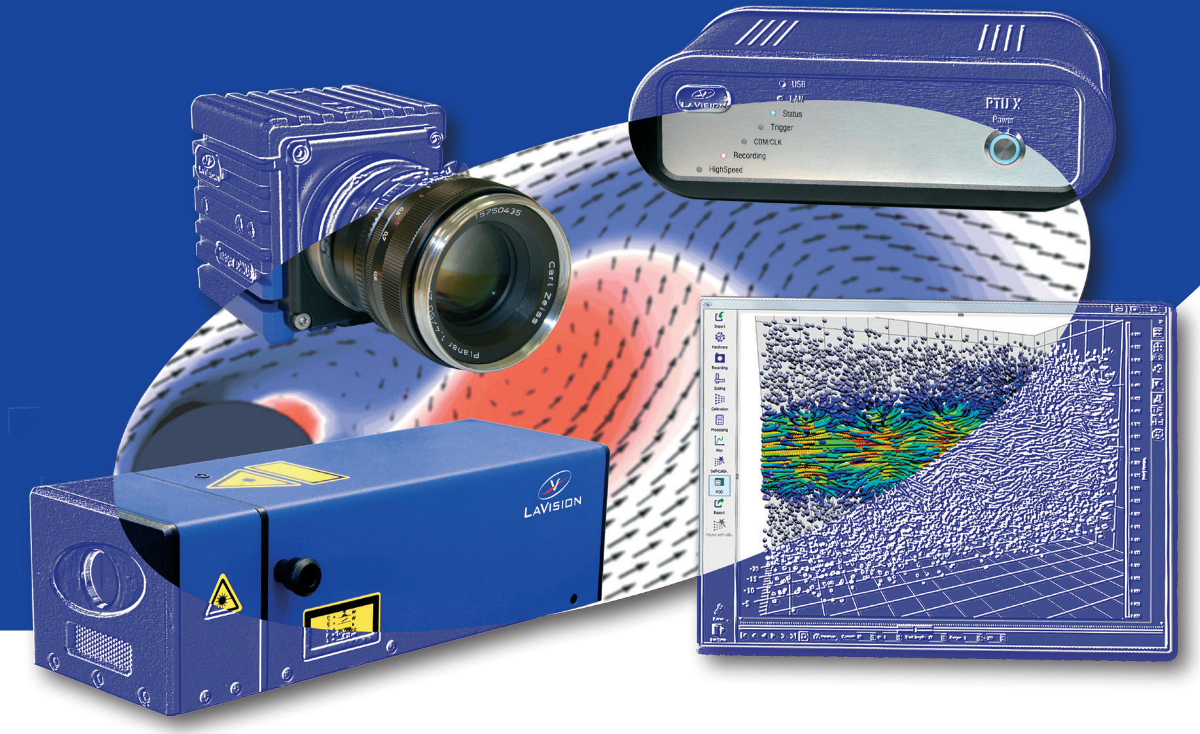
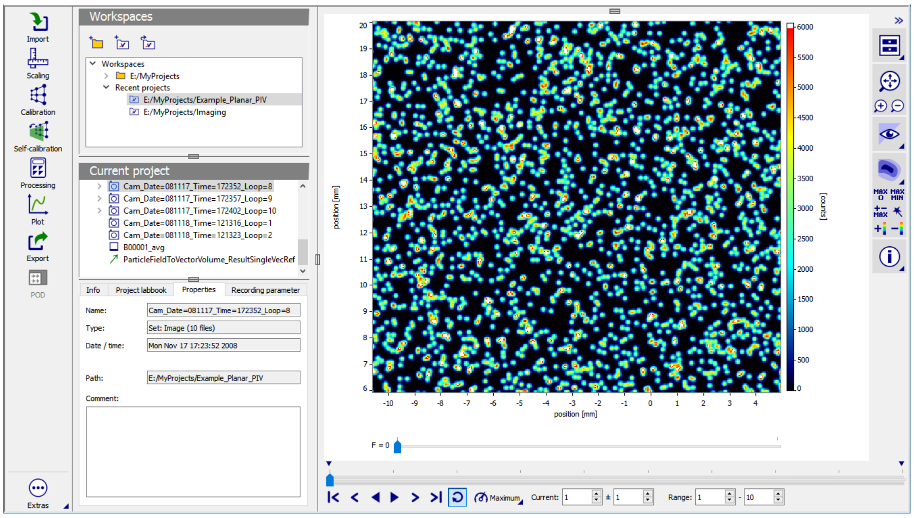


Product Manual

DaVis 10.2 Software

Item Number(s): 1105xxx



LaVision
FOCUS ON IMAGING

Product Manual for **DaVis 10.2**

LaVision GmbH, Anna-Vandenhoeck-Ring 19, D-37081 Göttingen
Produced by LaVision GmbH, Göttingen
Printed in Germany
Göttingen, November 2, 2022

Document name: 1003001_DaVis_D10.2.pdf

Contents

1 Welcome to the LaVision DaVis 10 Software

1.1 About DaVis 10

LaVision's **DaVis 10** defines a new landmark of cutting-edge innovations in Intelligent Laser Imaging software. Combining high-performance cameras with outstanding image processing algorithms opens new dimensions for your imaging applications. **DaVis 10** has been redesigned from the ground-up to deliver a software for modern demands for usability, data handling and visualization.

1.1.1 Leading-edge 4D imaging

DaVis 10 possesses LaVision's most innovative algorithms for 4D measurements of flow fields, 3D surfaces, sprays, mixing and combustion processes. High-speed cameras add the time domain as the 4th dimension to the 3 dimensions of volumetric imaging. For example, the innovative "Shake-the-Box" particle tracking algorithm utilizes the information gained from time-correlated high-speed measurements to optimize data quality and density. Our 12+ years of Tomographic PIV has led to an expansion to Tomographic Imaging for 3D/4D flame and spray characterization.

1.1.2 Contemporary software architecture

DaVis 10 has been redesigned from the ground-up to match the requirements of current and future imaging technology. A pure 64 bit structure, a fresh layout, and a responsive user interface improve the workflow and lead to a satisfying user experience. New data displays for visualizing 3D/4D data are powerful and user-friendly. The internal data format has been optimized to efficiently handle the huge amount of data and high transfer rates generated by state-of-the-art multi-camera systems.

DaVis 10 is built in an ISO-9001 certified agile SCRUM process from our highly motivated and experienced software development team.

1.1.3 Platform concept for multi-parameter imaging

DaVis 10 follows our successful strategy of a multi-parameter imaging platform from DaVis 7 and 8. It integrates various applications on a common user interface and aggregates all functionality in a single, optimized software.

Like previous versions of DaVis, *DaVis 10* is very flexible and can grow with your needs. The modern architecture ensures expandable system solutions and secures your investment for the long-term.

A huge variety of supported hardware and comprehensive data processing are combined on a common platform in **DaVis 10** and maximizes the technical capabilities of the system. As one example, the simultaneous measurement of flow fields (PIV) and surface deformation (DIC) with **DaVis 10** lead to a comprehensive investigation of fluid-structure-interaction (FSI) in a single integrated system.

1.1.4 Fully integrated hardware control

DaVis 10 offers flexible and scalable hardware support. LaVision works in close cooperation with several camera and laser manufacturers worldwide to be able to always offer our customers the latest imaging technology.

More than 100 camera and laser models are integrated into and controlled by DaVis 10. This high degree of integration offers an intuitive and seamless operation of the complete system, especially when combining different camera models and hardware components. Furthermore, DaVis 10 combined with our comprehensive Programmable Timing Unit (PTU X) delivers precise timing and hardware control with a clear and consistent user interface. The complexity of the system adapts to the measurement task: from a compact 2D imaging tool up to a complex multi-parameter and multi-camera Laser Imaging system, DaVis 10 offers field-proven working solutions.

1.1.5 Scientific and industrial applications

Innovations in **DaVis 10** are not only driven by demanding scientific and research applications, but from industrial applications as well. Based on the proven modular concept of DaVis 10, our innovative measurement technology can be successfully transferred and customized for industrial applications for quality control and testing.

1.1.6 Customization

DaVis is continuously being improved to fit the requirements and applications. Specific features for new customers can be incorporated into the software on request.

Please do not hesitate to contact LaVision! A team of experienced engineers, physicists and programmers will create a solution to fit your needs.

1.1.7 Customer-friendly update and support policy

LaVision believes that our customers should benefit from our constant innovation. We believe that our customer's success is also our success. Therefore, **DaVis 10** continues our long-standing philosophy of no cost updates and support. This customer friendly policy ensures that our customers have access to uncompromising support for the lifetime of the system.

1.2 Conventions

Throughout the manual certain types of information are printed in a special way or at special layout positions.

Courier

Anything which is a computer command (user input, macro programs, variable name etc.) or a file name is printed in Courier. E.g. the command to start **DaVis** is printed this way:

`DAVIS.EXE`

Capital letters

If certain keys be pressed on the keyboard, these keys appear in capital letters, e.g. ENTER denotes the Enter key (it is also called return key).

If two keys have to be pressed at the same time, this is denoted with a plus between the keys. For example,

CTRL+I

means that the user should press and hold the Control key CTRL (German keyboards: STRG) and press the key I at the same time.

Bold Letters

Bold letters are used for dialog names, button names or other text items in a dialog.

2 Installation

2.1 System Requirements

For the installation of **DaVis** 10, the following minimum specifications of the computer are required:

- A PC with 4 GB RAM, 2-GHz dual-core CPU or higher, operation system Windows 10 – 64 bit. Starting with DaVis 10.2 any language version of Windows 10 is supported.
- Windows 10 is required to start **DaVis**.
- All important Windows updates should be installed.
- The analysis version of **DaVis** is running on Windows 10 – 64 bit.
- Hardware support is available on Windows 10, 64 bit, only.
- Windows 10 N versions are not supported.
- 1,5 GB available space on hard disc is required for the software installation and 4 GB for optional example files.
- Display resolution should be at least 1280x1024 pixel. The software is optimized for 4K resolutions (for example 3840x2160 pixel) and for 1920x1080 pixel (FullHD).
- The user should either get administration privileges, or the **DaVis** directory and the data directory must be set up with complete access privileges for the users. Please ask your system administrator to define the correct privileges.
- Special settings may be needed for anti-virus software (see page 33) or a local firewall (see page **??**).

2.1.1 DaVis for Linux

The Linux version is not included into the standard installation program but can be downloaded from the **LaVision** website by registered customers.

Distributed Processing

The Distributed Processing feature of **DaVis** had been discontinued in the first version of **DaVis** 10 and has been released again in version 10.1.0. The installation guide can be found in manual **1003013** about **Distributed Processing**.

Cluster Script Processing

The Cluster Script Processing feature has been released with version 10.0.4 and is also available in version 8.4. The installation guide can be found in manual **1003013** about **Cluster Script Processing**.

Command Line Processing

The Command Line Processing allows to write own scripts to process a larger amount of projects and sets. The installation guide can be found in manual **1003013** about **Command Line Processing**.

2.2 Software Installation

Usually **DaVis** is already installed in the system delivered by **LaVision**. So, no further installation is necessary.

The manual installation of software and hardware is described in the following sections.

2.2.1 Operating System

Some hardware devices may be used on a special Windows version only and will not be supported on other Windows versions.

Whenever you want to change or upgrade your operating system, please contact **LaVision** before the installation to request information about support of your hardware and about new software versions. For cameras and other hardware devices, please contact **LaVision** to choose the appropriate main board. If **DaVis** detects an unknown operating system, it gives a message to the user and does not start.

2.2.2 Installation

When installing **DaVis** for the first time on a PC, you need to be logged in with administration privileges.

All installations can be completed only if a *dongle* is connected to your PC (see section on page 33) or if a network dongle is used. When using a DVD or the white USB flash drive as installation medium, please don't connect the local dongle to your PC before finishing the installation of **DaVis** software and all drivers.

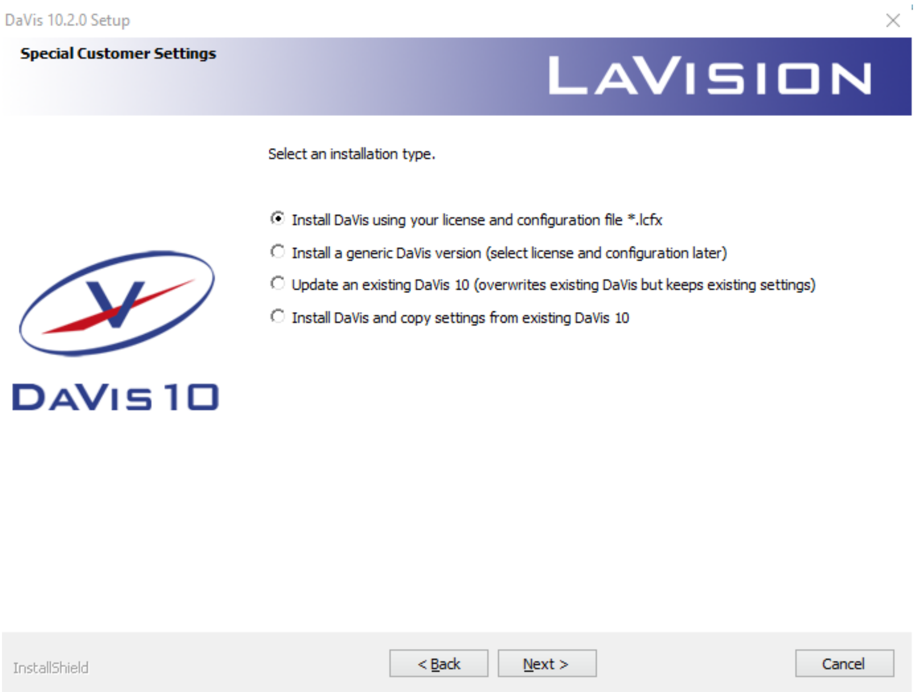
Please compare the **dongle identifier** from your dongle and your installation medium. Both numbers must be equal for the correct installation. In case of a black dongle with additional USB flash drive and delivered since year 2020 the flash drive is usually named with the dongle identifier, which can be seen in the Windows explorer. In case of a older delivery on DVD the dongle identifier is printed to the DVD and in case of white USB flash drives the flash drive is named with the dongle identifier.

For the installation of the **DaVis** please proceed as follows:

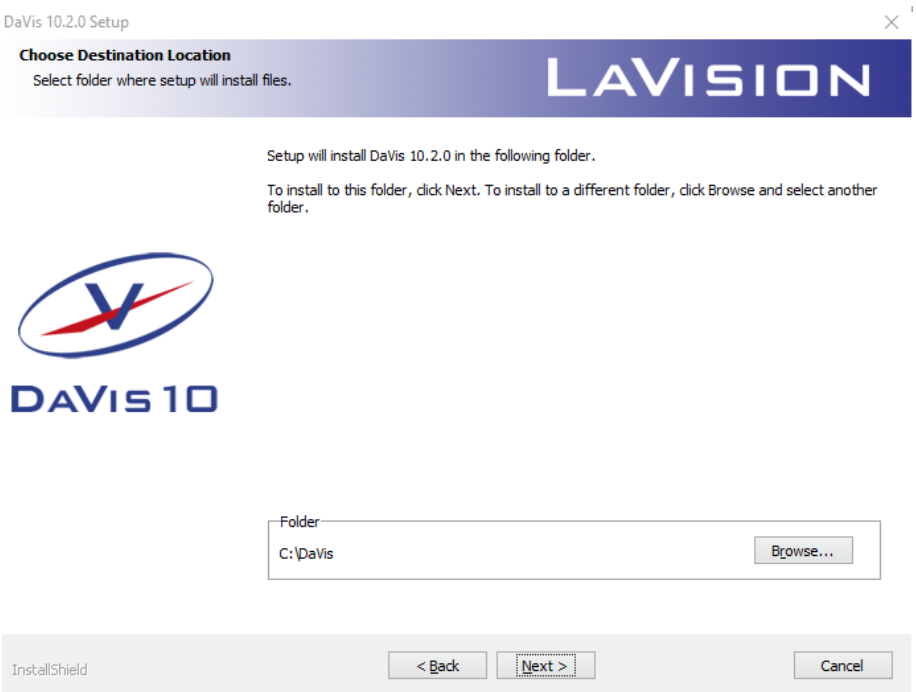
- Connect the **DaVis** installation medium to your PC and start the `SETUP.EXE` program from the new drive.
- Select the first option to **install using your license and configuration file**. This option uses the default settings from the configuration file `*.lcfx`, which is stored on the individual medium.

  The next option **install a generic version** is used for a simple installation without any individual configuration. In this case depending on the dongle (see chapter on page 33 below) the license is either detected automatically for a dongle with ten digits identifier or the license dialog opens (see page 62) and asks for a configuration file or a license key file during first start. A configuration file with e.g.

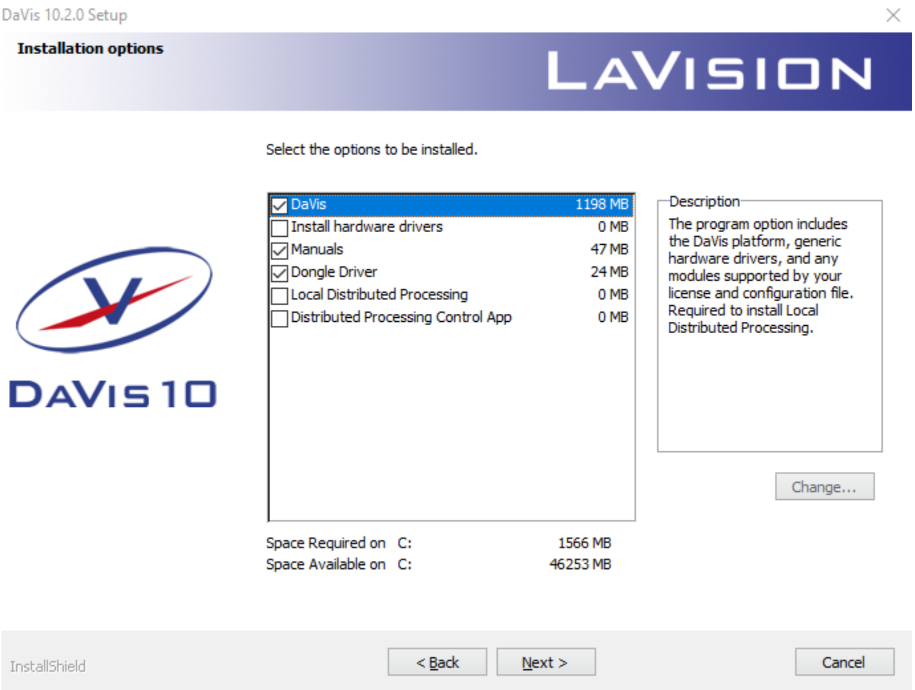
hardware settings can be installed at any time later, same for a copy of settings from another **DaVis** installation.



- Select a NEW directory for the installation. DO NOT OVERWRITE existing **DaVis** versions.



- Select the features: Program is needed anyway, the dongle driver is needed for new installations, manuals are optional, distributed processing is described in a certain manual.



- Wait for the end of installation. This may take some time in the section about additional drivers. Do not close small temporary installation dialogs in this section.
- Restart the PC. An evaluation version can be executed immediately after connecting the dongle.
- Start **DaVis** by double clicking the item on your desktop. This link starts **DaVis** in the binary subfolder.

2.3 Silent Install

Silent install, or unattended install, can be used to install DaVis on multiple Windows PCs with a software distribution system. Silent install is supported for generic installation or installation with a configuration file (`*.lcfx`). All other installation options are not supported.

To distribute the installation with a silent install a recording of an installation has to be created first. To do this start the setup like this:

```
SetupDaVis.exe /r /f1"C:/path/DaVis.iss"
```

The `/r` parameter starts the setup in recording mode. With the `/f1` parameter the file path for the recordings response file (`*.iss`) is specified. If the file path for the response file is not specified it will be created in the Windows system folder.

The response file can then be distributed with the setup to the other systems and used to perform the silent install. The silent install can be started like this:

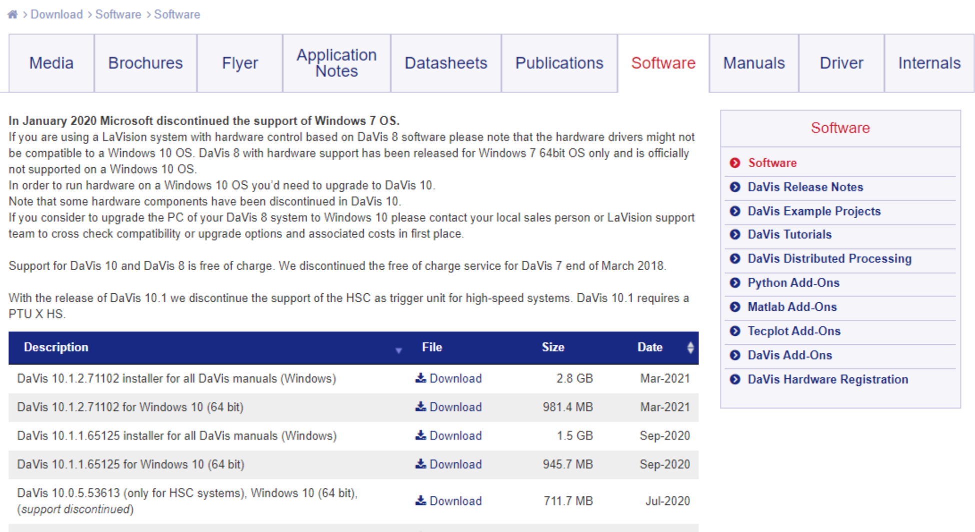
```
SetupDaVis.exe /s /f1"C:/path/DaVis.iss"
```

The `/s` parameter starts the setup in silent install mode. With the `/f1` parameter the response file is specified. A log file is created next to the `SetupDaVis.exe` during the silent install. Starting a silent install without a response file will not install DaVis.

2.4 Installation of an Update

LaVision provides the latest **DaVis** version in the download area of the `www.lavision.de` web site. Please note that for access to the download area you need to have a valid user account. In order to obtain this you need to register specifying your email address and order no. of your system or your dongle number respectively.

> **Note**
>
> Simple updates from DaVis 8 to DaVis 10 just by downloading the new version are not possible! A new license key is needed and settings can't be transfered.



> Download > Software > Software

Media | Brochures | Flyer | Application Notes | Datasheets | Publications | Software | Manuals | Driver | Internals

In January 2020 Microsoft discontinued the support of Windows 7 OS.
If you are using a LaVision system with hardware control based on DaVis 8 software please note that the hardware drivers might not be compatible to a Windows 10 OS. DaVis 8 with hardware support has been released for Windows 7 64bit OS only and is officially not supported on a Windows 10 OS.
In order to run hardware on a Windows 10 OS you'd need to upgrade to DaVis 10.
Note that some hardware components have been discontinued in DaVis 10.
If you consider to upgrade the PC of your DaVis 8 system to Windows 10 please contact your local sales person or LaVision support team to cross check compatibility or upgrade options and associated costs in first place.

Support for DaVis 10 and DaVis 8 is free of charge. We discontinued the free of charge service for DaVis 7 end of March 2018.

With the release of DaVis 10.1 we discontinue the support of the HSC as trigger unit for high-speed systems. DaVis 10.1 requires a PTU X HS.

Description	File	Size	Date
DaVis 10.1.2.71102 installer for all DaVis manuals (Windows)	Download	2.8 GB	Mar-2021
DaVis 10.1.2.71102 for Windows 10 (64 bit)	Download	981.4 MB	Mar-2021
DaVis 10.1.1.65125 installer for all DaVis manuals (Windows)	Download	1.5 GB	Sep-2020
DaVis 10.1.1.65125 for Windows 10 (64 bit)	Download	945.7 MB	Sep-2020
DaVis 10.0.5.53613 (only for HSC systems), Windows 10 (64 bit), *(support discontinued)*	Download	711.7 MB	Jul-2020

Software
- Software
- DaVis Release Notes
- DaVis Example Projects
- DaVis Tutorials
- DaVis Distributed Processing
- Python Add-Ons
- Matlab Add-Ons
- Tecplot Add-Ons
- DaVis Add-Ons
- DaVis Hardware Registration

⚠ Note

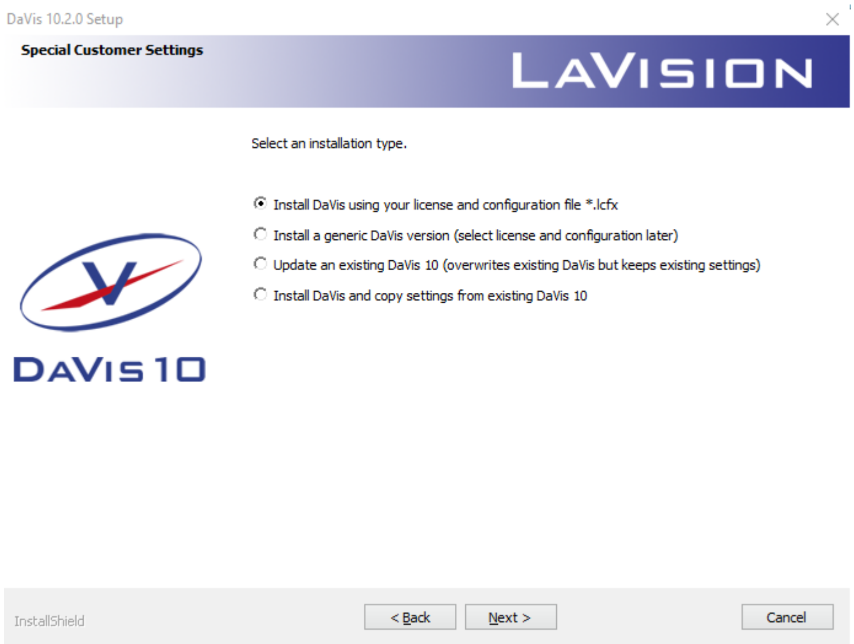
Do not delete the existing **DaVis** version. Keep this as a fall back until the new version has been installed and tested successfully.
Do not install or update any driver for the hardware components like the PTU since they are working properly in the old version! Updating the driver can be complicated. A corrupt driver will stop you from using the system in any **DaVis** version.

For the installation of **DaVis** please proceed as follows:

- Download the latest **DaVis** installer as exe-file from the download area of the **LaVision** web site and copy this into a temporary directory on the system PC.

- Start the **DaVis** install shield by a double-click on the exe-file.

  – Select the **Install using your license and configuration file *.lcfx** option in the install shield and refer to the config file. If you want to use the same configuration as in the initial installation of **DaVis** then use the config file from the `SETTINGS` directory of the original **DaVis** installation medium or contact **LaVision** by email to `service@lavision.com` to get a config file from our data base that is valid for the used dongle number.

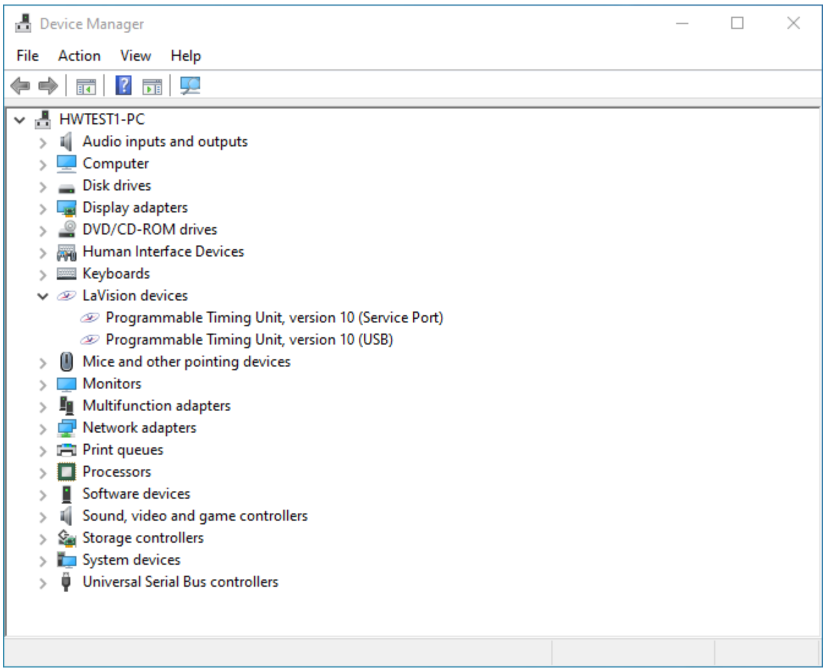


In the next step select a NEW directory for the installation. DO NOT OVERWRITE existing **DaVis** versions. Use the overwrite options instead, see next option below.

- The overwrite option replaces a already installed version of **DaVis** by the new version and copies all settings from the old version: general settings like the used license, the current hardware setup, customized acquisition items and saved operation lists. After selecting the option **Update an existing DaVis 10 (overwrites existing DaVis but keeps existing settings)** the old installation must be selected in the next step:

- Select **Install and copy settings from existing DaVis 10** to copy all settings but install in a new folder. On this way the old installations stays available and can be used as fall back or for comparison.

- DO NOT INSTALL any plug&play driver, even if the install shield will show corresponding instructions in pop-up dialogs. The install shield assumes that no driver are installed until now. In order to double-check that the drivers for the hardware components used are installed and working properly use the **Windows Device Manager**.



- Restart the PC and start **DaVis**.

2.5 Uninstall DaVis

All additional hardware drivers must be uninstalled with their own uninstalling programs!

For German-language versions of Windows: Click on **Start - Einstellungen - Systemsteuerung**. Doubleclick **Software**, select the entry **DaVis** in the software list and press button **Hinzufügen/Entfernen**.

For English-language versions of Windows: Click on **Start - Settings - Control Panel**. Doubleclick **Add/Remove Programs**, select the entry **DaVis** in the software list and press button **Add/Remove**.

Then delete the **DaVis** directory and the remaining backup files in the subdirectories.

2.6 Multiple Installations and Backup of DaVis

DaVis can be easily backed up and copied: Simply copy the entire folder of your **DaVis** installation to a new location. Besides the installation of the hardware drivers and the dongle driver, there is nothing extra needed. If you like to run a different hardware configuration on the same PC, simply copy an existing **DaVis** installation and run **DaVis** from this copy. This is also a good practice if you like to test different configurations or settings. **DaVis** writes settings back into its program folder and does not make use of the Windows registry. Even though this intentionally does not follow the Microsoft Windows standards, it gives you the freedom to copy **DaVis** as often as you like onto the same or to a different machine. In order to have a productive system backed up properly, we strongly recommend to use a full disk backup, like provided by Microsoft in the standard operating system or by 3*rd* party software.

2.7 Installing Hardware Devices

For some cameras or other hardware devices special installations are required. Special instructions can be found in the **Hardware** manuals. The following instructions are general and working for most devices:

When rebooting the PC the Windows device manager detects a new hardware device. The system asks the user to select the driver position. Please select the subdirectory `Drivers` on your **DaVis** medium and press button **Continue**. Now the device will be included in the device manager. Then **DaVis** is able to use the device.

2.8 LaVision Hardware Icons

Many LaVision devices are listed in the Windows Device Manager while they are connected. They can be identified by the familiar LaVision icon. With older DaVis installations, it might occur that these icons are not being displayed correctly. This can be fixed by installing the registration file found in the DaVis installation directory at
`ProgramData\Hardware\install-lavision-class-icon.reg`

Use a double-click to do so and confirm the two messages that you are presented with. Note that administrator privileges are required to perform this change.

2.9 Windows – Processor Groups on Systems with More than 64 Cores

A system with more than 64 cores (processor units) can not be used at 100 % of its processing capacity for **DaVis**. This is because Windows cannot use more than 64 cores for one application:

Windows will group the available cores into pools (*processor groups*) of identical size. It will increase the number of pools until the maximum number of cores per pool is smaller than or equal to 64.

For example, 68 cores will be grouped into 2 pools (of 34 cores each) as this is the smallest number of pools such that each group contains ≤ 64 cores.

An application running on Windows, such as **DaVis**, can access only one of these pools.

You will find more details on this on the Microsoft web site, e.g. using the following link: `https://msdn.microsoft.com/en-us/library/windows/desktop/dd405503(v=vs.85).aspx`

2.10 Remote Desktop and OpenGL 3D Display

What does it mean when **DaVis** starts with warning message: *Remote desktop connection detected. Sufficient OpenGL support might not be available over a RDP connection. For improved 3D support, try starting DaVis using the "DaVis remote desktop starter" shortcut.*

DaVis 10 uses modern OpenGL graphics technology for its 3D views and thus needs an OpenGL version of 3.3 or higher (see also "The graphics card you are using does not support OpenGL version 3.3 or higher, which is needed for 3D graphics in DaVis. All 3D views will be disabled."). Via a Windows remote desktop connection (using the Windows RDP protocol) this might not available to DaVis on some systems, although the host computer might have a suitable graphics card and driver and be technically able to show 3D graphics in DaVis. This is an inherent limitation of the Windows remote desktop system (especially on Windows 7 hosts), which DaVis cannot work around. Possible solutions for this error message:

Try starting DaVis on the host computer using the **DaVis remote desktop starter** shortcut. Note that administrator privileges are required to use this starter. The application will terminate your current remote desktop connection and restart DaVis on the host computer in the background. You can then reconnect to the remote computer. This can help to make 3D graphics available on the host computer.

The **DaVis remote desktop starter** can also be used for local dongles, which don't allow RD connections. See page 36 for more details.

2.11 Anti-Virus Software

To reach a faster storage speed, please configure the filetypes to be scanned by your anti-virus software: The **DaVis** filetypes `IM7`, `VC7`, `IMG`, `IMX` and `VEC` and other file types typically used by **LaVision** stream set format should not be scanned, because these files include image or vector data and no executable code. Filetype `CL` includes macro commands to be executed by the **DaVis** software and don't run as standalone software. Filetypes `CLS` and `XML` are used for configuration settings.

2.12 The Dongle

LaVision protects the usage of the **DaVis** software by a dongle. Several different hardware types of dongles are available and have been delivered over the years for DaVis 7, 8 and 10, see figure 2.1 for an image and some more descriptions.

Multi-License Network Dongle

If more than one or two PCs with **DaVis** are used, e.g. one system for the experiment in the laboratory and some more PCs for data processing in offices, and if local dongles should not be connected to all PCs, then a network dongle from **LaVision** can be used to replace all local dongles.

The network dongle is available for the USB interface.

The dongle server should be running the whole day, e.g. on a file server with Windows operating system. Network drivers for Linux and MacOS are

Figure 2.1: Blue USB dongle delivered up to version 8.2 or black HASP USB dongle, supported since version 8.2, for single license on a local PC. Green USB network dongle for license management from a server, delivered up to version 8.2. Not displayed in this picture is the HASP USB network dongle with red color, used since version 8.3 and needed since Windows 10 and Windows Server 2016, and the yellow network dongle used since version 10.0.5. Starting with year 2020 the local dongle and network dongle come in the same style like the former black dongle. The DaVis 10 dongle has the license stored on the dongle and has a big **10** printed next to the **LaVision** logo.

available from the supplier even for older dongle types. The current dongle driver for Linux systems is included in the archive of DaVis for Linux, see section 2.12.5 for installation instructions.

Search for more information about the **Sentinel HASP** key or for special drivers e.g. at

`https://sentinelcustomer.gemalto.com/sentineldownloads`

The green network dongle is delivered with a fixed number of licenses and upgrading the number of licenses always requires an exchange of the dongle. The red or yellow or black network dongle can be upgraded via license patch file at the customer's side.

Please contact **LaVision** sales department for a network dongle.

2.12.1 Installation of a Local Dongle

> **Note**: *During the installation from a data medium the dongle driver is automatically installed. Please do not connect the USB dongle to the system before finishing the installation!*

The dongle driver installation may take a little time without any further action in the installation program.

Manual Driver Installation

The setup program can be executed manually, for example if the dongle driver is downloaded from the **LaVision** homepage or if **DaVis** is copied to another PC without installation. For a downloaded driver version please change into your download folder. When using the installed **DaVis** then open the local Windows Explorer and change into **DaVis** subdirectory `Driver\Dongle`. Then execute `install.bat` to install the driver.

2.12.2 Device manager and services

Figure 2.2 shows the entries of the HASP dongle in the Windows device manager. Depending on the dongle type there can be up to three lines of **Sentinel Key** devices.

The installed HASP driver uses one service, see figure 2.3, the **Sentinel LDK License Manager**. This service must be running when a dongle is connected to the PC. In case of DaVis using a network dongle from another PC the service should run but must not. A network dongle can even be used without installed dongle driver.

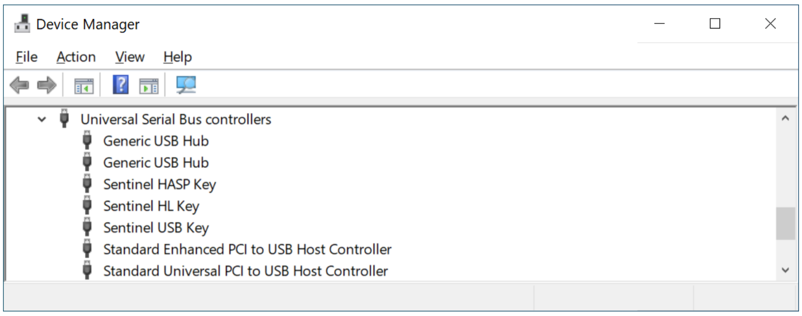


Figure 2.2: HASP dongle in the device manager.

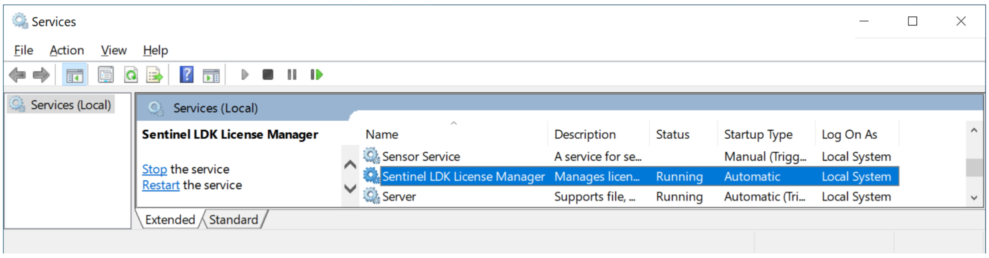


Figure 2.3: Sentinel LDK License Manager in the services list.

2.12.3 Dongle Driver Problems

If **DaVis** is missing the dongle driver, then repeat the installation and follow the above given instructions.

If the driver is installed and the dongle connected to the PC, but not detected, then the hardware configuration of the PC should be checked, e.g. if the USB port is activated.

USB Dongle

For usage of USB dongles the USB interface must be enabled.

If the USB dongle shows a flashing red light, then the dongle has a hardware fault and must be replaced. Please contact **LaVision** service department.

If the light of the dongle is off, then the dongle is not recognized by the Windows operating system. Try to reconnect the dongle or change the USB port. If nothing works, then reinstall the driver. In worst case please contact **LaVision** service department.

Old blue dongle and modern dongle driver

The old blue dongle is fully supported until driver version 8.15. The next versions until 8.31 don't support the old blue dongle on all PCs. The support is removed since driver version 8.41.

In case of an old blue dongle to be used for both **DaVis** 8 and **DaVis** 10 an older version of the driver must be installed and never be updated, neither via installation of a new **DaVis** version or by automatic Windows updates. The latest driver with blue dongle support, version 8.15 , is available in the download area of the **LaVision** website.

2.12.4 Remote Desktop

The usage of DaVis via Windows remote desktop (RDC) on another PC is possible with a new 10-digit **DaVis 10** dongle. There could be restrictions because of the OpenGL driver for 3D views in **DaVis**. During startup, **DaVis** checks the conditions of the graphical device and driver, and gives some information about possible problems. The software cannot be started with an old 5-digit **DaVis** dongle via remote desktop. This would require a decent remote tool. Please use the `DaVisRemoteDesktopStarter.exe` instead that is located in the `~DaVis\win64` directory.

DaVis 8 dongles and remote desktop

Using a **DaVis 8** dongle with a license upgrade to **DaVis 10** may restrict the start of **DaVis** via remote desktop (RDC)! The error message **Terminal Server detected** will be displayed at the startup. On those Windows operating systems **DaVis** is allowed to start on the real desktop only. There are some possible solutions for this problem:

1. Contact **LaVision** service department about a full upgrade of your dongle. In case of blue dongles and even black dongles with a 5-digit ID the dongle must be exchanged.
2. Do not use RDC but another remote desktop software like **TeamViewer** or **pcAnywhere** or **RAdmin**. Those programs able to control the real desktop, they are not opening a virtual desktop.
3. Use a network dongle connected to a dongle server somewhere in your network or even on the local PC. This avoids all problems with local dongles and works without installation of the dongle driver.

2.12.5 Installation of a Network Dongle (red and yellow and black version)

Since **DaVis 8.3.1** the red or yellow or black network dongle can be used. For installation on a server without **DaVis**, start the **DaVis** installer and deselect **DaVis** and the manuals, just select the dongle driver. When the installation is finished, the red or yellow or black dongle can be connected to the server. The server can even be a virtual server running e.g. on VMware as long as the host system allows the virtual machine to control USB devices.

To verify the installation and dongle connection open the following link and click on **Sentinel Keys**: `http://localhost:1947/`

The website is generated by the dongle driver and displays all connected dongles both local and network dongles, see figure 2.4 as example. Local dongles are displayed with a **Key Type** of **HASP HL Time** and network

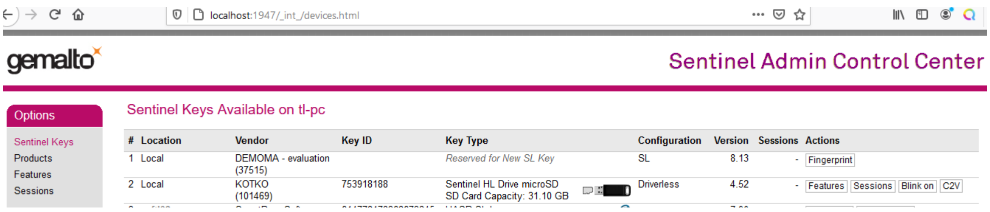


Figure 2.4: Driver website with list of detected dongles.

dongles by **HASP HL Net**. Software based licenses are displayed by a **Key Type** of **HASP SL**. The **Key ID** of the dongle is a physical unique device identifier and has nine or ten digits. For old style dongles **DaVis** uses its own five-digit number, which is printed on the dongle, but stores both numbers in the service file. The short id is usually requested by **LaVision** service department and can be given on the website to initiate a service request.

Define the IP address of the dongle server

If the dongle server is in the same network as all DaVis computers then there is no need to define the IP address of the dongle server. In case of different networks open the driver's website

`http://localhost:1947/_int_/config_to.html`

and enter the IP address of the dongle server in the box next to **Remote License Search Parameters**, see figure 2.5. Sometimes even the switch **Aggressive Search for Remote Licenses** must be used e.g. when the network dongle is located in another subnet. Press button **Submit** to activate the settings. More information about those settings can be found when pressing the **Help** button on bottom right of the website.

If these settings must be used on several computers, the configuration file can be copied from the first computer from the following location to the same folder on another PC: `C:/Program Files (x86)/Common Files/Aladdin Shared/HASP/hasplm.ini`

Installation on a Linux system

Most modern Linux distributions can be used as dongle server. Download the Linux version of **DaVis**, which includes the dongle driver, from the **LaVision** website or the latest driver from the manufacturer's website:

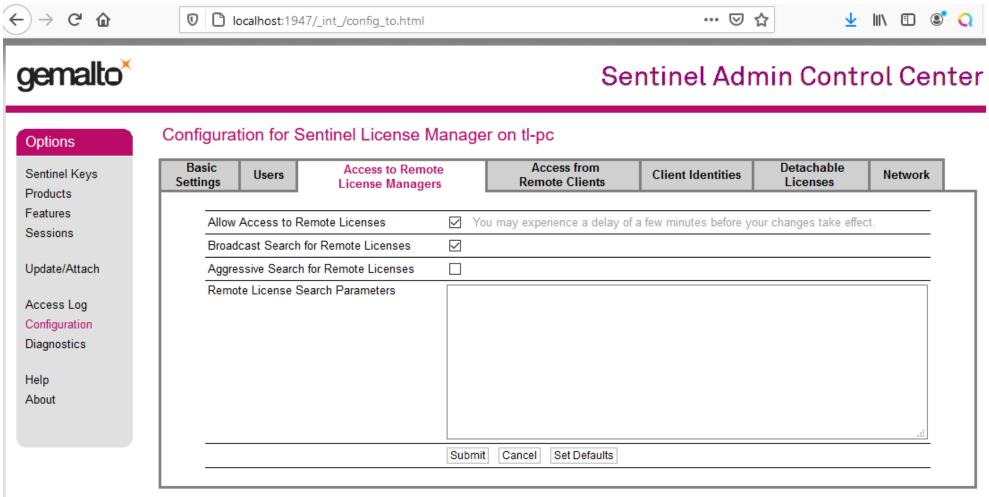


Figure 2.5: Driver website with configuration of network serahc for a dongle.

`https://sentinelcustomer.gemalto.com/sentineldownloads/`

After extracting the **DaVis** archive the driver can be installed by

```
sudo ./davis-setup.sh dongle
```

To verify the installation just start a web browser and open the driver's website `http://localhost:1947/`

Now a network dongle can be connected and should be visible on that website after a short time.

Firewall

Usually the installation of the dongle driver changes the firewall configuration and allows the **Sentinel License Manager** to access the local network (domain and private network). For manual configuration port 1947 must be allowed.

There is a way to change the port but this had to be done on all PCs using the network dongle. Stop the licensing service after installing the driver, then modify the registry to add a *Port* key with the desired new port number, and then restart the service:

```
sc stop hasplms
reg add HKLM\System\CurrentControlSet\Services\hasplms\Parameters
/v Port /t REG_SZ /d 1948
sc start hasplms
```

To restore the port setting to the default port 1947, delete the *Port* key from the registry:

```
sc stop hasplms
reg delete HKLM\System\CurrentControlSet\Services\hasplms\Parameters
/v Port
sc start hasplms
```

Verify your changes by browsing the Sentinel Admin Control Center interface at `http://localhost:<portnumber>/`

2.12.6 Installation of a Network Dongle (green version)

The green USB dongle has been used for **DaVis 7** and **8** and also can be used for **DaVis 10**. It must be connected to the USB port of a Windows PC. In most cases a file server running 24 hours per day is used. In principle each free PC could be used, even a PC running **DaVis**. The server PC should get a fixed IP address and must not use DHCP.

Network Dongle Server for Windows

Please put the **DaVis** installation medium into the destination (server) PC and change into folder `Drivers\Dongle`. Install the dongle driver by executing `install.bat`. Now change into folder `Drivers\AKS Monitor` or open subfolder `driver\AKS Monitor` of a installed DaVis version. Execute the installer files `hlsw32.exe` and afterwards `aksmon32_setup.exe`. The first one installs the dongle server `HL-Server` and the second one installs the network dongle monitor.

Start the monitor program, which can be found under **Programs – Aladdin – Monitor – AKS Monitor**. If the name of the server PC is not listed on the left, then install and start the **HL-service** via menu **Services – Hardlock**.

Please select the server PC in the tree on the the left side, see figures 2.6 and 2.7. Enter the **Module address** with value **11324** in the middle of the right area and press the **Add** button. Now some entries should be visible in the lines above the **Add** button. When starting DaVis on this or on another PC, the number in the current logins column will change.

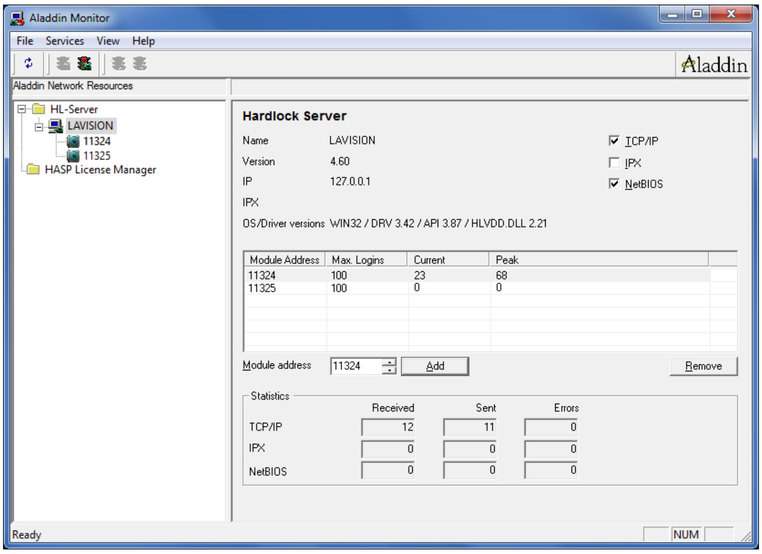


Figure 2.6: Network dongle monitor with dongle overview.

Two Network Dongles at same Server

If two network dongles of the old green type should be used on the same server, then the second dongle must be a special one. It is not possible to enable two *default* network dongles at the same server. This special dongle must be prepared by **LaVision**. Please contact **LaVision** sales department when such a setup is planned.

The **Module address** of this special second dongle is **11325**.

Firewall

When using a network dongle, the firewall of the server PC must be configured to accept communications between the network dongle server and the local host. The server's firewall must accept connections to the **DaVis** hosts. Local **DaVis** and dongle server are communicating on UDP port 3047.

The firewall asks during the first start of **DaVis**, if the software should be blocked. Please answer with **never blocking**. When running the **dongle server** on a system with firewall, the administrator has to add the communication port as exception: Select the second card of the firewall setup dialog, the press button **Port** and type as name e.g. `hlserver`, as port 3047 and select UDP. After pressing the **OK** button and closing the firewall setup dialog the **AKS Monitor** will locate the dongle server's PC.

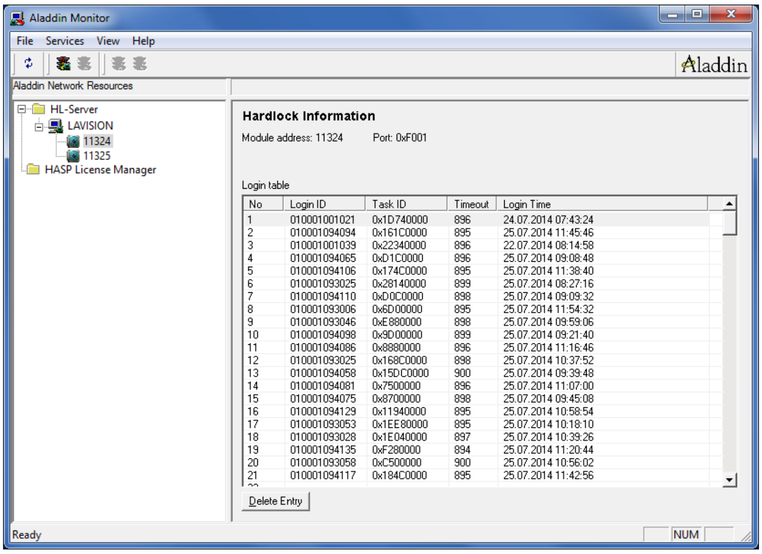


Figure 2.7: Network dongle monitor with connection list of all license users.

Connection Timeout

The dongle server uses a default timeout of 15 minutes = 900 seconds for each dongle. **DaVis** regularly asks the dongle server for a valid license. If the server is not asked for a certain time, then the server can give the license to another **DaVis** in the network. When **DaVis** works on a time consuming processing, there might be a long time without request for a license. Or when **DaVis** is stopped irregularly, the server gets no command to free the license.

The timeout can be changed by command line only. Please search for the folder of program `hls32cmd`. This should be copied to hard disc during installation of the dongle server. The default folder is `C:/Program Files/HL-Server`. Then open a command shell (Windows Start - Run) and execute e.g. to set a timeout of 15 minutes:

```
"C:/Program Files/HL-Server/hls32cmd" -timeout 15
```

Two or More Network Devices

On PCs with more than one network device, e.g. a second ethernet device or a WLAN device, the IP address of the network dongle must be entered in the **License Manager** dialog, see page 62. By default the dongle driver searches on the first network interface, which is often the WLAN device.

2.12.7 Installing local and network dongle

Sometimes a local dongle is connected and also a network dongle is available in the local network. In this case the detection of the right dongle might be a problem: If **DaVis** is not configured to use a certain dongle but searches for available dongles and then finds the network dongle first but the user wants to use the license from the local dongle.

In this case please make sure that the dongle identifier is entered in the configuration of **DaVis**. Either send the settings to **LaVision** service department or edit a settings file with a text editor. Open a folder in your **DaVis** installation: `Users/<name>/CL/Licence.cls` For easier access with a text editor this file can be renamed to `Licence.txt` and after changing the settings renamed back to the original name.

The head of the file should look like this:

```
DongleAddr = 4002;
DongleAddrIp = "";
DonglePorts = "";
DongleCreationDate = 20224;
DongleIdentifier = "1574665240";
```

Please replace the number in the line with `DongleIdentifier`) by the long ID or your local dongle. This ID can be copied from the driver's website: `http://localhost:1947/_int_/devices.html` Afterwards store the file and don't forget to rename to the original name. Then start **DaVis**.

2.12.8 USB dongles on virtual machine host system or locked dongles

When using a virtual machine as server for the dongle then the connection from this VM to the USB port may be difficult. In those systems a VM might not be allowed to use the USB ports of the host machine. Also locking a dongle is useful in some environments when nobody should be allowed to remove the dongle from a PC.

For those systems external devices with network connection are available mapping USB ports to any PC or VM in the local network. Some examples are:
`https://www.seh-technology.com/us/products/usb-dongle-servers/myutn-800.html`

and
https://www.digi.com/support/knowledge-base/anywhereusb-usb-device-compatibility-list

2.12.9 Error Messages During Startup

If **DaVis** detects a problem with the dongle, then this message is given very early in the startup procedure. In case of a driver problem the dialog 2.8 is displayed and the dongle driver has to be installed again.

Figure 2.8: Problem with dongle driver during startup.

Another error appears, if the last used dongle could not be detected, or if the detected dongle does not fit to the installed license key. This opens the **Dongle License Manager** dialog, which is described on page 62.

2.12.10 How DaVis Remembers the Dongle Type

When **DaVis** is opened for the first time, it searches for the connected dongle. At first, the USB port are asked for a dongle. By default **DaVis** is not searching for network dongles!

When **DaVis** is closed, it stores the type of the connected dongle in some static variables in the file `License.cls` in the CL folder. For more information about variables in general please read the *DaVis Command Language Manual*. The names of the variable storing the dongle type is `DongleAddr`. All these variable values can be changed easily with a text editor.

By default **DaVis** starts to search for a local dongle and then checks the network for a network dongle. To increase the searching speed, the variable `DongleAddrIp` should be set to the IP address of the dongle server.

Each dongle gets a unique number, which is stored in the dongle and read out during the startup of **DaVis** and then stored in variables `DongleCreationDate` for the five digits identifier and `DongleIdentifier` for the HASP dongle identifier. This numbers are written on the dongle below the **LaVision** logo and on the installation medium.

The available packages are enabled by the license code in variable `DongleCode`. When a customer gets a new project package, the dongle must not be changed, but a new license code is delivered. A description of the way to transfer the new license code into the **DaVis** software is given in section 3.4.2 on page 65.

Defining the port for dongle search

String `DonglePorts` variable in settings file `Licence.cls` can be used to define the ports, where **DaVis** should search for the dongle. By default, if the string is empty, **DaVis** searches at first on the USB ports and then in the local network. To change the string, please close **DaVis**, then edit the file with a text editor, save the changed file and start **DaVis** again. This may be needed in rare cases on strange errors e.g. if no USB port is available on the PC mainboard.

2.12.11 Uninstall the Dongle Driver

Disconnect all dongles from the PC first, then restart the PC and do not start **DaVis**. Use the standard uninstallation functions of your Windows operating system. In normal cases there is no need to uninstall the driver.

3 Startup

3.1 Startup Window

After starting **DaVis**, the **Startup** window (see figure 3.1) appears on the screen. At first **DaVis** reads some important macro files, then checks for a dongle, reads other (customer dependant) macro files and initializes the hardware. A progress bar gives information about the way to go. To stop the startup you can press the left mouse button somewhere on the startup screen.

If the dongle is not detected and an error message is displayed, please follow the instructions on page 33. If the license key does not correspond to the connected dongle, please read the section on page 65.

During startup **DaVis** tries to detect if another **DaVis** is already running. It is allowed to run several **DaVis** instances from different installations at the same time. But **DaVis** complains and does not start if the same installation (from the same folder) is running by the same user. Means different users on a Windows terminal server are able to run the same installed version of **DaVis**. This is possible because all local settings are stored in a user dependent folder.

At the end of the startup procedure, the **DaVis** main window is displayed and fills the complete screen.

Figure 3.1: The startup window of **DaVis**.

3.2 Screen Layout and Extras Menu

When the program is started, the **DaVis** task window is set up (see figure 3.2). This figure is an example for an Imaging Project, but the main items (menu, tool and status bar, dialogs) are visible in every package.

The top line of the task window includes the program's name "*DaVis 10*" on the left and the typical Windows title bar buttons on the very right: minimize, maximize or close the program. This close item in the title bar is the only way to close and exit DaVis!

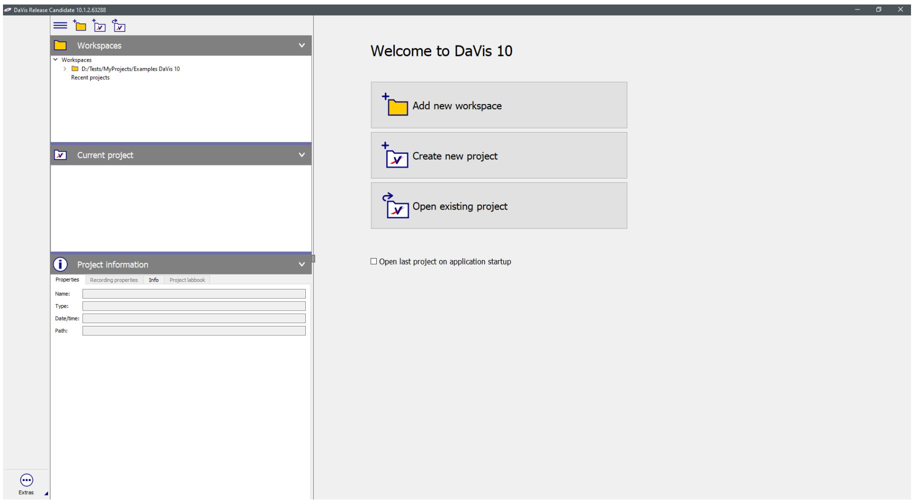


Figure 3.2: Layout of DaVis with project manager but without open project.

Toolbar

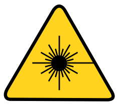
The **Toolbar** on the left side provides **buttons** which allow execution of the most important functions with a single mouse click. When moving the mouse cursor over the toolbar buttons, a short description about the button function is displayed in the **Status Text**.

The largest part of the **DaVis** task window is the **work area** in the middle. Dependent on the layout style either the complete area is filled by a single **Project Manager** dialog or by a special function dialog like **Recording** or **Processing**.

Tool Tips

For the first help many of dialogs give so-called **Tool Tips** whenever the mouse rests for a short time on a dialog item: A small window appears on screen, gives a description about the item, and disappears when the mouse is moved again.

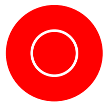
3.2.1 Toolbar Stop Laser



Stop Laser stops immediately all trigger signals and automatically switches the light sources to off.

Please note that this button does not replace any emergency stop switch or other safety devices!

3.2.2 Toolbar Stop Devices



Stop Devices stops immediately all devices, such as light sources and moving devices, and automatically switches them off. The trigger signal will be abruptly interrupted by the software. It is recommended to reinitialize the hardware setup.

Please note that this button does not replace any emergency stop switch or other safety devices!

3.2.3 Toolbar Extras - Service

Create service file for LaVision Support

Whenever a problem occurs and the support at **LaVision** asks for the settings of **DaVis**, e.g. to check the parameters of a dialog or some hardware settings, this menu item should be selected to create a copy of all parameters (static variables) of **DaVis**. This `LSFX` file should be send by email to **LaVision**.

Load service file from LaVision Support

Use this menu item to load an `LSFX` file from the **LaVision** support. After selecting the `LSFX` file **DaVis** restarts automatically to activate the new settings.

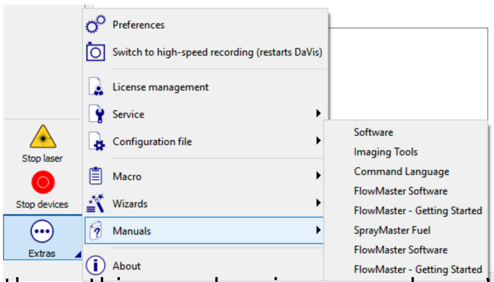
3.2.4 Toolbar Extras - Macro

This menu gives access to macro functions and macro logging. See detailed descriptions in the **Command Language** manual.

3.2.5 Toolbar Extras - Wizards

Some useful wizards are available in the **Extras** menu. See detailed descriptions in the chapter on page 413.

3.2.6 Toolbar Extras - Manuals



The available manuals for **DaVis** are available in PDF format and equal the printed manuals. The latest Acrobat Reader® should be installed to view the files. If another PDF viewer is installed and registered as default PDF reader, than this reader is opened. When selecting a **Help on** item in the **Extras - Manuals** menu then **DaVis** starts the Acrobat Reader with the selected help file and opens the contents page. The general manual about the **DaVis Software** and the programmer's **Command Language** manual are always available here. Depending on the license key some other manuals may be added to this list, e.g. the **PIV FlowMaster** manual.

Help about this Window

Whenever questions arise about a certain dialog or window, pressing the **F1** key immediately supports the user with the corresponding manual. **DaVis** searches for the manual, the PDF file, in the `help` subdirectory and opens the file either on the contents page or in the chapter or section, which describes the active dialog.

If a dialog includes different subdialogs, which are described in different parts of the manual, then a chapter selection box appears and asks the user to select the preferred chapter (see figure 3.3).

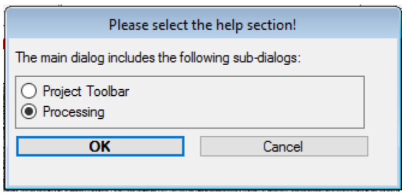


Figure 3.3: Select manual section for help on active window.

3.2.7 Toolbar Extras - About

On top of the **About** dialog the software version is displayed. In the lower area of this dialog all licensed packages are listed. The full **DaVis** version number has four parts, the 3-part version number given in large font on top plus the **Version ID** given in the left column.

Figure 3.4: About **DaVis**

The dongle ID had been displayed in this **About** dialog until version 10.1.2 but move to the **License Management** dialog in version 10.2.0. Press **Open license dialog** to change into the new dialog (see page 62) and indicate the dongle ID and the order number whenever asking **LaVision** for support (see p. 427).

Press the link `http://www.lavision.de` to start your preferred web browser and open the **LaVision** homepage. Press the link **E-Mail** to start your mail program and open a new message to **LaVision** service. Press the button **OK** to close this dialog. Two extra buttons open dialogs and display **Patent information** and **Third party license information**.

3.3 Preferences

The preferences dialog is opened from toolbar item **Extras** and gives access to a number of global system settings. The settings are presented in different sections on the left side (fig. **??**). Some of these settings are available for certain license keys only and in this case are described in the corresponding manuals.

3.3.1 General

Startup behavior

In the **General** category of the preferences automatic steps of the **DaVis** startup 1 can be customized.

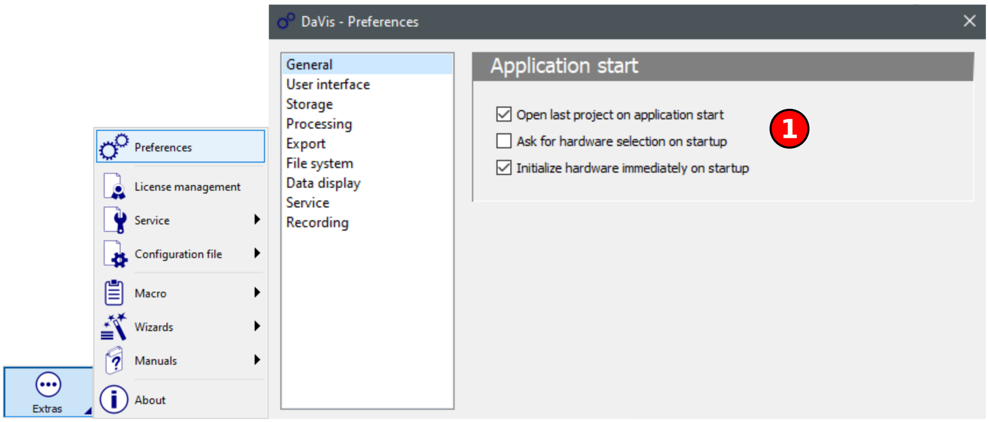


Figure 3.5: Preferences - Application start

Open last project on application start: The project used at the time **DaVis** was closed, is opened automatically at the next startup.

Ask for hardware selection on startup: If different hardware profiles are used **DaVis** can be configured to ask the on startup to select a hardware profile. This is helpful in situations where different configurations or different sub systems are in use on the same computer.

Initialize hardware immediately on startup: Initializes the hardware during startup or postpones the initialization until the hardware is used for the first time (e.g. when entering the recording). This can be useful when the hardware is not used every time **DaVis** is used (e.g. if it is started for data visualization or processing only).

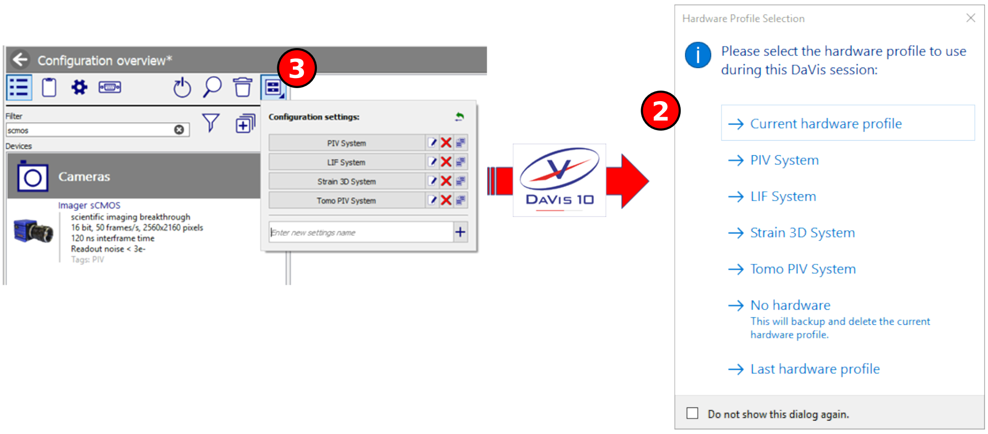


Figure 3.6: Hardware profile selection

If the **Ask for hardware selection on startup** option is checked, **DaVis** will pop up a list of available hardware profiles 2 during startup. These different profiles are those defined in the hardware shelf. To add, remove or modify profiles use the shelf button 3 in the hardware setup.

Additionally to the shelf items **DaVis** offers the extra option *Current hardware profile* which represents the profile the last **DaVis** session. The option *No hardware* backs up and clears the current hardware configuration. This configuration backup is available as *Last hardware profile backup* option at the next startup.

3.3.2 User interface

Define the **font**, **font size**, and **color** for display of scales in image or vector views and in the dashboard, as well **keyboard shortcuts**. Moreover, select the **position of info label visibility**.

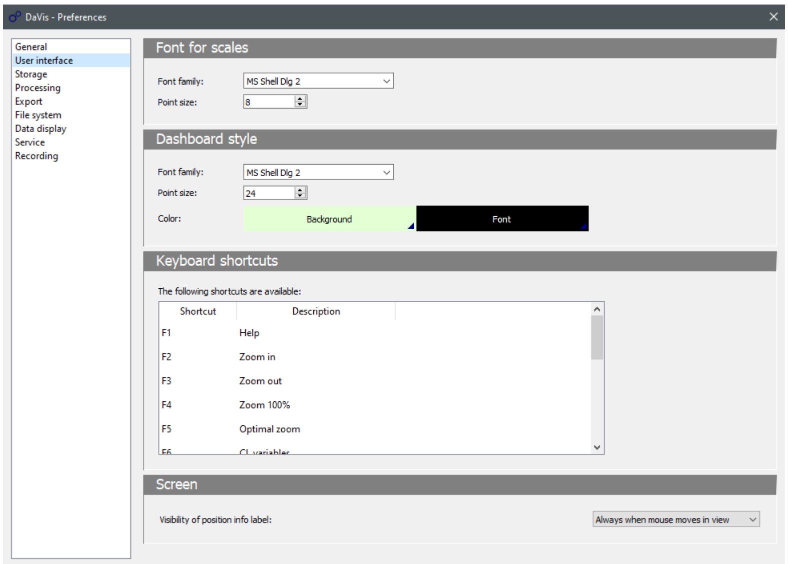


Figure 3.7: Preferences dialog with user interface settings.

3.3.3 Storage

The first section, see figure 3.8, gives information about the location and folders with the variable settings (**User data path** with e.g. license key and display settings) and with fixed settings (**Program data path**, stored during installation).

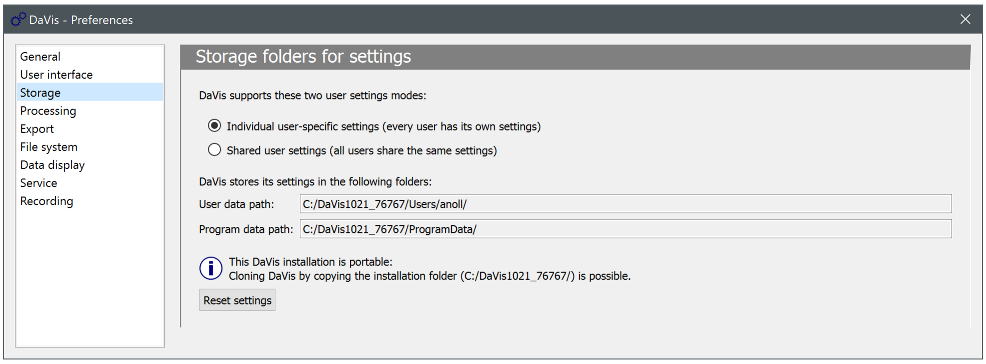


Figure 3.8: Preferences dialog with storage settings.

DaVis supports two user setting modes:

- **Individual user-specific settings**: Every user has its own settings. The user account is linked with the MS Windows user account. This means that if another MS Windows user is logged into the system, the user will have its own **DaVis** settings separated from all other users. This mode is useful if you want to separate work of multiple users. Settings are stored under `Users/<name>` with `<name>` as the user name of the active MS Windows account.

- **Shared user settings**: All users share the same settings. This means that **DaVis** uses the same settings for all MS Windows users. If user A changes settings, user B will see these changes. This mode is useful if you have multiple users that do the same tasks with the DaVis software. Settings are stored under Users/All users.

The active mode can be seen by the activated radio button. You can change the mode by clicking the other mode's radio button. A dialog will guide you through the process of switching between the two modes:

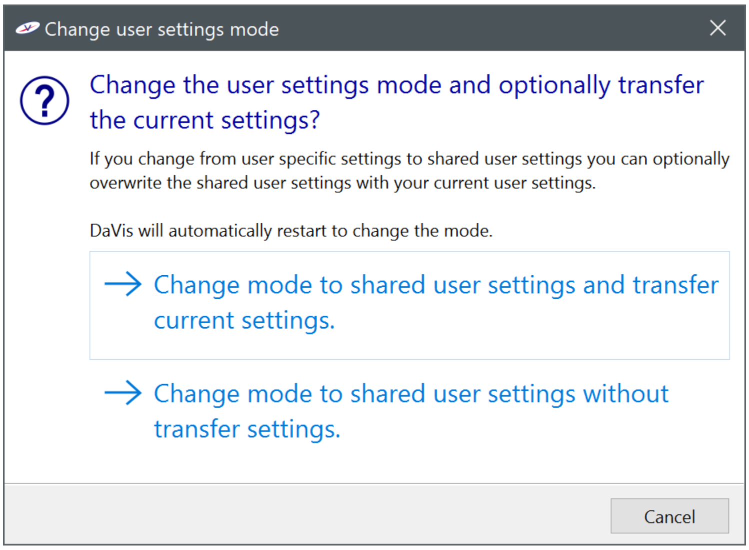


Figure 3.9: Dialog shown when changing the user settings mode.

When switching the mode, the user can define if the active settings shall be transferred to the other new mode or not. This makes is easy to switch to the other mode without loosing your settings if required. **DaVis** will perform a restart if the user settings mode is changed.

Settings folder, portable and non-portable installation

All variable settings of the software are stored in the **DaVis** subfolder `Users` with different settings for the different Windows users. With this **portable mode** the user can copy the settings folder to any other PC into a **DaVis** installation of the same software version and use the same (copied) settings.

If a (empty) file of name `useAppData` exists in the binary folder next to the **DaVis** executable, then the so called **non-portable mode** is enabled: the **DaVis** folders are not touched and can be used in read only mode. In this case all settings are stored in folder `"%APPDATA%/LaVision GmbH/DaVis/10.x.y/`

3.3.4 Processing

Storage

The processing dialog stores newly created image and vector results in `IM7` and `VC7` files which have configurable compression settings.

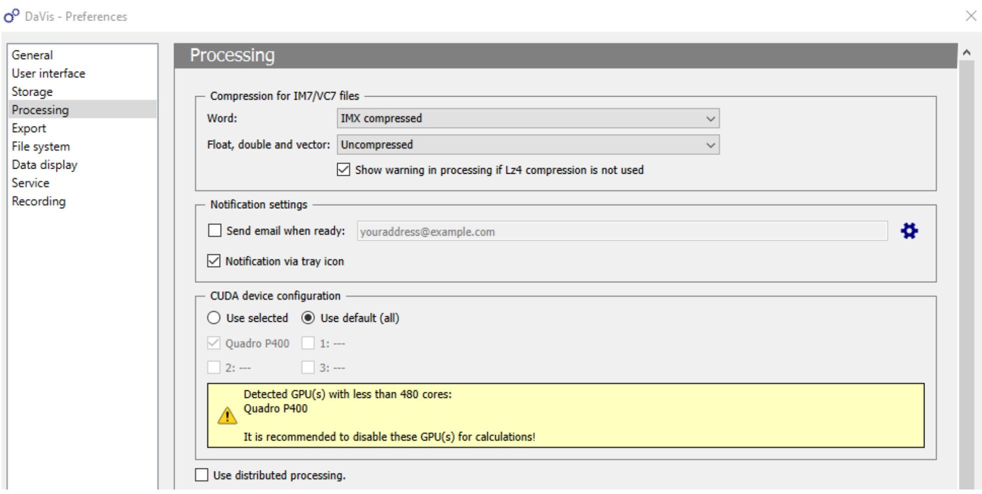


Figure 3.10: Preferences dialog with processing settings.

The compression modes can be used to change the size of the resulting files but also may change the writing speed. Details about the different modes are given in the section above and in table 3.1. Changing the storage mode is possible in the top part of the **Preferences - Processing** dialog of figure 3.10 and is only applicable to the processing functionality.

DaVis supports a number of different compression algorithms for the file formats `IM7`,`VC7`. More information on the file formats are given on page 138.

Different types of compression (see table 3.1) can be selected for 16 bit (word) buffers and for floating point buffers (float, double and vectors).

Mode	Description
Uncompressed	Fast but large files and almost no compression time
IMX	Simple and fast compression for WORD buffers
Zlib	Slow but smallest file size, working for WORD and FLOAT buffers
LZ4	Fast for WORD/FLOAT images and vectordata. Utilizes the full power of the hardware and is the fastest compression mode supplied by **DaVis**. Yields small file sizes. Offers the best compromise between performance and file sizes compared to all other compression modes available on **DaVis**. Available since **DaVis** 10.2. Previous **DaVis** versions are not able to open data sets stored with the lz4 compression.

Table 3.1: Compression modes for file storage

Notification settings

Sometimes a function needs a long time to work and you do not want to wait in front of the PC or visit the PC from time to time until the function finishes. Therefore e.g. the **Processing** and **Hyperloop Processing** dialogs are sending notifications by email if this option is enabled in the middle part of the **Preferences - Processing** dialog of figure 3.10. The setup (see figure 3.11) asks for a valid email address, the IP address of a mail server and the port to send mails (by default 25 for SMTP). Press the button **Send test mail** to validate the settings.

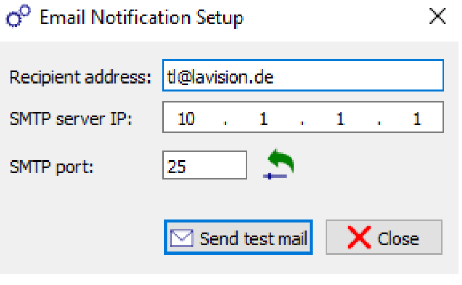


Figure 3.11: Setup email and server address for email notifications.

CUDA device configuration

Select CUDA devices for processing in the middle part of the **Preferences - Processing** dialog of figure 3.10. Mode **Use default (all)** uses all detected CUDA graphics devices. Mode **Use selected** uses the selected devices only. In this mode **DaVis** searches for device changes during startup, gives a message to the user and switches to the default mode.

Distributed Processing

Details about **Distributed Processing** are given in manual **1003013_DistributedProcessing_D10.1.pdf**. All settings are defined in the bottom part of the **Preferences - Processing** dialog of figure 3.10.

3.3.5 Export

For the export of data via clipboard and for the storage of buffer information in TXT files **DaVis** uses free values for the **decimal point character** (comma or dot) and the **separator** (tabulator, space or carriage return character). The **Floating point precision for CSV export** can be changed between 0 (no digits, like integer) and 15 (even very small values are exported in standard format and not in exponential format). The default value is 6 digits. The default value for **Indexed file digits** is 4 digits.

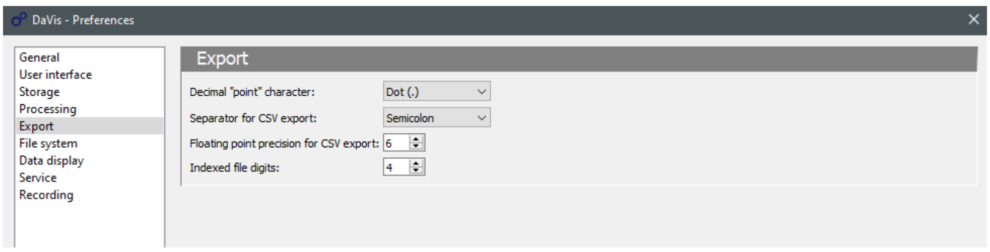


Figure 3.12: Preferences dialog with export settings.

3.3.6 File system

Before executing a macro, **DaVis** optionally checks if a CL-macro file has changed and asks the user, if the changed file should be reloaded. This is used by CL developers and avoids restarts of the software. Three modes about **check for changed CL files...** are available: By default the CL/Autostart folder is checked only. The checking can be switched off (**Never**) or processed for **all CL files** loaded so far.

By default, a dialog appears, if a changed file has been detected and asks the user if the changed files should be loaded. Deselect this mode and load all changed CL files automatically via checkbox **load updated CL files without asking**.

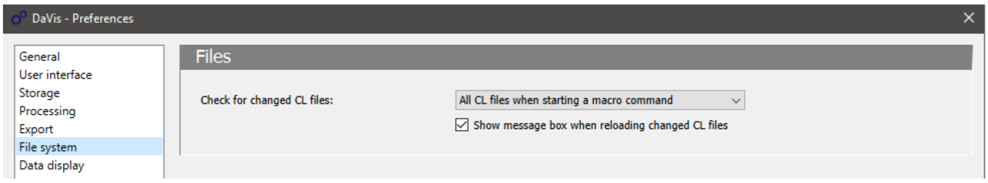


Figure 3.13: Preferences dialog with file system settings.

3.3.7 Data display

The **Data display** category of the **DaVis - Preferences** dialog provides information about the graphics card and OpenGL version. Chose an **Anti-aliasing** multiplier step between off, 2x, 4x, 8x, and 16x. **Transparency** of 3D graphics can be enabled by setting a checkmark.

Figure 3.14: Preferences dialog with data display settings.

Usually error bars show the **uncertainty** of a value (1x standard deviation). In some communities other values like 2x standard deviation or 3x standard deviation might be more appropriate (corresponds to 95% or 99.7% confidence interval). This selection affects **all** error bars in all plots.

3.3.8 Service

Select a **Visibility level** between User, Service, and Developer. If Service is selected, production tools can be loaded. In case Developer is chosen, **locally provided tools** can be additionally selected via checkbox.

Figure 3.15: Preferences dialog with service settings.

3.3.9 Recording

The **Recording** category of the **DaVis - Preferences** dialog contains four groups: *Hardware*, *Timing*, *Recording memory*, and *Stream format*.

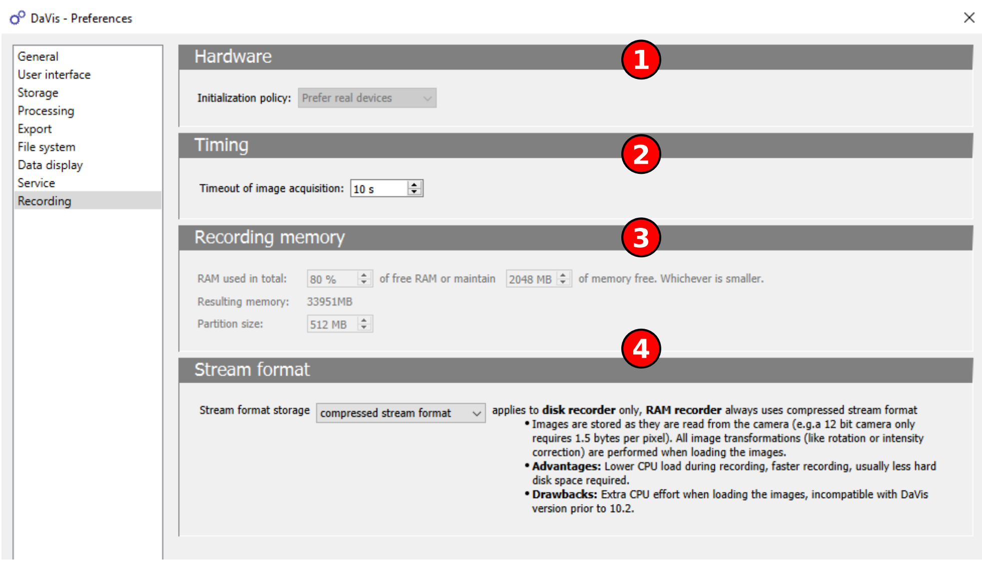


Figure 3.16: Preferences - Hardware and Recording

1. **Hardware:** The current initialization policy is shown. This option is only active in service mode (requires an extended license). It controls how **DaVis** searches for hardware. The default is to search for real devices and fall back to simulations if no real device was found. Other options are to either only use real devices or explicitly use sim-

ulations.

2 **Timing:** The image acquisition timeout can be set. This time is applied when **DaVis** waits for an image after a trigger has been issued or a trigger should have been supplied from an external source.

3 **Recording memory:** The amount of computer memory (RAM) can be specified which is reserved for cameras when the recording dialog is entered. This pre-allocated memory allows starting a recording with very short latency. In RAM recording mode this memory is used by all streaming cameras as their virtual camera RAM. In disk recording mode it is used as a buffer between the cameras and the storage medium (e.g. RAID system) to ensure the a constant data transfer rate.

The size to reserve can be set as percentage of the free RAM. For systems with less amount of memory a second parameter sets a minimum size which is at least maintained free for the application itself and the operating system.

The *Partition size* specifies the size of each allocated memory block. Smaller blocks might increase allocation speed but also increase the number of unused bytes at the partition boundaries. The default of 256 MB works fine in most cases.

Note: It is recommended to use not more than 80% of free RAM to retain enough memory for the application itself. Since the reserved recording memory is only used by the cameras, additional memory is required for some metadata and e.g. AD converters.

Note: In rare cases if the camera frames are set to a very small size (e.g. 32x32 pixels), the recording memory might be set to smaller values since the original assumption that the images need the vast majority of the memory becomes invalid.

4 **Stream format:** can be chosen between **compressed stream format** and **uncompressed stream format**.

- **compressed stream format**: applies only to disk recorder, since RAM recorder always uses the compressed stream format. The images are stored as they are read from the camera (e.g. a

12 bit camera only requires 1.5 bytes per pixel). All image transformations, for instance rotation ore intensity correction are performed when loading the images.
Advantages: Lower CPU load during recording, faster recording, usually less hard disk space required.
Disadvantages: Extra CPU effort when loading the images, incompatible with **DaVis** version prior to 10.2.

– **uncompressed stream format** :applies only to disk recorder, since RAM recorder always uses the compressed stream format. The images are expanded to 16-bit images before storage without increasing dynamic range. All image transformations, for instance rotation ore intensity correction are performed before storage.
Advantages: Faster loading of images, compatible to all **DaVis** versions.
Disadvantages: Higher CPU load during recording, slower recording.

3.4 License Management Dialog

With the help of the **License Management** dialog of figure 3.17 a new license key can be loaded and problems of the dongle detection can be solved.

The list of enabled software packages and modules is displayed in the **Licenses** tab for general features like hardware or evaluation support and in the **Modules** tab for the different modules of **DaVis**. In case of network dongles the columns **Licenses used** and **License limit** give information about available licenses for possible use from other PCs. For licenses with expiration date the date is given for each module.

The available features and modules are defined by the connected dongle and the license key. See chapter on page 33 for more information about dongles.

The lower block in the dialog gives all information needed by the service to identify the system: The customer name and order number, the full version information and the identifiers of the last used dongle and of the detected dongle.

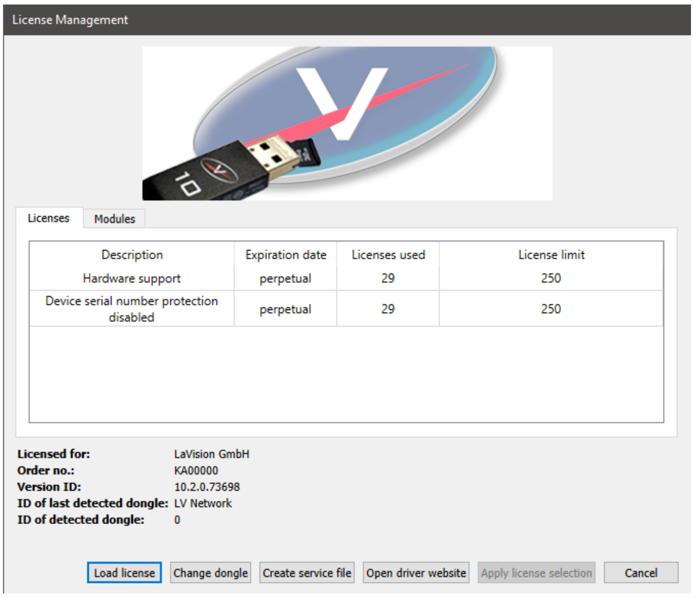


Figure 3.17: License management.

When pressing the **Load license** button a fileselectbox opens and asks for the location of the new license file either of type `LCFX` (default: LaVision Customer File) or `SET` (for simple licenses without lots of other software settings) or `V2C` for upgrades of **DaVis** 10 dongles. Only the license information and no other settings are read from the selected file. Then **DaVis** restarts automatically.

With the button **Change dongle** the type of dongle can be changed between local dongle and network dongle and between the old dongle types (blue and green) and the new types (black for local and red for network dongles). This is useful when several dongles are connected to the PC or available in the network.

With the button **Create service file** collects all information from the **DaVis** installation and finally opens the Windows explorer with the folder of the created `LSFX` file. This file can be sent to the **LaVision** service department in case of service requests. See the chapter about **Customer Service** on page 427 for more information.

If a network dongle of the green type is used, then it can either be detected automatically or by a defined IP address. The IP address must be given if the PC has more than one network connection, e.g. wireless and wired networks at the same time, because the driver searches on the first network device only.

If two network dongles of the green type are connected to the dongle server via **AKS Monitor**, then **Network Dongle 2** must be selected in dialog 3.18 to connect to the second one. Otherwise **DaVis** is searching for the first (default) network dongle only.

If there are more than one black local dongles connected to the local PC or other black or red network dongles available in the network, the **Available Hasp keys** list can be used to switch from automatic connection to a certain dongle.

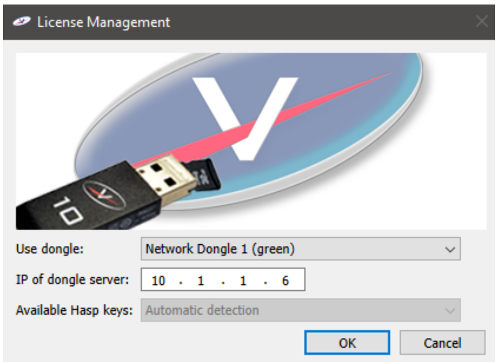
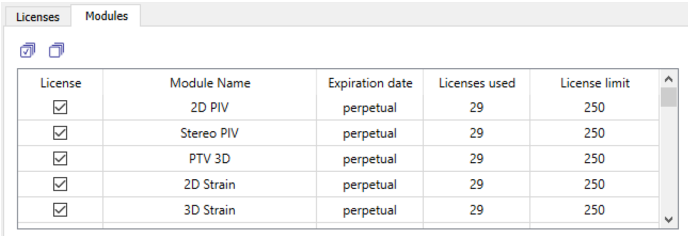


Figure 3.18: License management: Define IP address of network dongle server.

3.4.1 Activate or deactivate certain modules

This function is available for all licenses but has special use for network dongles with a different number of licenses for different modules, see figure 3.19. As example a network dongle could give three PIV licenses but only one Tomographic PIV license. By default **DaVis** reserves all available licenses from the configured dongle. This means the first PC starting **DaVis** would get the Tomographic PIV license. If one PC should be chosen as the Tomo PC, the Tomo module must be activated by **DaVis** on this PC but deactivated by all other **DaVis** installations on other PCs.

When changing any activation state, most buttons at the bottom of the dialog are disabled, only **Activate license selection** becomes available and the **Cancel** button allows to return to the former settings. With the button **Activate license selection DaVis** will store the new activation states and restart. After restart the deactivated modules are given at the end of the module list.



License	Module Name	Expiration date	Licenses used	License limit
☑	2D PIV	perpetual	29	250
☑	Stereo PIV	perpetual	29	250
☑	PTV 3D	perpetual	29	250
☑	2D Strain	perpetual	29	250
☑	3D Strain	perpetual	29	250

Figure 3.19: Module management: Activate or deactivate a certain module from the full license.

3.4.2 Upgrade License Key

In some cases a change of the license key is needed:

1. Installation of a extended license when upgrading a package (application) without re-installing the complete software.
2. Exchange of the dongle, e.g. after a hardware failure of the old device. The new dongle has another unique ID and needs a new license key.
3. Switching the same software installation between different packages with different license keys. We would prefer a second installation of **DaVis** instead.

During a complete software installation the key is automatically loaded. During a software update or when the dongle is connected to another computer, the license key can be loaded manually.

Note: Every dongle includes a unique serial number. Every license key includes the same serial number as the connected dongle. So a license key can be used only in combination with one dongle. It is not possible to use a license key with another dongle.

3.4.3 Loading a New License Key

The license file is delivered by **LaVision** support via email. At first store the `LCFX` or `SET` or `V2C` (means: Vendor To Customer) file somewhere on hard disc. Then start DaVis and select toolbar item **Extras – License Management** to open the **License Management** dialog. Section 3.4 above describes the way to load the new license.

In case of **DaVis 10** dongles with the license on the dongle itself and not in a license key file the upgrade procedure is very similar to the procedure of old style keys. Those old style keys are used when the same dongle gives the license both for **DaVis 8** and **DaVis 10** or when a **DaVis** 8 dongle gets a license upgrade to be used for **DaVis 10**.

3.4.4 Manual license upgrade of a red/yellow/black network dongle

In case of a red/yellow/black network dongle the upgrade with the `v2c` file can be done at the dongle server without **DaVis** installation. Login via Remote Desktop to the dongle server first. Then open the driver's website, see figure 3.20:

`http://localhost:1947/`

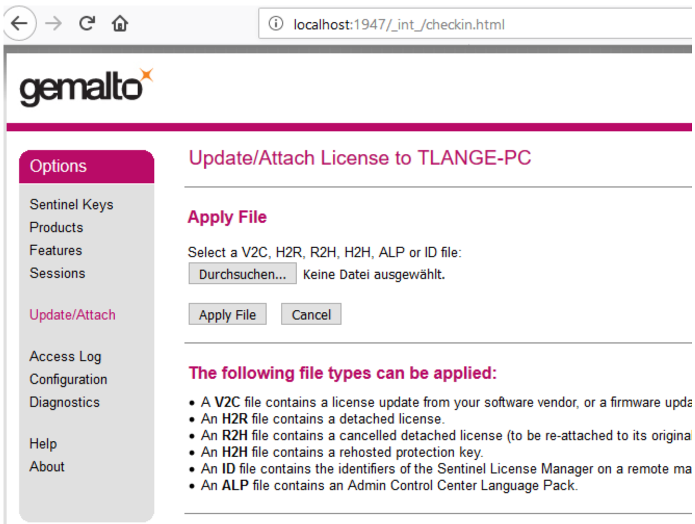


Figure 3.20: Upgrade the license via driver's website.

Select item **Update/Attach**, then **search** for the `v2c` file and press button **Apply File**. The driver will switch to the updated features of the dongle or display a error message. In case of an error please contact the **LaVision** service department.

3.4.5 Error on Wrong License Key

In case of a dongle hardware problem or an invalid license for the connected and detected dongle the dialog of figure 3.21 is displayed with some information about the problem.

The license key and the dongle itself include the unique dongle number. This is checked during startup and must be equal.

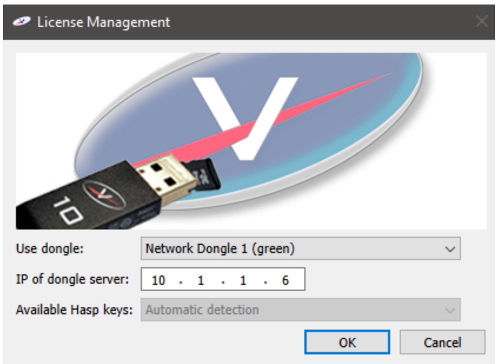


Figure 3.21: License management: Error description during startup.

4 Projects

4.1 Project Concept

The **DaVis** project concept supports the user in execution, administration and evaluation of data. Therefore, all system parameters, user annotations and experimental results will be stored in a hierarchically file structure. Depending on the software package the project provides a number of dialogs for acquisition and evaluation of data.

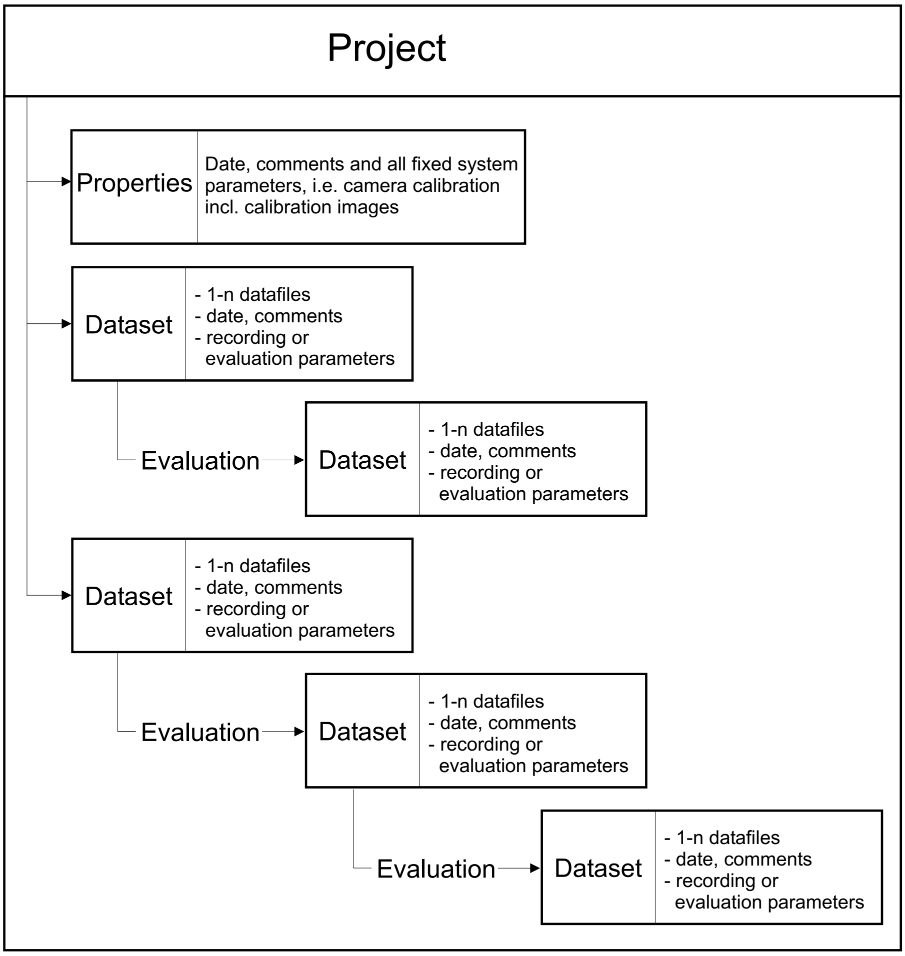


Figure 4.1: The project concept

4.1.1 About Projects

After opening a project the treeview of the manager shows the data within this project only. All system parameters, user annotations and experimental results are stored in a hierarchical file structure as outlined in figure 4.2. This example shows a project `Example PIV Planar` which includes some recordings (`Cam_Date_Time_...`). The datasets of the calculated velocity fields `PIV_Vec` and a postprocessed vector field `PostProc` are stored in the respective subfolders.

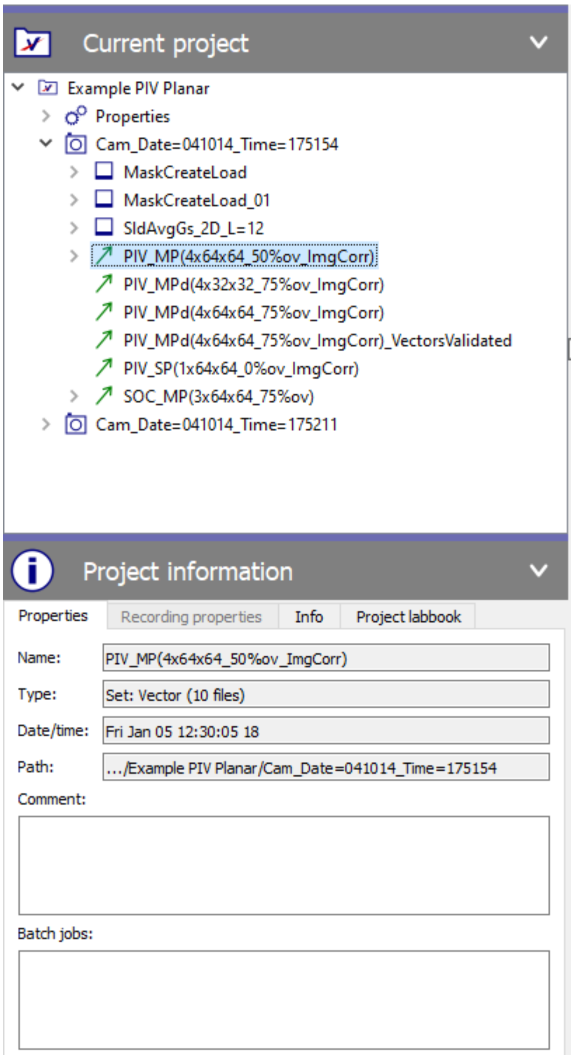


Figure 4.2: Project example

Most of the dialogs are directly accessible in the project manager by clicking the project name or a data set with the right mouse button. Depending on the type of dataset (image set, vector set, properties, . . .) different options are available in the context menu. Clicking on the project name provides operations to import data, make new recordings or (re)calibrate the cameras. Clicking on data sets gives access to image processing or evaluation routines. The data type (e.g. images, vectors, profiles) restricts the available processing functions, e.g. RGB image operations are not available for intensity images.

4.2 About Workspaces

Workspaces are locations on the harddiscs and network drives to manage and analyze your data. Like the *Favorites* in the Windows© *File Explorer*, each workspace is linked to a data folder on a disk drive of the computer.

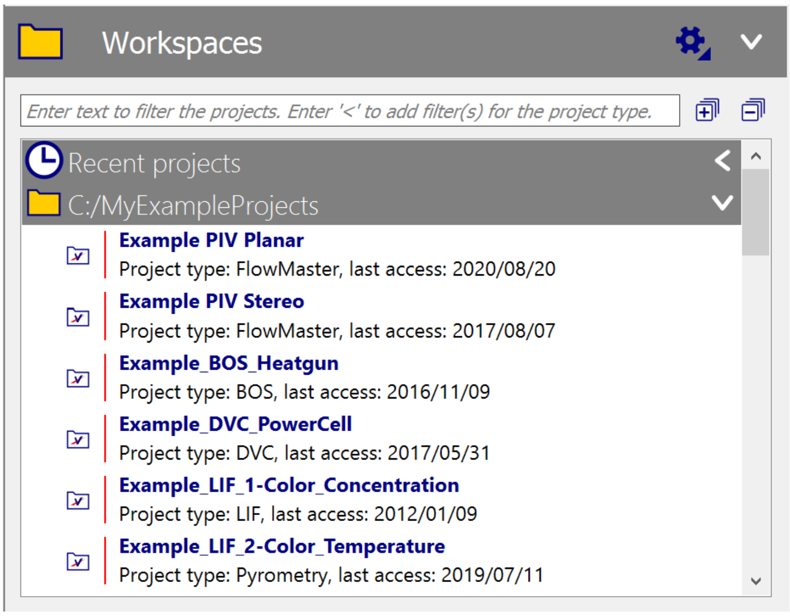


Figure 4.3: The **Workspaces** window

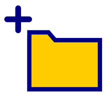
The Workspaces window is situated at the upper left of the screen. Its width and height can be adjusted using the two splitters below and to the right. It shows the projects within each workspace and a list of recently used projects. For each project the name, project type and last access is displayed. You open a project by clicking on it.

Its width and height can be adjusted using the two splitters below and to the right.

To structure your data, you can create as many workspaces as you like. For example, you might want to create one workspace

- per collaborator, with every user having direct access to their data,
- per type of sample or material or per method used,
- per measurement session,
- . . .

The **Workspaces** window contains a toolbar with three options:



Add workspace: Adds a new workspace in the root folder of the project manager. The folder must physically exist on your hard disk to be selectable.

Open project: Opens a previously created project (data set). The project does not have to be part of a workspace and can be stored on remote locations such as a USB drive.

Create new project in current workspace: Creates a new empty project in the currently active workspace.

Below the toolbar, all workspaces and included projects are shown as a directory tree.

✓ To collapse/minimize the display of a workspace, click on the section header that displays its name.

✓ Right-clicking on a workspace name opens a menu in which you can:

- Assign an alias name to the workspace - this avoids the display of the (long) directory path.
- Create a new project in this workspace.
- Remove this workspace from the list

  ! Note: This does not delete any data from the hard drive!

✓ Right-clicking on a project name opens a menu which allows you to:

- Copy a project.
- Move a project to another directory. This physically moves your data on the disk drive!
- Remove a project. This physically removes the data from the disk drive!

The section **Recent projects** offers quick access to the projects last edited. The number of projects displayed therein can be set to 5, 10, or 20 after right-clicking on the section header.

You can filter the list of displayed projects by entering a filter criterion in the textbox located at the top of the **Workspaces** section. You can enter single words in that textbox which are applied instantly as a filter. Additionally, you can filter on a specific project type by entering an ’<’ which then opens a selection of project types. If you select a project type only projects of that type will be displayed.

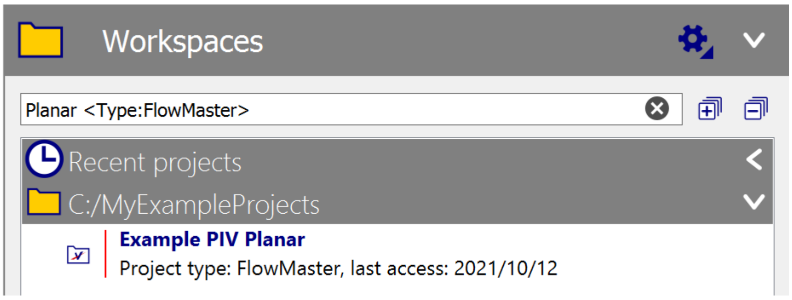


Figure 4.4: Filtering **Workspaces**.

! With the very first use of **DaVis**, no workspace is defined yet and no project is loaded.

✓ Click on the **Add new workspace** button in the **Welcome to DaVis** dialog (Fig. 4.5) to define an existing folder on your disk drive as workspace for future data acquisition.

✓ Click on **Open existing project** (Fig. 4.5) if you have an existing **DaVis** 8 or **DaVis** 10 project.

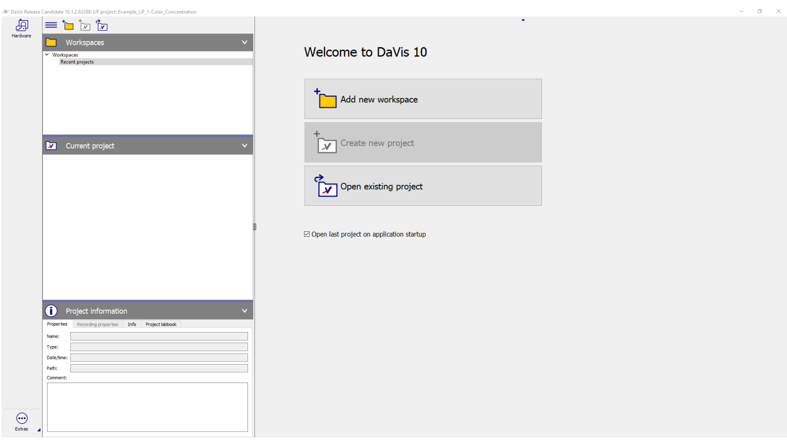


Figure 4.5: The **Workspaces** dialog with welcome screen at first start of **DaVis**.

If you install a new DaVis version on a computer on which you already have used former DaVis versions the **Import Workspaces** dialog will be opened on the very first startup of DaVis. It allows you to quickly import formerly used workspaces into your new DaVis installation (requires DaVis 10.2.1 or later). The **Import Workspaces** dialog is documented in the next chapter.

4.2.1 Import Workspaces

You are able to import workspaces from previously used DaVis installations by using the Import Workspaces dialog. The dialog can be accessed from the cogwheel menu:

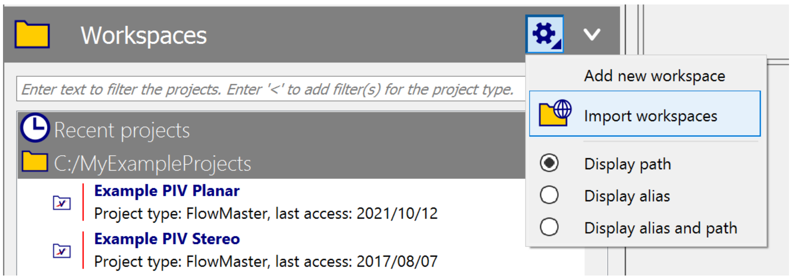


Figure 4.6: Open the **Import Workspaces** dialog.

The dialog shows a list of workspaces that were used on the computer before. You can import workspaces to the current DaVis installation by ticking their checkboxes and clicking the **Import workspaces** button.

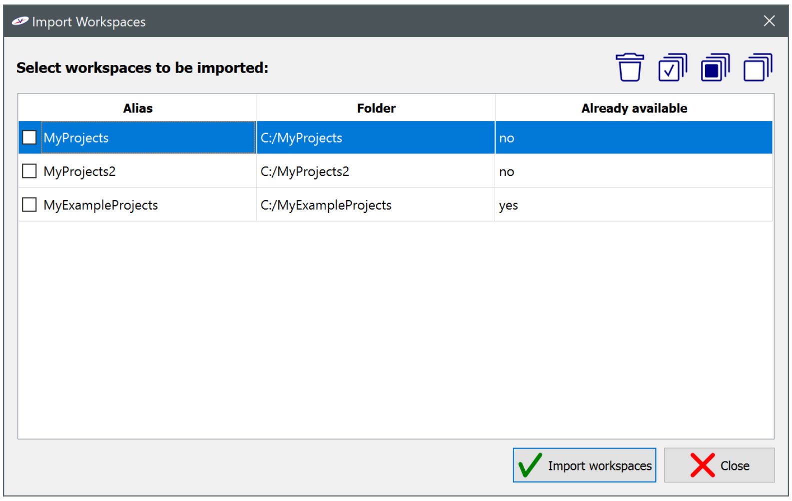


Figure 4.7: The **Import Workspaces** dialog.

With the buttons in the top right area of the dialog you can un-/check multiple workspaces with a single click. The trash can button can be used to remove ticked workspaces from the list (the workspace in your filesystem is not touched).

4.3 Project Manager Dialog

The **Project Manager** dialog appears after logging into **DaVis**. It is the central dialog where data acquisition, data evaluation and data storage is organized.

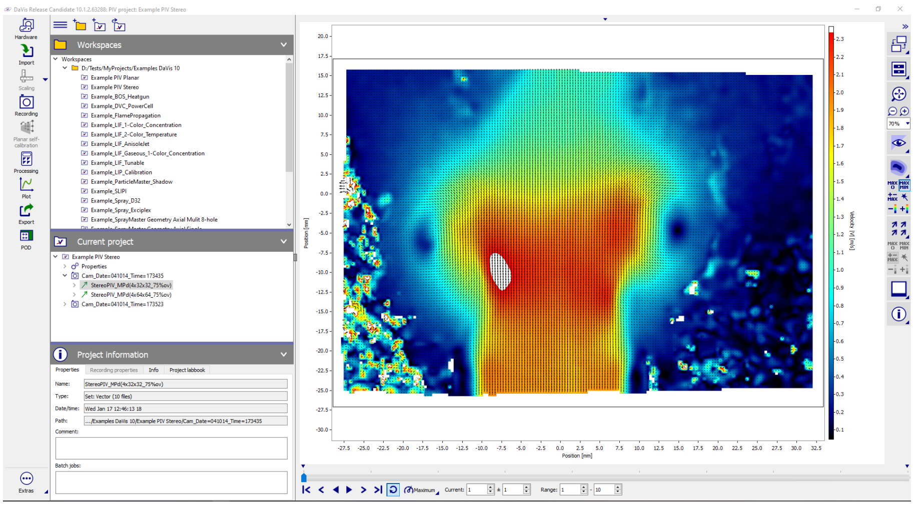


Figure 4.8: The Project Manager dialog after logging in

The dialog is divided into several parts:

- The toolbar on the left gives access to new dialogs for different purposes, e.g. hardware setup or processing. Some of these toolbar items are available for certain set types only.
- The trees in the left area present the workspaces and the current project.
- The data viewer in the right part shows a image, vector field, plot or special dialogs of the selected set.
- The subdialogs in the lower left part give some additional information about the previewed file or the current project:

  Info: General information from the last actions (processing, recording,...).

Project labbook: This labbook is available after opening a project. It can be used to store additional information about the current project (e.g. hardware environment). All important events (e.g. image calibration) are stored automatically.

Properties: This card shows the properties of the current file in the treeview depending on the file type (single file, dataset, folder...). The **Comment** can be used to store additional information about the dataset. The **Batch jobs** field shows the name and date of all processing operations, which have been applied to this dataset. Both text fields can be edited directly. The information is stored immediately.

4.3.1 Current Project

The **Current project** window displays the whole content of the project which is currently in use.

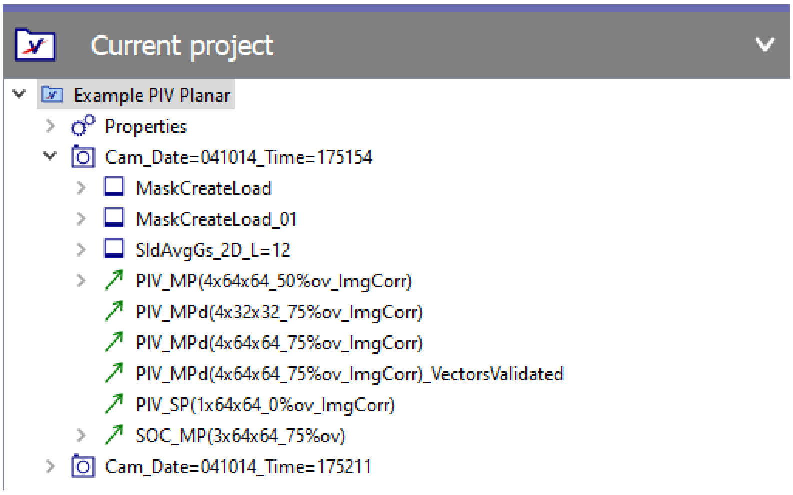


Figure 4.9: The **Current projects** window

The **Current project** window (Fig. 4.9) is situated at the lower left of the screen, below the **Workspaces** area. It displays a directory tree of the currently selected project with all its saved data sets:

- ▷ Calibration data under the *Properties* node shows the currently active calibration.

  ! A project can only have one *active* calibration!

- ▷ Calibration history (date and parameters of previous calibrations)

✓ Right-clicking on a previous calibration allows to reactivate it, i.e. all further calculations concerning this project will be performed based on this calibration until the active calibration is changed again.

▷ Raw images from recordings.

▷ Processed images, vector fields or plots.

▷ Imported images or vector fields.

4.3.2 Context menu of the selected set

Right-clicking on a data set of the current project opens a context menu with the following options:

Processing opens the **Processing** dialog to evaluate the selected set, see page 209.

Hyperloop: Execute the same processing for a number of sets, see page 81.

Processing history: All parameter settings of a processing are stored in the result set. Using the history an old setup can be displayed and even loaded for a new calculation.

- Reactivate the parameters used to calculate this data set for use in further calculations.
- Check properties to view previous calculation parameters.

Open in file explorer: Open the data folder with the Windows file explorer. This function is also available via shortcut keys `Shift`, `Ctrl`, `Alt` and `E`.

Rename the data set.

Delete the data set. This physically removes your data from the hard drive! A dialog opens that provides options to delete data from a set (see Fig: 4.10).

1. **Delete data set and all subdirectories:** Deletes the selected set with all the subdirectories.

2. **Delete all subdirectories:** Deletes all the subdirectories but not the selected set.
3. **Delete ranges of images/vector fields:** Deletes the selected images/vector fields from the selected set.
   You may enter numbers of individual images or several ranges (from-to) separated by semicolon as visible in Fig. 4.10. The corresponding files will be deleted (no recycle bin!) and the remaining images renamed to B00001, B00002, ..., B0000N.

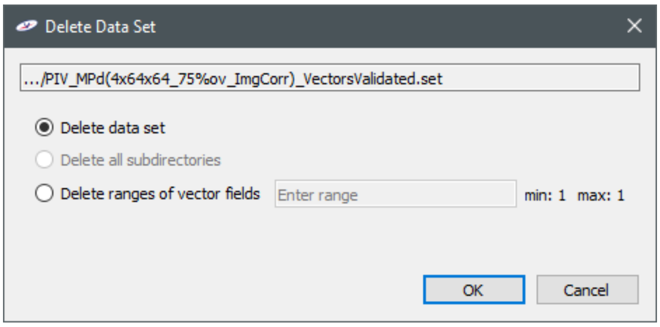


Figure 4.10: Delete data sets

Copy to: Copy the selected set to a selectable location in the file system.

Move to: Move the selected set to a selectable location in the file system.

Create new folder within the data set.

Export data to foreign file formats or to AVI movies, see page 145.

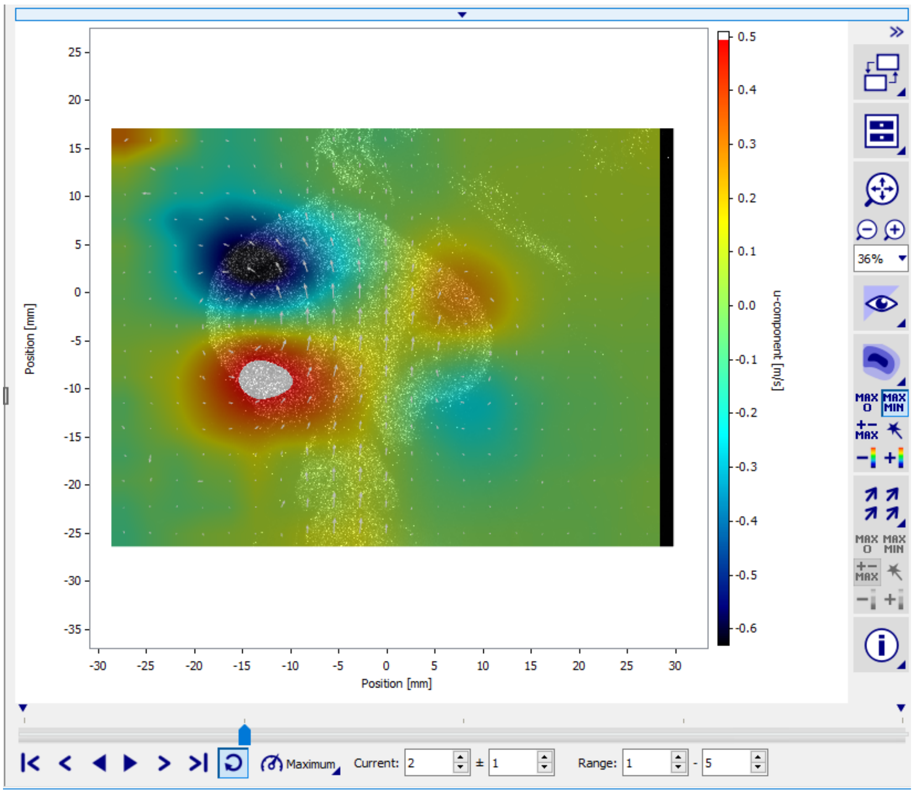
Export data set for support (incl. calibration data) in order to send it to the **LaVision** technical service.

Move items by drag&drop

Moving sets in the current project tree is also possible with drag&drop: Click to an item, hold the left mouse button and move the item to another set or to the root item, then leave left mouse button. **DaVis** will ask if the the set really should be moved.

4.4 Data Viewer

The **Data Viewer** dialog can be used to animate recorded or processed datasets. The functions of the player are described in the table 4.1. Detailed descriptions about the view configuration itself are given on page 103.



Button	Description
<	Step one image backwards.
|<	Plays the movie backward to the first image.
>|	Plays the movie forward to the last image.
>	Step one image forward.
■	Stops the playback.
↻	When this button is turned on, the playback is repeated until you stop it. This is like an endless loop.

Table 4.1: Data Viewer control buttons

With the small markers above the position slider the first and last image of the movie can be defined. Click to the left or right slider and move the position. The selected range is painted in gray color.

In case of high-speed recordings the unit of the movie player can be switched between image index, acquisition time and crank angle.

4.5 Export Data for Support

In the dialog **Export data for support** of figure 4.11 a large number of files can be exported and send to the **LaVision** service department:

- single files
- zip archive including a whole set with the corresponding source files in the project manager hierarchy and (optionally) the image calibration of the project (plus calibration images if needed). Additionally the calculation parameters within the result set files are archived.

This way it is easy to hand over sample images (with calibration if needed) to reproduce calculation results on other computers or to solve calibration issues. Open the dialog shown in Fig. 4.11 via the context menu on the selected set.

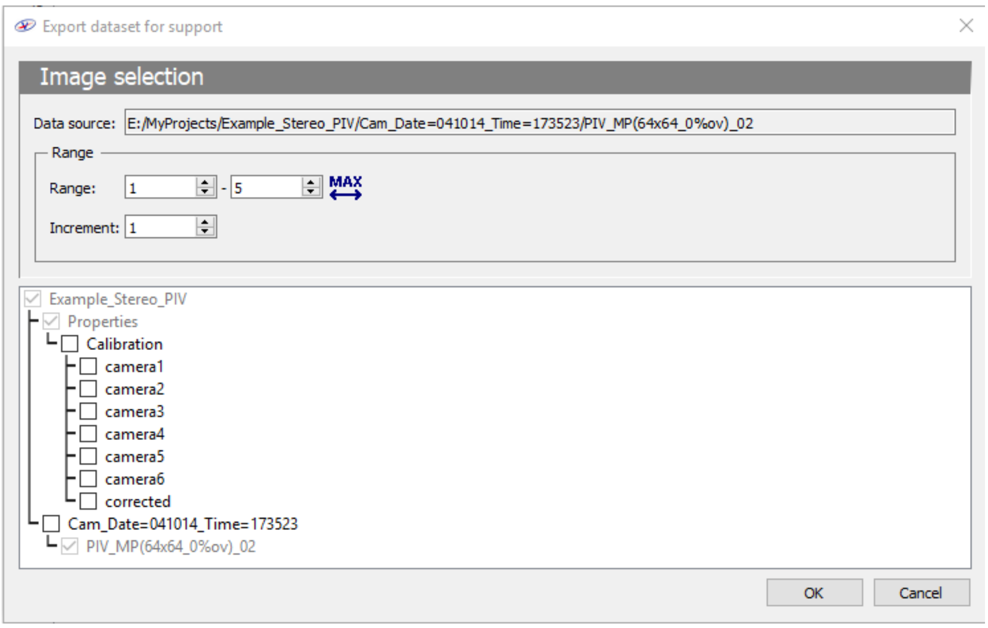


Figure 4.11: Dialog to create support file for **LaVision** service.

The data range is given for a single file, for a whole set or for any subset of images in the set file. In the lower part some selection items are used to include the calibration and/or the calibration images. The corresponding source images (or source vector fields) in the project manager hierarchy can also be included to allow the service to repeat the same processing.

After clicking the **OK** button a fileselectbox opens to enter path and name of the zip archive. This file can be send by email to the **LaVision** service department.

4.6 Hyperloop

With the **Hyperloop** dialog (Fig. 4.12) an operation can be executed for an arbitrary number of datasets of the **same type and hierarchy level** within the project. The dialog can be reached by clicking with the right mouse button on a dataset and then selecting one of the following options:

Hyperloop – All Sets: Select all sets within the project and open the **Hyperloop** dialog.

Hyperloop – Current folder: Select only the sets of the current folder.

Hyperloop – Root folder: ... : This option can be used to specify a parent root folder. All sets below this folder will be selected.

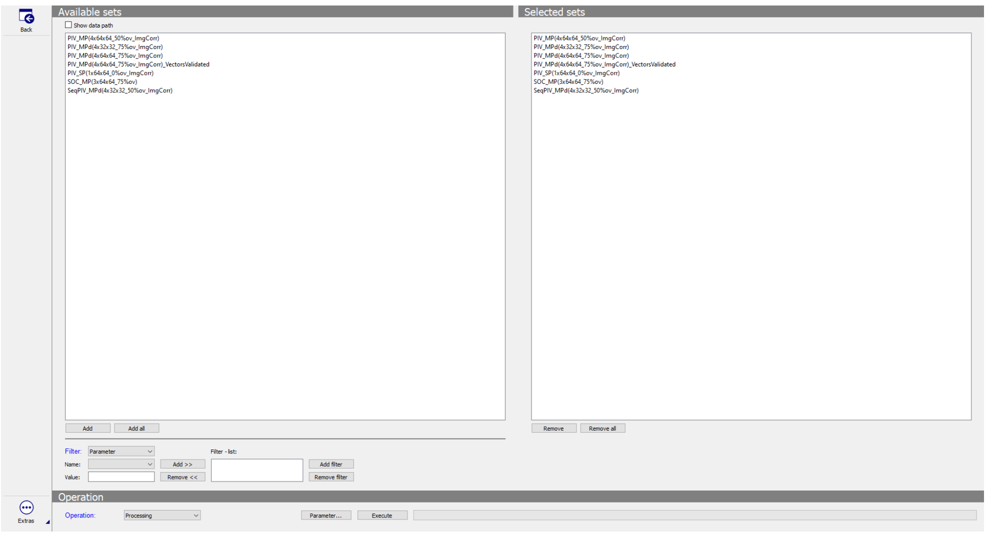


Figure 4.12: Hyperloop dialog

The dialog is divided into the following sections:

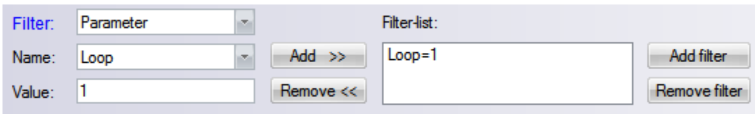
1. **Available datasets:** A list of all available datasets is given. Press buttons **Add** and **Add all** to append one or more datasets to the list of **Selected datasets**.

Show data path: Enable or disable the display of the complete path of the datasets within the current experiment.

2. **Filter:** Create some filters to append or remove datasets to or from the list of **Selected datasets**. This is very useful for processing datasets with specific name characteristics.

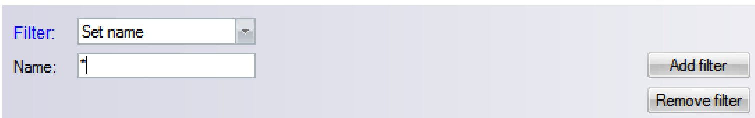
Filter: Choose the filter type **Parameter** or **Set name**.

Parameter: This filter uses parameters which are coded into the set names. The parameter has to be of the format:
..._<PARAMETER>=<VALUE>_...
for example: CamImages_StageX=110_PivDt=3
This name contains the following parameter:
StageX = 110 , PivDt = 3



The **Name** item contains all parameters which are found in the **Available datasets**.

Dataset name: This filter uses wildcards to filter the datasets, for example:
CamAngle=*PivDt=*



Add filter: Add the filtered datasets to the list of **Selected datasets**.

Remove filter: Remove the filtered datasets from the list of **Selected datasets**.

3. **Selected datasets:**

Remove: Remove one or more datasets from the **Selected datasets**.

Remove all: Remove all sets from the **Selected datasets**.

4. **Operation:** The available operations depend on the current project type. Press the **Execute** button to start the selected operation for all listed datasets. Some standard operations are:

Processing: Execute all selected datasets with the current **Processing** operations. Press button **Parameter** to change the **Processing** settings (see chapter on page 209).

Reorganize data: Reorganize the data of the selected sets in one/several new sets. Press button **Parameter** to change the **Reorganize** settings.

Delete: Delete all selected datasets and subdirectories.

Move: Press the **Parameter** button to choose between:
Move to: Move selected sets to the specified location.
Move n level up: Move each set to a parent directory.
Attention: Make sure that there are no sets with the same name.

Export: Export all selected datasets. Press button **Parameter** to change the **Export** settings. See chapter **Export** on page 145.

Rename: Press the **Parameter** button to choose between:
New name: Enter a new name for all sets.
Replace substring: Replace a substring in all sets with a new one.
Attention: Make sure that there are no sets with the same name.

4.7 Reorganize data

With the dialog **Reorganize data** some specific files can be copied from an arbitrary number of source sets and saved in certain order in one or several new sets. This dialog is opened from the **Hyperloop** dialog (see page 81).

For example (Fig. 4.13) a user wants to create one set with the average images of several source sets. The result set **B00001_avg** consists of 10 average files for the 10 loop steps.

1. First, specify the source sets using the **Hyperloop** dialog: Click with the right mouse button on a set in the project treeview and select **Hyperloop – All Sets** to include all sets of this hierarchy level, see ref 4.14.

   Then, the **Hyperloop** dialog includes all selected sets in the list of source files (Fig. 4.15).

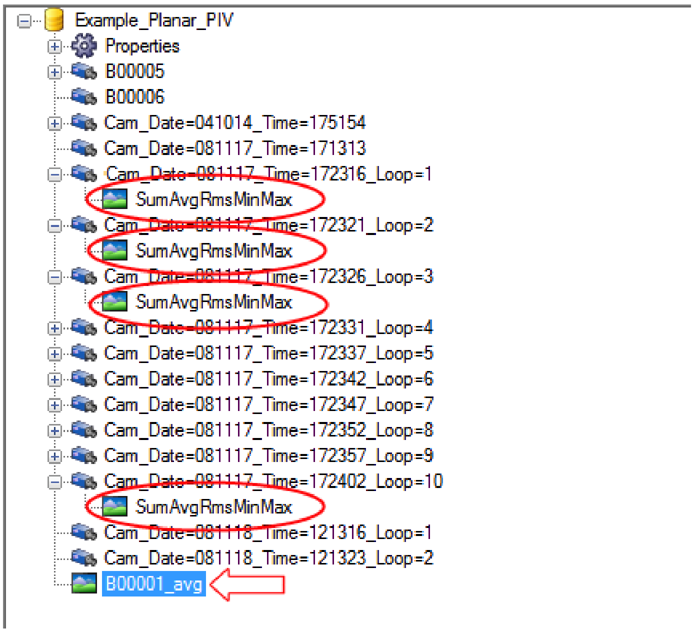


Figure 4.13: Example of reorganization

2. In a next step, specify the sets to be reorganized using the **Filter** and the **Add/Remove** buttons (Fig. 4.15).

3. Select the operation **Reorganize data** and press the **Parameter** button to open the setup dialog (Fig. 4.16).

   In the **Reorganize data** dialog of Fig. 4.16 one file of the first data set has to be selected. This file is copied from all selected sets to one new set, e.g. with 10 source sets the result will be 1 set with 10 files. The files are sorted by the variable parameter in ascending order if the parameter is available in the format ..._<PARAMETER>=<VALUE>_....

   - **Available files:** List includes all files of the first set. One (and only one) file has to be selected to be extracted from each set.
   - **Selected file:** Displays the name of the file which is copied to the result set.
   - **Ignore variable parameter:** Specify the nonrelevant parameter which must not be used for the reorganization (for example: Date, Time).
   - **Select variable parameter:** Specify the main variable parameters. This parameter is varied in each result set.
   - **Path:** Select the path of the result set.

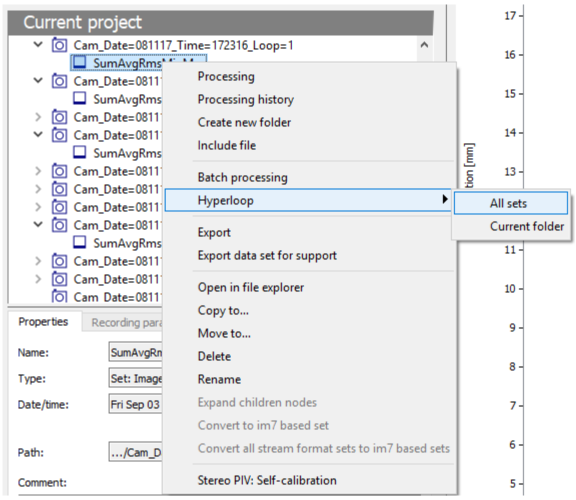


Figure 4.14: Open the hyperloop dialog for reorganization.

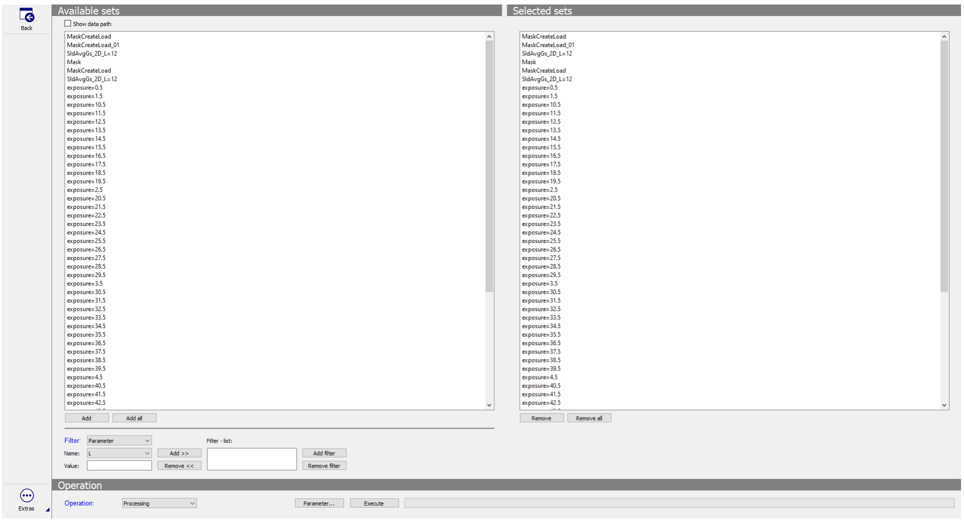


Figure 4.15: Selection of data sets to be reorganized

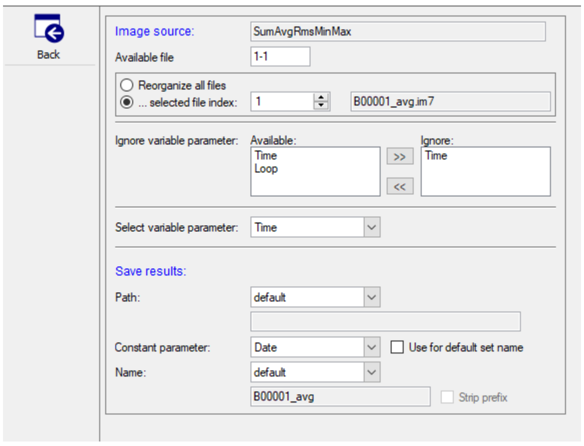


Figure 4.16: Reorganization parameters

- **Use for default set name:** Choose for each constant parameter if it should be used to create the default set name of the result.
- **Name:** Specify the name of the result set.

4. The reorganization can be started with the **Execute** button in the **Hyperloop** dialog.

Example

Figure 4.17 gives an example for the reorganization of this setup.

- **Source:** Scan with 2 translation axis, X-axis: 50-60, Y-axis: 10-12
- **Reorganize:** Variable parameter: X-axis
- **Result:** 3 sets (one for each Y-axis position) with 6 files (for each X-axis position: 50,52,54,56,58,60).

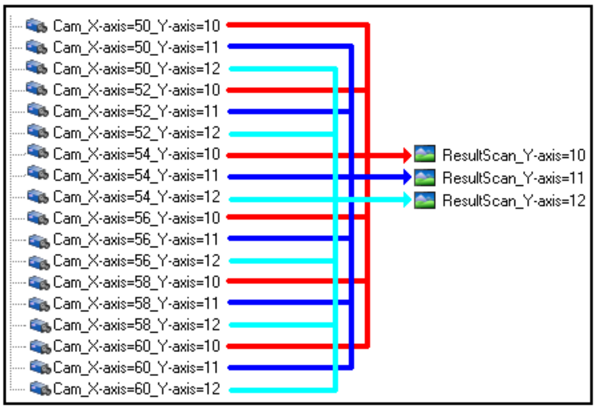


Figure 4.17: Reorganization example

4.8 Include File

The **Include File** option can be used to add a file of any extension into the tree structure, e.g. a PDF document, an Office document or images like BMP or PNG or JPG. This function is available in most project types and can be reached by the context menu on any level of the project tree.

After clicking **Include File** in the context menu a file select box opens. Please select the file to be copied into the project and added to the project tree. **Caution:** It is not possible to add **DaVis** files (extension `im7`, `vc7`, `imx`, ...). For this files the **Import** dialog must be used.

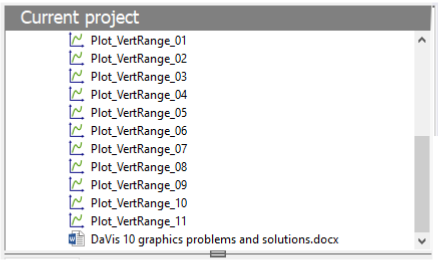


Figure 4.18: Include File example.

With a double click on the included item the file will be opened by the default application. The file can be deleted with the context menu item **Delete File**.

4.9 The SET file and its directory structure

4.9.1 Simple SET

The format of simple SETs has not changed in former versions of **DaVis**. Old SETs created by **DaVis** 7 or 8 and can be loaded into **DaVis** 10. The `SET` file itself is a simple text file and lists all static CL-variables of the included groups of variable, see manual **1003002_CommandLanguage_D10.0.pdf** for details. A subdirectory of the same name as the `SET` file includes the image or vector files:

```
MyData.set
MyData/B01.im7
MyData/B02.im7
MyData/...
MyData/B17.im7
```

The image or vector files are ordered and named with a starting `B` for **buffer** followed by the index in the set. The example of the SET above contains 17 image files.

Sometimes optional parameters are coded into the file name in a style of name/value pairs:

```
MyData/B01_delay=0.03.im7
MyData/B02_delay=0.08.im7
```

When DaVis is loading such files, the name/value pairs are automatically converted into attributes and can be accessed later by some processing operations or by the display e.g. to show the values as overlays.

4.9.2 Stream SET

The stream SET came into the **DaVis** software with version 8.4. This format uses the same SET file as the simple SET but organizes the image data, scales, attributes and other information in another way. There are different binary files for all data types, see figure 4.19. The image data is usually stored in one file per camera with all images of e.g. a recording instead of one file per image. This decreases the number of files per SET and increases the writing speed dramatically. Therefore the recording dialog stores stream sets instead of simple sets.

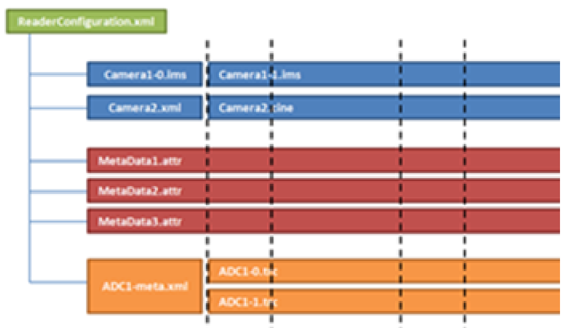


Figure 4.19: Example for files stored in a stream set.

All standard dialogs and functions in DaVis are able to read with SETs of all types, e.g. the processing or export dialogs. Processing results are usually stored as simple SETs and not as stream SETs.

Stream SETs can be converted to simple IM7 based SETs by using the context menu in the **Project Browser** dialog, see figure 4.20. This is useful when a set is recorded with DaVis 10 and should be loaded into a older version of DaVis (up to version 8.3) or by other software such as Matlab.

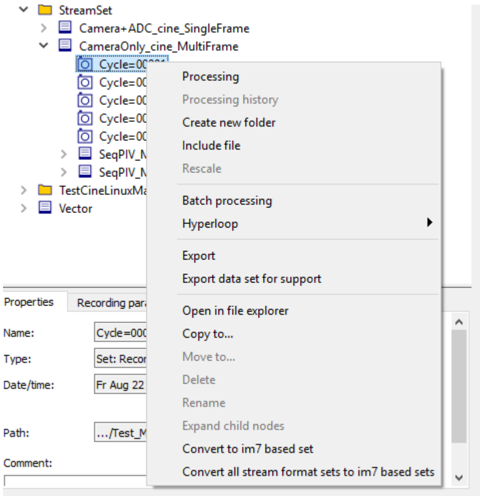


Figure 4.20: Convert stream set to simple set.

4.9.3 Multi SET

The multi SET contains several simple sets or streams sets and is created e.g. when scanning in the recoding dialog. Each scan point creates one set which is often named as **cycle** with ordered index. The multi set can be accessed like one *normal* set in the **Project Browser** dialog and processed by the **Multi Set Processing** dialog. This opens when the multi set is selected in the project tree dialog when pressing the processing dialog toolbar button.

The order of multi sets is like figure 4.21. Here the loop recording with highspeed cameras stored either time and loop or phase and cycle in two dimensions.

Multi set processing allows to process all child sets with the same operation e.g. to average each child set (cycle). The result would be again a multi set with e.g. one averaged image per cycle. The second mode of the multi set processing dialog allows to process e.g. all first images, all second images of every cycle and so on. As result a multi set is created in a reordered way. In case of averaging the result can even be a single set with all averaged first images, seconds images and so on.

By default the project tree hides the cycles as shown in figure 4.22. After opening the context menu of a multi set the option **Show subsets** enables the display of cycles like in figure 4.23. This display is useful when a single cycle should be processed or exported.

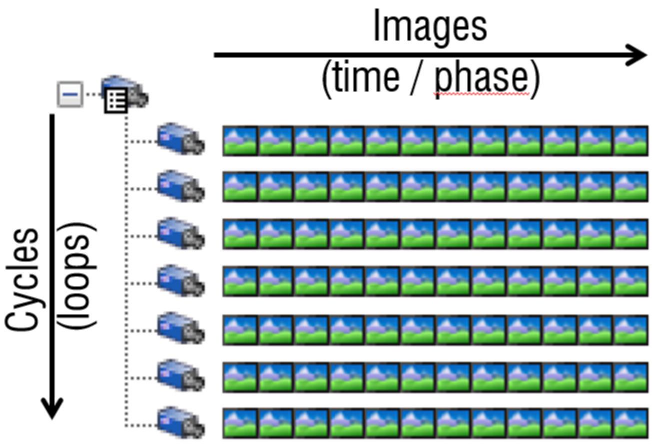


Figure 4.21: Example for loop recording in two dimensions.

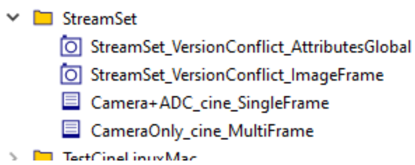


Figure 4.22: Default display of a multi set with hidden cycles.

4.10 Repairing corrupt SETs

Corruptions of SETs sometimes come from harddisc errors or errors with external USB drives. Another known issue is a copy process to a destination without enough space, which leads to missing or incomplete copied files.

In case of **Simple SETs** a `IM7` or `VC7` would be corrupt or missing. This corruption state is not detected when opening such a `SET` but later e.g. when playing through all images. In case of a missing or corrupted image

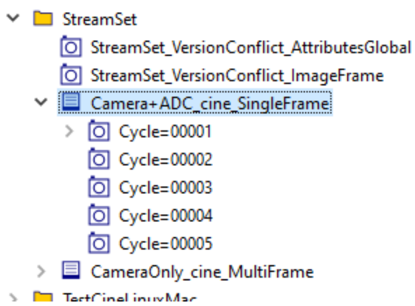


Figure 4.23: Display of a multi set with its cycles.

the player changes to red color. A processing on such a corrupt SET will work in most cases just by skipping the missing file.

For **Stream SETs** corruptions are more difficult to handle because of containing a small number of large files. When opening a corrupt **Stream SET** in the project browser a warning dialog (see figure 4.24) will open, give information about the corrupt files and what would happen in case of repairing.

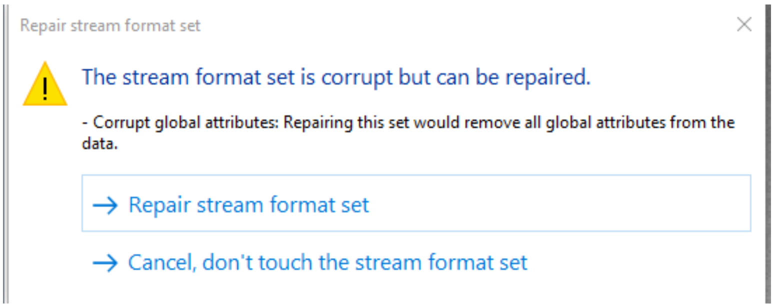


Figure 4.24: Information and repair dialog for corrupt Stream SETs.

The following issues are detected:

- **Images:** In case of a corrupted images file with missing images at the end of the file the number of images can be reduced. Even if only one the file of one camera is corrupt, the files of the other cameras also can not be read completely. If a image file can't be recovered at all then the complete set is unreadable.
- **Attributes:** A corrupt attributes file either with global attributes or with frame attributes can be removed. Then the attributes are partially missing but the images can be used for display and processing.
- **Scales:** A corrupt scales file can be removed and e.g. replaced later by a processing operation to define new scales. If the set includes a recording then the scale is not important when using the calibration information during processing.
- **AD traces:** Corrupt AD traces would be removed from the set.
- **Configuration:** A corrupt configuration file can't be repaired automatically. There is a possibility to repair the xml-file manually or replace it by a configuration file from another but equal recording.

5 Data Buffers

5.1 About Buffers

DaVis is designed for the acquisition and processing of data from CCD and video cameras.

A **CCD** (*C*harge *C*oupled *D*evice) or **CMOS** (*C*omplementary *M*etal-*O*xide-*S*ilicon) is a 2-dimensional image sensor. It consists of an array of separated image elements, so-called **pixels**. The whole set of intensities, pixel by pixel, builds up the image.

An image or profile is (in **DaVis**) internally stored in a block of main memory, called a **Buffer**, including all pixel in a 2-dimensional table and some additional information like scales and attributes. The image buffers are either of the type WORD (16 bit per pixel, integer values 0-65535), FLOAT (32 bit, floating point) or DOUBLE (64 bit, floating point). Other buffer types are available for color images (32 bit RGB, 8 bit for each color component red, green and blue) and for vector data (two or three floats: Vx, Vy and Vz).

A buffer is displayed on the screen in an **Image Window**, which is a **view** of the raw data and can be of the types **image, plot, profile** or **3-D**.

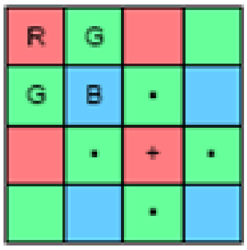
5.1.1 Buffer Properties, Attributes and Scales

The basic properties of every buffer are size and format. This properties define a buffer and its raw data. The scaling properties give additional information about each dimension of a buffer and about the intensity.

It is possible to add a unlimited number of free information (strings, arrays), the so called **attributes**. These can be used by the customer or macro programmer for own desire, while the standard properties (and some special attributes) are defined and handled by **LaVision**'s projects. A list of standard system attributes had been given in the appendix of the *Command Language Manual* until version 10.1.2.

5.1.2 Bayer Pattern

This applies only to buffers containing raw images from single-chip color cameras whose red, green and blue sensitive pixels are arranged in the so-called Bayer pattern.



These buffers are marked by the buffer attribute `RGBFrame<n>` (n = frame number) with a value > 0. In this case a color interpolation takes place which converts the intensities from the Red, Green and Blue pixels to a color image to be displayed on the screen.

5.2 Buffer Information

The properties of a buffer are displayed in the **Buffer Information** dialog (figure 5.1). This can be opened with the information icon in the view's toolbar on the right side of the view.

5.2.1 Data Properties

The **Data Properties** dialog displays the main properties of a buffer: size, type, additional data fields like mask and scalar fields and the acquisition time.

The meaning of the displayed information are:

- **Data type:** Images of data type WORD (16 bit integer number), FLOAT (32 bit floating point) or DOUBLE (64 bit floating point), a vector format (two or three values per vector: Vx, Vy and Vz) or a RGB color format (32 bit RGB, 8 bit for each color component red, green and blue).
- **Size** in pixel of horizontal (X) and vertical (Y) direction, depth (Z) and frames.

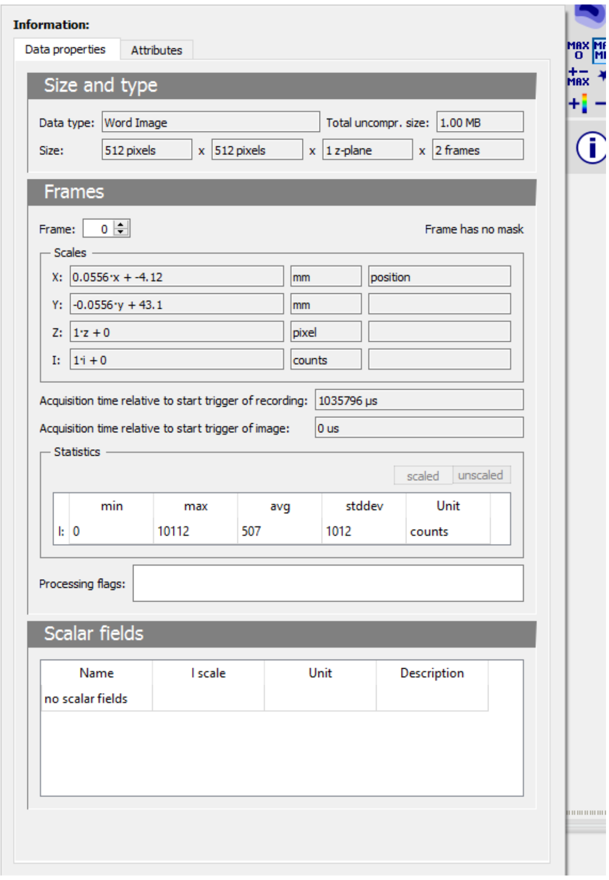


Figure 5.1: Buffer information.

- **Grid** size is displayed for vector buffers only.
- **Mask** is attached to the buffer. The mask information is used by some processing functions and can be displayed on screen e.g. to disable unmasked pixel or vectors.
- **Frames** can have different **scales** and different **acquisition times**. Select the frame index, counted from 0 to n-1 for a buffer with n frames, to display the scales, times and statistics of a certain frame.
- **Processing flags** give some information about the former processing operations if the buffer already has been processed and e.g. has been corrected using the project's calibration.
- **Scalar fields** are additional information for each pixel or vector. By default each pixel has one information only, the intensity. Some processing functions are calculating additional information, e.g. the height information or peak ratio for vector buffers. The number of components and the name list are available.

A scalar field has a **name**, intensity scale (**I scale**, **Unit** and **Description**) and a certain **format** (WORD, FLOAT or DOUBLE) and usually has the the same **size** as the master image or vector. Scalar fields with free **size** are not displayed here. Own **scalings** can be defined.

5.3 Buffer Scales

A scale is a linear mapping of the X (horizontal), Y (vertical), Z (depth, used for real 3D buffers only) or I (intensity)-axis to a new range of values. Scales can be assigned to an image buffer or predefined for each camera.

The linear scaling function for real intensity value I to scaled value I_s uses a factor a and an offset b:

$$I_s = I * a + b$$

Each frame of a buffer has its own set of scales. They are a **property** of that particular frame and are stored with it.

5.3.1 Default Scales

The **default scales** for the axis X (horizontal) and Y (vertical) are the pixel locations on the CCD: e.g. for a 1k x 1k camera ranging from 0 to 1023 for X and from 0 to 1023 for Y.

The I-scale corresponds to the intensity values ("counts") as a default. For a WORD buffer it ranges e.g. from 0 to 65535 counts.

Press and hold the `SHIFT` key and move the mouse cursor inside the buffer view. The pixel's coordinate and intensity are given in the **position info label** in pixels and counts.

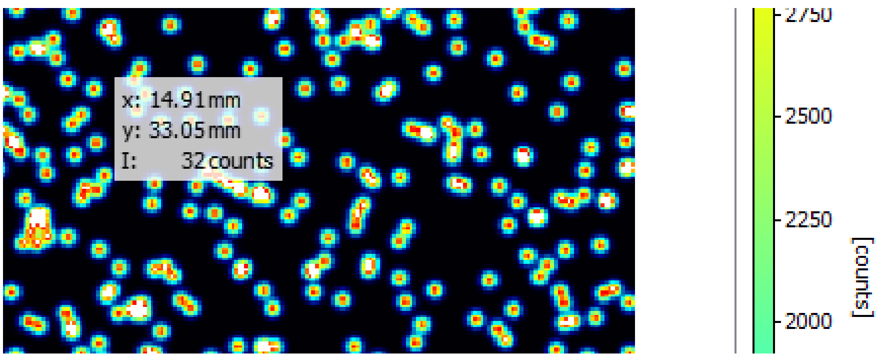


Figure 5.2: Position info label at mouse cursor position in unscaled values.

5.4 Attributes

Attributes are additional information (strings and arrays of integer or float values) of the data object and are often set by the image and analog data acquisition functions and by the processing functions. A list of used attributes is given in the appendix to the *Command Language Manual* and should be read if user processing operations are using their own attributes or changing standard attributes. Attributes can either be stored together with the whole image or vector field, or they can be bound to a certain frame and store different values in the different frames of an image or vector field.

5.4.1 Attributes in DaVis 10.2.0

This dialog is part of the **View information** dialog. Card **Attributes** of figure 5.3 presents a list of all attributes, which are defined for the displayed buffer.

The list starts with a section of attributes, which are not bound to a certain frame. The second section includes the so called **Device Data**, which are acquired during recordings usually from A/D converters. See the chapter on page 175 for detailed descriptions. The last section gives access to all frame attributes. This list is partially displayed in readable form in the first part of the dialog (**Data properties**) e.g. with the acquisition timings.

5.4.2 Attributes in DaVis 10.2.1

This dialog is part of the **View information** dialog. The tab **Attributes** (see figure 5.3) shows all attributes in a hierarchical display (tree), which are defined for the displayed data object.

The tree displays the attributes grouped by themes and sub-themes. For example, all attributes that are related to the data acquisition are displayed under the tree node **Recording**. The attributes related to cameras are grouped under the tree node **Camera**, which itself is under the tree node **Recording**. Most attributes have additional data that is displayed. A short description is shown directly beneath the attribute's name. When an attribute is selected, the lower part of the dialog shows a longer description of the attribute and the attribute's value(s). In case of frame attributes, means attributed bound to frames, the values of each frame are given in a list separated by the pipe symbol.

The second block in the dialog gives additional information about the selected attribute: a short description and the real attribute name. The real name has to be used in the processing dialog when attributes should be modified.

The third block in the dialog displays a table of frames and value per frame in a table, or e.g. in case of the AD trace attribute the trace is displayed as simple profile. There are some more attributes with individual views in this block.

The attributes can be filtered by entering text in the line above the tree, where the filter text is matched against the attributes' names. With the button to the right of the filter line the attributes can be exported in a `csv` format.

Selecting the checkbox on the left of the attribute allows to display the attribute's value as overlay on top of the image or vector field. The attribute item can be moved in the view. With a right mouse click on the item a configuration editor opens to change size and color of the display or to fix the position of the item relative to one of the four corners of the view. In this mode zooming and panning will not change the position. Otherwise, using the pointer in the middle, the attribute item is pinned to the current pixel position of the image or vector field.

5.5 Buffer Statistics

The buffer statistics can either be displayed in the **Buffer information** dialog in section **Data properties** or in the dashboard. The first dialog calculates the statistic on complete frames while the second dialog on top of the image or vector view can be used to select a user defined rectangular region as source for the statistics.

Click into the small triangle on the bottom right of the statistics field in the dashboard and then **Set region of interest**. Press the left mouse button in one corner of the rectangle, e.g. upper left, and then move the mouse with pressed button to the lower right position.

Statistics are computed in the selected rectangle from the displayed buffer. For buffers with masks only the valid pixel are taking into account.

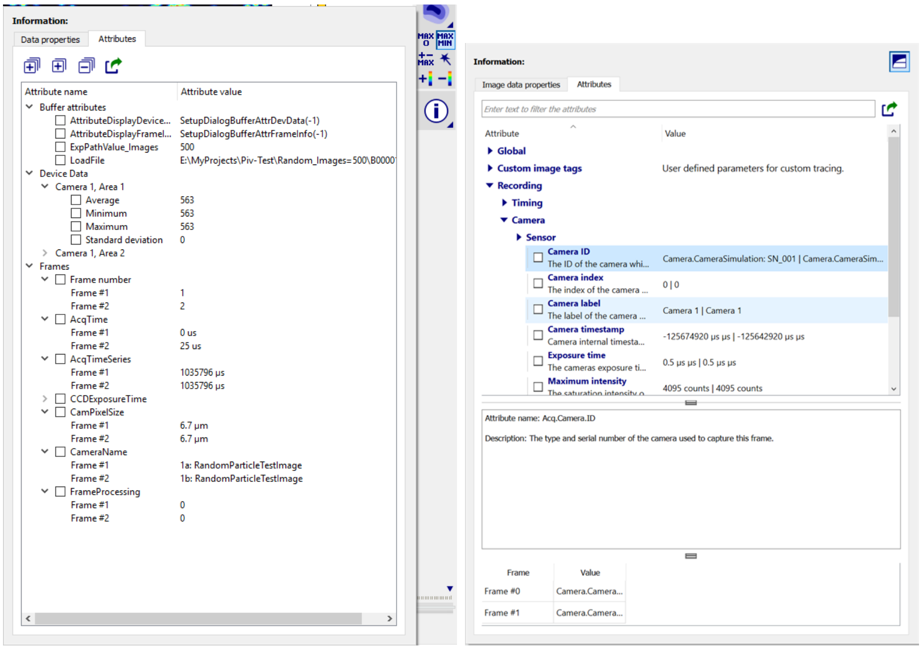


Figure 5.3: Attributes tab of the view information dialog in version 10.2.0 on the left and in version 10.2.1 on the right.

The selected rectangle is painted in the same color as the dashboard item on the view.

The statistics are shown in scaled units (switch **scaled**) as average with standard deviation and the values for minimum and maximum intensity.

5.5.1 Standard Deviation

There are two different ways to compute the Standard Deviation (stddev):

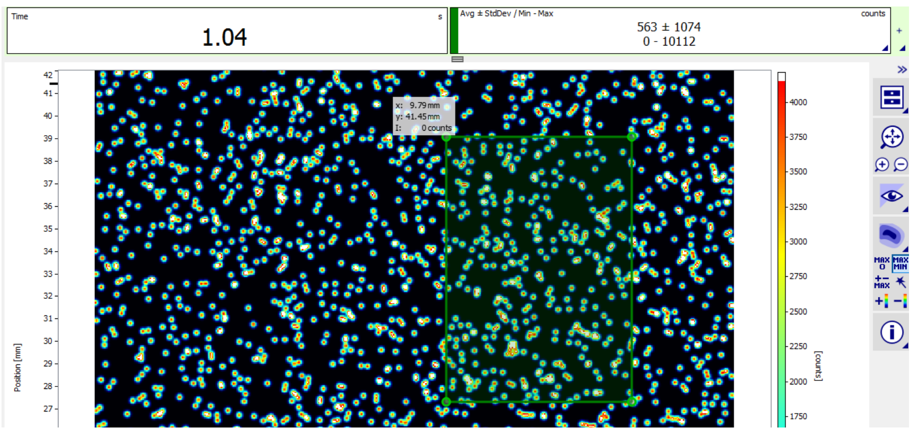


Figure 5.4: Buffer Statistics card of the Data Properties dialog.

Statistics can be calculated on a single image, where each pixel has a different intensity. Then the average intensity and the stddev based on the complete image is calculated. The results are scalar values. This is the standard deviation from each pixel intensity from the mean intensity.

The second way works with a number of images: Average and deviation are calculated for each pixel position separately. The result is either an averaged image or an image of deviation results at each pixel position. Therefore the algorithm needs to determine for each position the average and deviation.

There are two different options to execute the stdev calculation. The first options needs two loops: At first average all images, at second calculate the stdev, which needs the average to sum up the differences. The following formula gives the standard deviation of n images with term x_i representing the pixel intensity of any pixel of image i and $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ being the average:

$$\sigma = \sqrt{\frac{1}{n-1}\sum_i (x_i - \overline{x})^2} \tag{5.1}$$

Implementing this formula with two loops creates a time consuming algorithm, especially when the images can not be hold in PC memory, and each image must be loaded from hard disc for two times.

A second way is possible, which needs one loop only! The following formula can be used to calculate the stddev of a complete image with one loop. Again x_i represents a pixel of image number i from n images in total.

$$stdev = \sqrt{\frac{n\sum_i x_i^2 - (\sum_j x_j)^2}{n(n-1)}} \tag{5.2}$$

5.5.2 Stddev on Vector Buffers

For vector buffers the average, stddev, minimum and maximum are given for each direction (X, Y, Z) and in total.

The total Standard Deviation `stddevTotal` is calculated as $R_t = \sqrt{R_x^2 + R_y^2 + R_z^2}$ with R_x as stddev in X-direction.

5.5.3 Histogram

The **Histogram** item of the dashboard (see figure 5.5) creates a histogram of a complete image or of a rectangular region of an image. A rectangle can be defined by mouse after clicking the **Rect** button.

The histogram profile gives the number of pixels with the same intensity vs. intensity in counts. The given intensity range of the source buffer is divided into the number of slots depending on the with of the screen range.

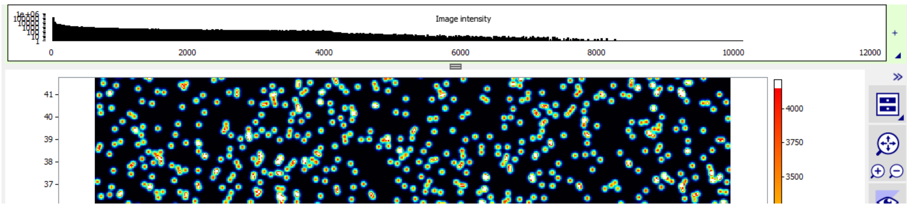


Figure 5.5: Histogram card of the Data Properties dialog.

6 Data Views

This chapter gives a general overview about the display of images, vector fields, plots and other data and set types. Most display options are described for general cases and not in detail.

The image view is completely described in the general section below and with the additional functions including **2D View** in section 6.2 and **3D View** in section 6.3.

6.1 General view layout and functions

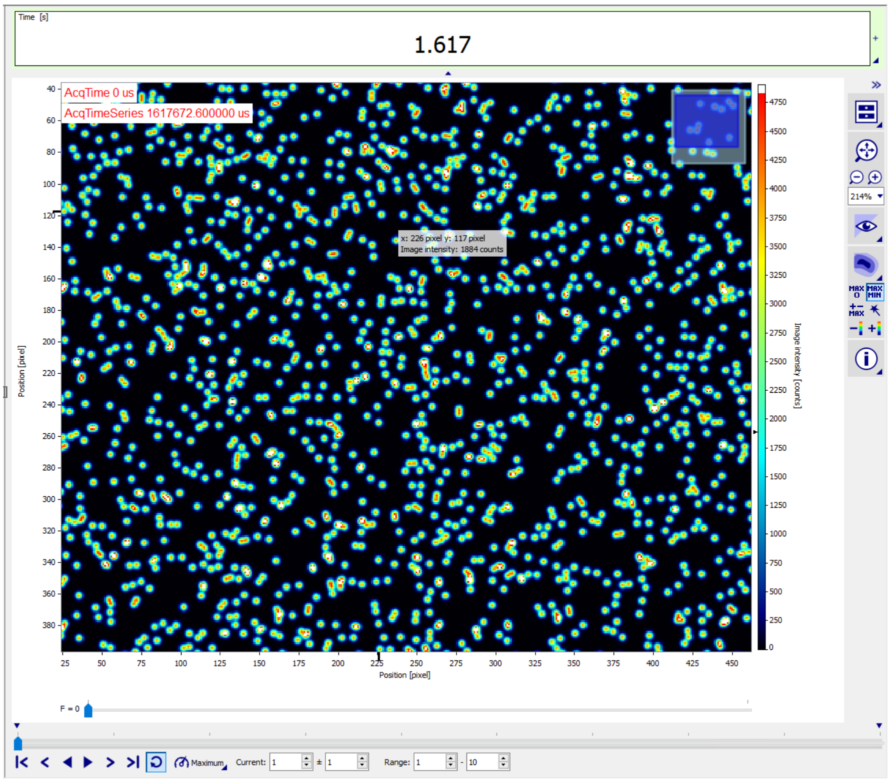


Figure 6.1: The general layout of image and vector field views with the image or vector field in the center and other parts like scales and toolbar around.

The layout of the views includes in most cases the image or vector field or plot in the center. On bottom the **player** can be used to select one item of the full data set. If the image or vector field contains several frames and/or planes, then a **frame slider** and/or a **plane slider** are displayed between view and set player. Both the frames and planes are 0-bases values, means the first frame is named as **F=0** on the left side of the slider. Below and left of the view are the **scales** with ticks, values, name of the scaling and units of the scaling.

On the right side of the view a **color bar** gives information about the selected color mapping together with intensity scales. On the very right the **tool bar** gives access to all display options. Above the view the **dashboard** can be used to display additional information about the data like acquisition time (see figure 6.1) or statistics.

Some overlays might be visible on the view: If the image or vector field is zoomed in and not completely displayed then a **position map** becomes visible in the top right corner. The displayed part of the data is marked in blue color, the full data as gray background. **Attributes** of the buffer can be displayed on any position like `AcqTime` and `AcqTimeSeries` in the top left corner of figure 6.1.

The **position info label**, see white rectangle with two lines of text in the middle of the view of figure 6.1, gives the scaled position of the mouse cursor and moves around with the mouse cursor. For images the scaled pixel intensity is given as second line. For vector fields the two or three components are given in the second line. When pressing the shift key while moving the mouse all values are given in unscaled values e.g. the position in pixel coordinates.

Distances within the view can be measured and displayed by the **Ruler** option within the context menu, see figures 6.2 and 6.3. To set manually the distance, place the cursor to the starting position, where the measurement should begin, and keep the left mouse button pressed until the end position is reached. The ruler is not bound to the specific location and can be moved freely within the view display.

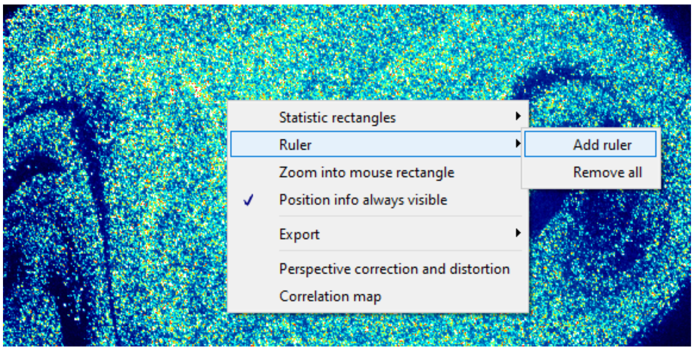


Figure 6.2: Ruler option via context menu.

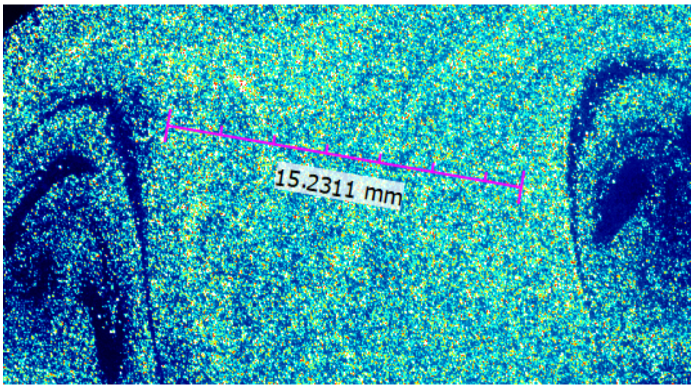
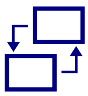


Figure 6.3: Measured distance.

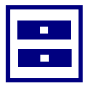
6.1.1 Switch view



Images can be displayed in 2D and depending on the data and project type also in 3D. Press the **Switch view** icon to switch between all available modes.

Vector fields can be also displayed in 2D and 3D. Other special modes are the Probability density function plot, the Scatter plot, and the FSI 3D view.

6.1.2 Settings shelf



The settings shelf stores the active display settings with a free user selected name and can be used to switch between different display settings easily.

To create a new shelf item enter a name and press the plus button on the right, see figure 6.4. If no name is entered, the shelf item gets the name with date and time. To select another settings shelf just click on the name area. The shelf item can be renamed and deleted with the buttons on the right.

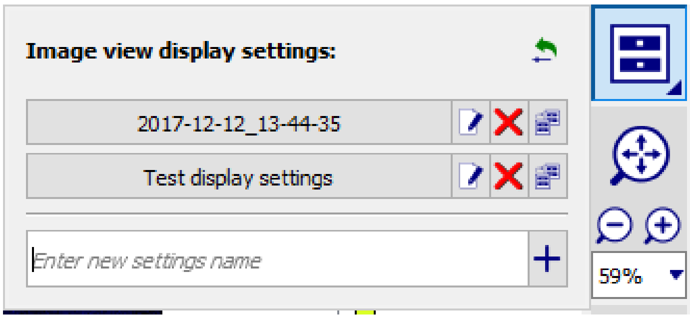


Figure 6.4: Shelf dialog for different sets of display settings.

6.1.3 Zoom items

By default when opening a new view the image or vector field is displayed completely fitting into the dialog area. With both buttons the user can zoom in or out. Click on the list of predefined zoom values (like 50%, 100% and 200%) or enter a zoom value of free choice. The 100% zoom value means that every pixel of the image is displayed as one pixel on screen.

If a specific area needs to be enlarged, right click on the source image and select **Zoom into mouse rectangle** via context menu, see figure 6.5, in order to zoom in manually in that area of interest.

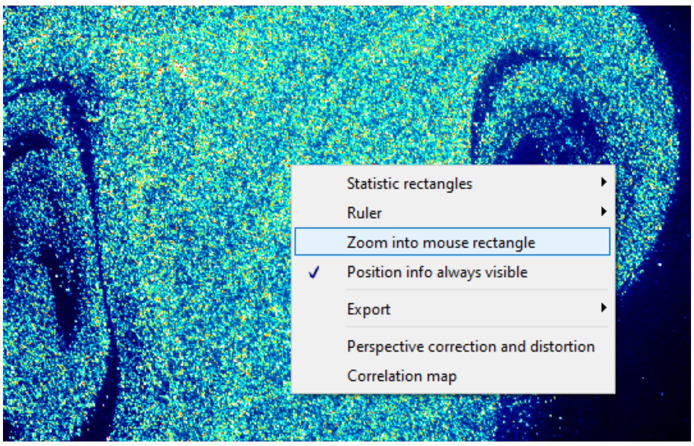
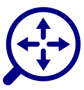


Figure 6.5: Zoom into mouse rectangle via context menu.



To return to the default zoom press the **optimal zoom** button. Zoom setting and view point (the position when zooming in and the moving the visible area of the image) are not stored in a shelf!

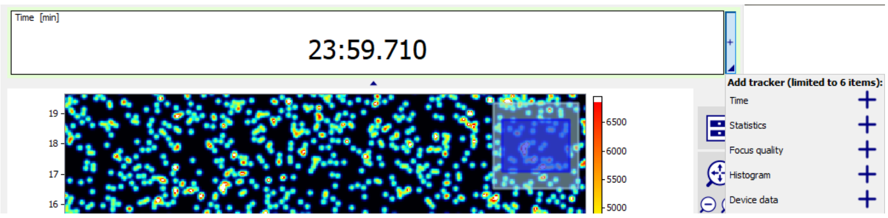
6.1.4 Information

Depending on the buffer type the information dialog gives details like size and data type. Depending on the selected frame the scales of this frame, the acquisition times and some statistics are given. See chapter on page 94 for more details. The second card lists all buffer attributes, see page 97.

6.1.5 Dashboard

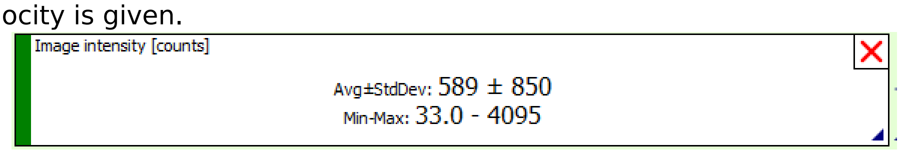
The dashboard displays additional information about the buffer and can be closed by clicking the small arrow head in the center between dashboard and view. There is only one dashboard even in multi view dialogs like the recording dialog.

Click to the small green column on the right of the dashboard to add new dashboard items. Up to six items can be displayed at the same time. The minimum number of items is one. The last item can't be removed. To remove another item press the red X-button which is displayed in the top right corner when the mouse moves above the item.



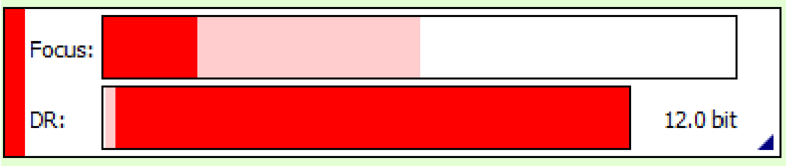
The following item types are available:

- **Time**: Acquisition time of this buffer in the recording. For processed data this could be the real time if information about the recording are missing.
- **Statistics**: Averaged pixel intensity with standard deviation, minimum and maximum intensity for images. For vector fields the total velocity is given.

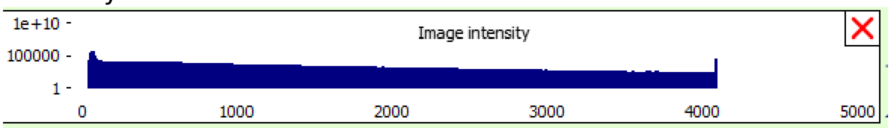
  

  By default the statistics of the complete image or vector field are calculated. Press the small black triangle in the bottom right corner to select a region of interest by mouse in the view or reset to the complete frame.

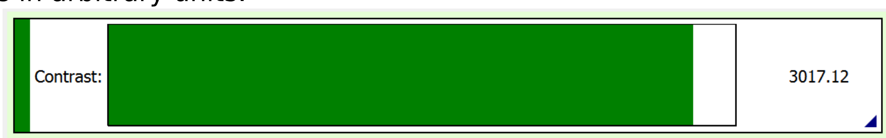
- **Focus quality** and **Dynamic Range** (**DR**) can be displayed for images only. Also the bit depth of the camera is given. The area of interest can be selected on the same way as for statistics.



- **Histogram** plots the number of pixel with the same intensity versus the intensity.



- **Contrast** shows the image contrast based on local intensity gradients in arbitrary units.



- **Device data**: Select a channel by clicking to the small black triangle on the bottom right of the item.

6.2 Images - 2D View

Some functions are available within both views, 2D and 3D, and are therefore only described in this section.

6.2.1 Component visibility

Enable or disable the display of **scales** (x and y) and **color scale**. A **row profile** and a **column profile** can be displayed optionally. Further, **data overlays**, a **navigator**, and **frame** can be switched to on.

6.2.2 Color mapping range

After pressing the color mapping button in the toolbar the dialog of figure 6.6 opens. In the first **range** card the **modifier** of the color mapping mode can be switched between **0 ... max** , **min ... max** , **+-max** and user defined range.

Further, the color mapping range can be increased or decreased . The **optimal color mapping range** button leads to the best resolution determined by the program taking the maximum intensity of the whole buffer into account. By pressing this button, the intensity range is fixed, and differences in several recordings can be compared to each other. The image is displayed in the maximum range of colors. For FLOAT buffers with a small range of intensities the optimal resolution is a min-max mode. Then, there is no appearance of "turn-overs" with abrupt changes, when the highest color is reached and the next higher intensity is displayed in color "0" again.

These settings can be also accessed directly in the toolbar by pressing these buttons.

In the **color** tab, displayed in figure 6.7, all **color** tables are build from 256 different colors each and the defined intensity range is mapped to this 256 color between minimum and maximum intensity. All intensities below the given minimum of the range, if painted in the first color, and all intensities above the given maximum value are painted in the last color. The order goes from bottom of the color table to the top. The color tables can be changed in the **Color Table Editor**, see section 6.2.3 below.

The default color tables number 27 and 31 show a **transparency** in certain areas of the table. The transparency is allowed to be between full transparent and solid. Color tables with **transparency** are used e.g. in the 3D display to make some background structures visible if foreground structures are transparent.

In the last card about **overlays**, depicted in figure 6.8 a **cross hair** can be placed in the center of the image or vector field and a **grid** of a selectable raster size (in pixel) can be painted, too. The option **Blending** allows to stack images on top of each other.

Frame offset specifies the index of the frame being stacked on top of the frame selected within the frame slider. **Example:** Setting frame 1 via

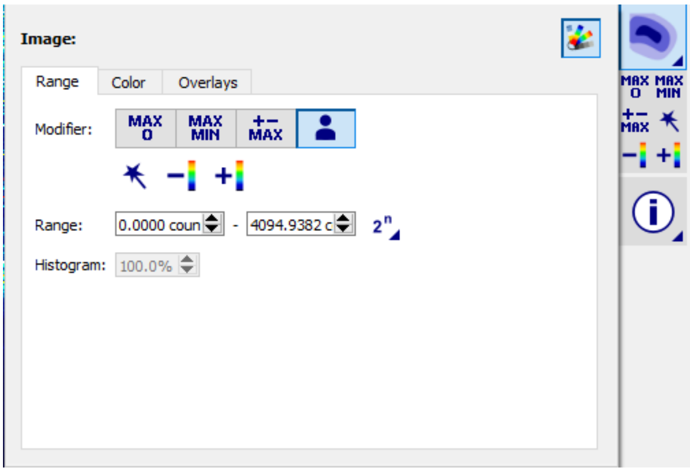


Figure 6.6: Setup dialog for color mapping of images, in this case for the different ranges.

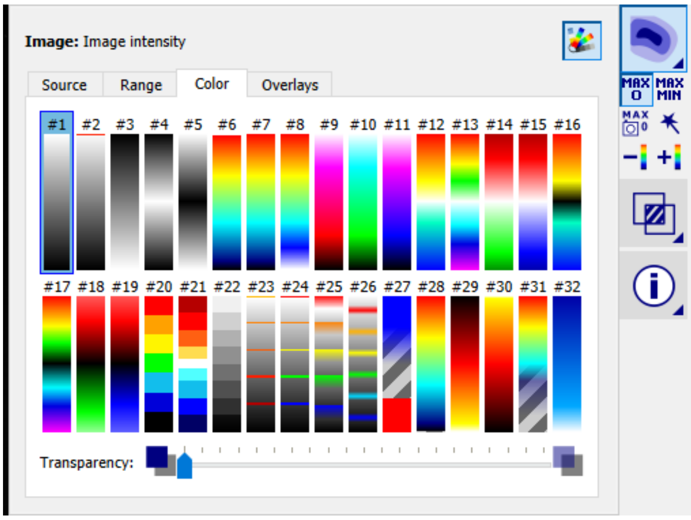


Figure 6.7: Setup dialog for color mapping of images, in this case for the different color tables.

frame slider and **frame offset** = 2 results in overlaying the frames 1 and 3. **Opacity** defines the transparency of the stacked frames. Low opacity values induce high transparency of the stacked frames, and thus only the initial selected frame is visible. Contrarily, high opacity values induce low transparency of stacked images. As a result, only the last frame is visible.

6.2.3 Color table editor

The **Color Table Editor** can be opened vis menu **Extras - Wizards - Color table editor** and opens the dialog of figure 6.9. At the first start the three user defined color tables are empty. Start by adding some **stops**. The

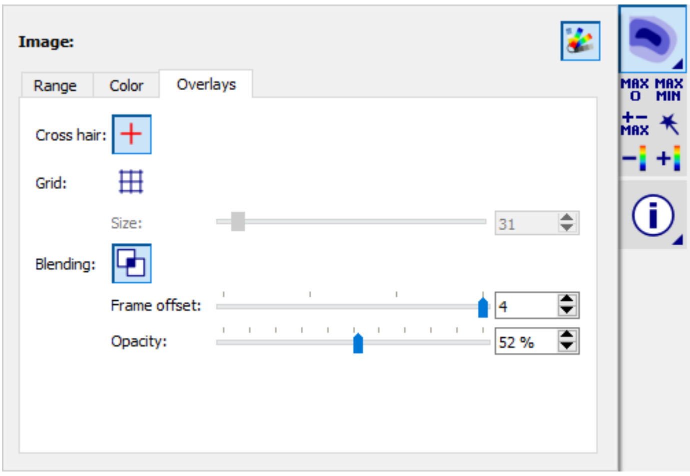


Figure 6.8: Setup dialog for color mapping of images, in this case for the overlays.

position of each stop can be changed between color index 0 and color index 255. The color table is defined by 256 colors in total which are interpolated between the defined color stops. Each stop is build from a certain color and an alpha value. The alpha values are 255 by default meaning a full solid color. Change the alpha value in the range between 0 as fully transparent to 255. The so defined color table is displayed on the left and automatically updated whenever a value is changed.

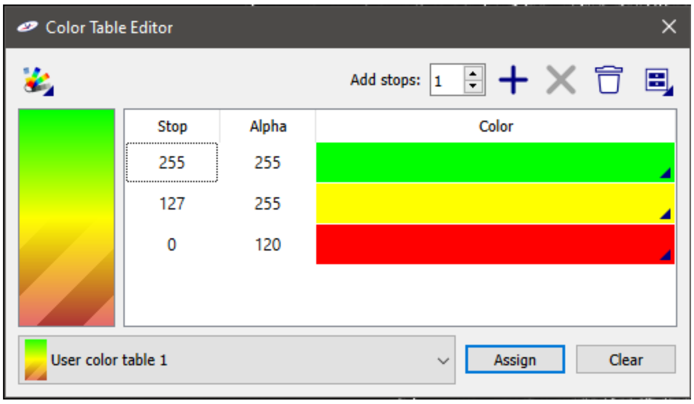


Figure 6.9: Color table editor dialog with a user defined table of three color stops.

After finishing the definition select one out of three available user tables on the bottom left and then press **Assign** to use the current color table in the **DaVis** views. With the shelf button on the top right different definitions of the user defined color table can be stored and reactivated later. With the button on top left one of the pre-defined tabled can be loaded into the editor as a base for own changes. After closing the editor the currently visible views are updating and the new color table can be used.

6.2.4 Overlay tools

The overlay tools allow to blend specific information over the currently displayed image. There are several tools to choose from to support the user in different tasks. The available tools may differ between project types. In the **Overlay tools** dialog it is possible to choose one tool from the drop-down list. Each tool has its own options. Therefore, the displayed settings differ from tool to tool. On top of the dialog the overlay functionality can be enabled or disabled and the scale corresponding to the active tool can be enabled and disabled. These two buttons are also available under the **Component** visibility button. The overlay's transparency can be configured with the **Transparency** slider at the bottom of the dialog.

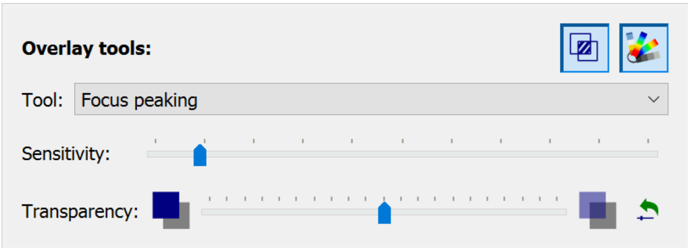


Figure 6.10: Overlay tools configuration dialog.

The **Overlay tools** are available in every 2D image view in any **DaVis** dialog. Some of the tools are most useful while setting up the system and preparing the data acquisition with the live view of the **Recording** dialog.

The following tools are available:

- **Focus Peaking**: In-focus areas of the image are highlighted in red. This tool is useful when setting up your system for data acquisition and you want to manually focus the lenses of your camera(s). The peaking depends on the contrast of your scene. Therefore, the sensitivity of the focus peaking can be adapted accordingly. This tool is also helpful for aligning cameras attached to a Scheimpflug when taking images of calibration plates for calibration purposes.
- **Image overlay**: This tool allows to overlay an existing image from a set onto the current image by using a defined color scale. This tool is helpful when you need to re-arrange your experimental setup to a previous state. It supports the user by repositioning the cameras to the same FOV as used for the overlayed image recorded in a previous similar experiment. Another use case are long term measurements in which it is necessary to track if the FOV of the camera(s)

have not changed meanwhile, e.g., because somebody has accidentely touched and moved the camera. The following tools are useful for PIV experiments:

- **Pixel shift**: The image is overlayed with a heatmap that represents detected pixel shifts within different areas of the image. The user can specify a desired pixel shift as required by the experimental setup. Areas of the image that match that desired pixel shift are highlighted in green. Areas with a smaller pixel shift are shown in red and areas with a higher pixel shift are shown in yellow. This tool helps a lot to set up an appropriate dt for double frame or time-resolved PIV recordings. Using this tool in the recording dialog's live view while changing the dt gives the user instant feedback on how good the current dt matches with the flow of the experiment.

- **Particle seeding density**: The image is overlayed with a heatmap that represents the particle seeding density in different areas of the image. The user can specify the preferred seeding density in [ppp] as required by the experimental setup. Areas of the image that match that seeding density are highlighted in green. Areas with lower or higher seeding density are shown in red. Using this tool in the recording dialog's live view while applying the seeding gives the user instant feedback on how good the seeding is.

- **Particle size**: The image is overlayed with a heatmap that represents if the sizes of seeding particles match a specified size. The user can specify the desired range of particle sizes in [pixel] as required by the experimental setup. Areas of the image that contain particles that match that size are highlighted in green. Areas with lower or higher particle sizes are shown in red. Using this tool in the recording dialog's live view while setting up the particle seeding gives the user instant feedback if particles are captured with the ideal size on the camera's sensor. For good PIV data acquisitions particles must not be too small nor to large.

- **Laser alignment and beam overlap**: Use this tool in the recording dialog's live view while setting up the laser beams for PIV double pulse/double frame recordings. Usually, one can see from a high correlation value of a PIV calculation that the laser shots in both correlated images overlap quite well. This is utilized by this overlay tool. The image is overlayed with a heatmap that represents the correlation value in different areas of the image. High correlation values (around a specified value) are display in green, lower correlation values are shown in yellow or red. Please note, that a good correlation value also requires a proper seeding.

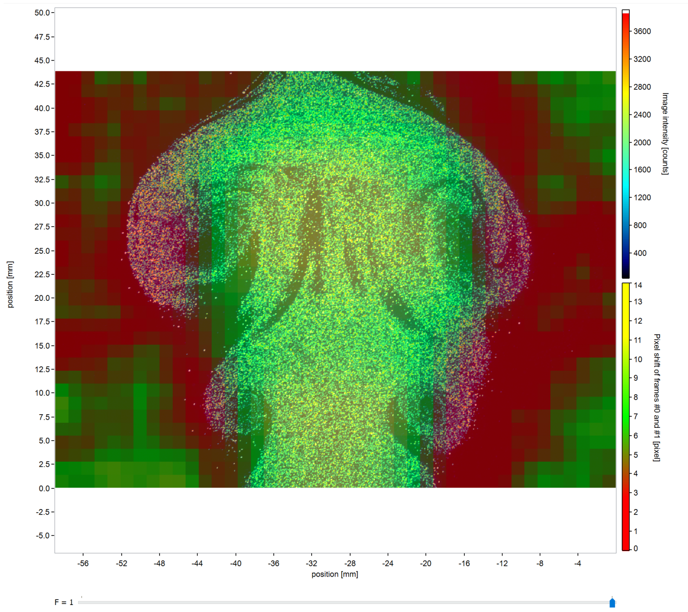
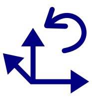


Figure 6.11: Example of the **Pixel Shift** heatmap overlay: green areas have a suitable pixel shift for good results if later PIV is applied on.

6.3 Image - 3D View

6.3.1 Reset view



Dragging the mouse through the view rotates the 3D display. Dragging with pressed shift key moves the viewpoint. Press toolbar button **reset view** to return to the default orientation.

6.3.2 Component visibility

Enable or disable the display of **scales** (x and y), **color scale**, and the **bounding box**. The **coordinate system indicator** can be also activated in order to display the current orientation of the 3D view.

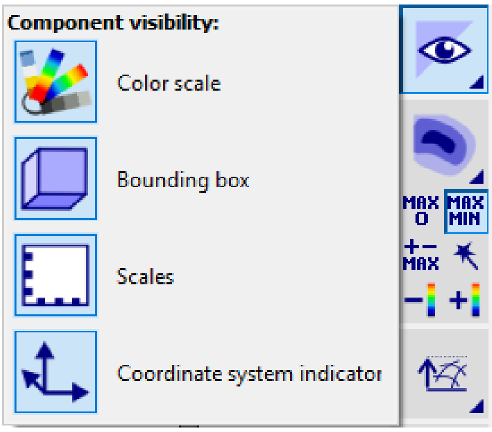
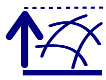


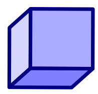
Figure 6.12: Setup dialog for component visibility.

6.3.3 z-Scaling



This button defines the scaling factor of the z-axis. Chose a scale factor between 0.5 and 100.

6.3.4 3D view configuration



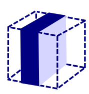
The **3D view configuration** dialog, see figure 6.13 gives access to parameters like the font size of the scalings, the 3D perspective, the bounding box color and the **light**.

The **font size** can be set between 1 and 40pt. Further, chose as **view mode** between orthographic and perspective. The perspective can be changed via the slider.

For each, axes box and view background, a **color** can be chosen. The axes box can be also set transparent.

The **light** can be shown or hidden.

6.3.5 Volume clipping



Within the **Volume clipping** dialog, see Fig. 6.14, the data can be clipped on the corresponding axis by either dragging the slider or by defining the range. The **enable clipping** checkbox allows to activate or deactivate clipping, without modifying the range.

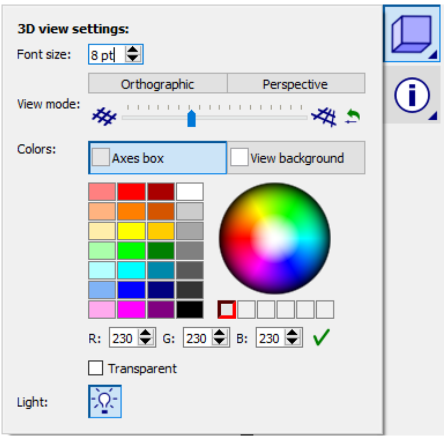


Figure 6.13: Setup dialog for 3D view configuration.

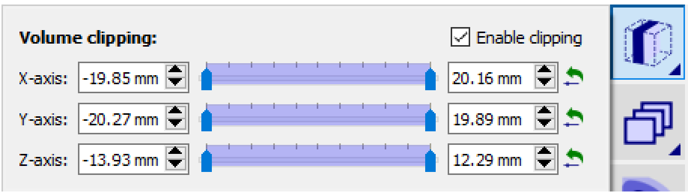
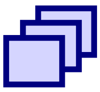


Figure 6.14: Setup dialog for volume clipping.

6.3.6 Volume slice configuration



This dialog allows to visualize data in selected slices. In each tab of this dialog the respective **YZ**, **XZ**, and **XY** planes can be configured. Additionally, scalar slices can be shown or hidden by pressing the button . Choose in **Mode** how many slices, (**No slices**, **Single slice**, or **Multiple slices**) should be visualized.

If **Single slice** was selected, displayed in Fig. 6.15, the appropriate index can be set either manually by defining the index, or by moving the slider. The reset button resets the value. The index can be either set to **Scaled** or **Unscaled**. Scaled defines the unit of the index, while **Unscaled** shows the voxel value. As an example, Fig. 6.17b, for instance, displays the data, when the **single slices** mode was selected.

If **Multiple slices** was selected, illustrated in Fig. 6.16, the appropriate **range** can be set either manually by defining the start and end, or by moving the slider. The reset button resets the value. The corresponding **step** defines the distance between the slices and can be specified via the slider or again manually by defining a value. The index can be either set to **Scaled** or **Unscaled**. Scaled defines the unit of the index, while **Unscaled**

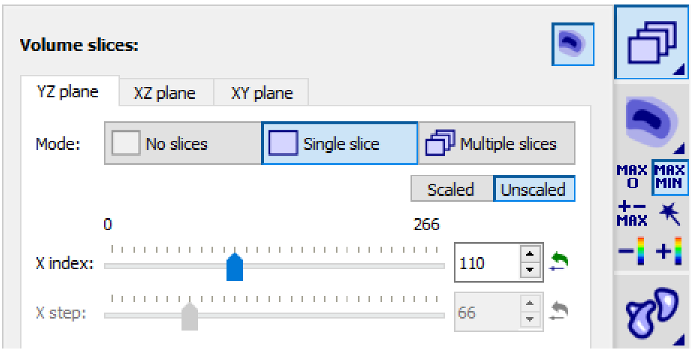


Figure 6.15: Configuration for volume slices. Mode: Single slice.

shows the voxel value. Fig. 6.17c, for instance, visualizes the data, when the **multiple slices** mode was selected.

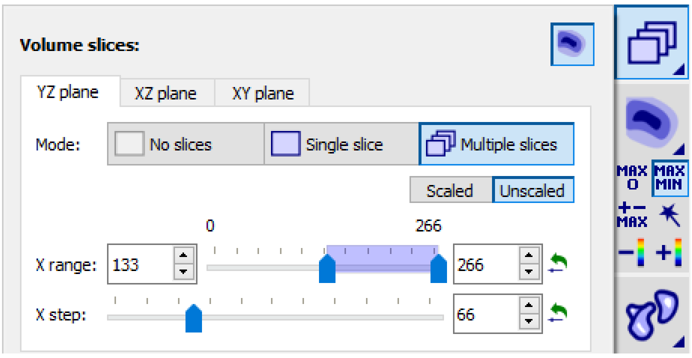
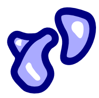


Figure 6.16: Configuration for volume slices. Mode: Multiple slices.

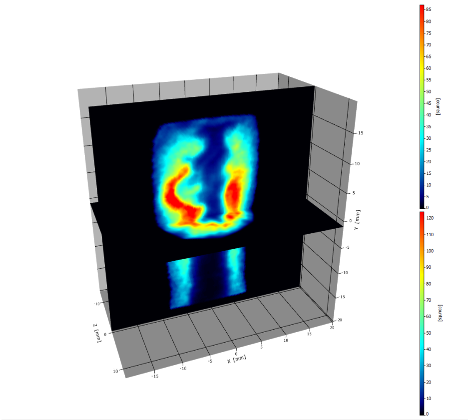
If **No slices** was selected, all settings are disabled, see Fig. 6.18. As an example, Fig. 6.17c illustrates the data, when the **multiple slices** mode was selected.

6.3.7 Isosurface configuration

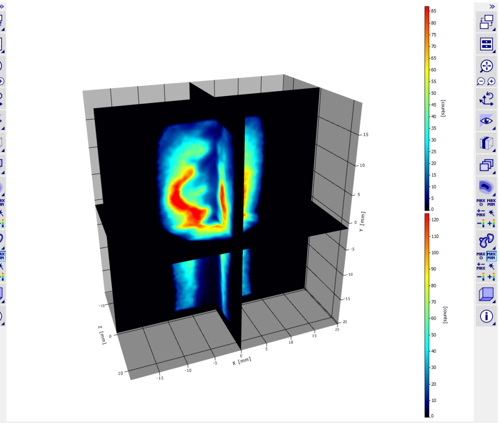


In this dialog surfaces with the same intensity can be visualized. The tab **Iso values**, depicted in Fig. 6.19, allows to configure specific iso surfaces. When **Individual** is selected, the sliders can be enabled via checkboxes. Each slider represents a surface. A surface is defined by the number of counts that can be set manually.

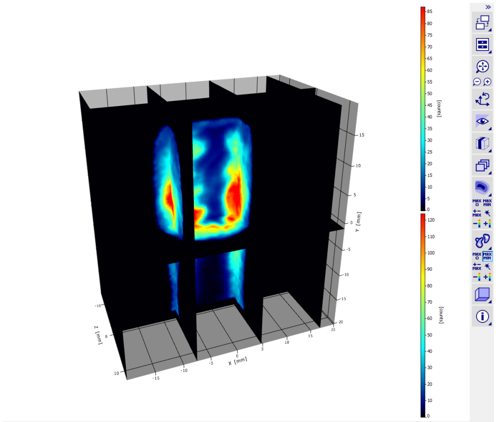
If **Equidistant** is selected, a **range** and a number of **surfaces** can be set, see Fig. 6.20. The number of surfaces can be set between 2 and 8. The specified surfaces are then equally spaced in the defined range. The



(a) No volume slices on YZ plane.



(b) Single volume slices on YZ plane.



(c) Muliple volume slices on YZ plane.

Figure 6.17: Volume slices on YZ plane.

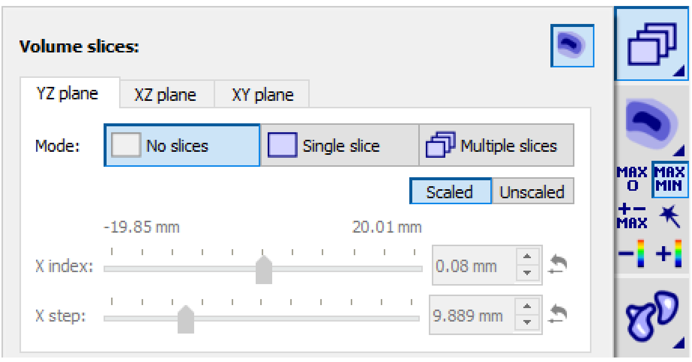


Figure 6.18: Configuration for volume slices. Mode: No slices.

legend below includes checkboxes that give account of the displayed iso surfaces.

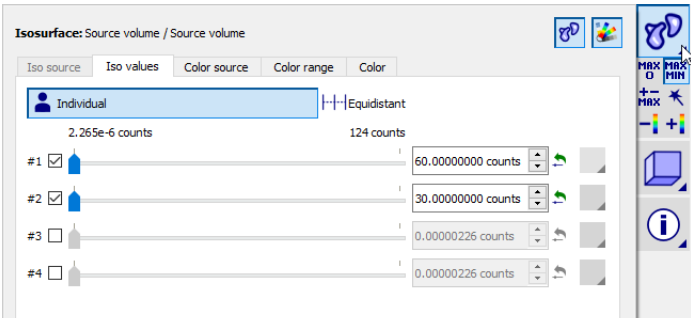


Figure 6.19: Configuration of individual isosurfaces

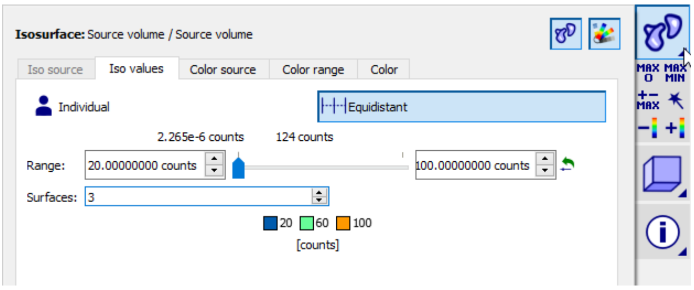


Figure 6.20: Configuration of equidistant isosurfaces.

In the tab **Color source**, the isosurface color can be set to **same as iso value source** displayed in Fig. 6.21. Another possibility is to choose a **Uniform color** or **Source volume**. If **uniform color** was selected, the color range tab is disabled. Following, in the **Color** tab, the uniform color can be defined for each isosurface, see Fig. 6.22a. The **Transparancy** can be modified via the slider.

In case **source volume** was selected, the **color range** tab is enabled and can be modified. Please refer to Section 6.2.2 for more information regarding the color range. Information on the **Color** tab can be also obtained from this section.

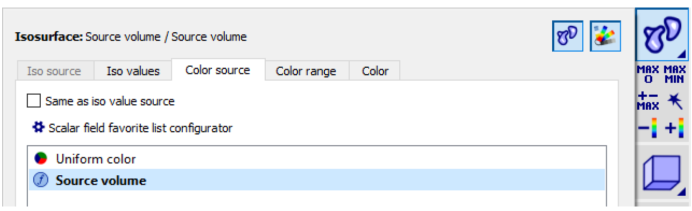
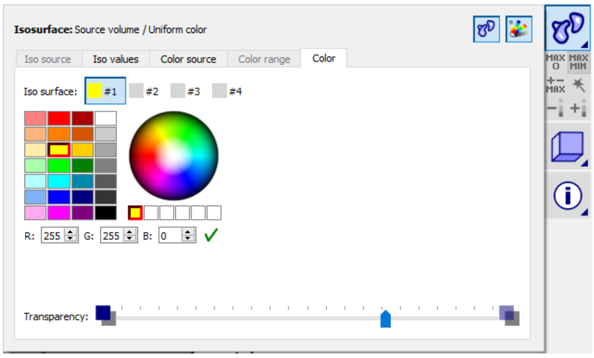
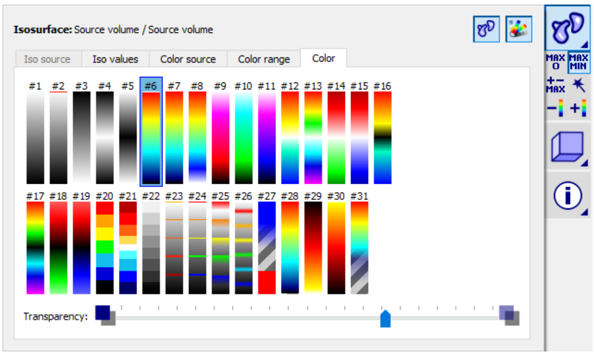


Figure 6.21: Color source of iso surfaces.



(a) Color of isosurfaces.



(b) Color of isosurfaces.

Figure 6.22: Color of isosurfaces.

6.4 Vector fields - 2D View

Some display options of the vector field display are equal or likewise to the general settings, see sections above.

6.4.1 Component visibility

Enable or disable the display of **vector field / grid**, **scalar field**, **source image**, **color scale** of vector field and/or scalar field and **scales**. Furthermore, the **bounding box** and the **coordinate system indicator** can be activated or deactivated. Finally, you can activate or deactivate the **reference vector display**.

The reference vector display can be configured by using its context menu which is shown when the reference vector is clicked with the right mouse button, see figure 6.24.

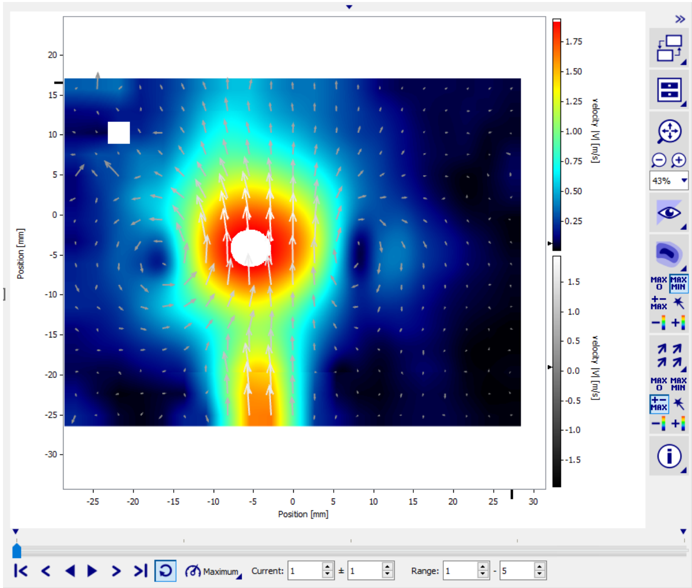


Figure 6.23: Vector field view.

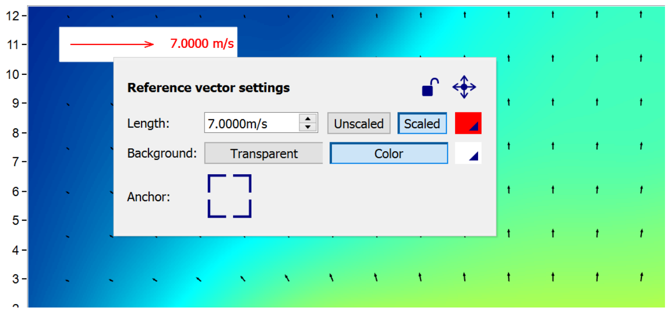


Figure 6.24: Reference vector configuration dialog.

You can define colors, the appropriate length and position of the reference vector.

6.4.2 Scalar field

The scalar field is displayed as background and can optionally be combined with the source image of vector computation. On the top of the dialog, the scalar field as well as the color scale can be shown or hidden. Again the scalar field can be calculated from different **sources**, has its own **range** of intensities for color mapping and its own **color** table. Figure 6.25 displays the source tab. In addition to the sources, a **Unit** and a **Smoothing** factor can be set. The **Interpolation** can be switched off or set as biliniar or bicubic.

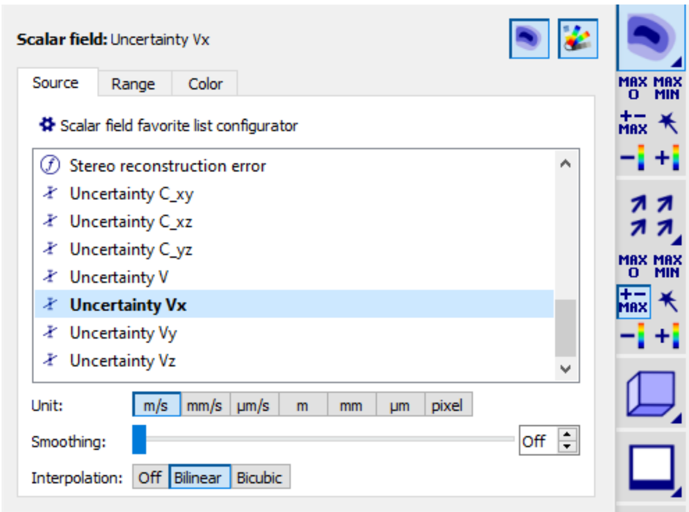


Figure 6.25: Component visibility in Scalar fields.

6.4.3 Vector field settings

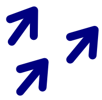


Figure 6.26 shows the **Display** tab of the **Vector field settings** dialog. Here, the changes between **display** of vector arrows or of grid can be set. Depending on the project, also strain axes can be selected. A **horizontal** and/or **vertical skip** can be adjusted. Further, the **length** of the displayed vectors can be changed and the **center** of the vector can be set either on the **origin** or on the **grid**. The **thickness** of the displayed vectors can be modified. If the thickness is set to **zoom dependent**, the vectors are shown thicker the more the view is zoomed. If thickness is **zoom independent**, the vector appearance do not change when zooming.

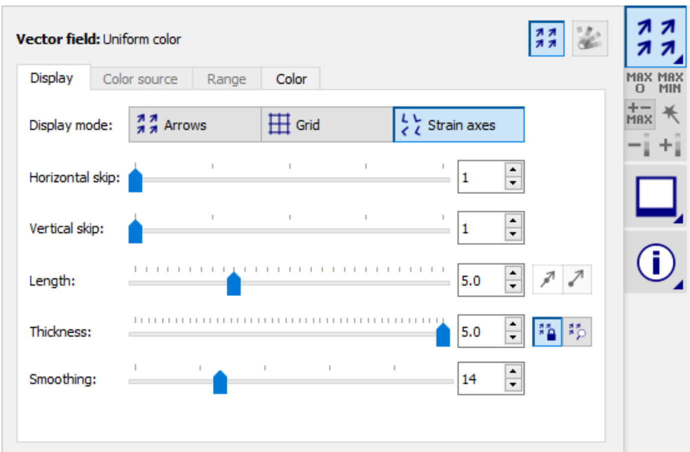


Figure 6.26: Vector field settings.

In the second card about **color source** the different vector directions (Vx, Vy, Vz) and the total length of a vector can be selected. More options can be available depending on the results and parameters of processing functions. The third card is used to define the **range** of the mapping between the values of the vector source and the color table. The **color table** itself can be changed in the last card. The color table immediately changes when the mouse moves above the displayed color tables. Click onto a table to fix the selection.

6.5 Vector fields - 3D View

Most display options of the 3D vector field display are equal or likewise to the 2D vector field settings, see sections 6.4, or to the 3D View image settings, see sections 6.3. Therefore, this chapter provides information about additional 3D vector field configurations that were not described so far.

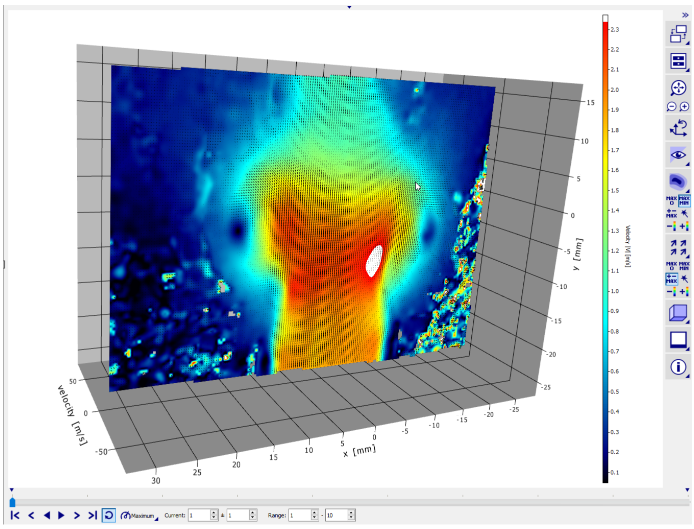
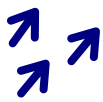


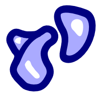
Figure 6.27: 3D vector field view.

6.5.1 Vector field settings



Additionally to the settings described in section 6.4.3, a **z-scaling factor** can be set within the 3D view.

6.5.2 Isosurface configuration



This dialog is already described in detail in Section 6.3.7. Additionally, in the tab **Iso source**, the vector **source** and its corresponding **unit** can be chosen.

In the tab **Color cource** again vector sources are available for displaying the isosurfaces.

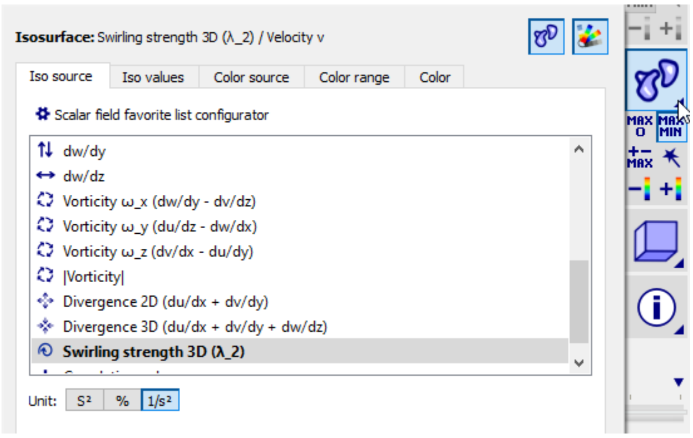


Figure 6.28: 3D vector field view.

6.6 Particle track fields

The result of particle tracking operations, like Shake-the-Box, Particle Tracking Velocimetry (PTV) and Tomographic PTV is a particle field. They are stored in particle field sets and are characterized by the following icon in the project tree view: 

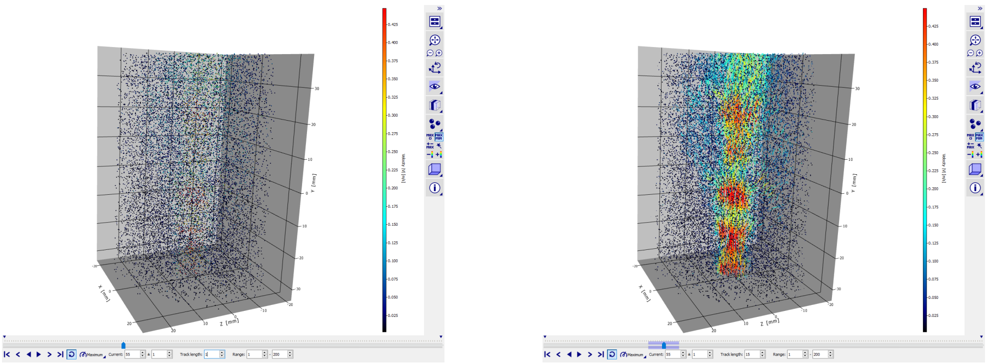
Particle fields can contain data from either time resolved particle tracking or multi pulse particle tracking (2-pulse or 4-pulse). In all cases, a list of particle positions is stored for each time step. Particle tracks are made up by links, that connect particles from different time steps. For time resolved recordings, a particle track can start and end at any time step during the recording. So, particle tracks can have different length for different particles. Some tracks may only exist for a few time steps, others may be present over many time steps. For a single particle, one track may even end at some time step and a new track from the same particle may start again a few time steps later.

Different from velocity or acceleration vector fields, no velocity or acceleration data is stored in a particle field. Velocity data is always calculated when needed using a polynomial fit or, e.g. in the case of two-pulse data, finite differences of particle positions. Also, the velocity data is available at the particle positions and not on a regular grid.

Particle fields can originate from Particle Tracking Velocimetry (PTV), from Shake-the-Box, or from Tomographic PTV. The properties and post processing options are the same independent of the origin of the particle fields.

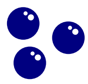
In the case of 2D-PTV the same particle fields are created as for the 3D techniques. In the 2D case however, the z-coordinates of all particles are zero. The same display and post processing options are available as for the 3D cases.

The particle field view in **DaVis** allows the 3D display of particles and particle tracks. For time resolved particle fields, particles from an individual time step can be displayed or particles from multiple time steps at once with a selectable number of particles that are shown at once.

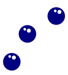


(a) Single time step.   **(b)** 15 time steps.

Figure 6.29: 3D view of single and multiple time steps of particle fields.



The **Particle tracks configuration** dialog allows to set configurations concerning the particles. The first tab **Display** is shown in figure 6.30. One of the following **display modes** can be selected:

-  Show only the lines of the track
-  Show only head particle and lines of the track
-  Show only particles of the track

-  Show all particles and lines of the track

Further, the **Particle size** as well as the **line width factor** can be set either manually via the slider, or by defining a specific value.

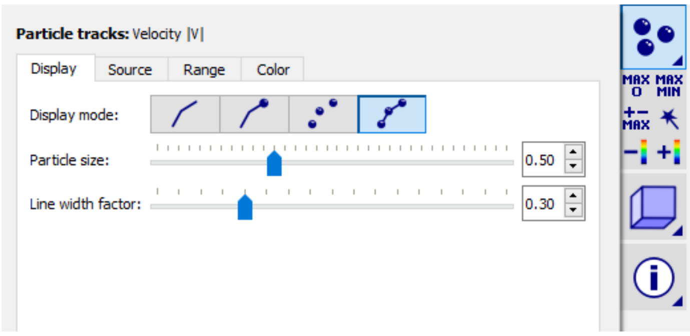


Figure 6.30: Setup dialog for Particle tracks configuration. Display tab.

The second tab **Source** is displayed in figure 6.31. This dialog is used to define the source of particle calculation. Select a **unit** for the axis. Moreover, the **Filter order** and the **Filter length** can be set manually via the slider, or by defining a specific value.

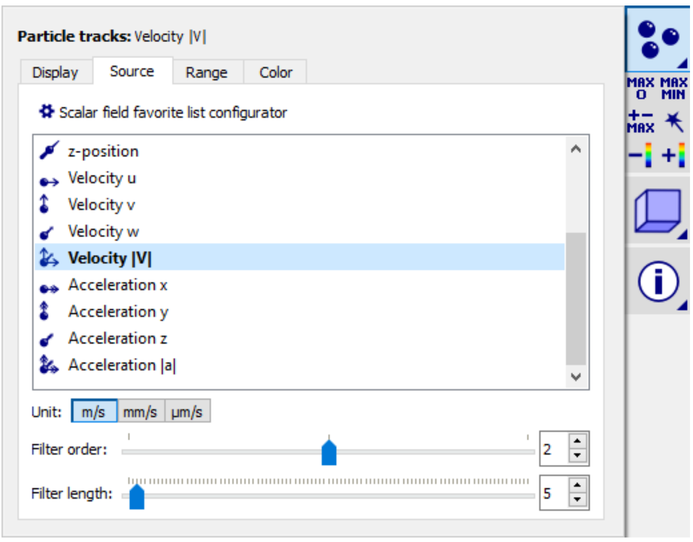


Figure 6.31: Setup dialog for Particle tracks configuration. Source tab.

The **Range** and **Color** tabs are described in figure 6.7 and figure 6.6.

6.7 Plots

Some display options and toolbar icons of the profile view are equal or similar to the general settings options, see sections above.

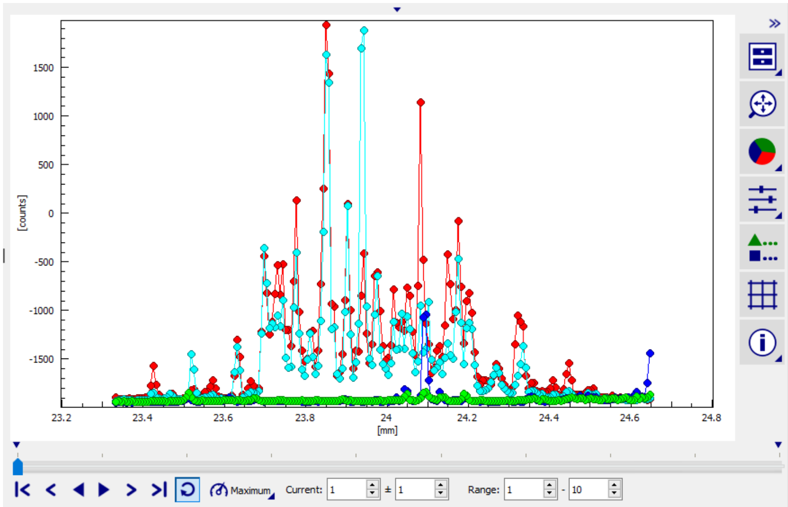
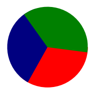
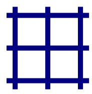


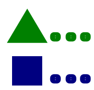
Figure 6.32: Profile view



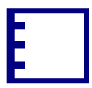
- The background color can be changed by clicking on the icon shown in the margin and then on one of the predefined colors. Moving the mouse cursor over the color matrix gives you a preview: the background color is changed immediately but not fixed.

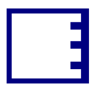


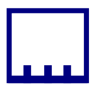
- Show or hide the **grid**.



- Show or hide the **legend**.







- Press the toolbar buttons for **left, right or bottom axis** to change the value displayed on this axis. Additionally the displayed value range can be changed or entered manually and the scaling can be switched between linear and logarithmic.
- When the mouse is moved above the plot view, the plot item next to the mouse cursor is highlighted as a thick circle. The **position info label** gives x- and y-positions of the highlighted plot item.
- Data reduction on XY plots is important in case of very long plots, e.g. for AD traces from highspeed recordings with millions of data points.

To increase the speed of display the number of displayed points is reduced down to the order of the screen size.

6.7.1 Probability density function plot

This special plot is used to detect peak locking of the recorded images by vector field computation. The processing operation described on page 316 is used to calculate the plot.

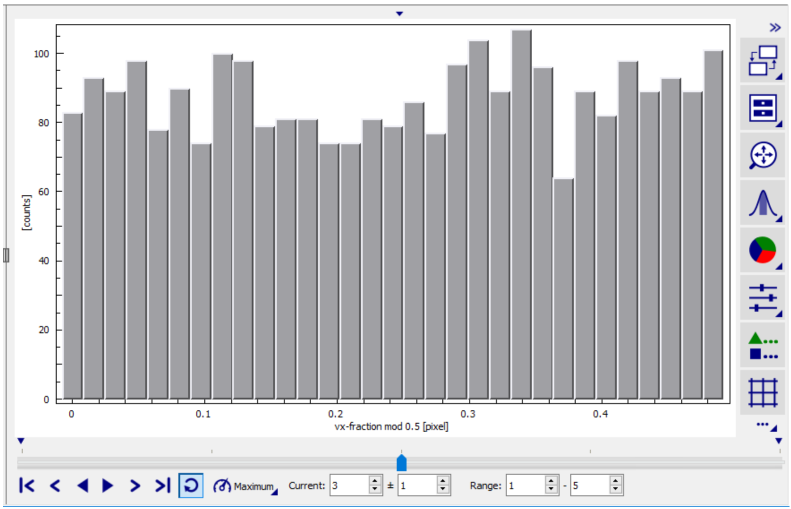


Figure 6.33: Probability density function plot of a vector field.

Press the toolbar item to open the display settings dialog of figure 6.34. Select a **Component** to be extracted for all vector fields in the set.

Select the component displacement via the option **mode**:

- **V mod 0.5** defines the fraction mod 0.5 in the range of 0 to 0.5.
- **V mod 1** defines the fraction in the range of 0 to 1.
- **Complete V** defines the whole range in steps of 1 pixel.

The number of plotted bars can be set via the **pixel** slider.

6.7.2 Scatter plot

The scatter plot displays the distribution of vector direction and length in Vx- and Vy-direction.

Press the toolbar buttons for x- and y-axis to change the displayed value on this axis (Vx, Vy, Vz) or specify a range of the plot area.

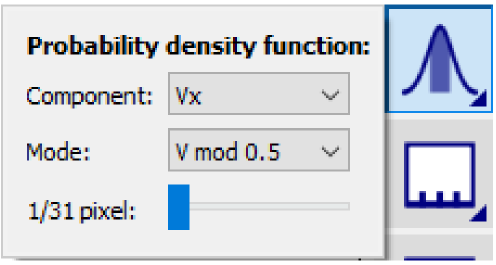


Figure 6.34: Display options for probability density function plot of a vector field.

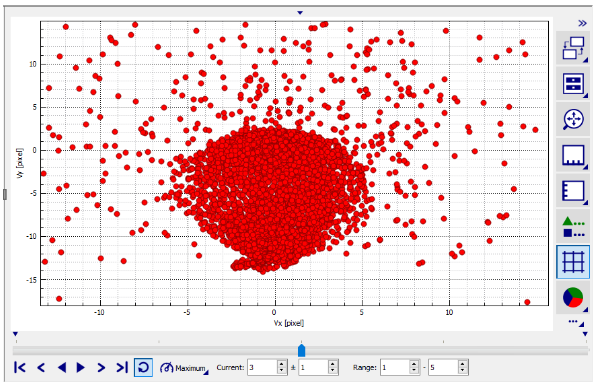
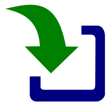


Figure 6.35: Scatter plot of a vector field.

7 Import



The **Import** dialog is reached via the **Import** button in the **Project Manager** toolbar. Using this dialog you have the possibility to import datasets or files to the current project. **Note**: The files will be copied and not moved!

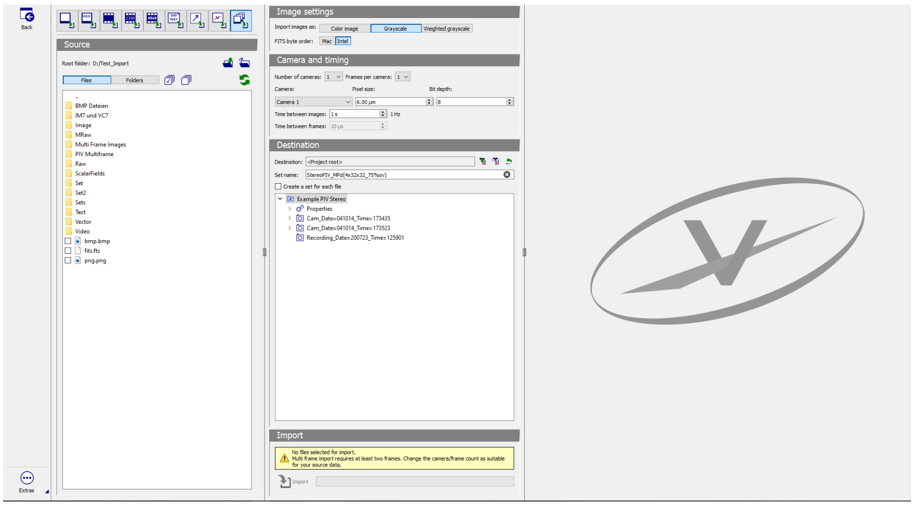


Figure 7.1: The **Import** dialog

The dialog is divided into five areas:

1. The top left area gives some buttons to switch between different source types, e.g. bitmap files, movies or raw data files. When holding the mouse cursor without movement above a button, a tooltip appears and gives a short description about the available source type.

2. The left area presents a treeview of the file system to change the source folder.

3. The upper area of the column in the middle of the dialog allows to change some special settings for the selected type of source data.

4. The lower area in the column in the middle displays the tree of the active project. Here the destination of the import can be defined.

5. The right area, which displays the **LaVision** logo when opening the dialog, is used as preview for the selected source file.

7.1 Select source file(s) or folder(s)

The tree with the selected folder of source data allows to navigate within a drive only. Press the left button above the tree to open a fileselectbox to change to another folder and especially to change the drive.

Click to an item of the file list to display the file in the preview area. Mark the item for import by clicking the checkbox on the left side of the item.

Switch between **Select files** and **Select folders** to import either single files or complete folders with all files of the selected type. Depending on the mode either the files or the folder can be selected via checkbox on the left side of the list.

To select or deselect a range of items at once please click to the first item, then press the `shift` key, click to the last item, then click to one of the checkboxes in the selection list. To select all items at one, press the keys CTRL + A and then click into one of the checkboxes.

7.2 Source types

The following source types are available and allow to import some file types:

- **Image** for bitmap and data files of types `bmp`, `jpg`, `png`, `tif` and `tiff`, `fits` and `fts`, the **DaVis** formats `im7` and the old formats `img` and `imx`.
- **Raw image data** for dumped raw data files from some camera vendors: `raw` (byte format), `raww` (word format) and `raf` (float format).
- **Video** of type avi.
- **Text file importer** of formats `txt`, `csv`, `dat` and `xyz` e.g. with Tecplot or Matlab data.
- **Vector fields** of **DaVis** formats `vc7` and the old format `vec`.
- **DaVis sets** of format `set` and the corresponding folder.

- **Multi frame** for pairs (or quadruples) of image files to be stored as frames in a single image file.
- **Volume** for multi folders, each folder creating one image. Available since **DaVis** 10.2.1.

7.2.1 Image import

The settings of **Image import** give access to three options for colored bitmap images:

- Import real RGB (red/green/blue) images, which can be useful for background images, but in most cases not for calculations.
- Simply average the three color components into an intensity value.
- Convert the color into an intensity value via a function, which uses different factors for each component and conserves the gray level of each pixel.

7.2.2 Raw image import

Raw data files give no information about the image size. Therefore the image size must be specified by **width**, **height** and **frames**. The number of **planes** can be defined since **DaVis** 10.2.1, see figure 7.2. The number of frames is defined via camera parameter. If the given size is larger then the amount of data in the file, the file cannot be read and the **LaVision** logo is displayed instead.

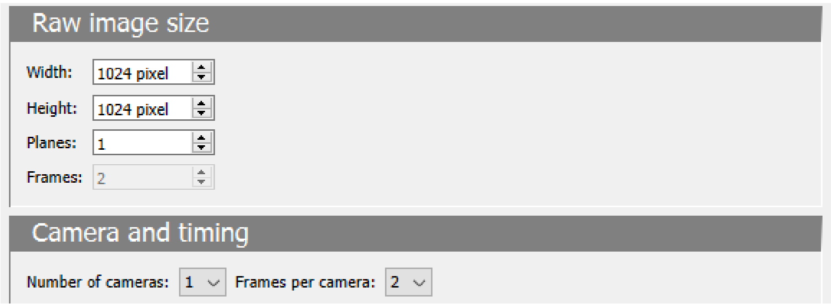


Figure 7.2: Raw image import

7.2.3 Text file import e.g. from Tecplot or Matlab

The file types **dat** and **xyz** are well defined and can be read without any options.

The file types **csv** and **txt** usually contain kind of a header followed by the data in clear text style with a separator. Therefore the number of lines in the **header** and the separator can be defined in the parameter dialog, see figure 7.3.

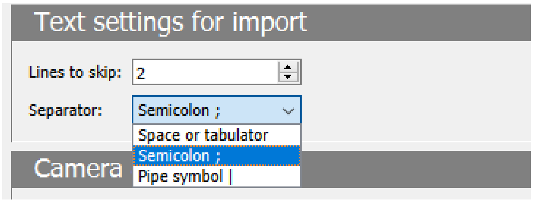
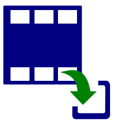


Figure 7.3: Import parameters of csv and txt files.

7.2.4 Video import

Videos can be imported the same way as colored images, see above. Additionally the number of frames in the AVI file is displayed and range and increment can be defined.

The most important error messages of video import are:



- **Video import disabled:** The video import needs a special library from the graphic device driver: `OpenGL.dll`. If the movie button is disabled, then please update your graphic device driver!
- **Missing codec for AVI:** If the AVI import stops with an error message about missing codecs, **DaVis** is running on a 64 bit system and the Windows Media Player is able to displays the AVI, then there could be a problem with corrupt 64 bit codecs. In this case either the conversion of the AVI into another codec like **Microsoft RLE 1** helps, or the installation of another codec library like:
  `http://www.codecguide.com/klcp_64bit.htm`

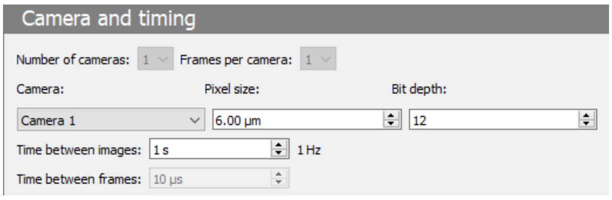
7.2.5 Volume import

Select a folder with several child folders. The image files from one child folder are read into a multi plane image. The result of this import is a set of multi plane images. Both child folders and file names should be sorted in alphabetic way, e.g. names with an index. There is no way to change the order during import.

7.3 Camera and timing

This parameters are used to define special buffer attributes describing timings and other information about a recording. These attributes are very important for many operations in **DaVis** and must be added after importing files without these attributes (e.g. BMP and TIF).

In all normal import types only one camera (with one frame) can be defined. The special **Multi frame import** gives access to more settings, see section below.



The following buffer attributes are set:

CameraName#n: Camera number of frame *n*. Used to identify the corresponding camera number (needed for calibration, image correction etc.).

AcqTime#n: Arrival time of frame #*n* within one acquisition (one image). Used to compute the time between two frames (vector computation, general time series operations).

AcqTimeSeries#n: Arrival time of image #*n* within one recording (one data set).

Pixel size#n: Camera pixel size in µm. Used in the calibration to get correct pinhole parameters. If this value is set to a wrong value, all pinhole parameters that are scaled in mm, such as like focal length or camera–plate distances, will be wrong by the same factor. This will *not* affect the quality of the calibration! So if the camera pixel size is unknown, just enter 5µm, which is a common value of CCD chips.

Bit depth#n: Bit depth of the image can be set between 8 and 16 bits.

Please note: Existing attributes (like when importing old im7 files) will not be changed!

7.4 Multi frame import

Use this mode to import single files as multi-frame images. The format of the files must be: $<$ *prefix* $><$ *number* $><$ *frame* $>$. $<$ *ext* $>$

The file names have to start with the same prefix characters, followed by a number. The $<$ *frame* $>$ part identifies the different frames.

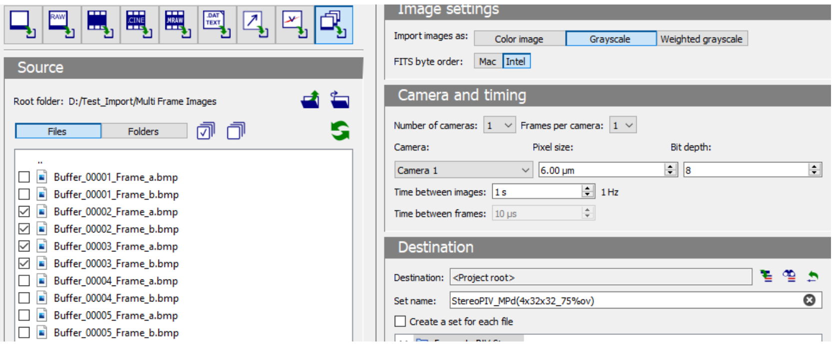


Figure 7.4: Multi frame import

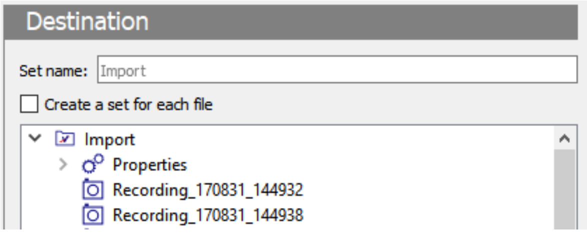
In the example in figure 7.4, the files `Buffer_00002_Frame_a.bmp` and `Buffer_00002_Frame_b.bmp` are imported as first and second frame of the first image, and `Buffer_00003_Frame_a.bmp` and `Buffer_00003_Frame_b.bmp` are combined into the second image.

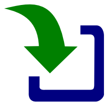
The **Number of cameras** and **Number of frames** must be defined and then the **wildcards** for all frames must be given. When these values are entered, a click to one of the selected source images loads all virtual frames of the selected file. E.g. when clicking on `Buffer_00002_Frame_b.bmp`, the file `Buffer_00002_Frame_a.bmp` is also loaded and both frames are displayed in the preview.

7.5 Destination and Import

The tree gives an overview about the folder structure in the active project. Either import new data in the root folder of the project or open and select folder as destination of the import. By default the imported data is stored in a new set of name **Import** but this **Set name** can be changed if wanted.

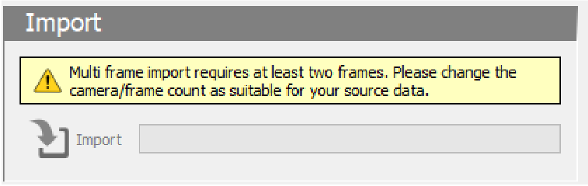
Optionally with **Create a set for each file** all selected files are stored in individual sets with the name of the source file.





The **Import** button in the lower part of the dialog starts the import process.

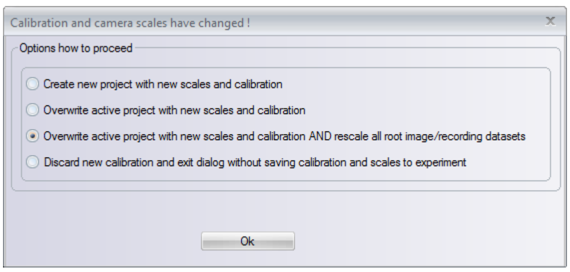
The yellow section next to the **Import** button gives information about problems or incomplete settings. In case of the following example the number of cameras and frames in the **Camera and timing** dialog does not fit to the number of defined frames for multi frame import.



7.6 Next steps after import

7.6.1 Scaling

When the import has been finished a new scaling should be created for the imported images. Therefore the **Scaling Wizard** is used. See chapter 17 for a detailed description. For the scaling process some suitable images are needed. They have been recorded in the measurement plane and provide information on a know distance.

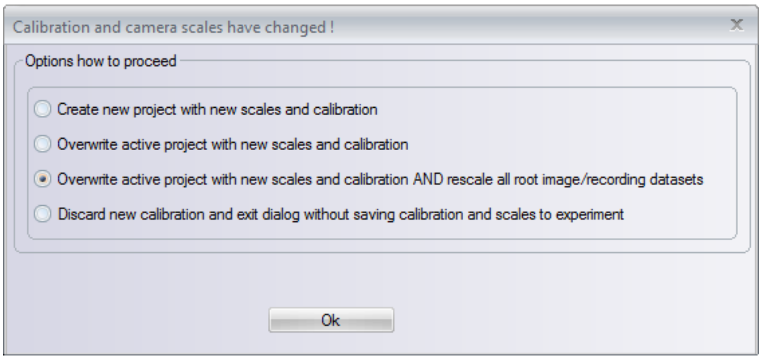


After the scaling is finished, it is very important to use the **Overwrite active project with new scales and calibration AND rescale all root**

image/recording datasets option in order to make sure that the new imported pictures get rescaled.

7.6.2 Calibration

The next step creates a new calibration for the imported images – see chapter Calibration about the **Calibration Wizard** for detailed descriptions. For the calibration process some suitable images are needed. They have to be recorded in the measurement plane and provide information on a known distance.



After finishing the calibration, it is very important to use option **Overwrite active project with new scales and calibration AND rescale all root image/recording datasets**, to make sure that the new imported images are rescaled.

7.7 File Formats

DaVis supports the input and output of the following data formats. The standard formats are `IM7`, `IMX`, `IMG`, `VC7`, `VEC` and `PRF`, which are the only formats to include the complete buffer information defined by the image acquisition functions and by the calculation functions of **DaVis**. If a buffer is saved in another file format, some information is lost!

The standard formats are supported by the **Processing** dialog, **Recording** dialog and by the **Project Manager**. Foreign images of other formats must be imported into **DaVis**.

7.7.1 DaVis File Types for Images and Vectors

IM7

This is the default storage format for all types of image buffers since version **DaVis** 7. All buffer information are included in this file type. The format got new features in **DaVis** 8. Older version may not be able to read all additional data from the file, but image and vector data can be read even in old versions.

The data can be compressed, and the compression mode can be selected in the **Preferences** dialog, see page 56. Different compression algorithms are implemented for a short storage time or to get a small file size.

IMX

This is the default format for image buffers in the old versions **DaVis** 5 and 6. For WORD buffers the data is compressed. FLOAT buffers are always stored uncompressed, no matter if the extension is `IMX` or `IMG`.

The data compression scheme is very fast and efficient. Therefore, there is no reason to avoid the compressed format, unless you would like to use your own software for data analysis. On the **LaVision** website, C code is available for reading `IM7`, `IMX`, `IMG` and `VEC` file formats.

IMG

Used for uncompressed data storage by the old software versions **DaVis** 5 and 6. After a header of 256 bytes, the data is stored in binary format (not `ASCII`) line by line from the top left to the bottom right of the buffer as unsigned integers (2 bytes each pixel, 0-65535) or, for FLOAT buffers, as single precision floating point numbers (4 bytes each pixel).

The raw data table is followed by a coded list of the buffer attributes (strings and arrays), scales, and the comment.

VC7

This is the default storage format for vector buffers, including the complete buffer information. The format is very much equal to `IM7` and the data compression can be used.

VEC

Used for vector data by the old software versions **DaVis** 5 and 6.

7.7.2 General File Types for Export and Import

TXT

It is possible to store and load the contents of an image or vector buffer in the **ASCII** text format (*A* merican *S* tandard *C* ode for *I* nformation *I* nterchange). No comments, scales or other additional data will be stored.

Some parameters of the export can be defined in section **Export** of the **Preferences** dialog, see page 58:

- decimal *point* character: comma or dot
- *separator* between values: tabulator, space character, or carriage return
- precision of exported float values: 0 – 15 digits

BMP

There is also a data transfer support for bitmaps. Images can be loaded in the Windows bitmap format `BMP` (in 16, 256, 65536 or full colors) with the help of the **Import** dialog. The `BMP` format does not support storing attributes, scales or comments.

DaVis only supports storing image buffers in the 8 bit format. This means to lose lower significant bits. Do not use `BMP` to save images acquired by a camera which supports more than 8 bits if you want to use the high-resolution data for later computing!

When loading `BMP` files into **DaVis** buffers, a conversion mode is used, which can be selected in the **Import** dialog.

B16

A 16-bit bitmap format which supports long comments, but no attributes.

JPEG

The *Joint Pictures Experts Group* format is lossy, meaning that the output image is not exactly identical to the input image. The JPEG writing and storing algorithms in **DaVis** are based in part on the work of the *Independent JPEG Group*. For more information see

ftp://ftp.uu.net/graphics/jpeg/jpegsrc.v6b.tar.gz

PNG

The *Portable Network Graphics* format can be used to store RGB color images and 16 bit grayscale images.

TIFF

Tagged Image File Format is an uncompressed 8- or 16-bit color format and does not store comment lines or other attributes. **DaVis** can import grey and color (RGB) TIFF images and color palette images. Exported TIFFs are grey images only. **DaVis** does not support compressed TIFF images. Please import uncompressed files only.

FITS

The Flexible Image Transport System stores comment lines, but no attributes. The so-called Fits Byte Order can be switched between **MacIntosh like** and **Intel Byte Order**.

DAT

The `DAT` format is used e.g. by Tecplot. **DaVis** is able to read and write data files for Tecplot of multi-plane and multi-frame buffers, both image and vector type.

AVI

Audio-Video Interleaved movies can be imported. During import the color tables (RGB) can be mapped into grayscale pixel values. See the chapter about importing AVI on page 134 for more information.

RAW and RAWW

The 8 bit `RAW` and 16 bit `RAWW` image data formats are created by some external image acquisition programs. Row per row and from left to right in each row the pixel data are stored without additional information. The image size (width, height, depth and frames) must be entered in the parameter area of the **Import** dialog because the size is not specified in the file itself. When **DaVis** stores such a file, all planes and frames are stored.

7.7.3 Special File Types for Profiles, Sets and Overlays

PRF

A single profile is stored in an ASCII file format. This profile format is the only file format to be loaded into the **Profile** window. In the first line the number of data points and the scale value of the first and last point are stored. The data points follow line by line.

PRM

One or more profiles are stored in an ASCII file format. This format can be read e.g. by Microsoft Excel. Each profile is stored in a different column. A column gives the scaled values of the profile.

SET

DaVis uses `SET`s to store parameters and data files as “single objects” . For example all acquisition parameters and the acquired images are stored as a `SET`. All dialogs are supporting the `SET` structure to make the usage of large numbers of files easier, e.g. the movie player of the **Project Manager** and the **Processing** dialogs.

With the help of SETs the user does not need to care about the location of parameter files and about the image files. The dialogs only ask for a SET as input and another SET as output, and then the dialogs manage the SET creation and file storage automatically.

The SET file itself is a simple ASCII text file and lists all static CL-variables of the included groups of variables. Additionally this file can include a comment string, the storage time and a type flag, which is used e.g. by Projects. A subdirectory with the same name as the SET file includes the image, vector or profile files:

```
MyExperiment.set
MyExperiment/B01.im7
MyExperiment/B02.im7
MyExperiment/...
MyExperiment/B17.im7
```

In this example we have a SET with the name MyExperiment, and the SET includes 17 image files of type IM7.

Stream Set

A **Stream Set** is the **DaVis** file structure for recordings and processings since **DaVis** 10 with some basic support in **DaVis** 8.4. The base infomation are stored in the SET file also. The folder of the same name does not contain one file per image but one or more files per device: During recording each camera stores its images into one file. This increases the writing speed during recording and allows to use own optimized file formats of camera manufacturers.

OVG

The OVG files don't include any image information but overlay graphics. This graphics can be edited in the **Overlay Editor**, see manual **Tool Dialogs** for more information. The overlays of a complete buffer, of a single frame or of a range of frames can be stored in OVG files and reloaded. Whenever the **Project Manager** displays a SET of OVG files, the manager searched for a source image and displays the image together with the overlay.

7.7.4 Reading DaVis File Types with Other Software

The **DaVis** file types `IM7`, `IMX`, `IMG`, `VC7` and `VEC` and the stream sets can be read from other software like MatLab®, Python®, LabView® and own C/C++ programs.

A `ReadIMX.DLL` can be downloaded from the **LaVision** website. The complete source code of this DLL is available in C language and can be compiled on Windows, Linux and Mac systems. The library includes functions to be called by LabView or own C/C++ programs to read image and vector data and the attributes. Examples for LabView and in C code are available. The version of the `ReadIMX` library is old and will never be updated again. Stream sets from recordings of **DaVis** 10 can't be read by this old library.

For **MatLab**, a special DLL can be called from some example M-files and is available for Windows and MacOS. The source code of this `ReadIMX-Matlab.DLL` is not available. But the MatLab library is updated together with **DaVis** and allows to read images from all of the above listed formats and also from the recorded stream sets.

For **Python**, libraries and example code are available for Windows and Linux.

8 Export

This chapter gives an overview how to export data and views in **DaVis**. We distinguish between

- **Screenshot/movie export** (e.g. `.png`, `.mp4`) of a view, which is available in the **Export dialog** (see section 8.1) and via context menu on the view depicted in Fig. 8.1 and
- **data export** (e.g. `.dat`, `.csv`), which is available as operation in the processing dialog and via the context menu on the view (see section 8.2).

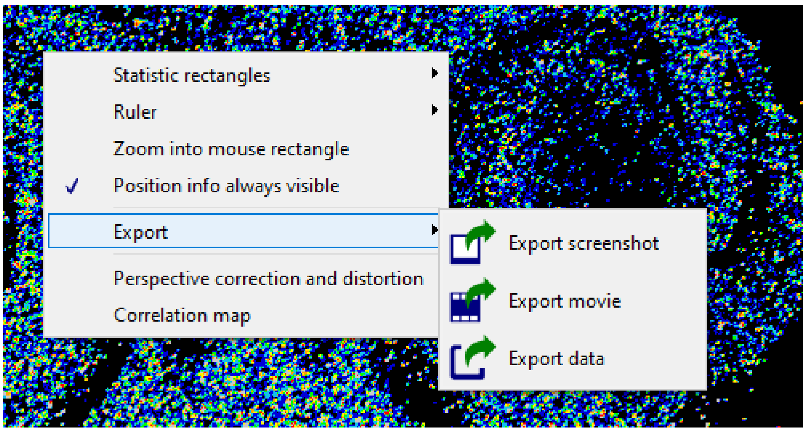
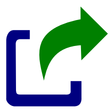


Figure 8.1: Open the **Screenshot Export** dialog via context menu.

8.1 Export Screenshot and Movie

The **Export** dialog (see Fig. 8.2) can be used to export data views as screenshots or movies.



The dialog can be reached using the **Export** button in the **Project Manager**. The dialog is able to export multiple sets in one step. Therefore the sets must be selected in the **Project Manager** using multi selection before opening the dialog.

8.1.1 Overview

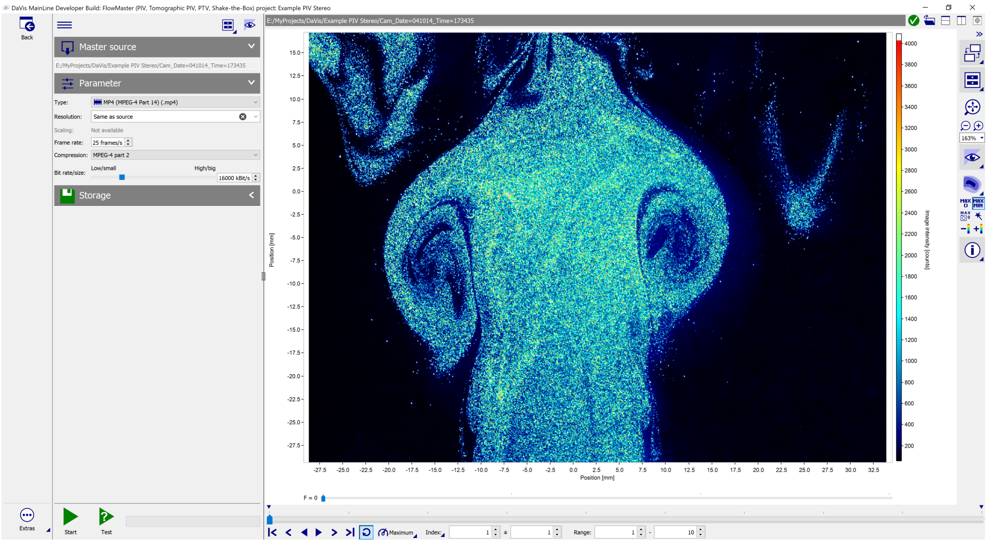


Figure 8.2: Export dialog for exporting data views.

The dialog is subdivided into two parts. The left part contains the sections for the export **Parameter**, the storage **Options**, and the **Start** and **Test** button. The right part which is utilized to prepare the data view(s).

The first step after opening the dialog is to set up the data view(s) in the right part. The **Preview** button can be used to show how the exported view looks like (without player and controls) before starting the export.

After setting the export parameters and the storage options the **Start** button will export the set (or all sets if multi selection is used) to the specified folder. The selected range and increment in the player control can be used to export only a part of the set (see Fig. 8.3):

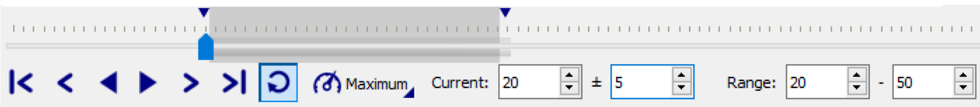


Figure 8.3: Export range.

8.1.2 Parameter

The following parameters can be set:

- **Type**: Choose between screenshots (`.bmp`, `.png`, `.jpg`) and movie file formats (`.avi`, `.mp4`). In screenshot mode the export creates a single file for each element in the selected set. If `.avi` or `.mp4` is selected, the export creates one movie file for each set.

- **Aspect ratio/Resolution**: Select the aspect ratio and/or the size of the exported screenshot or movie. **Note:** The **aspect ratio** setting is removed in **DaVis** 10.2.1. Either select a predefined value or enter the resolution.
  If **Same as source** is selected, the export dialog uses the aspect ratio and size of the displayed image or vector field. If a 3D view is displayed, the aspect ratio of the view is used.
  Mode **original image size** is available for 2D images only since version 10.2.1 and uses the real pixel size on screen. By disabling the view addons like scales and colormap the image can be exported 1:1 from screen. In this mode the export can be increased by the given scaling factor.

- **Frame rate**: Select the frame rate of the video.

- **Compression**: Depending of the selected **Type**, different compressions can be selected for this container.

- **Quality/size**: The quality of the resulting files can be defined as a value between 1 and 100. For more information about this value please refer to the corresponding definition. A quality of 100 percent is not lossless and a value of 33 percent does not create a file with a size of one third of a lossless file. A very good value is 80 percent.

- **Export full range**: This option is available for multiple source sets (hyperloop or multi selection). Use this option to export the full data range of all selected data sets. This option is useful, if the number of files differ between the selected data sets.

The settings shelf button  stores the active dialog parameters and can be utilized to load previously saved parameter settings. More detailed information can be obtained from section on page 105.

8.1.3 Storage

Please refer to page 153 for the storage process of the exported screenshot or movie.

8.1.4 Side by Side Export

The display can be subdivided horizontally and vertically into multiple views, as displayed in Fig. 8.4, by the buttons and , respectively. Therefore, a further required dataset can be loaded by clicking the button . The button closes the corresponding view.

To export the multiple data views, one view has to be selected as a master source with the button . The screenshot or movie is then exported within the defined range of this master source view.

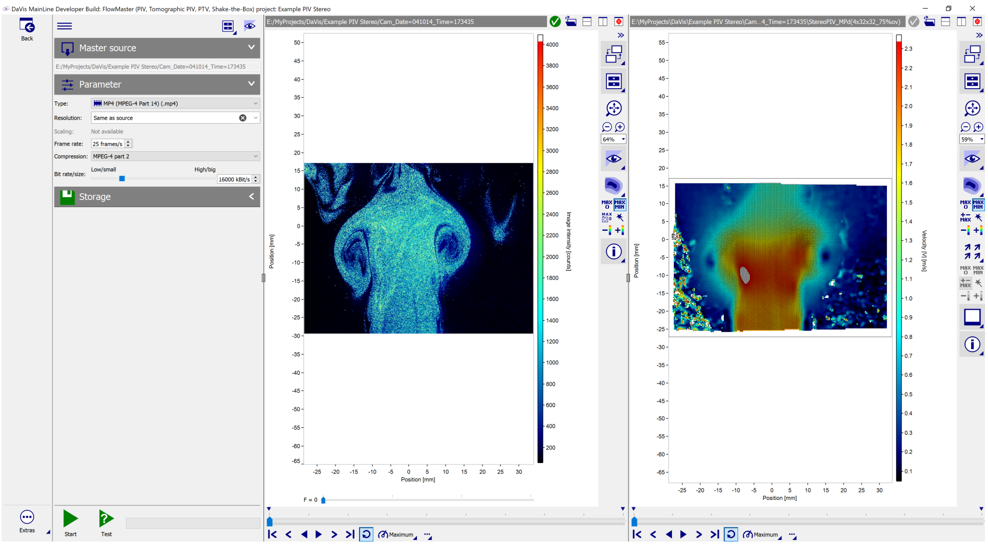


Figure 8.4: View of multiple datasets.

8.2 Export Data

8.2.1 Purpose and Location

The Export Data Dialog is the user interface for exporting images and vector fields into other file formats. Some of the target file formats are text formats (e.g. csv), others are binary formats (e.g. tif). The Export Data Dialog is available for images and for vector fields only and can be accessed in two ways. First, via the context menu of an image or vector field view, depicted in figure 8.5 and second via the processing dialog, illustrated in figures 8.6 and 8.7, respectively.

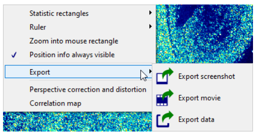


Figure 8.5: Access via context menu.

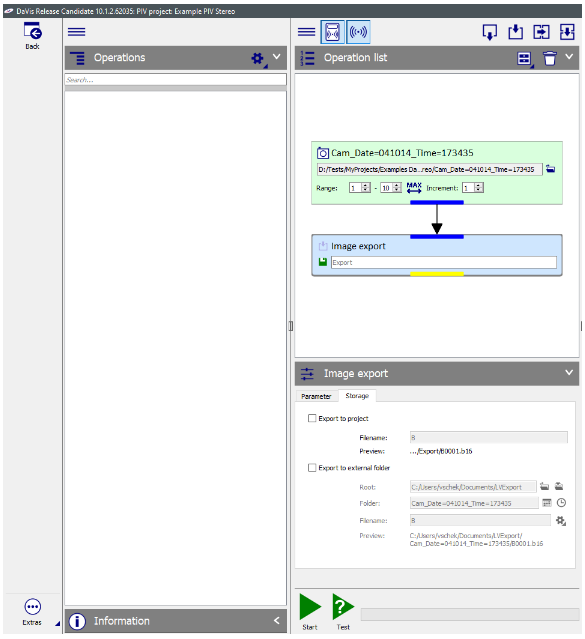


Figure 8.6: Access via processing dialog (image).

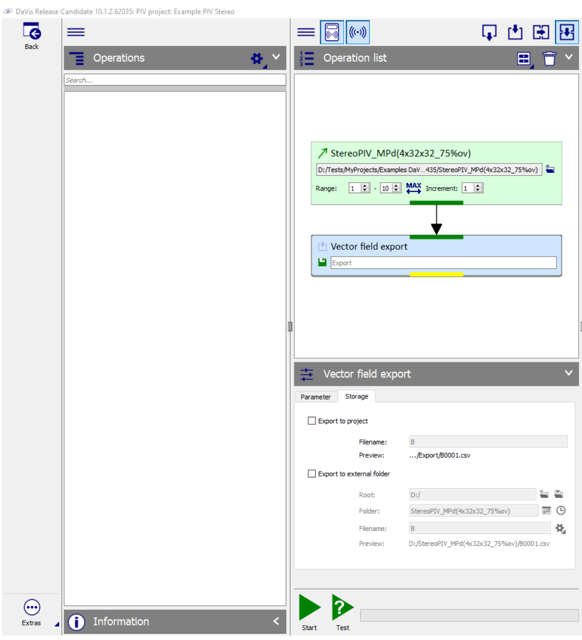


Figure 8.7: Access via processing dialog (vector field).

8.2.2 Export Data Dialog

The Export Data Dialog contains three sections, **Range**, **Parameter** and **Storage**.

Figure 8.8: The Export Data Dialog.

The "Range" section

The "Range" section is meant to define the range (minimum, maximum and increment) of the set that is exported. This section does not appear in the processing dialog because here the range is defined in the source operation item. The section also does not appear, when the dialog is opened via the context menu of a view that displays a single image or a single vector field. Since there is no range to define in these cases, the "Range" section would make no sense.

The "Parameter" section

The "Parameter" section defines the export format. There are different formats available for images and vector fields.

Export formats for images

For images the following export formats are available:

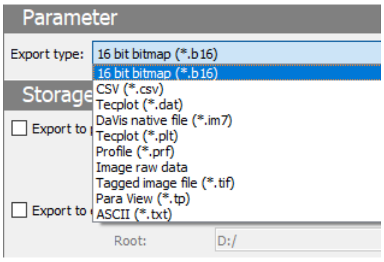


Figure 8.9: Export formats for images.

Most of these formats are trivial and do not require or offer any further parameters. All formats with parameters are described here:

When **Image raw data** is selected, the **Parameter** section will show additional parameters, where the data format of the exported data can be selected. The selected data format also defines the extension of the export file(s) (byte → `*.raw`, word → `*.raww`, float → `*.raf`).

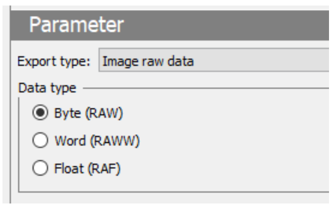


Figure 8.10: Image raw data parameters.

The **TIF** file export allows to convert image data of float data type into word data type before exporting. As a word data type the `TIF` contains 16 bit integer values and but 32 bit floating point values as a float data type.

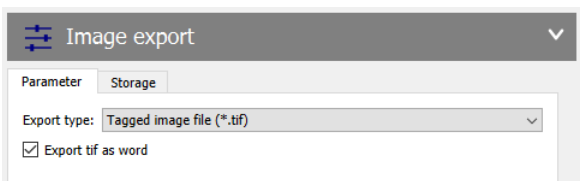


Figure 8.11: TIF export parameters.

Export formats for vector fields

For vector fields the following export formats are available:

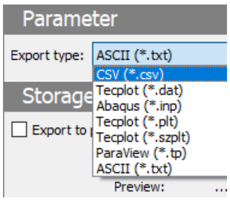


Figure 8.12: Export formats for vector fields.

Unlike the export types for images, only two of the export types for vector types are trivial and have no additional parameters: CSV and Para View. All other export types have additional parameters.

Tecplot

The Tectplot export format is available three times in the list, the differences for the additional options are only minor. The most obvious difference is the extension of the export file(s). While `*.dat` is a ASCII-based format, the extensions `*.plt` and `*.szplt` represent two binary formats. For further information about these formats please refer to the "Data Format Guide" of Tecplot.

When either of the Tecplot export formats is selected, the dialog displays the following additional parameters:

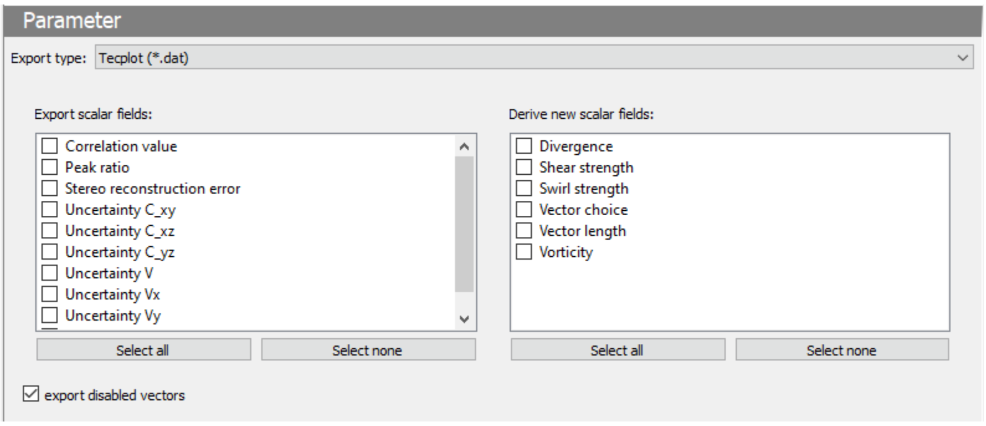


Figure 8.13: Tecplot parameters.

In the left list all existing scalar fields can be selected. If a scalar field is checked, its data is included in the export file. Likewise, in the right list some more scalar fields can be selected and included in the export. The difference to the scalar fields in the left list is that the scalar fields in the right list do not exist yet. They are created temporarily during the export.

The check box "export disabled vectors" includes all disabled vectors in the export as well. Please note that this check box is only available and visible for the Tectplot (`*.dat`) format!

Abaqus

The additional parameters for the Abaqus export are very specific to Abaqus. Please refer to the appropriate documentation in Abaqus.

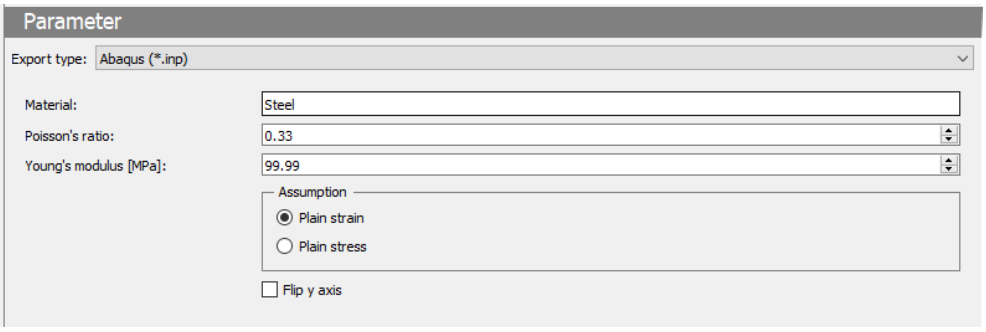


Figure 8.14: Abaqus parameters.

The "Storage" section

The export folder and export file can be defined in the "Storage" section. Select, whether data should be exported to **project** or to **external folder**. The **Preview** displays the complete storage path of data.

If **Export to project** was selected, choose a filename.

If **Export to external folder** was selected, three input lines are enabled.

Root

Figure 8.15: The "Root" folder.

The "Root" is a general root folder for the complete export. The root folder can be set/changed by clicking on the left icon. The right icon opens the folder in the windows explorer.

Folder

Figure 8.16: The export "Folder".

The export "Folder" is a sub folder of the root folder. It provides a default name that is derived from the exported set but can be defined with an arbitrary name. Please keep in mind that the usual file system restrictions of the operating system apply to the "Folder" name. The "Folder" has some sophisticated options to generate a name from the current date and time, as well as from the data of the exported set. By clicking the left button in the "Folder" line, the current date is added to the current cursor position in the edit field ("<date>"). Likewise, by clicking the second button the current time is added ("<time>"). Please note that <date> and <time> are only placeholders that are filled with the current date and time at the start of the export, not the date and time when they are added to the edit field. When the "Folder" name is edited manually, it is possible to use some placeholders derived from the attributes of the exported set. By typing "<" a list of available placeholders is displayed. They can be selected by the mouse or by scrolling with the cursor keys to the desired attribute and pressing the enter key.

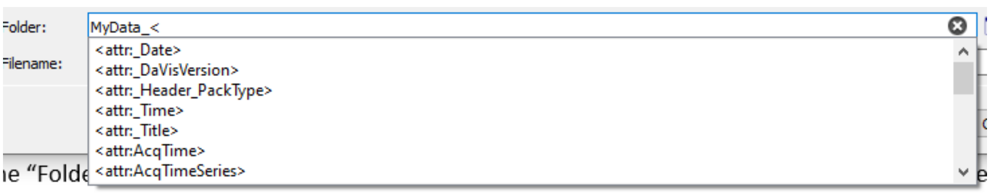


Figure 8.17: Attributes for the export "Folder".

During the actual export the placeholders are filled with the respective attributes of the currently exported set element.

Filename

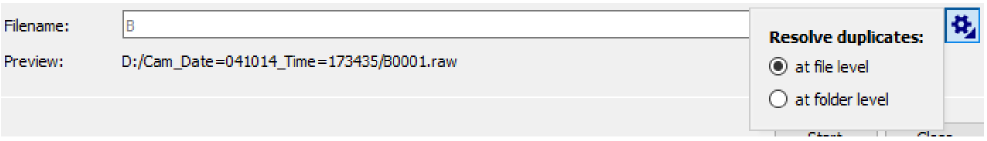


Figure 8.18: The "Filename".

The "Filename" nearly provides the same functionality as the "Folder", except for the date and time fields but the set attributes are equally available. If the mouse is hovered above the icon with the question mark, the tooltip displays the generated full qualified filename of the resulting export file. For this purpose, the data from the first set element is used.

Besides, pressing the cogwheel next to the input line allows to select, where duplicate named data should be resolved. It is possible to select between **at file level** and **at folder level**.

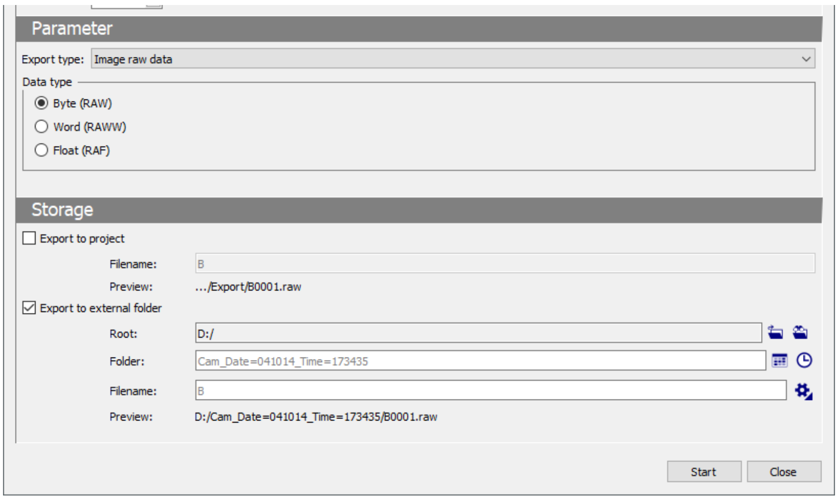


Figure 8.19: The generated Filename.

Start Button

The export is started by clicking the "Start" button. When the export is started, the "Start" button changes to a "Stop" button, so the export can be stopped at any time by the user. The progress of the export is displayed next to the "Start"/"Stop" button. The "Close" button is disabled during export.

Figure 8.20: During export.

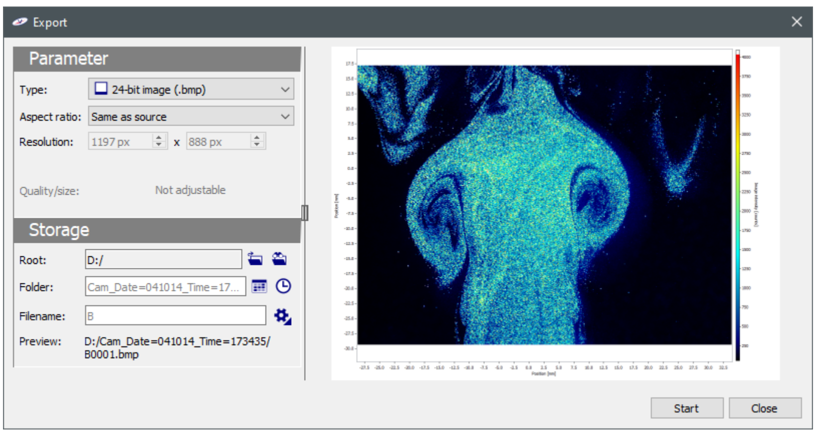
After the export is finished, either by finishing regularly, by interruption, or by being stopped due to an error, the "Stop" button is renamed in "Start" button again, the progress bar has vanished, and the "Close" button is enabled again. In case of an error an error message will appear.

8.3 Screenshot Export Dialog

8.3.1 Export Screenshot

The selected image or vector field is displayed in the preview area. Change the **type** between bitmap, jpeg and png, then select the **aspect ratio** and in case of **free** mode enter the **width** and **height** of the exported file in pixel.

Select a **root** and **folder** as storage destination, and enter a **filename** for the screenshot. Then press the **Start** button to store the file.



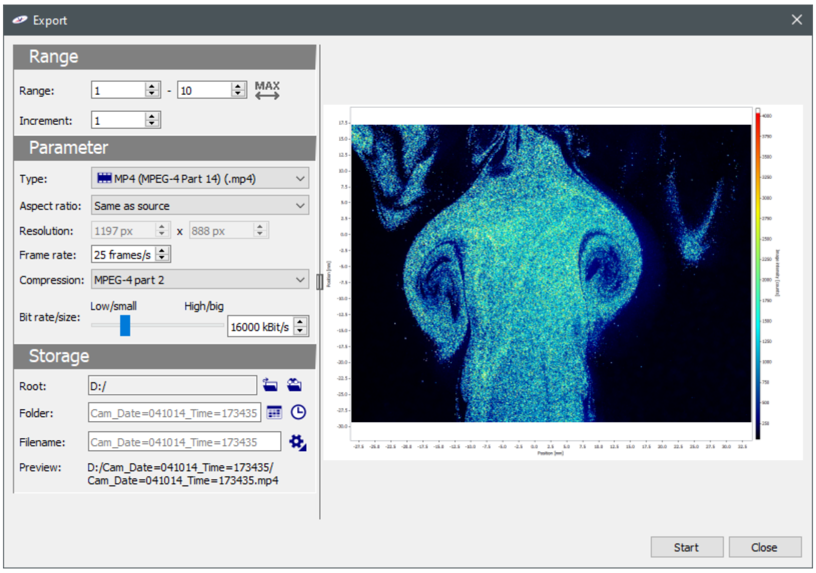
8.3.2 Export Movie

Some **Parameter** and **Storage** settings are equal to the screenshot export. For movie export, additional parameters as **Frame rate**, **Compression**, and **Bit rate/size** can be defined. The frame rate is defined by default as 25 frames per second.

The selection of images or vector fields from the source set can be changed with the **Range** parameters. By default all items are exported, means **range** is 1 to **maximum** and **increment** is 1.

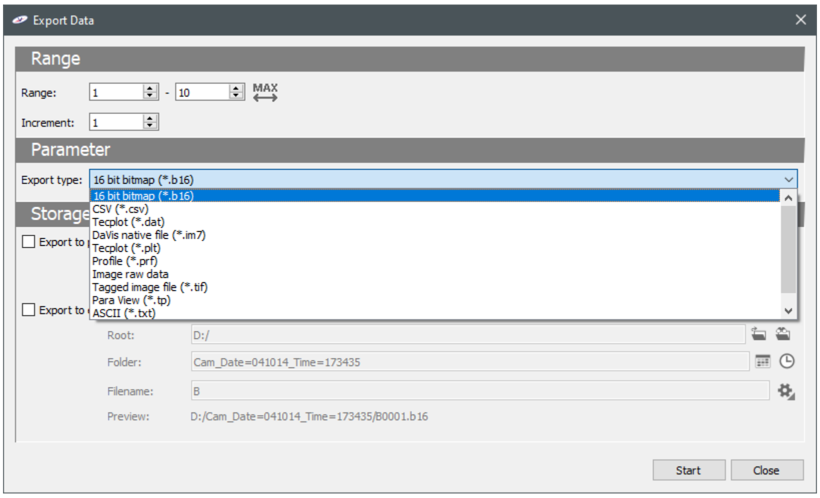
The **Root** and **Foldername** can be defined as already described on page 153. The destination **Filename** can be used as basename and automatically gets the suffix of data and/or time. The last suffix is added anyway and gives the source item index, counting from 1 to maximum (if increment is 1).

During export, the **Start** button is renamed to Stop and can be pressed to interrupt the export. Already exported items are not removed. A progress bar is displayed during export.



8.3.3 Export Data

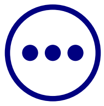
Available data types are depicted in the screenshot below. Like movie export the range and increment of source items can be changed.


Export Data
Range
Range: 1 - 10
MAX
Increment: 1
Parameter
Export type: 16 bit bitmap (*.b16)
16 bit bitmap (*.b16)
CSV (*.csv)
Tecplot (*.dat)
DaVis native file (*.im7)
Tecplot (*.plt)
Profile (*.prf)
Image raw data
Tagged image file (*.tif)
Para View (*.tp)
ASCII (*.txt)
Storage
Export to
Export to
Root: D:/
Folder: Cam_Date=041014_Time=173435
Filename: B
Preview: D:/Cam_Date=041014_Time=173435/B0001.b16
Start
Close

9 Hardware Setup

The **DaVis Hardware Setup** dialog is a graphical dialog to setup the hardware configuration used with the system. It displays the relationship between the used devices and their cabling. Hardware profiles can be created and stored in or loaded from a shelf for quick access. A **DeviceFinder** dialog is available to search for hardware components connected to the system.

The **Hardware Setup** is opened by pressing the **Hardware Setup** button on the **DaVis** project browser's toolbar on the left hand side.



Also on the project browser's toolbar the **Extras** button 1 is visible at any time. It opens a menu of which the first item opens the **DaVis - Preferences** dialog.

Further, on the project browser's toolbar the **Stop Laser** and **Stop Devices** button 2 are enabled in cases light sources are switched on or movable devices are moving. For further details on this buttons, please refer to Section 3.2.1 and Section 3.2.2, respectively.

Please note that these two mentioned buttons do not replace any emergency stop switch or other safety devices.

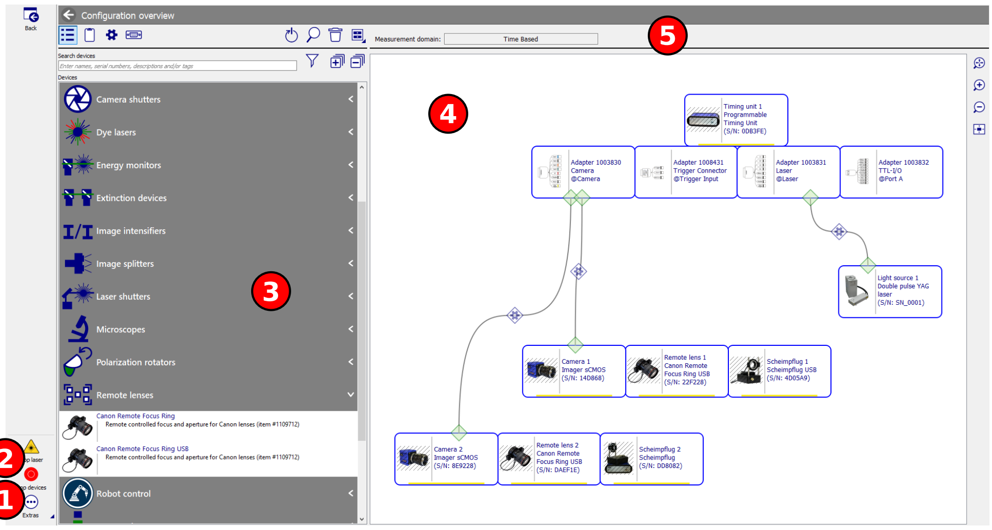


Figure 9.1: Configuration overview window

9.1 First steps to set up the hardware

The **Hardware Setup** opens with the *Configuration overview* window (cf. Fig. 9.1) containing two major parts: On the left hand side it shows by default a list of supported devices 3 . On the right side the configuration area contains a graphical view of the chosen hardware components 4 . A configuration is filled by dragging items from the device list and dropping them into the configuration area. If possible **DaVis** assists the user by preselecting that device category which is most relevant for a following step.

Starting with an empty setup the configuration area contains a hint to select a timing unit first, then add other devices and afterwards setup the device details. So first drag a *Timing Unit* and drop it into the configuration area.

Dropping a *Timing Unit* which can be used for low-speed and high-speed domains **DaVis** asks to choose a domain (cf. Fig. 9.2). This selection connects the domain specific default adapters to the *Timing Unit*. The active domain is displayed on top of the *configuration* area (cf. 5 in Fig. 9.1) and can be changed in the *Timing Unit's* device settings at any time. Depending on the selected domain the corresponding camera group is automatically expanded for the next step.

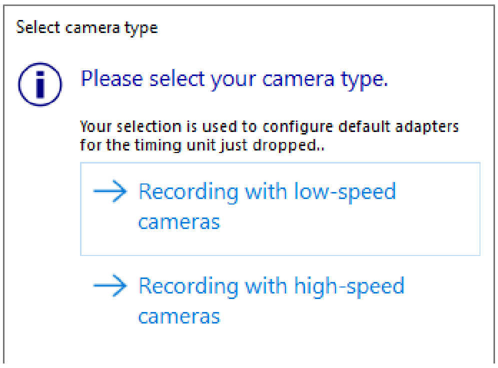


Figure 9.2: Preselecting camera speed domains

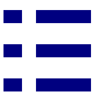
When dropping additional items **DaVis** tries to connect the appropriate cables and to automatically attach new devices to already available devices as required. The device list can be filtered to easily find the desired items.

Once all devices are configured and connected properly the hardware must be initialized and optionally can be stored to the configuration shelf. The appropriate tools are described in detail in the following sections.

9.2 Configuration overview toolbar

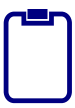
Figure 9.3: Hardware setup toolbar

Device list



Shows the device list tab on the left part of the *Configuration overview* window (default when calling the **Hardware Setup**).

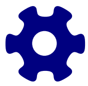
Message list



When initializing a setup, messages of different categories (information, warning, error and fatal error) are generated and are displayed in this tab. Displaying this tab is alternated with the device list. If messages are generated before the initialization the button flashes in red color to indicate that messages are available.

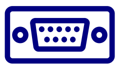
continued on next page...

Configuration details



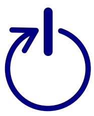
Opens the *Configuration details* view to graphically set up the cabling and the connector mapping of the setup. This button can also be found on each cable line in the graphical view and has the same function.

COM port configurator



Opens the **COM port assignment** dialog which shows all available COM ports in the system and their association to a device. It is important to configure COM port devices to allow **DaVis** to detect them correctly.

Initializing hardware setup



Initializes the hardware configuration as composed in the configuration area and applies the specified settings to the physical devices connected to the computer. In case of errors or warnings, the message tab is shown automatically. (Is called automatically when leaving this dialog, messages are shown in a separate dialog if needed.)

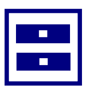
Searching devices

Opens the **DeviceFinder** dialog to scan the system for connected known hardware. Selecting product families to search shortens the scan time. After scanning the detected devices are displayed in the device list. The list can be filtered to only display detected devices (see **device filter** below).

Clear configuration

Clears the entire setup by removing all items from the current configuration in the configuration area.

Configuration shelf



Opens the hardware profile shelf to store or reload hardware profiles by user defined names. The stored profiles can also be recalled (if activated) at startup of **DaVis** (cf. Sec. 3.3.1 *Startup behavior*).

9.3 Device list area

The device list area contains groups of devices available for the use with **DaVis**. A toolbar on top helps managing the list. Having found the right device item it can be added to the configuration by dragging and dropping it in the configuration area on the right hand side.

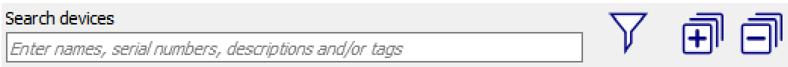
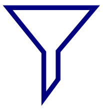


Figure 9.4: Device list toolbar

Search devices

Typing terms (such as "PIV", "camera") into the search text box filters the device list and only those items are shown which include one of the search terms in their name, description text or in their device tags.

Device filter



To restrict the device list to only those components found by the **DeviceFinder** dialog, press the **Filter** button next to the search bar.

Expand all

All available categories are expanded at once.

Collaps all

All available categories are collapsed at once.

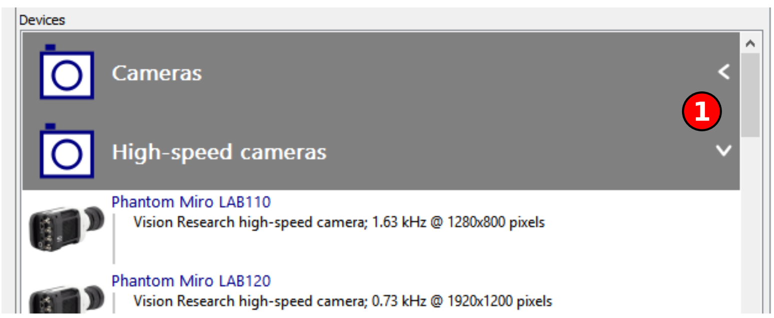


Figure 9.5: List of grouped devices

For a better overview of the device list the devices are grouped by their categories (cf. Fig. 9.5). The different groups can be collapsed and expanded by an arrow 1 in their caption. Depending on the users action **DaVis** assists the user by opening that device category which is most relevant and closing categories which are irrelevant for an outstanding configuration step.

9.4 Message list area

The message list area shows the messages generated when initializing a new hardware configuration. The messages are divided in the different types information, warning error and fatal error. Displaying the message list and device list can be alternated in the *Configuration overview's* toolbar.

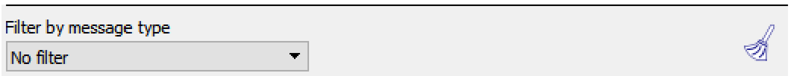


Figure 9.6: Message list toolbar

Filter by message type

The messages type to be shown can be selected in this combobox.

Clear messages

To clear the generated messages in the list, press the **Clear** button.

9.5 Configuration area

The configuration area is a graphical interface to configure the hardware ensemble to be controlled by **DaVis**. A device can be added to the configuration by dragging it from the device list on the left and dropping it into the configuration area on the right hand side. To add further components, drag them as well and drop them close to the icon of the device they shall be connected to.

There are two types of possible connections of devices in the configuration area:

1. The line configuration which represents the trigger connections between the *Timing Unit* and the various devices.
2. The physically interconnection of devices represented by docking the two device icons in the configuration area.

When drag-and-dropping a device, its trigger lines will be automatically connected to the matching connectors of the *Timing Unit* (as far as they are available). Devices which can be docked (like a remote lens control or Scheimpflug adapter) are automatically docked to the next matching partner (like a camera). To see which components are matching, all incompatible ones are faded. Manual docking is achieved by dropping the device on its partner.

Note: It is important to drop the device ON the device it shall be docked to. It is not sufficient to align them next to each other.

9.5.1 Device icons

Once a device has been set to the configuration its icon in the graphics offers further features. When hovering over the icon with the mouse cursor it becomes colorized and shows additional buttons.

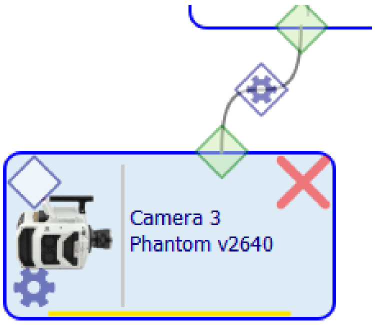


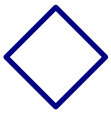
Figure 9.7: Device icon specials

Remove device from configuration

When pressing the red "X" the device and its connections are removed from the configuration.

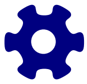
continued on next page...

Change cabling



Pressing and dragging the green rhomb allows to disconnect a present connection. Dragging the white rhomb to the *Timing Unit* induces a triggering connection to the *Timing Unit* if not already present.

Edit device settings

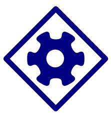


Pressing the cogwheel opens the **Edit Device Settings** dialog. Depending on the device category some type specific parameters can be changed. Descriptions of theses specific settings can be found in the respective device manuals. Also device information like type, serial number or version information are shown. This dialog is described in Sec. 9.5.2 *Device settings* in more detail.

Show device image

Doubleclicking on the image of the device enlarges the view of the picture.

Call configuration details window



With the cogwheel in rhomb button placed on the connection line the **Configuration details** window is shown. See Sec. 9.6 *Configuration details* for more information.

9.5.2 Device settings

The **Edit Device Settings** dialog shows some common device properties like name, type specification, device serial number and version information and allows to edit device type specific settings. In the **Properties** group 1 the device name can be changed as desired for easier identification of certain devices in a complex system (e.g. 'front camera', 'back camera'). This name will be shown throughout the use of **DaVis**, e.g., in the **Recording** dialog. The device type together with the serial number is used to uniquely identify the device in the system. For cameras, an additional camera number is available as known from former **DaVis** versions.

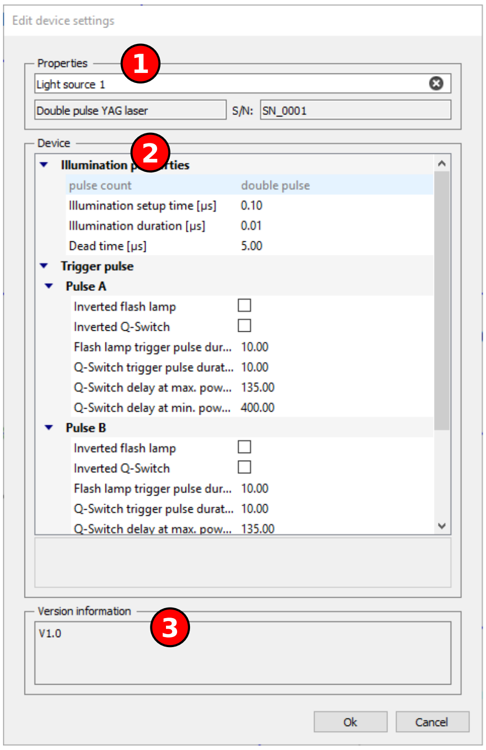


Figure 9.8: Device Settings dialog

The **Device** group 2 offers the device specific settings. All these settings are explained in detail in the respective device manual. The example in Fig. 9.8 shows the settings for a generic double pulse YAQ laser mainly concerning the timing behavior of the laser. The version information at 3 contains device specific information like driver or firmware version. Changed settings are only applied by pressing the OK button.

9.6 Configuration details

The **Configuration details** window displays the cabling of the configured devices. The view can be accessed by either clicking the settings button in the **Configuration overview** toolbar or by clicking the settings button on any cable.

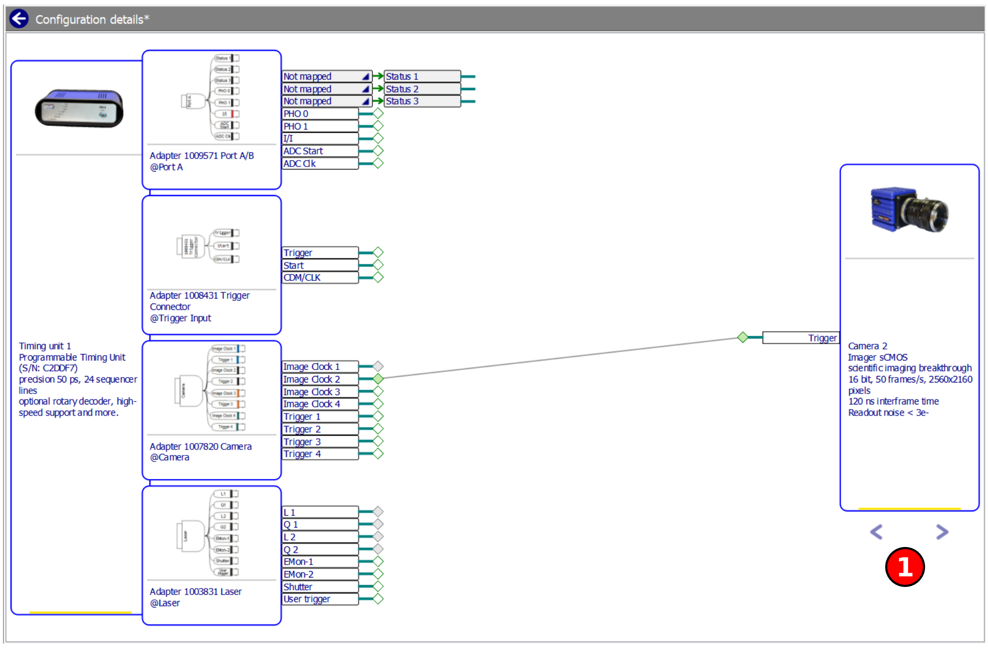


Figure 9.9: Configuration Details

On the left hand side the *Timing Unit* and all its connectors are shown while on the right side one of the added devices is visible. With the navigation buttons 1 below the device the other devices of the current configuration can be reached.

Between the *Timing Unit* and the selected device the trigger cables are shown in detail. The goal is to use this scheme as instruction to properly connect the hardware. Different cabling options can either be changed graphically by dragging and dropping to match the hardware or vice versa. In any case the graphical configuration in **DaVis** and the hardware cabling should match for a proper working system.

Figure 9.10: Impossible connection

Figure 9.11: Possible connection

When dragging a connection in the graphics **DaVis** checks and indicates if a connection is allowed or not (cf. Fig. 9.10 (impossible) or Fig. 9.11 (possible)). Ports used by not shown devices are connected to are grayed-out. To swap cables first disconnect one and leave it, connect the other then connect the first to the new port.

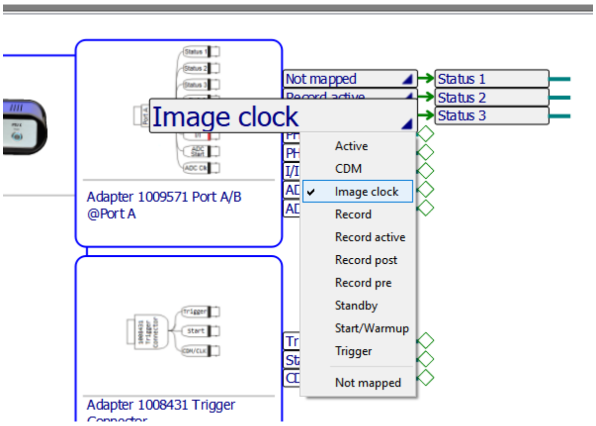


Figure 9.12: Configurable output connectors

The output signal for the status connectors of the *Timing Unit's* port adapter can be configured in the **Configuration details** window (cf. Fig. 9.12). A pulldown menu offers several different types of output signals which can be selected by a mouse click. The named signal can be detected on the port after the next hardware initialization.

10 Recording

The **Recording** dialog provides the functionality for device control and image acquisition from live view through simple recordings to complex hierarchical scans. The recorded image sets can afterwards be processed in the **Processing** dialog (see Chap. 11 *Processing*) which contains all available processing functions.

The **Recording** dialog can be accessed from the **Project Browser** pressing the **Recording** button in the **DaVis** project browser's toolbar.

10.1 Overview

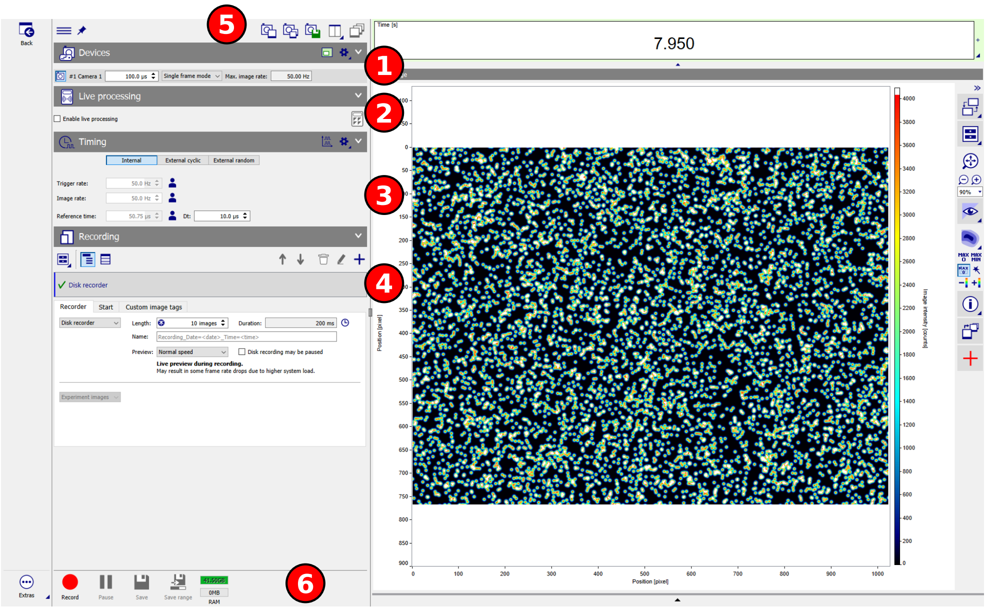
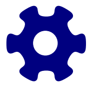


Figure 10.1: Recording dialog

As shown in Fig. 10.1, the **Recording** dialog is divided into two major parts. On the left hand side you find the **main control** section and on the right the **image display** which shows either the live image or the movie player with the recorded data set. The **main control** in a common system contains



the three control sections **Devices** 1 , **Live processing** 2 **Timing** 3 and **Recording** 4 . Each section shows the most important or most frequently used controls for direct and immediate access. **Devices** and **Timing** details are found in pop-up windows which open when clicking the corresponding settings button at the lower right corner of each section. Additional sections are available if a microscope or a robot is configured in the hardware setup. Specific projects might also add a section to the recording dialog (e.g. the live gauge in StrainMaster projects).

The **main control** has a toolbar at its top 5 and another toolbar at its bottom 6 with further controls and status indicators. The toolbars are permanently visible while the sections can be collapsed with an arrow on the right in their caption or can be scrolled up and down if not all controls are visible contemporaneously.

To enlarge the **image display** one can either move the separator between the two parts or can minimize the **main control** to a vertical buttonbar on the left with all controls accessible.

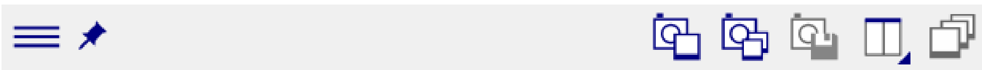


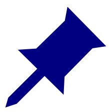
Figure 10.2: Main control toolbar

The header toolbar for the **main control** offers the following buttons:

Minimize/Maximize sections

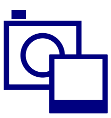
Switches the **main control** section to a buttonbar to the left side and back. So the **image display** area can fill most of the **Recording** dialog. Once minimized, the sections can still be accessed through respective buttons in the buttonbar. The header and footer buttons and controls are included in this buttonbar to stay visible.

Auto collapse



Auto collapses the sections. When opening a required section, all other sections are collapsed. However, sections can be pinned so that auto collapsing is omitted.

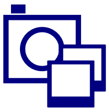
Take a snapshot



Takes a snapshot with all active cameras and other devices like AD converters into the live image display.

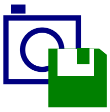
continued on next page...

Enable/Disable live view



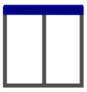
Toggles between live mode and idle. In live mode all active devices are continuously grabbing images which are shown in the **image display**. In live mode all device setting can be modified and they will be applied once the current image acquisition is completed. Usually one should not notice the latency between changing a parameter and its effect. In some cases e.g. when having some slow responding devices or a slow image rate, the latency becomes noticeable.

Save live image



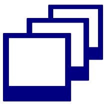
Saves the currently displayed live image as is to the 'Live images' folder of the active project. This is intended for quick backing up interesting data.

Configure view layout



A grid layout of the live image display can be defined. This is intended for adjustment purposes when multiple camera should be visible simultaneously or when different regions of one camera are of interest e.g. for adjusting the Scheimpflug angle and focus.

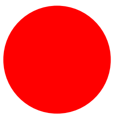
Show last recorded dataset



Switches the **image display** to the movie player which displays the last recording or scan. The button is only active if a recording has been acquired.

The footer toolbar is equipped with the following buttons and information:

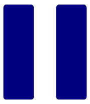
Record



Starts the recording. This can either be a quick recording of a couple of images or a complex scan. Once started the button toggles to a cancel button which allows to stop the current recording in a controlled manner. **Note that data may be lost when stopping a RAM recording.**

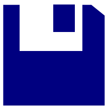
continued on next page. . .

Pause



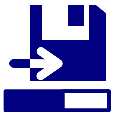
Pauses the recording. This is only available for disk recordings and has to be enabled on the **Recorder** tab of the **Recording section** (cf. Sec. 10.5 *Recording section*). Pausing a recording means actually pausing the data storage.

Save RAM recording



Saving a recording becomes enabled if an unsaved RAM recording is present. This can be useful when checking the data before starting the sometimes time consuming storage process. During a scan or if the 'Automatically store images' option is checked on the **Recorder** tab in *RAM recorder* mode (cf. Sec. 10.5 *Recording section*) this button is inactive. Disk recordings are always directly stored to disk and thus do not need to be stored manually.

Save range of a RAM recording



This is an alternative to store the full recorded data range from a RAM recording. The range can be selected in the movie player using either the small arrows above the image slider or by setting in the desired first and last image in the concerning spinboxes. The range can be decreased using an increment other than one.

The image viewer on the right contains the image itself and the dashboard which allows to display e.g. focus quality, a histogram or measured device data like ADC data or energy values from an Energy Monitor. For adjustment purposes it might be useful to activate focus quality items which are associated with different areas in the image e.g. to adjust focus and Scheimpflug angle. Details on the usage of the dashboard are found in Sec. 6.1.5 *Dashboard*.

The 'i' button in the image toolbar on the right allows to select any of the available image attributes and custom image tags to be superimposed on the **image display**. The appearance of these attributes can bet set in a dialog which is opend by right clicking on such an item. The position can be changed by dragging the item with the mouse.

Details about the **image display** and the **movie player** with all their options are described in Chap. 6 *Data Views*. Basically the appearance of the **image display** can be controlled through the toolbar right beside the image. For multi frame images, which are available when using multiple cameras or having cameras with double exposure mode, a slider becomes visible below the image to scroll through the frames.

The **movie player** has its controls below the image to allow simple playback, reverse playback, different speeds and single stepping. After a scan the **movie player** has two control bars to navigate through the recorded **Multi Set** (cf. Sec. 4.9.3 *Multi SET*), the upper one moves (horizontally) through the current scan point while the lower one moves (vertically) through all scan points at the same image number (cf. Fig. 4.21).

10.2 Devices section

The **Devices** section contains all device controls. This section is composed dynamically depending on the devices in the current hardware profile so it might look slightly different than in the screen shot shown in Fig. 10.3.

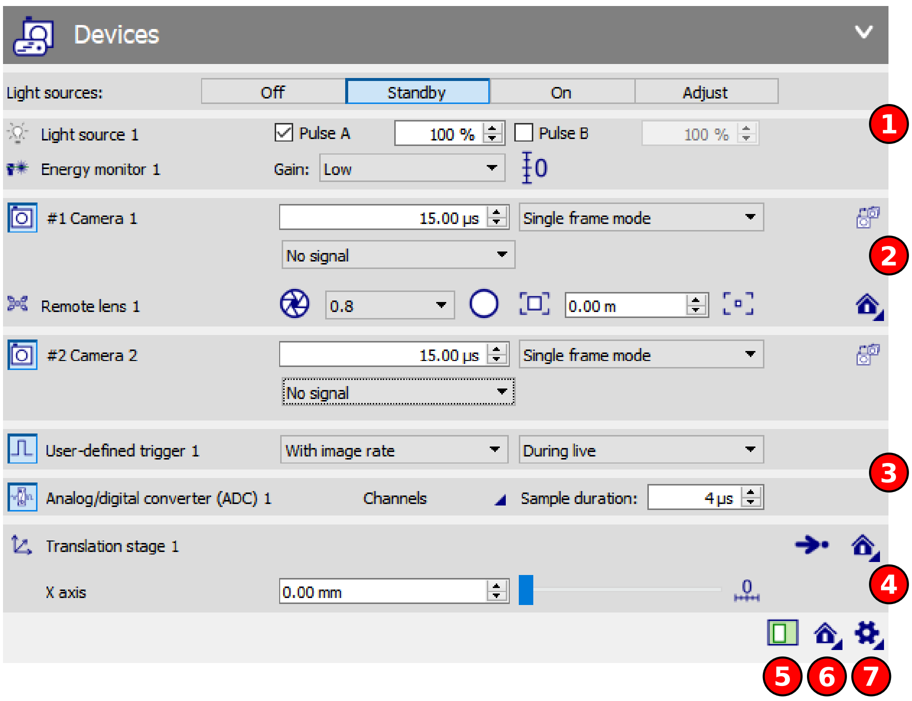


Figure 10.3: Devices section

The top area belongs to the light sources 1 followed by cameras 2 with their optionally attached devices like remote lens, Scheimpflug adapter or

Image Intensifier. Other devices like AD converters, user triggers 3 or translation stages 4 are positioned below the camera area. Devices that are somehow connected to each other are grouped together in a shaded area.

For each device the most important and most often used controls are available directly in this section. Settings which are more rarely used are available in the so called **device details**, a pop-up window (see Fig. 10.5, Fig. 10.6 and Fig. 10.8) which opens when clicking the settings (cogwheel) button 7 at the right lower corner of the section. To leave this window click somewhere on the underlying main window.

The **device details** window is organized into horizontal and vertical tabs. The horizontal tabs represent the root devices, e.g, those which create a group in the **devices** section like a camera or a light source. The vertical tabs contain general and specific settings for the root device and usually one additional tab per attached device (like remote lens control or Image Intensifier).

The camera AOI (area of interest) can be configured in a separate view which opens by clicking the green rectangle 5 . The AOI can either be changed interactively or by typing coordinates to the corresponding controls. If any motored devices are included in the hardware profile, the home button 6 appears to reference of all respective devices simultaneously.

10.2.1 Light sources

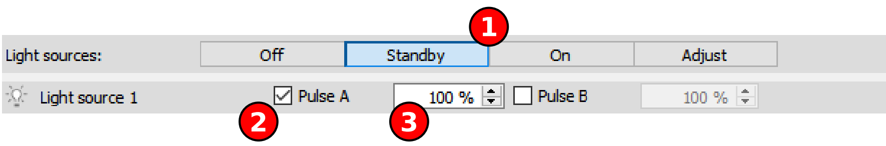


Figure 10.4: Light source controls

The upper row of buttons 1 define the system light state. This state controls all activated light pulses with a single click.

Off: Light sources are all off. No light is emitted in any case.

Standby: Light sources are all set to standby. No light is emitted. Usually required for lasers to keep the pumping circuits running (e.g. flash lamps of a Nd:YAG laser). Some high power lasers which require Q-switch always operation would also fire the Q-switch in this state but they are most likely equipped with a mechanical shutter to avoid light emission during standby.

On: Light sources are all set to on. Light is emitted whenever the camera system, more precisely the PTU is in record state, meaning that the cameras are active and triggered. This is the case during snapshot acquisition, active live view or during recording.

Adjust: Light sources are all set to adjust. **Light is always emitted.** This mode is intended for adjustment purposes.

Each light source has its own row with controls to enable the individual pulses and to set the pulse power if the light source provides power variation. For Nd:YAG lasers this is typically realized by detuning the delay between pumping lamp and Q-switch triggers. An LED's power is for instance controlled though its pulse width. For quick access only the major power controls are available.

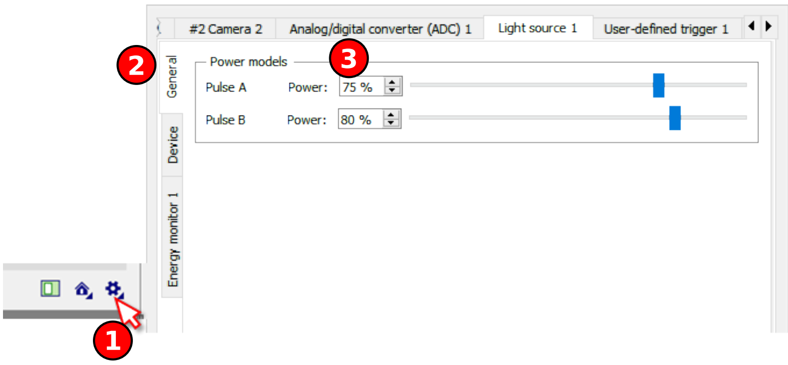


Figure 10.5: Light Source Details - General tab

If a light source has additional options to set the pulse power, these power settings are found in the **device's details** window.

The **device details** window is opened by the settings button 1 of the devices section (cf. Fig. 10.5). Each light source has its own settings tab on the horizontal tab bar. On the vertical tab bar it has a *General* 2 settings tab. The *General* tab in shown example includes the same power controls 3 as in the **devices** section but with additional sliders. Other light source types can contain additional controls here. Please refer to the specific light source manual for further explanation.

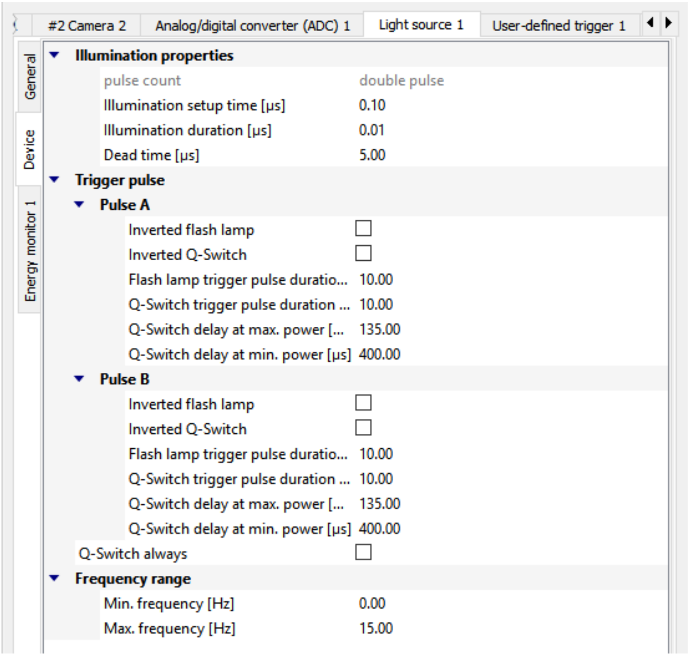


Figure 10.6: Light Source Details - Device tab

The second horizontal tab contains the light source type specific settings. These are most likely the same settings as in the hardware setup's **Edit device settings** dialog (cf. Fig. 9.8). For details about the listed settings please refer to the corresponding light source manual.

Usually one specifies timing properties for the light sources to allow Davis to place the light at the right point in time with respect to the camera exposure. Since many in **DaVis** available light sources are generic devices, they need to be parameterized to match the behavior (trigger delays, light emission or light pulse width) of a connected real hardware.

All other tabs on the vertical bar are optional and depend on the devices docked to the light source (like Pulse Energy Monitors or Shutters).

10.2.2 Cameras

Figure 10.7: Camera controls

The camera controls in the **devices** section (cf. Fig. 10.7) allow to activate/deactivate 1 the camera, set the exposure time 2 and set the exposure mode 3 . For multi camera configurations a ’copy to all’ button 4 is available to copy the settings for exposure time and mode to all other cameras as well. This usually works for cameras of same type. If the types differ the settings might not be exactly equal due different limitations of the specific models.

The acquired frames of all activated cameras are added to the image. The arrangement of camera frames in an image from left to right on the frame slider equals the order of the cameras in the **devices** section from top to bottom. The order of cameras themselves is specified by the order of adding them to the hardware configuration.

The exposure time can be edited in single and in double frame mode for most models. In double frame the exposure time only applies to the first frame since the second frame is exposed as long as the readout of the first frame takes. For double frame measurements with very short PIV dts (in the order of the cameras inter frame time, usually below 50 μs) it is recommended to set the exposure time to its minimum value. Doing so ensures that **DaVis** can accurately align the laser shots to the camera exposures.

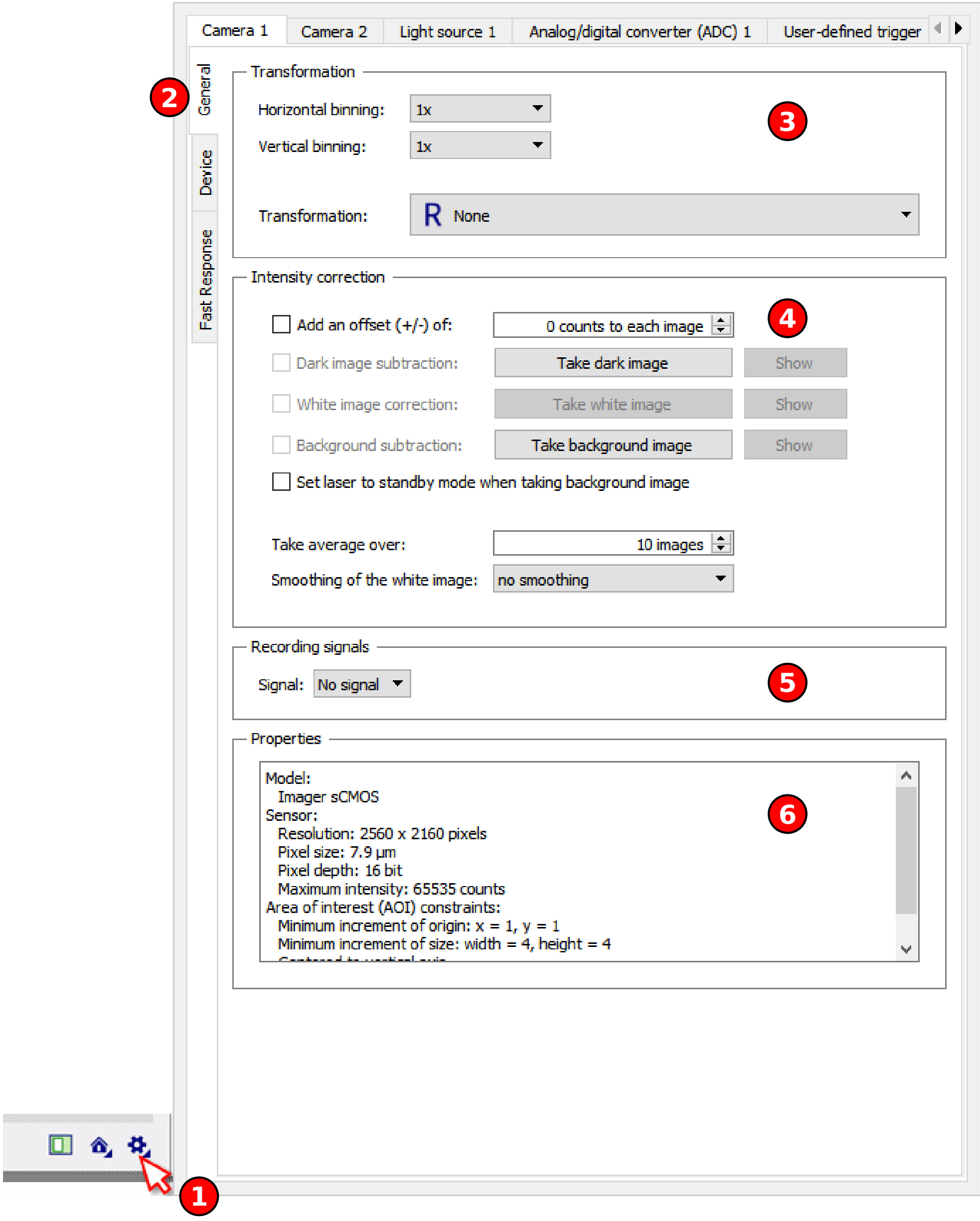


Figure 10.8: Camera Details - Device tab

The camera details are opened by the settings button in the **devices** section 1 in Fig. 10.8. The desired camera can be selected on the horizontal tab bar. The camera tab itself always has a *General* and a *Device* tab on the vertical tab bar. Optionally there are more vertical tabs depending on the specific camera model and additional devices attached to the camera (like remote focus lenses, shutter or Scheimpflug angles).

The *General* tab 2 contains image transformation controls 3 , intensity correction controls 4 and signal options 5 (see Fig. 10.8). In addition several camera properties are shown 6 . All image processing activated here is performed live when streaming the data from the camera to the PC for snapshot, live and disk recording. For RAM recordings the processing is applied prior to displaying or storing the images.

Note: If maximum performance for disk recording has a high priority it is recommended to not use any of the general camera options which are live processed by the software and prefer to execute them off line later on using the **DaVis** processing engine.

General tab - Binning

The first option in the *Transformation* group 3 in Fig. 10.8 is the camera binning in X and Y direction. Binning means to sum up the intensities of multiple pixels as one enlarged pixel. Binning can be performed in hardware and software. Hardware binning on CCD sensors sums up the charges analogue on the sensor before digitizing the image and benefits higher sensitivity and noise reductions. Software binning sums up the intensities after digitalizing and creates a higher dynamic range. For CMOS sensors some techniques exist to do a kind of a hardware binning before digitization but some just do software binning in the camera processor. Both types of hardware binning usually allow higher image rates be because the transfered data size is reduced with binning.

Depending on the camera type the available hardware binning settings are limited. **DaVis** always uses maximum hardware binning first and then calculates the remaining binning afterwards. Once a binning > 1 is selected, the used hardware and software fractions are printed behind the controls in 1 .

Note: When binning is used, the spatial resolution decreases. Accurate PIV measurement requires the highest possible resolution and so binning should not be used for PIV!

General tab - Image transformation

The second option in the *Transformation* group 3 in Fig. 10.8 is the frame transformation (like mirroring or rotating) which can be selected from a combobox.

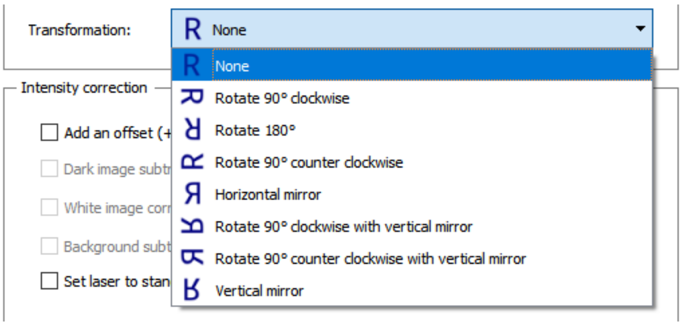


Figure 10.9: Camera Details - Transformation options

The ’R’ symbol shown in the listed items (Fig. 10.9)clarifies how the image is transformed.

General tab - Intensity correction

In the *Intensity correction* group 4 in Fig. 10.8 the use of live dark images, white images and background subtraction can be configured. Some camera models can provide slightly different options if they use a build in dark and white image correction (e.g. the Imager sCMOS).

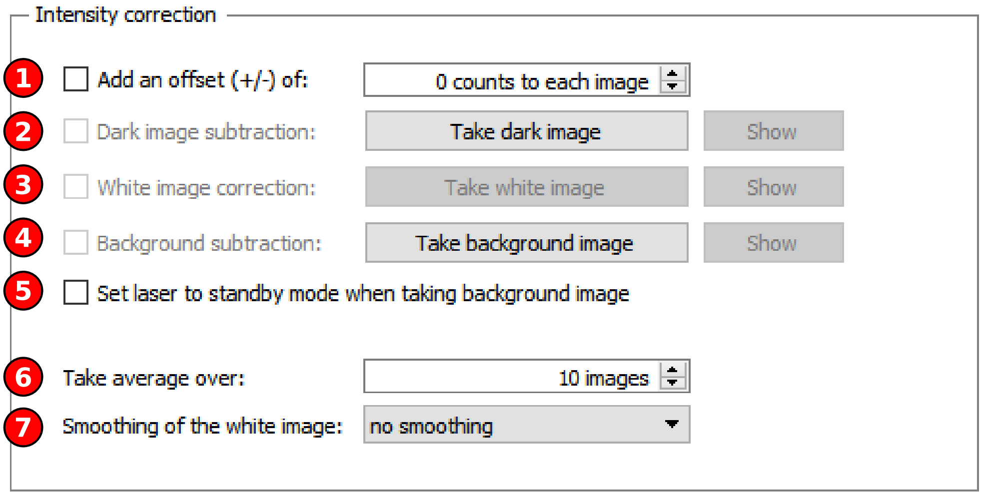


Figure 10.10: Camera Details - Intensity Correction

The control’s tasks of the intensity correction group (cf. Fig. 10.10):

1 **Add an offset:** Simply adds a constant intensity value to each pixel intensity in all images.

2 **Dark image subtraction:** CCD and CMOS cameras have a dark current in the scale of a few electrons which causes a dark level intensity. This offset is included in all images. The dark current increases with the sensor's temperature and is usually decreased by cooling the chip. These changes can be in the range of +/- 10 counts.

If the camera lens is capped so that no light reaches the detector, the so called dark image can be acquired. It is recommended to acquire a new dark image first when starting a measurement series if the intensity itself is responsible for the measured quantity (like in LIF experiments).

3 **White image correction:** The image of a homogeneously white plane shows the spatial intensity distribution of the detector's efficiency. If needed it can be corrected by acquiring an image of a homogeneously illuminated object (e.g. defocused white paper or milky glass plate), normalize this white image and divide later acquired images by this normalized white image.

For the white image correction the dark image has to be subtracted first.

4 **Background subtraction:** A background image considers a background of the recorded scene. Do not confuse this with the dark image 2 . A background could simply be some shiny object behind the scene which superposes the desired signal or it is caused by a scattering e.g. from molecules (e.g. a Laser Induced Fluorescence (LIF) experiment).

Apart from the desired fluorescence signal, e.g. a Rayleigh dispersion induced signal might give an undesired background (also depending on the used filters). In this case a Rayleigh correction image (LIF-experiment conditions without fluorescence molecules) can be acquired and subtracted from recorded images.

5 **Set laser to standby mode:** For background image acquisition **DaVis** can be advised to automatically set the laser temporarily to standby to avoid laser scattering in the reference image.

6 **Take average over:** To acquire any of the correction images a number of averaged images can be set to reduce noise.

7 **Smoothing of the white image:** Set a smoothing filter and select the size of then convolution kernel.

DaVis builds up a database of reference images in the

<DaVisFolder>/ProgramData/IntensityCorrection

folder. It stores images for different AOI, Binning and exposure mode settings for each camera identified by type and serial number. This avoids acquiring a new set of reference images every time these camera parameters are varied.

Note: Acquiring reference images switches off any active image transformation to be able to use the same reference image for any transformation.

The sequence of processing is as follows (all steps are optional):

1. apply software binning
2. subtract constant offset
3. subtract dark image
4. apply white image correction
5. subtract background image
6. apply image transformation

General tab - Recording signals

The only control in the *Recording signals* group 5 in Fig. 10.8 is a combobox to select the type of signal. These signals can be used to tag the images for the following processing. The available signals are specific for each project and so described in the corresponding project manuals.

General tab - Properties

In the *Properties* group 6 in Fig. 10.8 some read only camera properties are listed like camera type, sensor size, pixel sizes, bit depth and resulting max. number of counts, ADC resolution with its resulting maximum intensity and the conditions for setting the area of interest (AOI).

Device tab

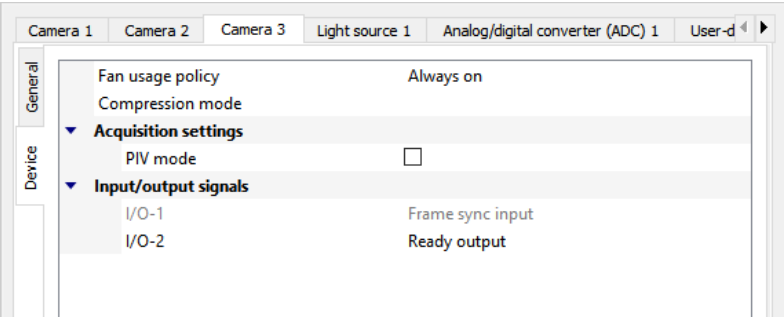


Figure 10.11: Camera Details - Device tab

All camera tabs contain a vertical *Device* tab which contains camera type specific settings. As example Fig. 10.11 shows the settings of a Phantom high-speed camera arranged as a tree view. Here it is possible to (de)activate the PIV mode of the third camera listed in **devices**. All camera type specific settings are explained in the respective camera manual.

Area of Interest

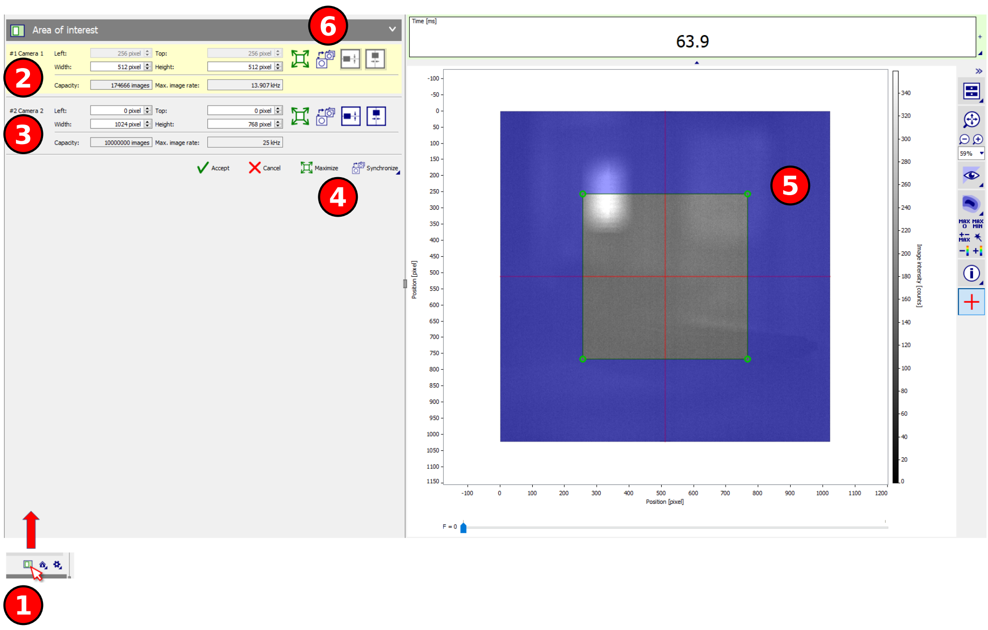
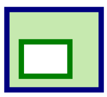


Figure 10.12: Area of Interest Editor



The camera AOI can be edited very comfortably in the interactive AOI editor which is opened when clicking the AOI button 1 in Fig. 10.12 at the bottom of the **devices** section.

Note: To edit the camera AOI it is recommended to either run the camera in live mode or to have acquired a snapshot at the full AOI before opening the AOI editor.

The left part shows a group of controls for each camera 2 and 3 as well as some common controls 4 . On the right the current live image is displayed as full camera AOI whereas the set AOI is superimposed as a green rectangle. The AOI can either be interactively changed in the image area 5 or by entering the upper left coordinates and the dimensions in the corresponding controls. Also the AOI can be reset to its maximum for each camera or all together 6 .

The camera is changed by moving the frame slider or activating any of the controls associated with the desired camera (e.g. clicking into the 'Width' spinbox for Camera 1). The active camera is highlighted.

The AOI selected for one camera can be synchronized to all others using the **copy to all** button 6 . The two buttons right to the **copy to all** button indicate the symmetry constraints. If a camera has no symmetry requirement the button can be checked to enforce a symmetry.

In the **Synchronize** menu at 4 all auto synchronizations can be specified (e.g. to let **DaVis** automatically copy the AOI coordinates to all other cameras when editing one of the cameras).

Once all cameras have been set up the editor is closed by the **Accept** button. To abort the editing and to revert the AOI settings back to their previous coordinates the editor can be closed via **Cancel**.

10.3 Live processing section

Live processing allows to monitor the usability of desired processing operations that will be applied to the current **Live View**. Via the **Configure view layout** button processing results can be displayed besides the raw camera images. This view is depicted in Fig. 10.13.

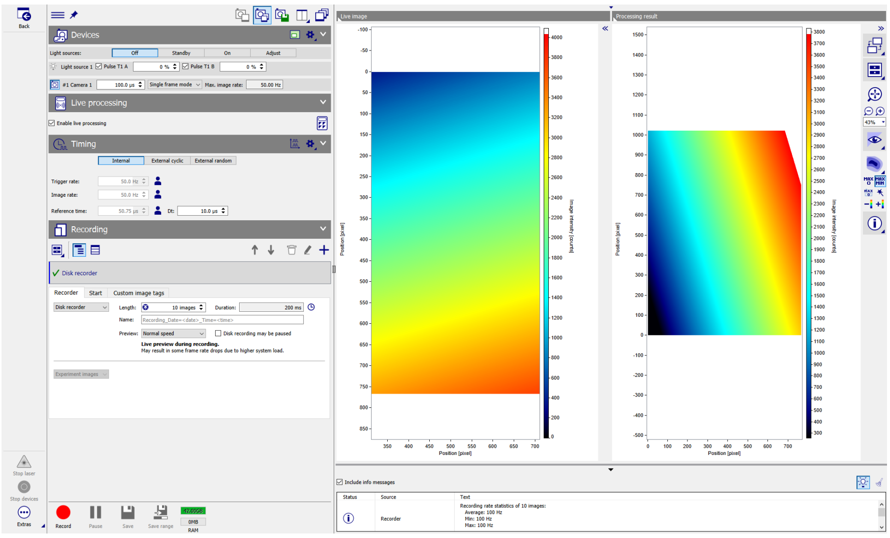


Figure 10.13: Live processing. The left image contains the live image. The right image presents the processing result. In this case, the operation **Rotate** was applied to the data.

Live processing can be enabled by checking the checkbox **Enable live processing**. Clicking the **Processing** button, opens the **Processing** dialog. For this step, it is necessary to have a previously grabbed image or a running live grabber, otherwise the processing dialog cannot be entered. Here, various operations can be applied. Live processing is only executed during grabbing, i.e. results are not stored. During recording, the live processing is not executed. In case the operation list is invalid or does not contain any processing operations,live processing cannot be performed. Other errors that occur are displayed in info messages. The live frame rate depends on how many operations are utilized. The more operations are applied, the lower the live frame rate. To switch views between **Live view** and **Processing result** click on the gray bar as depicted in Fig. 10.14

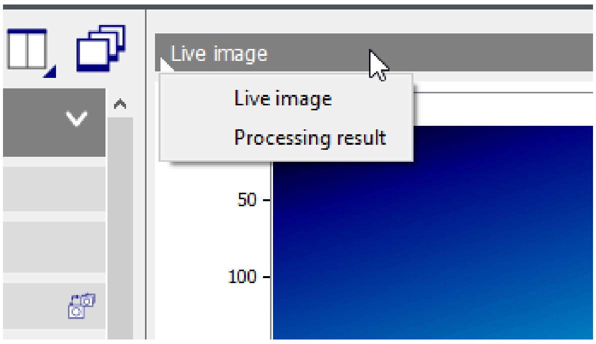


Figure 10.14: Live processing. Switch between views.

10.4 Timing section

The timing section bundles all settings related to triggering, image rates and reference times. This section provides a brief description for the timing settings. A detailed description can be found in the PTU-X manual.

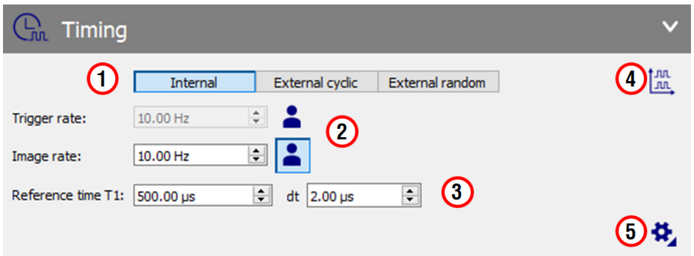


Figure 10.15: Timing section - internal

1 **Trigger source:** The trigger source setting determines the origin of the PTU trigger and its general behavior.

- Internal: The PTU internal clock generates the trigger signal. This trigger signal has a precise constant frequency. This is the default trigger source for free running of the system.
- External cyclic: A TTL trigger signal needs to be sent to the PTU. It is assumed, that this signal is periodic and has a frequency, which stays constant during a measurement.
- External random: A TTL trigger signal needs to be sent to the PTU. The trigger has no frequency and could be even a single trigger.

2 **Trigger rate settings:** The settings for the trigger rate depend on the trigger source settings above. Details are described in the PTU-X manual.

3 **Reference time:** The delay of the reference time is set with respect to the trigger. In case a reference time has double events (like for PIV), the time between these events, the "dt", can be set here. The number of reference times and if they are double events is set in the details of the PTU hardware settings. Depending on the measurement domain of the PTU, the Reference time can also be set in phase angle degrees instead of time.

4 **Timing diagram:** Clicking on this button opens the timing diagram. The timing diagram shows the electric TTL signals and the active times of the devices. Note: The timing diagram reflects the actual state of all recording settings. E.g.: Laser pulses will only be shown, if the laser is activated ('On' mode).

5 **Device Offset:** Clicking on the **Details** button allows to fine tune the trigger timing of each device (the time offset to add to the automatically calculated trigger time).

10.5 Recording section

In the **Recording** section (see Fig. 10.16) all settings regarding the recording of image sequences are specified. A simple recording is performed by just typing in the number of images to be recorded and click the **Record** button. Alternatively a very complex automated scan of e.g. a reference time combined with an X and Y axis could be parameterized all within this section.

A scan could either be configured as a hierarchical or as a table scan. The hierarchical scan is composed of different scan values and loops nested on different levels usually used to scan through continuous parameter fields. A table scan consists of a list of scan points where each column describes a specific scan value. This type of scan is intended for large sparse parameter fields.

A simple recording produces a single recording set while a scan produces a **Multi Set** (cf. Sec. 4.9.3 *Multi SET*) with the images in 'horizontal' dimension and the scan points in 'vertical' dimension. These **Multi Sets** can easily be viewed with the movie player which has movie controls for both dimensions. Also processing these **Multi Sets** can easily be done in either direction.

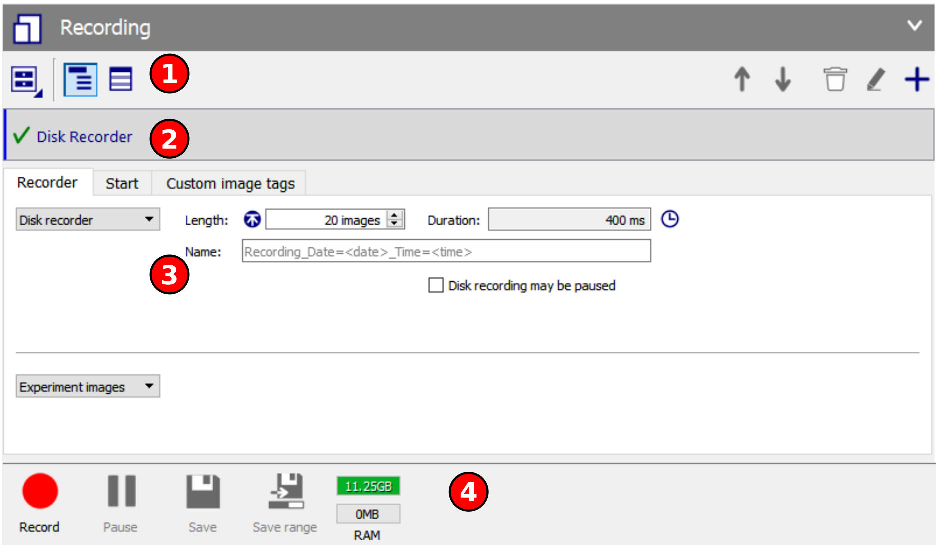


Figure 10.16: Recording section

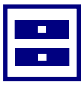
With low-speed cameras and thus in low-speed domains (cf. Sec. 9.1 *First steps to set up the hardware*) recordings can either executed with a disk recorder or a RAM recorder. A disk recorder is very flexible and the more convenient option while the RAM recorder is intended for highest performance with respect to frame rates. For high-speed domains a RAM recorder is available only because high-speed cameras always record to their built-in memory (cf. Sec. 10.5.1 *Recorder type*).

The left hand side of the toolbar at the top 1 of the **Recording** section in Fig. 10.16 contains tools to control the scan sequence or scan table. (The tool buttons on the right are described in Sec. 10.5.4 *Scanning modes*.) A composed scan sequence or scan table is displayed in 2 (cf. Fig. 10.25). Therefore this row expands vertically. The central area 3 contains the tabs *Recorder*, *Start* and *Custom image tags*. The controls in the footer toolbar 5 are used to execute the current recording (cf. Sec. 10.1 *Overview*).

The *Recorder* tab contains controls to set the essential recording parameters such as recorder type (if available), recording length and recording name. These controls are described in the following sections in more detail.

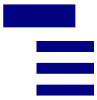
In the **Recording** section's toolbar 1 the following items are included:

Scan Sequence Shelf



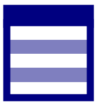
A shelf for the scan sequence. Allows to quickly store and load scan sequences using the shelf mechanism.

Hierarchical Scan Sequence



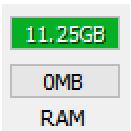
Switches the recording to a hierarchical scan mode. For details on the scanning modes, see Sec. 10.5.4 *Scanning modes*.

Table Scan Sequence



Switches the recording to a table scan mode. For details on the scanning modes, see Sec. 10.5.4 *Scanning modes*.

RAM status display



Displays the amount of available RAM (green) and RAM used by the recorder.

10.5.1 Recorder

In the following sections the controls of the **Recorder** tab are described in detail.

Recording name

The recording name defines either the name of a simple recording or the name of the multi set containing an entire scan. The name is automatically generated by **DaVis** using wildcards for the current date and time.

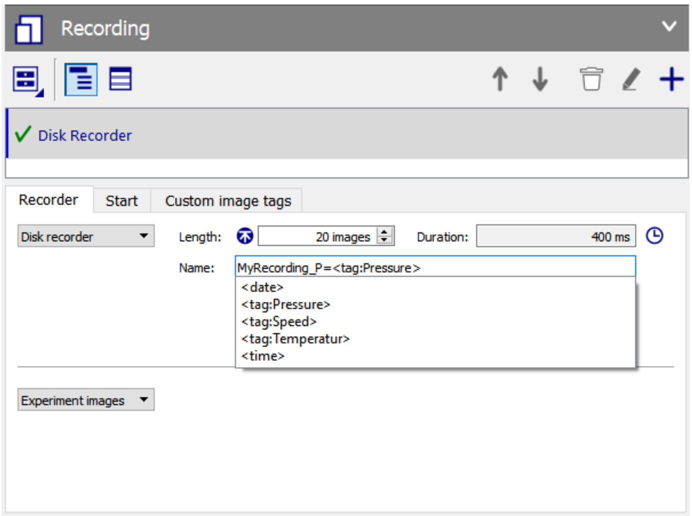


Figure 10.17: Recording name - wildcards

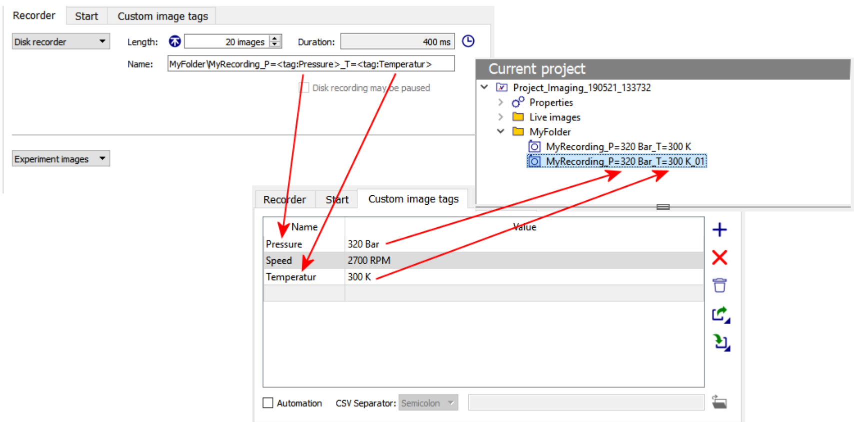


Figure 10.18: Recording name - wildcard substitution

To record to a sub folder in the current project a path can be entered like *MyFolder/MyRecording* or *MyFolder\MyRecording*. The folder does not need to exist. It will be created during recording if necessary.

For dynamic name generation, wildcards may be used. A wildcard is given as a name part enclosed into <>, e.g. <date> or <time>. Other wildcards are available by **custom image tags** (cf. Sec. 10.5.3 *Custom image tags*). The **custom image tag** related wildcards are prefixed with 'tag:' like e.g. *<tag:Pressure>*, where 'Pressure' has to be a valid **custom image tag** name. If all wildcards are identical multiple recording names are counted up.

Note: When editing the recording name a list of available wildcards pops up each time a < is typed.

Example: *MyRecording_P=<tag:Pressure>_T=<tag:Temperature>* creates a recording name with *<tag:Pressure>* and *<tag:Temperature>* being replaced by the values of the tag 'Pressure' and 'Temperature' to *MyRecording_P=320 Bar_T=300* (cf. Sec. 10.5.1 *Recording name*).

Recorder type

As briefly introduced above, there are two recorder types available: disk recorder and RAM recorder.

The **disk recorder** only works with streaming cameras (which are almost all low-speed cameras). It streams the data from all devices into the computer, composes it to measurement point (a multi frame image), sends it through an optional processing stage and pushes it into a queue which is drained by a high performance disk writer engine using the **DaVis** stream format where basically the data of each device are written to a separate file. The queue uses the dedicated recording memory as defined in the preferences (cf. Sec. 3.3.9 *Recording*). If the hard disk's write speed is slower than the camera system's data rate, the queue will continuously fill up which is visualized in the lower RAM fill level indicator (see 4 in Fig. 10.16). Once the RAM is filled, the triggering of the devices is paused (the timing unit enters the record idle state, which e.g. only fires the laser flash lamps but not Q-switch and camera) until a slot in the queue becomes available. This decreases the frame rate during recording. The frame rate statistics are stored in the comments of each recording (displayed in the project browser's properties windows). The images are always displayed during recording similar to live mode since the data is streamed directly to the computer.

The **RAM recorder** works either with streaming cameras or RAM cameras (usually high-speed cameras). It acquires the data either to the camera RAM or to the computer RAM. Once the recording is finished the data are written to disk (either automatically after recording or by manually saving or save range, respective option is available in the recording settings). In case of streaming cameras this mode potentially allows higher continuous frame rates since the computer has to handle less data traffic as it just needs to transfer the data from the frame grabbers (or GigE or USB cards) to the memory whereas in case of the disk recorder it has to move the data from memory to disk at the same time. A live preview of image data is possible but optional. The preview will pick the most recent image and display it. Thereby the display refresh rate can be chosen. For maximum performance, the image display should be turned off (checkbox is available in the **Recording** tab if RAM recorder is chosen).

10.5.2 Start

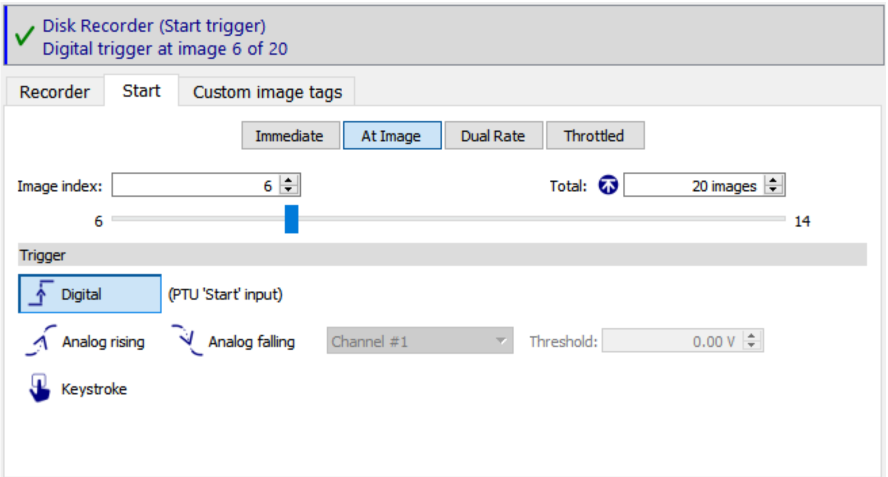


Figure 10.19: Start tab - At Image

The **Start** tab contains controls to set the start conditions of a recording in four different ways:

Immediate: Recording starts immediately. No further settings are needed.

At Image: Defines the triggered image in a post triggered recording. Several options for the trigger source can be chosen with this recording mode (cf. Fig. 10.19 and corresponding description).

Dual Rate: A method to change the recording frequency during a recording between full and throttled rate (cf. Fig. 10.20 and corresponding description). Different sources for rate selection can be set as well.

Throttled: A constant time delay between storing images can be set. This allows recordings of e.g. very long time periods.

With the **At Image** mode a triggered image number can be specified to define the number of image before and after the trigger. The images are acquired into a ring buffer while the expected trigger signal is used as a post trigger: From the image acquired at the time of the trigger signal, the image index is counted back in the buffer and these past images are included in the recording. Then the recording is filled to its configured size with images acquired after the trigger signal.

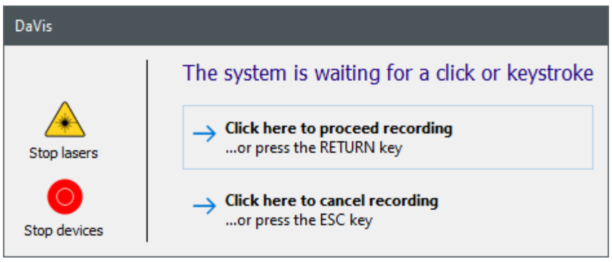
The number of trigger position within the image sequence can either be set in a spin box or by a slider. On the left side near the slider the number of images which are acquired before the trigger signal is shown. The remaining recorded numbers of images is shown on the right.

To define which method will start the image acquisition there are three different trigger options:

Digital: A digital trigger signal which is sent to the 'Start' input of the PTU.

Analog: If an ADC is used a threshold for an analog signal can be set to generate the start trigger.

Keystroke: The trigger can be sent by the user through a dialog which pops up during the recording (once all pre-trigger images have been acquired). Further, all light sources or all devices can be stopped by pressing the buttons **Stop lasers** or **Stop devices**, respectively. More detailed information regarding these buttons can be obtained from page 49.



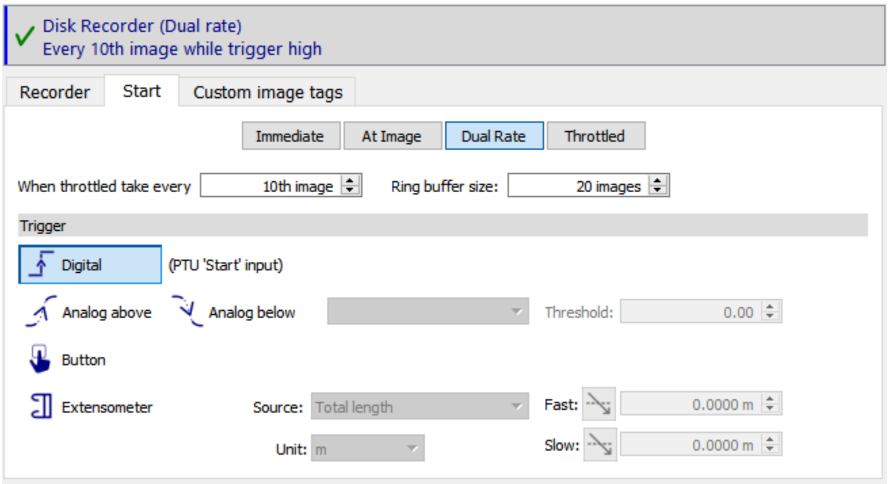


Figure 10.20: Start tab -Dual Rate

The **Dual Rate** mode allows to switch between a high and a low recording rate during one and the same recording. The change between the recording rates is initiated either by an external TTL signal, an ADC signal, a button click or by reaching a user-specified condition for the live extensometer (in **StrainMaster** mode only).

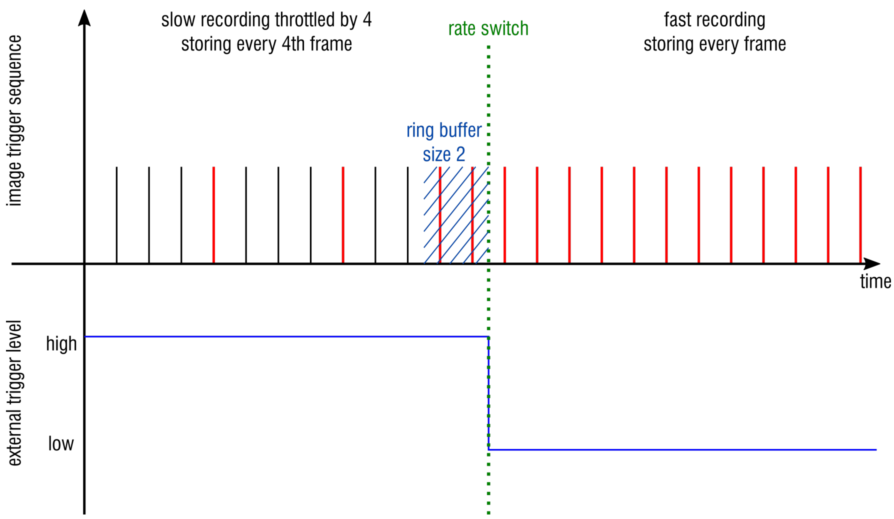


Figure 10.21: Change Frequency function principle

The functional principle of the **Dual Rate** mode is explained using the example of an external trigger. Black lines represent trigger signals sent to the camera. Red lines denote images that are stored to disk. The blue line in the lower part of the image represents the external trigger level. Throttling acts as a filter storing every Nth triggered camera image (here every

4th) when the external trigger level is high. Upon dropping the external signal, the recording rate is switched to fast mode, now storing all triggered images. The ring buffer temporary saves the last N received images at full rate (here 2) and dumps them to disk when the rate switch is triggered.

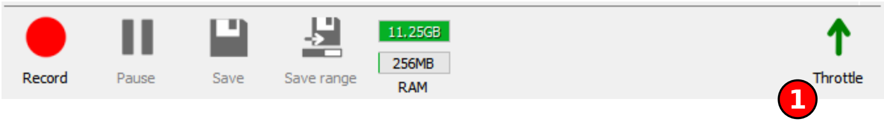


Figure 10.22: Toolbar including -Dual Rate

To define which parameter will change the recording frequency there are four different trigger options:

Digital: A digital trigger signal which is sent to the 'Start' input of the PTU.

Analog: If an ADC is used a threshold for an analog signal can be set to select the rate.

Button: The speed selection is toggled manually by pressing a button in the footer toolbar (cf. 1 in Fig. 10.22).

Extensometer: This option is only available with the **StrainMaster** project and its settings are described in the **StrainMaster** manual in detail.

10.5.3 Custom image tags

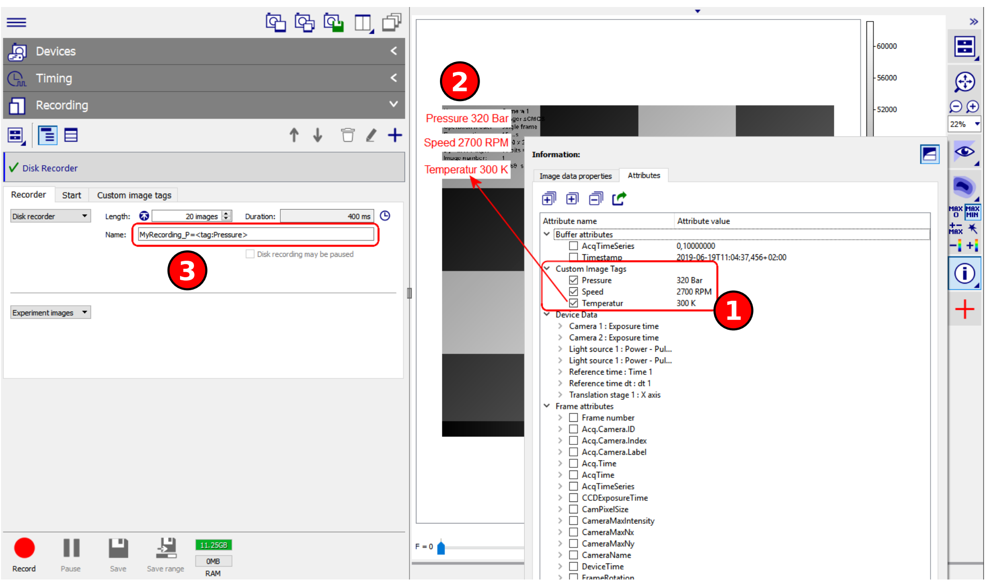


Figure 10.23: Custom image tags - Overview

With the **Custom image tags** the images acquired in **DaVis** can be tagged with user defined properties. A tag consists of a name and an associated value. The tags are defined on the 'Custom image tags' tab of the **Recording** section. How to define the tags is explained in Sec. 10.5.3 *Defining Custom Image Tags*). These values are stored as special buffer attributes in the **Information** list which is opened by the ⓘ button in the **image display's** toolbar. On the **Attributes** tab they are grouped in the *Custom image tags* 1 node. They can easily be superimposed on the image 2 itself by checking the corresponding attribute item. Also all image tags can be used as wildcards to generate the recording name 3 .

Example: *MyRecording_P=<tag:Pressure>* creates a recording name with *<tag:Pressure>* being replaced by the value of the tag 'Pressure' to *MyRecording_P=320 Bar* (cf. Sec. 10.5.1 *Recording name*).

Note: When editing the recording name 3 a list of available wildcards pops up each time a < is typed.

Defining Custom Image Tags

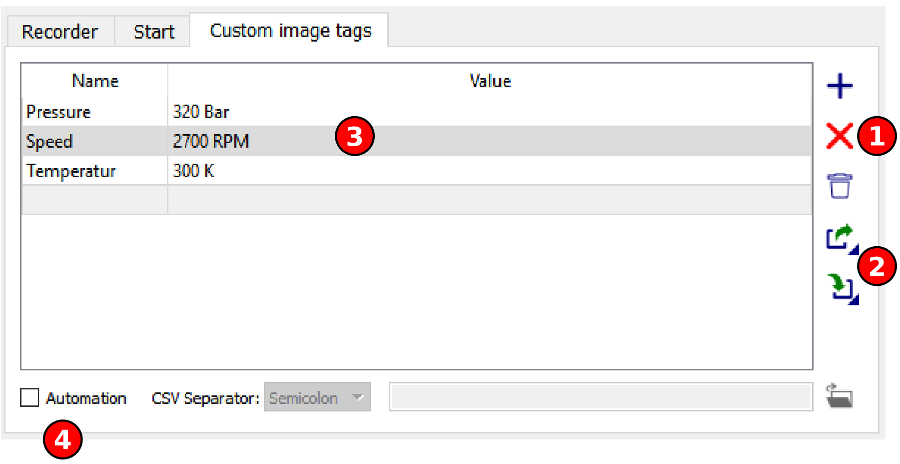


Figure 10.24: Custom image tags - Editor

Custom image tags can be set on the 'Custom image tags' tab of the **Recording** section (cf. Fig. 10.24). The controls on the right at 1 are used for editing the table (add or remove rows, delete the table). The controls at 2 allow to export and import the table items to e.g. edit them external (e.g. MS Excel or Libre Office).

The (name, value) pairs 4 can easily entered and edited in the table. To add a new empty item click the + button or by clicking in an empty row. Rows can be deleted with the × button. The entire list is deleted by clicking the button. Both names and values can contain any character. If they are intended to be used as file name wildcard, the value should only contain characters which are valid for file names.

The export button opens a file save dialog to save the table to a text file in `csv` format. The import button opens a file open dialog to load (and overwrite) the table with the contend of a file. The column separator can be selected in a drop down menu of the export/import buttons. A semicolon as separator is default for direct import into MS Excel.

Optionally for automation purposes 4 a file can be specified which is loaded prior to each recording to automatically set all tags. A possible scenario is a test bench which writes all environmental parameters to a text file and **DaVis** reads this file to store these values with acquired images. The automatically loaded files just add new tags or modify existing ones (e.g. replace their values). Other tags are not changed.

10.5.4 Scanning modes

There are two different scanning modes available, the hierarchical scan and the table scan. With hierarchical scans it is possible to automate the scanning of nested parameter fields where each parameter's value range is defined by a scan schema. The table scan allows to build up a large table of values for different parameters usually to scan sparse parameter fields.

Hierarchical Scanning

A hierarchical scan consists of different scan values which are altered during the scan depending on the selected scan scheme. Additional to scan values there are some control elements like loops, delays, CL command execution or a 'Wait for Trigger' item available. Beside that a recorder entry is always part of the scan sequence.

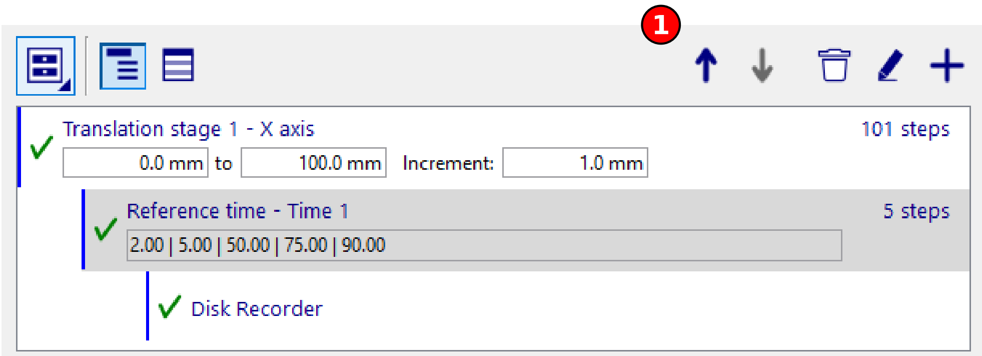
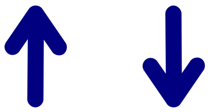


Figure 10.25: Hierarchical scan mode

The right hand side of the toolbar 1 in Fig. 10.25 provides the controls to manipulate the hierarchical scan sequence.

Change hierarchy

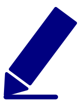


Moves the selected scan item a up or down in the sequence to move it through the levels of the sequence.

Clear scan sequence

Clears the entire scan. Only the recorder item remains.

Edit selected scan item



Enables the edit mode for the selected entry. This is equivalent to a double click on the scan item entry itself.

Add new scan item

Opens the scan item selection dialog to add a new item to the sequence.

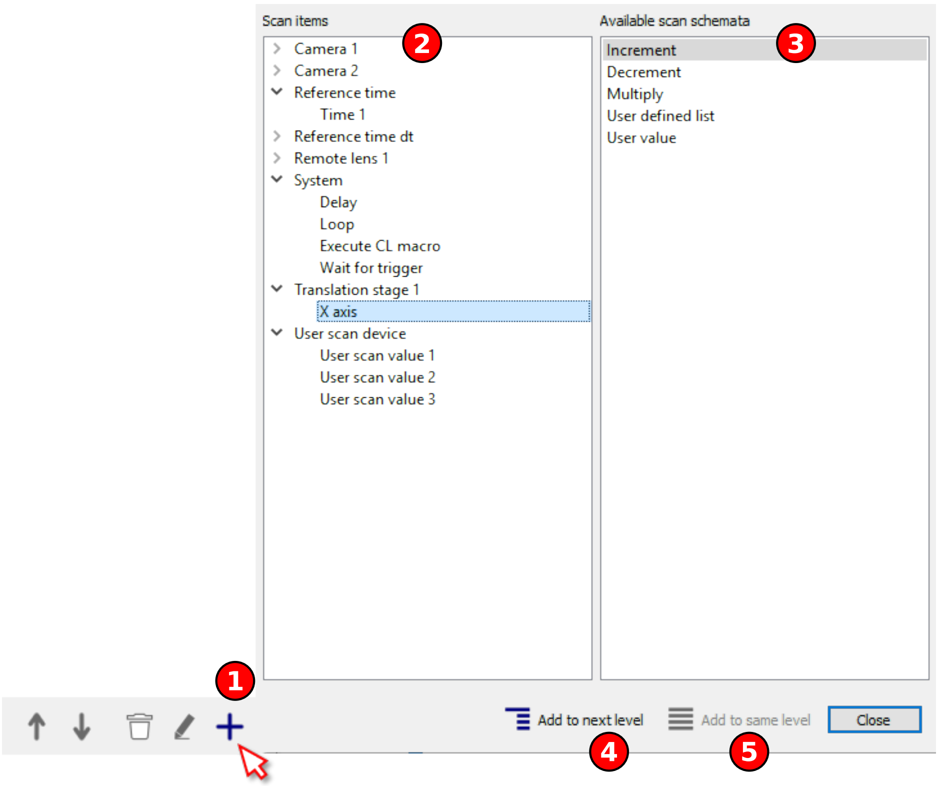


Figure 10.26: Hierarchical scan - Add scan item

To start building a scan sequence, open the scan item selection (+ button 1 in Fig. 10.26).

The left part 2 shows all available scan values and system control elements. The values are grouped by device or context. Once a specific value is in use, it disappears from the list of available values. Once an item is selected on the left, here the 'X axis' of the 'Translation Stage 1', the schemata available for the current selection are shown in the list on the right 3 . To add the selected value considering the selected schema simply double click the schema or press either **Add to next level** 4 or **Add to current level** 5 .

Available schemes are:

- **Increment:** increments the value in a given range with a given step size
- **Decrement:** decrements the value in a given range with a given step size
- **Multiply:** multiplies the value in a given range with a factor

- **Discrete values:** walks through a list of discrete values in a given range. The possible values are defined by the underlying device (e.g. an aperture scan allows only discrete aperture values)

- **User defined list:** a list of arbitrary user defined values

- **User value:** queries the next value interactively from the user during the scan

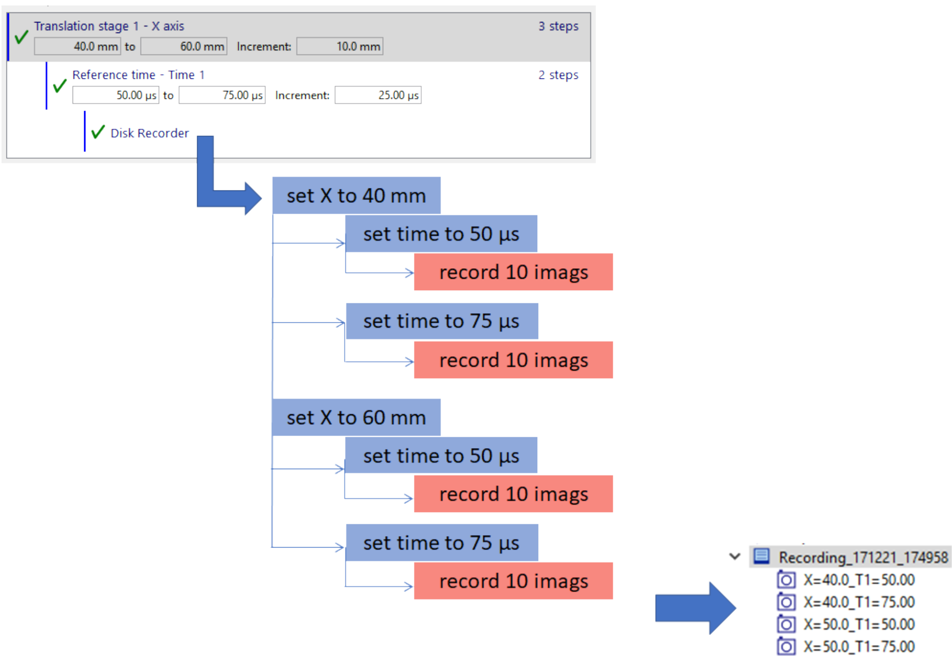


Figure 10.27: Hierarchical scan - An example for a nested two level scan

There are different ways of combining multiple scan items. One scenario is to scan a traverse along the x axis and scan the reference time at each x location. In this case the two scan items are nested on two levels. Another scenario would be to scan e.g. two axis X and Y of a traverse simultaneously to move from one 2D point to another. Now the two scan items for X and Y axis are arranged on the same level and change their positions together before recording at the next point.

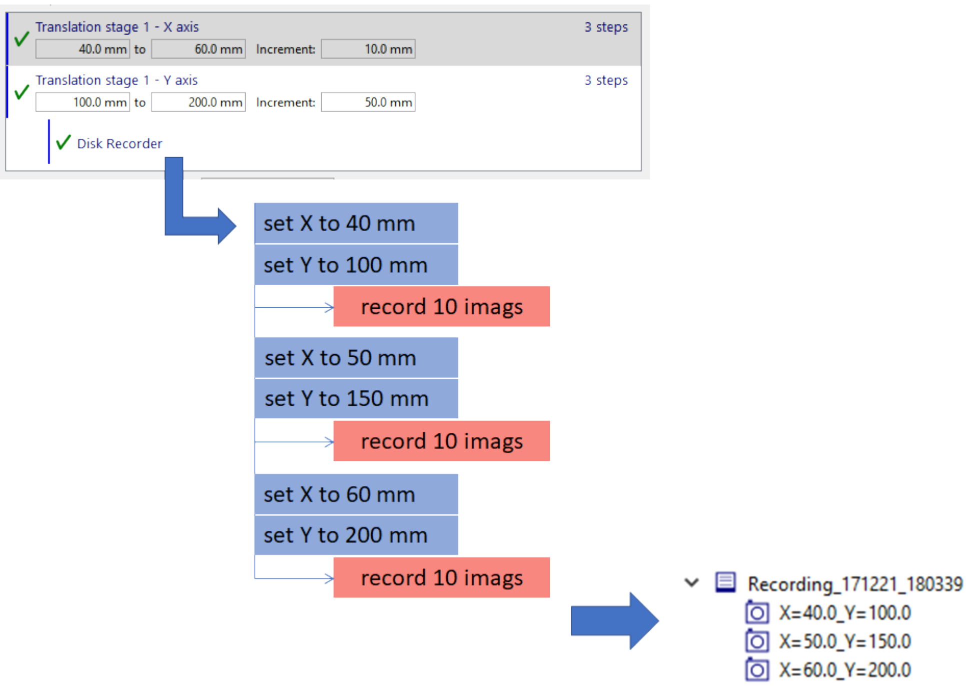


Figure 10.28: Hierarchical scan - An example for multi value single level scan

These differences are illustrated in two figures. Fig. 10.27 shows a nested scan of position and time while Fig. 10.28 illustrates a scan of two axes on the same level. The resulting recording multi set is shown (this detailed view of the multi set is only available if the 'Show subsets' option is activated in the project browser).

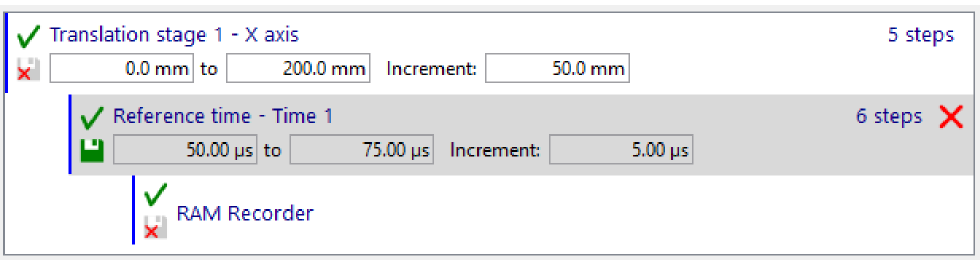


Figure 10.29: Hierarchical scan - An example RAM scan with save points

For RAM recordings it is possible to define the level where the recorded data should be stored to disk. These so-called scan points can be activated by clicking the disk symbol . Setting the save point to another level than the recorder allows to speed up the scan. Imagine a nested scan of location and time as shown in Fig. 10.29. Changing the reference time usually takes a single PTU cycle while moving the traverse to the next position might take a couple of seconds. So it might be desired to acquire all reference times into memory and then store the data while the traverse is traveling to its next position. The save point can only be set at levels which completely fit into the available recording RAM. If a sve point is not possible, **DaVis** will indicate this.

In cases a certain scan item should not be used during a scan, it can either be removed from the sequence or just be temporarily disabled by clicking the ✓ on its left side.

Table scan mode

In this mode **DaVis** executes the recording by interpreting the table items row by row. The goal of a **Table scan** is to selectively set the measurement points instead of scanning continuous parameter fields. This can be useful if just specific e.g. positions of a 3D traverse system should be recorded. For ease of use in complex scenarios the table scan sequences can be created externally (e.g. in MS Excel) and imported into DaVis. For systems using a robot it is recommended to scan the target positions using the table scan since it allows to add safe way points (scan points without recording images).

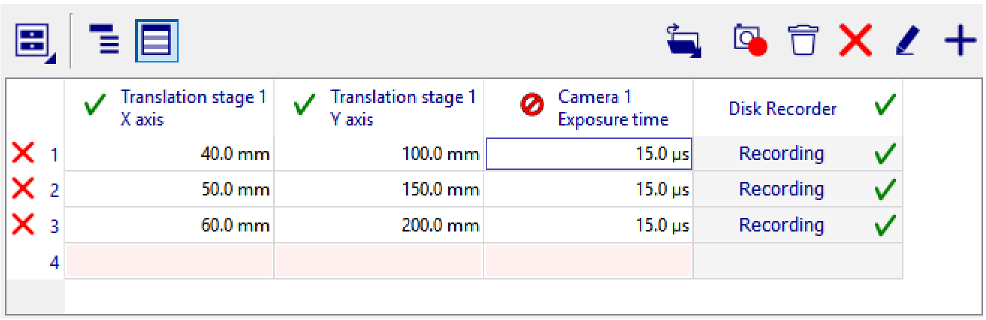


Figure 10.30: Table scan mode

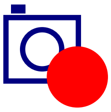
In the table itself columns can be activated (✓) and deactivated (⊘) for different scans. Rows can be removed by pressing the delete button (✗) (cf. Fig. 10.30).

For setting up the individual recording columns of a scan table the following buttons are available at the left of the **Table scan's** toolbar:

Import/Export CSV

Import or export a CSV table to handle templates of scan tables. E.g. to create complex scan tables it might be advisable to create a short sample table containing all necessary columns, export this as CSV table, complete it with all measurement points as external spread sheet and finally reload it as scan table.

Dry run



Executes the scan as defined in the scan table without acquiring images for testing the scenario.

Clear scan sequence

Clears all previously set scan rows from the table.

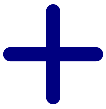
Remove current column from table

Removes a single selected column with all its items from the scan table.

Edit selected scan item

Enables editing the value of a single selected scan item. It has the same effect as double clicking the item itself.

Add New Scan Item



Opens the scan item selection dialog to add a new item to the table. The dialog is shown in Fig. 10.31 and is explained there in detail.

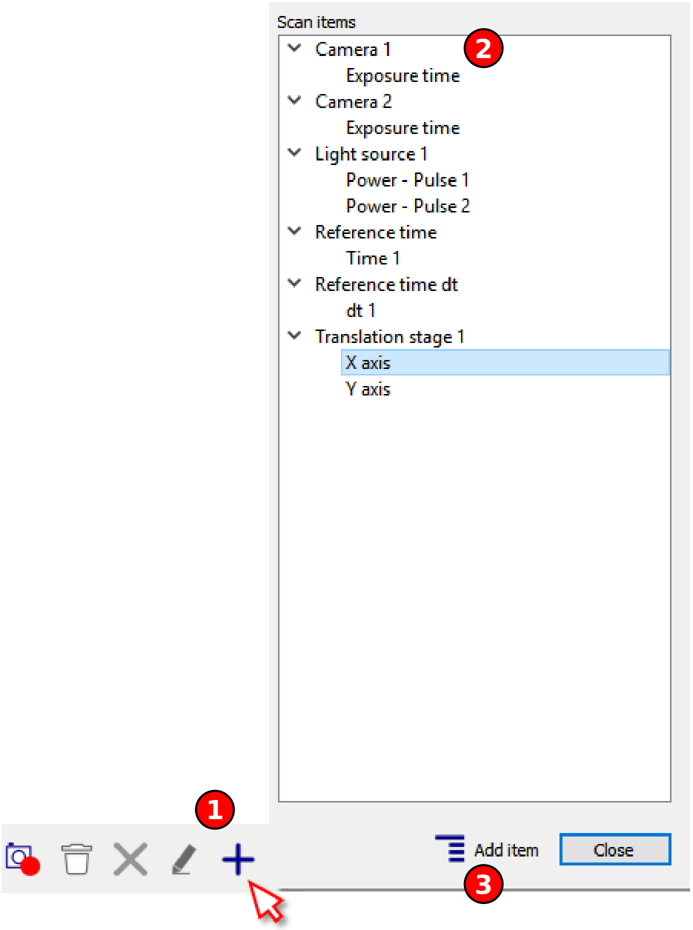


Figure 10.31: Table scan - Add scan item

To start building a scan sequence, open the scan item selection (+ button 1 in Fig. 10.31).

A tree view 2 shows all available scan values. The values are grouped by device or context. Once a specific value is in use, it disappears from the list of available values. To add the selected item simply double click the item or press **Add item** 3 .

In Fig. 10.32 a scenario of a three rows table scan is shown to illustrate the process. The example shows a scan with two axis X and Y of a traverse together with the reference time T2. All three scan values a re set to their values simultaneously before recording the desired images at the given scan point. This is repeated three times to record all scan points. After such a scan the data is stored in a multi-set with a set for each scan point. The detailed view of the multi-set as shown in Fig. 10.32 is available if the 'Show subsets' option is activated in the project browser.

Figure 10.32: Table scan - An example for a X/Y/T scan

11 Processing

The **Processing** dialog (Fig. 11.1) can be used for automatic processing of an arbitrary number of operations on a data set.

The dialog can be reached using the **Processing** button in the **Project Manager**.

11.1 Sections of the Processing Dialog

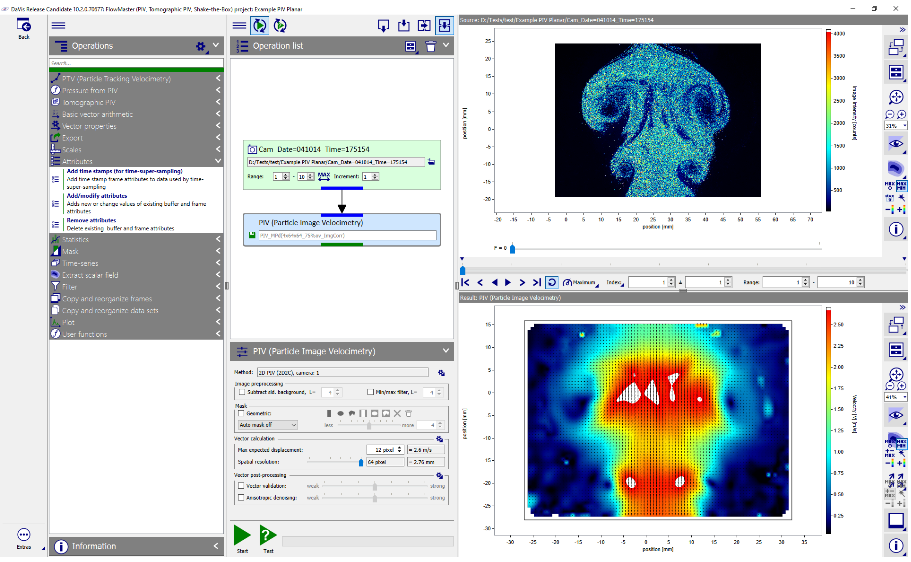


Figure 11.1: Processing dialog

The dialog is subdivided into three parts:

- the **Operations** section in the left part, where all currently supported operations are listed, and the **Information section** in the lower left;
- the middle part with the **Operation list** (top) and the **Parameter** section (bottom);
- the right part with the views on the source set and the intermediate or final result set respectively.

On the first startup, the **Operation list** contains only the item for the source set (green in Fig. 11.1). With a double click on an operation in the **Operations** section, it is possible to append an operation item to the **Operation list**. After that, this item is highlighted as selected in the **Operation list** and the **Parameter** section shows the corresponding parameters.

The **Start** button starts the processing of the **Operation list** for all set elements in the selected range with the selected increment. If storage is enabled (default), the result sets are stored in a subfolder of the respective source set in the current project.

With the **Test** button the operations are executed only for the displayed set element of the source data set. Test processing without source display will execute the operations to the first file of the selected range. The progress bar shows the status of the complete calculation.

Further, it is also possible to **minimize/maximize sections**. The “Hamburger" button in the **Operation** section as well as the **Operation list** and **Parameter** section switches the corresponding sections to a buttonbar to the left side and back. So the image display area can fill most of the **Processing** dialog. Once minimized, the sections can still be accessed through respective buttons in the buttonbar. The header and footer buttons and controls are included in this buttonbar to stay visible.

During processing, **display updates can be turned on/off**. By default display updates are enabled. As soon as the processing starts,the result image gets updated and shows the current applied operation.

If the button **Live processing** is enabled, a test processing is performed on the current image, whenever a new operation is included into the operation list, or a parameter is changed. This choice enables a better overview of the processing, since the result of a previous operation is utilized for adjusting parameters of a new operation. There is a notification, if some operations with specific parameters do not support live processing . This is the case when time-consuming operations, for instance tomographic PIV, are processed, or CL-coded operations appear several times in the operation list. To show the result of a specific operation, press the **Pinning** button  of that operation in the operation list.

11.2 Operations

Depending on the result data type of the selected item in the Operation list, all supported operations are listed here. Each operation has a short description and some internal tags (such as "smoothing", "masking") to find a certain operation when using the **Search** input field.

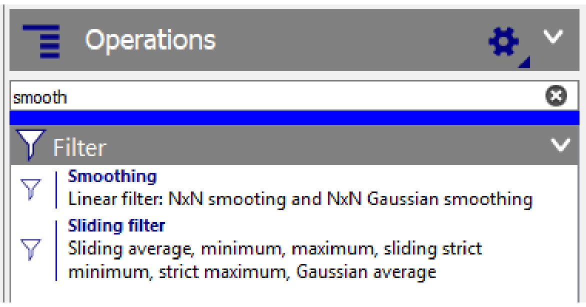


Figure 11.2: The **Operations** section lists all operations available for the selected source set – here restricted to operations matching the search term

The colored bar below the **Search** field shows the input data type of the listed operations. Common data types are:

Data type	Color
Image	blue
Vector	green
Plot	violet

Operations can be added to the **Operations list** via

- double click: When using a double click on an operation, it is always inserted directly after the active item in the **Operation list**.
- drag and drop: An operation can be added to the **Operation list** by dragging an item into the **Operation list**. It is possible to insert an operation between certain operations by dropping it there. A red arrow in the **Operation list** indicates that the input and/or output data types are wrong, if the operation is inserted at this position.

Favorite and recent operations (Fig. 11.3):

Recently used operations: This group shows the last used operations for the current data type. The list can be cleared by using the trash can button on the **Recently used operations** header.

Favorite operations: The **Favorite operations** group appears when at least one operation is marked as favorite using the yellow star button in the lower right part of each operation.

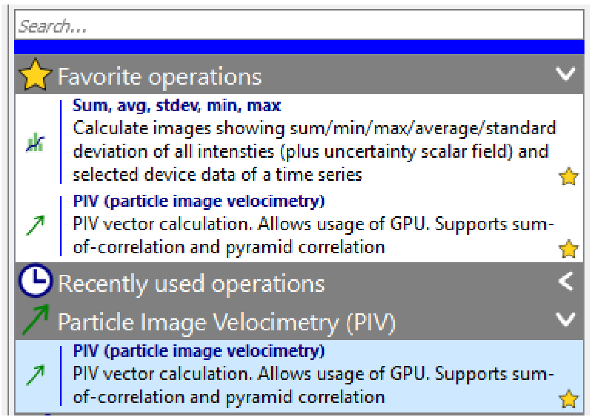


Figure 11.3: The **Favorite operations** and **Recently used** groups in the **Operations** section

11.3 Operation List and Parameter

The **Operation list** (Fig. 11.4) defines an arbitrary number of operations to be executed on the source data set. The result of the first operation is used as source of the second operation and so on.

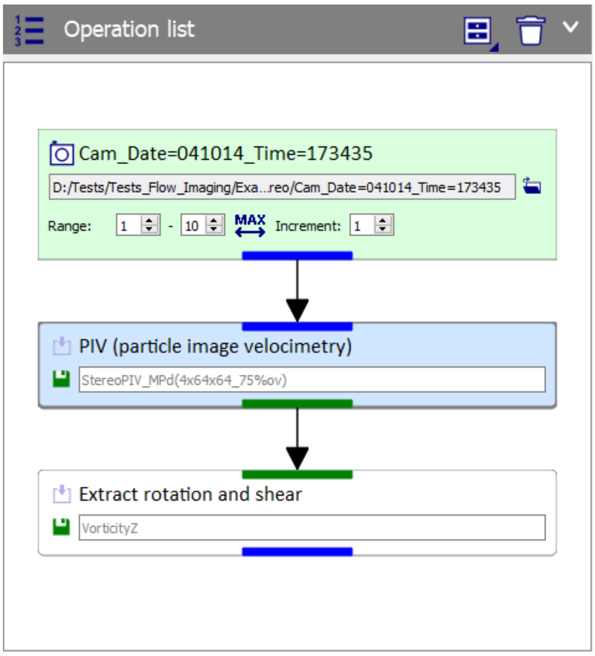


Figure 11.4: The **Operations List** with items added from the **Operations** section (via drag and drop or doubleclick). The top item is the source data set. The middle item is selected.

The currently selected operation in the list is marked with a light blue background. This selection can be changed by a mouse click on an item and affects the **Operations** section, which always shows all supported operations for the selected item. Additionally, the corresponding parameters are shown below the **Operation list**.

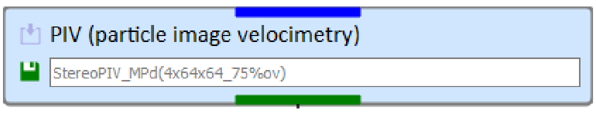


Figure 11.5: An operation item. Note the top and bottom bars color-coding the input and output datatypes.

Within the operation item it is possible to change the storage which determines if and how the results are stored:

Save result: The result is stored in the current project in a subfolder of the source data set. Each operation has a default storage name which can be overwritten.

No storage: The results are not stored in the current project but temporary on hard disc. When the storage is off the software will show a Warning above the progress bar.

Overwrite source: Overwrite source data set. This option is available for a few functions only.

The colored bar in the top area of the item indicates the input data type, the bar at the bottom indicates the output data type. A red cross between operations indicates that the input and output data types are different. In that case the **Operation list** is invalid and it is not possible to execute the processing.

11.4 Information

Messages: Shows all errors that occurred during processing. Additionally some information messages are available (e.g. processing time). All messages contain a timestamp.

Log: Shows all messages from the CL InfoText command.

11.5 History

The processing operation list is stored together with the operation results. No information about old processings are lost. A click with the right mouse button to the **Project** tree opens a context menu. Select **Processing History** to open a dialog like in figure 11.6.

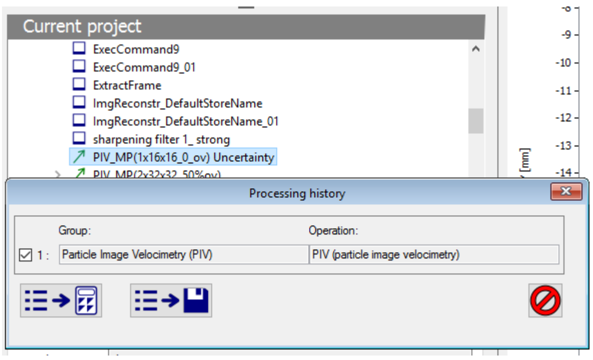


Figure 11.6: History of processing.

This dialog displays a list of all operations which have been processed to get from the source image to the selected result. Each operation of this list can be selected seperatly. Press the left button to load the selected operations as new active operation list. The **Processing** dialog is not opened automatically. At first a source set has to be selected, then open the **Processing** dialog and process the selected source set with the old operation list.

Press the second button to store the selected operations to hard disc. This exported list can be imported later into the **Processing** dialog.

The right button cancels the history dialog and returns without changes of the active operation list to the **Project** manager.

11.6 Hyperloop

The **Processing** dialog is able to process an arbitrary number of datasets with one mouse click. This can be done using the **Hyperloop** dialog (see page 81). In this case the dialog provides one additional option:

Use full data range for all datasets: This is a very useful option if the number of files differs between the selected datasets.

11.7 Processing operations for all data sources

The operations of this section can be executed for all types of data sources: images, profiles and vectors.

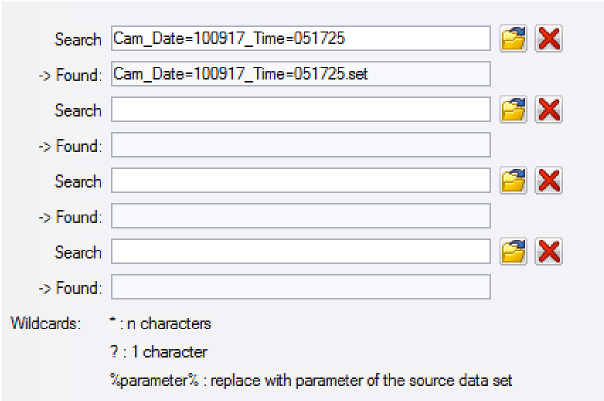
11.7.1 Operation: Copy and reorganize data sets

copy: Copy the source set using the selected range and increment of the **Processing** dialog and create a new set with less files.

copy selected data: Enter a list of set items or ranges of set items and create a new set with this files. With definition `1-3;5;7-10` the files 1 to 3, 5 and 7-10 would be taken from the source set and thrown as files 1 to 8 into the destination set. Sources files 4 and 6 are skipped. With definitions like `-100` or `20-` either the first 100 files or the files starting with index 20 up to the end are copied. Use a definition like `!5` to skip single files of the source set.

reverse order: Copy all sources and reverse the storing order.

merge data sets to multi frame: Merge all sources with the corresponding file of the specified datasets to a multiframe buffer:



Press the **folder** button to select a dataset or enter the name of the dataset in the edit field. The **cross** button can be used to clear the current entry. It is also possible to use the following wildcards to find a corresponding dataset:
* : n characters
? : 1 character
%parameter% : replace with parameter of the source dataset
The files of all found datasets will be merged.

append data set: Copy the source data set and append the buffers of the specified data set. See operation **merge dataset**.

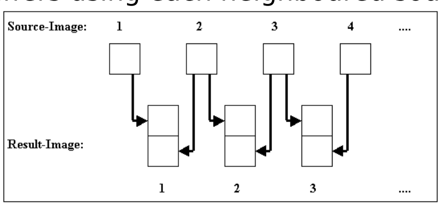
11.7.2 Operation: Copy and reorganize frames

extract frame: Extract the specified frame of every source buffer.

reorganize frames: Reorganize the frames of every source buffer. Choose for every source frame between the **destination frame** number and **delete**.

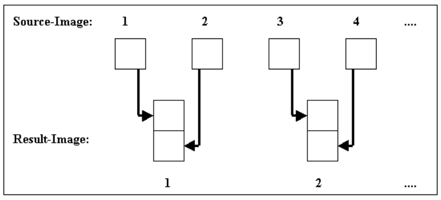
create multi-frame buffer from time series: Choose between this options in the parameter dialog:

- **create (n-1) images: 1+2, 2+3, 3+4...:**
  Create two-frame-buffers using each neighboured source files.

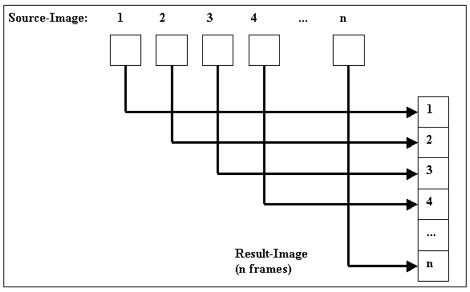


- **create (n-1) images: 1+2, 1+3, 1+4...:**
  Same as above but always use the first buffer as first frame.

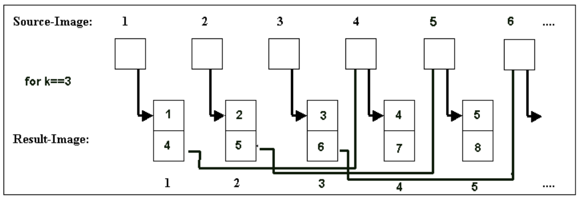
- **create (n/2) images: 1+2, 3+4, 5+6...:**



- **create 1 image with n frames: 1+2+3+4...:**



- **create (n-k) images: 1+(1+k), 2+(2+k), 3+(3+k)...:**



create time-series from multi-frame buffer:

- **create one buffer for each frame:**
  Extract all frames of the source buffers and store in different buffers of the result set. Start with first frame as first buffer, followed by the other frames of the first buffer, then followed the same way by the other source buffers.

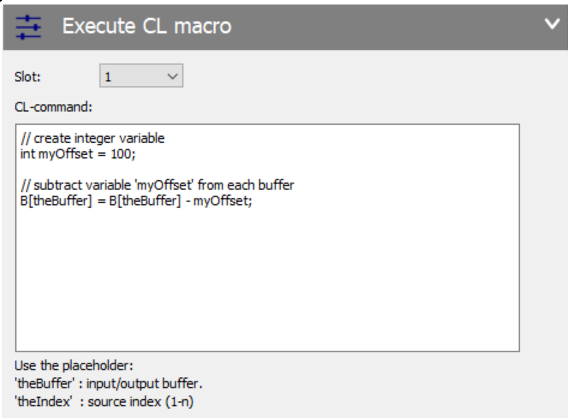
- **create one buffer for each time stamp:**
  Collect all frames with the same time stamp into the same buffer. If the source set had n files with each m frames then the resulting set is created with m files with each n frames.

11.7.3 Operation: User Function

execute CL macro 1-9: Execute user defined macro commands for each source buffer. Use the following placeholder:

- theBuffer: Input/output buffer. The source buffer is taken from buffer *theBuffer*. At the end of processing the result must be copied into buffer *theBuffer*.
- theIndex: Source index. The value is one for the first processed source and then increased up to the number of sources.

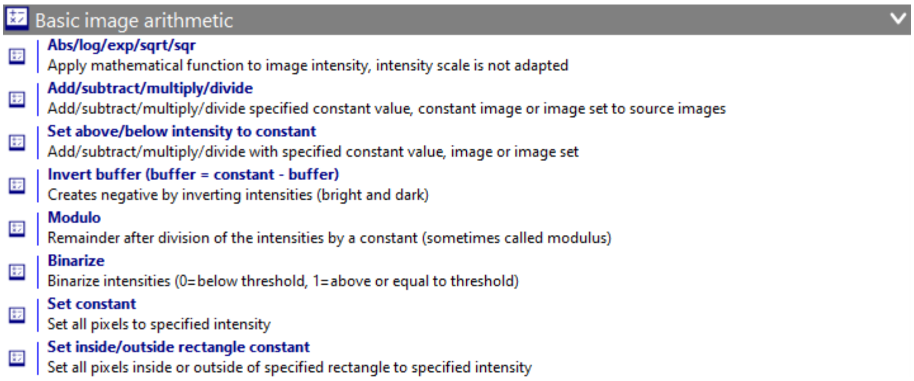
The following example subtract a constant offset of 100 counts from the source image.



12 Processing Images

12.1 Operation: Basic Image Arithmetic

The basic arithmetic gives access to simple mathematical operations with constant values or other data sets. Other functions are used to apply masks or to change the buffer type or dimension.



12.1.1 Mathematical Functions

Apply a mathematical function to each pixel of the source image.

Absolute: Get absolute value.

Logarithm: Calculates natural logarithm. Result is undefined for negative pixel intensities.

Exponential: Calculates the exponential function. A too large result will be a floating point overflow value.

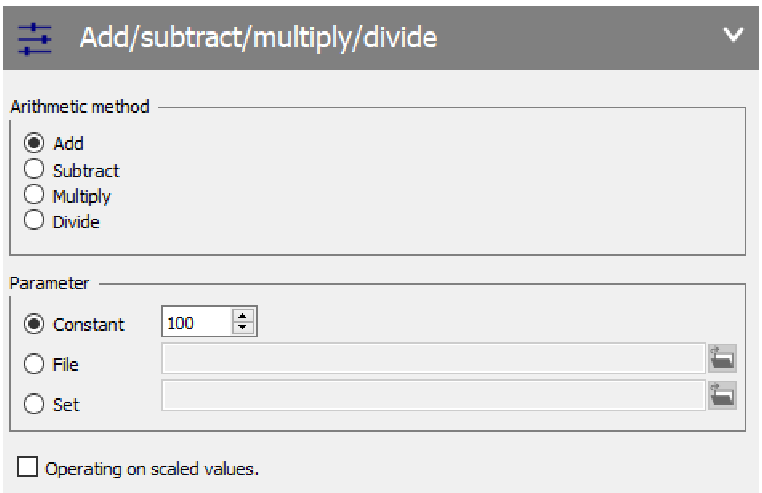
Square root: Calculates the square root. Result is undefined for negative pixel intensities.

Square: Pixel intensity is squared.

12.1.2 Add, Subtract, Multiply, Divide

Add, subtract, multiply or divide values from or to an image. The **constant** value (in unscaled counts) is operated to each pixel of the image. The second image can be taken from a fixed **file** or from another **set** of images. In the last case, the images with the same index in both sets are processed.

If the operation is done with another image, then the values of pixels at the same pixel position are operated. The operation works on the raw intensity by default and does not take different intensity scales into account. If **operating on scaled values** is selected, then the intensities of buffers with different intensity scales are normalized to the same scale before operating both intensities. However, in this case the scale units must be equal.



12.1.3 Set Pixel to Constant Value on Condition

Set constant

Set all pixels to specified intensity.

Set above/below intensity to constant

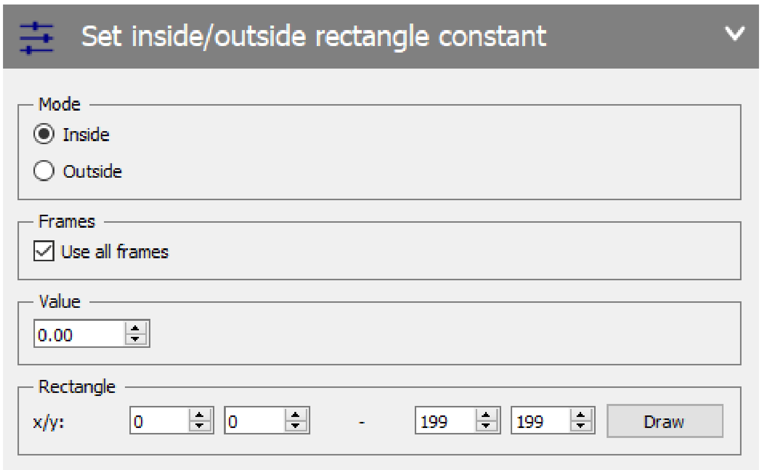
Set all pixels **above** and/or **below** a specified value to a **constant** value.

Set inside/outside rectangle constant

inside rectangle = const: Sets all pixels inside the rectangular region to a constant **value**. Press button **Draw rectangle** to define a region with the mouse in the source image. The coordinates can be edited directly in the dialog by their unscaled pixel position.

By default the region is applied to the source buffer once. However, when the same rectangle should be applied to each frame, then select the rectangle in one frame and change to mode **Use all frames**.

outside rectangle = const: Set all pixels outside the rectangular region to a the constant **value**. Same interface as in the last function.



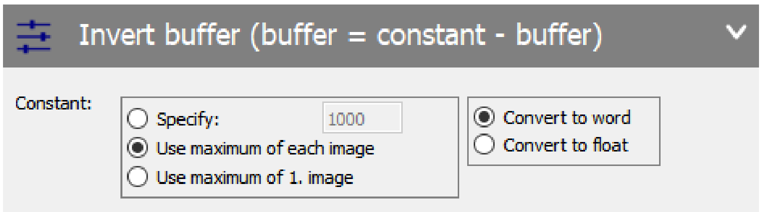
12.1.4 Invert buffer (buffer = constant - buffer)

Invert the source image using a constant value. Then **convert** the result to data type word or float.

Specify: Enter the constant value.

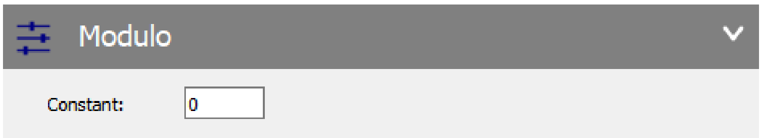
Use maximum of each image: Calculate the maximum intensity from all source images of the complete data set and use this as constant value for the inversion.

Use maximum of first image: Calculate the maximum intensity from the first source image and use this as constant value for the inversion.



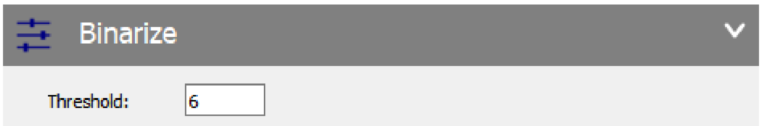
12.1.5 Modulo

Calculates pixel value modulo a **constant** value for images of type WORD.



12.1.6 Binarize

Set all pixels with an intensity above the **threshold** to 1 and the remaining pixels to 0.



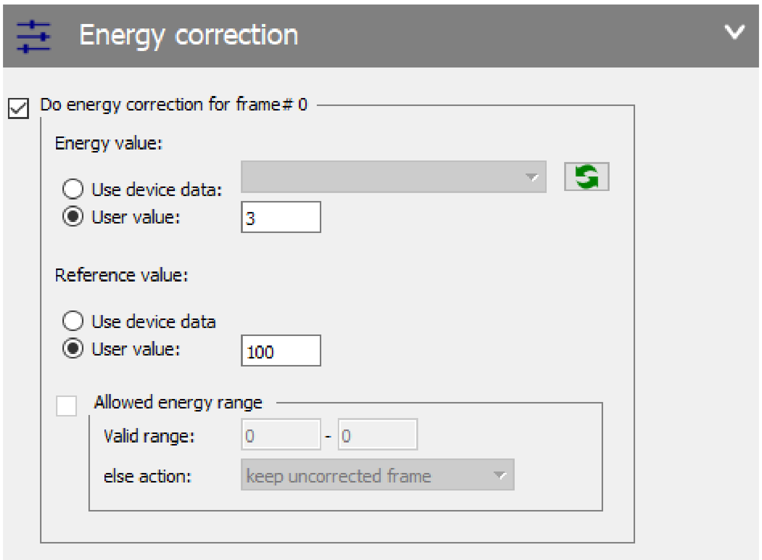
12.2 Operation: Intensity Correction

12.2.1 Energy correction

Correct the intensity level of an image with the help of the energy value of a device. At first, select the **Energy value** either from a constant **User value** or from a **device data** attribute.

As second step, define a reference value. If the energy value is equal to the reference value, then no intensities would be changed. The reference value can either be taken from the **device data**, because most device data get their default reference value. Or the reference value can be defined by the **User**.

If selected, the energy correction can be restricted to those images with the energy value in a certain **range**. If the value is outside of the range, the image can either be left uncorrected, or set to 0 intensity, or simply not stored in the destination SET.

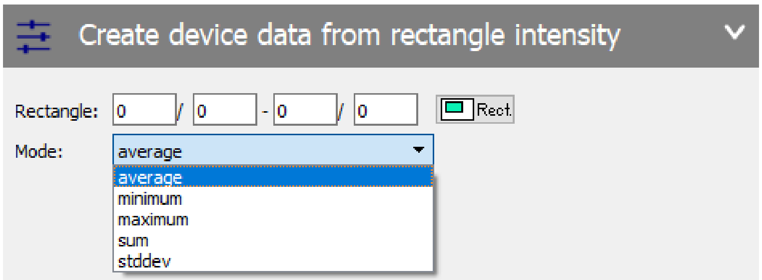


12.2.2 Create device data from rectangle intensity

The intensity of a rectangular area in the source image is calculated and stored in the result as new device data. Available **Modes** are the average, minimum, maximum, or sum. Press button **Rect** to define the rectangular area by two mouse clicks in the source image.

This function can be used to correct the intensity level of an image corresponding to the level in a rectangular area. Therefore use the **extract intensity** function as first operation in the operation list, followed by an **energy correction**.

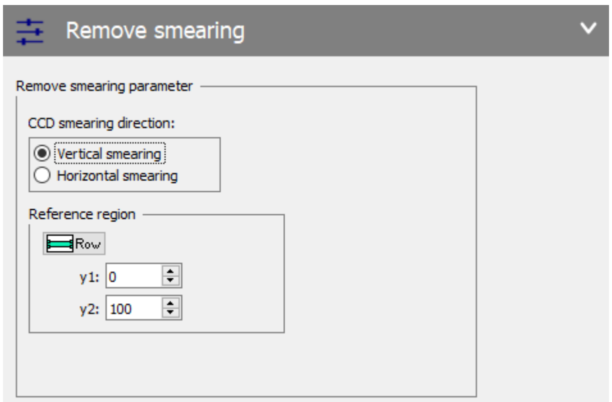
Press button **Rect.** to define a region with the mouse in the source image. The coordinates can be edited directly in the dialog by their unscaled pixel position.



12.2.3 Remove smearing

Because of the restricted shutter ratio of a CCD and the continuously lightning, a part of the image information is smeared in vertical direction. In case of short exposure times, this part is high enough compared to the general image intensity to influence the measurement. In some systems the limit is at about 1000μs even the camera could have been be operated with 40μs.

Options of this intensity correction are **vertical smearing** or **horizontal smearing** as CCD smearing direction. A **reference region** must be defined by two mouse clicks into the source image as range of rows or range of columns. The operation calculates the average in direction of the range and then subtracts this row/column from the complete image.



12.3 Operation: Image properties

12.3.1 Data Type Conversion

word → float: Changes the buffer type to float.

float → word: Changes the buffer type to word.

12.3.2 Scalar Fields

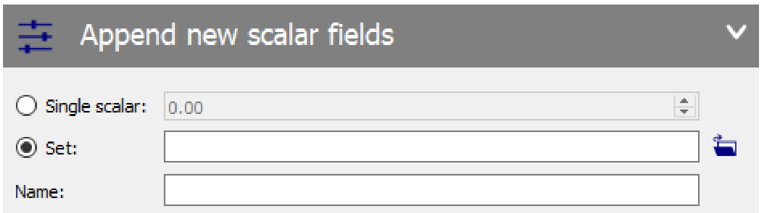
Scalar fields are additional information for each pixel. By default, an image includes for each pixel the intensity value. However, some operations create additional values, e.g. the height value of vector processing, and store this as a new scalar field to each pixel. Scalar fields are defined by a unique name.

Extract attached scalar fields

Select the name of the scalar field to be **extracted** into a resulting image.

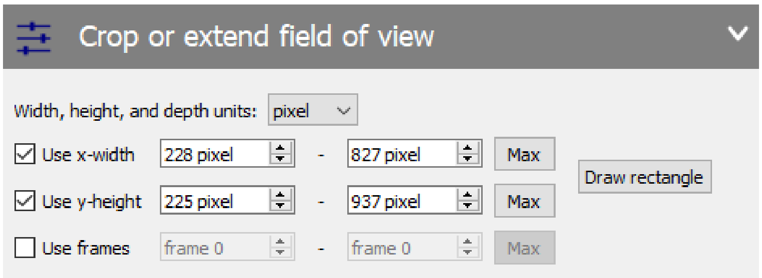
Append new scalar fields

Create a new scalar field of the given **Name**. Either set this component to a **single scalar** value or use a **set** of image files and take their intensity values as initial value for the new scalar field. In this case, the corresponding pixel positions are taken.



12.3.3 Crop or extend field of view

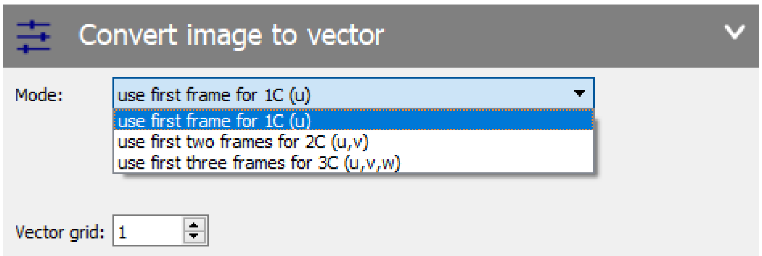
This function crops (removes pixels or a scaled length at the rim) or expands (adds 0-value pixels at the rim) the image size without changing remaining pixels. Select a width, height, and depth unit between **pixel** or **scaled**. Each dimension of width, height, depth and frame number can be changed separately to a new size or left untouched. Press button **Draw rectangle** to define the rectangular area with the mouse in the source image.



12.3.4 Convert image to vector

Convert image data into simple vector format. Useful e.g. after importing vector files from other software.

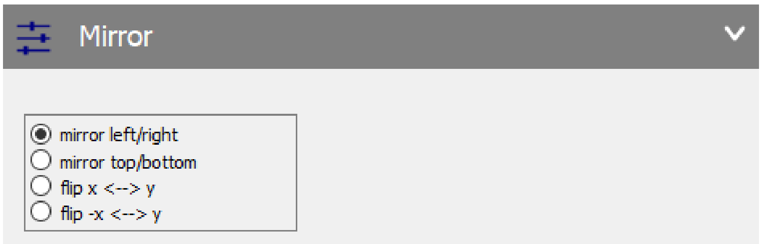
The source image must include one frame for 1C vector results, two frames for 2C vector results or three frames for 3C vector results. The complete first frame is taken as u-component, the second frame as v-component and the third frame as w-component.



12.4 Operation: Image mapping

12.4.1 Geometric Transformations

mirror: Mirror/flip all source images. Choose between **mirror left/right**, **mirror top/bottom**, **flip x/y** or **flip -x/y**.



rotate: Rotate all source images:

- Rotate around the image center by 90° left/right or by 180° . The scales can be rotated optional. If selected, then the origin of the result will be at the same image position as before.

- Specify: Rotate images by a **specified** angle. The center of the rotation can either be the center of the image or a free defined pixel position with **X** and **Y** coordinates. Depending on the angle the resulting image may become larger than the source image. Select **fit image size** to increase the size of the result so it includes all pixels from the source.

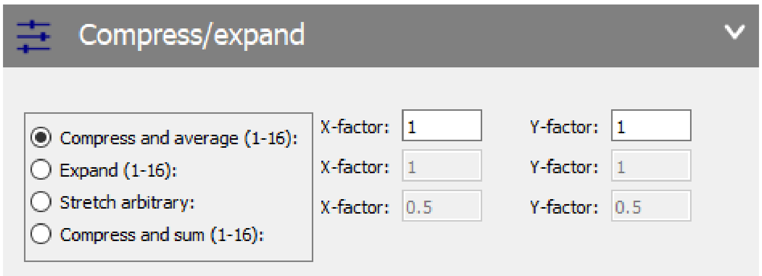
compress and expand: Change the size of the source images:

- Compress and average (1-16): Compress all images by an integer factor between 1-16 and average all source pixel which are combined into one result pixel.

- Expand (1-16): Expand all images by an integer factor between 1-16.

- Stretch arbitrary: Resize all images by an arbitrary factor.

- Compress and sum (1-16): Compress all images by an integer factor between 1-16 and sum up all source which are combined into one result pixel. This works like software binning during recording.



deform with vector field: Deform an image with a specified vector field. The vector field can be either a single file to deform all images by the same vector field. Or select a vector set to deform each image by the corresponding vector field.

12.4.2 Image correction and distortion

perspective correction (raw to world): Apply the correction coefficients of a camera to all source images.

perspective distortion (world to raw): Apply the inverse correction coefficients of a camera to all source images.

12.4.3 Image Reconstruction

Apply a geometric image correction or distortion and/or a pixel shift correction using specified vector field to an image. The result will be a float buffer.

In the group box **Perspective correction or distortion** checking the **Apply** checkbox switches on the image correction / distortion. An additional parameter is the z position for which this is done. The default is at $z = 0\ mm$ which is the position of first view calibration plate and usually also the light sheet position in PIV. If you need to readjust the value to a different z position (second option in the radio button) to match images from two mapped cameras, you may have to think about doing a stereo self calibration to readjust the $z = 0\ mm$ plane of the calibration.

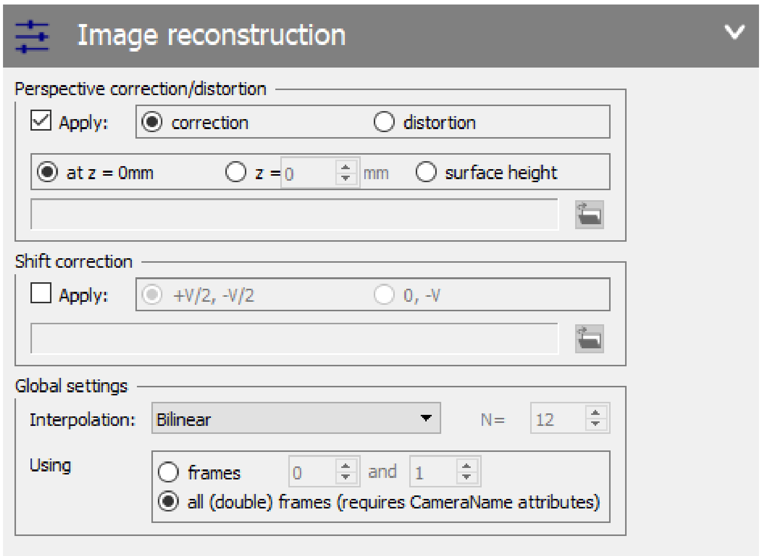
In case you have a surface height buffer (e.g. StrainMaster 3D or Surface-Flow package), the surface height file (IM7) holding the z height information for every pixel in world coordinates can be specified. It will be used to correct all images in the source set. Or you specify a set file (SET) with the height information for every buffer in the source set. Then the Nth file in the surface height set is used as the reference height to correct the Nth image of the source set. Press the **open file** button to select a different source for the reference surface height(s).

In the group box **Shift Correction** checking the **Apply** checkbox switches on to undo an additional pixel shift in the source images using the given reference vector field. If you select a single vector field (VC7) this information will be used for all images in source set.

Also select if to take out the specified shift from the second frame only (**0, +V**) or if to take out half the shift symmetrically from both frames (**-V/2, +V/2**). While the asymmetric shift (**0, +V**) is usually done in StrainMaster systems, the symmetrical shift (**-V/2, +V/2**) is the default case for PIV. Press the **open file** button to select a different source for the reference vector field(s).

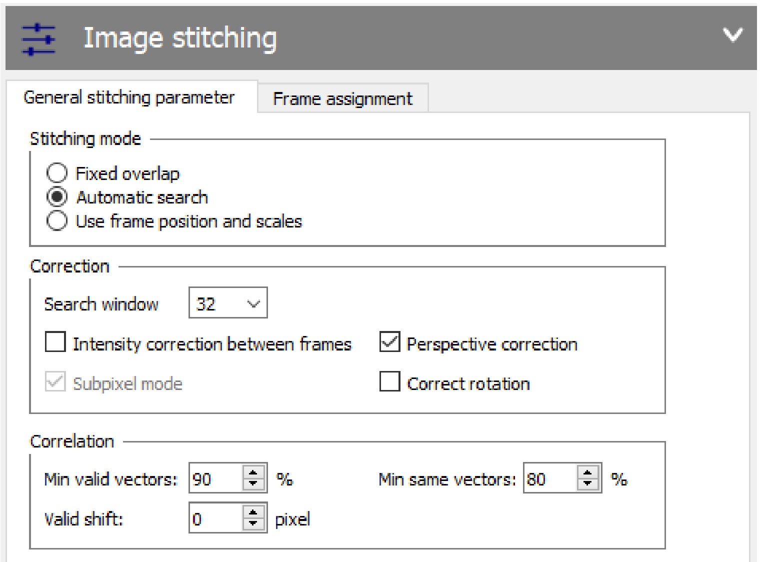
In the group box **Global settings** the interpolation algorithm and the frame range to which the image reconstruction is applied is selected. While the **bilinear** interpolation using only the nearest neighbors to the left, right, top, and bottom is very fast, it has a smoothing effect on the pattern. On the other hand, **Whittaker** reconstruction and **Lanczos** interpolation use all pixels in a 10x10 neighborhood. While they are significantly slower, the corrected images will maintain the sharpness of the original. **Bicubic** uses a bicubic interpolation of the neighboring 4x4 pixels. B-spline-6 uses a basic spline of degree 6.

The frame range is specified to treat either all frames of a buffer (**all (double) frames**) or only one or two user specified frames. In the first case all frames are scanned for matching pairs of double frames belonging to the same camera and the transformation is applied to those. Matching double frames is important to apply the shift correction properly. In the other case, the two specified frames are used and treated as a double frame even if they belong to different cameras. All frame numbers start from 0. Specifying a frame number of -1 means to ignore this frame. So, a frame combination of 0 and 1 will correct the first (0) and the second (1) frame in the buffer applying a shift correction of $(0, V)$ or $(-V/2, +V/2)$ as indicated. A frame combination of 0 and -1 will correct the first frame only applying a shift correction of 0 or $-V/2$ as indicated. A frame combination of -1 and 0 will correct the first frame only, but this time applying a shift correction of V or $+V/2$ as indicated.



12.4.4 Image Stitching

image stitching (frame): Perform image stitching for multiframe images from multiple cameras.



On the general parameter card the stitching mode can be selected between:

Fixed overlap: Uses the fixed overlap for all (16, 32, 64 or 128 pixel) frames that may be specified in the Overlap selection box below. A search algorithm tries to find the correct stitching position within the specified overlap.

Automatic search: A search algorithm uses the specified window size (16x16, 32x32, 64x64, 128x128, 256x256 pixel) to scan the overlap region for the correct position. The search window may be specified in the Window selection box below.

Use frame position and scales: Uses the position and scale of each frame only to arrange the frames in the stitched image.

Search window: Size of window used for pattern matching in automatic mode to adjust shift for image stitching e.g. 64x64 pixels.

Additional correction options are:

Intensity correction between frames: Applies an intensity correction during stitching of the frames to avoid an intensity gradients.

Subpixel mode: Will make sure that the shift of the frames with respect to each other will be detected by correlation with subpixel accuracy.

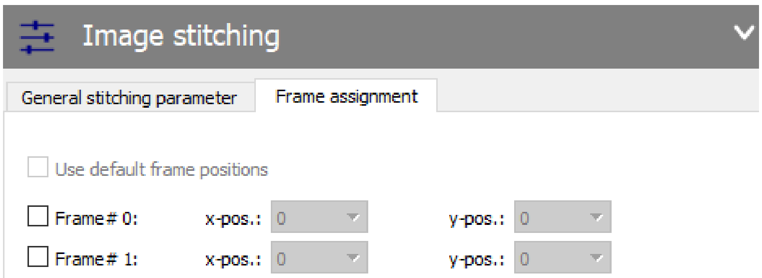
Use image correction: Will apply the image correction that is valid for the corresponding camera before starting image stitching.

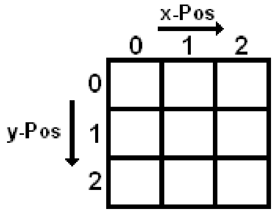
Correlation settings: Minimum valid vectors: Allows to specify the minimum of the valid vectors detected by the correlation function. The higher the value the more accurate is the detected overlap.

Minimum same vectors: Allows to specify the minimum of same vectors, i.e. vectors that indicate the identical shift. The higher the value, the more accurate is the detected overlap.

Valid shift: Allows to specify an amount of pixels that can be corrected within the stitching shift.

Frame assignment: Defines the global arrangement of the frames in the patched-up image.





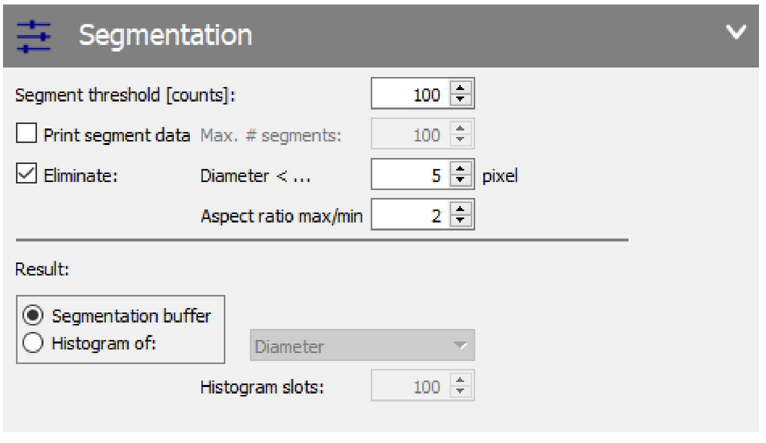
12.5 Operation: Image Analysis

12.5.1 Segmentation

Computes segmentation regions of the input buffer. The detected segments are marked with intensities 1,2,3,... in the output buffer. The **Segment threshold** (in counts) defines a segment as a region of pixels where all pixel intensities above the threshold.

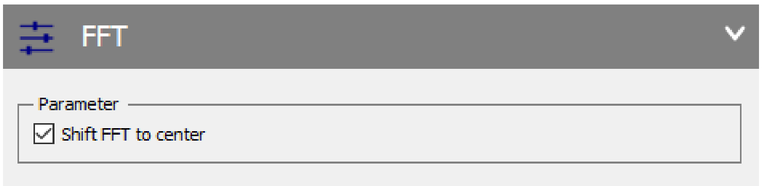
If **print segment data** is selected, then the Info window is filled with detailed information about the detected segments. In this case the "**Max. # segments**" can be used to limit the maximum number of segments outputted into the Info window. With the **Eliminate** option all small segments (**diameter lower than . . . pixel**) are removed from the result. Additionally, all regions with elliptical size can be removed (**aspect ratio**). The aspect ratio is the ratio between the major axis and the minor axis of a segment (Rotational invariant).

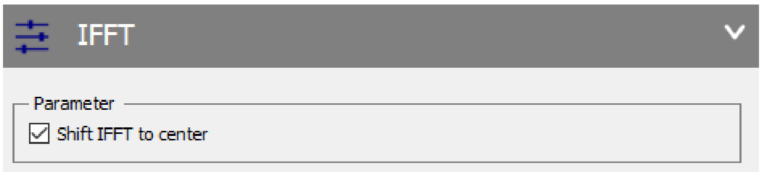
The result is given as **segmentation buffer**, which includes the marked areas and colored intensities for each area. Or the result is given by a **histogram** displaying e.g. the number of segments per diameter.



12.5.2 Fast Fourier Transform (FFT) and inverse FFT

Calculates a Fourier transformated image and converts it back to original image by shifting the FFT/IFFT to center. More details concerning FFT and IFFT are described in chapter **Fourier Transform (FFT)** on page 451.





12.5.3 Speckle Quality

Calculates scalar indicating relative pattern quality across the image. Uses local intensity gradients to access contrast.

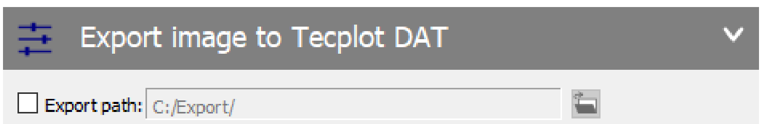
References

Pan, B., Lu, Z., & Xie, H. (2010) ”Mean intensity gradient: an effective global parameter for quality assessment of the speckle patterns used in digital image correlation.” Optics and Lasers in Engineering, 48(4), 469-477.

12.6 Operation: Export

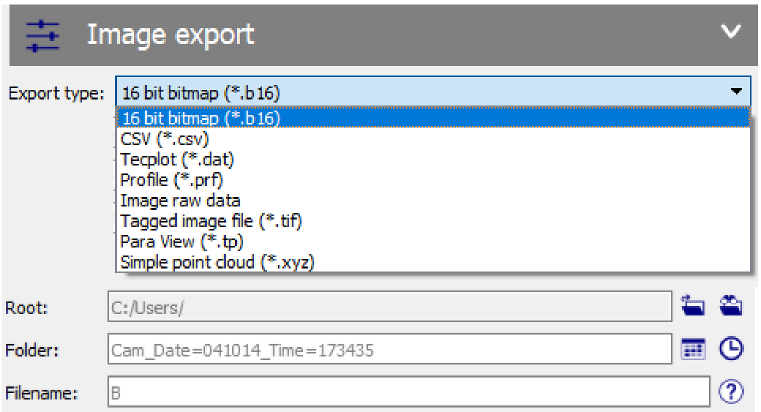
12.6.1 Export image to Tecplot DAT

Exports image in DAT format in Tecplot. Chose **Export path**.



12.6.2 Image Export

Exports pixel intensities in various file formats, such as PRF, TIF, BMP, TP, TXT, XYZ. Therefore, chose **Export type**. To select **Root**, click on either to select a directory, or on to open export directory. Moreover, select a **Folder** -, and a **File** name.

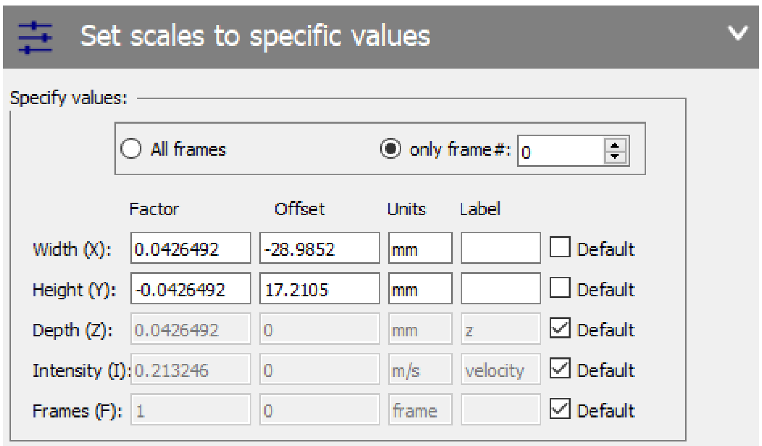


12.7 Operation: Scales

12.7.1 Set Scales

Change the scaling of the source buffers either to direct defined values or to values taken from another source.

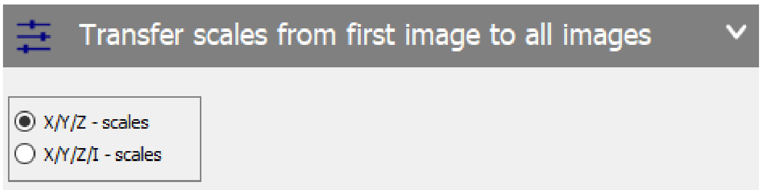
set scale to specific values: Change the image scales of all directions: X, Y, Z, I(intensity), F(frames):



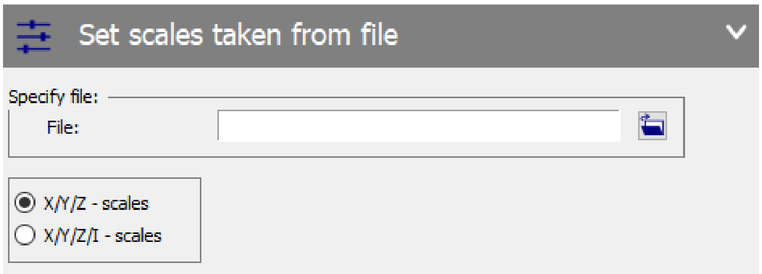
Choose between:

- **All frames**: Define the scales for X,Y,Z,.. in the lower part of the dialog. These scales will be set for all source frames.
- **only frame#**: Use the up/down button of the **frame#** item to toggle between all source frames. With this it is possible to set each frame scale separately.

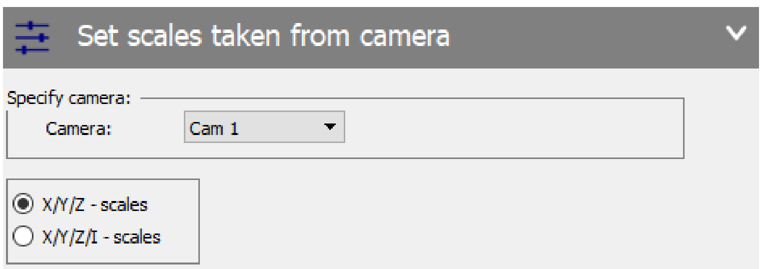
Transfer scales from first image to all images: Extract the scale parameters from the first source buffer (choose between X/Y/Z - or X/Y/Z/I - scales) and apply to all other image buffers.



Set scales taken from file: Extract the scale parameters from a file (choose between X/Y/Z - or X/Y/Z/I - scales) and apply this to all images buffers.



Set scales from camera: Select a camera as scaling source and choose between X/Y/Z - or X/Y/Z/I - scales.



12.7.2 Transfer Scales

Reset scales: Reset all scales to the default values with 1 as factor, 0 as offset, pixel or intensity as unit and a empty description in the X/Y/Z-direction and additionally in I-direction.

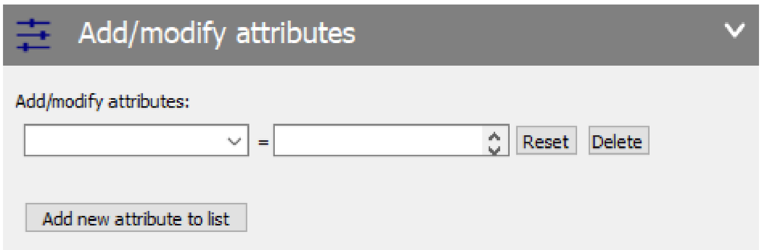
Adjust XY-scales to MITAS scan position: The image scales are adjusted by an offset in the X- and Y-direction that originates from a translation stage scan in the corresponding directions. Make sure that the images are scaled in mm to use this operation.

12.8 Operation: Attributes

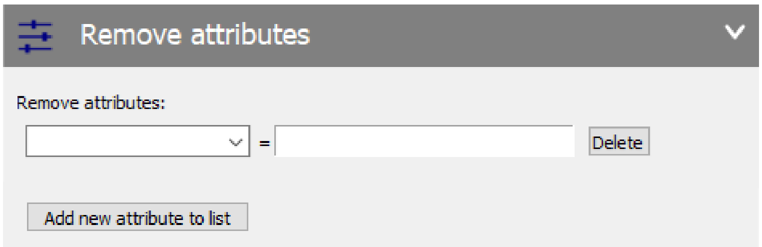
12.8.1 Basic Attributes

Create a list of attributes to be changed or removed. Press button **Add** on bottom of the parameter dialog to append a new item to the list. Press button **Delete** to remove the selected item from the list. For each item the attribute name can be selected and, when adding new attributes, the string value can be defined.

add/modify attributes: Create a new string attribute or modify the value of an existing attribute.



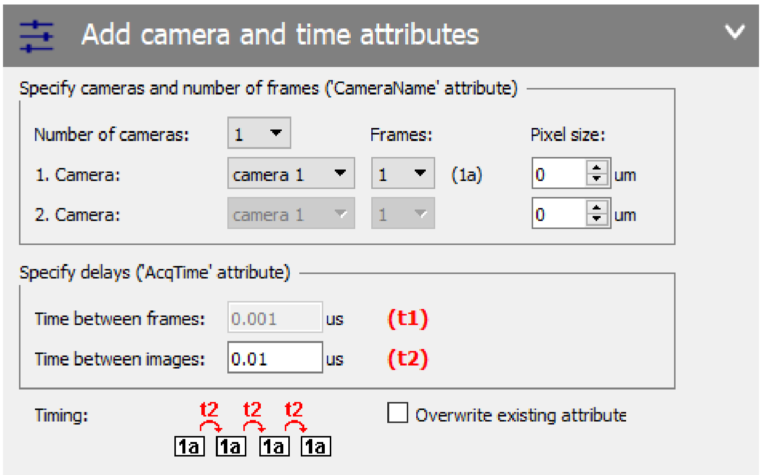
remove attributes: Delete the given attributes from the buffer.



12.8.2 Add camera and time attributes

This operation can be used to add or modify the following default attributes to the selected data set:

- CameraName#n: Camera number of frame *n*.
- AcqTime#n: Arrival time of frame *n* within one acquisition (1 image).
- AcqTimeSeries#n: Arrival time of image *n* within one recording (1 data set).
- Pixelsize#n: Camera pixel size in µm.

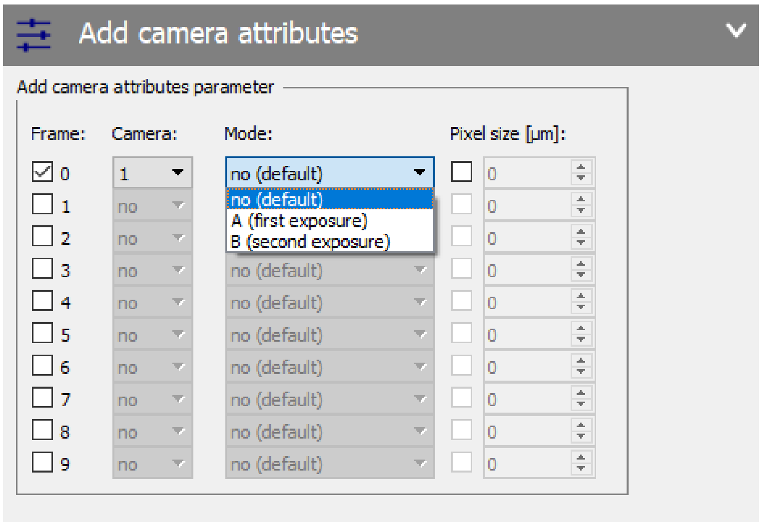


These attributes are very important for many operations in **DaVis** and must be added after importing files without these attributes (e.g. BMP and TIF). The `CameraName` attribute is used to identify the corresponding camera number (needed for calibration, image correction etc.). The `AcqTime` attributes are used to compute the time between two frames (vector computation, general time series operations). The `CamPixelSize` attribute is used in the calibration to get meaningful pinhole parameters. If this value is set to a wrong value, all pinhole parameters that are scaled in mm, such as focal length or camera-plate distances, will be wrong by the same factor. This will *not* affect the quality of the calibration! So if the camera pixel size is unknown, just enter 5 µm, which is a common value of CCD chips.

If the checkbox **Overwrite existing attributes** is not set, then existing attributes will remain and only the non-existing will be added. Otherwise all attributes will be overwritten or added.

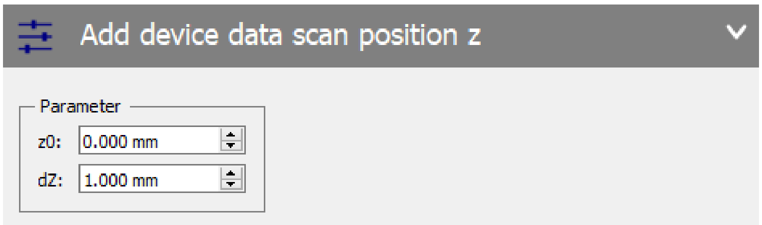
12.8.3 Add camera attributes

Select the frame(s) where to add a new `CameraName` attribute. For each frame the **camera** index can be selected. The **mode** of the camera is either **no** for single exposure images, or **A (first)** or **B (second)** for double exposure images. Optionally the pixel size of each camera can be given.



12.8.4 Add device data scan position z

Adds scan position information to existing data from older **DaVis** versions.



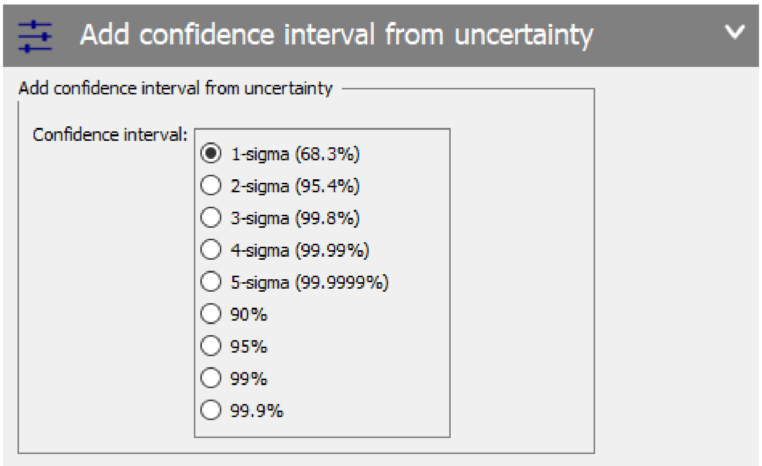
12.9 Operation: Statistics

12.9.1 Add confidence interval from uncertainty

This functions adds an extra scalar field "Confidence interval N%" if an uncertainty scalar field exists in the input vector field for $\mathbf{V_x}$, $\mathbf{V_y}$, $\mathbf{V_z}$ **or V** The confidence interval is defined as:

$$I_C = k \times \sigma \tag{12.1}$$

with σ the stored uncertainty and **k** a value dependent on the percentage value given for the confidence interval.



In the dialog the confidence interval can be selected either as a factor of sigma or as a percentage interval:

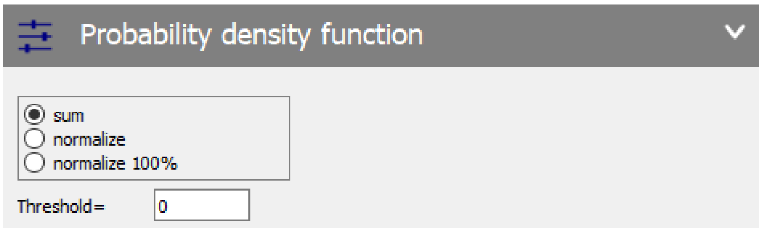
k	**Confidence interval**
1	68.3%
2	95.4%
3	99.8%
4	99.9%
5	99.9999%
1.645	90%
1.960	95%
2.576	99%
3.291	99.9%

The value of **k** corresponds to the width of a Gaussian bell that contains the given percentage of the total integral. E.g., a value of k = 1.96 contains 95% of all values around the center and leaves the 0.025% at either end outside.

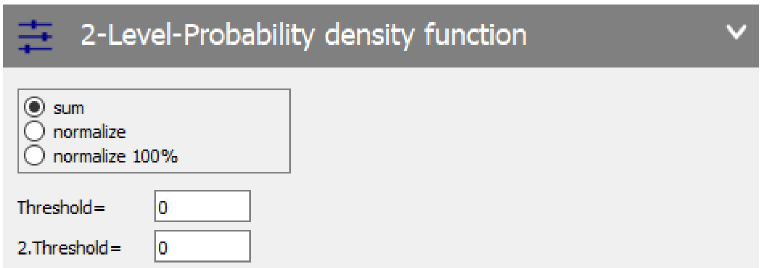
12.9.2 Probability Density Function

Simple Probability Density Function (PDF): This function uses a threshold to calculate the probability density function. All pixels above the threshold are set to 1, all pixels below are set to 0. The resulting image is the sum of the 0/1 images. Select

- normalize: to present the resulting image in the range 0...1.
- normalize 100%: for the range 0...100.

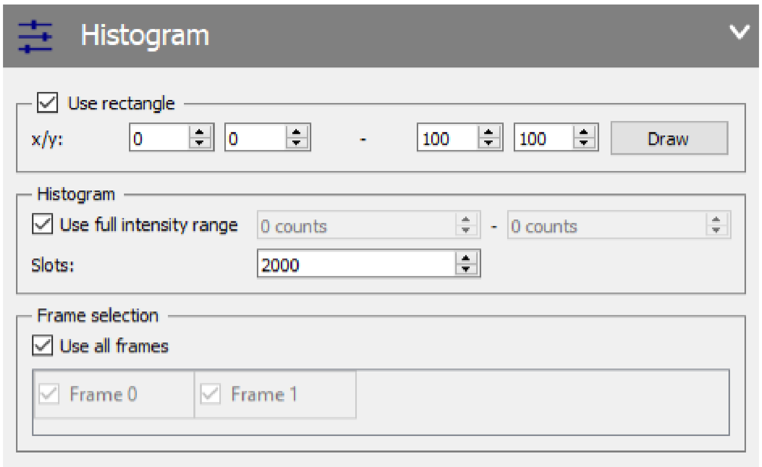


2-Level-Probability Density Function: This works like **PDF** but calculates two resulting buffers for different levels.



12.9.3 Histogram

Create a histogram of the intensities in the image and store result as profile. Each **slot** of the profile gives the number of pixels with the corresponding intensity. Optional a **rectangle** can be defined or selected with mouse clicks. The **intensity range** must be given with a minimum and maximum value. Leave this 0 to 0 for the complete range. The range is divided into the given number of **slots**. In the group box **Frame selection** checking the **Use alls frames** checkbox switches on to select all frames to create a histogram. Otherwise, select frames manually, by switching off the checkbox.



12.9.4 Sum, Average, Standard Deviation, Min, Max

Calculate the following results:

- average: of all source images.
- standard deviation: of the source images, see chapter on **Standard Deviation** on page 99 for the definition.
- min, max: for every pixel find the minimum or maximum intensity of all source images and store the min- or max-value in the resulting buffer.
- sum: add all source images.
- max-min: calculates the difference ("range") of both above described resulting images.

The mask is taken into account when adding up all image intensities and calculation all results. Masked pixels are not counted. If e.g. 10 images are processed and a pixel is masked out in 2 images, the average is calculated from the remaining 8 images.

The same processing can be done on device data, if existing. E.g. if a image file includes additional profile data from an A/D-converter, averaging may be done for the image data itself but also for the device profile.

Image statistics with uncertainty can compute the previously mentioned mathematical operations (see chapter 5.5.1) for scalar fields over time by taking the uncertainty of the scalar values into account. The formulas for using and computing the uncertainty is equivalent to the vector statistics uncertainty part and is described in chapter 13.6.

- If **Calculate uncertainty** is enabled, the result will contain an extra scalar field `Uncertainty` giving the error of the calculated values. For formulas used please refer to chapter 13.6: vector statistics.

- **Use effective number of samples (time-correlation)** calculates the effective number of data samples if the input data is highly correlated over time.

- Select **Subtract uncertainty from standard deviation** to subtract the estimated noise from the standard deviation due to input values containing uncertainty variations which should contribute to the true intensity variations:

$$\sigma_{true} = \sigma^2 - \overline{U^2} \tag{12.2}$$

  with U uncertainty, σ standard deviation.

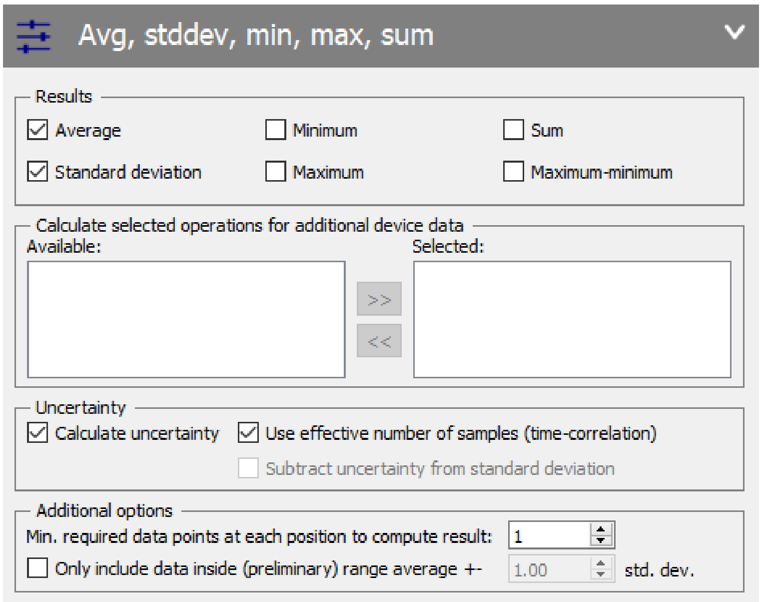
Additional options:

- **Min. required data points at each position to compute result**: allows to set the minimum number of samples necessary to calculate a mean / standard deviation value. If fewer samples are present for a pixel position, the result will be a disabled pixel.

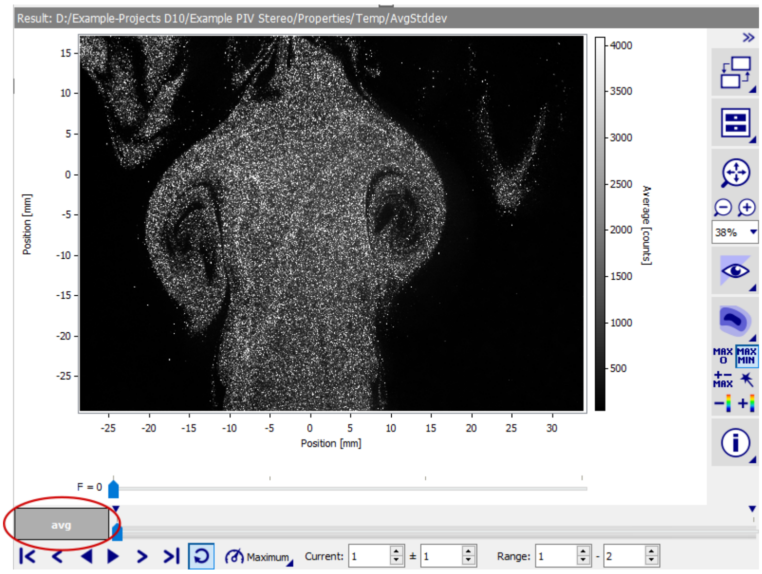
- **Only include data inside (preliminary) range average +- ... Std. dev:** This *sigma clipping* only uses values w with $\mu - p \cdot \sigma \leq w \leq \mu + p \cdot \sigma$ (μ average using all values, p specified value, σ standard deviation using all values). This way outliers are left out, but keep in mind that the calculation time will increase by a factor of two as two passes are made: one to calculate μ and σ and one to calculate the final result mean and standard deviation using only values w fulfilling the specified criterion.

- The final results will also contain a scalar field "Number of data points" to reflect how many samples were used to generate the final result at each position.

- If **Calculate uncertainty** was chosen, also additional scalar fields **Integral time scale** are stored, and **Neff** if **Use effective number of samples** had been activated.



The resulting set includes all selected statistics. The movie player gives information about the displayed data:



12.10 Operation: Mask

12.10.1 Concept of Masks

In **DaVis** every image, scalar field or vector field may have an additional masking information that is marking a pixel or a vector valid or masked out.

This information is used in the display and by many algorithms so that only valid pixels are used for further calculations. The statistics area in the **Buffer information - Data properties** dialog for example calculates the average / stdev / min / max values only from the visible pixels / vectors (that are not masked out). Please note that the information (pixel count values in images or shift values in vector fields) of masked out positions are not deleted, but they are simply not used. Deleting the mask can bring them back. Some processing operations let you choose, if you want to use all available information, or just the part that is marked as valid information (and not masked out).

In this section, processing operations are described that are used to add a mask, delete a mask, or delete the information in the masked out area permanently from the image or vector field.

Sometimes you cannot define a fixed mask for a series of images because the sample under study is moving or due to deformation or because the seeding of a flow has a fluctuating density and distribution. Rather than defining a individual masks for many images, the solution can be to define some criteria to distinguish areas of interest from areas that should not be considered in further calculations.

So, if you need to take into account a moving area of interest, you should adjust your experimental setup before the recording to facilitate this process. E.g. use a wavelength filter corresponding to your laser illumination on your camera lens, or just hang a black curtain in the background when using normal lighting to suppress background light facilitating the detection of the contour of your sample, or for increasing the visible difference between highly and poorly seeded areas.

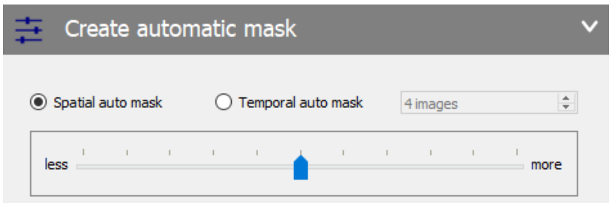
12.10.2 Create automatic mask

In the automatic mask mode a *copy* of the image is taken to run algorithms using the local pixel intensity information to decide if the mask for that pixel is enabled or disabled. The resulting mask is applied to the original image without changing the intensity values in the result.

As automatic mask either a spatial or temporal automatic mask can be calculated. For the temporal mask, also the number of images taken into account in the calculation is specified.

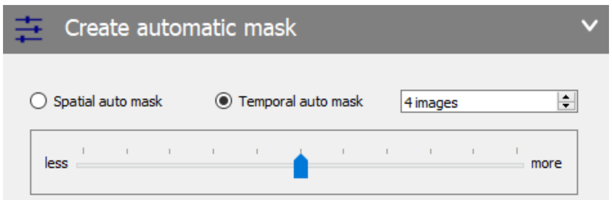
Spatial auto mask

The aim of a spatial auto mask is to distinguish between the area in the image with seeding (high pixel count values) and without seeding (low pixel count values). As between single particles there might be low count pixels as well, first a sliding maximum filter. This will effectively fill in all low count areas with high values **if** high pixel count seeding particles are in the neighborhood! Areas where no high pixel count seeding is nearby maintain their small intensity values.



Temporal auto mask

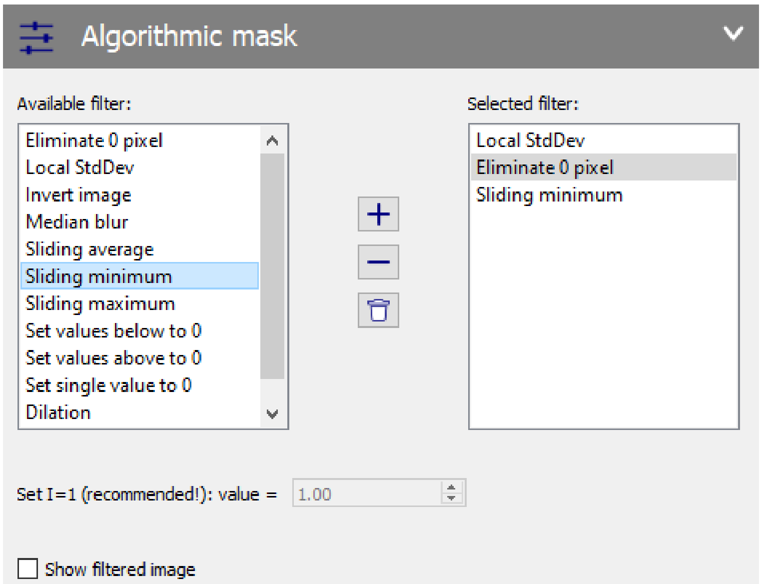
The temporal auto mask takes the ±4 images into account to reduce the influence of temporal fluctuations in the recording series.



12.10.3 Add Algorithmic Mask

In the algorithmic mask mode a *copy* of the image is taken to run some algorithms using the local pixel intensity information to decide if the mask for that pixel is enabled or disabled. The resulting mask is applied to the original image without changing the intensity values in the result.

To create a mask in **algorithmic** mode a number of different operations can be selected in the list on the right.



- **Eliminate 0 pixel**: If selected, all zero count pixels in the original image will be replaced and will hold the given count value. This may be required if a previously masked image contains many zero count pixels that disturb subsequent algorithms. Here, these can be given a count value close to the average of the rest of the image.

- **Invert image** and add an intensity offset **N** using the formula $I(x,y) = I_{max} - I(x,y) + N$.

- **Sliding average/maximum/minimum** filter over $N \times N$ pixel.

- **Median blur** replaces the current value with the median value of its neighbors defined by the median blur length.

- **Set values below/above to 0** sets the intensity above or below a given threshold to 0 counts.

- **Set single value to 0** sets all pixels of the specified value to 0. This options enables using masks with a mask out value different from 0. Note that sometimes it is easier to define the areas which must **not** be used e.g. for vector calculation.

- **Erosion:** Just like the median filter this filter will sort the pixels of a $N \times N$ pixel-area according to their intensity. But instead of taking the middle element, the erosion will take the element before. The effect is a kind of erosion, valleys become deeper and finally the image will be eroded away toward the background level.

- **Dilate:** The dilatation filter does exactly the opposite of the erosion. It will take the next element after the middle one in the sorting list. The effect is that edges become sharper while plateaus flatten and become more even.

- **Local Stdev:** Any pixel value is replaced with the local standard deviation to the average of its neighbors. This is a useful filter operation to mask out areas with high particle seeding for example. Image areas without or with poor particle seeding show a homogeneous (low) count distribution, whereas areas with many particles have high local contrast (count value variations). The same applies to probes with speckle pattern.

Enable the checkbox **Show filtered image** in order to show the image with applied filters. If this checkbox is disabled, the masked image is shown.

A typical scenario is using a intensity threshold to distinguish between area in the image with seeding (high pixel count values) and without seeding (low pixel count values). As between single particles there might be low count pixels as well it is recommended first to apply a sliding maximum filter over a certain distance N (typically the mean distance between seeding particles). This will put the values to the highest value in a $\pm N$ neighborhood in x and y direction. This will effectively fill in all low count areas with high values **if** high pixel count seeding particles are in the neighborhood! Areas where no high pixel count seeding is nearby maintain their small intensity values. See fig. 12.1 for an example with the operation **below threshold** of 400 counts alone and a combination of **Sliding maximum** over 10 pixels and **below threshold** of 400 counts.

If the average intensity of the seeded area differs very little from the background intensity with no seeding it is recommended to calculate the **local standard deviation (stdev)** value first which is the intensity variation which gives high values for seeded areas and low values for non seeded areas. Typical values are stdev=10 for highly seeded regions and stdev=0 for poorly seeded regions. Starting from this you can apply the above **Sliding maximum** and **below threshold** functions.

As the **Sliding maximum** filter tends to enhance the region of interest to non seeded areas you can erode the region using the **erosion** filter as last function. This effectively eats the outer N pixels of the mask away like peeling several layers from an onion.

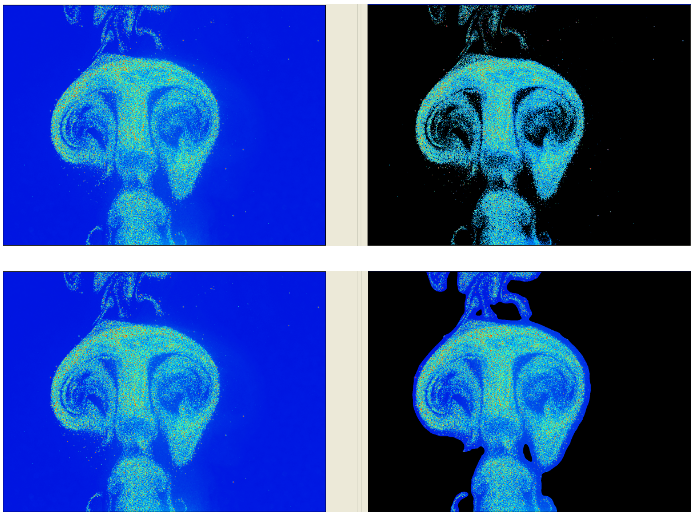
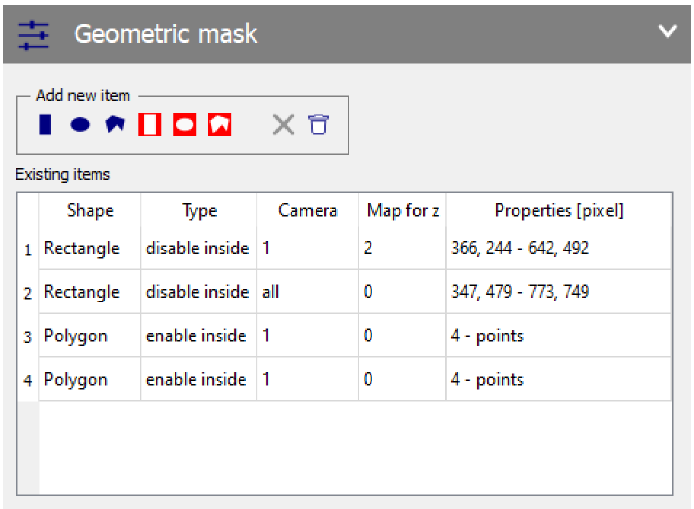


Figure 12.1: Example: original image (left), image masked with **below threshold** (upper right) or additionally with **sliding maximum** (lower right)

12.10.4 Add Geometric Mask

This is the standard mask definition mode suitable for fixed geometrical shapes i.e. filled rectangles, circles, polygons, or ellipses for all frames of an individual camera or for all camera frames in the same coordinate system (mapped calibration).



	Shape	Type	Camera	Map for z	Properties [pixel]
1	Rectangle	disable inside	1	2	366, 244 - 642, 492
2	Rectangle	disable inside	all	0	347, 479 - 773, 749
3	Polygon	enable inside	1	0	4 - points
4	Polygon	enable inside	1	0	4 - points

Figure 12.2: Operation Geometric Mask.

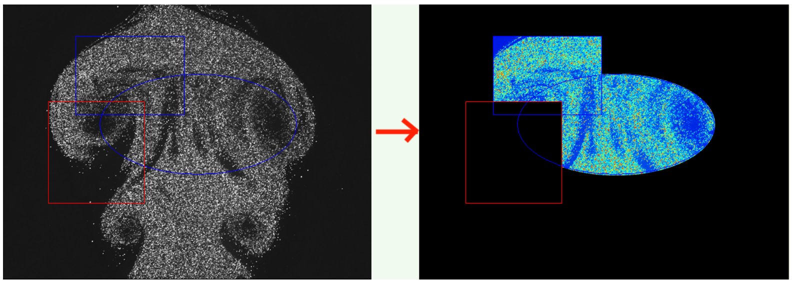
Typically, this is required in the case of limited optical access. As a result, the image has certain fixed areas with no useful information. Another typical reason is that you want to take out areas with reflections and other artifacts that stay at a fixed position and would negatively influence in the derived result of an algorithm (e.g. would appear as ghost particles).

The shape of a masks can be defined in **Add new item**.

You define a region by clicking on one of the following 6 buttons, and then select the positions on the source image in the batch processing with the mouse. The blue buttons enable all pixels inside the defined element and masking out all pixels at the outside. The red buttons enable all pixels outside the defined element and masking out all pixels inside.

- The first mouse click defines a corner of the rectangle, the second mouse click defines the opposite corner. Here it is not important which corner you click first. You might start with the upper left, followed by the lower right, or vice versa.
- The elliptic mask is defined analogously to the rectangular mask.
- For a polygon any number of points can be added. Clicking on the first point closes the polygon and stops adding further points. "Esc" quits the entry and discards the polygon.

If you define more than one element, then the intersection of all elements is calculated the following way: first the single blue enabled areas are added up (if a pixel is inside of any one of the elements then it is enabled). Independently the single red enabled areas are added up (if a pixel is inside of any one of elements then it is disabled). Then the intersection of those two areas is calculated (if a pixel is enabled in the added red *AND* enabled in the added blue areas then it is enabled in the final result). Please refer to the following figure for an example:

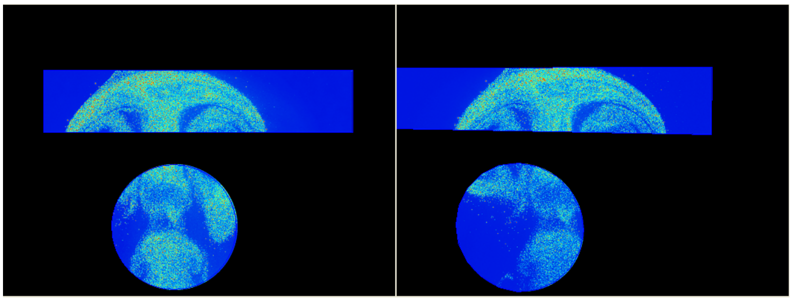


All defined elements appear in the **Existing items** table.

Type of the item shape can be changed by switching from **enable inside** to **disable inside** and vice versa.

Additionally, you can select in the **Camera** column of every defined item, whether the mask should be applied to all frames of **one camera** - the camera corresponding to currently visible frame, or to **all cameras** - that is to all frames of all cameras *in the same coordinate system*. **1** applies the certain mask only to that camera where the mask was defined, whereas **All** applies the certain mask to all cameras.

In the latter case, the mask will be mapped accordingly to take into account the image distortion. In order to use this function a proper calibration for mapped cameras is required (e.g. as necessary in a stereo system). Further, a z-position has to be given and can be defined in the **Map for z** column. This is used to map the masked camera raw image to world coordinates first and then determine which pixels need to be disabled in the other camera raw images. This is also the reason why a rectangle / circle in the original camera most likely looks deformed in related camera images. Usually, you would specify $z = 0\,\text{mm}$, which is at the position of the calibration plate in the calibration process.



12.10.5 Load mask from file

Masks can also be predefined in files.

You may have a different source to calculate the masked areas. Using this mode you can load an image (or a set of images) holding intensity values equal zero (disabled pixels) and pixel count values greater than zero (enables pixels). This intensity information is used to define a mask for the source camera images. 12.3 depicts the corresponding dialog.

An external set can be loaded via choosing **Use local subset MASK.SET**. Taking into account this option, the referred set is automatically accessed. Alternatively, a **mask from a specified set** can be utilized. Here, the exact path to the external set has to be defined. In case the file path is not valid, a warning appears. It is important to have the same amount of

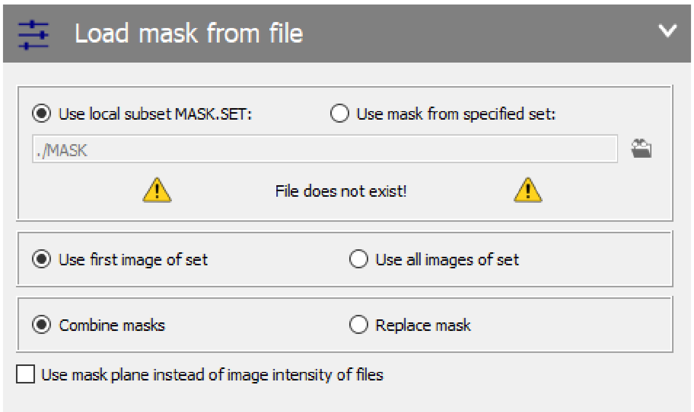


Figure 12.3: Operation Load mask from file.

images per buffer in both, MASK.SET and the input set. The number of images in the input set and in the external set has to be equal, too.

Further, in the second selection, the images of the set to be used can be determined. Choose between **Use first image of set** and **Use all images of set**. If the first option is selected, only the frames of the first buffer in the external set are utilized as a mask for each buffer in the input set. In this case, one buffer in the external set is sufficient. Contrarily, when using all images of the external set, the same amount of buffers as in the input set is required, since each buffer of the external set is assigned to the according buffer in the input set.

The options **Combine masks** and **Replace masks** define how previous masks in the input set are dealt with. Additional masks from the external set are included to the input set, if combine masks was selected. Otherwise, when replace masks was chosen, all masks of the input set are replaced with masks from the external set.

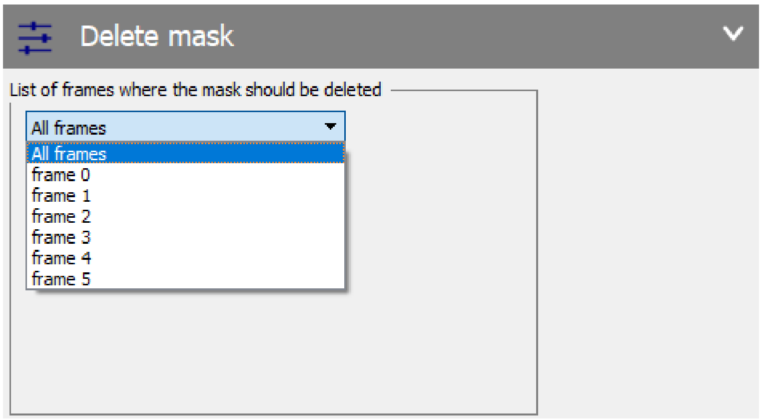
If **Use mask plane instead of image intensity of files** is enabled, mask values of the external set are utilized in place of images intensities.

12.10.6 Delete Mask

There might be a successfully defined mask, that was used by an operation but is no longer needed by subsequent operations in the operation pipeline. Then this mask can be deleted. In this case the original intensity values of all previously disabled pixels are restored. You have to be careful if you use this function as several functions define a mask as result of the operation.

For example the *Rotation* or the *image correction* processing function will add a mask to the result to mark tag pixels that are not valid (pixels at the image border after a 45^o rotation where the image was turned away to leave an empty area). Removing the mask will leave these pixels as valid with a 0 *count* value which might influence on further calculations. So be careful and check if taking away the mask from your image does not have side effects.

In the stand-alone function you can select to remove the masks on all frames or to remove the mask on a single frame only.

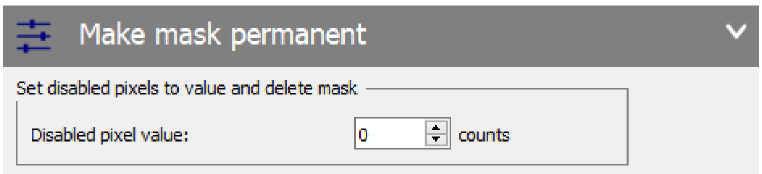


12.10.7 Extract mask as binary image

Store masks from source set files as new image set. A pixel will be set to value 1, if the mask position is selected, or 0, if disabled.

12.10.8 Make mask permanent

You might have successfully defined a mask, but the operation you want to use does not respect the masked areas or it does not the way you wish. Then you can make the mask permanent by this function. It deletes the mask and instead replaces the intensity count values of all previously disabled pixels by a **disabled pixel value**.



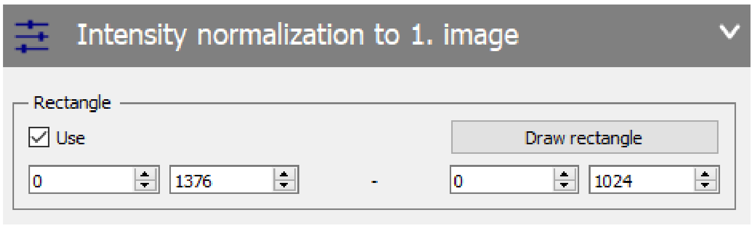
12.11 Operation: Time Series

12.11.1 Difference to First Image

Subtract the first image from every source and add an intensity offset, if needed.

12.11.2 Intensity normalization to First Image

Normalize the intensity level of all source images inside a rectangular region to the first image. You can either draw manually a rectangle in the source image by clicking **Draw rectangle** or edit the coordinates directly in the dialog by their unscaled pixel position.



12.11.3 Shift and Rotation Correction

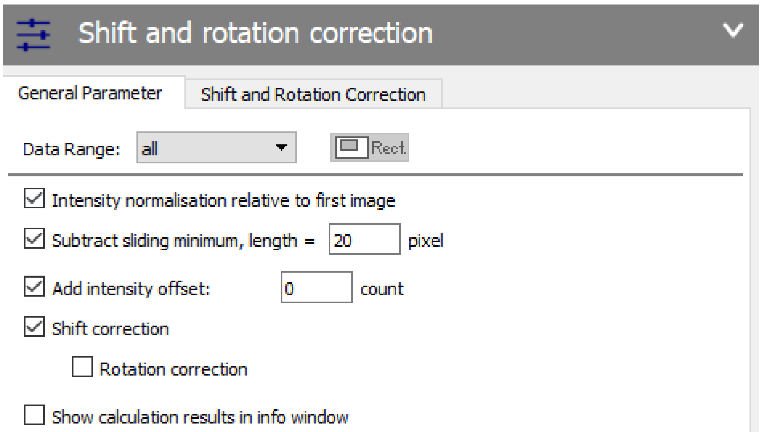
Shift and intensity correction will adjust all images to the image intensity and position of the first image of the set. Thus, non-uniform images can be adapted to match the first one in brightness (e.g. due to changing background) and/or position. A shift correction might be necessary, if your object or the cameras moved during the image recording (e.g. due to vibrations), or if your images were derived from scanned photographs that are not aligned very well from one image to the other. To match the position select one or more objects that are resting in the frame of reference (for example a part of the objects housing) and all images will be shifted (and rotated) to match that position. Images with non-uniform lighting or background noise can be adapted to reach the same **mean intensity** of the first image.

In the top of the first parameter card (General Parameter), the **Data range** is defined. Here, select a rectangular region of interest that will be cut out first before proceeding on the whole image. All the data outside will be left out in the result. If you have defined an image correction for your camera be aware that cutting out data in that way will make your image correction invalid, as now the pixel (0/0) correspond to a wrong position of your calibration plate. So either cut out the cross plate pictures as well using the same region and redo the calibration process, or correct the images first and performing the shift and intensity correction afterwards. Then of course the evaluation should not use image correction again as the images are already corrected!

The available operations are carried out in succession and add their affects accumulatively.

Intensity normalization relative to first image will adapt all images to the mean brightness of the first image. This might be adequate for image series with varying background intensities or non constant illumination of the object. **Subtract sliding** minimum is a high pass filter that filters out the local mean background intensity (over a scale of the specified number of pixels) leaving only the local fluctuations. To avoid zero clipping when subtracting the local mean value you can specify a number of counts in **Add intensity offset** that will be added to the image before subtracting the local mean value of the original image.

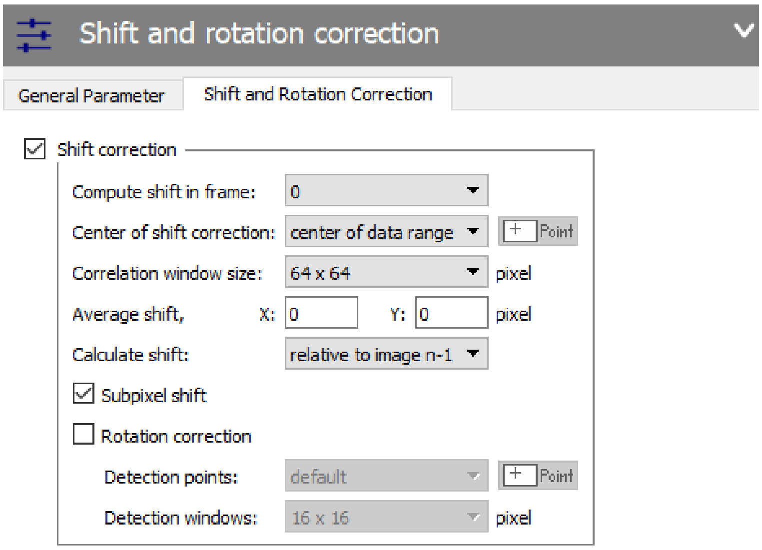
Enabling the group box **Shift correction** enables the translation correction (and rotation correction, if **Rotation correction** is specified) of all images relative to the first image. The parameters for shift and rotation correction are defined on the second parameter card (Shift and Rotation Correction).



As **center of shift correction** you can either leave the default **center of data range** or specify any other point that lies on a static structure (e.g. housing of the object) that is fixed with your frame of reference and is not affected by the studied structures. To specify another, more suitable point of reference for shift detection press the button **Point** to the right of the **shift correction** check box.

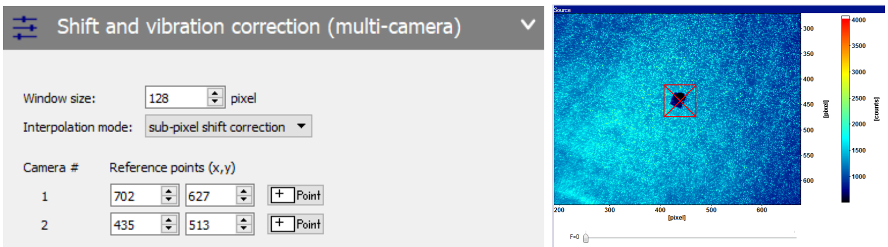
In **Correlation window size** you specify the correlation window. This window is shown as a red box in your image. Please select its size so that the static structure used for the image matching lies well inside, but (almost) no contrast is included, as this random information disturbs the correlation. Larger windows usually lead to more stable results. In any case the window size should be two times greater than the maximum shift among images, or the algorithm will fail. If this cannot be achieved for the current static structure look for a more suitable one. If you already know that a linear movement is taking place in your image series (linear trajectory of your static structure through the images) you might specify an **Average shift in x and y**. This will be applied to every image before starting the correlation. So the correlation window size can be kept small while still allowing to detect large shifts. The **Calculate shift** option defines how the algorithm works: The **relative to first image** mode uses the first image to calculate the shift. This mode is very exact but can only be used for small absolute shifts within one set. The **relative to image n-1** mode calculates the shift relative to the last image and adds the absolute shift from the last. This can be used for larger shifts but is less exact because a small error will be added from image to image. Enable the **Subpixel shift** option to correct shifts smaller than one pixel. The **shift correction** will only perform a translational correction and does not detect rotations!

To eliminate all remaining rotation around the position specified for the shift correction (these do match already after the shift correction), you might add a **rotational correction**. In this second step several areas marked by the d**etection points** are matched and the best rotation angle is computed. The default position of this detection points are around the specified static structure. If there are more than one static structures visible in the image, it is recommended to **specify** your own **detection points**. Pressing the button **Point** you can select up to 20 reference points for the rotational correction. The farther away from the static structure used in shift correction the greater the level for the rotational correction and the better the estimate for the optimal angle. Again, the **detection window** should be adapted to the objects used for **detection points**, and as large as possible for the same reasons as described above.



12.11.4 Shift/Vibration Correction (Multi-Camera)

This processing is especially designed to take out shifts from all images of a Stereo-PIV system independently for each camera. This could be necessary if e.g. vibrations caused the cameras to wiggle during the recording. Please note that this processing should be used with care as it has some limitations.



Similar to the shift and rotation correction (see above), for each camera a fixed reference point in the first image needs to be set that is visible in all images and serves as the matching pattern to shift all images to the position of the first image. A good choice would be a part of your model with a unique feature (e.g. a screw). A straight line (e.g. a reflection on an edge of your model) is a poor choice as the matching is ambiguous along the direction of the line. A curved line is better suited. In Stereo-PIV with two (or more) cameras, any vibration between the cameras leads to 2D2C-vector positions for each camera, which do not correspond any more to the same position in the light sheet. While a constant misalignment can be corrected for by the usual Stereo-PIV self-calibration procedure (REF), for vibrations any displacement varying from image to image needs to be corrected first by the function here.

Since the vibration of the cameras is usually present in all images of the recording including the first image, the shift correction will shift all images to the shifted position of the first image and a new Stereo-PIV self-calibration is required to correct the calibration function accordingly.

For double-frame camera images both frames are shifted by the same shift rounded to the next integer to avoid interpolation artifacts. For time series of single-frame camera images rounding the shift to integer values is not optimal because any residual fractional shift in one image will lead to a bias shift when calculating the vector field relative to the following image. In that case a sub-pixel shift and a sub-pixel interpolation are used.

The routine works for any number of cameras.

12.11.5 Time Filter

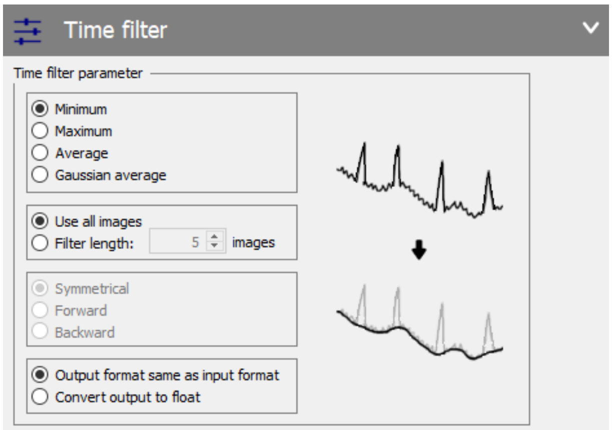


Figure 12.4: Shift and Rotation Correction parameter

For each image calculate the **minimum, maximum, average** or **Gaussian average** intensity for each pixel position from n (**filter length**) consecutive images. Output has same number of buffer. Or calculate the **minimum, maximum or average** intensity for each pixel position from the complete source set (**use all images**). Output is one buffer. If the **Filter length** n is used, then the mode **symmetrical, forward** or **backward** can be set. An example with n = 3 is shown in figures 12.5, 12.6 and 12.7:

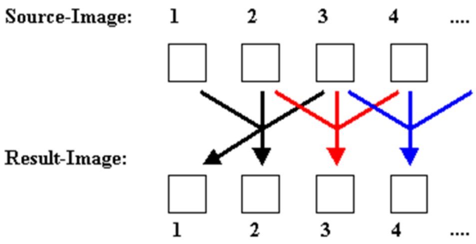


Figure 12.5: Symmetrical

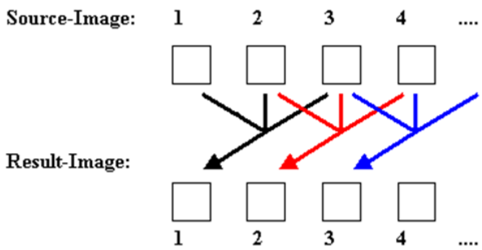


Figure 12.6: Forward (last n images get same information)

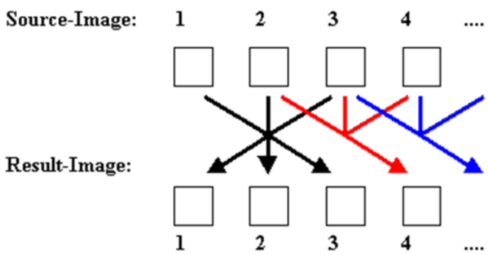


Figure 12.7: Backward (first n images get same information)

Convert output to float will force the output format to be float, else the **output format is the same as the input format**. This may proof useful in the case of the algorithm producing negative values, which are clipped in the case of WORD output.

12.11.6 Subtract Time Filter

Eliminate background noise (bright particles on dark background). Calculate the **minimum**, **average** or **Gaussian average** intensity like in **time filter**. Then subtract this from the source.

Here, also the **Butterworth** filter can be used to eliminate the background noise. This method can be used to remove unsteady laser light reflections from images of high-speed PIV systems. The movement of the reflection

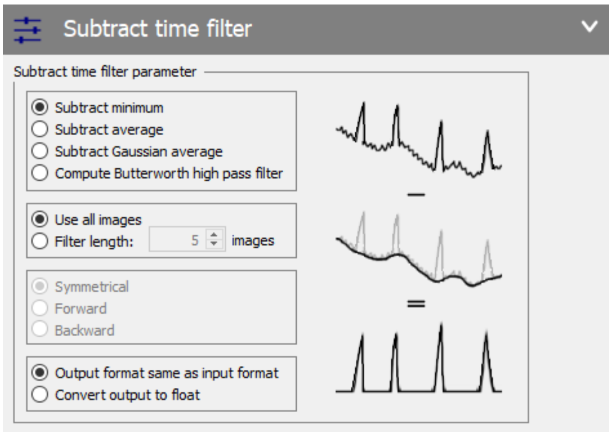


Figure 12.8: Subtract time filter parameter

can either be caused by movement of the reflecting object itself or by movement of the whole imaging system due to vibration. In both cases, traditional methods such as subtracting the **minimum** or **average** intensity observed over time yield only good results in areas where the reflection intensity is constant over time. But since the laser light reflections typically move magnitudes slower than the particles, it is possible to separate both signals in the frequency domain. By applying a suitable high-pass filter, only the low frequencies of the slow moving reflections are attenuated, resulting in an elimination of them while the intensities of the particles remain unaffected. In this case, a Butterworth filter can be chosen for having a maximum flatness in its passband.

The cutoff frequency is the frequency where the passband passes into the stopband, above it, frequencies are unaffected, below it, they are filtered. The cutoff frequency is specified as a fraction of π. With the data sample frequency S, the cutoff frequency F can be expressed as a fraction of π with the equation

$$2\frac{F}{S} = \pi.$$

With the **filter length** L, the equation can be transformed to

$$\frac{2}{L} = \pi.$$

The slope of the filter is determined by the filter order. The higher the order, the stronger lower frequencies are attenuated and the sharper the cutoff at the transition from pass- to stopband is. An order of three is sufficient for most data sets and is set to the default value.

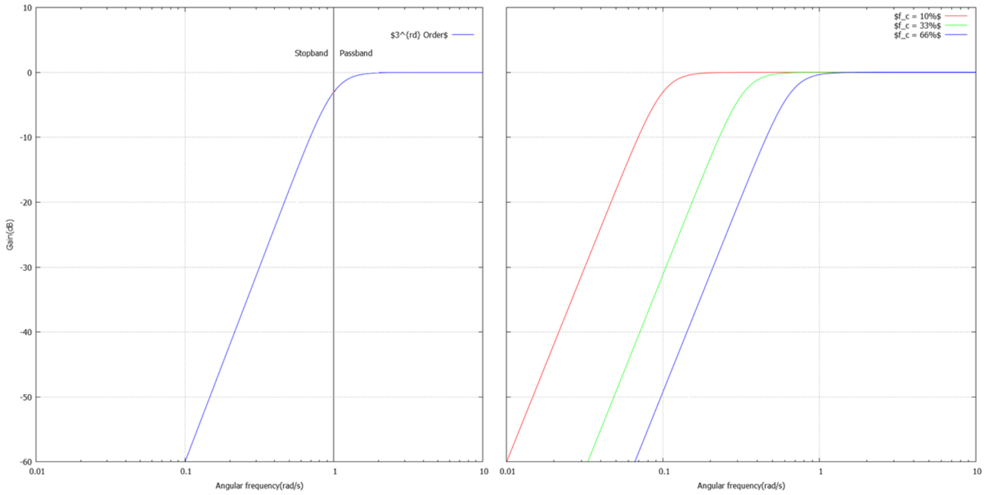
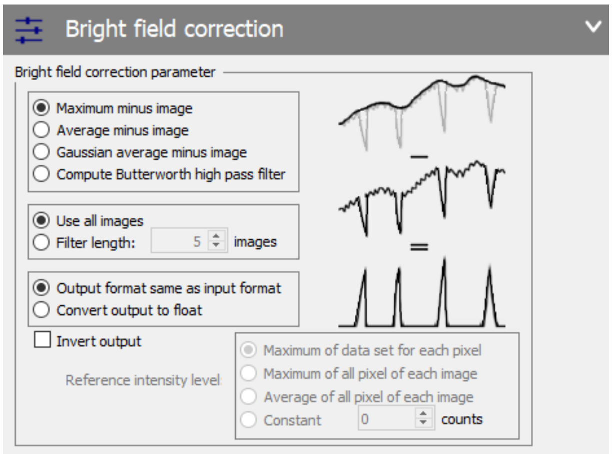


Figure 12.9: left: stoppband and passband; right: variation in cutoff frequency

Note that two prerequisites must be met for this method to produce viable results: The particles should move at least three times faster than the reflections and, depending on the speed difference, the time series must consist of a certain number of frames. The transformation to frequency domain produces half as much discernible frequency bins as there are frames, so with a frame rate of n Hz, n frames would yield the maximum resolution in frequency domain. While this being the best case, half of this optimal number of frames should be more than enough in most cases. For a more in-depth explanation of this method, please refer to: Scarano, F. and Sciacchitano, A. (2011) Robust elimination of light reflections in PIV. 9th International Symposium on Particle Image Velocimetry - PIV'11, 21-23 Jul 2011, Kobe, Japan

12.11.7 Bright-Field Correction



Compare with **subtract time filter**. This method subtracts the source from the **maximum**, **average** or **Gaussian average** intensity (over time). **Invert output** will invert the output. Therefore the **maximum of the dataset**, the **maximum of each image**, the **average of each image** or a **constant** value is used.

12.12 Operation: Filter

Filter operations play an important role in image processing. They are used for smoothing, edge detection, texture or contrast enhancement, and many other special tasks. There are numbers of filters, especially the NxN-filters, which only operate on the close neighborhood of a pixel. On the other hand, the filter operations done using the Fourier transform are nonlocalized and operate on frequencies, i.e. structures like sharp edges are analysed in the realm of their frequency representation.

The FFT-filter are described in an extra chapter. In this chapter the linear and non-linear neighborhood filters are described.

Among them the most often used ones are the ones, which calculate the new value of a pixel from a linear combination of the surrounding 8 pixel values and the value of the center pixel itself. Mathematically this can be represented as a 3x3-matrix, which is sort of moved over the image to calculate a new center pixel each time it is applied to a 3x3 area:

$$f'_{i,j} = \begin{pmatrix} a_1 f_{i-1,j-1} & a_2 f_{i,j-1} & a_3 f_{i+1,j-1} \\ b_1 f_{i-1,j} & b_2 f_{i,j} & b_3 f_{i+1,j} \\ c_1 f_{i-1,j+1} & c_2 f_{i,j+1} & c_3 f_{i+1,j+1} \end{pmatrix} / divisor$$

where

$$\begin{pmatrix} a1 & a2 & a3 \\ b1 & b2 & b3 \\ c1 & c2 & c3 \end{pmatrix} / divisor$$

is the filter matrix. 'divisor' is the final division factor of the sum. Only if 'divisor' is the sum over the filter matrix elements, the average image intensity stays the same. For example, if all of the filter elements are 1 (=smoothing) then 'divisor' is set to 9 in order not to change the average image intensity.

12.12.1 Smoothing

Filter the source by using the Smoothing Filter. Smoothing filters are used to eliminate high frequency noise by taking an average over the neighborhood of a pixel.

Smoothing 3x3

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} /9$$

This is the simplest smoothing filter and it works well. Function **smoothing k x k** is the same matrix, but using up to 9x9 pixels. Of course, any smoothing filter will also smooth-out any sharp edges, which are not noise, but physically real.

Gaussian LowPass

This is another good smoothing filter. It takes the center pixel more into account than the simple smoothing-3x3-filter. Mathematically it is more like a gaussian curve. Viewed in the frequency realm, the gaussian lowpass filter is smoother than the simple 1-1-1-filter.

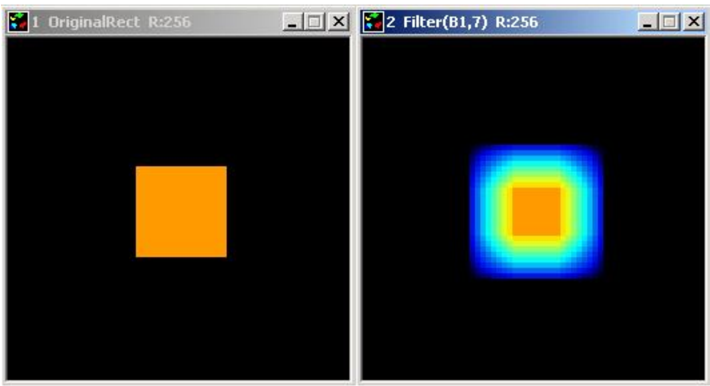
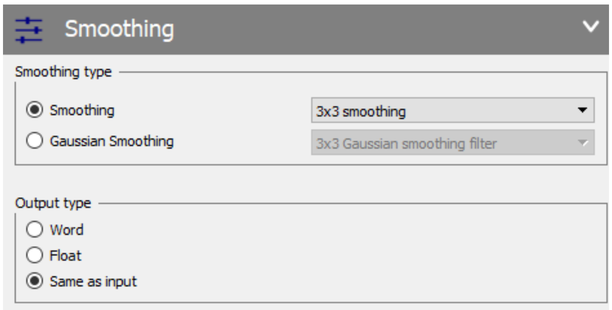


Figure 12.10: Original image (left) and 9x9-smoothed image (right).

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} /16$$

Choose a **Smoothing type** and the corresponding **Filter matrix** (see page 475) from the list. In the dialog the **Output type** can be selected. The result can be stored either in a word or a float buffer, or in the same type as the input.

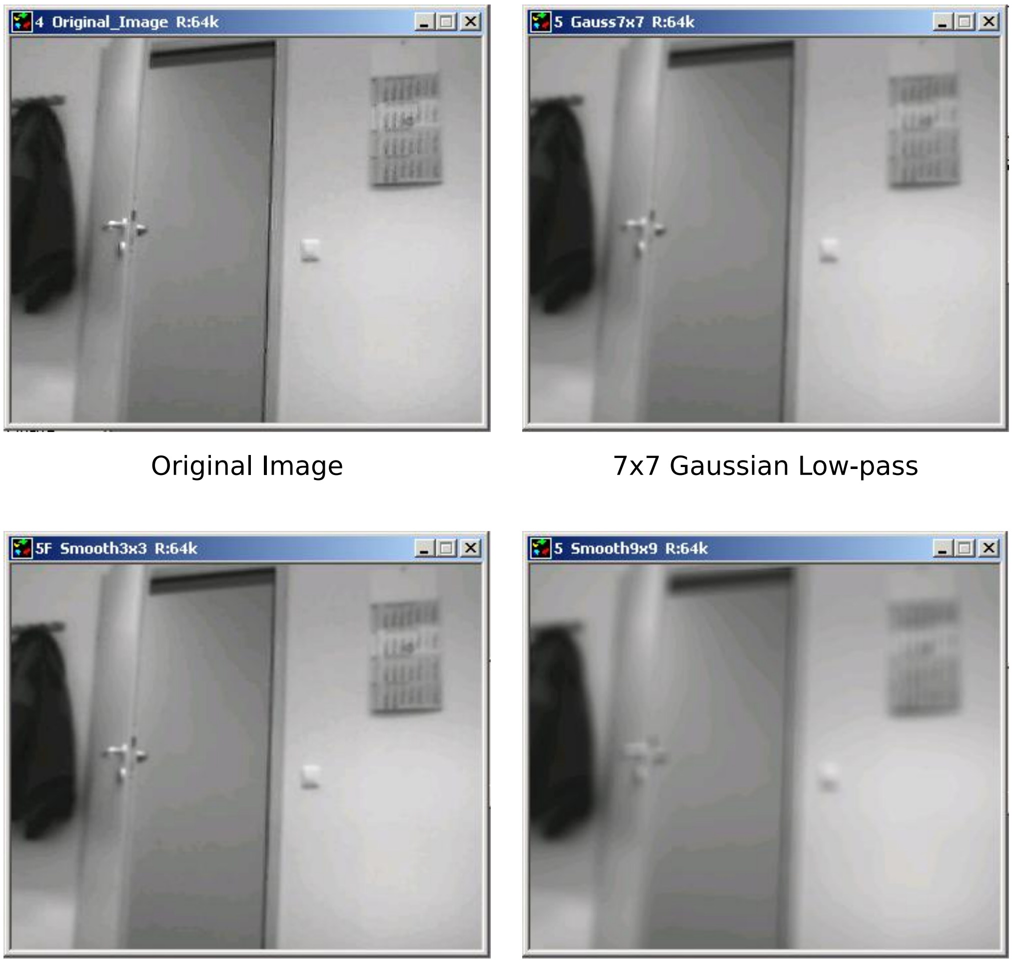


12.12.2 Linear Filter

Filter the source by using one of the Linear Filters **Sobel**, **Compass**, **Laplace**, or **Sharpening filter**.

Sobel Filter

The Sobel filter will enhance edges. There are four different ones, each one preferring a different direction. The effect is one of introducing a pseudo-3D representation of the data.



Original Image | 7x7 Gaussian Low-pass

Smoothing 3 x 3 | Smoothing 9 x 9

Figure 12.11: Gaussian and Smoothing filters on an example image.

All of the Sobel filters do not use any divisor (=1). The pixels are simply multiplied with the filter matrix and summed up.

- Sobel 1 – vertical: $\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}$
- Sobel 2 – horizontal: $\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$
- Sobel 3 – diagonal: $\begin{pmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{pmatrix}$

- Sobel 4 – diagonal $\begin{pmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix}$

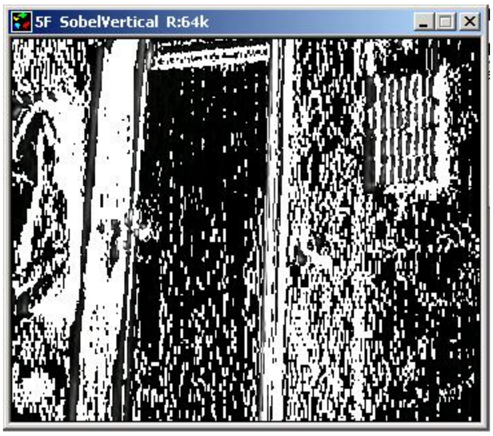


Figure 12.12: Example for Sobel-1 vertical filter.

Compass Filter

The compass filters are also edge detection filters. They are named according to their preferred direction N, NE, E, ...

- Compass-N: $\begin{pmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{pmatrix}$
- Compass-NE: $\begin{pmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{pmatrix}$
- Compass-E: $\begin{pmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{pmatrix}$
- and so on...

Laplace Filter

Laplace filter will detect edges in any direction. Simply try out the three different laplacian filter. Laplace-3 is a laplacian of a gaussian filter, i.e. it incorporates a second derivative to provide a noise-independent edge detection filter.

Just like the sobel filter the divisor is 1.

- Laplace-1: $\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$
- Laplace-2: $\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$
- Laplace-3: $\begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}$

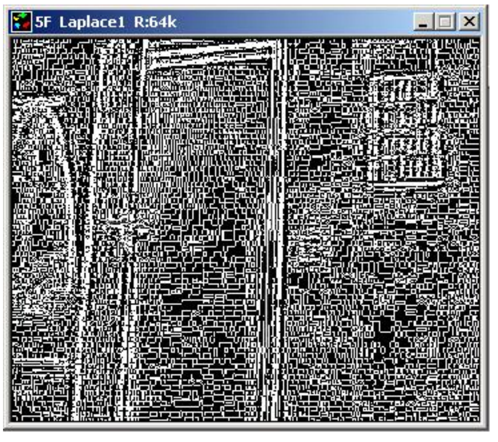


Figure 12.13: Example of the Laplace filter.

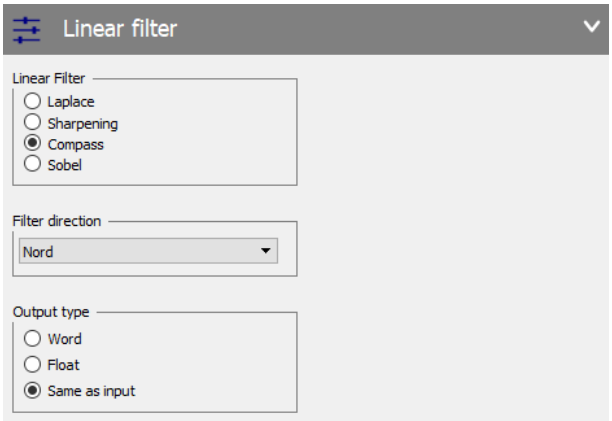
Sharpening Filter

These filters enhance the contrast of the image, i.e. the image is sharpened. Mathematically this is done by adding the original image to the laplacian. The factor **a** determines the addition of the edge-enhancing second derivative to the image. For a = 1 this is a 50% addition.

- Sharpness-1: $\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4+a & -1 \\ 0 & -1 & 0 \end{pmatrix} / a$
- Sharpness-2: $\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8+a & -1 \\ -1 & -1 & -1 \end{pmatrix} / a$

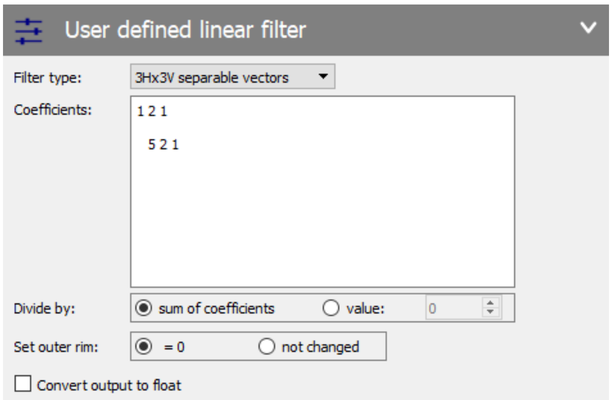
Figure 12.14: Example of Sharpness filter.

Select a **Linear filter** and the corresponding **Filter direction** from the list. In the dialog the **Output type** can be selected. The result can be stored either in a word or a float buffer, or in the same type as the input.



12.12.3 User defined linear Filter

The user can change any of the settings of the predefined filters but not change the name and not add a new filter.



Choose the **type** of linear user filter.

Define the filter by entering the matrix or vector coefficients inside the corresponding field directly.

Two further actions (**divide by** and **set outer rim**) are defined at the bottom of the dialog box. They are performed before the result of the filter is stored.

The **description** entered here will appear below the list of linear filters, when the filter has been selected (see the page before).

Basically, there are three **types** of user filters.

- **NxN matrices**: Usual NxN-matrices (N= 3, 5, 7 or 9) that are applied in the same way as the predefined linear filters. They are defined by entering the matrix elements in the center field.

- **NxN separable matrices:** They work much faster than the usual matrices. If a matrix is separable (is the product of a vertical times a horizontal 1D-vector), several steps of the full matrices operations are redundant. Therefore it can be applied in two steps of vector filter operations: first a horizontal vector, second a vertical vector. These filters are also defined by entering the elements directly into the vectors.

- **1D vectors**: Vector filters in one dimension.

For **seperable matrices** and **vector filters** enter the **coefficients** into the vectors (horizontal and/or vertical). Enter **normal matrice** filters inside the matrice.

Negative or floating point values can only be stored in image buffers of the type FLOAT. So it might be necessary to **convert the output buffer** to a FLOAT buffer, in order not to lose significant digits of the data.

After performing the matrix (or vector) operation, the result buffer can be **divided by** a specified value.

- **sum of coefficients:** Choose this divisor in order not to change the average intensity of the image buffer.
- **value:** Any other divisor can also be defined.

At the **rim** of the image a filter will not work over the same amount of pixels as in the middle. Therefore the result might differ drastically at the rim, espacially if a gradient filter is used. Then it might be useful to set the whole **outer rim** equal to 0. In the default setting the rim of the result buffer is not changed.

12.12.4 Non-Linear Filter

The non-linear filters do not work with the simple matrix equation as the linear filters do. Each non-linear filter has its specific advantages, but on the other hand they are non-linear operations, i.e. the average intensity of the image may change.

For the three filters **Median, Erosion and Dilatation** a $k \times k$ area (k= 3,5,7,9 or 11) is taken and all elements are sorted according to their intensities. Then the elements are stored back following a special rule for each filter.

Median Filter

The median filter takes a k x k-area (k = 3, 5 or 7) and **sorts** all elements according to their intensity. Then the **middle element** is stored back as the new center pixel. Just like the other smoothing filter, it will eliminate high frequency noise. As an advantage it does not eliminate sharp edges as much as the other filters. On the other hand this filter is non-linear, i.e. the average intensity of the image may change. The median filter works mathematically very similar to the erosion and dilation filter discussed below, which have a very different behaviour.

Figure 12.15: Median filter on 3 x 3 area and on 7 x 7 area.

Erosion and Dilation Filter

Just like the median filter this filter will also sorts the pixels of a 3x3-area according to their intensity. But instead of taking the middle element, the erosion will take the element before. The effect is a kind of erosion, valley become deeper and finally the image will be eroded away toward the background level.

The dilation filter does exactly the opposite. It will take the next element after the middle one in the sorting list. The effect is that edges become sharper and plateaus flatten and become more even.

Figure 12.16: Examples of erosion (left) and dilation filter (right).

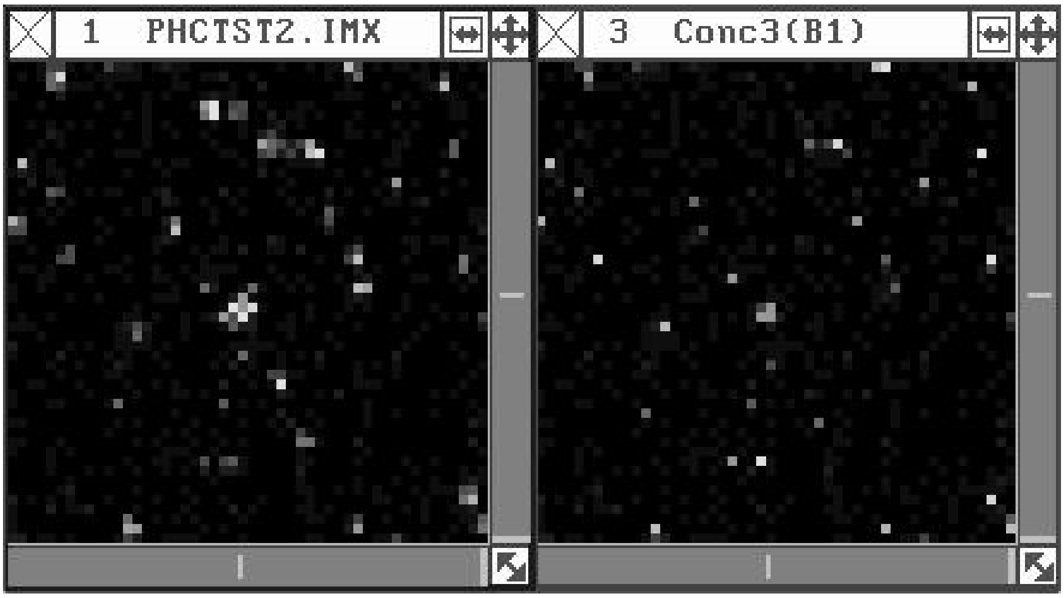


Figure 12.17: Example of concentration filter: original image (left) and filtered image (right).

Concentration Filter and Photon Counting

The **Concentration** filter will search for pixels, whose intensity will be a local maximum and also the intensity must be above the background level as determines by the parameter 'noise_level'. If found, it will move the intensities of the surrounding pixels toward the center to concentrate all power in the middle.

This can be used to locate the photon-'hills' when using an image intensifier. Since all power will be moved to the center, it can then be calculated from the intensity of that center pixel, if possibly more than one photon is located here. This requires the knowledge of how many counts a photon generates. This is dependent on the gain of the image intensifier and must be calibrated once.

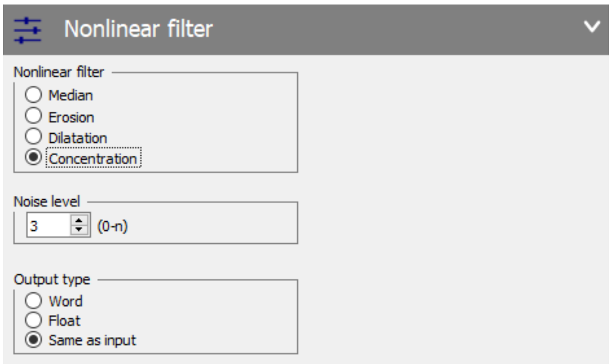
A possible **Photon Counting** function would perform all the necessary steps for calculating how many photons are located where. It will first perform the concentration filter a number of times using parameter 'noise_level' as a threshhold. Usually it is only required once, but in case the 'hills' are smoothed out more than 1 pixel in either direction, it may become necessary to do the filtering twice or three times.

Once the filtering is done, the background level is subtracted from the image and it is divided by the number of counts per photon to calculate the number of photons.

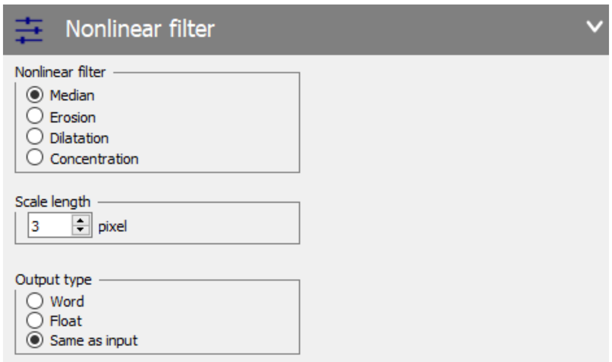
The following macro code gives a fragment of a photon counting function:

```
void PhotonCounting( int bin, int bout, float noise_level,
            int background_intensity, int counts_per_photon )
{
    NonlinearFilter( bin, bout, 4, noise_level );
    // Maybe repeat the above function for different noise levels.
    // Now subtract the background and scale the output.
    B[bout] -= background_intensity;
    B[bout] /= counts_per_photon;
}
```

Filter the source by using one of the Non-Linear Filters **Median**, **Erosion**, **Dilatation**, or **Concentration filter**. The dialog for nonlinear filters can be found in the processing operation. For Concentration the **Noise level** can be selected.



For the other filter types chose a **scale length**.



In the dialog the **Output type** can be selected. The result can be stored either in a word or a float buffer, or in the same type as the input.

12.12.5 Sliding Filter

Sliding Average Filter

This filter computes a local average over a specified scale length by computing the average according to the following equation applied 4-times, by going from left to right, from right to left, from top to bottom and from bottom to top through the image:

S^{avg} (0) = I (0)

and

S^{avg} (i) = (n-1)/n * S^{avg} (i-1) + 1/n * I (i) for $i \geq 1$

where I(i) is the pixel intensity at a certain pixel i and S^{avg} (i) is the computed average at that pixel and and S^{avg} (i-1) is the computed average at the previous pixel. N is the scale length (must be $n \geq 2$), which corresponds about to the distance over which the average is computed.

The effect is a sliding average, where each new pixel is added with a small weighting factor. This type of filter is much faster than any Fourier type of average filter or a linear $n \times n$ matrix filter, since very few multiplications are necessary for each pixel, independant on the scale length. There is a small edge effect at the image borders, but by far not as severe as when using a FFT-Filter. The only drawback is that the image intensities are shifted a bit in the direction of processing (an intensity peak becomes smoothed and the peak center is shifted by n/2). But this is compensated by going not only from left to right, but also from right to left.

The exact equations for the 4 passes are decribed below:

- Going from left to right: For all rows y do:
  S^{avg} (x,y) = (n-1)/n * S^{avg} (x-1,y) + 1/n * I (x,y)
- Going from right to left: For all rows y do:
  S^{avg} (x,y) = (n-1)/n * S^{avg} (x+1,y) + 1/n * S^{avg} (x,y)
- Going from top to bottom: For all columns x do:
  S^{avg} (x,y) = (n-1)/n * S^{avg} (x,y-1) + 1/n * S^{avg} (x,y)

- Going from bottom to top: For all columns x do:
  S^{avg} (x,y) = (n-1)/n * S^{avg} (x,y+1) + 1/n * S^{avg} (x,y)

For example, if the scale length is 10 pixel, then

S^{avg} (i) = 0.9 * S^{avg} (i-1) + 0.1 * I (i)

Sliding Minimum Filter

This filter computes a local minimum over a specified scale length. The filter has the same principle as the sliding average, but with slightly different equations:

With m = n / 2 half of scale length, do

Step 1: going from left to right: For all rows y do:

fmin = S^{avg} (0,y) = I (0,y)

and

if (I (x,y) $>$ fmin)

fmin = (m-1)/m * fmin + 1/m * I (x,y)

else

fmin = I (x,y)

S^{avg} (x,y) = (n-1)/n * S^{avg} (x-1,y) + 1/n * fmin

where fmin is an intermediate variable for storing a sliding minimum over half the scale length, which again is smoothed over the full scale length.

Step 2-4: same as for sliding average

Note that this filter can still produce sometimes pixel intensities that are slightly above the original values. The strict sliding minimum filter explained below is strictly below the original intensities, but is not as smooth as this filter.

Sliding Maximum Filter

Same as sliding minimum, only '$<$' instead of '$>$' in the above equation in step 1.

Strict Sliding Minimum Filter

Same as sliding minimum, except that none of the resulting pixel intensities will be below the intensity of the original pixel at a certain position. The equations are slightly different:

- Going from left to right: For all rows y do:

  fmin = S^{avg} (0,y) = I (0,y)

  and

  if (I (x,y) $>$ fmin)

  fmin = (m-1)/m * fmin + 1/m * I (x,y)

  else

  fmin = I (x,y)

  S^{avg} (x,y) = min(fmin, (n-1)/n * S^{avg} (x-1,y) + 1/n * fmin)

- Going from right to left: For all rows y do:

  S^{avg} (x,y) = min(S^{avg} (x,y) ,
  (n-1)/n * S^{avg} (x+1,y) + 1/n * S^{avg} (x,y))

- Going from top to bottom: For all columns x do:

  S^{avg} (x,y) = min(S^{avg} (x,y) ,
  (n-1)/n * S^{avg} (x,y-1) + 1/n * S^{avg} (x,y))

- Going from bottom to top: For all columns x do:

  S^{avg} (x,y) = min(S^{avg} (x,y),
  (n-1)/n * S^{avg} (x,y+1) + 1/n * S^{avg} (x,y))

Strict Sliding Maximum Filter

Same as strict sliding minimum, only 'max' instead of 'min', and '<' instead of '>' in the above equation in step 1.

Sliding Average Filter (gaussian profile)

The algorithm follows the same pattern as the sliding average filter described above: making horizontal passes left to right and back again and averaging it with the vertical passes top to bottom and back again.

But in contrast of using a sliding value and the current pixel only, the gaussian profile filter uses the last 3 pixels in an iterative convolution to create an approximation to a gaussian bell 2D filter. Below the horizontal pass is described exemplarily :

- set first three destination pixels $I_d(0,y)$, $I_d(1,y)$ and $I_d(2,y)$ to the average intensity of the three source pixels $(Is(0,y)+Is(1,y)+Is(2,y))/3$
- then starting from $x = 4$ calculate the next pixels $I_d(x,y)$ iteratively using the pixel $I_s(x,y)$ and previous destination pixels $I_d(x-1,y)$, $I_d(x-2,y)$ and $I_d(x-3,y)$:

$$I_d(x,y) = B \times I_s(x,y) + \frac{B_1 \times I_d(x-1,y) + B_2 \times I_d(x-2,y) + B_3 \times I_d(x-3,y)}{B_0}$$

with the following constants

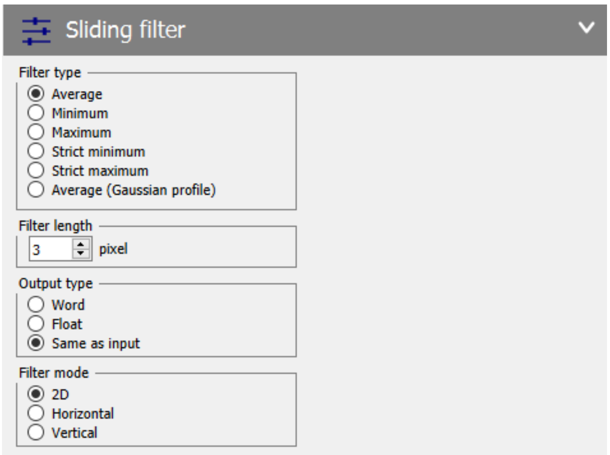
$$\begin{aligned}
\sigma =& \mathit{filter\ length}/2.0 \\
Q =& 0.98711 * \sigma - 0.96330 \quad \mathit{for} \quad \sigma \geq 2.5 \\
& 3.97156 - 4.14554 * \sqrt{1 - 0.26891 * \sigma} \quad \mathit{for} \quad \sigma < 2.5 \\
B_0 =& 1.57825 + 2.44413 * Q + 1.4281 * Q^2 + 0.422205 * Q^3 \\
B_1 =& 2.44413 * Q + 2.85619 * Q^2 + 1.26661 * Q^3 \\
B_2 =& -1.4281 * Q^2 - 1.26661 * Q^3 \\
B_3 =& 0.422205 * Q^3 \\
B =& 1 - (B_1 + B_2 + B_3)/B_0
\end{aligned}$$

The same procedure is done backwards again. The vertical pass is equivalent with x and y exchanged moving from top to bottom and back again working on the horizontally smoothed image.

Please note that the filter length of this filter is defined as a float value (rather than an integer value of the *sliding average* filter) and hence a finer adjustment of the smoothing level is possible.

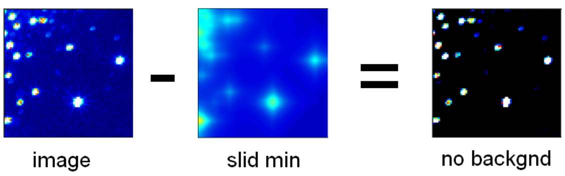
Reference: *A.Lukin* "*Tips and Tricks: Fast Image Filtering Algorithms*" in *Proceedings of GraphiCon'2007, Moscow, Russia, June 2007, pp. 186-189*

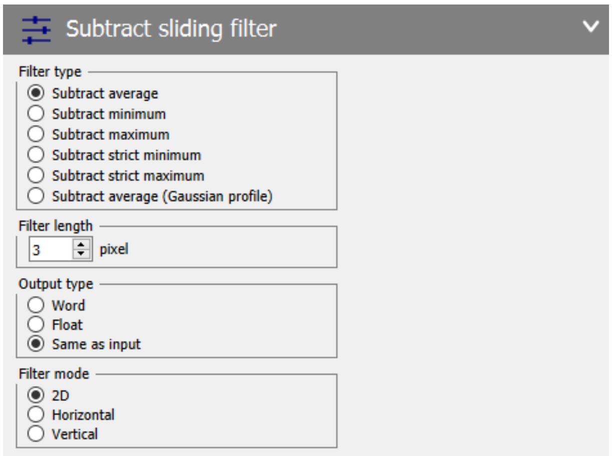
The dialog for Sliding filters can be found in the processing operation. Select a **Filter type** between **Sliding average**, **minimum**, **maximum**,**Sliding strict minimum**, **strict maximum**, and **Gaussian average** and the corresponding **Filter length** from the list. In the dialog the **Output type** can be selected. The result can be stored either in a word or a float buffer, or in the same type as the input. Additionally, a **Filter mode** can be chosen for **Sliding average (Gaussian profile)** (see page 474)).



12.12.6 Subtract Sliding Filter

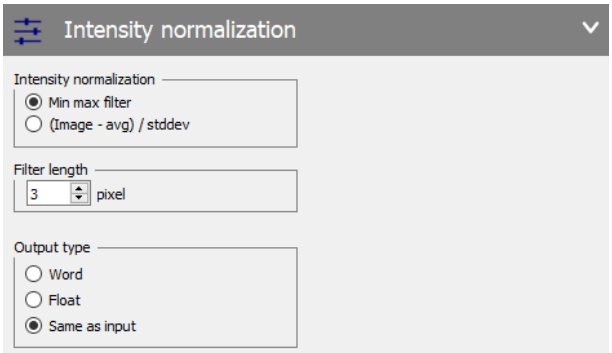
Subtracts a **Sliding Filter** (see page 273 for Sliding Filters) from the original image. Select the corresponding **Filter length** from the list. In the dialog the **Output type** can be selected. The result can be stored either in a word or a float buffer, or in the same type as the input. Additionally, a **Filter mode** can be chosen for **Subtract sliding average (Gaussian profile)** (see page 474)).





12.12.7 Intensity normalization

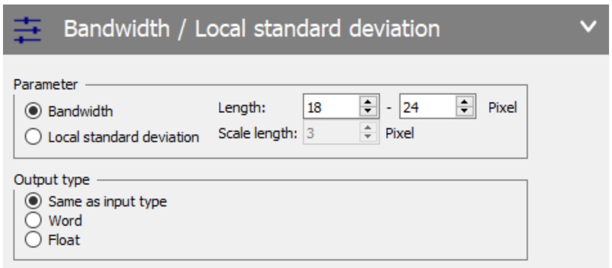
The Image normalization function **(Image - avg) / stddev** subtracts the local average (with the filter length given) and divides the result by the local intensity standard deviation. The result is an image with local mean intensity zero and a fixed dynamic range everywhere. Otherwise, choose **Min max filter** to normalize the intensity to minimum or maximum Intensity.



12.12.8 Bandwidth / Local standard deviation

Choose a **pixel length** as a threshold for the **Bandwidth** filter. Within this threshold a sliding average is computed. Choose a **Scale length** for computing the local standard deviation. A high local intensity change results in a high local standard deviation.

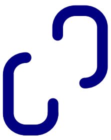
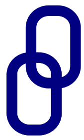
In the dialog the **Output type** can be selected. The result can be stored either in a word or a float buffer, or in the same type as the input.



12.12.9 Remove background / unsteady reflections

Removes unsteady background (e.g. laser light reflections on moving surfaces, like a propeller blade) from images while keeping the particles. The default mode is to compute the background, then subtract it from the original image and output the result of that subtraction.

One of the following algorithms can be selected to **compute the background**:

- **Sliding average or Sliding minimum**. The filter sizes for x and y direction may be defined independently, or you can use the button with the chain symbol to ensure that they are the same. More detailed information regarding these two filters can be obtained from pages 273 and 274, respectively.
- **Directional minimum** computes the background for every pixel by taking the minimum along multiple lines evenly-spaced over different spatial directions, then taking the maximum of these minima. The filter size represents the radius around each pixel in which the calculation is performed.
- **Average polynomial fit** computes the background by fitting a 2nd-order polynomial function to the local neighborhood of every pixel. The filter sizes for x and y direction may be defined independently by the open chain button  .The closed chain button  ensures that the filter sizes for both directions are the same.
- **Anisotropic diffusion** diffuses the intensity along the edges but not across the edges. Thereby, over multiple iterations, the reflection in the background image is preserved, while the particles are removed.

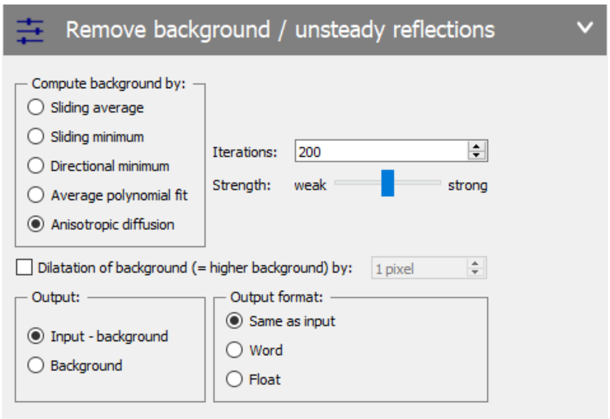
As a result, subtracting the background image from the original image, only particles should remain and not reflections. The recommended number of iterations is 200. Too few iterations may lead to particles disappearing, too many iterations lead to reflections reappearing.

Reference: *Adatrao, S., & Sciacchitano, A.* "*Elimination of unsteady background reflections in PIV images by anisotropic diffusion*" in *Measurement Science and Technology, 2019*.

You may choose to dilate the background before subtraction (see page 270 about dilation).

In the **Output** box, you can choose "Background" instead of "Input - background" to output the background instead. This is useful, if you want to precisely know what is being subtracted.

The **Output format** can be selected. The result can be stored in a word buffer, in float buffer, or in the buffer of same type as the input.



12.13 Operation: Copy and Reorganize Frames

12.13.1 Delete or change order of frames

Reorganize the frames of every source buffer of a set with images or vector fields. Figure 12.18 shows the parameter dialog with the **available frames** on the left and the **selected frames** on the right. During processing the source frames are copied to destination in the order of the right list from top to bottom.

Select a frame in the left or right list, then click on the arrows in the center to move the selected frame either from left to right or from right to left. When moving from left to right, means from **available frames** to **selected frames**, the frame is inserted at the end of the list as last frame of the destination. By marking a frame in the right list and clicking on the up and down sided arrows the frame order can be changed.

If a frame is in the list of **available frames** when processing is started, this frame will be deleted.

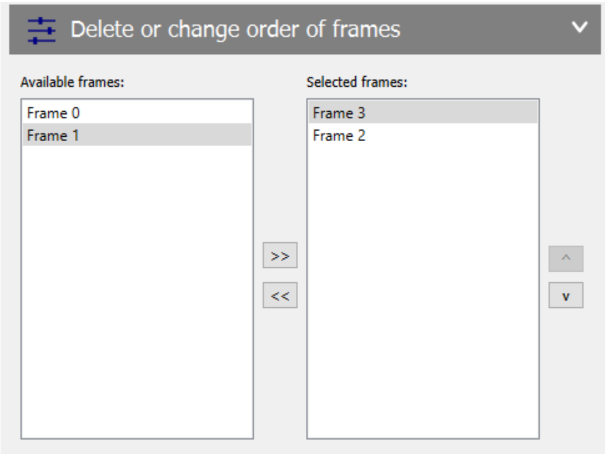
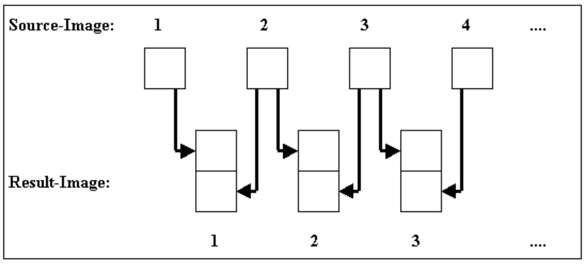


Figure 12.18: Parameter dialog to delete or change order of frames.

12.13.2 Create Multi-Frame Buffer from Time Series

A number of single frame source buffers are combined into one (or more) multi frame buffers. The following options are available.

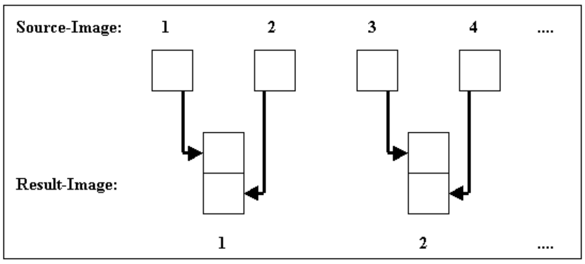
- **create (n-1) images: 1+2, 2+3, 3+4...:** Combine each consecutive pair of source images into one resulting double frame image.



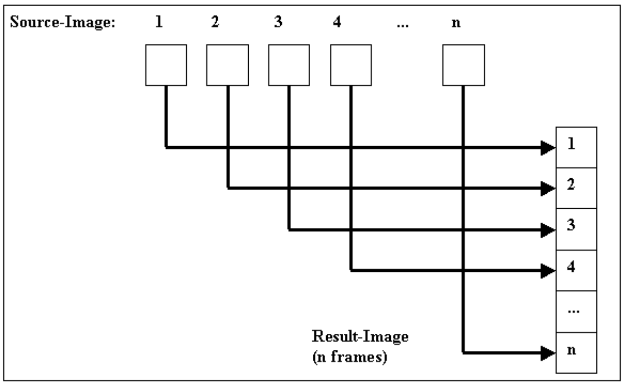
- **create (n-1) images: 1+2, 1+3, 1+4...:**
  Take the first image of the source set as first frame of each resulting

double frame image. All other source images are taken as second frame of each result.

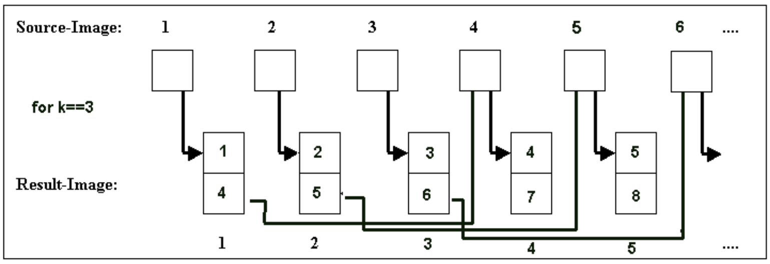
- **create (n/2) images: 1+2, 3+4, 5+6...:**

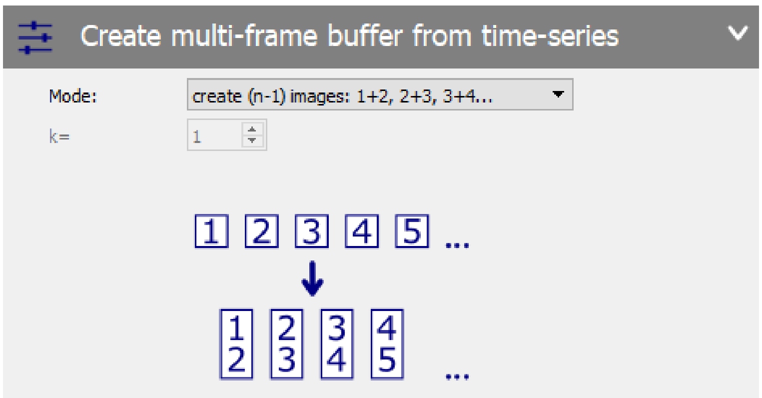


- **create 1 image with n frames: 1+2+3+4...:**



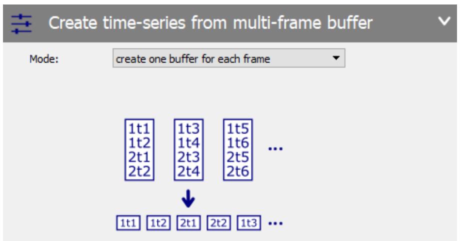
- **create (n-k) images: 1+(1+k), 2+(2+k), 3+(3+k)...:**

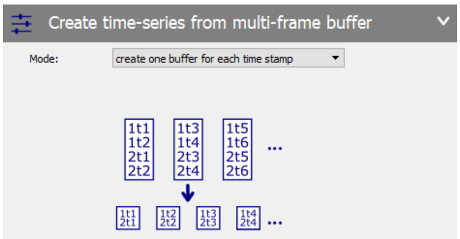




12.13.3 Create Time Series from Multi-Frame Buffer

All frames of a multi frame buffer are separated and stored as single frame buffers in the destination set. The first frame create the first result file, the second frame creates the second file and so on. It is also possible to create one buffer for each time stamp. Here, all frames with the same time stamp are collected into the same buffer. If the source set had n files with each m frames then the resulting set is created with m files with each n frames.



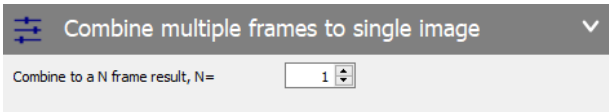


12.13.4 Exchange z <-> Frames

If a source buffer has planes instead of frames, then this function swaps both dimensions. The result will get n frames from n planes of the source.

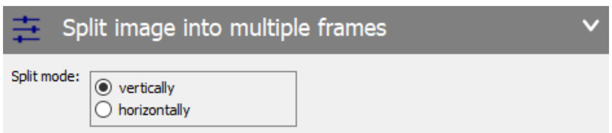
12.13.5 Combine multiple frames to single image

Choose a number **N** for frames to combine and join images vertically.



12.13.6 Split image into multiple frames

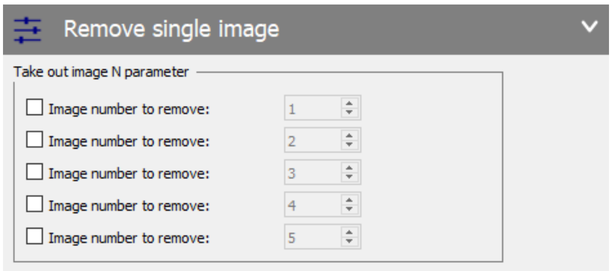
A single frame buffer as source creates a double frame destination buffer. In **horizontal** mode the left half of the image is taken as first frame and the right half as second frame. In **vertical** mode the upper half creates the first frame and the lower half the second frame.



12.14 Operation: Copy and reorganize data sets

12.14.1 Remove single image

Select images that should be removed from the source set.



12.14.2 Convert to plot data set

Convert image to list of plots. One horizontal line is converted to one plot.

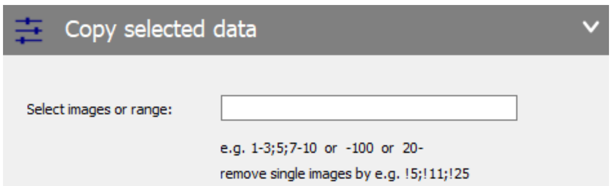
12.14.3 Copy

Copy the source set using the selected range and increment of the **Processing** dialog and create a new set with less files.

12.14.4 Copy selected data

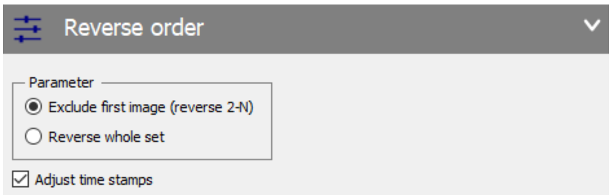
Copy the source set using the selected range and increment of the **Processing** dialog and create a new set with less files.

Enter a list of set items or ranges of set items and create a new set with this files. With definition `1-3;5;7-10` the files 1 to 3, 5 and 7-10 would be taken from the source set and thrown as files 1 to 8 into the destination set. Sources files 4 and 6 are skipped. With definitions like `-100` or `20-` either the first 100 files or the files starting with index 20 up to the end are copied. Use a definition like `!5` to skip single files of the source set.



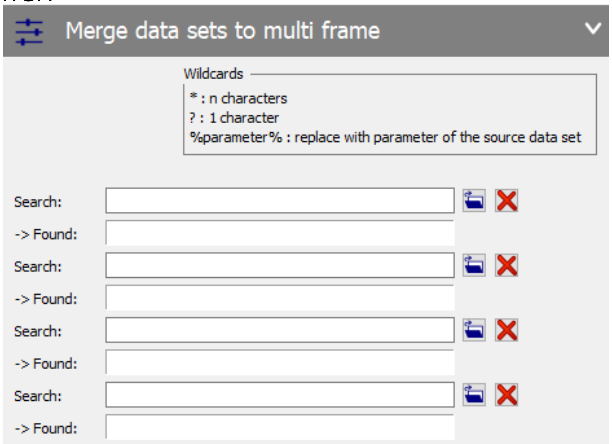
12.14.5 Reverse Order

Copy all sources and reverse the storing order. Choose as Parameter between **Exclude first image (reverse 2-N)** and **Reverse whole set**.



12.14.6 Merge data sets to multi frame

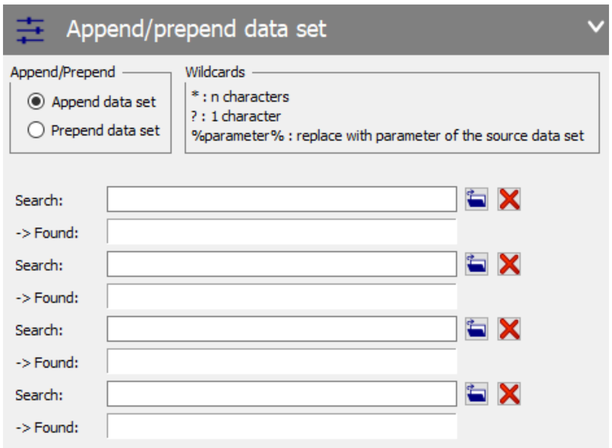
Merge all sources with the corresponding file of the specified datasets to a multiframe buffer:



Press the **folder** button to select a dataset or enter the name of the dataset in the edit field. The **cross** button can be used to clear the current entry. It is also possible to use the following wildcards to find a corresponding dataset:
* : n characters
? : 1 character
%parameter% : replace with parameter of the source dataset
The files of all found datasets will be merged.

12.14.7 Append/prepend data set

Copy the source data set and **append** or **prepend** the buffers of the specified data set. See operation **merge dataset**.



12.14.8 Convert image set to image volume

Converts a set of N single frame images to a set of one volume with N planes.

12.14.9 Interpolate hypersampling data to image set

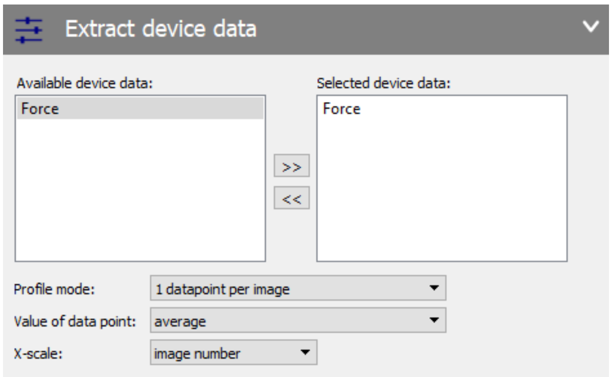
Create images at equidistant time stamps.

12.15 Operation: Plot

12.15.1 Extract Device Data

Extract device data from the source dataset:

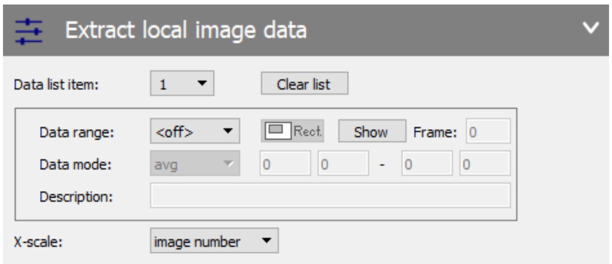
- **Data source**: Select the device source you want to extract.
- **Channel**: Select the analog input channel.
- Profile mode **1 datapoint per image**: Create one profile for the whole dataset with one datapoint for every source image.
- Profile mode **1 profile per image**: Create for every source image one devive data profile.
- X-Scale: Select the X-scale of the profile.



12.15.2 Extract Local Image Data

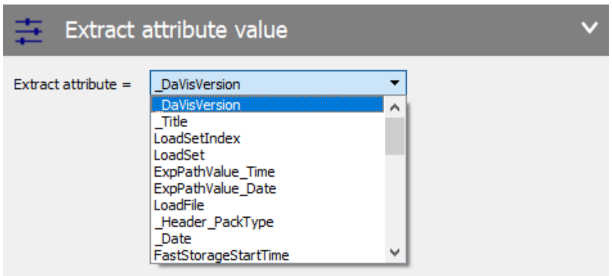
Specify up to ten rectangular regions in the source image which local data are stored in a profile:

- **Data list item**: Select 1 of the 10 items you want to specify.
- **Data range**: Select the data range of the selected item or **off** to disable this item. Range can be a frame, the complete image or a rectangular region. In **rectangle** mode press button **Rect** to select the region in the source window by dragging the mouse between two points. The pixel position of the rectangle is given.
- **Data mode**: Select the type of data which is calculated within the data range: This can be one of the average, stdev, minimum, maximum or sum of all pixels in the rectangular area.
- **Description**: Enter a description text which is displayed on the result profile.
- **X-Scale**: Select the X-scale of the profile. This can either be the image number from the source set or the time attribute of the image relative to the first image.



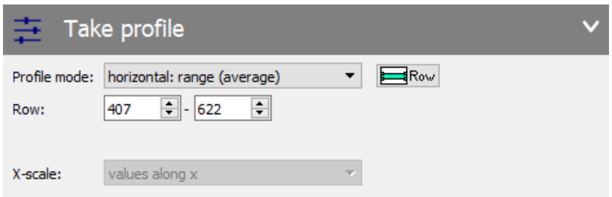
12.15.3 Extract Attribute Value

Extract the value of the specified attribute from the source dataset. The **parameter** dialog includes a list of all available attributes from the source buffer.



12.15.4 Take Profile

Extract a profile from the image or vector field. Press the button right to the profile mode to select a row, column, circle or arbitrary line in the source image.



Profiles can be taken in **Modes** *horizontaly* (from rows) or *verticaly* (from columns). Unless it is further specified, such a profile simply corresponds to the intensities of a single row or column. In the other modes the profile consists of the average values (*range-avg*) or the sum (*range-sum*) over a certain range of lines. In the mode *arbitrary line* a line at any angle and position can be chosen. In this mode the resulting pixel values along the line are computed by interpolating the closest four neighbouring pixels. In mode *circle* the user defines the center position and the radius of the circle. All (interpolated) pixel on the circle are taken clockwise from the right side of the circle. Mode *intensity histogram* calculates a histogram of the quantity of pixels for all intensity values of the complete source buffer.

For vector buffers a **vector profile** is calculated for the vector length or for the vector components (X, Y or Z).

Take profile of line/circle/lines: The user can enter the range **in pixel** directly in the text items left to the button. Depending on the mode some different edit items are displayed, e.g. the coordinates of the first and last point of a line, or the center coordinates and the radius of a circle. After entering the coordinates, the profile can be taken by pressing the button.

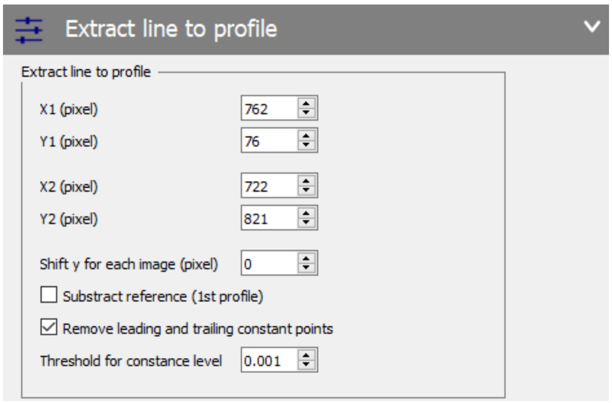
The user can take a profile by mouse directly inside the original image window of the source buffer after having clicked the right button in teh first parameter line next to **profile mode** selection. When the image window of the **source buffer** is opened, select the borders of the range with the mouse by clicking once on each border position. In mode *arbitrary line*, the line is defined by dragging the mouse between start position and end position of the line. In mode *circle* press the mouse button on the center of the circle, then move the mouse and hold the left mouse key to define the radius and select the desired radius by releasing the mouse key.

12.15.5 Extract Line to Profile

For each image extract a free line between two given pixel positions. The result of this operation is one profile for each source image. The scales of this profile are the original x-scale and the original intensity scale.

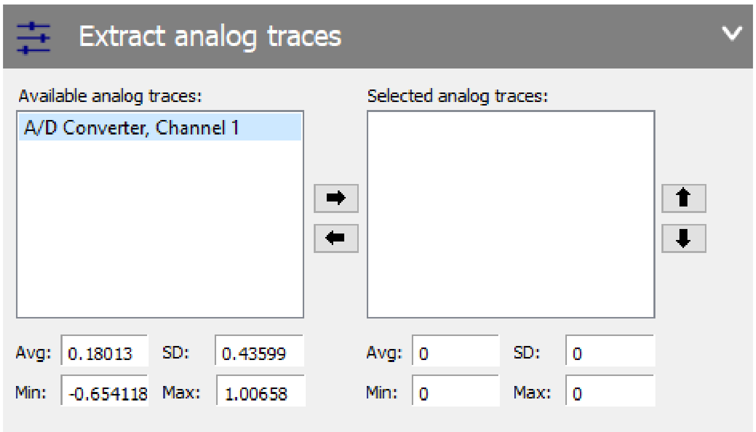
Optional the vertical position of the line can be **shifted** from image to image by a certain pixel shift. If e.g. the start position of the line is (123,5)-(564,800) and the shift value is 9, then the line of the second image is taken from (123,14)-(564,809) and the line of the third image from (123,23)-(564,818).

The line from the first image can be used as base line for all other profiles. If this **reference** is enabled, then the complete first profile is subtracted from the profiles of all other images. At the end the first profile is constant 0 while the other profile intensities give the difference to the first profile.



12.15.6 Extract analog traces

This operation extracts one or more profiles with data from recordings with AD converter. The result of each source file is a buffer containing one or more profiles.



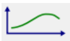
The list on the left side of the parameter dialog gives the names of all AD profiles from the source buffer. Selecting an item will display a preview of this profile on bottom of the parameter dialog. Both buttons in the center between both lists are used to include the selected profile into the result or remove from the result. The arrow buttons on the right can be pressed to change the order of all extracted profiles.

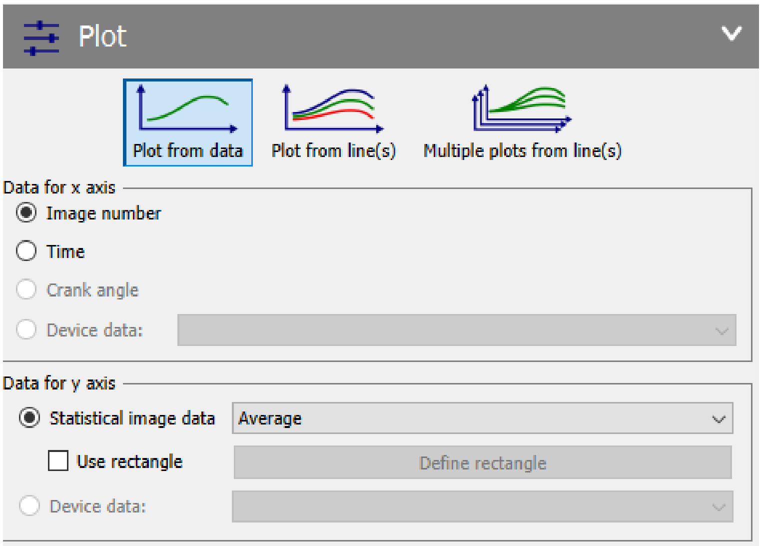
12.15.7 Extract Recording Rate

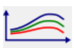
During the recording in **DaVis** each acquired image gets a time stamp relative to the start of recording. This operation calculates the time difference between two consecutive images and stores this differences as frequency value in a profile. The horizontal scale of this profile is the image number, the vertical scale is the frequency in Hz.

12.15.8 Plot

Create a plot between two scalar sources of the input buffer. The result will be an Plot-set for each source set.

For **plotting from data** , the x-dimension can be taken from the image number of the source set, time, or from the selected device data. Please note that device data can be only selected, if it is available in the source set. The y-dimension can be taken from statistical image data of the whole image or a defined rectangular region. For the latter, the check mark **Use rectangle** has to be enabled.



Select one of the modes **Plot from line(s)**  or **Multiple plots from line(s)**  in the plot mode group. Depending on the selection, you get one plot with *n* lines or a set of *n* plots with one line each, respectively. Then, select the extraction mode

- arbitrary line
- vertical
- horizontal
- vertical range

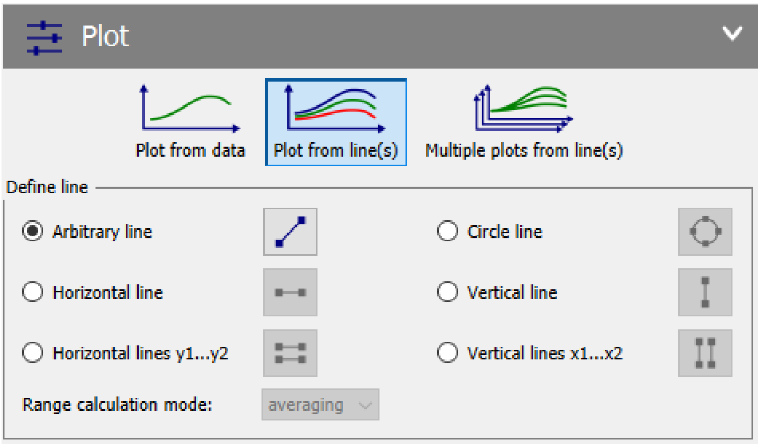
- horizontal range

and manually define the position in the vector field.

For an arbitrary line, a vertical line or a horizontal line, the image intensitiy is plotted along the line.

For horizontal range, the values are averaged inside the horizontal range (along x) for each y position and plotted against that y position. For the vertical range mode, it is vice versa.

The X-axis is set automatically to **values along line** and cannot be changed.



Please refer to Section **Plots for Image Sets** 20.3.2 for further details.

12.15.9 Extract data traces

If the recording also acquires A/D data together with the images, then those data can be extracted from the source set and stores as a new profile set.

12.16 Operation: Convert Image to RGB

The operations of this group create a RGB image from each grayscale source image.

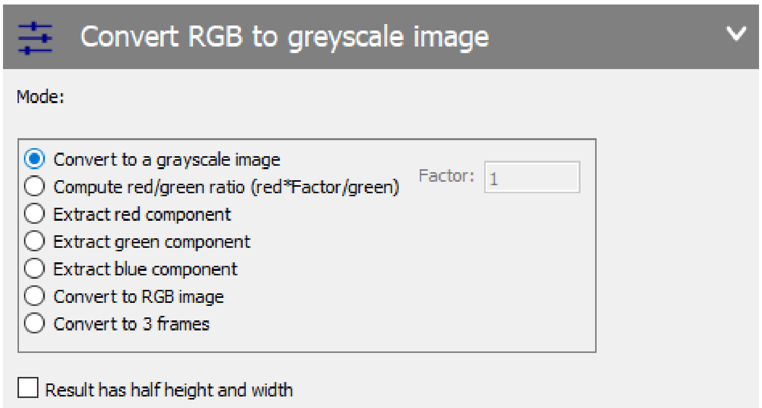
12.16.1 Combine R/G/B Frames to RGB Image

The source buffer must include at least three frames. The operation creates a colored RGB image using the first frame as red color component, the second frame as green component and the third frame as blue component.

12.16.2 Convert RGB to greyscale image

The Imager 3 camera uses the Bayer pattern (see page 94 for a detailed description) to store the three color components Red, Green and Blue in a grayscale image. This operation is needed to preprocess those special images.

The Bayer pattern includes a number of small 2x2 submatrices which store the red, green and blue color values. Therefore the real image has half of the width and height as the camera image. Select **Result has half width and height** to shrink the destination size from the source (virtual) size to the real size.



Convert to grayscale image simply takes the four pixels of the Bayer pattern and sums them up, using both green pixel as one averaged value.

Compute red/green ratio as $red * factor / green$ for every four pixels of the Bayer pattern.

Extract red / green / blue component takes one intensity from the four pixels of the Bayer pattern. In the case of green, the average of both green pixels is used.

Convert to RGB image results in a real RGB color image.

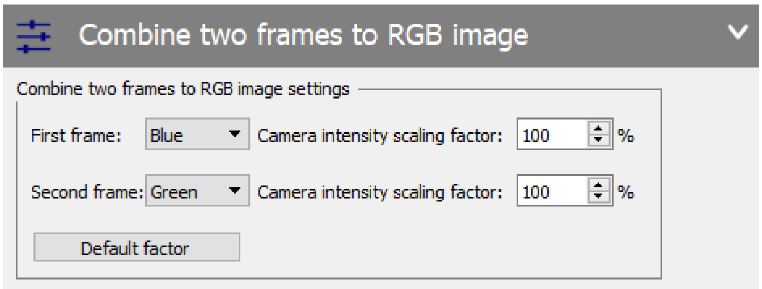
Convert to 3 frames results in 3 frame image, holding red, green and blue component.

12.16.3 Combine two frames to RGB Image

This function is used to merge information from a double frame into a single RGB frame. Each frame can be displayed as a different color. This function is useful when superimposing 2-phase information in a single frame while keeping the phase information separated.

The source buffer must contain two frames containing grayscale images. The operation creates a colored RGB image. The user can select the color component (Red, Green or Blue) into which each frame is converted. The color image format allows an 8-bit depth for each color component. The operation automatically converts the source images from the source image resolution to the 8-bit color component resolution (**Camera intensity scaling factor: 100 %**). By changing the **Camera intensity scaling factor** for each frame the relative display intensity for each color component in the RGB image can be adjusted. The **Camera intensity scaling factor** can be reset to 100 % using the **Default factor** button.

When adjusting the color and intensity settings the preview image is updated live when the **auto update** checkbox is active. If the **auto update** option is turned off the result image can be refreshed by pressing the **Update preview** button.



13 Processing Vector Fields

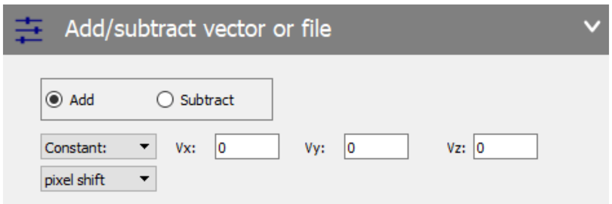
13.1 Operation: Basic Vector Arithmetic

13.1.1 Add/subtract vector or file

A vector is added or subtracted. This single vector or vector field is given

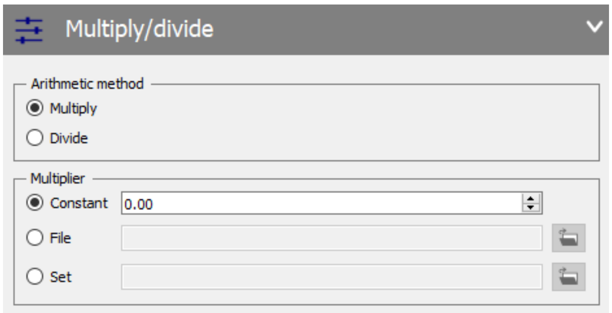
- as **constant** with specified components Vx, Vy, Vz,
- as **file** at given path,
- as **dataset** at given path.

The operation can be executed on the unscaled values as **pixel shift** or on the **scaled values**.



13.1.2 Multiply/divide

Each vector (i.e. all Vx, Vy, Vz components) is multiplied/divided by the specified factor.



13.1.3 Rotation 3D vector

Calculates the 3D-rotational vector with components

$$V_x = E_{zy} - E_{yz} \quad (13.1)$$
$$V_y = E_{xz} - E_{zx} \quad (13.2)$$
$$V_z = E_{yx} - E_{xy} \quad (13.3)$$

based on the information of a two-frame vector buffer (a buffer containing two vector fields).

13.1.4 Lamb Vector

Calculates the Lamb vector field, which is the cross product of the vorticity and the vector at each position in the vector field:

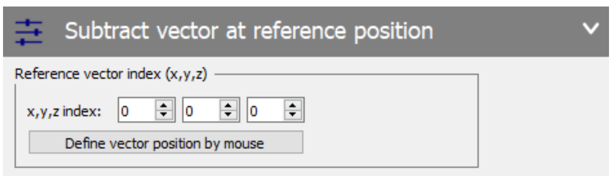
$$\vec{V}_L = \vec{\omega} \times \vec{V} = \vec{V}_{ROT-3D} \times \vec{V} = \begin{pmatrix} E_{zy} - E_{yz} \\ E_{xz} - E_{zx} \\ E_{yx} - E_{xy} \end{pmatrix} \times \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}.$$

One Application is the *Powell sound source* (PSS), which is the divergence of the lamb vector field (which can be applied to the vector field by a separate Processing Operation)

$$PSS = div\vec{V}_L = \vec{\nabla} \cdot \vec{V}_L = \frac{\partial (V_L)_x}{\partial x} + \frac{\partial (V_L)_y}{\partial y} + \frac{\partial (V_L)_z}{\partial z}.$$

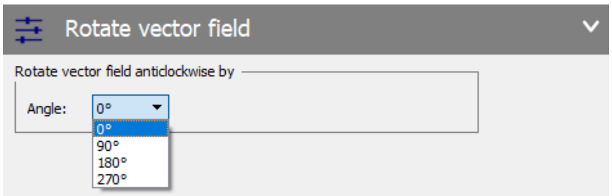
13.1.5 Subtract vector at reference position

This operation subtracts the vector at the reference index from the vector field, forcing the reference point to zero and changing all other vectors accordingly.



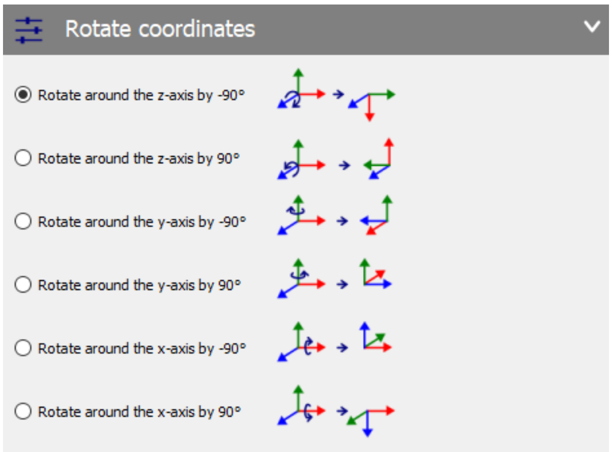
13.1.6 Rotate vector field

This function rotates a vector field anticlockwise by 0°, 90°, 180° *or* 270°. Existing scalar fields will be rotated as well and the X-/Y-scales of the result vector fields are adapted appropriately.



13.1.7 Rotate coordinates

Rotates the vector field around the x, y or z axis.



13.2 Operation: Vector Properties

13.2.1 Crop or extend field of view

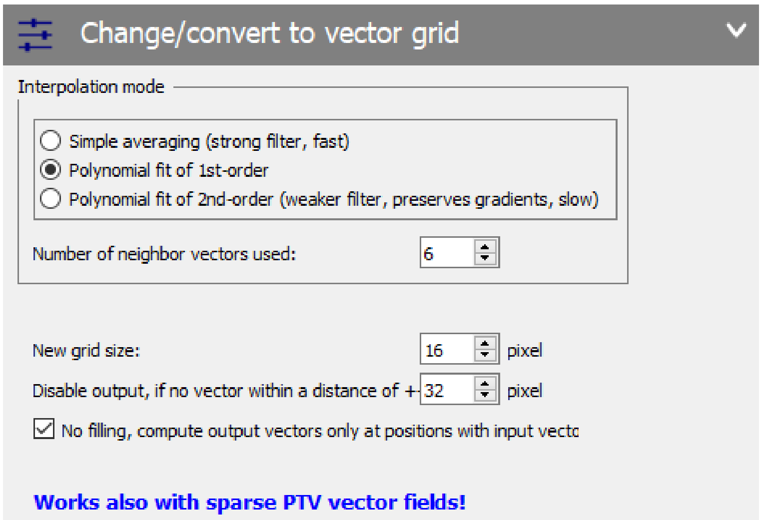
Copy a free volume from the source buffer into the destination. The vector ranges are given separately for each axis (X, Y, Z, F). Select a checkbox to enter a restricted range. Otherwise the complete range is copied. Copy a rectangular region out of the source image. On multi frame buffers the same rectangle is copied from each frame, and the result is a multi frame buffer again. The coordinates of the rectangle can either be entered directly into the edit items, or the upper left and lower right corner can be marked by dragging the mouse in the source image when pressing button **Draw Rect**.

13.2.2 Change / convert to vector grid

Converts a given vector field to a vector field with a different grid spacing. New vectors are interpolated using the specified algorithm (averaging, polynomial fit of 1st or 2nd degree).

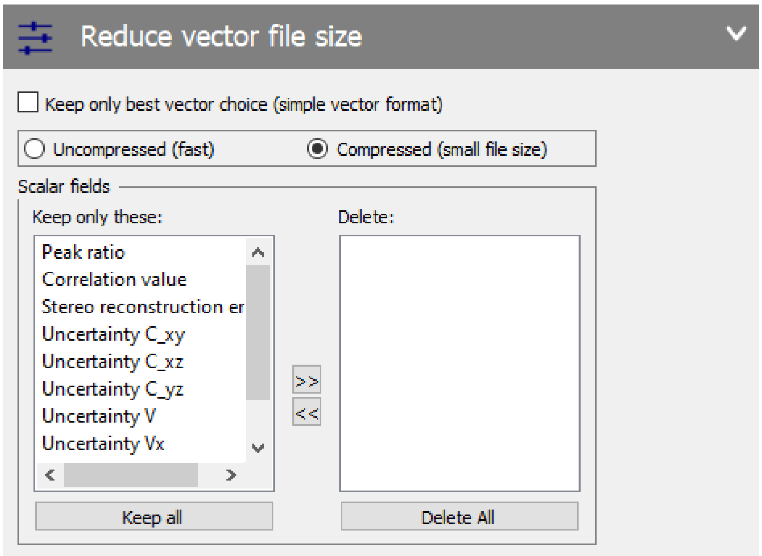
Parameters to be set are the number of required vectors used in the neighborhood, the maximum distance for the search of neighbor vectors (if not enough vectors are found than the vector at that position is disabled) and specify, if all vector positions should be calculated or only positions where before existing vectors were present.

This function can also be used to convert the legacy PTV vector fields (vector field with grid 1) to a vector field with vectors on a regular grid. The regular grid is required e.g. for vector postprocessing or for calculating derivatives. Please note that this operation only applies to vector fields and not to the particle tracks calculated in Davis 10.1 or later.



13.2.3 Reduce vector file size

This operation allows to delete data from existing vector files in order to save space on the harddisk. The deleted data cannot be retrieved later.



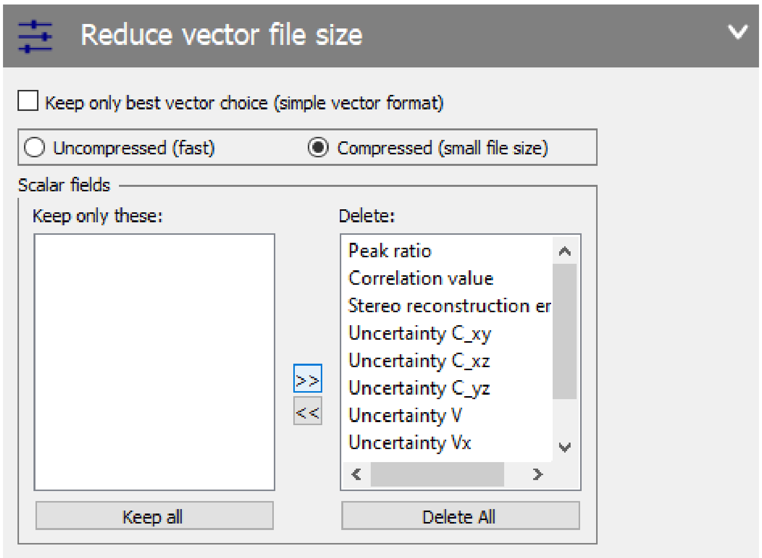
The options are

- **Keep only best vector choice (simple vector format)**- deletes the multiple vector choices (shifts that correspond to the 4 highest peak positions) and only keeps the currently active choice.

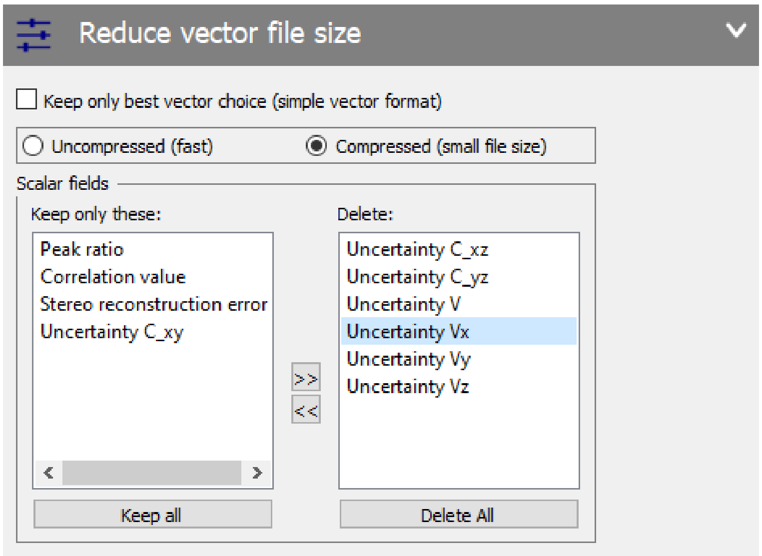
- **(Uncompressed fast)/Compressed (small file size)**- switches to the zlib compression format which leads to smaller file sizes but may require more time to load and decompress from disk. This may lead to slower processing or playback of data in the movie player

- **Keep/Delete Scalar fields**- depending on the type of evaluation (2D, Stereo) additional data is stored with each vector as the peak ratio, correlation values, uncertainty values, average particle sizes, which may be deleted. Please note that some post processing filters (e.g. delete if peak ratio < threshold) need these data to work.

You have two buttons to **Keep all** that data which moves all entries in the Keep box (as seen in the screenshot above) and to **Delete all** which selects all data as to be deleted by moving them to the right box (see figure below).



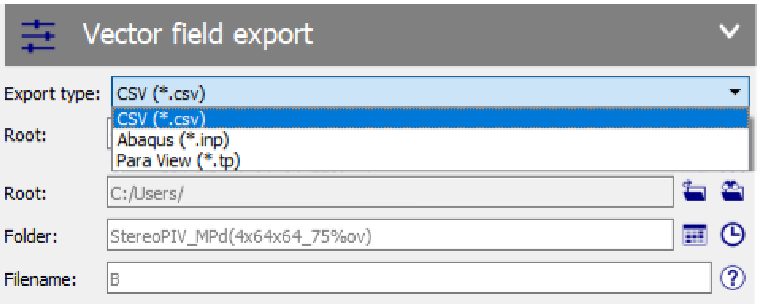
Furthermore you may use the » and the « Buttons which moves the highlighted entry to the other window (see figure below).



13.3 Export

13.3.1 Vector field export

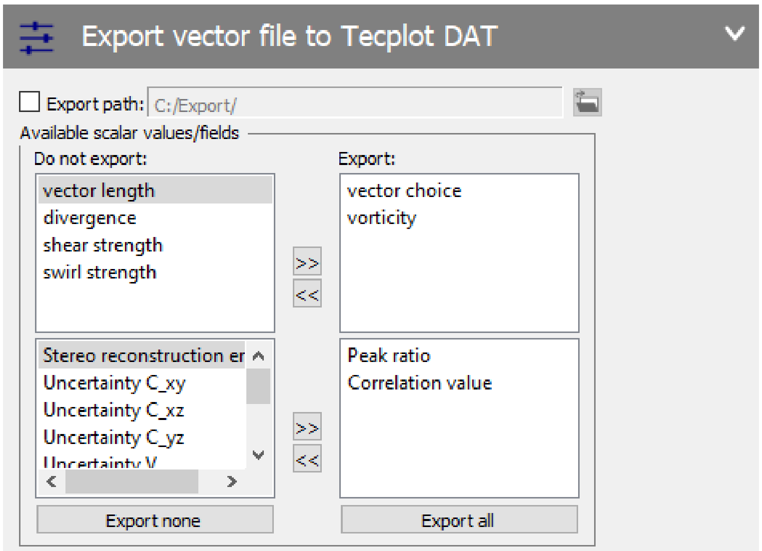
Exports vector fields in various file formats, such as CSV, Abaqus, and Para View. Therefore, chose **Export type**. To select **Root**, click on either to select a directory, or on to open export directory. Moreover, select a **Folder** -, and a **File** name.



13.3.2 Export vector file to TecPlot DAT file

Export vector fields into `DAT` format for later use with e.g. the **TecPlot** software. In the parameter dialog a free choice of additional components, such as vecctor choice, vorticity, etc. and scalar fields, e.g peak ratio, correlation value, etc. can be selected to be included into the created file. By default the vector components (u, v and w) are exported only.

Use the file select button on top of the dialog to define the location of the resulting files. Otherwise, the file will be stored in the current working direction. The operation store a single `DAT` file for each source vector field.

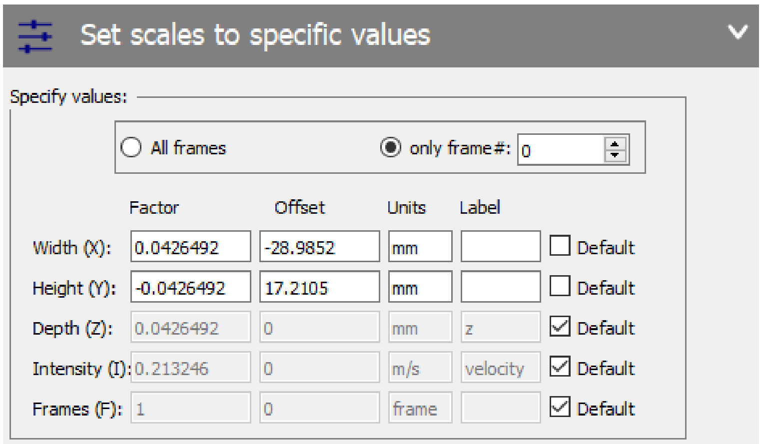


13.4 Operation: Scales

13.4.1 Set Scales

Change the scaling of the source buffers either to direct defined values or to values taken from another source.

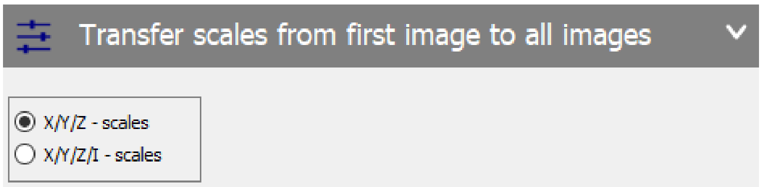
set scale to specific values: Change the image scales of all directions: X, Y, Z, I(intensity), F(frames):



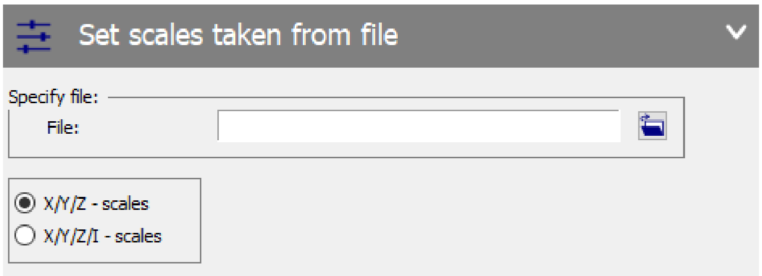
Choose between:

- **Use scale for all frames**: Define the scales for X,Y,Z,.. in the lower part of the dialog. These scales will be set for all source frames.
- **Set each frame scale separately**: Use the up/down button of the **frame#** item to toggle between all source frames. With this it is possible to set each frame scale separately.

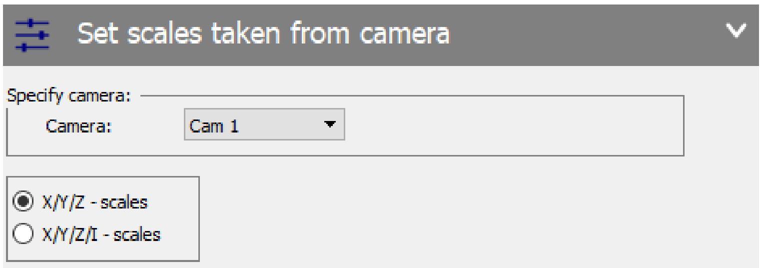
Transer scales from first image to all images: Extract the scale parameters from the first source buffer (choose between X/Y/Z - or X/Y/Z/I - scales) and apply to all other vector buffers.



Set scales taken from file: Extract the scale parameters from a file (choose between X/Y/Z - or X/Y/Z/I - scales) and apply this to all images buffers.



Set scales from camera: Select a camera as scaling source and choose between X/Y/Z - or X/Y/Z/I - scales.



13.4.2 Transfer Scales

Reset scales: Reset all scales to the default values with 1 as factor, 0 as offset, pixel or intensity as unit and a empty description in the X/Y/Z-direction and additionally in I-direction.

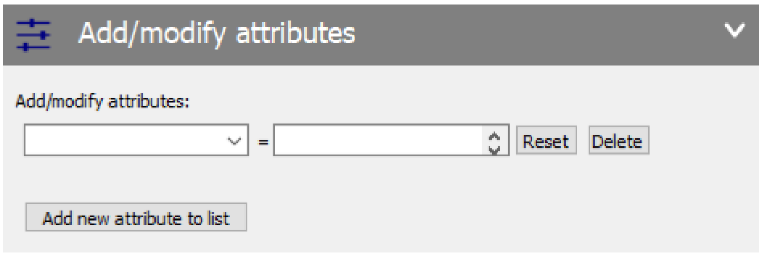
Adjust XY-scales to MITAS scan position: The image scales are adjusted by an offset in the X- and Y-direction that originates from a translation stage scan in the corresponding directions. Make sure that the images are scaled in mm to use this operation.

13.5 Operation: Attributes

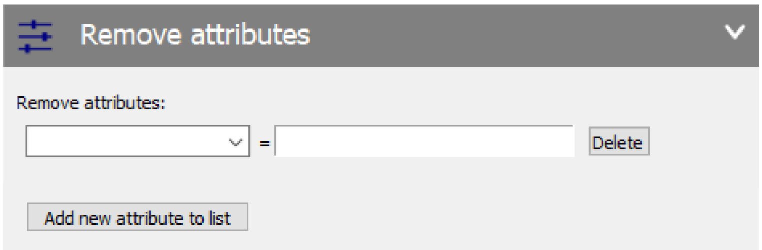
13.5.1 Basic Attributes

Create a list of attributes to be changed or removed. Press button **Add** on bottom of the parameter dialog to append a new item to the list. Press button **Delete** to remove the selected item from the list. For each item the attribute name can be selected and, when adding new attributes, the string value can be defined.

add/modify attributes: Create a new string attribute or modify the value of an existing attribute.

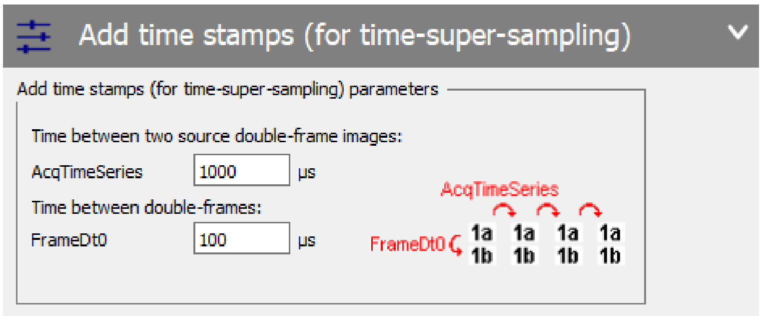


remove attributes: Delete the given attributes from the buffer.



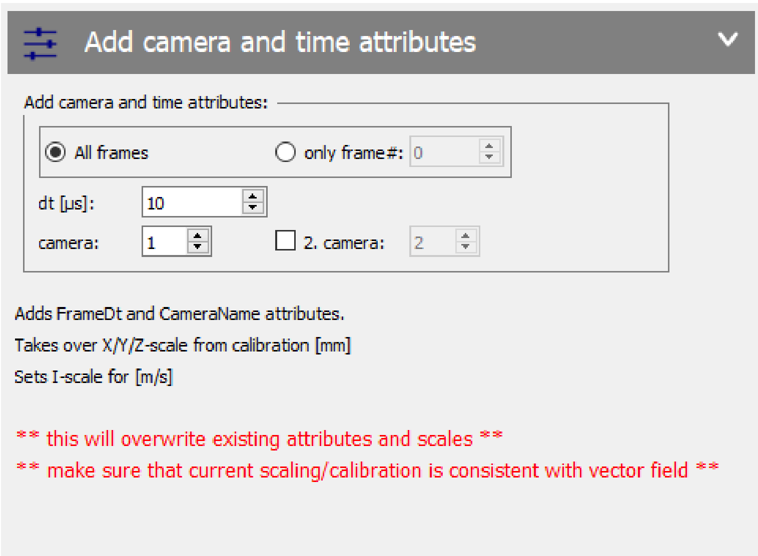
13.5.2 Add time stamps (for time-super-sampling)

This function adds time stamp frame attributes to data by time-super-sampling. In **AcqTimeSeries** enter the time between two source double-frame images. In **FrameDt0** enter the time between double frames.



13.5.3 Add camera and time attributes

This operation can be used to add or modify the following default attributes to the selected data set. Choose between **All frames** and **only frame#** to add camera and time attributes. Specify an acquisition time delay **dt [µm]** and a **camera** name attribute. If a second camera was utilized, select **2. camera** and its respective name attritubte.



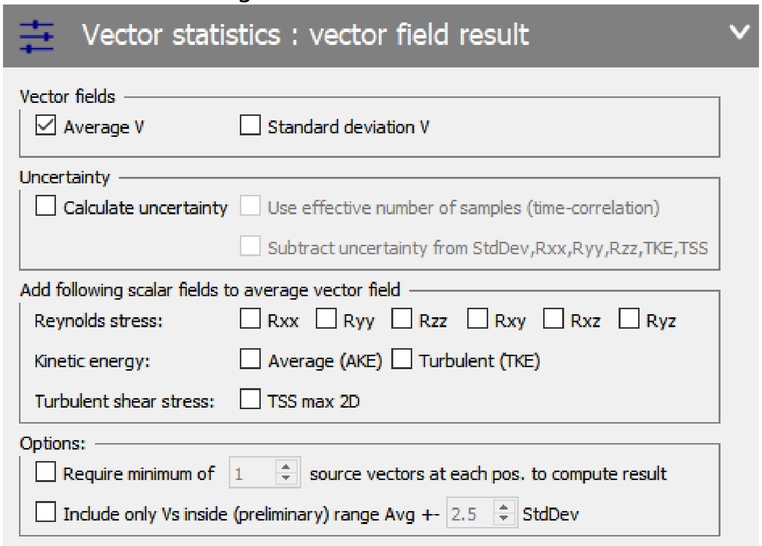
These attributes are very important for many operations in **DaVis** and must be added after importing files without these attributes (e.g. BMP and TIF). The `CameraName` attribute is used to identify the corresponding camera number (needed for calibration, image correction etc.).

This function will overwrite existing attributes and scales.

13.6 Operation: Statistics

13.6.1 Vector statistics: vector field result

This operation includes several functions to calculate statistics on a vector field and store the result again as a vector field.



- **Vector fields:** Here the result vector fields are selected. At least one choice needs to be selected.The average vector field is calculated for each vector position using the formula: $\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ where **N** is the number of available vectors in the input set at that position and $\mathbf{x_i}$ represents the vector components $\mathbf{V_x}$, $\mathbf{V_y}$ and $\mathbf{V_z}$ if available. The standard deviation vector field is a vector field in which each vector component contains the standard deviation σ_x of the corresponding component $x_i = V_x, V_y$ or V_z:

$$var(x) \equiv \sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2. \tag{13.4}$$

  Both vector fields contain a scalar field **Number of vectors** that contain the value **N** at each position. This value can be extracted using the **extract scalar field:vector components** (see Chapter 13.9.1) processing or simply shown as vector background / vector color coding in the vector field display. Please note that all calculations are done on scaled values to allow averaging of vector fields with different intensity scaling (e.g. by using different **dt**). If possible the vector fields are stored as pixel shifts using the intensity scaling of the first vector field with appropriate intensity scaling for the result to look at the average pixel shift as well as the average scaled vector components (typically **m/s** or **mm**).

- **Uncertainty:** If **Calculate Uncertainty** is selected the uncertainty U of the average value and the uncertainty of the standard deviation values σ is calculated using the number of vectors N:
Uncertainty of average:

$$U_{\overline{x}} = \sigma_x \sqrt{\frac{1}{N}} \tag{13.5}$$

Uncertainty of standard deviation:

$$U_{\sigma_x} = \sigma_x \sqrt{\frac{1}{2(N-1)}} \tag{13.6}$$

The switch **effective number of samples** can be selected which uses $\mathbf{N_{eff}}$ instead of the number of available vectors **N** in the above formulas. If the samples are correlated the uncertainties of statistical properties, e.g. mean, standard deviation are larger because the effective number of independent information carriers is smaller. For example, the variance or uncertainty of the mean of time-series data can be computed by

$$U_{\overline{x}}^2 = \sum_{i=T_1}^{T_N} \sum_{j=T_1}^{T_N} \frac{1}{N^2} \rho(x_i, x_j) \sigma_x^2 \tag{13.7}$$

using $\delta\overline{x}/\delta x_i = 1/N$ and $\sigma_{x_i} = \sigma_{x_j} = \sigma_x$ assuming as usual statistically stationary conditions. The autocorrelation coefficient for $j - i = n$, usually quickly decaying to zero for large n, is

$$\rho(x_i, x_j) = \rho(x_i, x_{i+n}) = \rho(n\Delta t) \tag{13.8}$$

It is only a function of $n\Delta t$ where Δt is the time spacing between samples. Note that $\rho(n\Delta t) = \rho(-n\Delta t)$ and $\rho(0) = 1$. Thus

$$U_{\overline{x}}^2 = \frac{\sigma_x^2}{N^2} \sum_{i=T_1}^{T_N} \sum_{n=-\infty}^{+\infty} \rho(n\Delta t) = \frac{\sigma_x^2}{N^2} N \sum_{n=-\infty}^{+\infty} \rho(n\Delta t) \tag{13.9}$$

Defining the effective number of samples N_{eff} as

$$N_{eff} = N / \sum_{n=-\infty}^{+\infty} \rho(n\Delta t) \tag{13.10}$$

leads to

$$U_{\overline{x}}^2 = \frac{\sigma_x^2}{N_{eff}} or\ U_{\overline{x}} = \frac{\sigma_x}{\sqrt{N_{eff}}} \tag{13.11}$$

Typically the summation in eqn. 13.10 is stopped when the correlation values reach zero for the first time. The computation of the effective number of samples is only accurate if one acquires a long enough sequence of data points capturing the longest flow fluctuations several times. Obviously, for uncorrelated data only $\rho(0)$ is 1, all others are zero, so N_{eff} reduces to N and eqn. 13.11 to 13.5. Often one defines an integral time scale as

$$T_{int} = \int_0^\infty \rho(\mathrm{d}t)\,\mathrm{d}t. \tag{13.12}$$

Then

$$N_{eff} = \frac{T}{2T_{int}} \tag{13.13}$$

with total recording time T. This illustrates the fact that, with δt smaller than T_{int} and a fixed recording time T, increasing the sampling frequency and number of data points doesn't improve the accuracy of derived statistical quantities (Smith, 2015, APS; Taylor, 1997). Instead it is advantageous to limit the sampling frequency to $1/T_{int}$ and rather increase the recording time T.

The option **Subtract uncertainty from Stdev, Rxx, Ryy, Rzz, TKE, TSS** corrects the values for Reynolds stress and turbulent kinetic energy if the input vector set has uncertainty values stored for each vector. If the uncertainty of each vector is known its extra deviation from the average vector can be assessed and corrected. The basic definition of the variance of a flow component u, v, or w contains already the sum of the true flow fluctuations and the measurement errors estimated by the average rms-uncertainty, here shown for the u-component:

$$var(u) \equiv \sigma_u^2 = \sigma_{u\_true}^2 + \sigma_{\delta u}^2 \cong \sigma_{u\_true}^2 + {U_u^{rms}}^2 \tag{13.14}$$

assuming that the true flow fluctuations are not correlated to the measurement errors which is usually the case. Note that ${U_u^{rms}}^2 = \overline{U_u^2}$. So following the strategy of correcting the bias one can compute the expected true value of the flow variance by:

$$\sigma_{u\_true}^2 = R_{uu\_true} = \sigma_u^2 - {U_u^{rms}}^2. \tag{13.15}$$

Assuming that the uncertainty fluctuations are mostly uncorrelated to the flow fluctuations leads to the uncertainty of R_{uu_true}

$$U_{R_{\mathrm{true}}} = \sqrt{R_{uu}^2 + \left(\sqrt{2}\sigma_{U_u}\overline{U_u} \cdot \sqrt{1 + \frac{\sigma_{U_u}^2}{2\overline{U_u}^2}}\right)^2} \cdot \sqrt{\frac{2}{N}} \tag{13.16}$$

- **Add following scalar fields to average vector field:** Additional scalar values can be calculated for each vector position that are stored as scalar fields to the average vector field only:
  - **Reynolds stress Rxx, Ryy, Rzz, Rxy, Rxz, Ryz**

$$R_{ij} = \overline{V_i' V_j'} = \frac{1}{N-1} \sum_{i=1}^{N} (V_{i,n} - \overline{V_i})(V_{j,n} - \overline{V_j}) \tag{13.17}$$

    for $i, j = x, y, z$
  - **Kinetic Energy AKE and TKE**
    AKE = $\frac{1}{2}|\overline{V}|^2$
    TKE = $\frac{1}{2}(R_{xx} + R_{yy} + R_{zz})$ for 3-component vectors
    TKE = $\frac{3}{4}(R_{xx} + R_{yy})$ for 2-component vectors
    In the planar PIV case it is assumed that for the turbulent kinetic energy the invisible 3^{rd} component $\mathbf{V_z}$ contains a similar turbulent contribution as the values calculated for $\mathbf{V_x}$ and $\mathbf{V_y}$ as the distribution of turbulence is isotropic. So the formula is corrected for this and leads to a value that is 50% higher.
  - **Turbulent shear stress**
    $\sqrt{\frac{1}{4}(R_{yy} - R_{xx})^2 + R_{xy}^2}$
- **Options:** The option **Require minimum of N source vectors at each pos. to compute result** only stores a result vector at any position, if more than specified **N** valid vectors were found in the input vector set at the corresponding position. This way the result vector calculation can be restricted to positions with enough statistics.
  The option **Include only Vs inside (preliminary) range Avg +- N Standard deviation** calculates the average and standard deviation using all available vectors in a first pass and then recalculates both values using only vectors that fall within the range of **Avg** $\pm$ **N** $*$ **standard deviation** This way outliers (spurious vectors) far outside the average vector are not taken into account. This strategy is also known as sigma clipping. Please note that if you are using this option, the calculation time doubles.

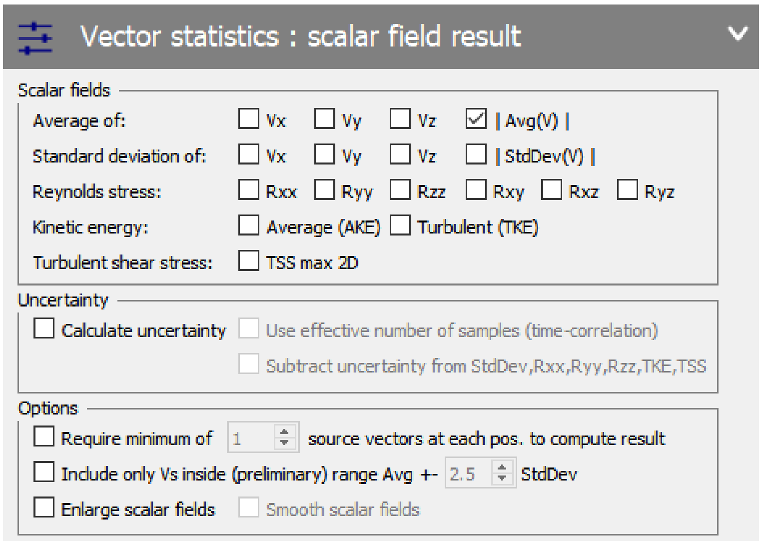
References

Wieneke B, Sciacchitano A (2015): 'PIV Uncertainty Propagation', 11^{th} International Symposium on Particle Image Velocimetry – PIV'15, Santa Barbara

Smith B, Neal D (2015): 'How To Efficiently Sample Data For Computation Of Statistics', Bulletin of the American Physical Society, 68^{th} APS-DFD, Boston

13.6.2 Vector statistics: scalar field result

This operation calculates statistics on a vector field and store the result as scalar fields. Each scalar field may contain an internal scalar field **Uncertainty** if calculation was selected. In this chapter only the formulas are given. For a more detailed derivation of all formulas (e.g. about effective number of samples) please refer to **section 1**.



The parameter settings are divided in three groups:

- **Scalar fields:** The magnitudes that can be derived are
  - Average V_x, V_y, V_z using the formula

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad (13.18)$$

  - $|Avg(V)| = \sqrt{V_x^2 + V_y^2 + V_z^2}$
  - Standard deviation $\sigma_x = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2}$
  - **Reynolds stress Rxx, Ryy, Rzz, Rxy, Rxz, Ryz**

$$R_{uv} = cov(u,v) = \frac{1}{N-1}\sum_{i=1}^{N} u_i^{'} v_i^{'} = \frac{1}{N-1}\sum_{i=1}^{N}(u_i - \overline{u})(v_i - \overline{v}) = \rho_{uv}\sigma_u\sigma_v \qquad (13.19)$$

    with u,v one of x,y,z
  - **Kinetic Energy AKE and TKE**
    AKE $= \sum_{i=1}^{N}\frac{1}{2}V_i^2$

TKE = $\sum_{i=1}^{3} \frac{1}{2}(V_i - \overline{V}_i)^2$for 3-component vectors
TKE = $\sum_{i=1}^{2} \frac{3}{4}(V_i - \overline{V}_i)^2$for 2-component vectors
In the planar PIV case it is assumed that for the turbulent kinetic energy the invisible 3rd component $\mathbf{V_z}$ contains a similar turbulent contribution as the values calculated for $\mathbf{V_x}$ and $\mathbf{V_y}$ as the distribution of turbulence is isotropic. So the formula is corrected for this and leads to a value that is 50% higher.

– **Turbulent shear stress**
$\sqrt{\frac{1}{4}(Rey_{yy} - Rey_{xx})^2 + (Rey_{xy})^2}$

where **N** is the number of available vectors in the input set at that position and $\mathbf{x_i}$ represents the vector components $\mathbf{V_x}, \mathbf{V_y} and \mathbf{V_z}$ if available

- **Uncertainty:** If **Calculate Uncertainty** is selected the uncertainty U of the selected components is calculated using the number of vectors N:
Uncertainty of average:

$$U_{\overline{x}} = \sigma_x \sqrt{\frac{1}{N}} \tag{13.20}$$

Uncertainty of standard deviation:

$$U_{\sigma_x} = \sigma_x \sqrt{\frac{1}{2(N-1)}} \tag{13.21}$$

Uncertainty of Reynolds stress:

$$U_{R_{uv}} = \sigma_u \sigma_v \sqrt{\frac{1+\rho_{u,v}^2}{N-1}} \tag{13.22}$$

Uncertainty of AKE:

$$U_{AKE} = V \times \sigma_{|V|} \times \sqrt{\frac{1}{N}} \tag{13.23}$$

Uncertainty of turbulent shear stress:

$$U_{TSS} = \frac{1}{TSS} \sqrt{\frac{1}{16}(R_{xx} - R_{yy})^2 \times ((U_{R_{xx}})^2 + (U_{R_{yy}})^2) + R_{xy}^2 \times (U_{R_{xy}})^2} \tag{13.24}$$

Optionally **use effective number of samples** can be selected which uses $\mathbf{N_{eff}}$ instead of the number of available vectors **N** in the above formulas. If the samples are correlated the uncertainties of statistical properties, e.g. mean, standard deviation are larger because the effective number of independent information carriers is smaller. For a

more detailed derivation refer to **section Uncertainty under**13.6.
The option **Subtract uncertainty from Stdev, Rxx, Ryy, Rzz, TKE, TSS** corrects the values for Reynolds stress and turbulent kinetic energy if the input vector set has uncertainty values stored for each vector. If the uncertainty of each vector is known its extra deviation from the average vector can be assessed and corrected. For a more detailed derivation refer to **section Uncertainty under**13.6.

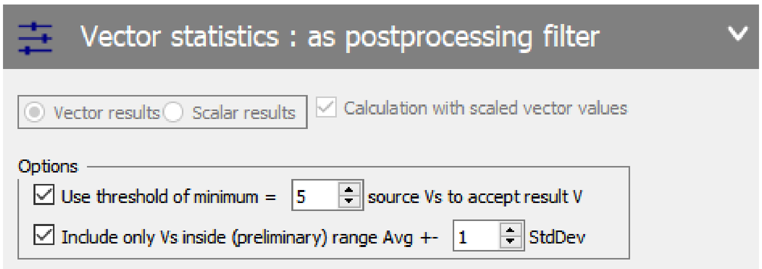
- **Options:** The option **Require minimum of N source vectors at each pos. to compute result** only stores a result vector at any position, if more than specified **N** valid vectors were found in the input vector set at the corresponding position. This way the result vector calculation can be restricted to positions with enough statistics.
  The option **Include only Vs inside (preliminary) range Avg +- N Standard deviation** calculates the average and standard deviation using all available vectors in a first pass and then recalculates both values using only vectors that fall within the range of **Avg** $\pm$ **N** $*$ **standard deviation**. This way outliers (spurious vectors) far outside the average vector are not taken into account. This strategy is also known as sigma clipping. Please note that if you are using this option the calculation time doubles.
  The Option **Enlarge Scalar field** will enlarge the scalar field results by a factor of the grid size of the input vector fields to fit the original / corrected image rather than storing one value per vector. The result is equivalent to what is displayed in the vector background.
  Optionally you can **Smooth the scalar field** over one vector grid when it is enlarged by bilinear interpolation.

13.6.3 Subtract avg vector field

Calculate the averaged vector field from all source vector fields in the first step and then subtract them from all source vector fields in a second step.

13.6.4 Vector statistics: as postprocessing filter

The AVG/stdev statistic is used to disable or to throw out spurious vectors from vector fields in a set.



- **Use threshold of minimum = N source Vs to accept result V:** When the flag is on a result vectors are created only in those places where at least **N** vector fields contributed with a valid vector.
- **Include only Vs inside (preliminary) AVG +- *N* stdev:** In this case the set of vector fields is scanned twice. In the first pass a preliminary average / stdev vector is calculated at each position using the vectors from all vector fields. In the second pass only vectors are used that fall within the range of $V_{avg} - N \cdot V_{stdev}$ to $V_{avg} + N \cdot V_{stdev}$ (componentwise) to calculate the final result. This way outliers (spurious vectors) far outside the average vector are taken out. Please note that if you are using this option the calculation time doubles.

13.6.5 Vector statistics: space-time-correlation

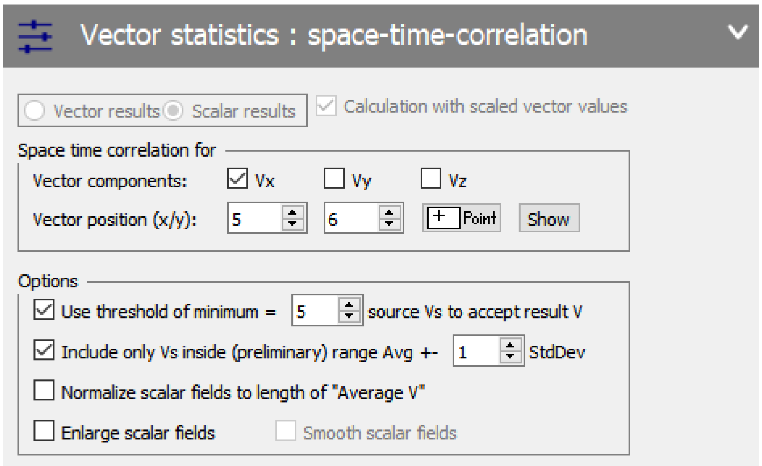
For a specified vector position a scalar map is calculated holding the correlation value of all the vectors (at the corresponding vector position) to that specified vector position.

The formula for the component i of the vector $V(x,y)$ and the specified position (x_o/y_o) is:

$$Corr(V(x,y)_i) = \frac{\frac{1}{N}\sum_{n=1}^{N} V(x,y)_i \times V(x_o,y_o)_i}{stdev(x,y) \times stdev(x_o,y_o)}$$

which is properly normalized so that correlation values fall between -1 (anti correlation = negative movement), 0 = noise (no correlation) and 1 = identical vectors (this is always the case for the specified position). Values between 0 and 1 show the degree of similarity in the behavior of each position (changes of the vector over time) to the specified position.

You can select for which vector components this is done (see following parameter dialog). The specified vector position can be shown on the processing dialog source image if you press the **Show** button. Pressing the button **+ Point** let you select the vector position a mouse click into the vector field in the source window. Alternatively you may adjust the vector grid position using the spin buttons to the side of it directly or enter numbers manually.



The other settings are the same as for the calculation of the average and stdev vector field (see above).

13.6.6 Power spectrum on complete vector field

For a given point (x_o/y_o) the power spectrum is calculated by:

$$\Psi(\omega) = \frac{1}{2\pi} F(\omega) \times F^*(\omega)$$

where $F(\omega)$ is the discrete-time Fourier transform of the series of values $V(x_o, y_o)_i$ over time.

This operation calculates the power spectrum as a scalar map showing the coefficients of the Fourier series at each corresponding position for all frequencies.

In the top group you specify the source of the calculation. It can either be a multi frame vector buffer or a series of single frame buffers. Furthermore you select a vector component or the vector length used for the calculation.

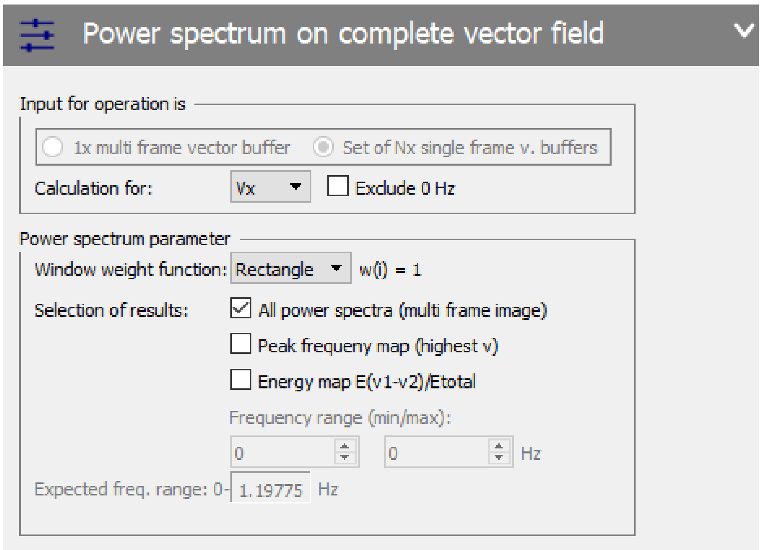
In the group below you set up parameters for the calculation. The window weight function is an additional weighting in the convolution of the data to avoid boundary effects at the beginning / end of the data.

Below that you select what kind of results you want to have. Available results are

- **all power spectra**: this will create a multi frame image with the coefficients of the corresponding frequency terms for each position

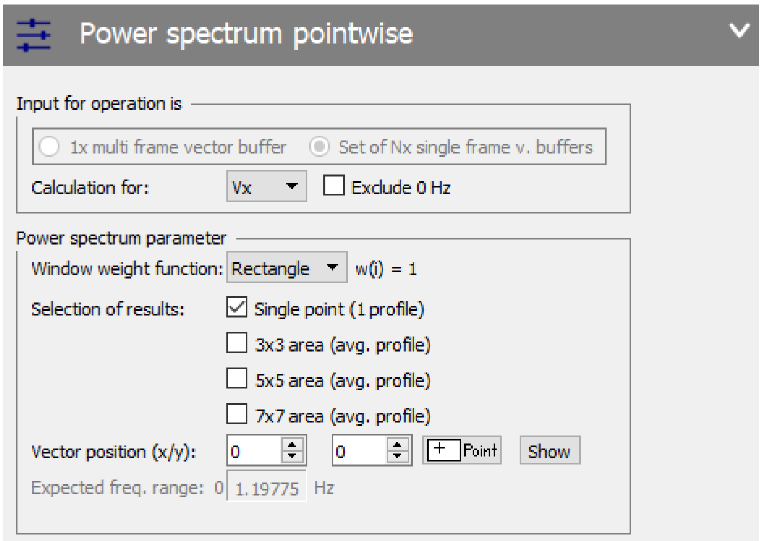
- **peak frequency map**: holds the frequency that had the maximum coefficient at the corresponding position.

- **Energy map (**$E(v1 - v2)/Etotal$: holds the fraction of the energy present in the coefficient of the frequency range $v1$ to $v2$ compared to the total energy.

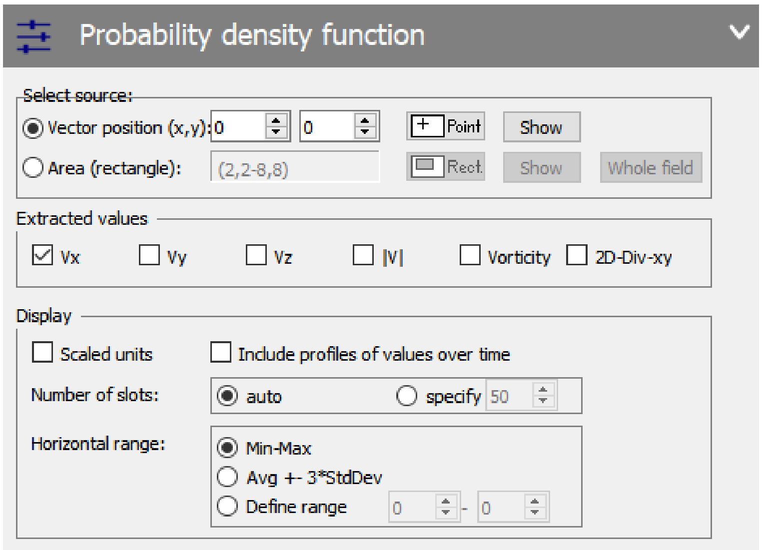


13.6.7 Power spectrum pointwise

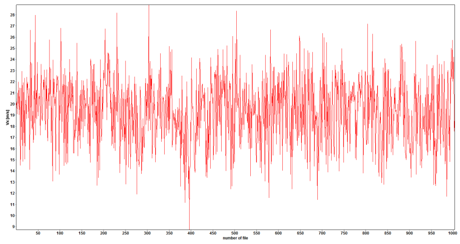
The parameters are basically the same as above but the calculation is done only for one point. The corresponding coefficients are shown in plot versus frequency in *Hz*. To suppress noise level additionally several profiles are calculated in an area of 3x3 (and/or 5x5 and/or 7x7) and then averaged to get a less noisy averaged result from the local neighborhood.



13.6.8 Probability density function

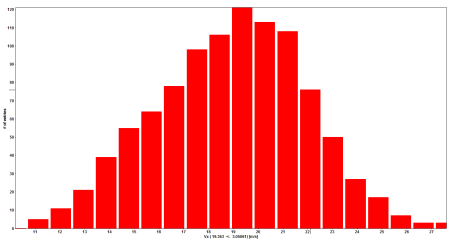


For a given vector position, which can be specified by entering a position in the text boxes or with a left mouse click, the selected values are extracted for all vector fields in the set.



The profile of the magnitude over time can be stored additionally for later analysis. To do so activate the **include profiles of values over time** option.

In any case a histogram is calculated showing the frequency of the selected values in the time series (number of entries in a certain interval). In addition the mean value and the standard deviation around the mean value is shown.



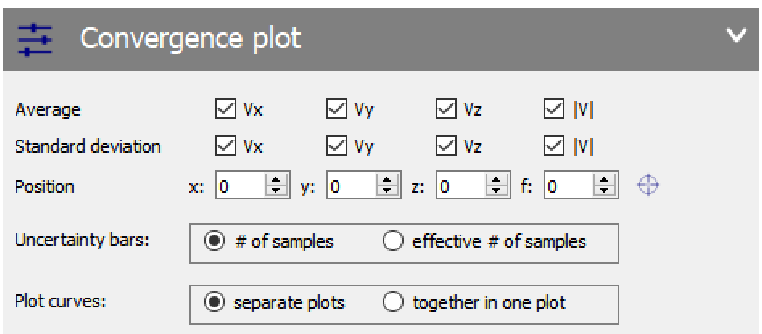
The **scaled units** checkbox allows to extract raw values e.g. pixel shifts or vorticity as gradients of the pixel shift vector field or as scaled values (e.g. in **m/s** or vorticity in **1/s** units). The **Number of slots** of the probability density histogram may be changed from **auto** (which calculates a default number of slots depending on the number of histogram entries) to a user defined number slots.

13.6.9 Add confidence interval from uncertainty

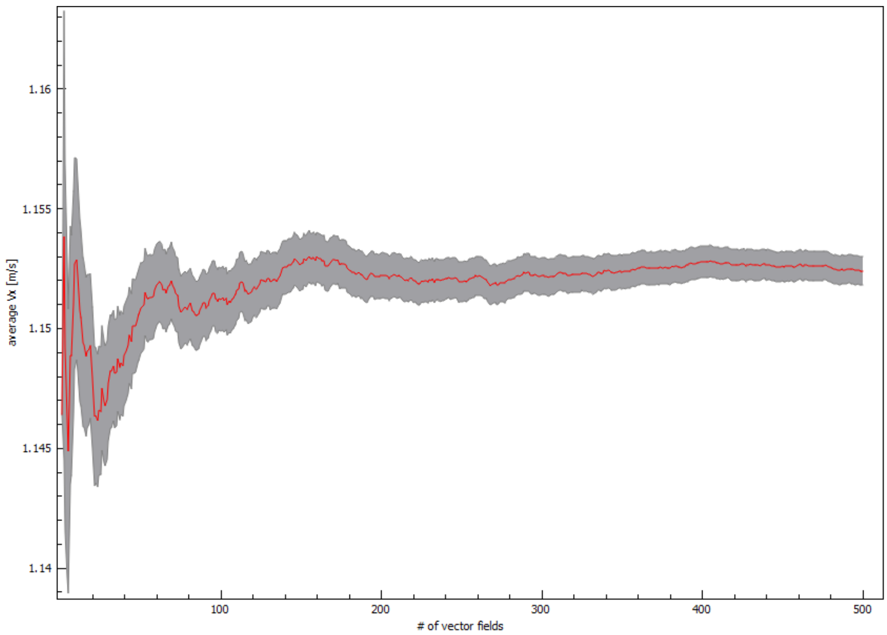
Please refer to Section 13.6.9 for detailed information on this function.

13.6.10 Convergence plot

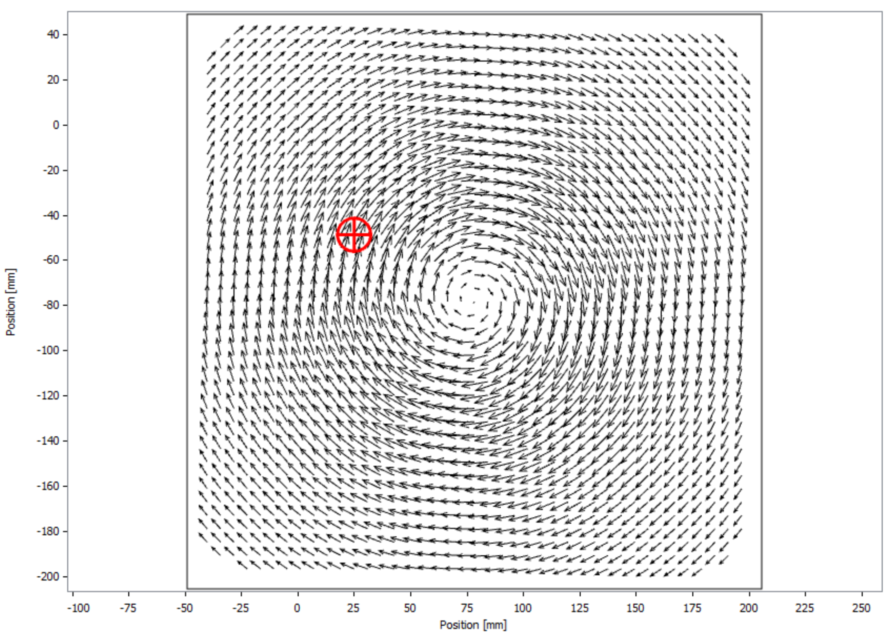
This vector processing shows the convergence of the average velocity and the standard deviation of the velocity and their uncertainty for a given position.



It will plot the selected average / standard deviation of single components or for vector length against number of vectors used showing the resulting uncertainty as uncertainty bars.

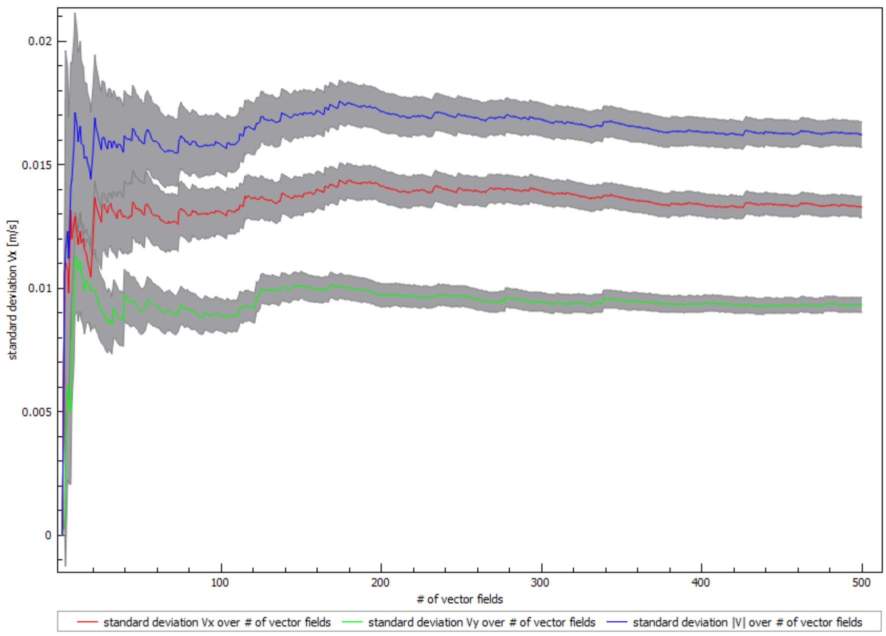


Select the position by pressing and selecting a vector with the mouse in the source window or drag the existing position to a new place (drag and drop)



Statistically the uncertainty of the average goes down by a factor of $1/\sqrt{N}$ when using N vector fields that are statistically independent. Using this operation allows to assess the optimal number of files to achieve the desired maximum uncertainty of the average / standard deviation at the selected position. Furthermore the convergence of the average value itself can be checked to ensure that not too few files have been used.

If the recording rate has been set too high, successive vector fields are no longer statistically independent and become correlated. In this case the effective number of samples N_{eff} that should be used is smaller than the number of files N_{total}. N_{eff} can be estimated using the technique described in the chapter "vector statistics : vector result". To use this method check the "effective # of files" option. This will first determine the effective total number of samples $N_{\text{eff;total}}$ using all Ntotal vector fields in the set and then calculate $N_{\text{eff}}(N) = N \cdot N_{\text{eff;total}}/N_{\text{total}}$ which is used to calculate the uncertainty of the average and standard deviation when using N vector fields (see formulas 13.20 and 13.21 in Chapter 13.6.1 Vector statistics: vector field result). Setting the option "use separate plots" all selected plots are stored separately (A set of 1-8 single result plots) which is most useful, if the value range of the average values / standard deviation is very different. Setting the option to "together in one plot" all selected plot are stored in one graph with different colors which makes it easier to compare all components at the same time. This is especially useful if the value range is very similar for all of them (see screenshot below).



13.7 Operation: Mask

Working with vector fields also might require masking out values (without deleting them !) for display purposes or subsequent operations that respects masked vectors.

13.7.1 Extract mask as binary image

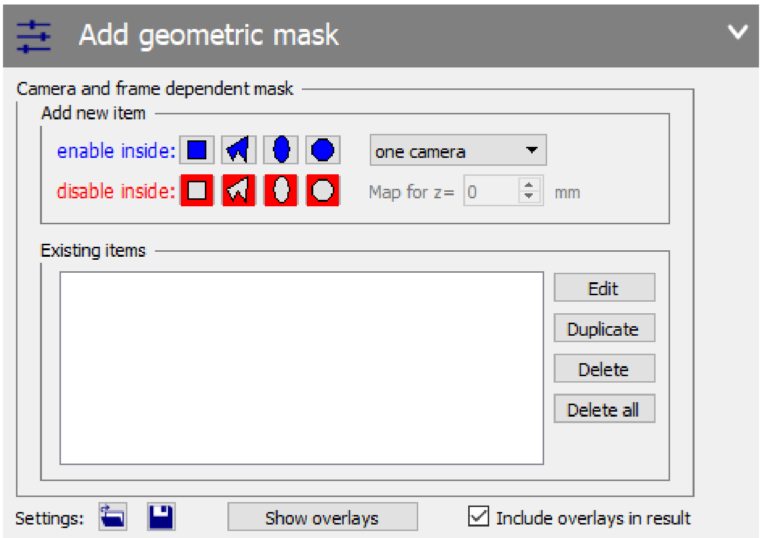
Extract mask from source buffer as new image buffer. A pixel will be set to value 1, if the mask position is enabled, or 0 if disabled.

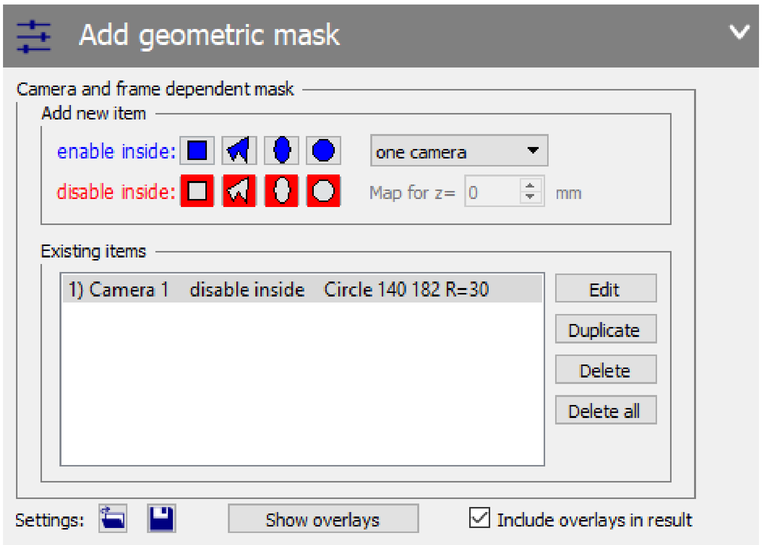
13.7.2 Delete mask

Also this operation is equivalent to the corresponding function for images. It will delete the mask and restore the original vector component values of all previously disabled vectors.

13.7.3 Add geometric mask

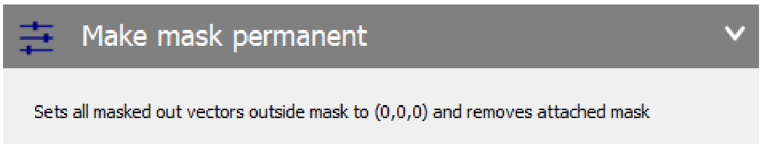
This operation is equivalent to the **geometric mask** mask function for images. Please refer for a detailed description to section 12.10.4.





13.7.4 Make mask permanent (delete all disabled vectors)

Also this operation is equivalent to the corresponding function for images. It will delete the mask and instead replaces the vector component values of all previously disabled vectors by a (0,0) shift or (0,0,0) in case of 3D vectors.

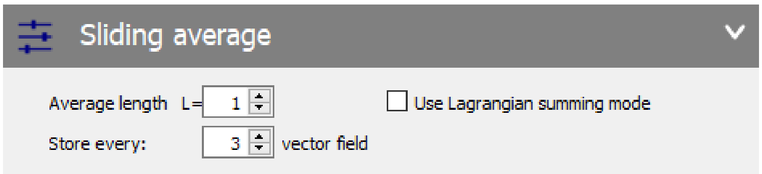


13.8 Operation: Time Series

13.8.1 Sliding average

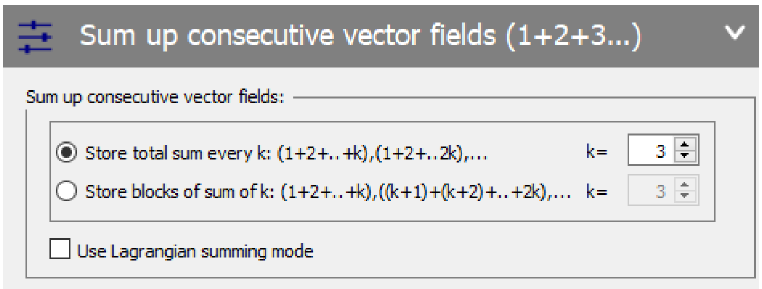
Calculates for each vector position the sliding average over n consecutive vector fields. The **average length** gives the number of consecutive vector fields which will be used for the sliding average calculation. With **store every** the number of result vector fields can be reduced.

Without **Lagrangian summing** vectors at the same vector index position are added up. With lagrangian sum, you follow a vector trail in a moving reference frame: A given vector of vector field 1 points to the position where the vector of vector field 2 lies that should be added up. Usually this vector is found by interpolation. This summed up vector (1+2) is pointing to the position of the vector in vector field 3 That is used for adding up etc.



13.8.2 Sum up consecutive vector fields (1+2+3...)

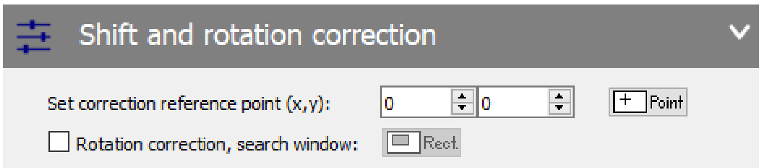
Sums up consecutive **k** number of vector fields. There are following options:
Store total sum every k: (1+2+..+k), (1+2+..+2k),... Here, all previous vector fields are added up. In **Store blocks of sum of k: (1+2+..+k),((k+1)+(k+2)+..+2k),..** the last k vector fields are added up. 'Use Lagrangian summing' mode adds not vectors at the same constant position but builds a track of positions: takes vector from vector field 1 at the grid position, adds the vector of vector field 2 at the position vector 1 has bee pointing to. Then vector 3 is taken from the position the summed up vector is pointing to. This way we follow the flow (Lagrangian reference frame) instead of taking the same laboratory coordinate (Euler reference frame).



13.8.3 Shift and Rotation Correction

Gets the vector at a specified position and subtracts this from the whole vector field. Use this operation to correct a different vector shift between consecutive vector fields. Set a correction reference point (x,y) by specifying the coordinates, or use the **Point** button to select the point directly in the image. If the **Rotation correction** is enabled, the operation corrects the angle between the reference position and the Rotation search window.

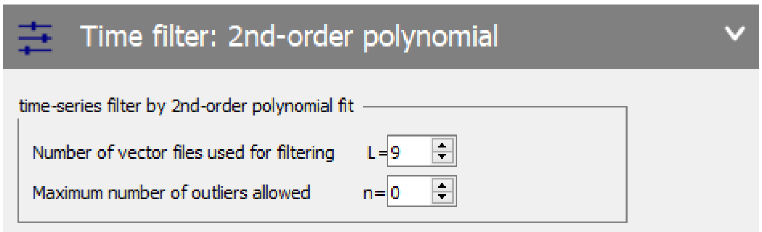
Press button **Rect.** to specify the position of the correction reference point. Enable or disable **rotation correction**. Specify the **search window** for the calculation of the angle between the reference point and this area.



13.8.4 Time Filter: 2nd-order polynomial

This advanced filter helps to remove noise (spurious vectors) from the vector field. It uses the number of vector fields in the set that corresponds to the preceding and subsequent vector fields.

It fits three polynomial of second order (least square fit) that best describe the development of the series of components V_x, V_y and V_z over time. Thus, first and second order changes in the temporal development are maintaining. Then the vector components are replaced by the value of that polynomial at the position in question.



For **Maximum number of outliers allowed > 0** this function works as follows:

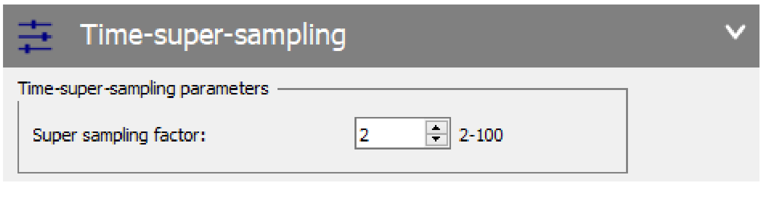
For **ignore up to** N **Outliers,** N**=1** this function works as follows: For the *Number of vector filed used for filtering* N all possible subsets of $N-1$ vector files are formed (in each a different *outlier* is thrown out). The fit is performed as described above for all those subsets. The vector is then taken from the polynomial with the smallest fit error, which is the one with the most irritating vector to the fit left out (a spurious vector if there really was any). This guaranties that the spurious vector is replaced by the polynomial of the surrounding vector fields AND at the same time does not influence badly on its neighbor vector fields in return!

Therefore, for a higher number *N* of outliers the necessary time to calculate the fit of all subsets is starting to grow very, very fast.

The advantage is that while a spurious vector can be replaced by a predicted value considering gradients of second order, it will not be included into the calculation of all neighbors because most likely the best fit for other positions can be classified as an outlier and thus be ignored. In areas where no spurious vectors are present, the gradients (up to 2nd order) will not be disturbed.

13.8.5 Time-super-sampling

Time-super-sampling increases the temporal resolution of PIV vector field SETs that are derived from high speed time series recorded in frame straddling mode. This operation has the parameter **Super sampling factor**, which defines the temporal resolution increase of the source SET. The number of generated intermediate vector fields between two consecutive source vector fields is equal to "super sampling factor-1".



The intermediate vector fields are calculated by using the so-called time-super-sampling (TSS) algorithm. This algorithm was described by Fulvio Scanaro in his paper *An advection model to increase the time-resolution of PIV time-series* of July 2010. The current implementation can be used for both 2C and 3C vector fields in 2D and 3D. The input vector fields should be derived from high speed time series recorded in frame straddling mode. Tomo/3D-vector fields are only supported, if the grid-values in X, Y and Z are equal. The TSS algorithm requires specific time stamps in the vector field buffers: In every case, it is necessary that the "AcqTimeSeries0"-attribute is set correctly. Additionally, the attribute "FrameDt0" must be available. If this attribute is missing, alternatively the X- and I-scales must be defined for frame0. The X-scale must have the unit "mm" and the I-scale must have the unit "m/s". If these requirements on attributes are not fulfilled, a

SET cannot be used with the TSS algorithm. (If necessary, these attributes could be copied to the vector field buffers from their image source buffers.)

This functionality is also available as a **send to**-dialog, see page 413 for details.

13.8.6 Time + space filter

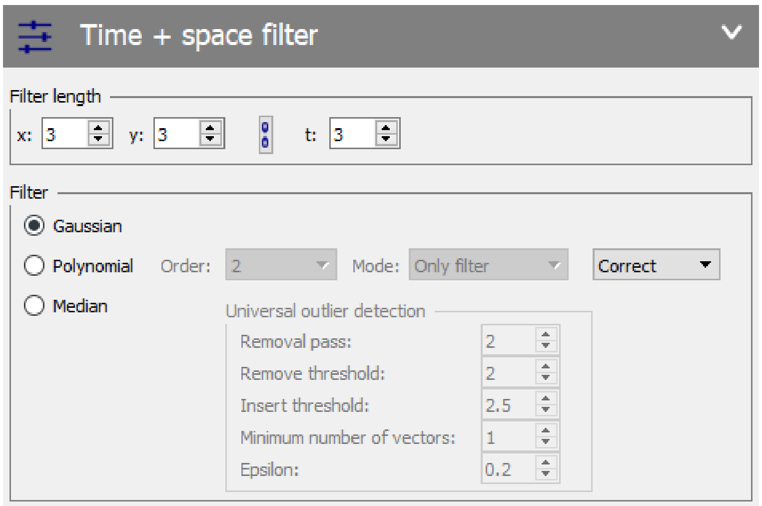
Choose a **Filter** type between **Gaussian**, **Polynomial**, and **Median** for each position fit to use the spacial neighbors as well as adjacent vector fields in time and set a **Filter length**.

Please refer to Section 12.12.1 for further information on the Gaussian smoothing filter.

The Polynomial filter removes high frequency noise from a vector field and fills up disabled vectors. Since the filter is mostly identical to the **polynomial filter** under the group **non-linear filter** please refer to section 13.10.6 for further information.

The main difference is that the filter operates over time, which means it operates on three dimensions for planar vector fields and on four dimensions for tomographic vector fields.

The operation requires that the number of SETs has to be equal or greater to the adjustable parameter **Filter length**.



The **Universal outlier detection** of the **Median filter** is an adaptation of the original median test for the detection of spurious PIV data. This normalizes the median residual with respect to a robust estimate of the local variation of the velocity.

Consider a displacement vector denoted by U_0, its 3×3 neighborhood data, denoted by $\{U_1, U_2, \ldots, U_8\}$ and U_m as the median of $\{U_1, U_2, \ldots, U_8\}$. A residual r_i defined as $r_i = |U_i - U_m|$ is determined for each vector $\{U_i | \, i = 1, \ldots, 8\}$ and the median r_m of $\{r_1, r_2, \ldots, r_8\}$ is used to normalize the residual of U_0:

$$r_0' = \frac{|U_0 - U_m|}{r_m}.$$

This method is quite general in outlier detection and can be used to process a large variety of inhomogeneous data. In fact, for purely uniform flow, the normalization factor r_m tends to zero. This can be compensated by assuming a minimum normalization level ε, i.e.

$$r_0^* = \frac{|U_0 - U_m|}{r_m + \varepsilon},$$

where ε represents the acceptable fluctuation level due to cross correlation. In **DaVis**, ε is internally fixed to 0.1 pixel, which has been found to be a suitable value that corresponds to the typical standard deviation noise level of the PIV data.

References

Westerweel J, Scarano F (2005) “Universal outlier detection for PIV data”, Exp. Fluids, Vol.39, No. 5, pp. 1096-1100.

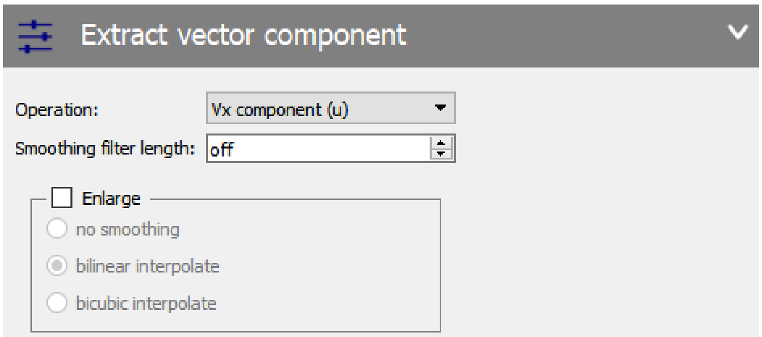
13.9 Operation: Extract Scalar Field

13.9.1 Extract vector component

When a scalar value is extracted from a vector field typically a single value is extracted for each vector. This vector has been calculated on the size of one interrogation window. To have the possibility to display the extracted scalar field in the background of a vector field you may select the **Enlarge** option. Then the scalar field is enlarged with the grid size of the input vector fields to fit the original image size. The enlarged scalar image may be smoothed over a fraction or a multiple of the vector grid selecting **Smoothing filter length**.

Enlarged scalar fields can be smoothed via sliding average with a filter length of 0.5*grid size, 1*grid size or 2*grid size or again bilinear interpolation. Both methods have their (dis)advantages:

- bilinear: constant gradients across background values stay smooth, nonlinear gradients lead to (slight) bends (piecewise linear tiles)
- nonlinear filter: constant gradients across background values produce (slight) stair steps, nonlinear gradients when enlarged lead to nicer and smoother nonlinear slopes



The following **Operations** can be performed:

Vector Length, Vx, Vy or Vz Component

Vx component (u): Extracts the u-component of the velocity vector = V_x.

Vy component (v): Extracts the v-component of the velocity vector = V_y.

Vz component (w): Extracts the w-component of the velocity vector = V_z.

$|V|$ **vector length:** Calculates $\sqrt{V_x^2 + V_y^2}$ or $\sqrt{V_x^2 + V_y^2 + V_z^2}$ for 2C- or 3C-vector fields.

Peak Ratio 1st/2nd or value 1st

Determines the peak ratio for each vector, i.e. the ratio of the correlation value of the highest and the second highest correlation peak – or (depends on the settings when the calculation was done) – the correlation value of the highest peak.

Vector Choice

Extracts the vector choice, i.e. 0 for a disabled vector and 1-4 for the 1st, 2nd, 3rd, 4th vector choice and 5 for a postprocessed vector.

Vector Angle

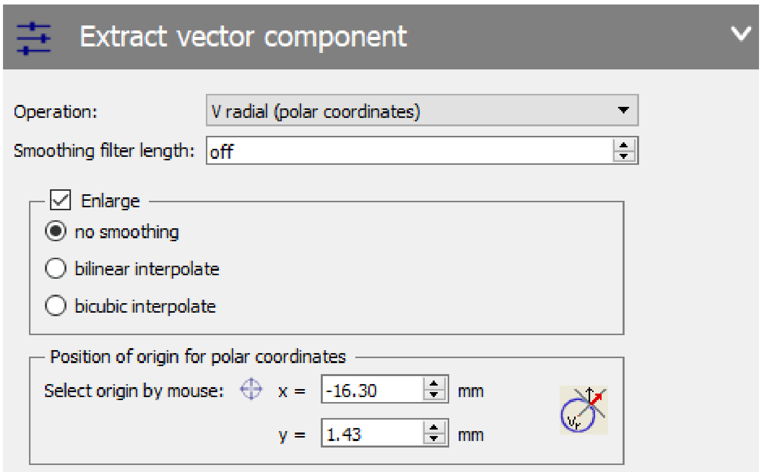
Extracts the angle in respect to the positive *x*-axis (counter clockwise) of each vector as arctan(Vy/Vx).

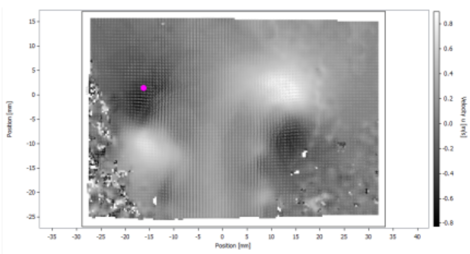
Vradial and Vangle (polar coordinates)

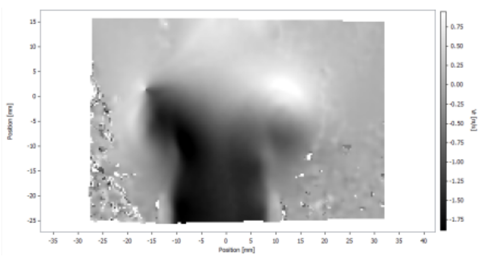
This two **Extraction modes** allow the extraction of scalar fields of V_r (radial component) and V_{angle} (polar angle component) in respect to a user specified origin. The origin can either be set with mouse (press the button labeled **Define origin by mouse**) or defined directly entering the pixel position in the corresponding text fields.

Please refer to the following four images which shows the result:

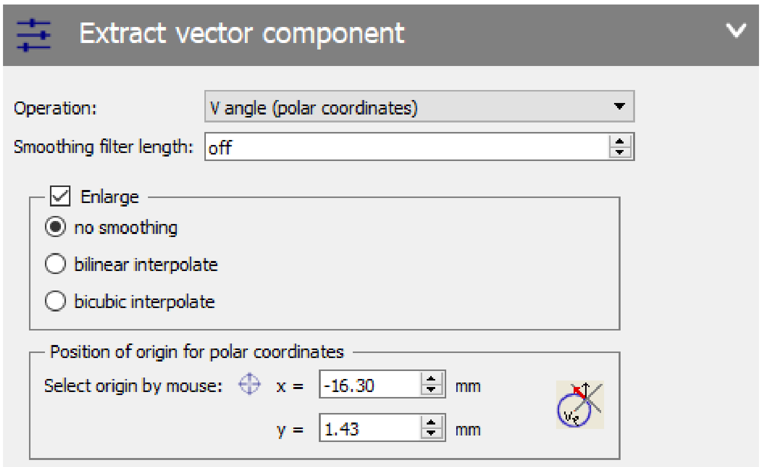
Extraction mode V_r:

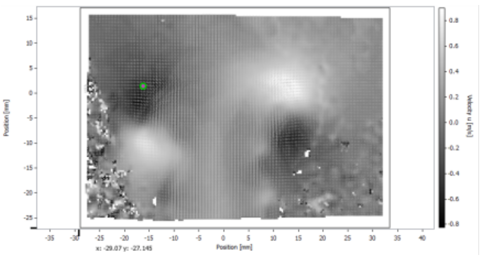


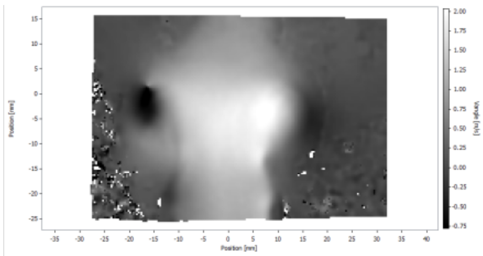




Extraction mode V_{angle}:





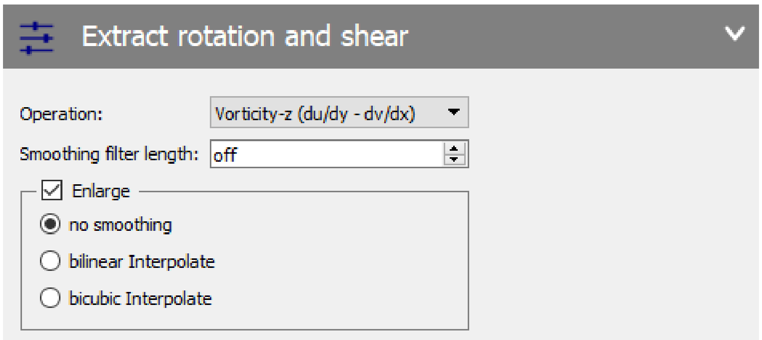


13.9.2 Extract divergence

All magnitudes in this group are based on the strain tensor

$$E_{ij} = \partial V_i / \partial j \quad with \quad i \in \{x, y, z\} \quad and \quad j \in \{x, y, z\}$$

Please refer to section 13.9.4 for a detailed description of how these values are calculated in **DaVis**. Again, the scalar image can be **enlarged** and a **Smoothing filter length** can be applied (see Section 13.9.1 for a detailed description of these parameters).



The following **Operations** can be performed:

du/dx: Calculates the divergence of the u-component in x-plane.

du/dy: Calculates the divergence of the u-component in y-plane.

du/dz: Calculates the divergence of the u-component in z-plane.

dv/dx: Calculates the divergence of the v-component in x-plane.

dv/dy: Calculates the divergence of the v-component in y-plane.

dv/dz: Calculates the divergence of the v-component in z-plane.

dw/dx: Calculates the divergence of the w-component in x-plane.

dw/dy: Calculates the divergence of the w-component in y-plane.

dw/dz: Calculates the divergence of the w-component in z-plane.

2D-Divergence-XY: Calculates the two-dimensional divergence in xy-plane.

-2D-Divergence-XY: Calculates the negative divergence in xy-plane.

2D-Divergence-XZ: Calculates the two-dimensional divergence in xz-plane.

2D-Divergence-YZ: Calculates the two-dimensional divergence in yz-plane.

|2D-Divergence-XY|: Calculates the absolute value of the divergence in the xy-plane.

3D-Divergence-XYZ: Determines the three-dimensional divergence. This option can only be applied to dual frame vector buffer e.g. with **LaVision** Dual Plane Stereo PIV system.

13.9.3 Extract rotation and shear

DaVis calculates the vorticity according to the central difference scheme with the four closest neighbors (see figure 13.1). Compared to other methods the small spatial kernel allows a high spatial resolution. If any sort of spatial smoothing is required, you may use the standard filter for scalar images.

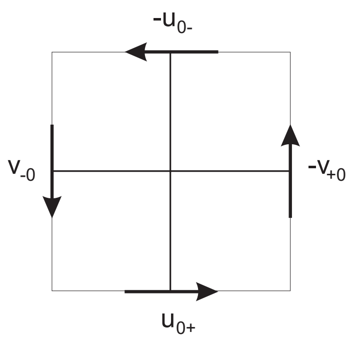
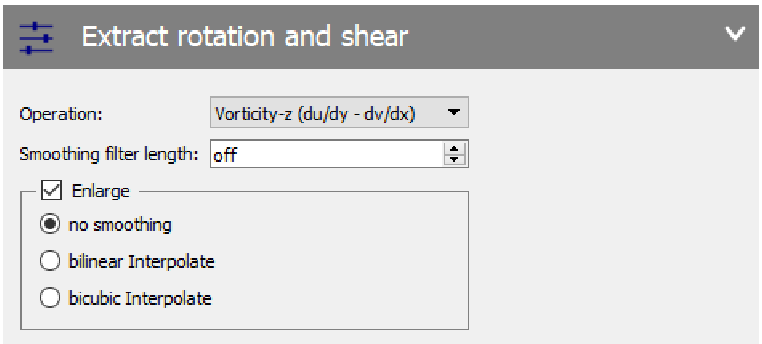


Figure 13.1: Example of vorticity calculation



The following **Operations** can be performed:

Vorticity-z: Determines the two-dimensional vorticity in xy-plane.

-Vorticity-z: Determines the negative two-dimensional vorticity in xy-plane

Vorticity-y: Determines the two-dimensional vorticity in xz-plane.

Vorticity-x: Determines the two-dimensional vorticity in yz-plane.

Swirling Strength: The eigenvalues of the Matrix

$$\begin{pmatrix} E_{xx} & E_{xy} \\ E_{yx} & E_{yy} \end{pmatrix}$$

in 2D have the form $a \pm \sqrt{b}$ with

$$b = E_{xy} \cdot E_{yx} - (E_{xx} \cdot E_{yy})/2 + (E_{xx}^2 + E_{yy}^2)/4$$

If b is positive, it is the shear. If b is negative - and the pair of Eigenvalues is complex - then it is swirl. In **DaVis**, the displayed values are $\max(0, -b)$ for swirling strength and $\max(0, b)$ for shear strength.

Calculates the swirling strength given by

$$max\left(0, -\left(E_{xy}\cdot E_{yx}-(E_{xx}\cdot E_{yy})/2+(E_{xx}^2+E_{yy}^2)/4\right)\right)$$

i.e. only the positive part of the swirling strength λ_{ci}. The rest is set to zero to display swirl (shear) only.

Shear Strength: See description on swirling strength above.

Swirl and Shear: Calculates the swirling- plus shear-strength given by

$$-\left(E_{xy}\cdot E_{yx}-(E_{xx}\cdot E_{yy})/2+(E_{xx}^2+E_{yy}^2)/4\right)$$

i.e. the imaginary portion of the complex eigenvalue of the local velocity gradient tensor. See section on swirling strength above.

3D shear: Calculates the value ('maximum eigenvalue - minimum eigenvalue')/2 of the strain tensor (only works for multi-z vector fields as derived in Tomo-PIV, where E_{zz} exists).

3D Swirl Strength (lambda-2): Calculates the value *lambda-2* in terms of the eigenvalues of the symmetric tensor $S^2+\Omega^2$ where S and Ω are the symmetric and the antisymmetric parts of the velocity gradient tensor ∇V (refer to J.Fluid.Mech. (1995), vol 285, pp69–94 : 'On the Identification of a vortex' by Jinhee Jeong and Fazle Hussain) (only works for multi-z vector fields as derived in Tomo-PIV, where E_{zz} exists).

13.9.4 Extract scalar field: strain

All magnitudes in this group are based on the gradients of the displacement field

$$\partial V_i/\partial j \quad with \quad i\in\{x,y,z\} \quad and \quad j\in\{x,y,z\}$$

where i in the vector component index going along the j-*th* axis.

For example:

- $\partial V_x/\partial x$ is the gradient of V_x along x axis direction and represents a compression or an expansion.
- $\partial V_x/\partial y$ is the change of V_x along the y-axis direction and represents a horizontal shear.

The calculation of the local derivatives is done the following way:

In the local neighbourhood of a vector (defined by the "Smoothing" parameter) a Gaussian weighted (displacements closer to the centre get a higher weight) affine fit is performed on the local displacement field.

The performed fit outputs six coefficients: $\partial V_x/\partial x$, $\partial V_x/\partial y$, $\partial V_y/\partial x$, $\partial V_y/\partial y$, t_x = translation in x, t_y = translation in y, leading to the transformation:

xÂ´ = x * (1+ $\partial V_x/\partial x$) + y * $\partial V_x/\partial y$ + t_x

yÂ´ = x * $\partial V_y/\partial x$ + y * (1+ $\partial V_y/\partial y$) + t_y

The coefficients t_x and t_y will be ignored and the coefficients: $\partial V_x/\partial x$, $\partial V_x/\partial y$, $\partial V_y/\partial x$, and $\partial V_y/\partial y$ describe the local scale and shear.

For stereo data - before applying the local affine - the local displacements are projected onto the surface of the specimen allowing us to compute the strain in the local coordinate system of the specimen. Depending on the strain tensor (see appendix **?? Engineering Strain Tensor**) these four coefficients will be used to compute the symmetric strain tensors $\varepsilon_{xx}, \varepsilon_{xy}, \varepsilon_{yy}$.

which can be represents as a two-by-two matrix:

$$F = \begin{pmatrix} \varepsilon_{xx} & \varepsilon_{xy} \\ \varepsilon_{xy} & \varepsilon_{yy} \end{pmatrix}$$

Many quantities are calculated from these components to give new quantities.

Available operations are:

Strain $\varepsilon_{xx}, \varepsilon_{xy}, \varepsilon_{xz}$: Calculates $\varepsilon_{xx}, \varepsilon_{xy}, \varepsilon_{xz}$

Strain $\varepsilon_{yx}, \varepsilon_{yy}, \varepsilon_{yz}$: Calculates $\varepsilon_{yx}, \varepsilon_{yy}, \varepsilon_{yz}$

Strain $\varepsilon_{zx}, \varepsilon_{zy}, \varepsilon_{zz}$: Calculates $\varepsilon_{zx}, \varepsilon_{zy}, \varepsilon_{zz}$

Maximum/Minimum Normal Strain, Maximum Shear Strain 2D: Normal strain (sometimes also called principal strain) are the eigenvalues of the matrix.

The eigenvectors $\vec{e_x}$ and $\vec{e_y}$ of that matrix define the coordinate system in which no shear is present and the deformation can be described by using only strain along $\vec{e_x}$ and $\vec{e_y}$. The greater eigenvalue is called *maximum normal strain*, the smaller *minimum normal strain*. The maximum shear strain is the shear in a coordinate system rotated by 45° to the coordinate system defined by the eigenvalues where the residual shear becomes maximal.

To see the formulas of the following modes more easily we will introduce two abbreviations:

$$d_1 = \frac{(\varepsilon_{xx} - \varepsilon_{yy})}{2}$$
$$d_2 = \sqrt{(\varepsilon_{xx} - \varepsilon_{yy})^2/4 + {\varepsilon_{xy}}^2}$$

Maximum normal strain 2D: $d_1 + d_2$;

Minimum normal strain 2D: $d_1 - d_2$;

Maximum shear strain 2D: d_2

Median normal strain: calculates the median eigenvalue of the normal strain

Maximum Strain Angle: θ_p=orientation of the maximum eigenvalue of F^tF

Maximum Shear Angle: θ_s=orientation of the maximum eigenvalue of $F^tF + 45°$

Poisson's Ratio : -strain max/strain min: Calculates the Poisson's ratio -$(d_1 + d_2)/(d_1 - d_2)$

Poisson's Ratio : -$\varepsilon_{yy}/\varepsilon_{xx}$: Calculates the Poisson's ratio $-\varepsilon_{yy}/\varepsilon_{xx}$

Poisson's Ratio : -strain min/strain max: Calculates the Poisson's ratio –$(d_1 - d_2)/(d_1 + d_2)$

Poisson's Ratio: $-0.5 * (\varepsilon_{xx} + \varepsilon_{yy})/\varepsilon_{zz}$**:**

Calculates the Poisson ratio $-0.5 * (\varepsilon_{xx} + \varepsilon_{yy})/\varepsilon_{zz}$

Poisson's Ratio: $-0.5 * (\varepsilon_{zz} + \varepsilon_{xx})/\varepsilon_{yy}$**:**

Calculates the Poisson ratio $-0.5 * (\varepsilon_{zz} + \varepsilon_{xx})/\varepsilon_{yy}$

Poisson's Ratio: $-0.5 * (\varepsilon_{zz} + \varepsilon_{yy})/\varepsilon_{xx}$**:**

Calculates the Poisson ratio $-0.5 * (\varepsilon_{zz} + \varepsilon_{yy})/\varepsilon_{xx}$

Strain EZZ estimation (plane stress assumption):

Another possible assumption is that we have a ***plane stress state σ_{zz} = 0 in a homogeneous and isotropic material, which is valid for flat and thin test objects*** (x and y dimension much larger than the thickness, no curvature or at least large curvature radii). This means that forces act on the material only in the x-y plane and there is no out-of-plane load component. Therefore, there are also no z-shear components ($\varepsilon_{xz} = \varepsilon_{yz} = 0$), but there will still be an out-of-plane strain component ($\varepsilon_{zz} \neq 0$). Consider a tensile test where the material is pulled along the y-axis and consequently shrinks in x- and z-direction. Therefore, under plane stress assumption, the stress tensor is

$$\underline{\underline{\sigma}} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{bmatrix} \tag{13.25}$$

and the corresponding strain tensor is

$$\underline{\underline{\varepsilon}} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & 0 \\ \varepsilon_{yx} & \varepsilon_{yy} & 0 \\ 0 & 0 & \varepsilon_{zz} \end{bmatrix}. \tag{13.26}$$

For isotropic materials, the out-of-plane strain can be estimated via Poisson's ratio ν (defined in Eq. 13.27):

$$\varepsilon_{zz} = \frac{\nu}{1-\nu}(\varepsilon_{xx} + \varepsilon_{yy}). \tag{13.27}$$

Taking all of this into account, the von Mises strain under plane stress condition becomes:

$$\varepsilon_{eq} = \frac{2}{3}\sqrt{\varepsilon_{xx}^2 + \varepsilon_{yy}^2 + \varepsilon_{zz}^2 - \varepsilon_{xx}\varepsilon_{yy} - \varepsilon_{xx}\varepsilon_{zz} - \varepsilon_{yy}\varepsilon_{zz} + 3(\varepsilon_{xy})^2} \tag{13.28}$$

Equation 13.28 is uses by DaVis or StrainMaster DIC for strain calculation in the plane stress case.

Tresca strain: Tresca strain is related to the Tresca yield criterion, which assumes that yielding is due to a maximum shear stress. Just like the von Mises criterion it is mainly used to ***describe the mechanical behavior of isotropic ductile materials.*** The Tresca criterion is designed for materials whose plasticity and failure mechanics are driven by shear stresses. Therefore, ***the applicable material range for the Tresca criterion is narrower than for the von Mises criterion.***

In strain terms, the Tresca equivalent strain ε_T is defined as the difference between the maximum and minimum principal strains.

If ε_1 is the maximum principal strain and ε_3 the minimum principal strain, the Tresca strain is given by:

$$\varepsilon_T = \frac{2}{3}\left|\varepsilon_1 - \varepsilon_3\right| \tag{13.29}$$

Note: For a general three-dimensional strain state, there is also a medium principal strain ε_2, which is not relevant for the computation of the Tresca strain. But, because of the three-dimensional nature of the problem, the minimum and maximum pricipal strains are not confined to the x-y-plane.

Note 2: There are different definitions of the Tresca strain in the common literature, but the only difference is the multiplication factor. Instead of the $\frac{2}{3}$ used here, other definitions use 0.5 or 1. These factors are a consequence of the fact that the actual Tresca criterion is defined in stress space and the corresponding strain equations are prone to specific assumptions.

Von Mises strain: The equivalent von Mises strain condenses the three-dimensional strain state at any given point into an effective scalar strain value which is equivalent to the strain of a uniaxial (tensile) test. The idea behind this is that the computed von Mises strain can be compared with material properties such as failure strain or the strain at the ultimate tensile strength which have been determined by standardized uniaxial tensile testing. Together with the associated von Mises stresses, these equivalent values allow safety assessments for engineered parts making sure that internal stresses and strains never exceed the stability limits of the material. Since the von Mises strain gives the effective strain equivalent to a uniaxial load state, it is sometimes also called the ***effective strain***.

Note: von Mises stress and strain are related to the von Mises yield criterion, which is ***only valid for isotropic ductile materials***, such as most metals. Consequently, von Mises strain has little meaning when dealing with brittle materials (ceramics, glass) or non-isotropic materials (fibre composites, wood, layered laminates). The von Mises yield criterion is a maximum distortion energy criterion which assumes that the total strain energy can be separated into two components: the volumetric (hydrostatic) strain energy and the shape (distortion or shear) strain energy. It is proposed that yield occurs when the distortion strain energy exceeds the value measured for the material at the yield point of a simple tensile test.

The following fomulas show how von Mises strain is computed and what the actual implementation in **DaVis** or **StrainMaster DIC** is doing.

The general three-dimensional strain tensor is

$$\underline{\underline{\varepsilon}} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{bmatrix} \tag{13.30}$$

where the ε_{ij}-components have been computed using any of the available strain tensors in **DaVis** (Engineering, Cauchy, Green-Lagrange, Almansi or Hencky). The von Mises equivalent strain ε_{eq} is defined as:

$$\varepsilon_{eq} = \frac{2}{3}\sqrt{\frac{3}{2}(e_{xx}^2 + e_{yy}^2 + e_{zz}^2) + 3(\varepsilon_{xy}^2 + \varepsilon_{xz}^2 + \varepsilon_{yz}^2)} \tag{13.31}$$

where e_{ij} are the deviatoric strains which can be computed from the components of the strain tensor according to:

$$e_{xx} = +\frac{2}{3}\varepsilon_{xx} - \frac{1}{3}\varepsilon_{yy} - \frac{1}{3}\varepsilon_{zz} \tag{13.32}$$

$$e_{yy} = -\frac{1}{3}\varepsilon_{xx} + \frac{2}{3}\varepsilon_{yy} - \frac{1}{3}\varepsilon_{zz} \tag{13.33}$$

$$e_{zz} = -\frac{1}{3}\varepsilon_{xx} - \frac{1}{3}\varepsilon_{yy} + \frac{2}{3}\varepsilon_{zz}. \tag{13.34}$$

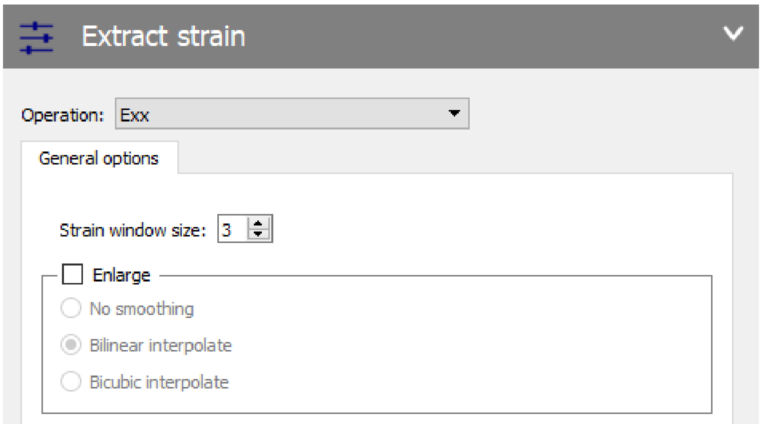
Using equations (13.32)-(13.34), the first part of (13.31) becomes

$$\frac{3}{2}(e_{xx}^2 + e_{yy}^2 + e_{zz}^2) = \varepsilon_{xx}^2 + \varepsilon_{yy}^2 + \varepsilon_{zz}^2 - \varepsilon_{xx}\varepsilon_{yy} - \varepsilon_{xx}\varepsilon_{zz} - \varepsilon_{yy}\varepsilon_{zz} \tag{13.35}$$

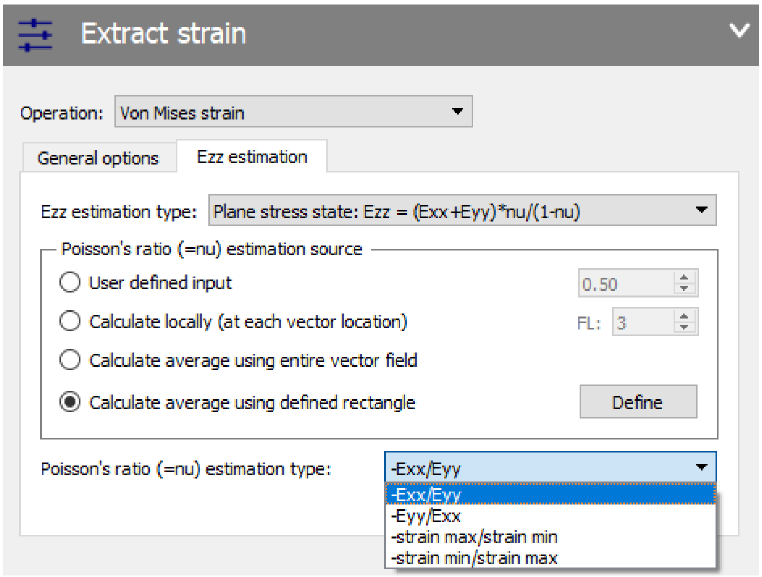
and thus, we can finally express the von Mises strain as a function of the strain tensor components:

$$\varepsilon_{eq} = \frac{2}{3}\sqrt{\varepsilon_{xx}^2 + \varepsilon_{yy}^2 + \varepsilon_{zz}^2 - \varepsilon_{xx}\varepsilon_{yy} - \varepsilon_{xx}\varepsilon_{zz} - \varepsilon_{yy}\varepsilon_{zz} + 3(\varepsilon_{xy}^2 + \varepsilon_{xz}^2 + \varepsilon_{yz}^2)}. \tag{13.36}$$

In **General options** the Strain window size can be set. Also the scalar image can be **enlarged**, see Section *Extract vector component* in the main **DaVis** manual for a detailed description of this parameter.

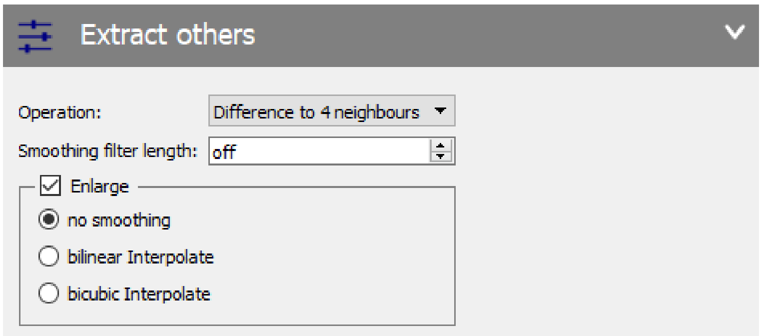


For the Operations Tresca strain and Von Mises strainand **Ezz estimation** can be set. Therefore, choose an **Ezz estimation type** between Plane strain assumption: Ezz = 0 and Plane stress state: Ezz=(Exx+Eyy)*nu/(1-nu), whereas *nu* depicts the Poisson's ratio. Additionally, if Plane stress state was selected, then the Poisson's ratio estimation source can be set by either a constant value as user defined input, by calculating locally at each vector location, by calculating the average using the entire vector field, or by calculating the average utilizing a defined rectangle. For the latter, click on **Define** to draw a rectangular region in the source image. Moreover, select a Poisson's ratio estimation type.



13.9.5 Extract others

Calculate differences to neighbours, surface curvature (needs Height scalar field) or surface angle to z as scalar field and store as image.



The following **Operations** can be performed:

Difference to 4 neighbours: Determines the local derivation from the average of four neighbouring vectors.

Difference to 8 neighbours: Determines the local derivation from the average of eight neighbouring vectors.

Max. surface curvature: Calculates the maximum of one of the two second derivatives in x and y direction of the surface height scalar field which is the equivalent to the maximum curvature in that direction (most suitable for cylinders). This only works for vector fields that contain a scalar plane with surface height information as derived in a StrainMaster3D system, where a local surface normal can be calculated.

Avg. surface curvature: Calculates the average of the two second derivatives in x and y direction of the surface height scalar field which is the equivalent to the maximum curvature in that direction (most suitable for samples with surface normal in any directions and arbitrary surface height distributions containing local maxima, minima and saddle points). This only works for vector fields that contain a scalar plane with surface height information as derived in a StrainMaster3D system, where a local surface normal can be calculated.

Surface angle to z-axis: Calculates the inclination of the surface height distribution in respect to the z-axis. This only works for vector fields that contain a scalar plane with surface height information as derived in a StrainMaster3D system, where a local surface normal can be calculated.

Additional components: If available the scalar value of a scalar component (stored inside the buffer) at the corresponding position is used. For example:

- Surface height
- Correlation value
- Correlation value
- Peak ratio
- Uncertainty V
- Uncertainty Vx

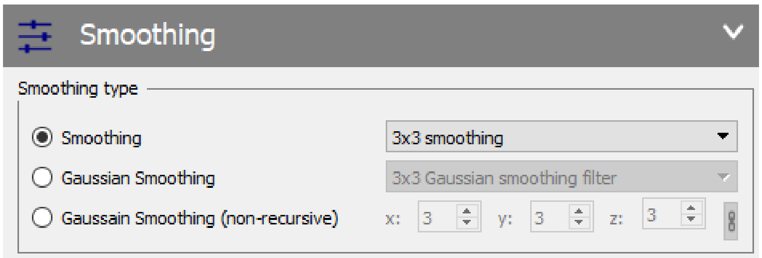
- Uncertainty Vy

Again, the scalar image can be **enlarged** and a **Smoothing filter length** can be applied (see Section 13.9.1 for a detailed description of these parameters).

13.10 Operation: Filter

13.10.1 Smoothing

Filter the vector field by using one of the **Smoothing filter types**: **Smoothing**, **Gaussian Smoothing** with the corresponding Filter matrix, and **Gaussian Smoothing (non-recursive)** with its corresponding xyz-coordinates. Please refer to Section 12.12.1 for more information on Smoothing filters.



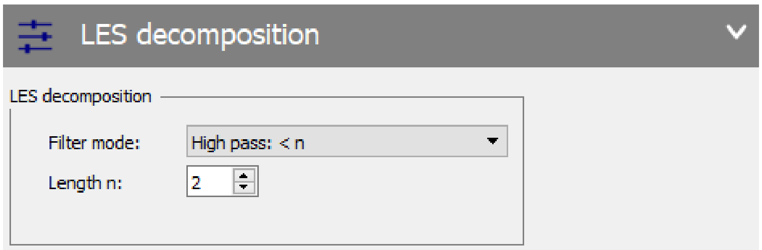
13.10.2 LES decomposition

Decomposition of a vector field according to large eddy simulation by separating the space spectrum of the field into two (or more) groups. This type of decomposition is extremely useful in visualizing small-scale turbulent eddies, as it removes the translation of the larger-scale field. The domain for the calculation of the local mean velocity can be specified using the **Length n** text box. Following filter modes can be selected:

- **High pass** $<$**n :** Returns high frequencies $< n$ by calculation of the local mean velocity for a given vector field and subtraction of the local mean velocity from the vector field.
- **Low pass** $>$**n :** Returns low frequencies $> n$ by calculation of the local mean velocity for a vector field.
- **High pass** $<$**n,** $<$**2n,. . . ,** 2^N**n:** High pass filter with gradually increased domain for the calculation of the local mean velocity. End

condition for the decomposition is $2^N n < min(N_{vh}, N_{vv})$ with N_{vh}, N_{vh} number of vectors in the given field in horizontal or vertical direction. The results are written to a multiframe vector buffer.

- **Low pass $>$n, $>$2n,. . . , 2^Nn:** Low pass filter with gradually increased domain for the calculation of the local mean velocity. Since $2^m n <$number of vectors (vertically or horizontally) the vector field is decomposed, the results are written to a multiframe vector buffer.
- **Range 0-n, n-2n,. . . , 2^{N-1}n $-$ 2^Nn:** Decomposition of the vector field according to a band pass filter of a gradually increased domain of the size $2^{N-1}n - 2^N n$ by calculating High pass $> 2^N n$ and subtraction of High pass $2^{N-1}n$.
- **Length n:** Specifies the domain for the calculation of the local mean velocity in units of interrogation windows (or vectors).

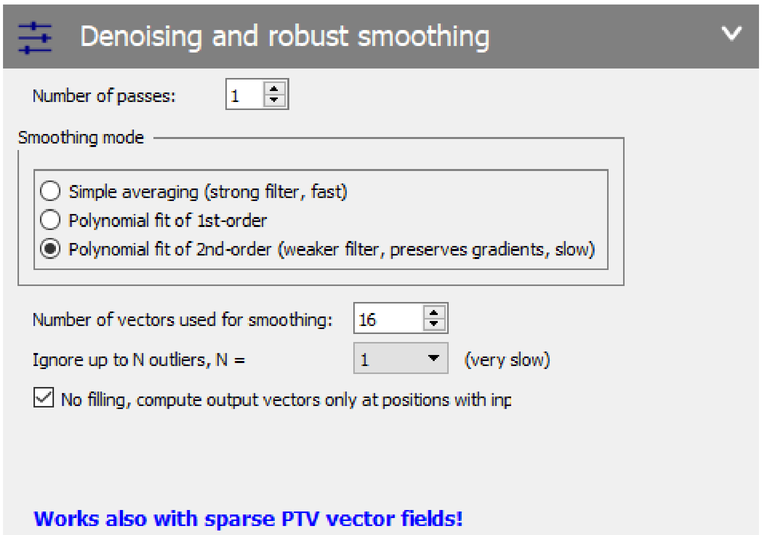


References

Adrian RJ, Christensen KT, Liu ZC (2000): 'Analysis and interpretation of instantaneous turbulent velocity fields', Exp. in Fluids, 29/3, p. 275-290

13.10.3 Denoising and robust smoothing

This advanced filter helps to take out the noise (spurious vectors) from the vector field. For all positions in the vector field it successively looks in the neighborhood for the specified number of vectors for smoothing *NoVectors* closest vectors to the given position in the source and replaces the vector in the result vector field as a function of the *NoVectors* vectors and the specified smoothing mode *mode*.



For **ignore up to** N **Outliers,** N**=0** this function works as follows: To all *NoVectors* vectors either a constant is fitted (*mode*=Simple Averaging), a polynomial of first order in x and y direction (*mode*=Polynomial fit of 1st order, which is a linear gradient in x and y) or a polynomial of second order (*mode*=Polynomial fit of 2nd order, which are a combination the functions changing with x, y, x^2, y^2 and $x \cdot y$). Then the vector in question (at the center) is replaced by the value of the fitted function at that position.

For **ignore up to** N **Outliers,** N**=1** this function works as follows: For the found *NoVectors* neighbors *NoVectors* subsets of $NoVectors - 1$ vectors are formed (in each a different *Outlier* was thrown out. The fit is performed as described above and the vector in question (at the center) is replaced by the value at that position of the fitted function *with the smallest fit error*. This is most probably the function, where an irritating (a spurious vector if there really was any) is missing because it has the smoothest gradients.

For **ignore up to** N **Outliers,** N**=2** this function works as follows: For the found *NoVectors* neighbors $NoVectors * (NoVectors - 1)/2$ subsets of $NoVectors - 1$ vectors are formed (in each a different *pair* of *Outliers* was thrown out). Again, the fitted function with the smallest fit error is used to replace the vector at the position in question by the value of the fitted function.

For **ignore up to** N **Outliers,** N**=3** this function works as for $N = 2$, only that : $NoVectors * (NoVectors - 1)(NoVectors - 2)/(2 * 3)$ subsets exist where different triplets are missing.

Therefore, for a higher number *N* of outliers the necessary time to calculate the fit of all subsets is starting to grow very, very fast. The advantage is that while a spurious vector can be replaced by a predicted value considering gradients of second order, it will not be included into the calculation of all neighbors because most likely the best fit for other positions can be classified as an outlier and thus be ignored. In areas where no spurious vectors are present, the gradients (up to 2nd order) will not be disturbed.

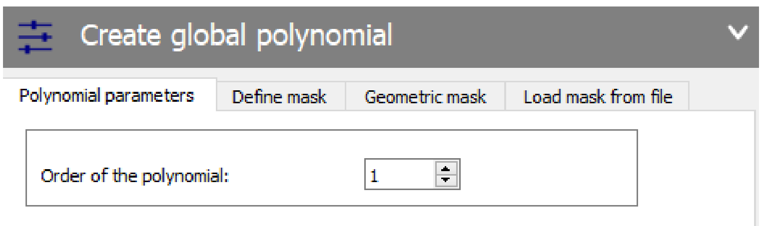
For a number greater than 1 in the field **passes** this process is repeated iteratively.

13.10.4 Create global polynomial

Creates a single polynomial, which optimally fits the given 2D vector field. The created polynomial will be minimal in terms of the least mean square difference to the given vector field. The parameter **Order of the polynomial** defines the order of the created polynomial. Possible values are orders between 0 and 10. An order of 0 creates a global mean vector field.

By defining a mask only those values, which are not masked, are used for the polynomial fit.

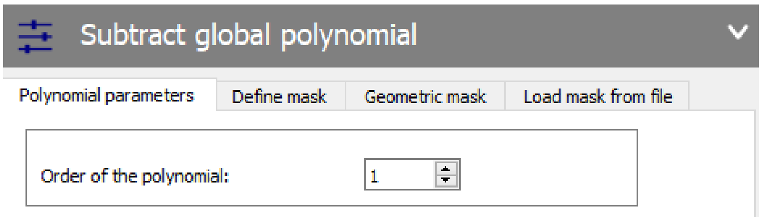
Please refer to Section 12.10 for details on **Define mask**, **Geometric mask**, and **Load mask from file**.



13.10.5 Subtract global polynomial

Subtract global polynomial internally uses the **create global polynomial** batch operation 13.10.4 to create a global polynomial. This global polynomial is used to subtract the given vector field from it.

The parameter **Order of the polynomial** is identical to the one in the **create global polynomial** batch operation, defining the order of the polynomial. Please refer to Section 12.10 for details on **Define mask**, **Geometric mask**, and **Load mask from file**.



13.10.6 Polynomial filter

The polynomial filter removes high frequency noise from a vector field and can be used to fill up disabled vectors.

For all positions in the vector field it fits a polynomial (in a least squares sense) with the order defined in the field **Order of the polynomial** to the active vectors inside the range specified by the parameter **Filter length**. Every vector component (x, y and z) will be fitted separately and the initial values get replaced by the value of the fitted polynomial.

The possible values for the parameter **Order of the polynomial** ranges from 0 to 4. The **Filter length** has to be a odd size between 3 and 11 and greater than the order of the polynomial.

The filter lengths in x,y and z can separately be chosen by deactivating the parameters connectivity by pressing the button with the chain symbol visible at the right side of the parameter block. If the chain is closed the parameters are connected, otherwise if the chain is open it is possible to define them independently.

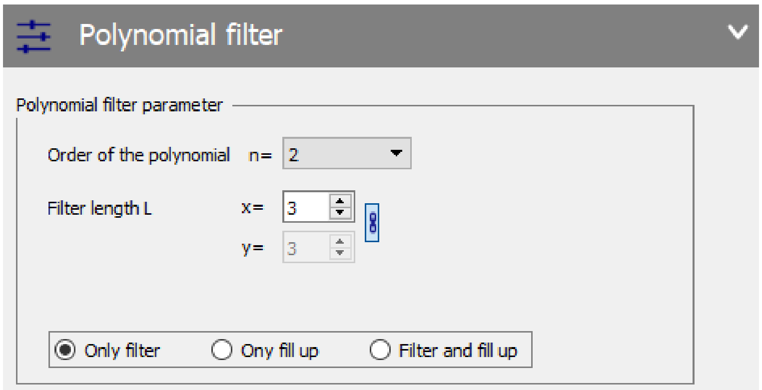
There are three possible settings for the polynomial filter:

- **Only filter**: Only filter enabled vectors.
- **Only fill up**: Only try to fill up disabled vectors.
- **Filter and fill up**: Combination of both.

It is feasible that a polynomial fit can not take place with the given polynomial order in a specific area because there are to less vectors for the fit.[1] If this happens the old values will be assumed unchanged.

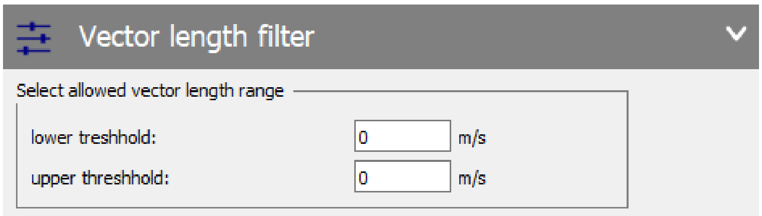
[1] Less than 33% percent from the maximum numbers of vectors inside the filter area or less vectors than coefficients needed for the fit.

When the input buffer is a volume vector field the batch operation fits polynomials not just in *x* and *y* direction but also in the *z* direction. If operating on tomographic vector fields the number of z-planes has to be greater or equal to the value of **Filter length**. Because the minimum value of the parameter **Filter length** is 3 the input volume also has to have a minimum depth of 3.



13.10.7 Vector length filter

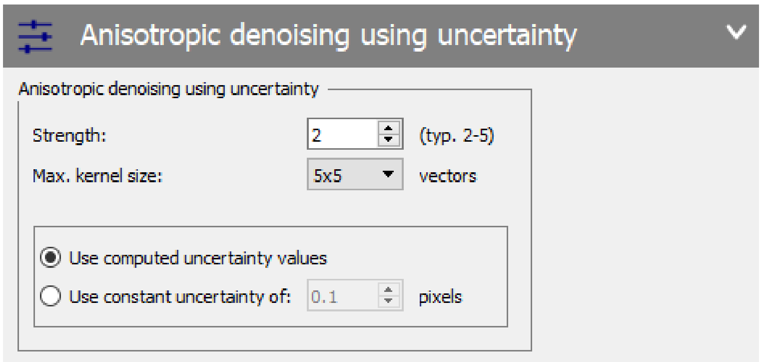
The vector length filter disables all vectors with a length larger and/or smaller than a given threshold. Therefore, select an allowed vector length range (**lower and upper threshold**).



13.10.8 Anisotropic denoising using uncertainty

This function applies polynomial fit to local neighborhood by using uncertainty. The filter length is extended so long as the deviation of values from fit is smaller than the uncertainty band around the values. Therefore, choose a **Maximal kernel size** defining the maximal filter length. The **Strength** value specifies the uncertainty band. The higher the value the higher the tolerance of the uncertainty band.

Choose, between **using computed uncertainty values** or **using constant uncertainty**. For the latter, enter the number of uncertainty pixels.



13.11 Operation: Copy and reorganize frames

Please refer to Section 12.13 for the functions of this Operation.

13.12 Operation: Copy and reorganize data sets

Please refer to Section 12.14 for the functions of this Operation.

13.12.1 Convert vector set to vector volume

Converts a set of N single frame vectors to a set of one volume with N planes.

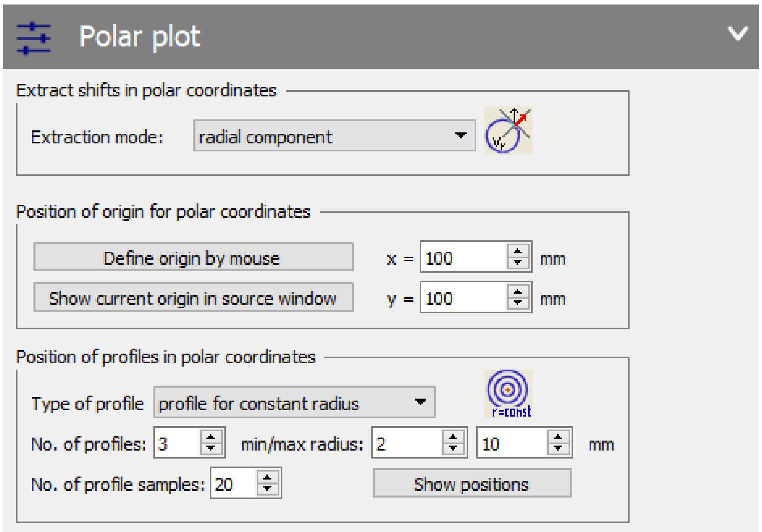
13.13 Operation: Plot

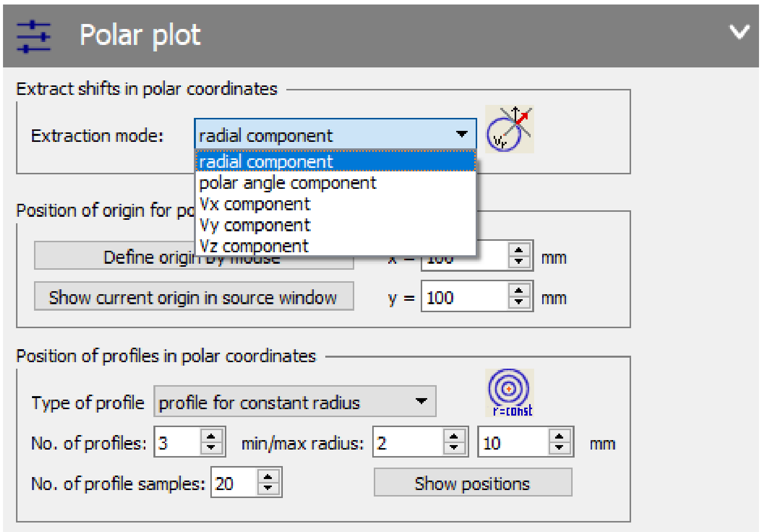
13.13.1 Polar Plot

The operation **Polar plot** allows to extract a family of profiles of V_r, V_{angle} (in respect to a user specified origin), V_x, V_y, V_z values along

- the circumference of concentric circles with fixed radius in respect to a user specified origin or
- radial lines in respect to a user specified origin.

See the following two screenshots of the parameter dialog:





The number of profiles in the family, the number of samples as well as the min/max value that define the boundaries of the family of profiles can be specified.

The origin can either be set with mouse (press the button labeled **Define origin by mouse**) or defined directly entering the pixel position in the corresponding text fields. Pressing **Show current position in Source Window** will draw the currently selected position in the source window buffer (if visible) as overlay graphics.

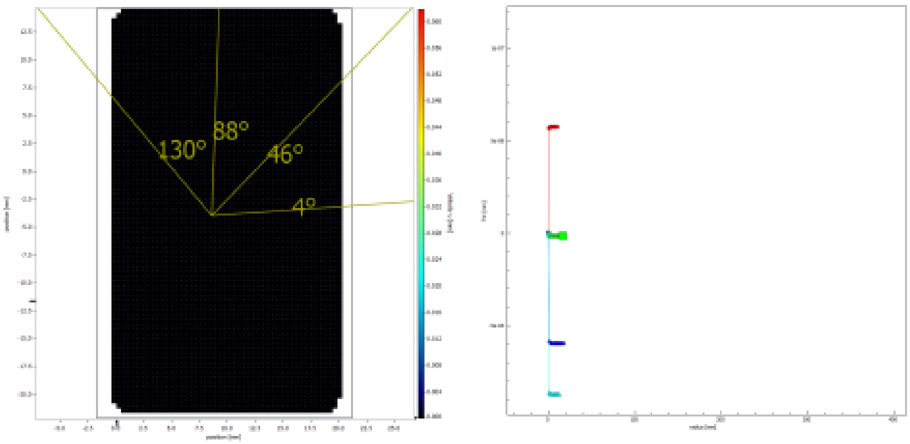
In the group box **Position of profiles in polar coordinates** you define how the profiles are taken (**profile for constant radius** or **profile constant angle**). For the current mode you select the number of profiles and the minimum and maximum radius (or angle) for which profiles should be extracted.

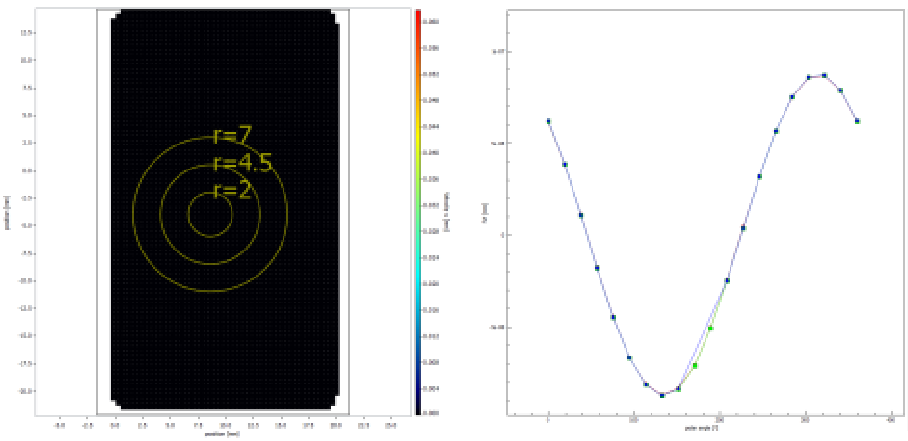
Furthermore you define the number of sampling points for each profile.

Pressing **Show positions** will blend in an overly graphics in the Source window (if visible) to show the position of the profiles for the currently selected min / max values and the specified origin.

The following screenshots show examples for the extraction of such profiles:

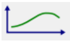
See the following two screenshots of the parameter dialog:



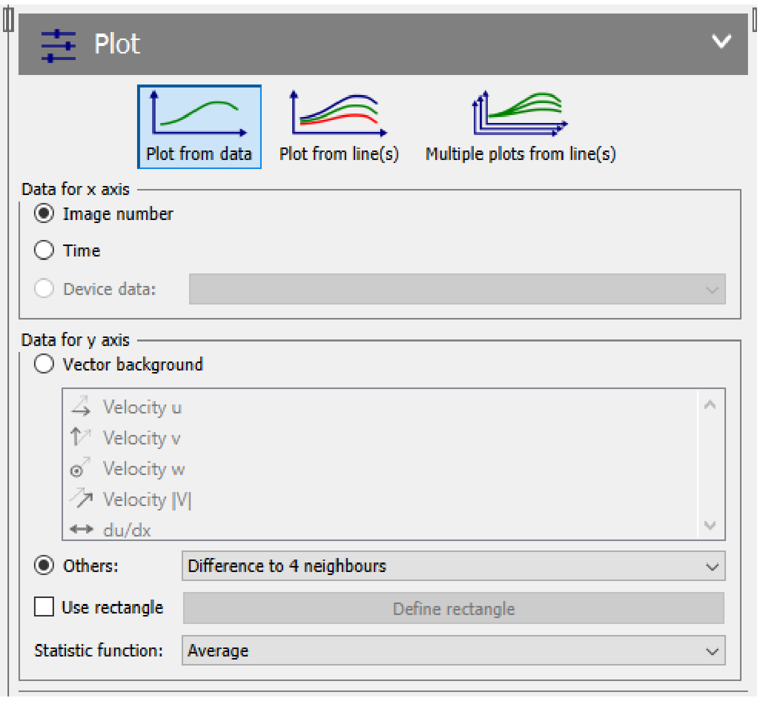


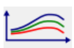
13.13.2 Plot

Create a plot between two scalar sources of the input buffer. The result will be an Plot-set for each source set.

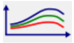
For **plotting from data** , the x-dimension can be taken from the image number of the source set, time, or from the selected device data. Please note that device data can be only selected, if it is available in the source set. This is typically an external signal that was recorded with an analog digital converter during the recording and is attached to each image and has been propagated to the vector field.

Select the data to plot on the Y-axis. The available options are **Vector background** or **Scalar field**. In Vector background, the values comprise functions as in the processing groups in processing dialog "extract scalar field:. . . ". Choose a Scalar field. The y-dimension can be taken from statistical image data of the whole image or a defined rectangular region. Specify **Others** to calculate differences to neighbours, surface curvature (needs Height scalar field) or surface angle to z as scalar field and store as image. For the latter, the check mark **Use rectangle** has to be enabled. Specify a **Statistic function** to calculate the average value, the standard deviation, minimum, maximum or the sum of values of all selected vectors inside the chosen area.



Select one of the modes **Plot from line(s)**  or **Multiple plots from line(s)**  in the plot mode group. Depending on the selection, you get one plot with *n* lines or a set of *n* plots with one line each, respectively.

To create a plot from line, you need to select the input set range you want to extract (start, end and increment) as in the processing in the **Source** group and do the following:

Select one of the modes **Plot from line(s)**  or **Multiple plots from line(s)**  in the plot mode group. Select the **Data source** to plot data on the Y-axis. The available options are **Vector background** or **Others**. Then, select the extraction mode

- arbitrary line
- horizontal line
- horizontal lines y1...y2
- circle line
- vertical line
- vertical lines x1...x2

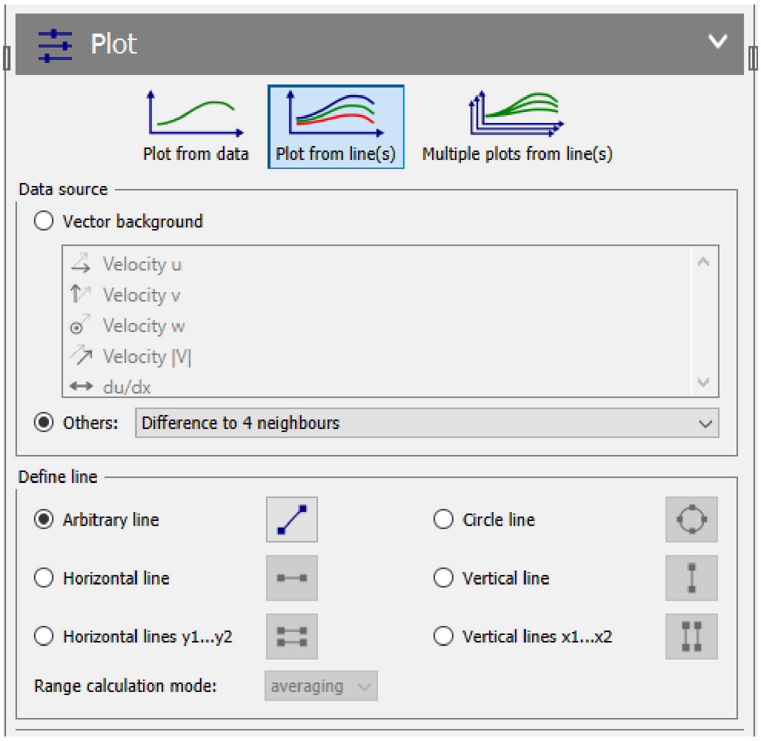
and manually define the position in the vector field.

For an arbitrary line, a vertical line or a horizontal line, the values for each vector along the line are calculated and shown versus the position along that line.

For horizontal range, the values are averaged inside the horizontal range (along x) for each y position and plotted against that y position. For the vertical range mode, it is vice versa.

For horizontal lines y1...y2 and vertical lines x1...x2 Range calculation mode can be selected: averaging or summing.

The X-axis is set automatically to **values along line** and cannot be changed.



Please refer to Section **Plots for Vector Sets** 20.3.1 for further details.

14 Processing Color (RGB) Images

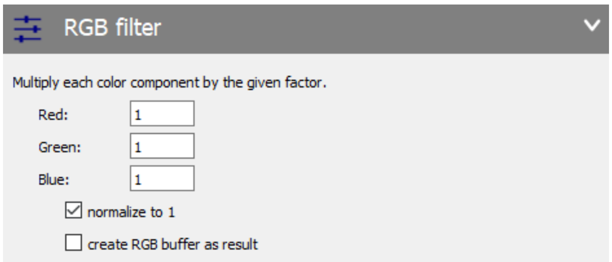
This chapter describes the special operations for color image source files. The general operations for image processing are available for RGB images, too, but a number of functions like filter are not working for color images.

14.1 RGB Filter

14.1.1 RGB Filter

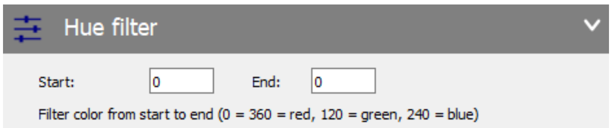
Each color component can be multiplied by a certain factor. The result is either a float buffer or, if **create RGB buffer** is selected, again a color buffer. For the first mode a **normalization to 1** is possible, then the resulting intensities are in range 0...1.

This function can be used to extract one color component from the RGB buffer: Just set one factor to 1 and the other factors to 0.



14.1.2 Hue Filter

Filter colors with the color circle in a given range are converted from RGB into intensities: Values for start and end are 0 = 360 = red, 120 = green, 240 = blue.



14.1.3 Create R/G/B Frames

The image is split into the three color components. Each component is stored in a single frame of the resulting float image buffer: The first frame includes the red component, second the green and third the blue component.

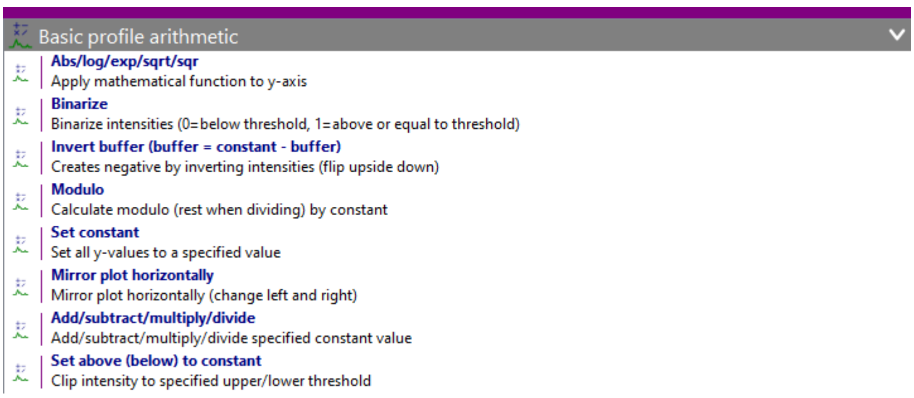
For example the statistics function may be executed as next operation. At the end the frames can be recombined into a RGB image by **convert image to RGB – combine R/G/B-frames to RGB image** (page 293).

15 Processing Profiles

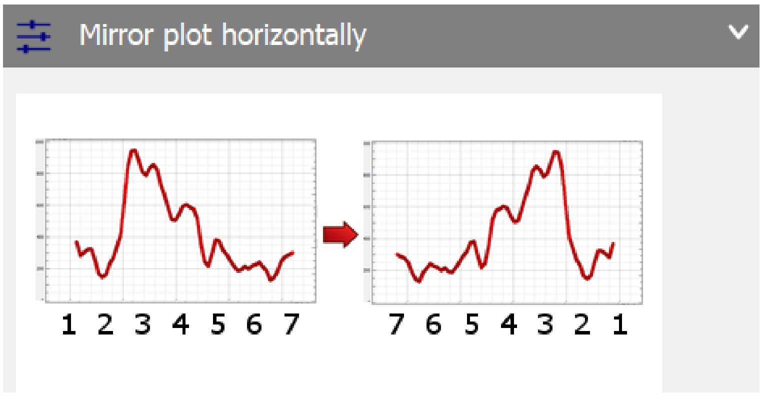
For profile sources only a very restricted list of operations is available. Most operations known from image processing are not working or don't make sense (e.g. nonlinear filter).

15.1 Basic Profile Arithmetic

The basic profile arithmetics are more or less equal to the image arithmetics. All operations are applied to all profiles of a buffer. E.g. when adding a constant value, then this raw value is added to each raw profile point. See sections starting on page 219 for detailed information about the basic image arithmetics.



15.1.1 Mirror plot horizontally

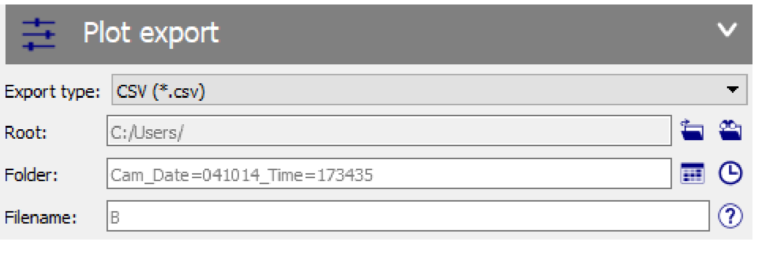


The x-axis of the source profile is plotted in reverse direction.

15.2 Export

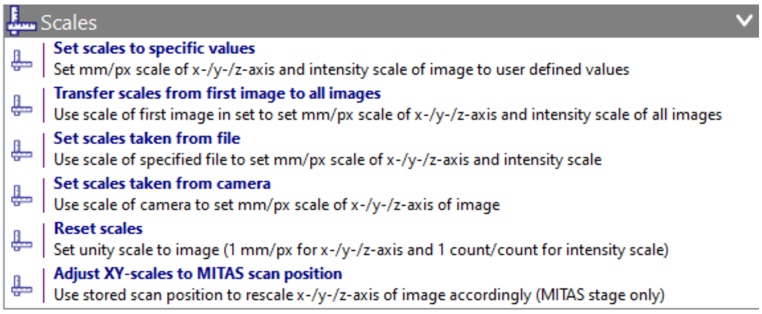
15.2.1 Plot Export

Exports plot as ASCII file in CSV file format. To select **Root**, click on either to select a directory, or on to open export directory. Moreover, select a **Folder** -, and a **File** name.



15.3 Scales

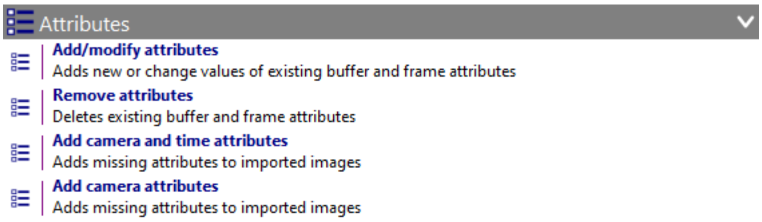
A scale is a linear mapping of the X (horizontal), Y (vertical), Z (depth, used for real 3D buffers only) or I (intensity)-axis to a new range of values. Scales can be assigned to an image buffer or predefined for each camera. See sections starting on page 234 for detailed information about functions in this operation.



15.4 Attributes

Attributes are additional information (strings and arrays of integer or float values) of the buffer and are often set by the image and analog data acquisition functions and by the processing functions.

See sections starting on page 236 for detailed information about functions in this operation.



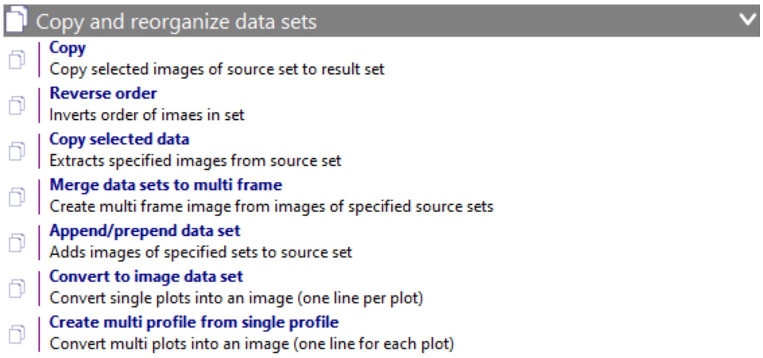
15.5 Statistics

15.5.1 Average Profiles

Average the profiles from all profile buffers is the source set. Store the averaged profile as result.

15.6 Copy and Reorganize data sets

This Operation allows to copy and reorganize data sets.



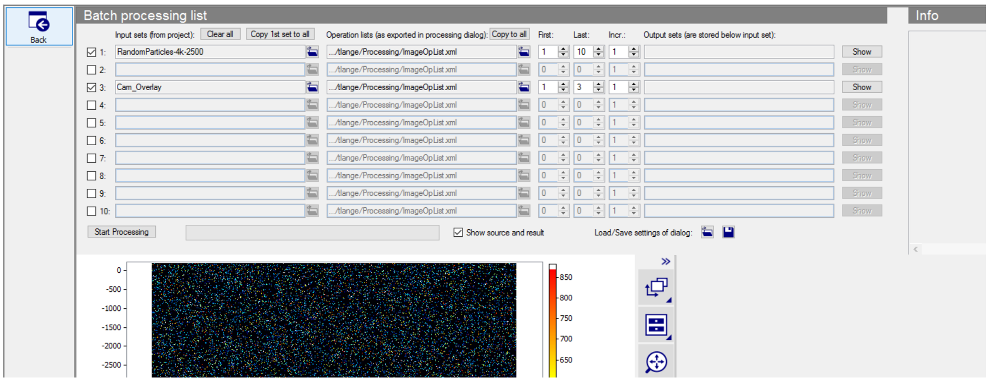
See sections starting on page 284 for detailed information about functions in this operation.

15.6.1 Create Multi Profile from Single Profile

A multi profile is a combination of single profiles in one buffer, where each profile is stored in another frame. All profiles of a multi profile buffer are painted with different colors in the display. With this operation the single profiles from all source buffers are combined into one multi profile as result.

16 Batch Processing Dialog

To access the dialog press right mouse button on a source set in the project tree and select item **Batch processing**.

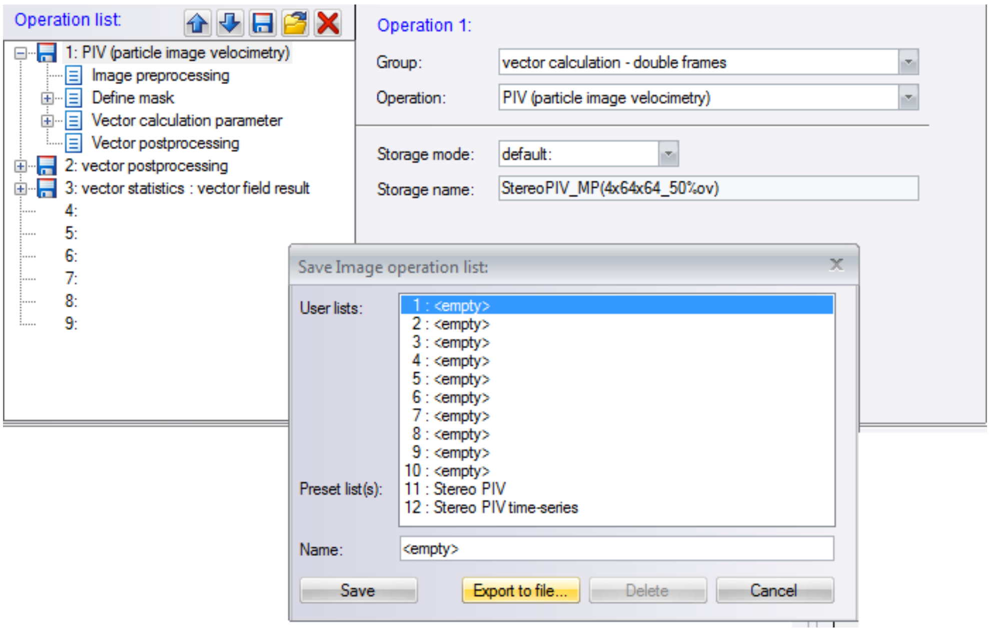


In the dialog any number of input sets can be specified. It is possible to select the same set several times. To select a new or different set please switch the checkbox at the left of the current line at first. Only activated lines are used during processing. Then press the yellow load button right to the field of the set name.

Also a previously stored operation list must be defined for each input set. The same operation list can be used for several input sets. The only restriction is that image operations have to be specified for images and vector operations for vector sets (see Check consistency). To select a operation list press the yellow load button right to the text field.

To create an operation list (stored as XML file) go to the processing dialog and set up the operation list. Then press

- the button above the operation pipeline (1) and
- select "Export to file..." button (2) and
- specify a name and a destination folder - refer to following screenshot



The name of the result set is shown to the right in each line and is composed of "OperationList_" plus the operation list name for easier recognition later. If the operations list has a pipeline of several operations the rest is stored beneath that folder like in the standard batch processing. Also the data range **First** and **Last** image of the input set to be processed can be specified as well as the **Increment** (e.g. only take every 2nd file). If you are done you can check the current settings for consistency by pressing Check consistency. Only if all input sets / operation lists exist and if input set type (image / vector / profile) is consistent with the input set type the Start Processing button is enabled. Otherwise a message tells you what kind of problem was encountered. Press **Start Processing** to start the job list and calculate all results.

The progress bar is an indicator on how many files are processed in the currently active set. The number **1:**, **2:**, **3:**, . . . in front of it shows the job number currently in operation. After all jobs are finished you can write down the final file names of the result sets: Pressing button **Start Processing** repeatedly does not overwrite the previous result set but rather adds a suffix _01, _02, _03,... to get a unique name for the result set. Pressing button **Show** (in the very right of each line) shows the source and result set of each job for a quick look on the results. Leaving the dialog returns to the project manager. The result sets in the project are now exactly as if the batch processing dialog would have been used manually.

Known restrictions:

- Global settings, e.g. the calibration of the project, are shared by all operations.
- Up to 10 jobs can be defined.
- Show source / result displays only one result and not every single file of all sets.
- Cluster script processing is not supported by this wizard.

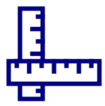
17 Scaling

17.1 Introduction

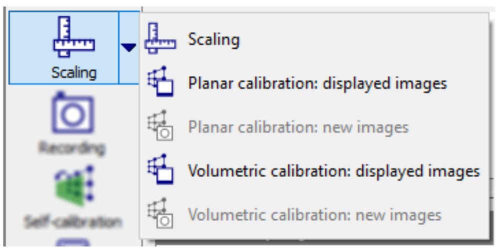
In all cases the term *scaling* refers to the calibration process. Doing a calibration is important because the results should be shown in scaled units representing the true (world) dimensions. So the image scale pixel/mm should be determined.

17.2 Scaling Dialog

A successful scaling procedure will store all acquired data and the scaling information under the **Properties** folder of the project in the subfolder **Calibration**.



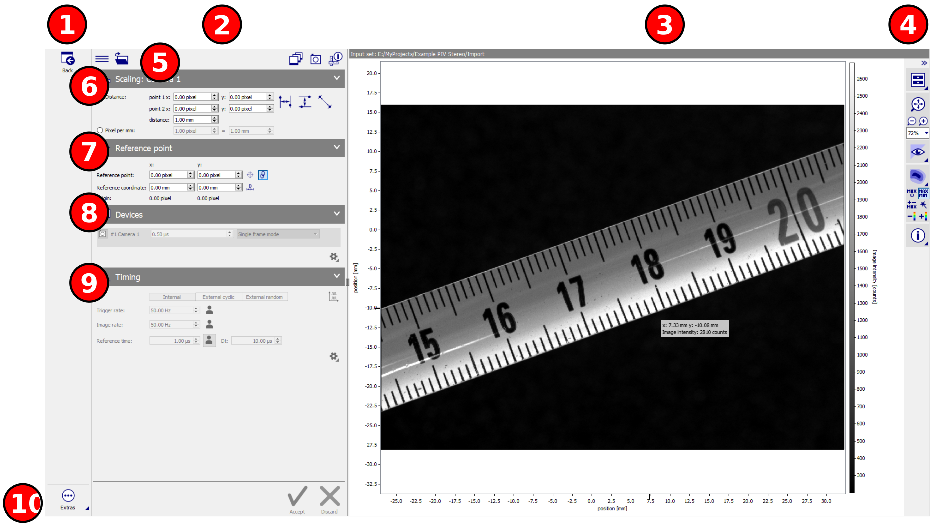
The **Scaling** button (see margin) in the project toolbar opens the **Scaling** dialog. You may have to click the triangle next to it to see all the available options for calibration.



17.3 Basic Layout

The scaling dialog consists of the following elements, which are explained in more detail afterwards:

1 In the **toolbar** the scaling process can be aborted and the dialog will be closed.



2 The center dialog area gives access to setting check boxes, entering numbers, or action buttons. It is divided into the sections **Scaling** 5 , **Reference point** 6 , **Devices** 7 and **Timing** 8 .

3 The dialog area to the right shows the image for a specific camera.

4 The view tool bar allows configuration of the image view, e.g. zoom or color mapping.

5 In the menu toolbar different behavior of the dialog can be selected (e.g. live recording on/off).

6 In the **Scaling** section all actions for defining a pixel/mm scaling can be found.

7 In the **Reference point** section the origin can be defined without changing the pixel/mm scaling.

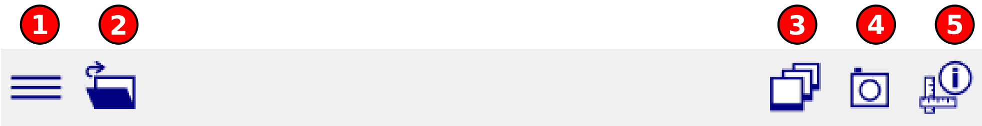
8 The **Device** section is only visible if hardware is connected and a hardware license available. In this section the device settings are defined, such as exposure time or light source control.

9 The **Timing** section is only visible if hardware is connected and a hardware license available. In this section the timing settings are defined, such trigger rate for image acquisition, trigger source, and delay times.

10 In the **Action** section the scaling result is accepted and stored to the project or discarded.

17.4 Menu Toolbar

The menu toolbar contains 5 action buttons:



1 Pressing this button hides the sections of the center dialog in order to have more space for the camera image while the sections are still available as overlay dialogs, see Fig. 17.1.

2 By default, the selected data in your project is the input when entering the dialog. With the **Load data from project** button a new recorded data set from your project can be selected.

3 The **Show preselected set** button switches back to the currently active loaded data set (selected when entering the dialog or opened using option 2)

4 The **Show live view** button switches the live recording on and off. This option is only available if hardware is connected and a hardware license loaded.

5 The **Show active calibration set** button shows the existing calibration of your project for reference if available.

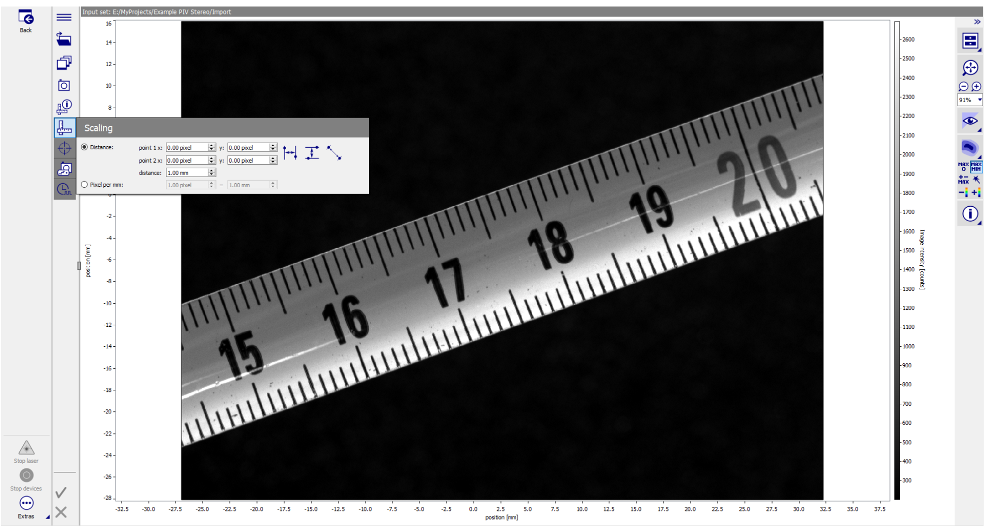
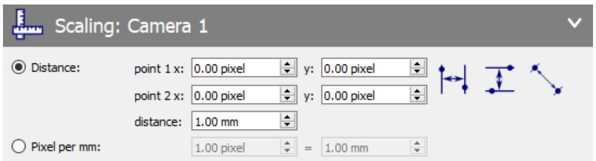


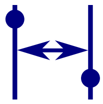
Figure 17.1: Scaling dialog with collapsed center dialog

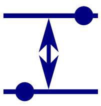
17.5 Scaling

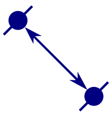
The dialog looks as in the following screenshot:



The goal of the current step is to define conveniently the mm/pixel scale of the image. This dialog allows to do this either by entering this ratio manually (radio button **Pixel per mm**) or by defining the ratio by specifying two points and their distance manually or interactively (radio button **Distance**).

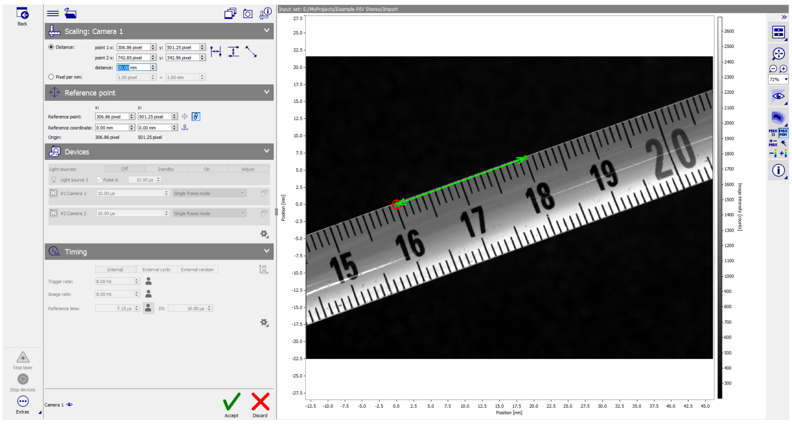






Pressing one of the three buttons shown in the margin allows interactive definition on the image with your mouse. To do so, please click the dialog button first, then left-click to point 1, drag the mouse to the position of point 2, and release the mouse button. The three dialog buttons define the horizontal distance, the vertical distance and the point-to-point (beeline) distance.

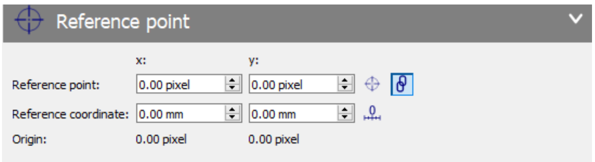
From this distance in pixels and the **distance** in mm defined in the dialog, the pixel/mm scale is calculated. Please refer to the following screenshot:

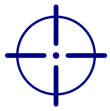


Please note that the first click defines the origin of the coordinate system if the option with the chain icon (see margin) in the **Reference point** section has been selected. Otherwise the previously defined origin stays at its position. Either way, the origin can be redefined in the **Reference point** section if needed.

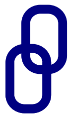
17.6 Reference Point

If the scaling is fine but the origin is not, a new origin can be redefined for the (already scaled) image. Furthermore, the value of any other reference point (not necessarily the origin) can be set.

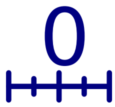




Either enter the numbers directly to the text boxes or click to the **Set reference point port mouse** to do this interactively by clicking with the left mouse button into the image.



With the "chain" option selected, the new origin is automatically selected by a new two point definition in the scaling section.

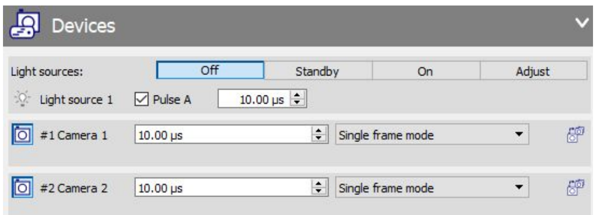


Zero adjust resets the reference point coordinates to the origin (0 mm | 0 mm), which is the default. The reference point coordinates can be specified to have any other desired laboratory coordinates (e.g. 10 mm | 200 mm).

17.7 Devices

The device and timing sections show all the relevant settings for acquiring images using your connected hardware. These sections may not be visible if you have no hardware license. The main elements allow to switch on and off light sources and cameras, to define the pulse length of your light source, and to set the exposure times of your cameras.

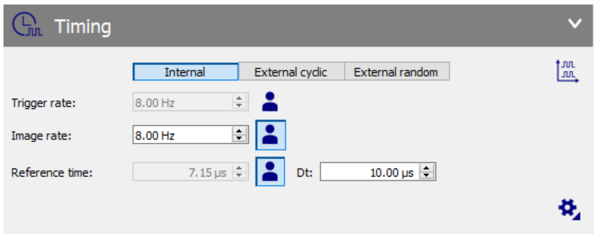
Depending on your available hardware, options might look different. Please refer to the "Recording Dialog" chapter for a more detailed explanation of all possibly available elements.



17.8 Timing

In the **Timing** section, the trigger source (generated internally or using an external trigger source, such as a hand trigger) can be selected. Furthermore, the trigger rate and the image rate can be set.

If the **User** button is not selected (as for "Trigger rate" in the following screenshot), the maximum rate is used. If the **User** button is selected, the rate can be specified in the now enabled text box.



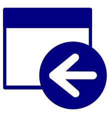
Depending on your available hardware, options might look different. Please refer to the "Recording Dialog" chapter for a more detailed explanation of all possibly available elements.

17.9 Action Toolbar

In the action toolbar you can check for which cameras a new scaling / a new origin have been defined so far. Clicking on the **eye** button shows the image and the specified points.

Pressing **Accept** will store the currently changed scalings to the project, overwriting all existing scalings / calibrations of the project. Those will be copied to the calibration history, though.

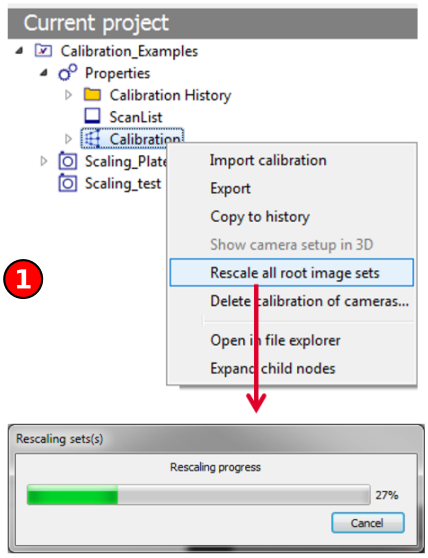
Discard will discard all changed made so far and reload the currently active calibration of the project. This will not close the dialog, allowing to proceed with the definition of scalings and reference points.

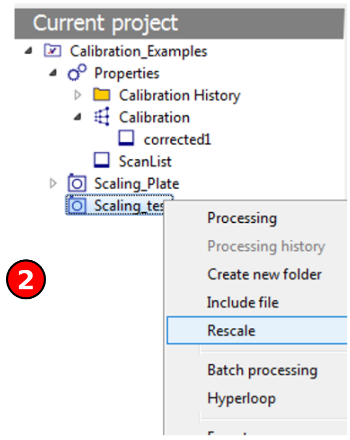


You can leave the dialog by pressing the **BACK** button in the upper left corner in the menu toolbar (which will also discard all changes).

17.10 Rescaling Existing Recordings

Outside the **Scaling** dialog you have the option to rescale existing recordings and apply the currently active scale to all images.





1 Rescaling all image and recording sets of your project:

Pressing the right mouse button on the **Properties/Calibration** entry in your project opens a context menu. Selecting the **Rescale all root image sets** option will rescale all image and recording sets at top level in your project. Please note that depending on the amount of data this might be time consuming!

2 Rescaling only one set of your project:

Pressing the right mouse button on a single image set or recording set in your project opens a context menu. Selecting the **Rescale** option will rescale only the selected set in your project. Please note that depending on the amount of data this might be time consuming!

18 Calibration

18.1 Introduction

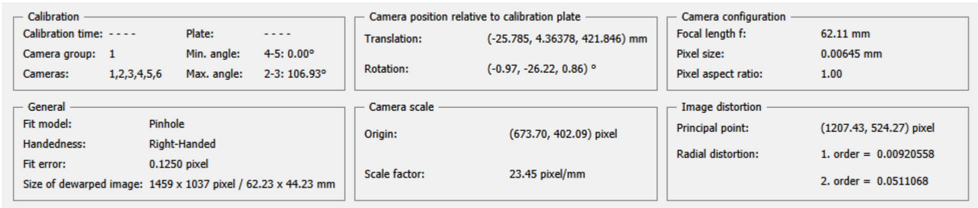
In all cases the term *calibration* refers to the image calibration process to determine and correct perspective distortions and define the mm-scaling of images. A calibration is important for processings because:

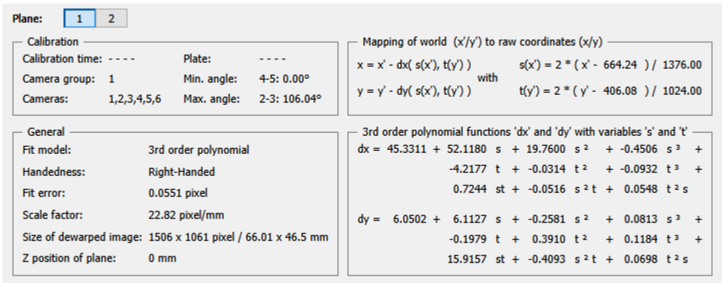
- The results should be shown in scaled units representing the true (world) dimensions. So the image scale pixel/mm should be determined.
- For oblique viewing setups or views through curved glass windows, the images show some image distortions due to perspective projection (and possibly inherent camera lens distortions). This can be corrected via the appropriate calibration.
- For stereoscopic and volumetric measurements, an internal representation for the geometrical setup of both cameras relative to the sample is needed. This is determined.

There are some suppositions for the calibration:

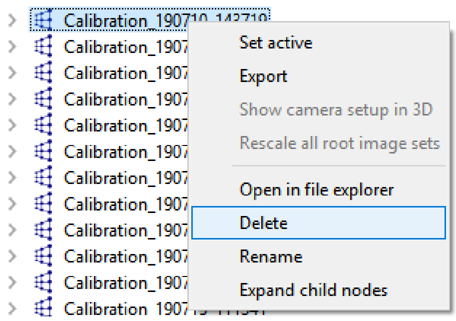
- Knowledge to operate the hardware in order to collect the calibration images must be available.
- If the camera(s) are in focus on the sample surface, it is a simple task to move the calibration plate into the focal plane - and when it is removed, you are immediately ready to start acquiring images. If the calibration is done first without being careful about the plate position, the system might be unfocussed on the sample, making it necessary to refocus and recalibrate the system.
- A calibration can be performed before or after the sample images have been acquired (providing the camera system has not been altered).

A successful calibration procedure will store all acquired data and the calibration information under the **Properties** folder of your project in the folder **Calibration**. A click on the calibration entry in the project manager shows the parameters of the perspective image correction function including the **Standard deviation of the fit** to check the quality of the calibration.



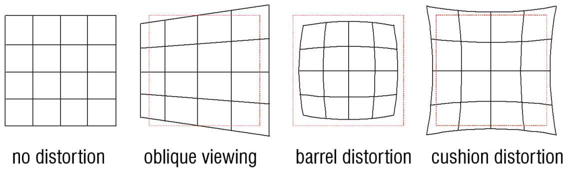


A new calibration – imported or when using the calibration dialog – as well as any additional adaptations of the calibration (e.g. setting a new origin, self-calibration procedure) will store the previously active calibration in the folder **Properties/Calibration History** before overwriting the **Properties/Calibration** files. Thus, it is possible to restore any previous calibration done in your project: Importing a calibration is possible via a right mouse click in the project manager tree item of the calibration. The **Calibration History** folder can be deleted anytime to save up hard disk space (right mouse click: delete), but it will not be possible to restore old calibrations afterwards.



18.1.1 When Perspective Calibration is Necessary

If the camera is not looking perpendicular to the measurement plane or lens distortions are present (typically radial barrel and cushion distortion), the image distortions need to be corrected. Otherwise, it is not possible to define axis orientations and a pixel/mm scaling that is valid for the entire image. In order to calculate further derivatives (e.g. a PIV calculation), an image correction is required as an intermediate step.



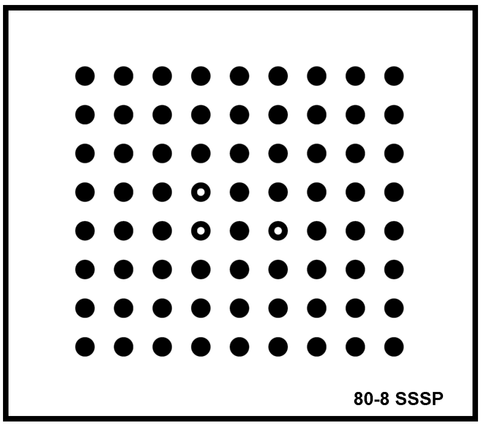
The perspective image calibration dialog and the calibration plates provided by LaVision are a simple tool to correct these distortions. In the following we provide some background of the calibration process.

Please note that DaVis 10 will not show a scaling on the recorded images after a calibration is completed. In order to show a scaling on the images or derivative data the image correction need to be applied in the Processing.

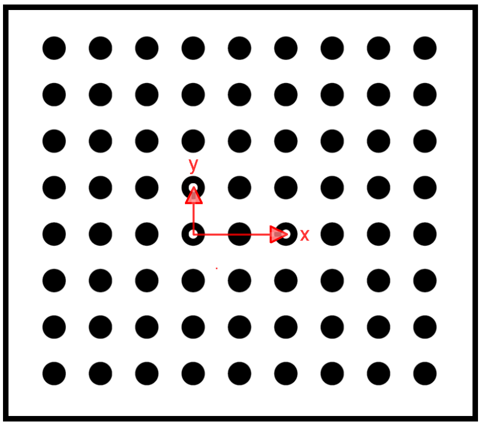
18.1.2 How to Calibrate Using a Calibration Plate

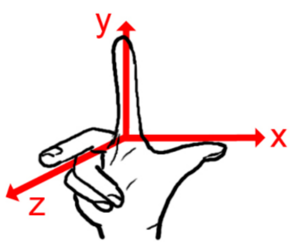
Single sided single plane plates (black dots on white background)

A typical *single sided single plane* (SSSP) calibration plate provided by LaVision looks like this:



Once the appropriate calibration plate is selected in the DaVis calibration dialog the number of marks, their size and distances is known. There are three open circles (fiducials) in an "L" shaped configuration that are detected automatically by the mark search in the calibration dialog to detect the orientation of the calibration plate. These fiducials define the coordinate system of the measurement plane. The first view (of a series of views) defines the z=0mm plane with the origin at the center of the "corner" open circle and the direction of the x-axis and y-axis by the other two circles of the "L" as depicted in the following figure:





In case the cameras are looking via a mirror at the calibration plate, the axis orientation is still as depicted here: x axis from the corner mark towards the circle at the longer arm and y towards the shorter arm of the "L" given by the fiducials.

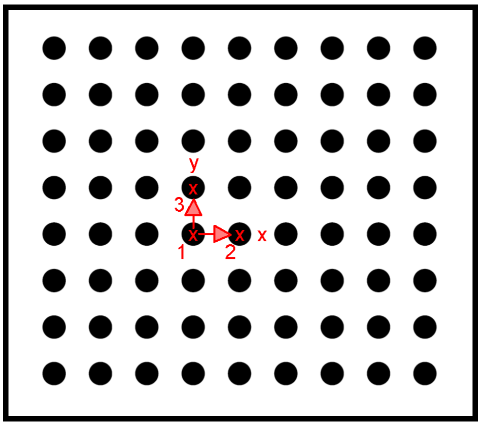
If the plate is tilted or rotated, so is the coordinate system. The orientation of the x- and y- axis can be corrected afterwards using the transformation dialog, see Section 18.8. The position of the z-plane can be adjusted to the measurement plane using the planar self-calibration dialog in some systems (2D-PIV, Stereo-PIV, PTV) while in other volumetric measurement techniques there is a 3D transformation dialog that allows the adjustment of x-, y- and z-axis on projected volume images (Tomographic PIV, Shake-The-Box, 3D-PTV). For these special dialogs please refer to the corresponding product manuals for more details.

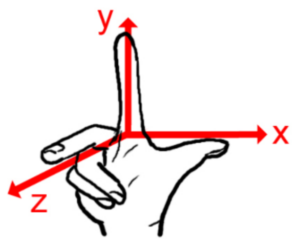
For customized plates without fiducials, or for images where not all fiducials can be detected in the field of view, there is the fallback option of the manual mark search. Here, the start marks are defined by the user moving the mouse cursor and clicking the corresponding marks in the recorded image.

The three marks must be clicked in following order:

1. Origin
2. Direct neighbor mark of the Origin to define the x-axis
3. Direct neighbor mark of the Origin at 90°to define the y-axis

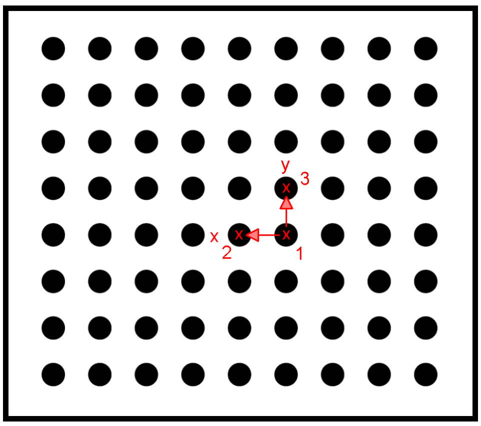
as shown in the following image:





Please note that the selected start marks must be the physically identical marks for all cameras.

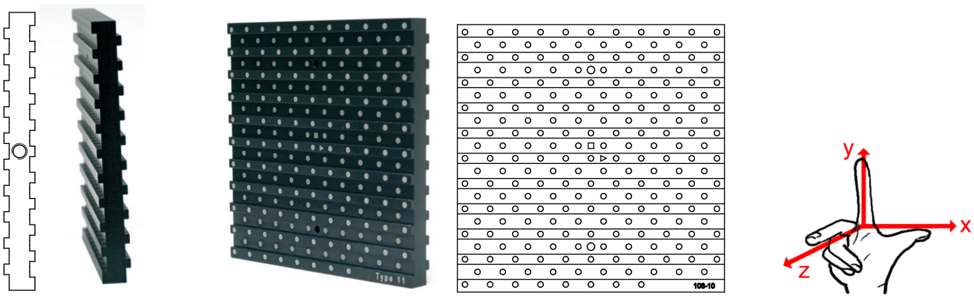
On mirrored images the corresponding cameras need to be specified in the calibration dialog and the three start marks need to be clicked as shown here:



Again, the start mark defines the position of the origin, the other marks define the direction of x- and y- axis of the coordinate system of the corrected image.

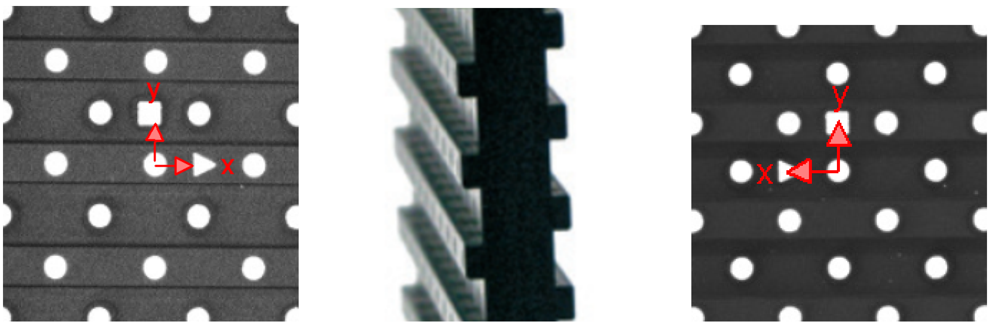
Dual sided dual plane plates (white dots on black background)

A typical *dual sided dual plane* (DSDP) calibration plate provided by LaVision looks like this:



The front side contains the calibration plate type imprinted. Usually the front side of the calibration plate with the 'Type' label on the bottom right needs to be imaged by the (first) camera in order to ensure proper orientation of the coordinate system in the corrected image.

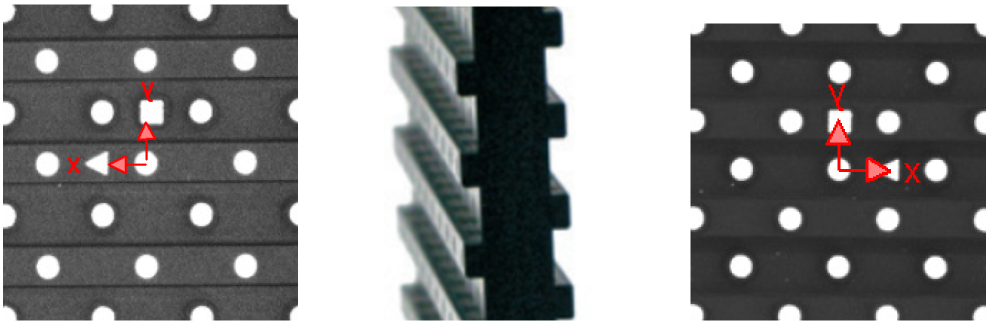
Calibration marks are printed on opposite sides in arrays on two levels. A square and a triangle fiducial marker are found automatically in the automatic mark search, and define the coordinate system of the corrected images as shown in the following figure:



(a) Front side (no mirror). **(b)** Side view. **(c)** Back side (no mirror).

Please note the different orientation of the triangle in respect with the square to distinguish between front and back side of the calibration plate. Also the direction of the x-axis is pointing to the left on the back side and to the right at the front side. A common coordinate system is defined this way. The z=0-plane is the top level plane on the front side.

A mirrored image can also be detected automatically. The cameras are looking through mirror at the measurement plane. It does not need to be selected in the dialog.



(a) Front side (mirrored). **(b)** Side view. **(c)** Back side (mirrored).

For special plates without ficudials or for images where not all fiducials can be detected in the field of view, there is a fallback option of the manual mark search. Here, the start marks are defined by the user using the mouse cursor on the top level plane on the front side and the marks exactly opposite to the marks on the back side (which might be found at the bottom level). The three marks on the front side must be clicked in the following order:

1. Origin
2. Direct neighbor mark of the Origin to define the x-axis
3. Direct neighbor mark of the Origin at 90°to define the y-axis

Please note that the selected start marks must be the physically identical marks for all cameras. The three marks on the back side must be clicked in the order:

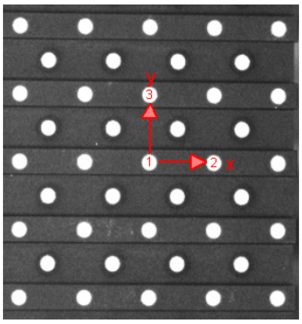
1. Origin
2. Direct neighbor mark of the Origin to define the x-axis
3. Direct neighbor mark of the Origin at 90°to define the y-axis

Please note that the selected start marks must be the physically identical marks for all cameras.

The following figure shows the click pattern for the unmirrored case:

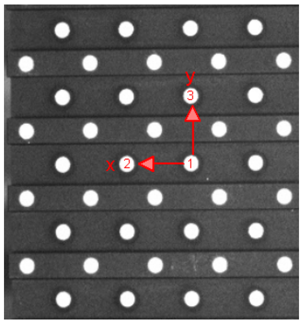
If using both sides of the calibration plate, make sure that the marks are exactly opposite pairwise: mark 1 on front and back side, mark 2 on front and back side, mark 3 on front and back side).

If both sides of the calibration plates are used, the z=0mm plane is the top level plane of the front side. For a PIV setup with limited depth of



(a) Front side (no mirror).

(b) Side view.



(c) Back side (no mirror).

field the plate might be placed centered to the laser sheet. Then, the top level plane will deviate from the laser sheet measurement plane. In this case you would need to specify the position of the first view (that differs from z= 0mm) at half of the plate thickness. However, using the SPIV self-calibration dialog would move the coordinate system with the z=0mm plane on the laser sheet plane without any drawbacks.

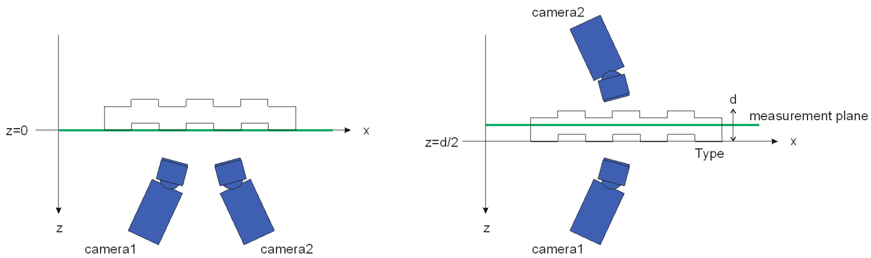
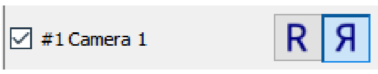


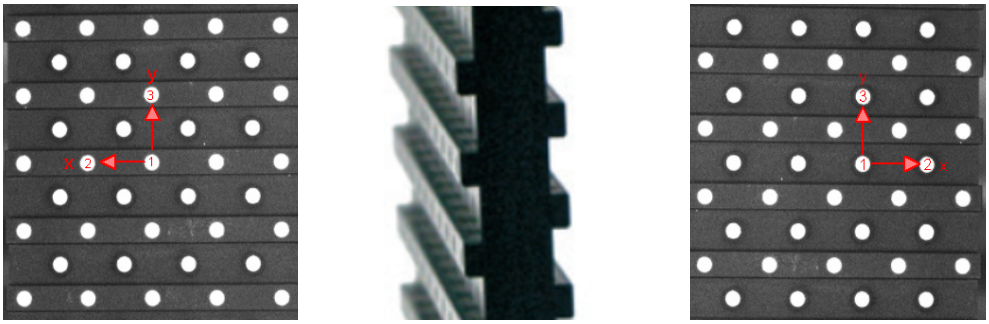
Figure 18.1: Left: Stereo setup with both cameras on the same side with top level of the calibration plate at measurement plane position (in green). Right: Stereo setup with both cameras on opposite sides with the center of calibration plate at measurement plane position and top level at z=d/2 (if d is the thickness of the plate) in order to compensate a limited depth of field.

In case one or more cameras are looking through a mirror at the experiment, specify those cameras in the calibration dialog to distinguish mirrored images from images showing the back side of the plate:



Then, marks should be clicked in the following order:

1. Origin
2. Direct neighbor mark of the Origin to define the x-axis
3. Direct neighbor mark of the Origin at 90°to define the y-axis



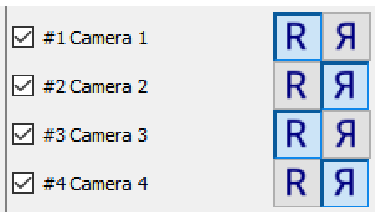
(a) Front side (mirrored). **(b)** Side view. **(c)** Back side (mirrored).

Again, on the front side the top level marks need to be clicked and the marks exactly opposite on the back side (which might lie on a lower level).

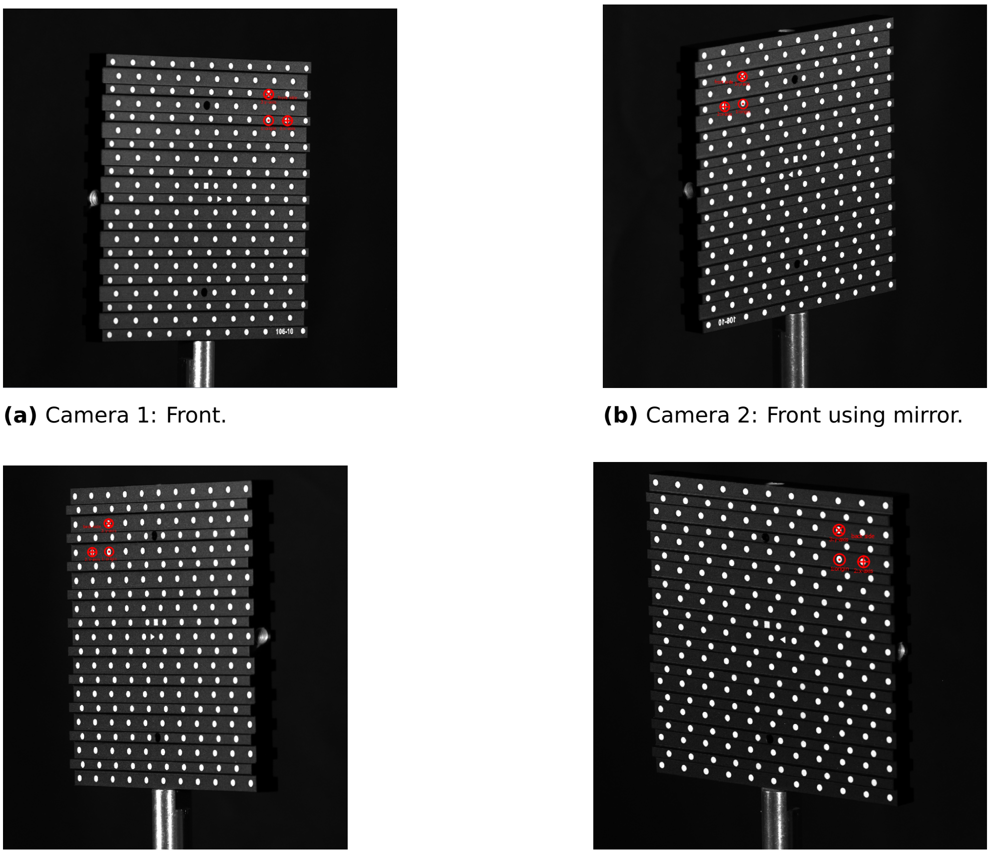
Finally, an example of a calibration with 4 cameras is shown.

- Camera 1 and 2 looking on the front side of the calibration plate
- Cameras 3 and 4 looking at the back side of the calibration plate
- Camera 1 and 3 look directly to the calibration plate
- Cameras 2 and 4 use mirrors and have mirrored images of the calibration plate.

The settings in the dialog should be made like this:



The clicks might be done as shown in the following figure:



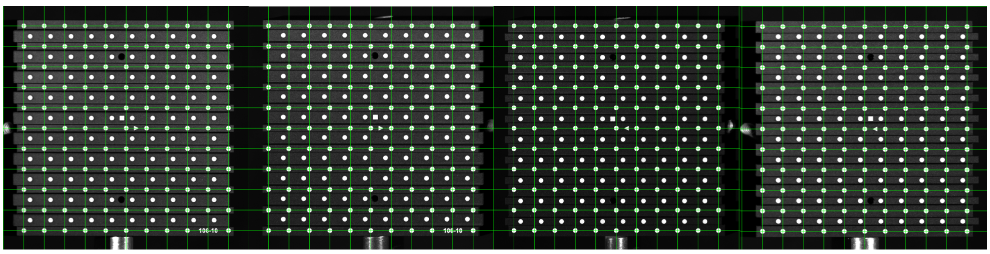
(a) Camera 1: Front.

(b) Camera 2: Front using mirror.

(c) Camera 3: Back side.

(d) Camera 4: Back using mirror.

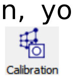
The following figure shows how all camera images look like when corrected for z=0mm. Please note that due to parallax the marks of the second level are not centered between the top level marks that fall on a rectangular grid as expected. This parallax is used for stereoscopic and volumetric systems.



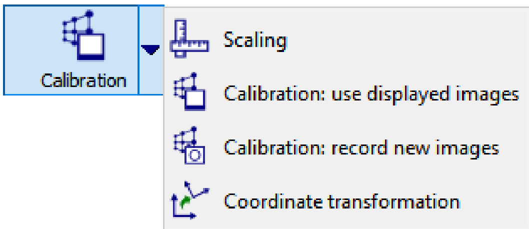
18.2 Calibration dialog

There are two ways to open the calibration dialog.

- If you already have recorded calibration images in your active project, you have to select the image set within the project browser and click the **Calibration: use displayed images** button. 

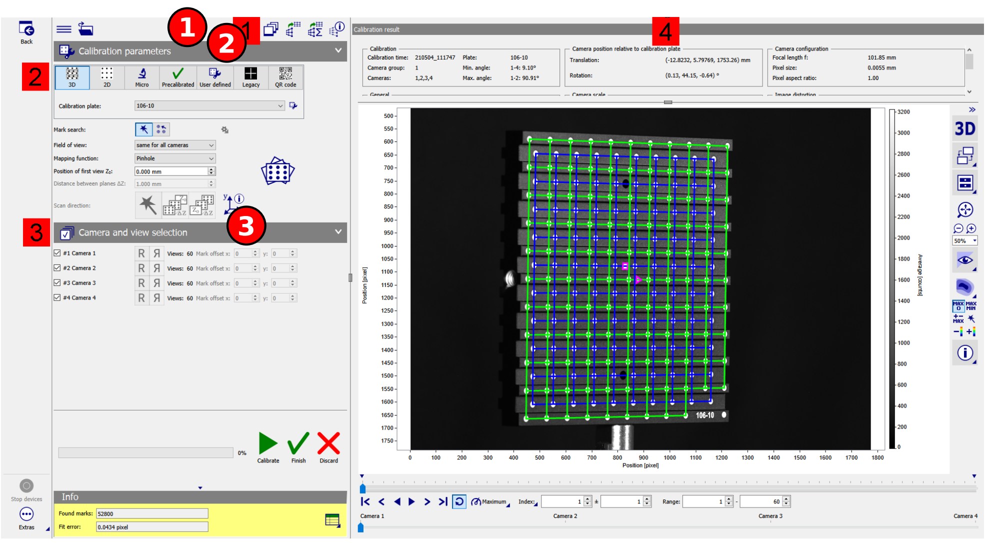
- In case acquisition hardware is connected and you wish to record new images to do your calibration, you have to click the **Calibration: record new images** button. 

You may have to click the triangle next to the calibration button to see all the available calibration options.

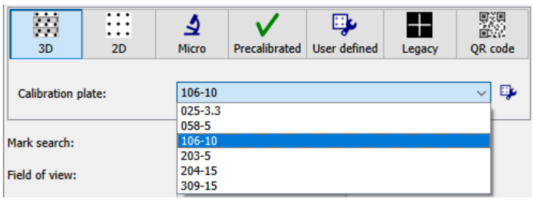


18.2.1 Basic layout

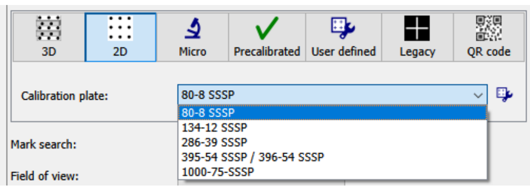
The calibration dialog consists of the following elements, which are explained in more detail afterwards:



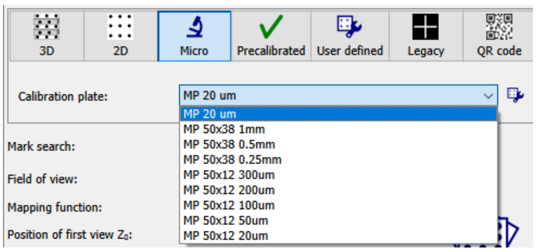
1. **View selection** 1 : selects what is displayed in the view on the right (6)
   - The input images for the calibration process.
   - The corrected images for each view and plane of the calibration plate.
   - The corrected images summed up over all views.
   - The parameter of the calibration result.
   - The parameter of the currently active calibration in the project.
2. **Calibration parameters** 2 : If a calibration plate from LaVision is used, select the type of plate you are using:
   - **3D**: A two level plate (white dots on black ground, square and triangular fiducials).



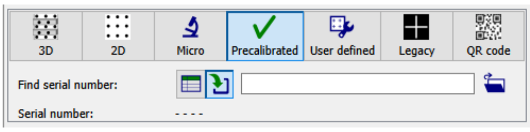
   - **2D**: A one level plate (black dots on white ground and 3 open circle fiductions).



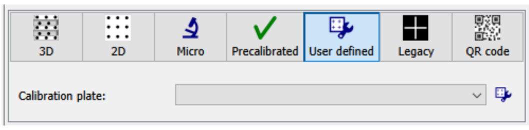
   - **Micro**: : Glass plate with microscopic calibration pattern of black dots.



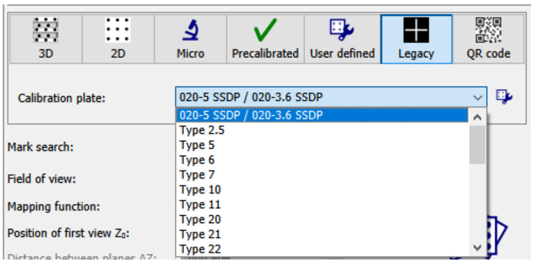
- **Precalibrated**: Calibration plate with measured mark positions of all dots. Either enter serial number of calibration plate, or select the MCP file provided by LaVision.



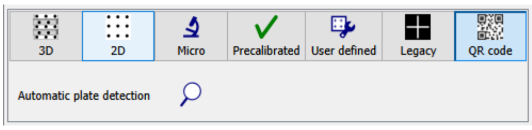
- **User defined**: All defined plates in the calibration plate editor are found here with the specified name.



- **Legacy**: Discontinued LaVision plate types are no longer sold.

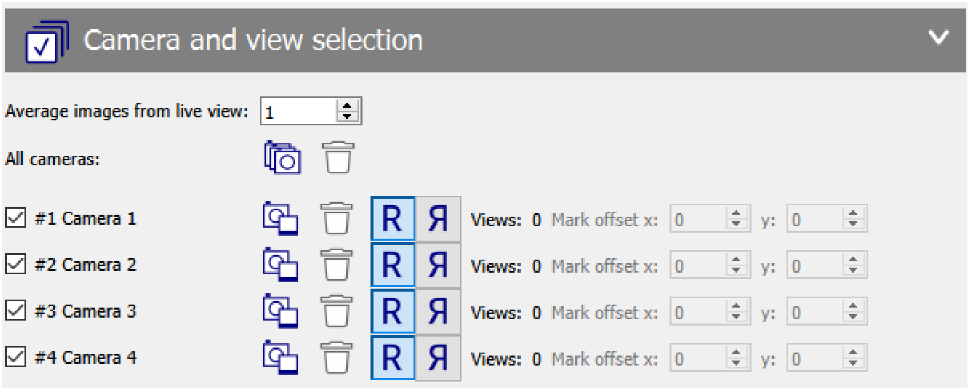


- **QR code**: LaVision type single sided, single level calibration plate with QR code that encodes plate type and dimensions. With this selection the calibration process can be started immediately. It is however necessary to detect the QR code during the calibration process in at least one camera and one view. Alternatively the identification can be done before starting the calibration process pressing the magnification glass button:
  - If you entered the dialog with prerecorded data then the set is scanned until the QR code has been identified.
  - In case of a life recording with the connected hardware the system switches to live recording scanning the camera images takes until a QR code has been detected. No images are stored for the calibration process. It is recommended to have the QR code either fully covering the field of view or to have the entire plate fully covering the field of view to optimize the QR code detection.



Select the appropriate plate in the corresponding section.

3. **Camera and view selection** 3 : defines, which cameras will be calibrated, simply by checking the desired cameras.



To define whether a camera faces the calibration plate directly or through a mirror the buttons R and Я can be used. If automatic mark search is used, these options will be disabled, since it will also be determined automatically. In case **Calibration: record new images** was selected, the following additional options to acquire images are available in this section.

- **Average images from live view**: This number defines how many images will be grabbed and averaged for each calibration view.
- Grabs one calibration view from all available cameras and adds it to the calibration input set. When the image acquisition is finished, the image view on the right shows the acquired images.
- Deletes the last acquired image.
- Allows to acquire images for each camera individually. Once image acquisition was started with this option, images from all remaining cameras must be recorded until the number of views is the same for all cameras.
- Views: Number of views taken for this camera.

- Mark offset this section is used for side by side calibration only (see chapter 18.4).

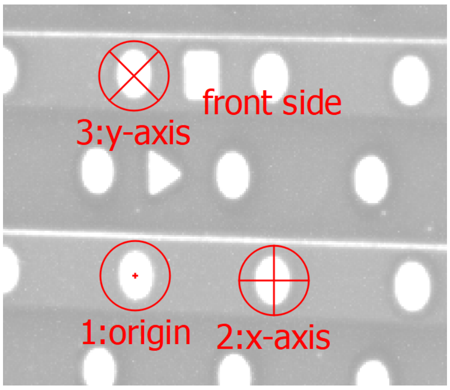
18.2.2 Start calibration

Once all settings are correct and all images are acquired the calibration can be started by pressing **Calibrate**.

In case of the automatic mark search, the software finds marks on all input images and calculates the selected mapping function.

If manual mark search is selected, the user will be guided through all images to place three start marks, which will serve as seed for the mark search. Instructions how to click will be displayed at the top of the input images.

The first mark defines the origin, the second the direction of the x-axis (relative to mark 1), and the third mark the direction of the y-axis (relative to mark 1).



One might wish not to place marks on some images because the calibration plate is not visible in these views. To do so, press ESC and one will be guided to the next image. Once all manual marks are placed, the mark search starts and calculates the selected mapping function. If manual set marks have been placed wrong on some image, navigate to the image in question within the input set and simply move the marks by click and drag. Afterwards, the calibration has to be started again by clicking **Calibrate** again. Now, the mark search for the altered view will be performed based on the moved marks.

When the calibration is finished, the info box at the bottom shows the total number of found marks and the reprojection error. For a detailed overview please click the button. This will open a spreadsheet showing the number of marks and reprojection error for each individual view/camera.

18.2.3 Coordinate system defined by calibration

Z-axis

The direction of the z-axis is defined by the orientation of the calibration plate in the **first** view of the calibration images. It is perpendicular to the calibration plate plane, and the Z=0mm plane is defined by the position of the front side of the calibration plate. In case of a two level plate, the top level defines the z=0mm plane. The coordinate systems are always right handed, so that the z axis is pointing towards the cameras viewing the front side of the target.

X-axis / Y-axis

On the calibrated z=0mm plane, the orientation of the x- and y- axis is defined by the orientation of the calibration plate.

1. **Manual mark search:**

   (a) 1-level plates and 2-level plates:

   The three marks clicked define the coordinate system. Please note that using 2-level plates, the three reference marks MUST be on the top level of the calibration plate with respect to Camera no. 1.

   - 1^{st} clicked mark defines Origin
   - 2^{nd} clicked mark defines (together with 1^{st} click) the direction of x-axis (usually to the right of 1^{st} click)
   - 3^{rd} clicked mark defines (together with 1^{st} click) direction of y-axis (usually to the top of 1^{st} click)

   Please be aware that for any cameras viewing the back side of the target, the clicked reference marks must be diametrically opposite of the marks on the front side of the target. Specifically, the reference marks would be origin-**left**-top, if the marks on the front side are clicked origin-**right**-top. On most LaVision calibration plates, the selected reference marks on the back side will be on the lower level! Use the triangle/square fiducial marks on the calibration plate to help to select the correct reference marks for each of the calibration images for all cameras.

If the camera looks at the calibration plate through a mirror, this information must be specified in the calibration dialog.

2. **Automatic mark search:**

   If the fiducial markers (triangle/square or open circles) are visible in the image, no manual mark clicking is necessary. **DaVis** will find all marks automatically and decide the origin and the orientation of the x-/y- axis based on the orientation of the calibration plate.

   (a) **2-level plate:**

   The two level/two-sided plates contain triangle/square fiducial marks that allow **DaVis** to recognize front and back side. Please note that the triangle on the back side has a different orientation pointing towards, rather than away from the square.

   Holding the calibration plate so that the plate type can be read (only printed on the front side), defines the coordinate system: x-axis pointing to the right, y-axis up in alignment to rows/columns of marks and z-axis pointing towards the camera. The origin is left to the triangle, and the z=0mm plane is defined by the top level plane on the target.

   If the plate is rotated/inclined, so is the resulting coordinate system. You can use the **Coordinate Transformation** dialog to change the orientation of the coordinate system, if desired.

   (b) **1-level-plate:**

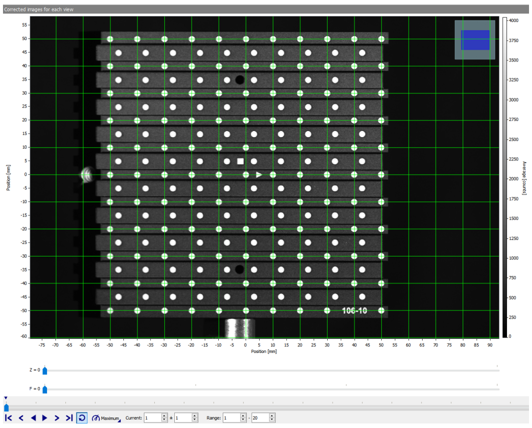
   The fiducials (open circles in an “L” distribution) define the coordinate system. The corner circle is the origin, while the other two circles define the orientation of the x-axis and y-axis.

   Holding the calibration plate so that the plate type can be read, defines the coordinate system: x-axis pointing to the right, y-axis up in alignment to rows/columns of marks, and z-axis pointing towards the camera.

If the plate is rotated/inclined, so is the resulting coordinate system.

18.2.4 Evaluation of the calibration result

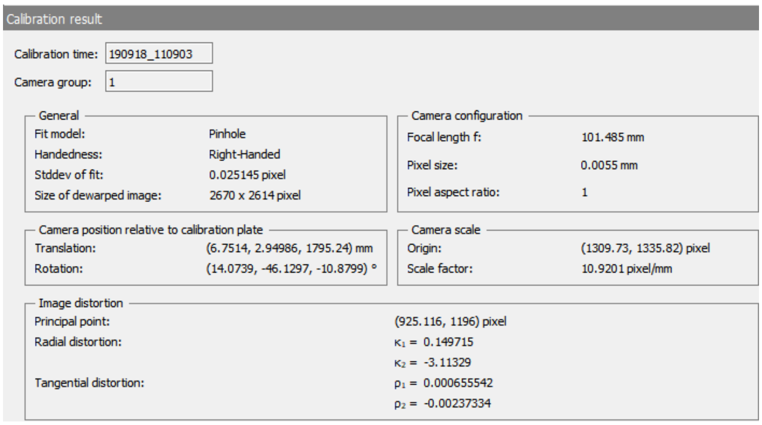
To evaluate the fit result visually, images of all camera/view/z-level combinations are corrected. To view these images click the button in the **view selection**. Also an image is created where all of corrected images are overlayed/summed up to check that all corrected images coincide. To view these images click the button in the view selection.



A grid of the ideal regular grid of marks is overlayed in green. This grid should pass through the center of all marks. The result can be checked:

- In single camera images if the marks fall to the ideal regular grid.
- If the marks in the summed up images of all views for all camera coincide for z = 0mm position.

In case of a 3D 2-level calibration plate, the same can be checked for the position of the second plane, too. Please note that, if there are coincidence marks for one level, the marks for the other level will be mismatched when superimposing different cameras due to their different viewing directions (parallax effect). This parallax can be used to check the 3D resolution of the experiment later. Let the distance *dz* between the 2 levels be 1mm. Then observe the mismatch in pixel for the mismatched plane - which we will assume here to be 5 pixels apart. As the deformation calculation can resolve movements of 0.05 pixel shifts on the CCD images shifts are detectable as low as 0.05 pixel/5 pixel x 1mm $\approx$ 10 μm in z-direction for the current camera setup. This resolution can be increased by increasing the angular separation of the system. (Attention: this will also reduce the resolution in x-direction. An angle of 30°- 60°is recommended). Clicking the button, shows a summary of all important fit parameter. In case of a pinhole calibration, parameters like focal length, translation and rotation can be used as a consistency check. So please check, if these values resemble the physical camera setup.



18.2.5 Finish calibration

When the achieved calibration is fine, it can be accepted by pressing **Finish** . This will store a new calibration in the Properties/Calibration folder of the project and make it the active calibration while moving the previously active calibration to the history folder.

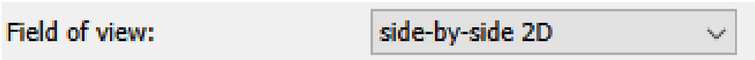
If one is not content with the calibration result, the **Discard** button can be used to reject all results.

18.3 Side-by-side calibration and image / vector field stitching

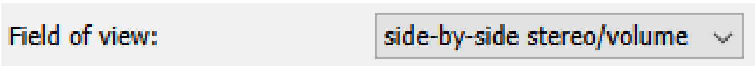
If several cameras are used in a side-by-side configuration - or a side-by-side Stereo configuration - an appropriate calibration is needed to define the coordinate system and the relation of all cameras to each other.

The goal of this type of calibration is to get a larger field of view using more cameras. The area of all corrected cameras images should fill one common view. In contrast to a Stereo setup where the intersection is taken in this kind of calibration the union of all places visible by individual cameras should determine the boundary of the corrected "world" view.

The requirement on the experimental side is to have a calibration plate that spans the whole field of view visible by all cameras. An image overlap is not required. On the software side select the field of view option side-by-side 2D



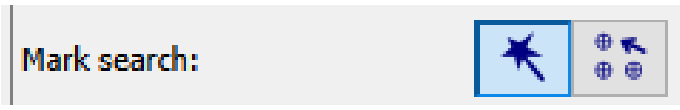
Or the option side-by-side stereo/volume.



The selection influences the way the boundary box (union of all corrected images) is chosen. In the 2D option all pixels fill be inside the corrected images that are visible by at least 1 camera. In the stereo/volume case at least two cameras must "see" the pixel.

You need this klind of calibration to use specific options later (e.g. image stitching and vector stitching).

18.3.1 Side-by-side configuration with overlap using automatic mark search



If there is an overlap region visible in all cameras and the fiducial markers can be seen in all camera images in the first view the automatic mark search can be selected and no further user input is needed. The offset of each camera to the field of view is automatically extracted from the fiducial positions.

Alternatively the procedure mentioned in chapter 18.3.2 using the manual mark search can be also used if the fiducials are visible.

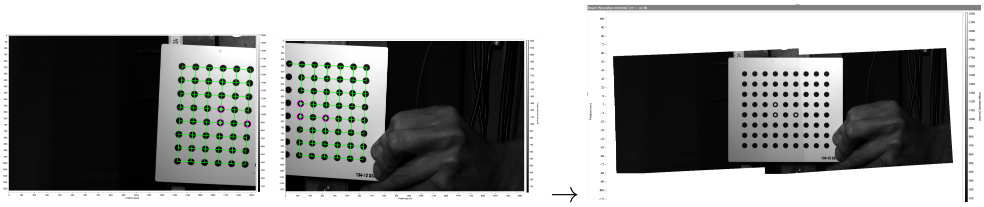
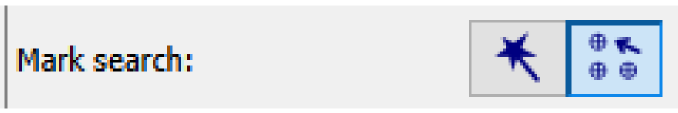


Figure 18.2: Automatic mark search on camera 1 and 2 and the resulting corrected image stitched together.

18.3.2 Side-by-side configuration using manual mark search (also possible without overlap)



In case there is no overlap a manual mark search needs to be done. The following is also valid if there is an overlap region but the user decides to do a manual mark search for other reasons.

For the manual mark search three marks must be clicked in each camera view as mentioned in chapter 18.2.3 section manual mark search. The three clicked marks define the coordinate system marks

1. first mark is the origin
2. the second mark clicked to the right defines the x-axis
3. the third clicked mark to the top of the center mark defining y-axis

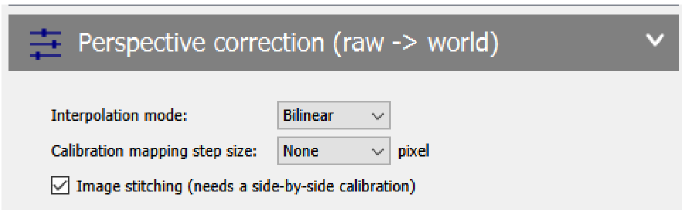
Three marks must be clicked in each camera view.

How ever in the side-by-side calibration procedure those three clicked marks do not necessarily need to be the same physical three marks on the plate for each camera view. If there is no overlap region this is not even possible. Instead the offset to a common reference mark needs to be defined. It is easiest to define this offset to be (0, 0) for the first camera (which becomes the reference mark now) and define the offset in mark positions for the other cameras (+1 counting to the right and to the top, -1 counting to the left and to the bottom) in section 3 of the dialog as shown below

During the calibration the offset is used to calculate the camera offset in the field of view in mm and define the calibration model accordingly.

18.3.3 Image stitching using side-by-side calibration

This special type of calibration is recognized by some processing operations as e.g. "Perspective correction (raw -> world)". Correcting a multi-frame image choosing the "Image stitching (needs a side-by-side calibration)" option results in a single frame corrected image stitched from all corrected camera image into one result. Intensity values of pixels in overlap regions visible by more than one camera are averaged.



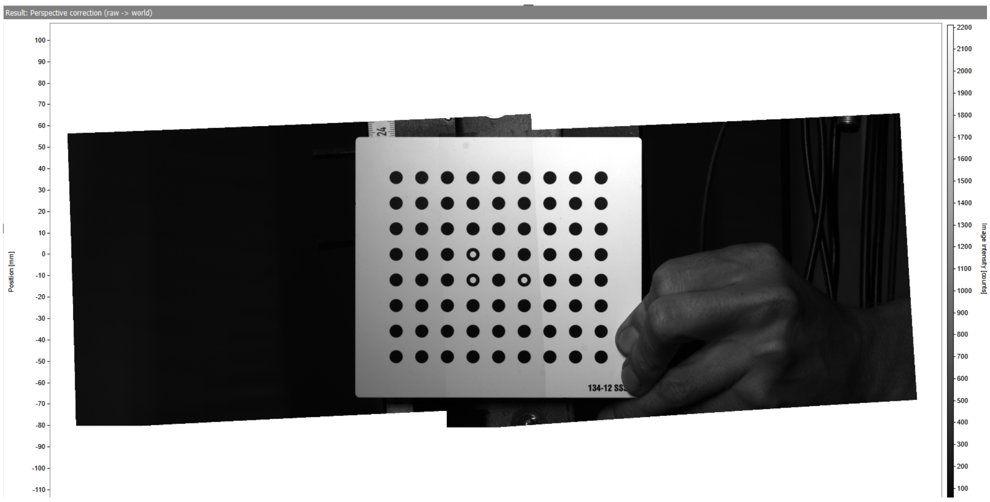


Figure 18.3: Result of the image stitching process.

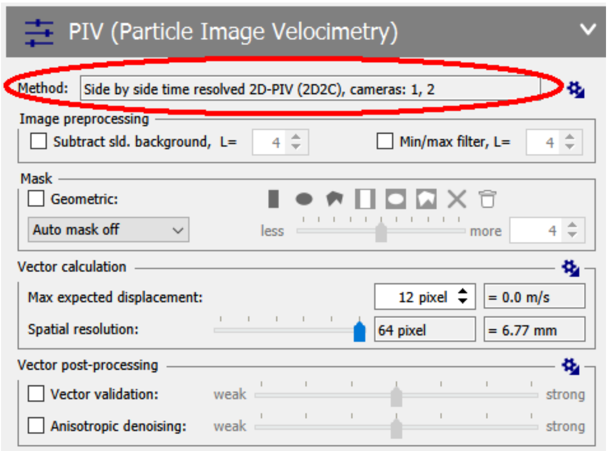
18.3.4 Vector stitching using side-by-side calibration

This special type of calibration is recognized by the PIV processing (refer to FlowMaster Manual for details) where an additional option Side-by-side PIV processing is available. The selection depends on the kind of calibration and may include Stereo PIV.

The result of such a vector calculation results in a single frame vector field.

For the 2D PIV case all vector world positions are checked if they are visible in each camera raw image. All cameras that "see" this position contribute with a calculated vector to the average vector stored at that position.

For the Stereo case all vector world positions are checked if they are visible in each camera raw image. A vector is calculated if that position is visible by at least two cameras. In that case all 2D vector shifts of each of the available cameras (two or more) are triangulated in one fit to get a 3D vector which is stored at that position.



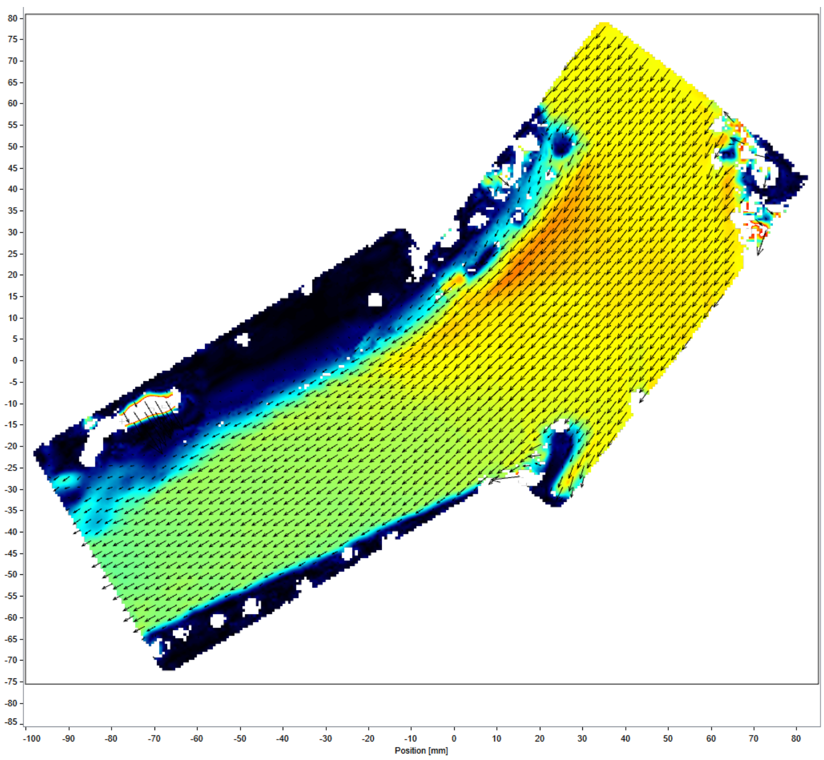


Figure 18.4: Example of a stitched vector field.

18.4 Example of a typical calibration workflow

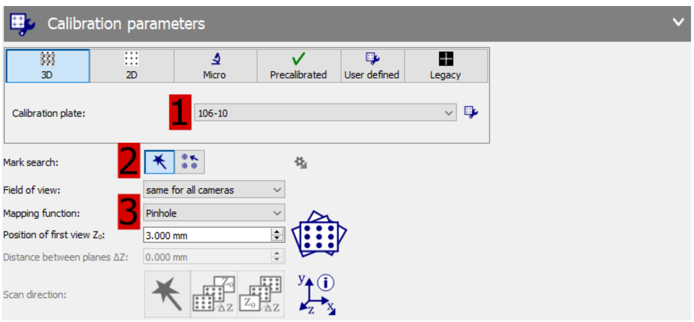
1. Select the used type of plate from the Calibration plate list in the corresponding section 3D, 2D, Micro, Precalibrated, User defined, Legacy. For this example it is the 106-10 calibration plate in the section 3D.

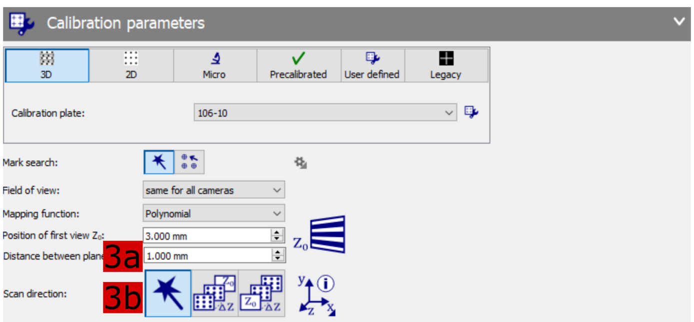
2. Depending on the calibration plate decide between automatic or manual mark search. Automatic mark search should work in most cases.

3. Depending on your physical setup, select the appropriate **Mapping function**. (see section Limitations and Requirements for Pinhole / Polynomial Mapping Function.)

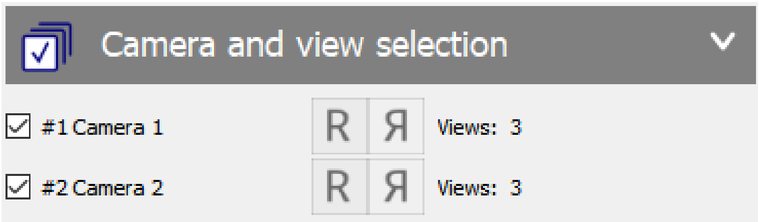
   (a) In case of a polynomial scan define the distance between each calibration view.

   (b) In case of a polynomial scan define the scan direction. Automatic mode should determine the direction correctly in most (not microscopic) cases.

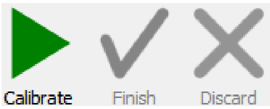




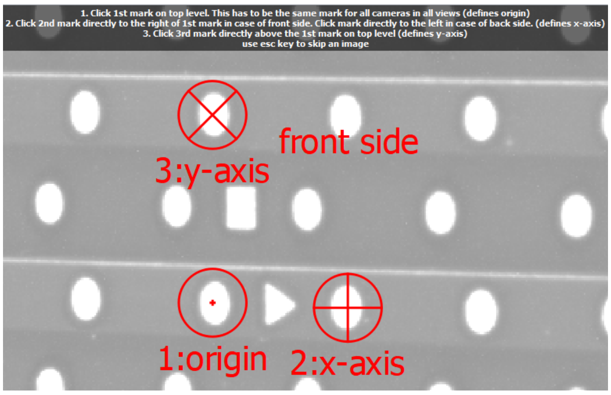
4. Select all cameras that should be calibrated.



5. Start the calibration process by clicking **Calibrate**.

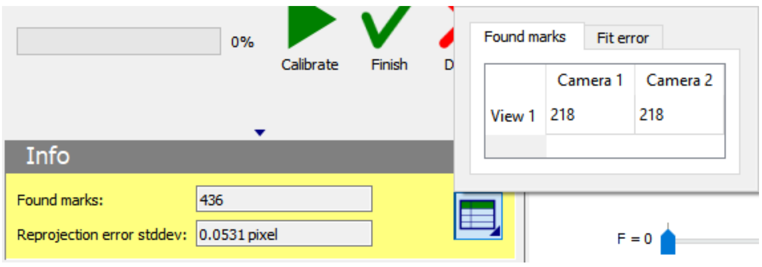


(a) If the manual mark search was selected, define three marks for each calibration image according to the information given on the screen.

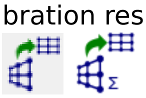


6. As soon as the mark search and calibration process has finished, the calibration result can be verified.

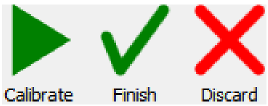
(a) Check the reprojection error and the number of found marks in the info section at the bottom.



(b) Check the corrected images (see section Evaluation of the calibration result).



7. Accept the calibration by clicking the **Finish** button.



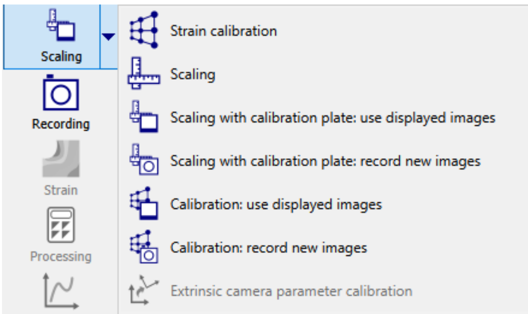
18.5 Limitations and Requirements for Pinhole / Polynomial Mapping Function

A **pinhole calibration** covers the whole volume visible in cameras, but needs an undisturbed optical access through air. Furthermore it is *not* necessary to put the calibration plate in place so that it covers the whole image. The pinhole calibration works with a single plane of marks, but accuracy greatly increases if you use a 2-level plate (2 planes separated by a known distance *dz*, like the one provided with the **LaVision** systems). It is recommended to have 3 views with tilted calibration planes to minimize mapping errors over the volume.

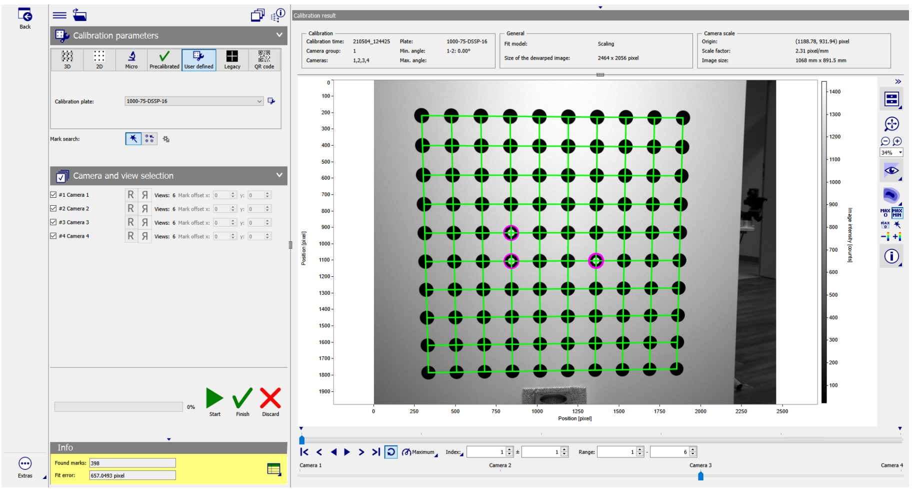
The **generic polynomial 3rd order function** works with arbitrary distortions contained in images, e.g., because the optical access to the experiment is blocked by a glass window that adds extra distortion. It is only possible to calibrate a volume with the generic polynomial 3rd order function using 2 or more equidistant coplanar planes. The **LaVision** 2-level plate will suffice. But as the distortion outside the volume covered by the calibration plate is extrapolated extra errors in the results dependent from the distance to the calibration plate are possible. If using multiple views, the calibration plate must be put in equidistant and coplanar positions. It is advisable to use a translation stage to do this.

18.6 Scaling with calibration plate

Please not that this dialog is not available in all project types.

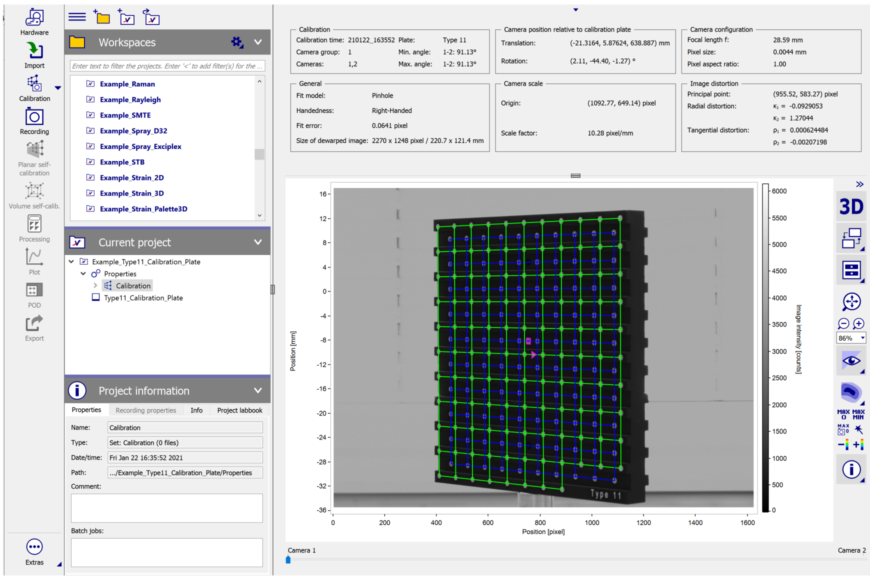


The dialog is meant to have a simple and reproducible scaling for a one camera system looking perpendicular to the measurement plane where the calibration plate is positions to extract the scaling information (factor pixel/mm) without any user interaction: This way sources for error are minimized and a high reproducibility can be achieved.

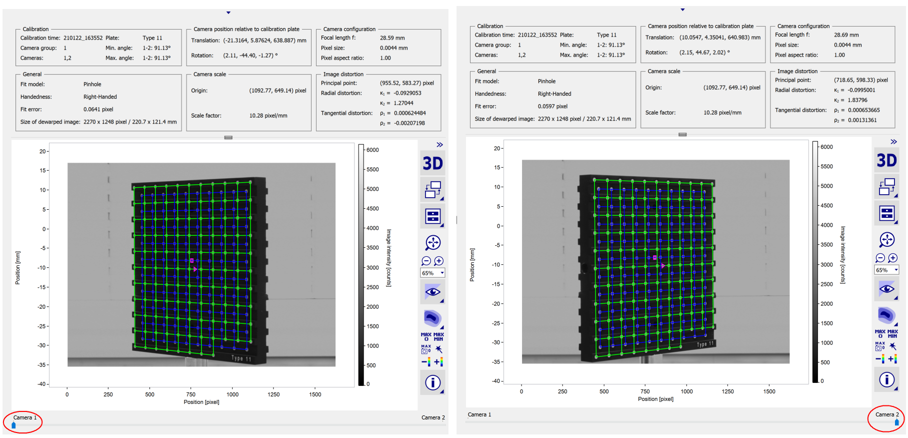


18.7 Calibration result view in the Project Manager

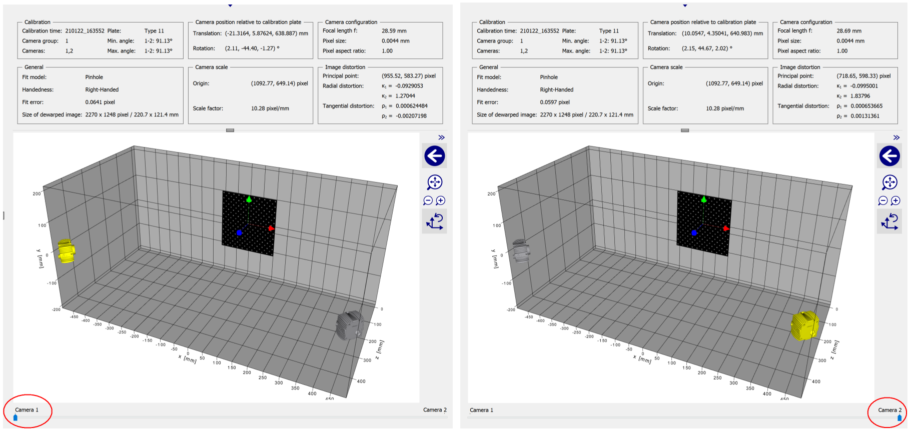
When the **Properties/Calibration** entry is highlighted in the Project Manager on the right side the currently active calibration is shown.



Switching the cameras via the camera slider on the bottom right the pinhole or polynomial parameters are shown as well as all views of the calibration plate with the marks found on all images.



In case of a pinhole model the user can additionally switch to a 3D view pressing the **3D** button in the tool bar to the right of the calibration plate images. This 3D view shows the relative position of the cameras to each other and to the calibration plate in a mm-coordinate system. Thus the camera setup can be checked for consistency. Again switching the cameras via the camera slider on the bottom right the corresponding pinhole parameters are shown as well as the selected camera is colored in yellow for better identification.



Inside the 3D view moving the mouse while pressing the left mouse allows to rotate the 3D view, while holding the SHIFT key on the keyboard at the same time moves the shows scenery. Using the mouse wheel zooms in/out of the 3D scene. Pressing the key x, y or z shows the system as seen from the x-, y- and z- axis. Pressing the back button  leaves the 3D view and returns to the 2D view of the recorded camera images.

18.8 Coordinate transformation dialog

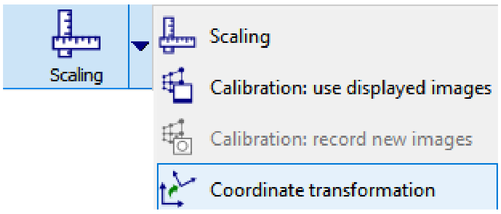
After the perspective calibration has been successfully created
(→ Calibration Dialog) the coordinate system can be further altered:

- The mm-Origin position in the corrected image can be redefined
- The pixel/mm scaling can be reset, which readjusts the corrected image size (or vice versa)
- The corrected image size can be cropped/expanded to contain fewer or more pixels on each side
- The direction of the x- / y-axis can be redefined (which rotates the corrected image)

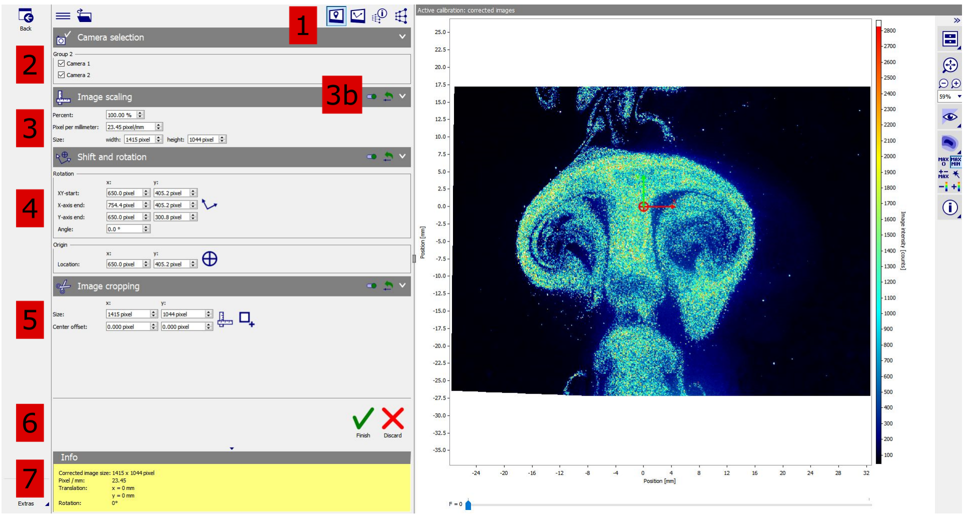
And any combination of the above.

Open the dialog by selecting a set of recorded images and press the tool bar button **Coordinate transformation**.

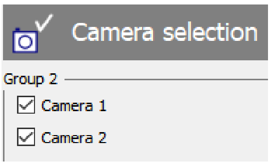
If a different entry is preselected, the choice can be switched by clicking on the triangle symbol on the right side where the **Coordinate transformation** entry can be selected. This becomes the new default entry for that button.



Once the dialog opens, it will look like this:

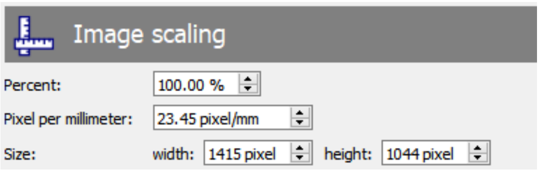


1. The **View selection** selects, which images are shown in the view on the right
   -  The images corrected with the original calibration
   -  The images corrected with the adjusted calibration
   -  The calibration parameters for the adjusted calibration
   -  The calibration parameters for the original calibration
2. In the **Camera selection** section the cameras are selected to which all the changes to the calibration are applied. For a Stereo / Volumetric system make sure that all transformations are applied to all mapped cameras. Otherwise, you will have an inconsistent group of cameras.



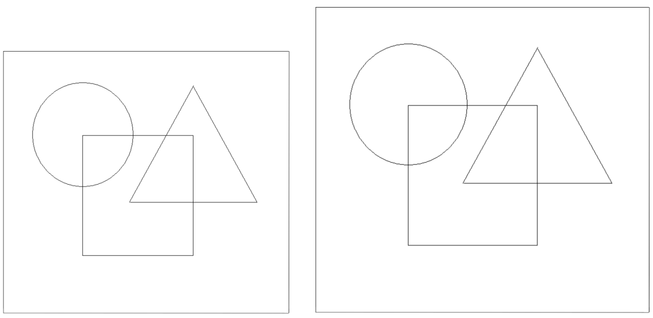
In the example above both camera 1 and 2 are selected, and all cropping/scaling/rotation adjustments are applied to the perspective calibration of both cameras.

3. In the **Image scaling** section the pixel size / corrected image size can be adjusted while maintaining the same field of view:



There are three ways of changing the current scaling:

- The resizing factor can be given as a percentage value based on the original scaling (100% = no change). This will change the pixel/mm factor, and the corrected image size in pixel accordingly. The field of view (of 'what is seen in the corrected image') stays the same.
- Alternatively, the absolute value of the new pixel/mm scale can be entered, which changes the **Percent** value accordingly, and changes the corrected images size in pixel.
- Or the width (or height) of the corrected image is adjusted, which changes the height (or width), the **Percent**, and the corrected images size in pixel accordingly.

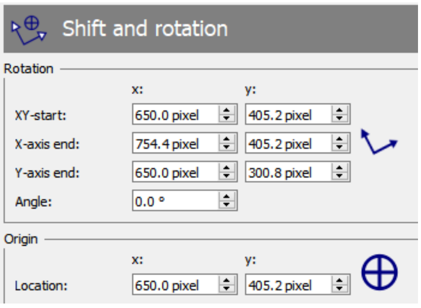


(a) Original **(b)** Adjusted

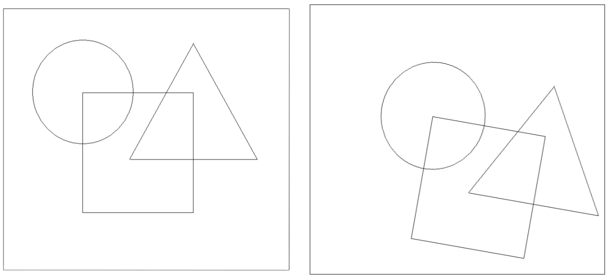
All changes are displayed on the right immediately, if the view selection is set to the second option

3b. Please note that the effect of each section can be switched off and on using the slider on the right side of the section. Furthermore, if you are unsure of the changes you have made, you can restore the settings of the original calibration clicking the left arrow button, which in this section resets the **Percent** scaling back to 100%. The same applies for the other sections.

4. In the **Shift and rotation** section the direction of x-/y- axis can be adjusted (the image is rotated). Also the position of the origin can be set to a different position.

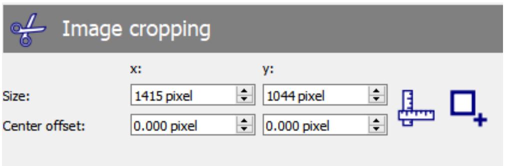


The values can be entered directly or changed by clicking on the spin box up/down arrows. Pressing the define coordinate system button , allows to define two points in the corrected image by click-drag-release, which defines the x-axis for the rotated, adjusted coordinate system. Both x-axis and y-axis tip of the coordinate axis can be readjusted with the mouse as well as the coordinate system base point. Clicking at the center of the axis allows to shift both x- and y-axis to a new position without changing the angle. The XY-start (the coordinate system base) does not change the mm position of the **Origin** (for that option see below). The resulting angle is automatically updated in the bottom **Angle** spin box. Alternatively, the angle of the x-axis can be changed here directly in degrees (0° horizontal direction to the right), which updates the position of x-/y- axis end coordinates automatically. Finally, the mm-Location of the **Origin** can be changed by entering the numbers manually in the spin boxes or by pressing the Origin button . After the origin has been defined, it can also be adjusted using the mouse on the overlay item in the image and move it to a different location, which updates the spin box values automatically.

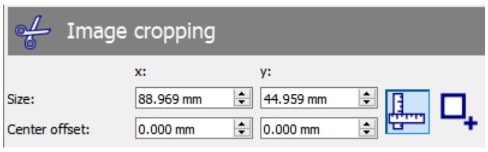


(a) Original **(b)** Adjusted

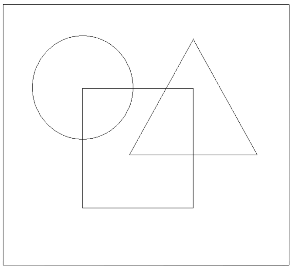
5. In the **Image Cropping** section the field of view can be adjusted. It changes the boundaries of the image by extending or restricting the field of view (what is visible in the corrected image).



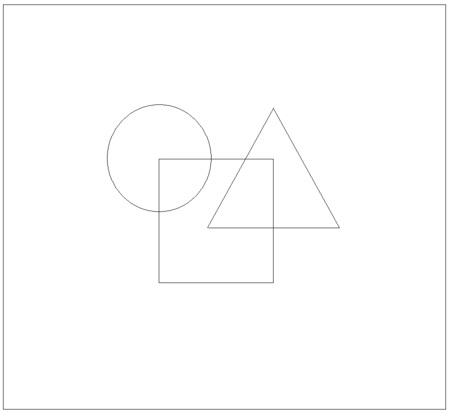
The width and height of the corrected image can be extended or restricted directly by entering the numbers manually. This will extend or reduce the field of view on opposite sides maintaining the image center while not changing the pixel/mm scaling factor. Also, the field of view can be shifted into one direction by entering the **center offset** manually. Pressing the **toggle** scale button  changes the number entry from pixel to mm.



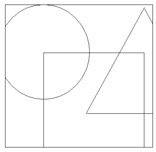
Pressing the **Set Rectangle** button  allows to set the field of view with the mouse by click-drag-and release a rectangular using the left mouse button. Please note that this enables also to extend the field of view, when zooming out first and selecting an area outside the visible image. The size can also be adjusted with the mouse clicking on the rectangle corners, or moved around clicking on the edges or the inside of the rectangle. If you have the view set to the second option, the current changes are shown immediately as a preview.



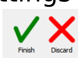
(a) Original



(b) Extended field of view



(c) Cropped field of view

6. Once all settings are correct you can accept the adjustments pressing **Finish** , which will store a new calibration in the Properties/Calibration folder of the project and make it the active calibration, while moving the previously active calibration to the history folder. Or you can **Discard** all settings, which will reset all entries to the original value (100% size, no shift, no rotation, and the original field of view) to start over again.

7. Please check the **Info** section at the bottom of the dialog to see what changes have been made to the original calibration before you store the calibration to the project.

Finally, using the Load set button  on the top left of the dialog allows to switch to a different input set without leaving the dialog, e.g. to check for different experimental conditions.

19 Camera Setup Wizards

19.1 Focal length calculator

Prior to setting up the camera, you may use the help menu to assist you in choosing the optimal camera working distance for a desired field of view (measurement area).

1. Select the menu item **Extras – Wizards – Focal length calculator** to open the dialog in Fig. 19.1.
2. Select the camera type used. This defines the size of the CCD sensor.
3. Enter the **field of view** to be monitored by the camera as size of the sample under study.
4. Enter the distance between camera and sample.
5. Press the button **Calculate** to find the **focal length** at which the sample optimally fits into the image.

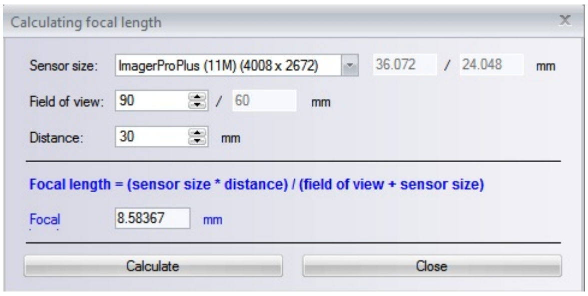


Figure 19.1: Focal length calculator

19.2 dt / velocity calculator

The dialog in Fig. 19.2 is useful to

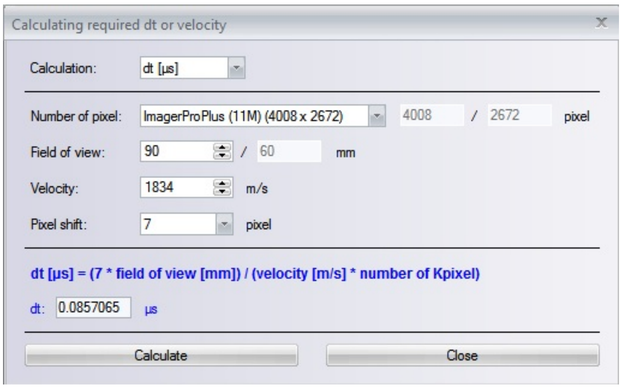


Figure 19.2: dt/velocity calculator

- find the time difference **dt** between two exposures of double-frame exposures for a given **velocity** (the expected mean velocity of the flow)
- or find the best **velocity** for a given **dt**.

1. Select the property to be calculated (**dt** or **velocity**).
2. Select the type of the camera used. The camera size is given in pixel.
3. Enter the **field of view** to be monitored and the **pixel shift** of the particles, which should be seen by the given **dt** or **velocity**.
4. Press the button **Calculate** to find the solution.

19.3 Synthetic image generator 2D

The dialog in Fig. 19.3 is useful to create a set of example images – single-frame or double-frame – if you have no hardware connected or no experimental setup available. It simulates a camera by creating a series of images with moving structures that behave over time as specified in the dialog. The ground truth (the vector field describing the pattern movement) is created with the images for later reference.

Open the dialog in **Extras – Wizards – Synthetic image generator 2D**.

The lower part of the dialog shows the created images on the left and the corresponding vector fields on the right.

In the upper part of the dialog there are four groups of parameters to be set: storage mode, simulated camera, particle, and vector field parameters.

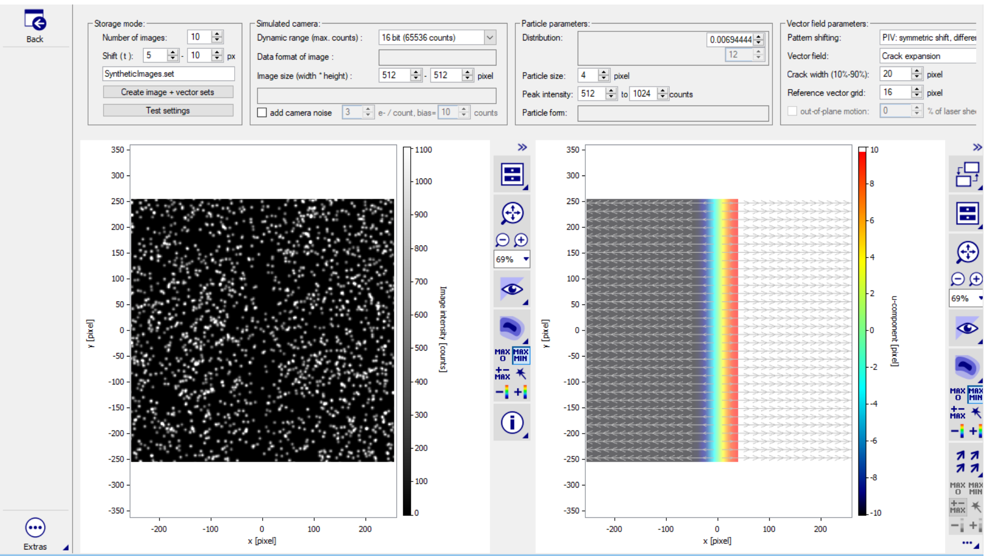
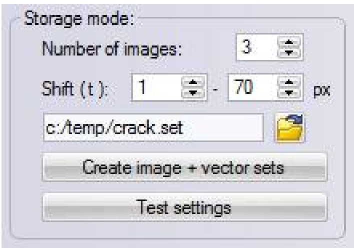


Figure 19.3: Synthetic image generator 2D

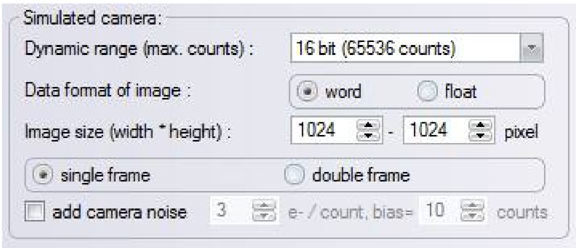
Storage mode:



- Specify the **number of images** to be created.
- The minimum and maximum pattern **Shift(t)** over time is defined for all images of the time series. Examples:
  - Setting both minimum and maximum shift to 10 and using double-frame images (refer to next section) will produce a series of double-frame images, each with a (max.) 10 pixel movement of the pattern between frames.
  - Setting this to 0 and 10 and using single-frame images (refer to next section) will produce a series of single-frame images describing an accelerated pattern movement, from images containing a 0-pixel movement (first image) to images containing a 10-pixel movement (last image) of the pattern.

- To specify where the result will be saved, press the **Open folder** button to select a new destination for the image set. If the name of the set already exists, a suffix will be added (`name.set` → `name_`*XX*`.set`, with an increasing number *XX*) to prevent overwriting existing data. Below that set a set of the name `vector.set` is created, in which the corresponding vector fields are stored as reference data.
- To check the parameter settings, press **Test settings**. This temporarily creates just one image and one vector field (using the specified maximum shift), which are not saved to your hard disk.
- To create and store the data, press the **Create image and vector sets** button.

Simulated camera:

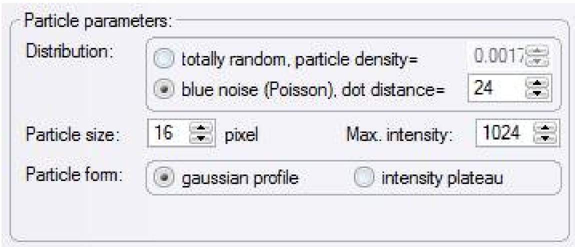


In this group of parameters, the type of the simulated camera is defined:

- The **Dynamic range (max. counts)** defines the maximum number of counts of the simulated camera before it saturates. There will be no higher intensity in the image.
- The **Data format of image** is either **word** resembling a real digital camera holding only integer count values or **float** with interpolated intensity values.
- The **Image size (width * height)** is the number of pixels of the images (camera chip size).
- Define if you want to simulate a **single-frame** or a **double-frame** camera. With a simulated double-frame camera a new random pattern is created for every (double-frame) image. For a single-frame camera, one random pattern is used for the whole time series to simulate time-resolved recordings.

- You can add simulated camera noise by checking the **add camera noise** checkbox. The parameters are:
  - the e^-**/count** conversion factor M/N: This defines pixels holding an intensity of N counts to be equivalent to M electrons. These would show an average noise of $\sqrt{M}$ electrons, which is reconverted to counts. The Poisson distribution of the real photon noise is approximated by a Gaussian distribution. A typical value of a real cameras is 5 e^-/count.
  - the thermal noise level, a background level Gaussian-distributed over all pixels (typical real value: 10).

Particle parameters:



In this group box, the pattern properties are defined:

- In **Distribution**, the spatial location of all particles is defined.

  Totally random will lead to random clustering of particles. This is more appropriate to simulate PIV seeding particles. You can define a particle density (particles per pixel). A typical value is 0.05 for a particle size of 3–4 pixels. Larger particles require smaller densities so that they do not overlap too much.

  Blue noise distribution is a random pattern where particles / pattern dots have a minimum distance to each other to prevent overlapping of particles. Typically, this value is a little larger than the particle size, e.g., 12 for particles sized 8 pixels. These dots are randomly distributed so that there is no preferred direction.

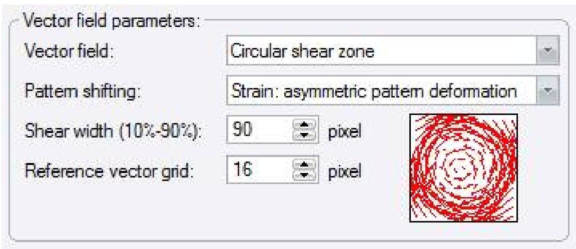
- **Particle size** is the size of each particle in pixel.
- **Particle form** is the radial intensity distribution of the particle image.

Intensity plateau creates a step-function-like intensity profile with a rather narrow fall-off to background intensity, resulting in dots of homogeneous intensity.

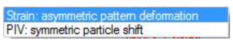
Gaussian bell approximates the look of typical PIV seeding particles. The radial intensity profile is a Gaussian distribution with the width defined by particle size.

- The **Max. intensity** is the highest intensity value of a *single* particle. Due to overlapping of particles, intensities will add up until the camera saturation is reached. For particles with a plateau distribution of intensity, you can avoid this by choosing the dynamic range (max. counts) of the camera (see above) for the maximum intensity. Then the plateau already saturates the camera and dots will melt rather than add up.

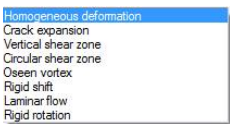
Vector field parameters:



Here the pattern /particle shift over time is specified. You have the choice of several distinctive behaviors:



- For **Pattern shifting** there are the two modes Strain and PIV:
  - In **PIV**, only the simulated particles are moved and maintain their individual intensity profile (like moving seeding particles in air or water). In double-frame mode, the shift is symmetrically distributed ($-V/2$ in the first image and $+V/2$ in the second frame).
  - In **Strain**, the pattern of each dot is smeared with the shifts defined by the vector field and become distorted. In double-frame mode, the shift is only present in the second frame.



- In the **Vector field** selection, the following choices are available, each of them with a parameter to be set:

Homogeneous deformation: This is a simulated expansion of a sample with a homogeneous stretch in y-direction. The size of the undistorted region can be set.

Crack expansion: This is a simulated crack opening with a nonlinearly growing stretch in x direction away from the image center. The size of the region that reaches 10 % – 90 % of the maximum shift can be set.

Vertical shear zone: This is a simulated shear zone with a nonlinearly growing shear in y direction with the maximum in the image center. The size of the region that reaches 10 % – 90 % of the maximum shift can be set.

Circular shear zone: The inner circular part performs a clockwise rigid rotation, the outer part rotates anti-clockwise, with a smooth (nonlinear) transition zone in between. The width of the region that reaches 10 % – 90 % of the final shift on either side can be set.

Oseen vortex: This is a simulated Oseen vortex, which models a line vortex that decays due to viscosity. The approximate distance from the vortex center at which the vorticity is maximal can be specified in pixel.

Rigid shift: This is a rigid shift. The parameter is the direction in degrees of that shift.

Laminar flow: This is a simulated laminar flow through a pipe in x direction with a parabolic velocity profile in y direction. The parameter specifies how far outside of the image the wall of the simulated pipe is lying.

Rigid rotation: This is a rigid rotation. The parameter is the distance of the rotation center in $-x$ direction from the center of the image.

- The **Reference vector grid** for the ground truth vector field is defined to be able to compare it to results calculated from the images more easily.

20 Single image/vector field dialogs

All these dialogs can be accessed via the context menu on the set display in the project browser main dialog. It takes the currently shown image or vector field as source to the dialog. Leaving that dialog brings you back to the project browser.

20.1 Perspective correction and distortion

The **Perspective correction and distortion** dialog of Fig. 20.1 checks the image dewarping function on images as well as the effect of the vector preshift used in the vector multi pass calculations. The example shows a calibration plate image to illustrate the dewarping. If an image should be used for dewarping, press right mouse button on the image and select **Perspective correction and distortion**.

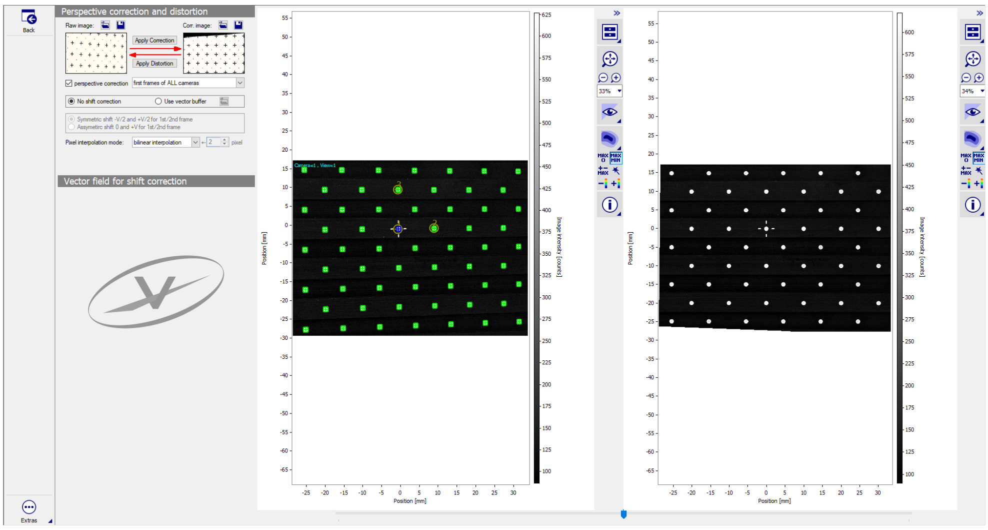


Figure 20.1: Image Reconstruction dialog

The left image holds the source image which has been sent to the dialog. The right image includes the result of the correction. At first, select the parameters as shown in Fig. 20.1. Then, press button **Apply correction** to start processing. The result might be like Fig. 20.2.

In this example, all distortions due to the perspective distortion as well as camera lens errors are taken out.

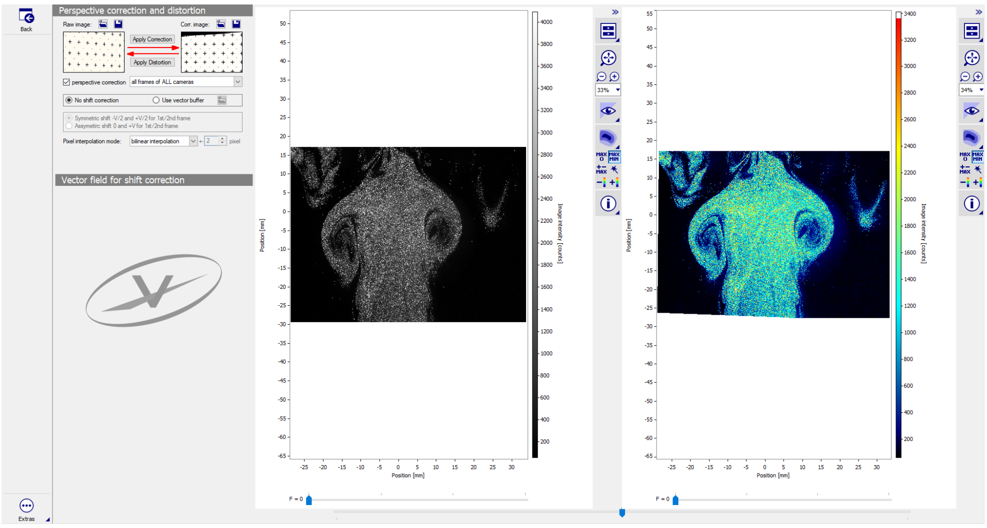
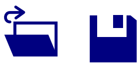


Figure 20.2: Image Reconstruction result

This can be done for the standard case of z = 0 mm height or for any other fixed height above the calibration plate position of view 1 used in the calibration process. Alternatively, a height map can be provided, holding the z-position for each pixel as used in the **StrainMaster 3D** package.

Another way is to provide a corrected image and distort via button **Apply distortion**. Both the source and result images can be saved and loaded by pressing the corresponding button.



In the dewarping process, a bilinear interpolation is used to access image information at subpixel positions. One optional mode is the **Whittaker reconstruction** with a specified filter length (e.g. 11 pixels). This mode is significantly slower but might provide sharper images.

A second optional mode is to shift the image contents as specified in a reference vector field. To use this mode, the flag **use image correction** has to be disabled and use **vector buffer** has to be enabled. The result is like the example of Fig. 20.3.

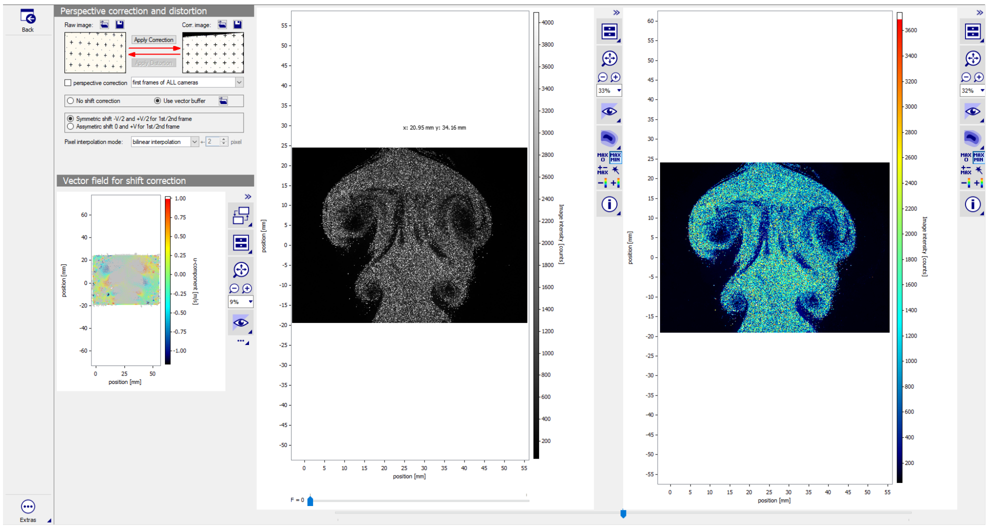


Figure 20.3: Image Reconstruction with vector shift

This second mode gives a possibility to check if the vector results really describe the shifts of the image well. If two frames are present for a camera, then another option is available: Either the whole shift can be applied to the second frame only as recommended in time series images. Or the half of the vector shift can be distributed evenly on both frames. For double frame images, the opposite sign is recommended.

After processing the reconstruction, the patterns of both frames should match exactly when switching between both frames. If there are still shifts present in the corrected image, this indicates that the vector field still does not describe the shifts in those regions well enough. More (multi pass) iterations and/or a finer grid should be tried.

For vector calculations with image correction, both **use image correction** and **use vector buffer** can be combined.

Expert users can choose the option to correct/distort frames for a specific camera only. Another expert option allows to select all frames with existing `CameraNameX` buffer attributes. The **auto mode** is recommended.

20.2 Correlation Map

To open the **Correlation map** dialog of Fig. 20.4 and Fig. 20.5, use the right mouse button on an image and select the **Correlation map** option, in case an image buffer is the source of the dialog. If a vector buffer was chosen as source, then select **Correlation map / vector choices**. On the top of the dialog, the **Correlation parameter** and the **Vector component values** are located. On the bottom left is the image/vector buffer, utilized as the source for the correlation. The respective correlation function is displayed on the right.

In image buffer source (see Fig. 20.4) the **Correlation window** provides information on which frames were utilized for correlation computation. Choose a correlation window size between 8x8, 16x16, 32x32, 64x64, 128x128, 256x256, 512x512, and 1024x1024 pixel. By clicking and moving the red Correlation window in the source image, the correlation is recalculated automatically.

By moving the cursor over the correlation, the correlation height is determined and shown in the status line. The cursor position is displayed, too, either scaled, or in pixel coordinates, when keeping the **SHIFT** button on the keyboard pressed.

If a vector buffer was chosen as source (see Fig. 20.5), the **Correlation window** can be set as described for image buffer source. Thereby the correlation window is snapped to a vector. Additionally, the tick at **Vector field was created using perspective correction** indicates, if the correlation calculation was taking perspective correction into account or not. The correlation map is not calculated from the vector buffer itself, but from the same image buffer from which the current vector buffer was calculated.

In **Vector component values** the positions of the 1st, 2nd, 3rd, and 4th choice vector in [pixel] are shown for the x-, y, and z-components. By moving the red correlation window in the source image, the correlation is calculated automatically. However, these values can be changed manually, or even disabled by choosing **disabled**. **Undo current vector** undoes all change in the current selected vector that were performed manually. **Undo whole vector buffer** resets all changes.

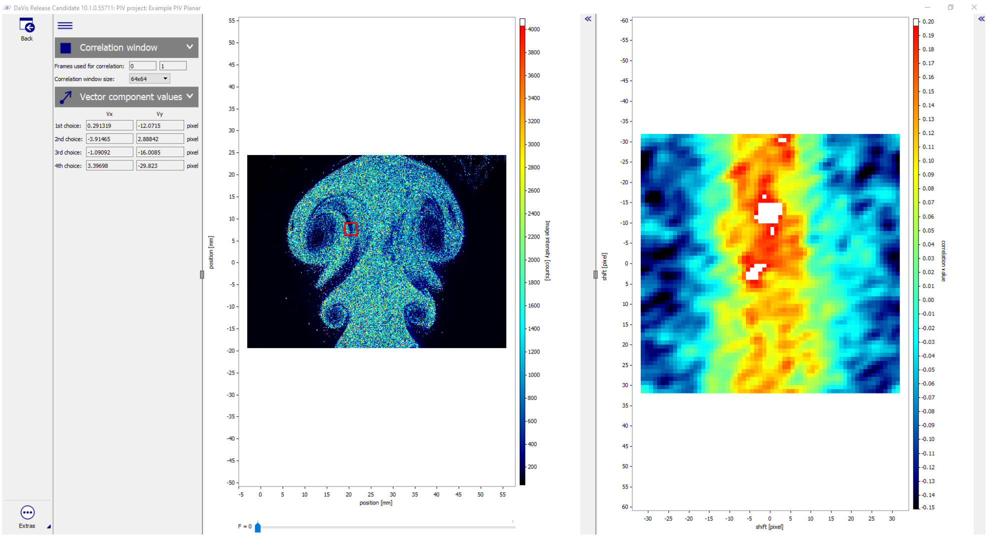


Figure 20.4: Correlation Map dialog for image source

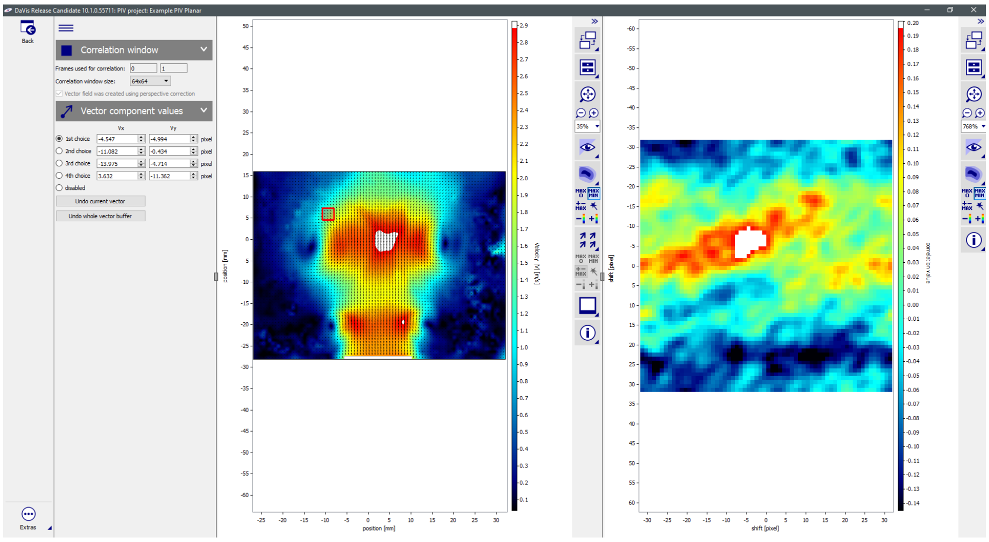
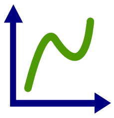


Figure 20.5: Correlation Map dialog for vector source

20.3 Plot



To open the **Plot** dialog, first, select an image set or a vector set and then press the **Plot** button in the tool bar.

This dialog helps to interactively create user-defined plots over time for a specific region in the input data or a plot over space taking a cut along a line in the input data. Depending on the type of the data set, different data sources for the x and y axis are available. The same function is available in the processing dialog as operation **Plot**.

The dialog, as shown in Fig. 20.6, is subdivided into a left part with the parameters for the plot creation and the left part with the views of the source and result set.

Source: The selected data set, which is used to create the plot.

Result: The resulting plot.

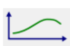
In the **Source** section, the source set name is shown and the maximum range of images or vector fields for processing. In **Selected range** and **Increment**. the start and end is defined, increment (default 1) allows to skip files, e.g. *Start-End* = 1-9 with *Increment* = 2 selects files 1,3,5,7,9 for processing.

20.3.1 Plots for Vector Sets

Only single-plane (planar or Stereo-PIV) vector sets are supported. Tomographic PIV (multi-plane) vector sets are not supported.

Plot from data

To create a plot from data, you need to select the input set range you want to extract (start, end and increment) in the **Source** group and do the following:

1. Select the first mode **Plot from data**  in the **Parameters**.
2. Select the X-axis (**Image number**, **time**, or **device data**).Please note that device data can be only selected, if it is available in the source set. This is typically an external signal that was recorded with

an analog digital converter during the recording and is attached to each image and has been propagated to the vector field.

3 Select the data to plot on the Y-axis. The available options are **Vector background** or **Others**. In Vector background, the values comprise functions as in the processing groups in processing dialog "extract scalar field:...". Otherwise, choose an Others. The y-dimension can be taken from statistical image data of the whole image or a defined rectangular region. For the latter, the check mark **Use rectangle** has to be enabled. Specify a **Statistic function** to calculate the average value, the standard deviation, minimum, maximum or the sum of values of all selected vectors inside the chosen area.

4 **Start processing** will perform the data extraction. The resulting plot is shown in the **Plot** window.

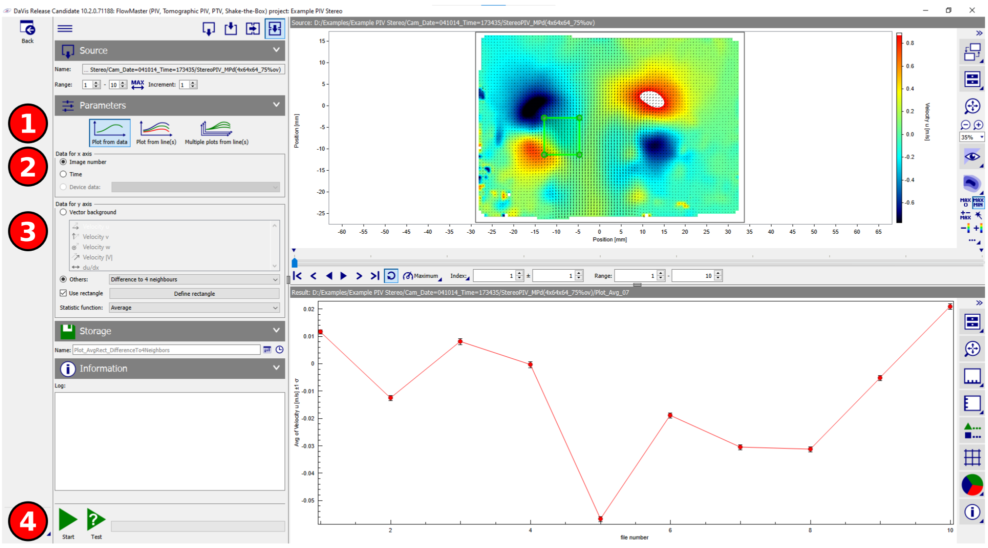


Figure 20.6: Plot dialog for vector sets. *X value / Y value (plot from data)*

Uncertainty bars in plot from data

Calculating a plot from data will calculate the average value inside the rectangle and show the uncertainty of the calculated average as an error bar.

If the plotted magnitude **does** contain an uncertainty value (e.g. of V_x, V_y, V_z, vector length) in the input buffer, then the uncertainty of the average is the averaged RMS uncertainty divided by the square root of the number of samples:

$$U_{\overline{x}} = \frac{\sqrt{\sum U_x^2 / N}}{\sqrt{N}}.$$

If the spatial resolution L_{sr} in x- and y-direction is known because neighbor values are correlated, then, instead of the number of samples N, the effective number of samples N_{eff} is used. Depending on the selected PIV parameters (window size, overlap, round/square windows, CPU or GPU), a spatial resolution is known. So N_{eff} is used and not the $N_x \times N_y$ entries inside a rectangle with L_{sr}=spatial resolution length (with N_{eff} at least 1):

$$U_{\overline{x}} = \frac{\sqrt{\sum U_x^2 / N}}{\sqrt{N_{eff}}} \quad \text{with} \quad N_{eff} = \frac{N_x N_y}{L_{sr}^2}.$$

If the plotted magnitude **does not** contain an uncertainty value in the input vector field, then the uncertainty of the average is the standard deviation of all values inside the rectangle divided by the square root of the number of values:

$$U_{\overline{u}} = \sigma_u \sqrt{\frac{1}{N_{eff}}}, \quad U_{\overline{v}} = \sigma_v \sqrt{\frac{1}{N_{eff}}}, \quad U_{\overline{w}} = \sigma_w \sqrt{\frac{1}{N_{eff}}}.$$

Plot & Multiple plots from line

To create a plot from line, you need to select the input set range you want to extract (start, end and increment) as in the processing in the **Source** group and do the following:

Select one of the modes **Plot from line(s)** or **Multiple plots from line(s)** in the plot mode group. Depending on the selection, you get one plot with n lines or a set of n plots with one line each, respectively. Please refer to Fig. 20.7 and 20.8.

1. Select one of the modes **Plot from line(s)** or **Multiple plots from line(s)** in the plot mode group.
2. Select the **Data source** to plot data on the Y-axis. The available options are **Vector background** or **Others**. In Vector background, the values comprise functions as in the processing groups in processing dialog "extract scalar field:. . . ". Otherwise, choose a Scalar field.

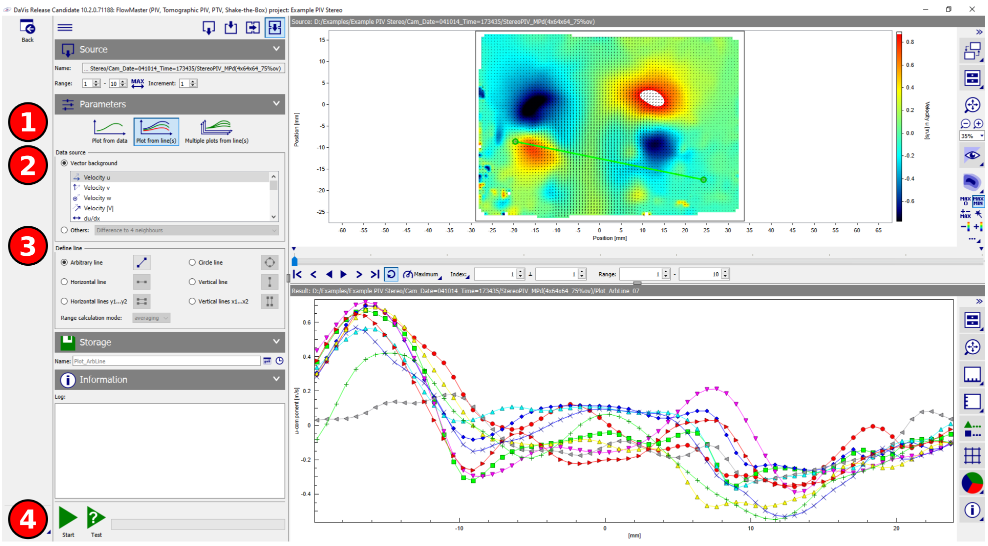


Figure 20.7: Mode *x line/y line (plot from line(s))*

3 Then, select the extraction mode

- arbitrary line
- horizontal line
- horizontal lines y1...y2
- circle line
- vertical line
- vertical lines x1...x2

and then manually define the position in the vector field.

For an arbitrary line, a vertical line or a horizontal line, the values for each vector along the line are calculated and shown versus the position along that line.

For horizontal range, the values are averaged inside the horizontal range (along x) for each y position and plotted against that y position. For the vertical range mode, it is vice versa.

For horizontal lines y1...y2 and vertical lines x1...x2 Range calculation mode can be selected: averaging or summing.

The X-axis is set automatically to **values along line** and cannot be changed.

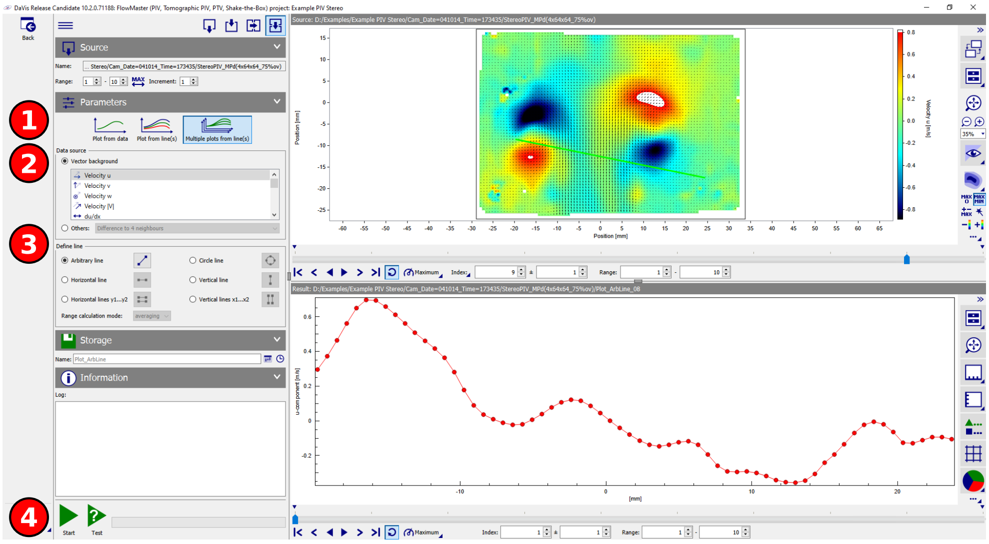


Figure 20.8: Mode *x line / y line (multiple plots from line(s))*

4 **Start processing** will perform the data extraction. The resulting plot is shown in the **Plot** window.

Uncertainty bars in Plot & Multiple plots from line

For the horizontal range/vertical range, the uncertainty of the averaged value is calculated depending on whether the input value contains an uncertainty information:

If the plotted magnitude **does** contain an uncertainty value (e.g. V_x, V_y, V_z, vector length) in the input vector field, then, the uncertainty of the average is the averaged uncertainty divided by the square root of number of samples:

$$U_{\overline{x}} = \frac{\sqrt{\sum U_x^2 / N}}{\sqrt{N}}.$$

If the spatial resolution L_{sr} in x- and y-direction is known because neighbor values are correlated, then, instead of the number of samples N, the effective number of samples N_{eff} is used. Depending on the selected PIV parameters (window size, overlap, round/square windows, CPU or GPU), a spatial resolution is known. So N_{eff} is used and not the $N_x \cdot N_y$ entries inside a rectangle with L_{sr}=spatial resolution length (with N_{eff} at least 1):

$$U_{\overline{x}} = \frac{\sqrt{\sum U_x^2 / N}}{\sqrt{N_{eff}}} \text{ with } N_{eff} = \frac{N_x N_y}{L_{sr}^2}.$$

If the plotted magnitude **does not** contain an uncertainty value in the input vector field, then, the uncertainty of the average is the standard deviation of all values inside the rectangle divided by the square root of the number of values:

$$U_{\overline{u}} = \sigma_u \sqrt{\frac{1}{N_{eff}}}, \quad U_{\overline{v}} = \sigma_v \sqrt{\frac{1}{N_{eff}}}, \quad U_{\overline{w}} = \sigma_w \sqrt{\frac{1}{N_{eff}}}.$$

20.3.2 Plots for Image Sets

To open the **Plot** dialog, first select an image set or a vector set and then press the **Plot** button in the **Project Manager**.

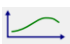
This dialog helps to create interactively user-defined plots over time for a specific region in the input data or a plot over space taking a cut along a line in the input data. Depending on the type of the data set different data sources for the x and y axis are available. The same function is available in the processing dialog as operation **Plot**.

The plot dialog offers two modes to create a plot.

20.3.3 Plot from data

The mode **plot from data** extracts one sample point from each image and plots it over file number, image time stamp or a specific device data value stored in the image (e.g. from an ADC device). See Fig 20.9.

To create a plot over time, you need to select the input set range you want to extract (start, end and increment) in the **Source** group and do the following:

1. Select the first mode  in the **Parameters**.
2. Select the X-axis (**Image number**, **time**, **crank angle** or **device data**).Please note that device data can be only selected, if it is available in the source set. This is typically an external signal that was recorded with an analog digital converter during the recording and is attached to each image and has been propagated to the vector field.
3. Select the data to plot on the Y-axis. The y-dimension can be taken from statistical image data of the whole image or a defined rectangular region. For the latter, the check mark **Use rectangle** has to be enabled. Otherwise, again device data can be selected.

4 **Start processing** will perform the data extraction. The resulting plot is shown in the **Plot** window.

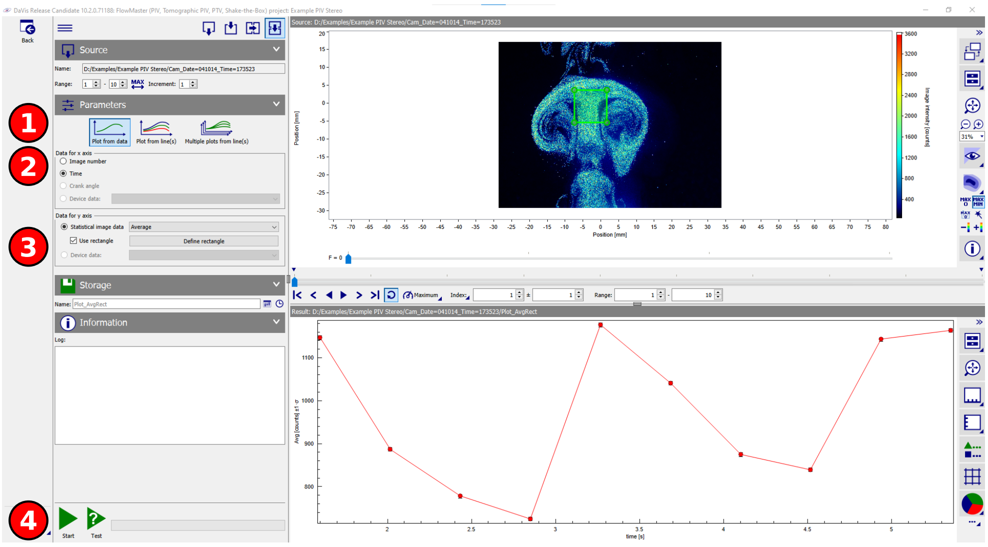
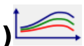


Figure 20.9: Plot dialog for image sets. *X value/ Y value (plot from data)*

Plot & Multiple plots from line

The two modes **Plot from line(s)**  and **Multiple plots from line(s)**  extract intensity values along a line. See Fig. 20.10 and 20.11.

1 Select one of these modes in the plot mode group. Depending on the selection, you get one plot with *n* lines or a set of *n* plots with one line each, respectively.

2 select the extraction mode

- arbitrary line
- horizontal line
- horizontal lines y1...y2
- circle line
- vertical line
- vertical lines x1...x2

and manually define the position in the vector field.

For an arbitrary line, a vertical line or a horizontal line, the values for each vector along the line are calculated and shown versus the position along that line.

For horizontal range, the values are averaged inside the horizontal range (along x) for each y position and plotted against that y position. For the vertical range mode, it is vice versa.

For horizontal lines y1...y2 and vertical lines x1...x2 Range calculation mode can be selected: averaging or summing.

The X-axis is set automatically to **values along line** and cannot be changed.

3 **Start processing** will perform the data extraction. The resulting plot is shown in the **Plot** window.

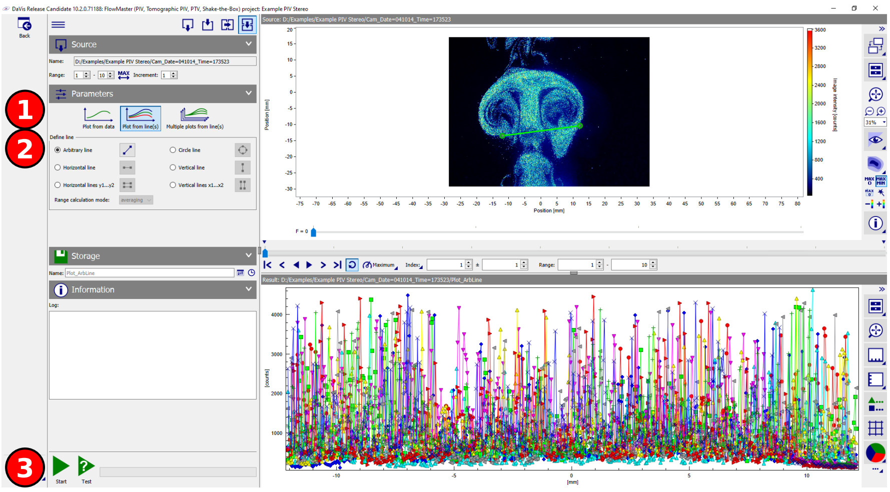


Figure 20.10: Mode *x line / y line (plot from line(s))*

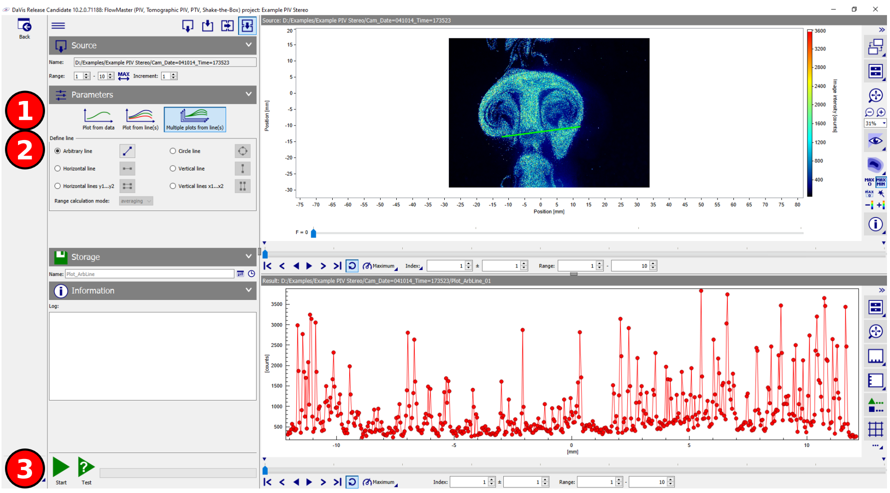


Figure 20.11: Mode *x line / y line (multiple plots from line(s))*

A Support

If you have a technical problem or a question regarding hardware or software which is not adequately addressed in the documentation, please contact your local representative or **LaVision** service directly.

You can contact service at **LaVision** GmbH by:

e-mail: **service@lavision.de**

phone: **+49 551 9004 229**

Alternatively, you may submit your problem using the **Support Request Form** in the **Support** section of the **LaVision** website `www.lavision.com`.

In order to speed up your request, please include the following information:

- The order number of your system (see section A.1).
- The number of the used dongle (see section A.1).
- A short description of the problem.
- The **LaVision** service file (see section A.2).
- Some logfiles if you have a reproducible software problem (see section A.3).
- Information on the Windows operating system and service pack used on the corresponding computer.

A.1 Order and Dongle Number

To be able to find information on the delivered hardware components and customer details in the **LaVision** database, your order number is required. This number can be found in the toolbar menu **Extras – About** or on the original **DaVis** installation medium (see Fig. A.1).

In the **About DaVis** dialog you find the dongle number and order number information. The **Version ID** is the build number of the **DaVis** version, shown on top of the dialog.

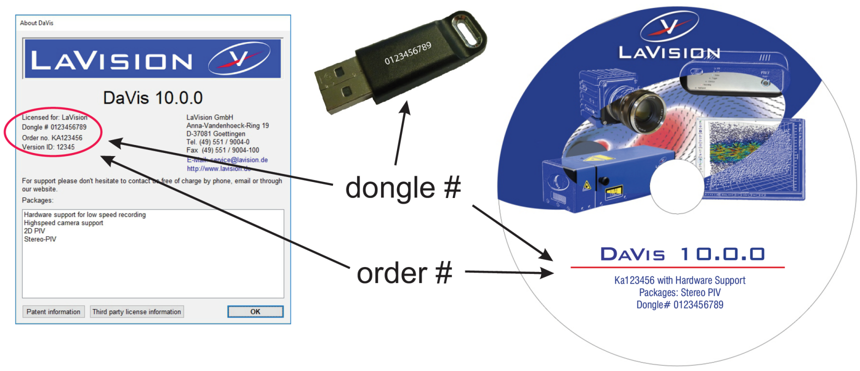


Figure A.1: Dongle and order no. in **Extras - About** and on the installation DVD.

The dongle number is required to exclude possible license problems. This number is printed on the hardware key as well. The dongle number and the order number can also be found on the original **DaVis** install medium.

Please include the order number and/or the dongle number in your service requests.

A.2 LaVision Service File

In order to be able to reproduce a software problem, it could be essential to know the exact hardware setup and software parameters in **DaVis**. All currently used parameters and all error messages that have been shown since the last **DaVis** start can be extracted using the toolbar menu **Extras – Service – Create service file for LaVision support**.

After you have selected this menu, the system will write all values for the relevant variables into a `LSFX` file. This file will also contain the current settings of the hardware setup, acquisition setup and processing operation lists. The procedure will take some seconds!

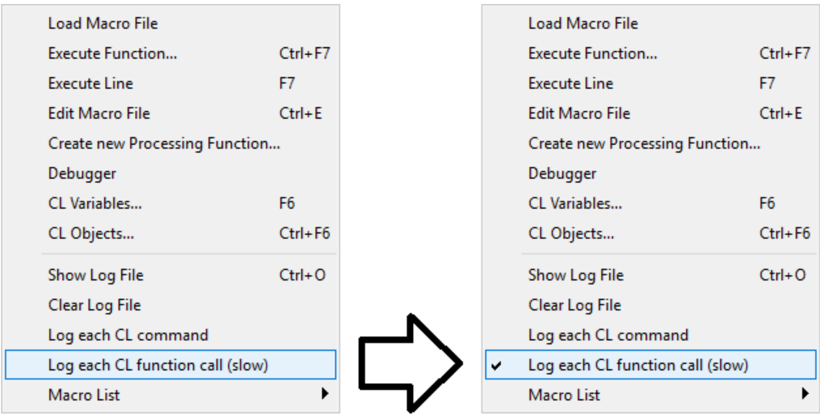
The `LSFX` file will be written automatically to a folder selected by the user and the Windows explorer opens at the end with this folder. The name of the file contains the order number and dongle number that is extracted from your software (`#ordernumber_donglenumber.lsfx`). Send the `LSFX` file as attachment to your email together with the description of your problem to `service@lavision.de`.

A.3 Log File

During startup of **DaVis**, some log files are generated in the **DaVis** subdirectory Users/<name>/log (until version 10.2.0) or in **DaVis** subdirectory ProgramData/log (since version 10.2.1). The standard log files are separated for certain areas of the software and get corresponding names for easier access by service. The log files from the Command Language are named like LOG_<date>_<time>.txt with date and time of the **DaVis** startup, e.g., LOG_170615_150343.txt. **DaVis** holds the last ten CL log files and removes older ones automatically.

If you have a reproducible software problem in **DaVis**, please send the complete log folder together with your email. These files contain all functions you have called and all error messages that have been displayed after you activated the log. Please proceed as follows:

1. Start **DaVis** and use the toolbar menu **Extras – Macro – Clear Log file**.
2. Enable the **Log each CL function call (slow)** entry in the menu. This feature is active if you see a flag at the left side of the entry. Every time you click on this entry, its status is changed.

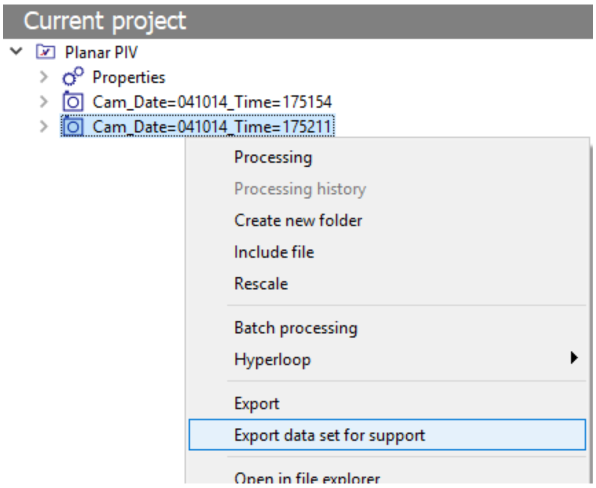


3. Try to reproduce your problem, e.g., until an error message is displayed.
4. A log file has been generated in the **DaVis** main directory. Send this text file attached to your email.
5. Disable mode **Log each CL function call (slow)**. This function is deactivated if you do not see a flag next to the entry.

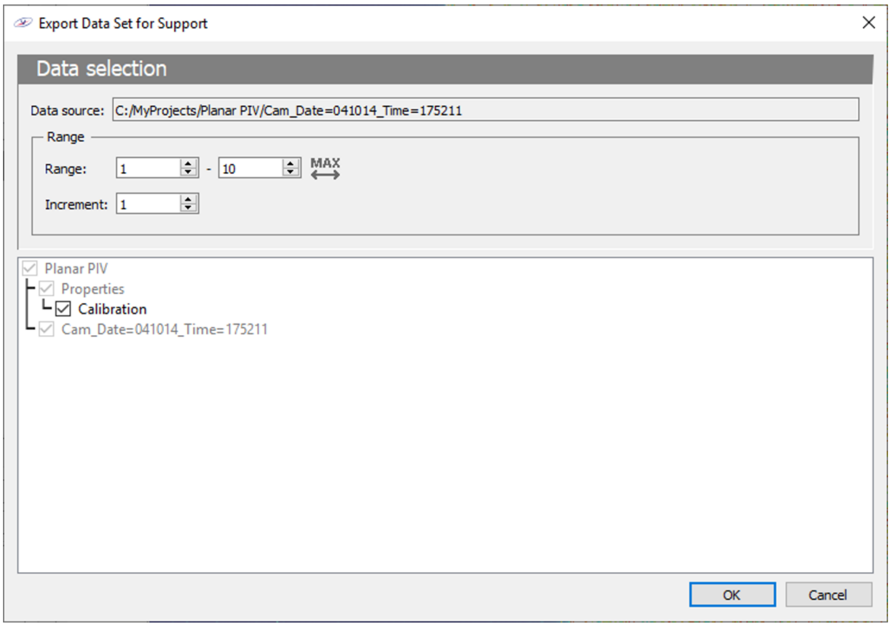
A.4 Export Data Set for Support

Some problems can only be reproduced using images or data that contain particular information or artifacts. For error analysis, it can be necessary to provide exemplary data that need to be extracted from the corresponding project.

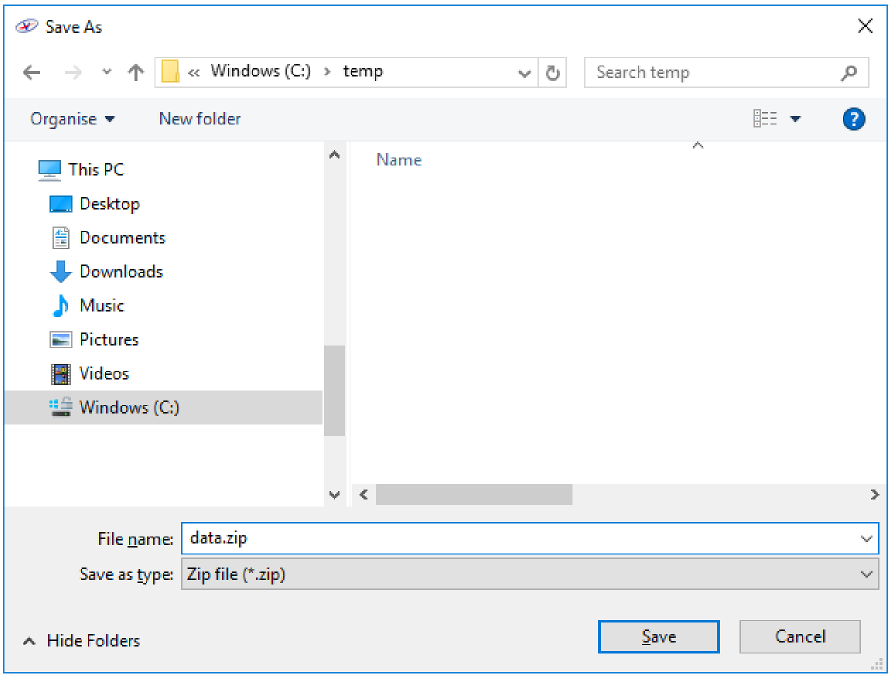
Depending on the project type, the number of cameras used, and the error, it can be necessary to provide the corresponding calibration (spatial, temperature, etc.) and derivative data as well. A convenient way to extract the data from the project is the **Export data set for support** option, which you can select by right-clicking on the corresponding data set in the tree view of the **Project manager**.



In the **Export data set for support** dialog, specify the range of data which you would like to extract from the data source by entering the range (i.e., first and last image). If a calibration is available in the project, this will be added by default. You have the option to deselect this part if it is not relevant.



After clicking the **OK** button, you need to specify location and file name for the `zip` file that contains the selected data.



DaVis will ask to open the containing folder or to send an email to `service@lavision.de`.

Note: Files with a size of more than 20 MB should not be sent by email. **LaVision** can provide a link for uploading data via file drop. Please contact `service@lavision.de` for details.

A.5 Shipment of Defective Items

If any item needs to be returned to **LaVision** GmbH for service or repair, please contact the **LaVision** service to obtain a **RMA** (Return Material Authorization) number together with an RMA form. This will list all items with SN and a short description of the problem. Place the RMA form in the box with the item(s) being returned. Return the authorized item(s) according to the shipping instructions.

Shipping instructions:

- Be sure to obtain an RMA number and RMA form.
- Add the signed RMA form to the shipping documents.
- Ship only the items that are authorized.
- Use the original boxes to avoid damages during transportation.
- **Remove cooling water from the laser!**
- **Use antistatic bags for computer boards!**
- Ship returned items to:

  LaVision GmbH
  Anna-Vandenhoeck-Ring 19

  37081 Göttingen

  GERMANY

Note: Shipments received by **LaVision** without an RMA number may be refused.

B Instructions for license upgrade

The license upgrade procedure depends on the used dongle and the number of digits of the dongle identifier, means the used license system.

In case of the old 5-digit-dongle **LaVision** sends a configuration file to be loaded in the **License Management Dialog** of **DaVis**.

In case of the new dongles with 10 digits two steps by the customer are required: At first create a file with current license state and send to **LaVision** and as second step receive the upgrade file and install.

Figure B.1: Different dongle types.

Please contact `service@lavision.de` in case of problems.

B.1 License upgrade of a 5-digit dongle

The license upgrade file is either delivered via email or on a USB flash drive. In first case just store the attached file somewhere on harddisc. In case of the USB flash drive the file is located in the `Settings` folder on the flash drive.

Start **DaVis** and open the **License Management** dialog of figure B.3. In case of a time limited and already outdated licenses this dialog opens automatically during startup. Otherwise select toolbar item **Extras – License Management**, see figure B.2, to open this dialog.

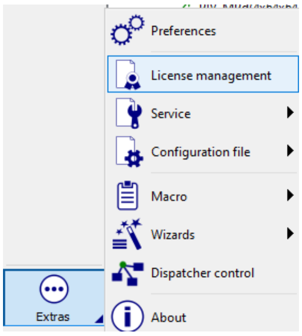


Figure B.2: Extras menu to open the license management dialog.

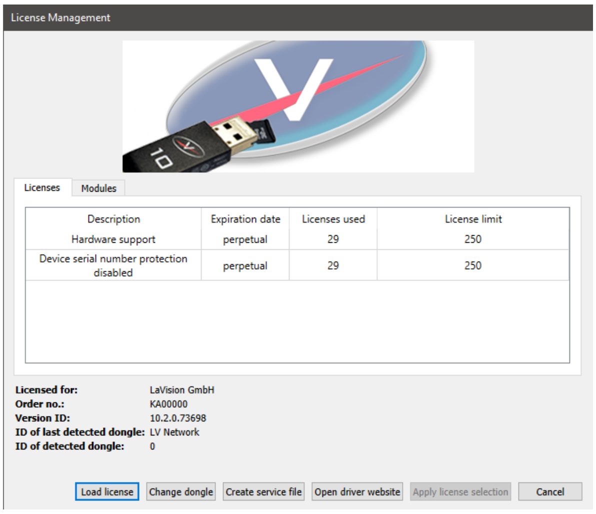


Figure B.3: License management dialog.

Press button **Load Licence** and select the new license file in the opened fileselectbox. The file is of type `LCFX` (default: LaVision Customer File). Only the license information and no other settings are read from the selected file. Then **DaVis** restarts automatically.

B.2 License upgrade of a 10-digit dongle

Open the dongle driver website `http://localhost:1947/` in a web browser or press button **Open Driver Website** in the **License Management** dialog of **DaVis**. If the website is not found the dongle driver is not installed!

B.2.1 Installation of the dongle driver

In most cases the dongle driver has been installed together with **DaVis**. A folder of a installed **DaVis** contains the dongle driver: Open the **DaVis** folder and then `driver/dongle`. Execute `install.bat`.

The dongle driver can also be installed with the **DaVis** installer. Select the dongle driver during installation in the list of features. See page 23 about details.

The current version of the dongle driver can be download by registered users from the **LaVision** website:
`https://www.lavision.de/de/downloads/driver/dongle.php`

B.2.2 Send current license state for a local dongle

Use DaVis for the connected dongle

The current license state of the connected dongle is stored in the service file. Start **DaVis** and create a service file via toolbar item **Extras – Service – Create service file for LaVision support.** Send the created `lsfx` file to **LaVision**.

Note about multiple connected dongles: Until version 10.1.2.67531 this does not work in case of two or more connected dongles. Please remove other dongles from the PC before creating the service file. Newer versions are writing the license state of all connected dongles into the service file.

Create license state from driver website

Note: This way is working for a new dongle generation only delivered since end of 2019. For older dongle types the **C2V** button as described below is missing.

Open the driver website on the PC with the connected dongle either from the **License Management** dialog of **DaVis** or directly with this link in a web browser:

`http://localhost:1947/_int_/devices.html`

Find the line with your dongle identifier, see figure B.4. Note that the dongle identifier is given as **Key ID** on this website and leading zeros are not displayed. Press button **C2V** on the right in the line with your dongle. On the next screen press button **Create C2V File**, store the file and send it to **LaVision**.

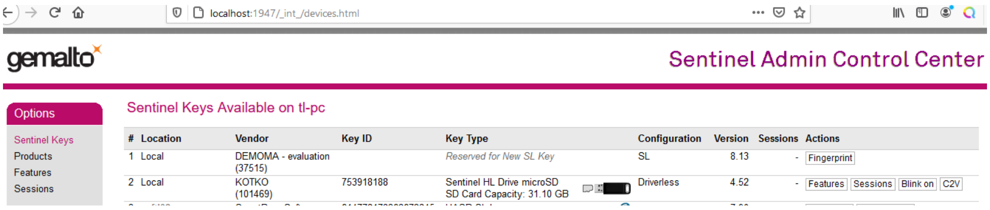


Figure B.4: Driver website with list of detected dongles.

B.2.3 Send current license state for a network dongle

In case of the license upgrade of a network dongle the direct access to the dongle or at least to the server machine with connected dongle is required! The state of the dongle can't be read from another machine via network.

If the dongle can simply be removed from the server PC and connected to the local PC, then please follow the instructions above for the local connection.

Otherwise a Remote Desktop connection must be established or another software like **TeamViewer** has to be used. Try the C2V-creation way via driver website first and follow the description above. If the **C2V** button is missing, the state has to be read with the help of a tool from the installed **DaVis**.

Using the LaVisionLicence.exe to get the license state

Copy the files `LaVisionLicence.exe`, `haspvlib\101469.dll`, `Fix-HASP-BaseFile.bat` and `Store-HASP-BaseFile.bat` from the **DaVis** `win64` folder to the server PC into a new folder without write protection.

Then execute `Fix-HASP-Driver.bat` with admin privileges. This should be executed once on the server and avoids an error in the next step in case of an incomplete driver installation.

Now execute `Store-HASP-BaseFile.bat` with admin privileges. This reads out the dongle and stores a `c2v` file with the current license state. Send this file to **LaVision**.

B.2.4 Upgrade the license state

Open the driver website on the PC with the connected dongle either from the **License Management** dialog of **DaVis** or directly with this link in a web browser, see figure B.5:

`http://localhost:1947/_int_/checkin.html`

Press button **Search** and select the `v2c` file sent by **LaVision**. Then press button **Apply** and wait for the success message of the dongle driver. Now **DaVis** can be started again using the upgraded license.

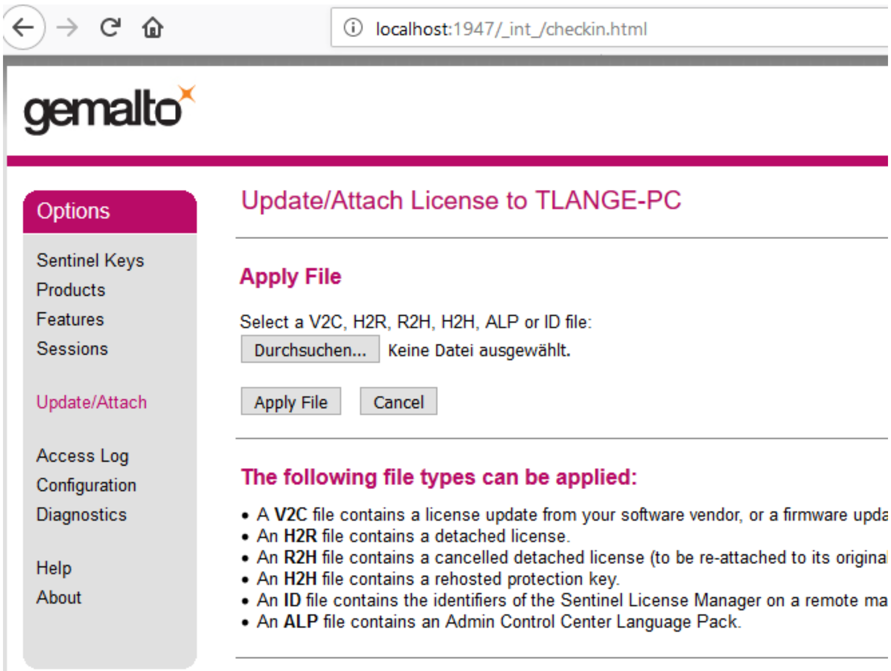


Figure B.5: Driver website to upgrade the license of a connected dongle.

B.3 How to install a license bundled to the PC without physical dongle

This instruction explains how to run **DaVis** without a physical USB dongle or a network license using a temporary software license. This software license is intended for cases, where the operator has no access to an existing USB or network dongle. It will run for a limited period of time.

B.3.1 Requirements

Please make sure that you can fulfill these requirements:

- Your institute/company is owner of a valid license and a physical dongle (USB or network).
- You personally have the permission to use this license.
- The access to your dongle is temporarily not possible due to working or health restrictions.
- Your computer is compatible with the DaVis version: most important is Windows version and language.
- You have DaVis 10 with dongle driver installed, according to your existing license.
- You can download DaVis free of charge from our website. This requires registration on `http://www.lavision.de/en/login.php`

B.3.2 Procedure (brief)

1. Install DaVis and the dongle driver on the target computer.
2. Verify the dongle driver installation in your browser at `http://localhost:1947/_int_/diag.html`
3. Open sub-folder `win64` in the DaVis installation and run as administrator the `Store-HASP-Fingerprint-SoftwareKey.bat`
4. The dialog tells about success or failure and about the name of the generated fingerprint file of your computer (c2v).
5. Please contact your sales representative to request a Software License File by email with the c2v file as attachment.
6. After identifying your existing license, you will get a Software License File sent back. This might take a working day to process.
7. Load the Software License File in your browser at texttthttp://localhost:1947/_int_/checkin.html

B.3.3 Procedure (detailed)

1. Install DaVis and the Dongle Driver

Install **DaVis** and the dongle driver on the target computer. You can download the setup file from our website. Please use the version according to your existing license.

Start the setup and select the installation type **Install a generic DaVis version**. Later choose as **Installation options DaVis** itself and the **Dongle Driver**, see figure B.6.

> **Note**
>
> The Dongle driver must be installed, even though you do not have a dongle now!

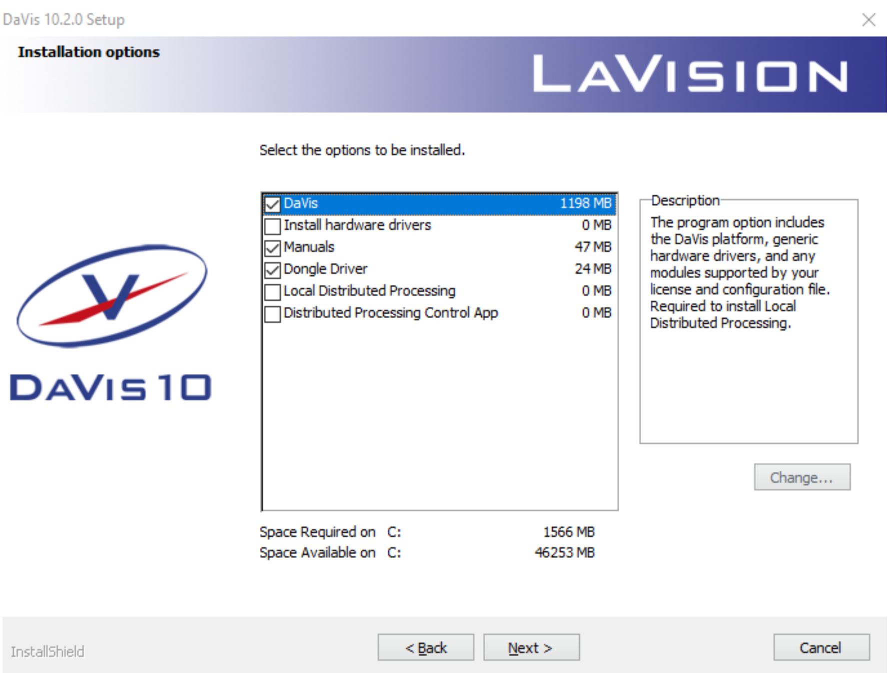


Figure B.6: Dialog for feature selection in the DaVis installer.

2. Verify the dongle driver installation

Start your browser at the internal website `http://localhost:1947/_int_/diag.html` If the **Diagnostics** website of the **Sentinel Admin Control Center** appears then the driver is correctly installed.

If this page is NOT visible then the dongle driver was not correctly installed! Reboot the computer. Install the driver again using administrator privileges.

3. Create PC fingerprint

Open sub-folder `win64` in the DaVis installation, select batch file `Store-HASP-Fingerprint-SoftwareKey.bat` with right mouse click, then **run as administrator**. The result should be like figure B.7. The name of the created file follows rule `fingerprint-<DOMAIN>-<COMPUTERNAME>-<USERNAME>.c2v` and can be renamed to hide internal company information.

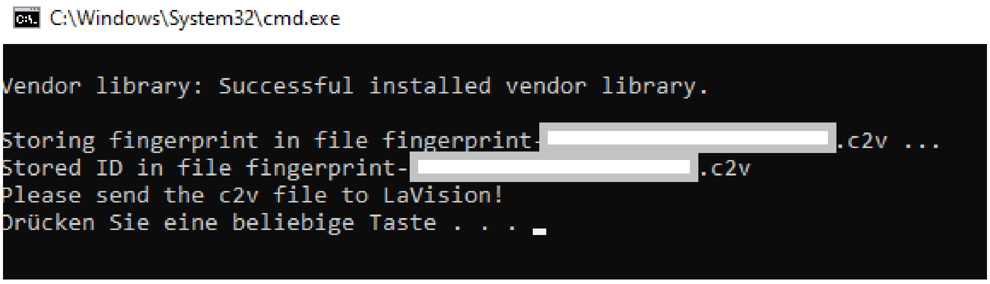


Figure B.7: Message after running `Store-HASP-Fingerprint-SoftwareKey.bat`.

4. Send the Fingerprint File (*.c2v) to LaVision sales and request a new license

Contact your sales representative to request a Software License File. Please include in this mail

- Your full name.
- Your institute or company name; if applies: department or group.
- If you are sending from your private mail address: your work mail address related to your institution.
- A brief description of the system (e.g.: "PIV", "HighSpeed", "Spray").

- If you still know the dongle number or order number - that helps to accelerate the process! But this is not required, if not available.
- Attach the Fingerprint File!

5. Install the Software License File (*.v2c) on your PC

Open the driver website on the PC with the connected dongle, see figure B.5:

`http://localhost:1947/_int_/checkin.html`

Press button **Search** and select the `v2c` file sent by **LaVision**. Then press button **Apply** and wait for the success message of the dongle driver. Now **DaVis** can be started again using the upgraded license.

C Calibration Plate Configurator (CPC)

C.1 Overview

The "Calibration Plate Configurator" is a tool for **DaVis** which allows the user to display official Calibration plates of **LaVision** GmbH, or possibly to create and view your own "User Calibrationplates". The program is split in two areas, on one card, the official Calibration plates that cannot be processed and on the other card, "User Calibration plates" which can be viewed, edited, created and removed.

C.2 Features

The application only includes a few features, but those are perfectly adequate for the purpose. The application supports things like:

- Automatic loading of "*.xml" files, if they are available
- The specific loading of "CalibrationPlates.xml" and "UserCalibrationPlates.xml"
- Displaying all the values of a selected calibration plate
- Adding new "User Calibration plates" in the "UserCalibrationPlates.xml"
- Removal of "User Calibrationplates" from the "UserCalibrationPlates.xml"
- Graphical interface, icons and labels, for better understanding

C.3 Location

You will find the "Calibration Plate Configurator (CPC)" in form of a tool witch access, within the calibration dialog. You just have to click on the button in the toolbar (Fig. C.1).

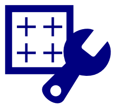
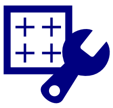


Figure C.1: Calibration Plate Configurator

C.4 User interface

Calibration Plate Configurator



✓ Click on this icon to open a menu where you can design your own calibration plate.

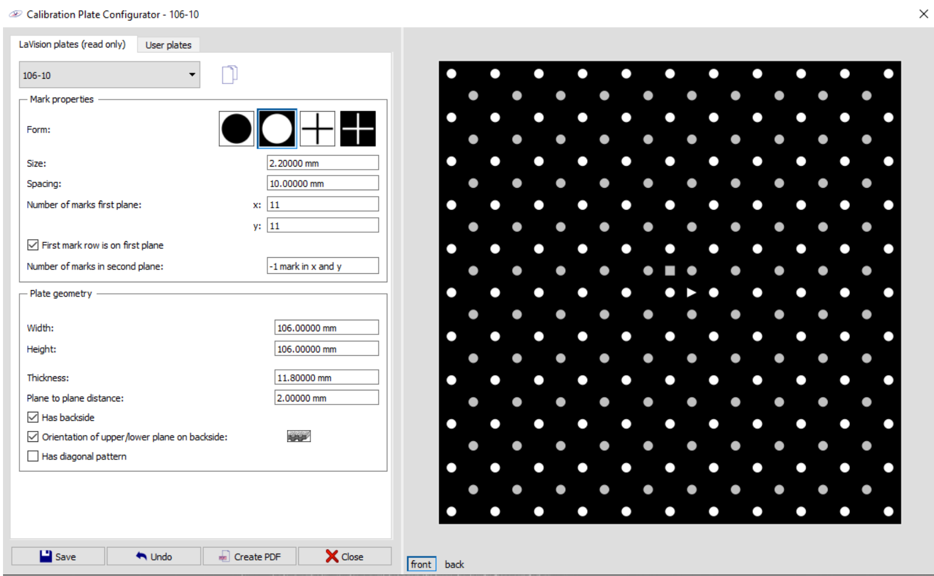


Figure C.2: Calibration plate configurator

✓ Click on the tab LaVision plates and select a plate type to display its specification (Fig. C.3).

! The plate specification cannot be edited.

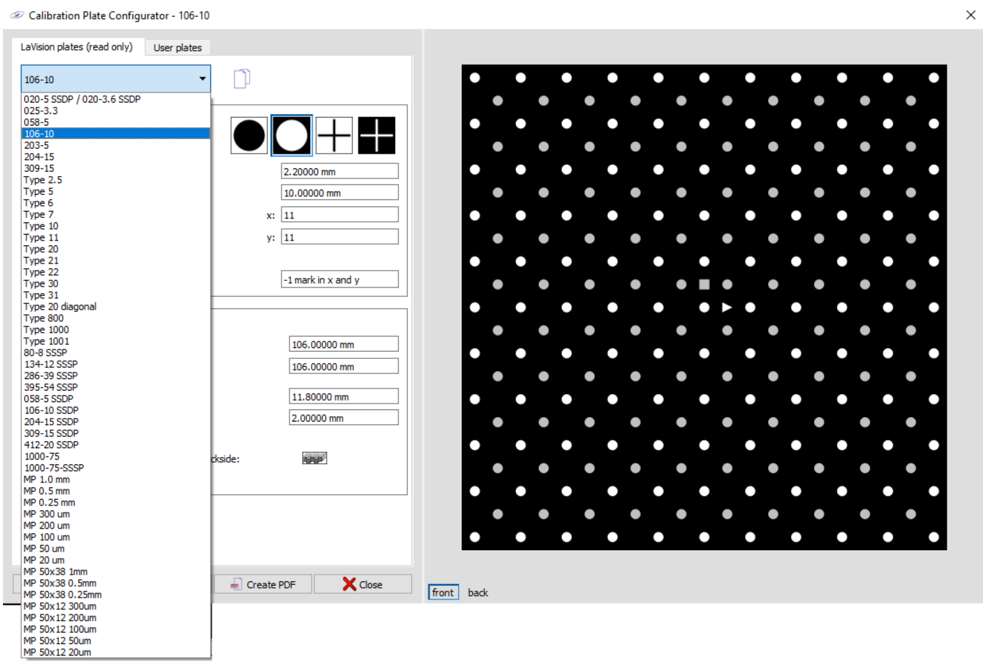
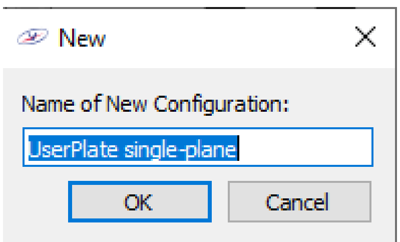
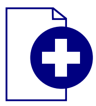


Figure C.3: LaVision Plate Specification view

✓ Click on the tab **User plates** and select a plate type to display its specification or to configure a new plate (Fig. C.4).

Create New Plate





▷ Option to design a new calibration plate.

✓ Type in the plate designation.

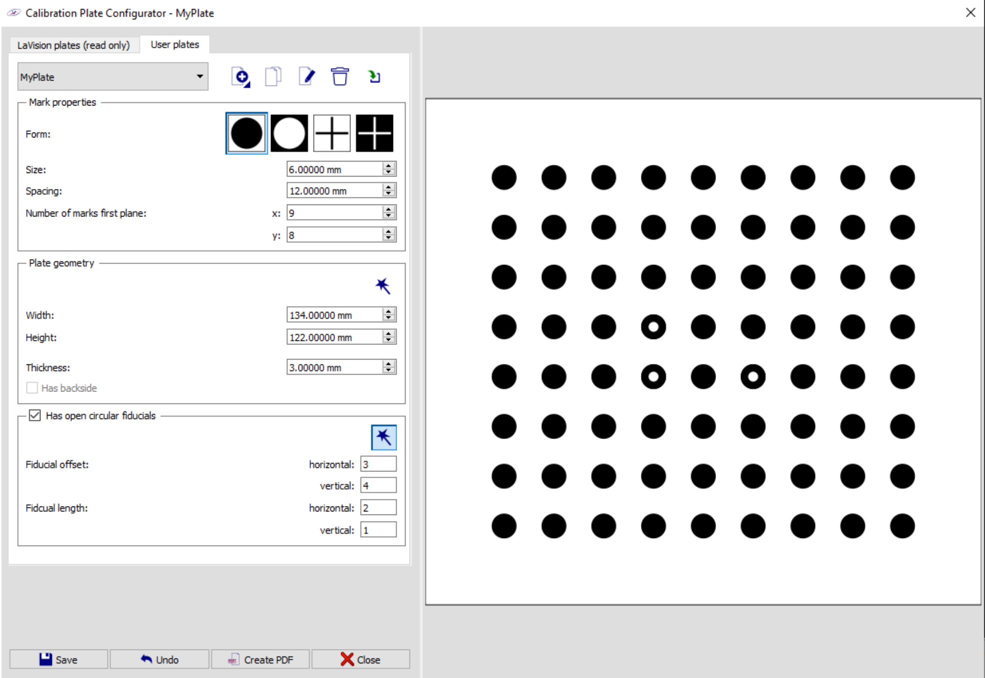
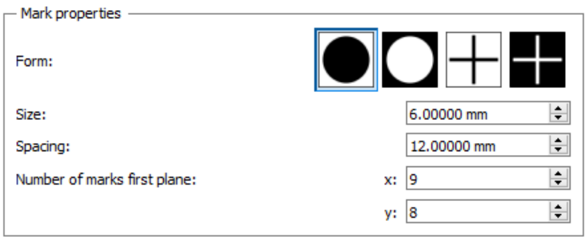


Figure C.4: User Plates view

Mark Properties



- ✓ Select the mark type of the calibration plate:
  - Black dot
  - White dot
  - Black cross
  - White cross
  - Black & white checkerboard
- ✓ Specify the diameter/the edge length of the marks.
- ✓ Specify the distance between the centers of the marks.
- ▷ The following configuration is recommended:
  - No normal mark between the two vertical fiducials.

- One normal mark between the two horizontal fiducials.

! The configuration must be unique; therefore, fiducials cannot use the same spacing in horizontal and vertical direction. Invalid configurations will be displayed in red.

✓ Specify the number of marks in x and y direction.

Plate Geometry

▷ Specifies the plate geometry.

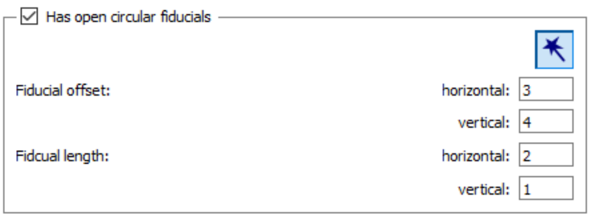
▷ The minimum size of the plate depends on your selection of number, diameter and distance of marks.

✓ Choose **Auto**  to calculate the minimum size automatically (recommended). Disable to manually set the outer dimension of the plate.

✓ Set width and height (if not **Auto** is ticked on).

✓ Define the thickness of the plate.

Fiducials



▷ For automatic detection of the plate orientation, special marks (fiducials) are required.

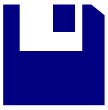
▷ Activating the option **Has open circular fiducials** will replace three normal marks with marks with open circles.

! This option is only available for circular marks, not for cross and checkerboard mark types.

✓ Choose **Auto**  to have the fiducials placed automatically.

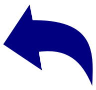
✓ At **Fiducial offset** specify the position of the lower left fiducial mark.

! Make sure that there is at least one line/ row of normal marks between the fiducials and the plate border! It is recommended to place the three special marks closely around the center of the plate.



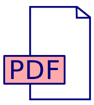
Save

▷ Saves the custom plate under the name (plate designation) you have specified.



Undo

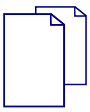
▷ Cancels all changes made since the last saved state.



Create PDF

▷ Creates a PDF-file of your designed plate.

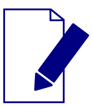
▷ The PDF can be used to let print your plate design onto a suitable plate.



Duplicate Selected Plate Specification

✓ Click on this icon to duplicate the displayed plate specification.

▷ This option is useful if you will create a plate with small variations of another given plate.



Rename Plate Designation

✓ Click on this icon to rename the plate designation.

Delete Plate Specification

✓ Click on this icon to delete a plate specification.

Import plates

✓ Click on this icon to import a plate specification, which is available as .XML-file on your computer.

- This option is useful if you will use a copy of a self-defined calibration plate specification on another computer.

D Other Filter Operations

D.1 Fourier Transform (FFT)

The Fourier transformation is a linear transformation from space or time to the frequency domain. Instead of displaying the data in the way it was measured, after applying the Fourier transform it is viewed as signal strengths of certain frequencies. Since the base functions of the Fourier transform are orthogonal and complete, there is no information lost in the transformation. Applying the inverse Fourier transformation will recover the original data again.

The Fourier transform (FT) is defined using integrals on continuous types of data. On the other hand, a computer stores digitised values at regular intervals in space. Here, the discrete Fourier transform (DFT), defined as the sum over discrete values, must be used. Finally, the fast Fourier transform (FFT) is just a fast algorithm for computing the DFT, which reduces the time for calculation tremendously. Due to the fast FFT-algorithm the FFT became so widely spread as a processing tool for all kinds of data in numerous applications.

For N-data points there are about N*N numerical calculations necessary using the DFT. Using the FFT this is reduced to $N\log_2(N)$ which for typical data sizes like 512 reduces the calculation time by a factor of 57!

The main application of FFTs are in performing filtering techniques, since in the frequency space many filters can be done by simple multiplication. Most of the times a low-pass filter is used to eliminate high-frequency noise, or a high-pass filter may suppress unwanted slow changes in the background.

FFTs are also used as a fast method for computing correlation between two data ranges, or to compute convolution and deconvolution of signals.

D.1.1 Basic FFT and Inverse FFT on Images

The *LaVision Pro Package* uses the basic 'butterfly'-algorithm enhanced by the possibility of using arbitrary long arrays of data. Instead of using the standard 128, 256, 512,... data length, any length of data points can be used (it must be even, though) while still leading to a very fast algorithm (Singleton).

The time of computation is dependant on how N (the number of data points) can be separated into factors. If N is close to a prime number (e.g. 2 * 137) the time of computation is not much faster than in the slow DFT-way of calculation. If N can be separated into low prime numbers (e.g. 2*2*3*3) the algorithm is very fast. Fortunately, the number of pixels used in CCD-cameras is most of the times ideal for applying the FFT.

If you take a part of an image for the FFT, simply be aware that some lengths of data points might require longer times.

The formula for calculating the FFT on some data array x_j is:

$$\begin{aligned} F_k &= \frac{1}{N}\sum_{j=0}^{N-1} x_j \; e^{i\;2\pi\, j\; k/N} \\ &= \frac{1}{N}\sum_{j=0}^{N-1} x_j (\cos(i\;2\pi\, j\; k/N) + i\sin(i\;2\pi\, j\; k/N)) \end{aligned} \tag{D.1}$$

where i is the imaginary number ($i^2 = -1$). The inverse FFT is defined as:

$$F_k = \sum_{j=0}^{N-1} x_j \; e^{-i\;2\pi\, j\; k/N} \tag{D.2}$$

The FFT can be applied to 2D-image data horizontally or vertically or both vertically and horizontally, which is called 2D-FFT.

The horizontal-FFT takes each row of the image and calculates the FFT from this data array. So, when applied to an image of size N*M (N-columns and M-rows) the resulting image is also of size N*M and the first row of the FFT-image is the Fourier transform of the first row of the original image. All rows are still independent. Spoken in terms of frequencies, the original image is analysed, in what respect there are horizontal frequencies (may be visualised as horizontal arrows) in each of the rows of the original image. The wavelengths corresponding to the FFT-frequencies correspond to the range of 1 pixel up to N pixel. The vertical-FFT works the same on columns of the image.

Since the FFT is linear, both the horizontal- and the vertical-FFT can be executed one after the other to yield the 2D-FFT, which is a measure of frequencies in arbitrary directions. The formula for the 2D-FFT is:

$$F_{nm} \quad = \quad \frac{1}{NM} \sum_{k=0}^{M-1} \sum_{j=0}^{N-1} x_{jk} \; e^{i\,2\pi\,k\,m/M} \; e^{i\,2\pi\,j\,n/N} \qquad \text{(D.3)}$$

Since the FFT works in principle on complex numbers and gives as a result almost always also complex numbers, even if the original data is real, we need to be concerned about the storage of the FFT-data in the buffers of the **DaVis**-program.

First of all, the FFT-operation only works on FLOAT-buffers and it always stores the result as a FLOAT-buffer. But since the result of the FFT is complex, we would need to store the result in **two** buffers, one for the real part and one for the imaginary part of the data. This would be true in general, but the software makes use of a feature which is true when the input data is just real instead of the general complex case.

When the input data is real, then the output data is symmetric in the real part and anti-symmetric in the imaginary part of the output-data. So only half the data needs to be stored, the other half is simply the same or negative. Exactly how the software stores the result of the FFT in a buffer is explained in an extra section, because it is not so simple to explain and, in general, it is of no interest for you.

What is important, though, is the way where the software stores the low frequencies and where it stores the high frequencies:

Rule:	**low frequencies are in the middle of the image,**
	high frequencies are at the outside.

This is different than what is expected from equation D.1, where the lowest frequencies are (k=0,1,...) at the beginning of the output data array, and the higher frequencies are to the right. The reason why the storage is done according to the above rule becomes apparent, when standard filters are applied to the FFT-images, since then the filters are always symmetric around the center, which makes them very easy to generate.

The lowest frequency (k=0 in eqn. D.1), stored at the center, is the average over all pixels (cos(..) = 1, sin(...) = 0). This average value is stored at the location N/2-1. For example, if N is 384, then the average is at pixel 384/2-1 = 191 (the first pixel is pixel 0). Or, if the 2D-FFT is used on an image of size 384 x 286 pixel, then the center with the average value is at position (191/142).

The inverse-FFT, of course, takes the storage details into account before using equation D.2. The result of the inverse-FFT is only real, since we started with a real image. Even after applying symmetric filters to FFT-data, the result of the inverse-FFT will always be real again.

Exercise

Let's use the **DaVis** program in Classic mode to try out the basic FFT and IFFT functions:

1) Load an image into buffer 1.

2.) Change buffer 1 into a FLOAT-buffer by making buffer 1 active and clicking on it again with the right mouse button to open the **Data Properties** dialog and the **Properties** card. There set the type of the buffer to FLOAT and click on **OK**. Remember that the FFT only works on FLOAT buffers.

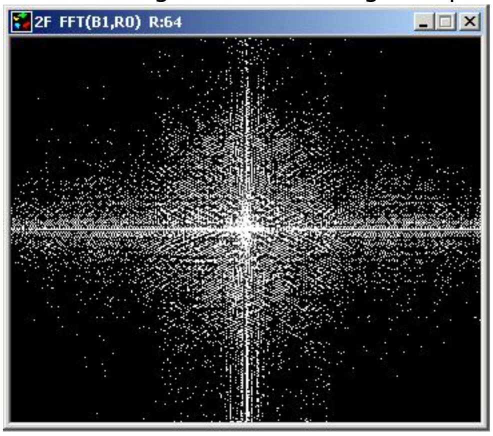
3.) Go to menu **Compute – FFT/IFFT**.

4.) Set the calculation **mode** to **2D**. You may also try **horizontal** or **vertical** to see the differences in the output.

5.) Run **FFT**, use all pixels of the image (select **Rectangle 0** as active rectangle) and store the result in buffer 2.

6.) Display buffer 2. Probably you don't see much. Go to resolution 64 or even multiply buffer 2 by 100 or 1000 with **F7** and `b2=b2*100` in order to see more details.

Notice how most of the image energy is in the center at the low frequencies and at the outer rim of the image, where the high frequency noise is stored.



7.) Run **IFFT** to convert the image back to normal. Store the result in buffer 3. If you have multiplied buffer 2 by 100 before, then the image intensity is 100 times higher in intensity.

Notice that the original image and buffer 3 are for all practical purposes exactly the same. Differences only appear in some minor digit due to rounding errors.

D.1.2 FFT-Filter

Once the data has been Fourier-transformed, it is easy to apply a standard frequency filter to it by simple multiplication of the Fourier data with an equal-length vector, which suppresses or enhances certain frequencies:

$$F'_k = F_k \bullet C_k \quad k = 0, 1, ..., N-1 \qquad \text{(D.4)}$$

where C_k is a frequency dependant filter function usually in the range of 0.0 to 1.0.

There are numerous ways of generating filters. For example, a low-pass filter is 1.0 for low frequencies and goes to 0.0 for high frequencies. A high-pass filter will be 0.0 for low frequencies and goes to 1.0 for high frequencies.

Translated to our way of storing the FFT-data with low frequencies in the middle, this means, for example for a low-pass filter, that the value of the filter function is 1.0 in the middle of the image and goes to 0.0 towards the outer rim of the image. So a very crude low-pass filter will simply set all pixels outside an inner rectangle to 0.0 (using the function **Set Outside Rect. = C**).

A filter needs to be defined only over half the image length, since it is symmetric around the center. For example, if the image size is 200 pixels (in either dimension) then it is sufficient to define the filter up to pixel 100.

In case the filter function length does not correspond with the image size, or in case a 2D-filter-function is applied to an image, where the width is not the same as the height (the standard case), the filter function is extended properly or cut to length if necessary.

For example, if the FFT has been computed in the 2D-mode, and the image size is 384 x 286 pixels, and the filter length is two times 100 as in the above example, then all pixels beyond the end of the filter (100) are set the the last value of the filter function at point N/2, which is 0 in the above example. This simply means that the low-pass filter is set to 0 for the very high frequencies. To be more precise, the center of the image is at 191/142, and in horizontal direction all pixels before pixel 191-100=91 and beyond pixel 191+100=291 are set to 0 and in vertical direction all pixels before 142-100=42 and beyond pixel 142+100=242 are set to 0.

In the 2D-case the image is multiplied with the filter function vertically and also horizontally. In the 1D-case, the image is only multiplied once either horizontally or vertically.

If the filter function is longer than the image data array it is simply cut to length. Of course, the program always centers the filter function correctly.

Applying a **high pass** filter it is very important to use the correct filter length, since cutting the filter to length or extending the filter in length may produce unwanted results. In the 2D-case it becomes necessary to use two different filters, vertically and horizontally, of different lengths to account for the fact that the image height is different from the image width.

Function **FFT Filter** in the **Fast Fourier Transform** dialog will use the first profile of the profile buffer. In the 1D-case the filter is only applied horizontally or vertically, in the 2D-case it is done in both directions.

Finally you can open the **Buffer Operations** dialog and use functions **Set Inside Rect. = C** and **Set Outside Rect. = C** for a fast low-pass or high-pass filter. We advise not to use these functions, though, since very crude filters will introduce extra noise into the source image as we show later on.

D.1.3 Using CL to design your own filter

You are not limited to the filter design menu to manipulate the FFT-data. Since the FFT-data is a standard FLOAT-buffer, it can be processed in any other way the program allows. Specifically you may design your own filter function (stored in a 'profile') and then multiply the FFT-image with the filter function. Or you may create a complete 2D-buffer to multiply the FFT-data with.

All functions in the filter design menu are actually CL-macros. The filters are generated using subroutine MakeFFTFilter() (profile 19), copied to profiles 17 (horizontal) and 18(vertical) and also processed further to account for the actual image size. Finally, the FFT-data is multiplied with profiles 17 and 18, either or both vertically and horizontally.

Below is a short CL-programming sequence to show how to use CL to apply a simple low-pass filter:

```
// simple horizontal low-pass filter

// let us assume the source image is in buffer 1

GetBufferSize( 1, nx, ny, type )
 // check image size

FFT( 1, 0, 0, nx-1, ny-1, 2, 0 )
 // horizontal FFT on whole image

// FFT is now in buffer 2

// here comes the definition of the filter function:

SetProfileBoundary( 19, 0, nx-1 )
 // set length of profile 19

for ( i = nx/2-1; i<nx-1; i++ )
   // right side of filter function

  F[ 19, i ] = 1.0 - ( i - nx/2 + 1 ) / ( nx /2 );
   // simple straight line to 0.0
```

```
for ( i = 0; i<nx/2-2; i++ )
     // fill left side of profile symmetrically

    F[ 19, i ] = F[ 19, nx - i - 2 ] // center = nx/2 - 1

    // pixel 0 = pixel nx-2, 1 = nx-3, ...

B[2] = B[2] * P[19] // apply filter horizontally

IFFT( 2, 3, 0 ) // horizontal IFFT
```

The next example shows how to multiply a buffer **vertically** with a profile. This is not so straight forward, since multiplication of a buffer with a profile is always done vertically. The trick is to generate a column, which is then multiplied with the FFT-data.

```
// Vertical multiplication of buffer 1 with profile 19.

// The result is stored in buffer 2.

B[2] = B[1] // copy buffer 1 to buffer 2

// the important side effect is that buffer 2

// has the same size as buffer 1.

C[2,0] = P[19] // copy profile 19 to column 0 of buffer 2

B[2] = B[1] * C[2,0] // this is the vertical multiplication
```

D.1.4 The Art of Filter Design

The important rule in defining FFT-filter is that the filter function should be as smooth as possible. There are only very few exceptions to this rule, for example, in case it is necessary to filter out a well known frequency, and only this one.

The transition functions in the filter design menu are chosen so that there is a smooth transition from one segment to the next.

To illustrate what the effect is of using a crude filter with sharp edges, a sample image is taken and filtered both with a crude filter and a smooth filter. The sample image has inside itself a rectangle of intensity 100 while the background is at 20 counts.

First of all we transform the image into Fourier space. Note that the Fourier transform of a square has the highest value in the center while decreasing

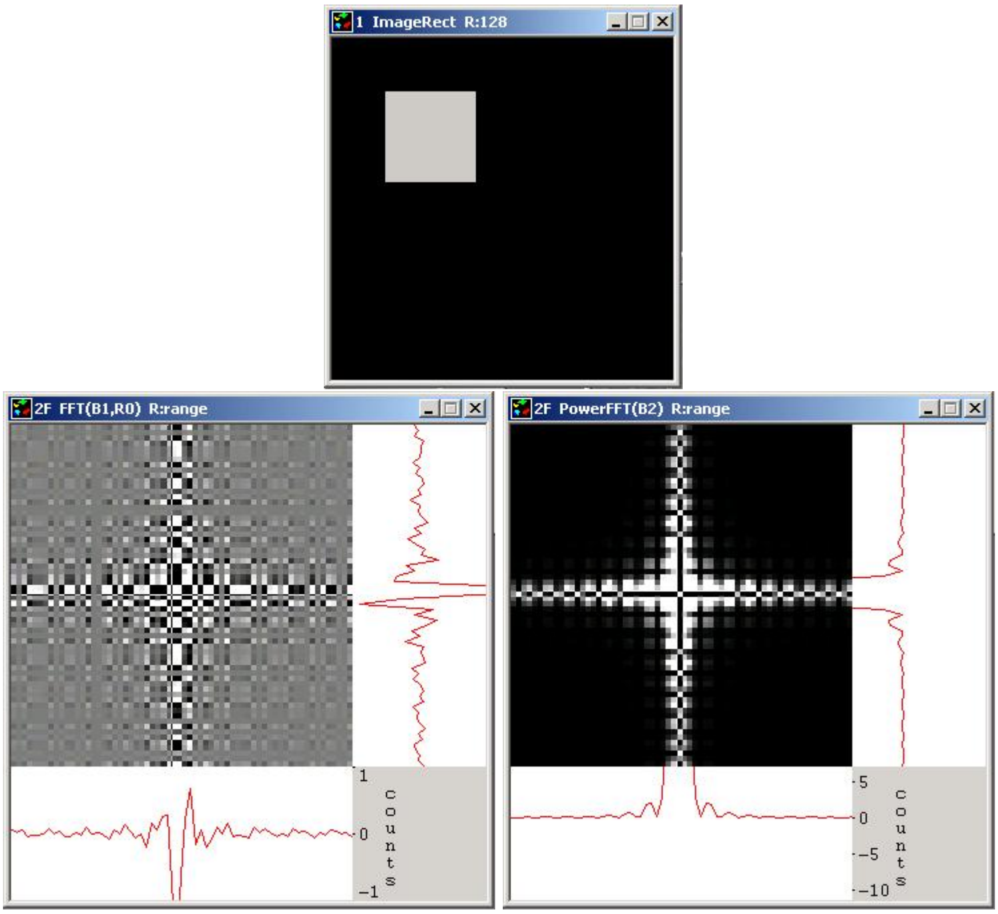


Figure D.1: Original image wth a small rectangle (top), the FFT (left) and PowerFFT (right).

progressively towards the outside. A sharp edge is the superposition of a series of progressively higher frequencies with progressively decreasing amplitude. Smoothing is the elimination of the higher frequencies so that the transition at the edge will be smoother.

Then a crude low-pass filter is applied with function **Set Outside Rect. = C** in menu **Buffer Operation**, setting all pixels to zero, which are roughly more than 10 pixels away from the center. After transforming back with **IFFT** we note an interesting effect in the final image. While the edges have been smoothed as we expected there is also a 'reverberation' of the square in the rest of the image. The background does not stay constantly at intensity 200, but is now swinging around 200. The reason for this is that the low frequencies are now no longer compensated by higher frequencies to produce a stable background. This means that our crude FFT-filter actually introduced **additional noise** into the image!

Let us compare it with a smooth filter designed with the filter design menu. Higher frequencies are not simply set to zero, but fade out towards zero. The result is that the background is now much smoother than before. Note that also the top region of the square is also now much more constant at intensity 100.

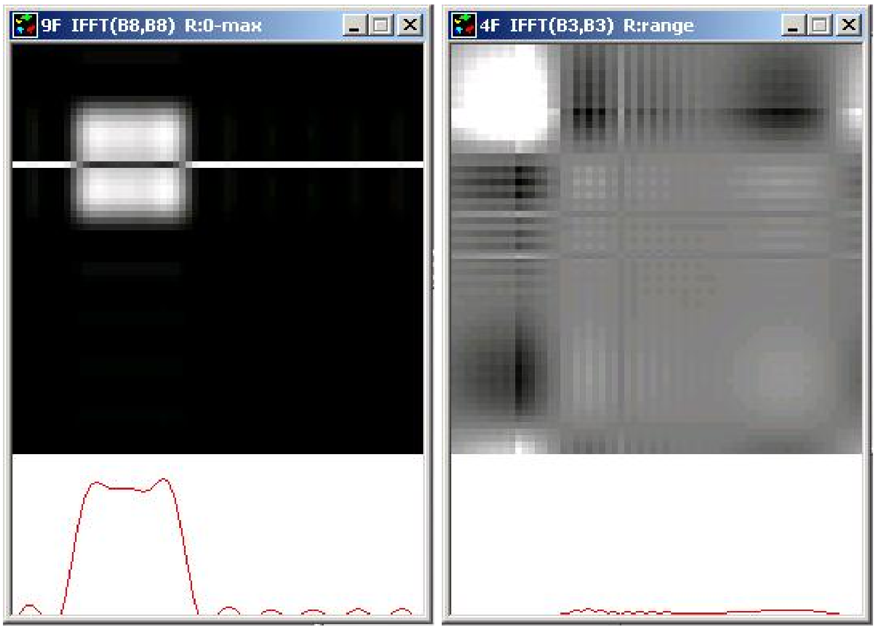


Figure D.2: Crude low-pass filter (left) versus smooth filter (right).

D.1.5 Mathematical Operation on FFTs

There are some useful functions in menu FFT math operations to work with FFT data in more detail.

First of all since the complex FFT data is stored in a kind of scrambled way to fit into a FLOAT buffer, it is useful to be able to separate the complex FFT buffer into two buffers where one is the amplitude part of the complex numbers and the other one is the phase. Mathematically every complex number can be expressed not only as a real part and an imaginary part, but also in terms of amplitude and phase, which in case of the FFT data is a much more useful representation of the real and imaginary values.

Any complex number F_k can be expressed as

$$F_k \quad = \quad R_k + iI_k \quad = \quad A_k \bullet e^{i\,P_k} \quad = \quad A_k \bullet (\cos P_k + i\ \sin P_k) \qquad \text{(D.5)}$$

with

$$A_k = \sqrt{R_k R_k + I_k I_k} \tag{D.6}$$

and

$$P_k = \tan(\frac{I_k}{R_k}) \tag{D.7}$$

where R_k and I_k are the real and imaginary part of F_k and A_k and P_k are the amplitude and the phase of F_k.

Function Amplitude computes the amplitude image of the FFT image, function Phase computes the phase. Again the low frequencies are toward the center of the image and the high frequencies are toward the outside. In the section on data storage there is more detail about where exactly every F_k of our fundamental Fourier equation D.1 is stored. Function Combine Amplitude and Phase will recombine an amplitude and phase image to an FFT-image, which can then be transformed back using IFFT. Note that these three function must know the type of FFT-data, whether it is a horizontal, vertical or 2D-FFT, because in each case the storage of the FFT-data is differently. For example, when the original image has been transformed using the 2D-FFT mode, then all subsequent operations must also been done in the 2D-mode (function Mode: must stay the same).

The separation of an FFT-image into amplitude and phase can be used to do some special filtering on the phase. In the standard case, when we apply a filter to the complete FFT-image, this is the same as if we apply the filter to the amplitude image only. But there are some special filtering techniques, which require, for example, to enhance or change the phase separately. There a number of good books on this topic. See the literature list for references (page 469).

The most often used application for the computation of the amplitude image is the computation of the power spectra, which is simply the square of the amplitude. Function Power-FFT simply computes the amplitude of the FFT-image and then squares the buffer.

Mathematically squaring the amplitude is the same as multiplying a complex number with it's complex-conjugate counterpart. 'Complex conjugate' means that the imaginary part is changed in sign.

$$\begin{aligned} F_k \bullet F_k^* &= (R_k + iI_k)(R_k - iI_k) \\ & (R_k R_k + iR_k I_k + iR_k I_k - i^2 I_k I_k) \\ & (R_k R_k + I_k I_k) \end{aligned} \tag{D.8}$$

In generalising the complex multiplication to two different numbers, function multiply $A \bullet B^*$ will compute the complex multiplication between two FFT-images A and the complex conjugate of B. If B is the same as A, then this function will compute the power spectrum according the above equation.

Similarly function multiply $A \bullet B$ does a complex multiplication without taking the complex conjugate of B. Those two function are mainly used for the computation of a correlation function or a convolution function.

There is no extra function for complex division, since it is rarely used and can be generated using the above functions. Defining complex division as the inverse of the complex multiplication, it can be best visualised in the realm of amplitude and phase. Here the complex multiplication is simply the multiplication of the amplitudes while adding the phases:

$$\begin{aligned} F_k \bullet G_k^* &= (A_k \bullet e^{i\,P_k}) \bullet (B_k \bullet e^{i\,Q_k}) \\ & (A_k \bullet B_k) \bullet e^{i\,(P_k + Q_k)} \end{aligned} \tag{D.9}$$

Therefore complex division is the division of amplitudes while subtracting the phases:

$$F_k / G_k^* = (A_k / B_k) \bullet e^{i\,(P_k - Q_k)} \tag{D.10}$$

In the same way the division by the complex conjugate G^* is done by adding the phases.

D.1.6 FFT Data Storage

The FFT data is stored in one FLOAT-buffer using the fact that the Fourier transform of a real data array is symmetric in it's real part and antisymmetric in it's imaginary part. This means, using the connotation of our basic equation D.1 and D.5:

$$F_k = R_k + iI_k \quad \text{with} \quad 0 \leq k \leq N-1 \tag{D.11}$$

the symmetry is given by

$$\begin{aligned} R_0 &= \text{data average} \\ R_1 &= R_{N-1}, R_2 = R_{N-2}, ..., R_{N/2} \\ I_0 &= 0, I_1 = -I_{N-1}, I_2 = -I_{N-2}, ..., I_{N/2} = 0 \end{aligned} \tag{D.12}$$

Stored in the FFT-buffer is only R_0 up to $R_{N/2}$ and I_1 up to $I_{N/2-1}$, i.e. a total of N real numbers. The storage is done in such a way that at the center of the image is stored R_0 and to the right are R_1 to $R_{N/2}$ and to the left are I_1 to $I_{N/2-1}$. The center of the image is at pixel location N/2-1. The highest frequency is actually N/2, indices higher than that are equivalent to lower frequencies.

Example: N=10, horizontal FFT. The FFT is stored as:

pix 0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
$I_4 = -I_6$	$I_3 = -I_7$	$I_2 = -I_8$	$I_1 = -I_9$	R_0	$R_1 = R_9$	$R_2 = R_8$	$R_3 = R_7$	$R_4 = R_6$	R_5

Horizontal and vertical FFTs are computed independently. This means that the 2D-FFT is done by first computing the horizontal FFT and storing the result in the above form in one buffer, then computing the vertical FFT on that buffer and again doing the rearrangement. Therefore in the 2D-case the final storage arrangement is quite complicated. There is actually no need to compute it, since it is much easier to use function **Amplitude** and **Phase**, in case direct access to each individual FFT-data point is required.

Using equation D.5 to D.7 as the definition as the amplitude and phase, functions **Amplitude** and **Phase** will store the result in the following way, again using our example of N=10:

pix 0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
$A_6 = A_4$	$A_7 = A_3$	$A_8 = A_2$	$A_9 = A_1$	A_0	$A_1 = A_9$	$A_2 = A_8$	$A_3 = A_7$	$A_4 = A_6$	A_5

and

pix 0	1	2	3	4	5	6	7	8	9
P_6 = $-P_4$	P_7 = $-P_3$	P_8 = $-P_2$	P_9 = $-P_1$	P_0 = 0	P_1 = $-P_9$	P_2 = $-P_8$	P_3 = $-P_7$	P_4 = $-P_6$	P_5 = 0

Similar in the 2D case the storage is (N=8, M=6, A_{nm}):

A_{54}	A_{64}	A_{74}	A_{04}	A_{14}	A_{24}	A_{34}	A_{44}
A_{55}	A_{65}	A_{75}	A_{05}	A_{15}	A_{25}	A_{35}	A_{45}
A_{50}	A_{60}	A_{70}	$\mathbf{A}_{00}$	A_{10}	A_{20}	A_{30}	A_{40}
A_{51}	A_{61}	A_{71}	A_{01}	A_{11}	A_{21}	A_{31}	A_{41}
A_{52}	A_{62}	A_{72}	A_{02}	A_{12}	A_{22}	A_{32}	A_{42}
A_{53}	A_{63}	A_{73}	A_{03}	A_{13}	A_{23}	A_{33}	A_{43}

Since the **power spectrum** is the square of the amplitude it is stored in the same way as the amplitude or phase image.

2D-**FFT-filter** must be designed in such a way that they are symmetric around R_{00} or A_{00}.

D.2 Correlation and Convolution

The Correlation between two arrays of data f_i and g_i of length N is defined as

$$Corr_k = \frac{1}{N} \sum_{i=0}^{N-1} f(i) \bullet g(k+i) \quad \text{(D.13)}$$

The factor 1/N is sometimes omitted, but simply out of practical considerations it should be included, since otherwise the numbers will get much too big to be displayed in any way.

$Corr_k$ is a measure for how closely the two data arrays agree with each other when they are shifted by k-data points. Therefore the correlation tells if a certain structure in one data array appears in the second data array again up to some displacement vector defined by k. Correlation are, for example, used for computing flow or particle velocity fields, where two images are taken shortly one after the other, and the correlation between the two images are computed. The correlation can be computed on the whole image, giving a measure of the average flow speed and direction, or it can be done on local sub-images to compute a 2D-velocity field. The sub-images, of course, must still be large enough so that in each one there is still enough information to compute a correlation.

The correlation can be easily extended into 2D:

$$Corr_{kl} = \frac{1}{NM} \sum_{i=0,j=0}^{N-1,M-1} f(i,j) \bullet g(k+i,l+j) \tag{D.14}$$

Mathematically it would be very time-consuming to compute the correlation according to the above equation, especially for a 2D-correlation of two 2D-image with a large number of pixels. But there is a mathematical relation between the correlation and the FFT. The correlation is the inverse FFT of the multiplication of the Fourier transform of the first data array with the complex conjugate FFT of the second data array:

$$Corr = IFFT(FFT(f) \bullet FFT(g)^{*}) \tag{D.15}$$

The time saving is more than a factor of 1000 for a 512 x 512 image. In menu Correlation/convolution function Correlation simply computes the correlation according to the above recipe. First the FFT is computed of the two source images, the complex multiplication is done and finally the result is transformed back.

D.2.1 Autocorrelation

The autocorrelation is the correlation of an image with itself. Mathematically this is simply the inverse FFT of the power spectrum, since the power spectrum is the result of the complex multiplication $F \bullet (F*)$. Of course, this can be computed with function Correlation by simply entering the same buffer for both source images. The auto correlation has always the highest value for k=0, since correlating the image with itself (without displacement) is a perfect correlation.

D.2.2 Convolution

The convolution between two arrays of data f_i and g_i of length N is defined as

$$Conv_{k} = \frac{1}{N} \sum_{i=0}^{N-1} f(i) \bullet g(k-i) \tag{D.16}$$

The convolution is nearly the same as the correlation, except that the second data array has been flipped over. So, for example, it can be used to detect similar structures in two different images, where the structure has been reversed in the second image.

Similarly the convolution can be computed via FFT by using

$$Conv = IFFT(FFT(f) \bullet FFT(g)) \tag{D.17}$$

Note that the multiplication is without taking the complex conjugate of FFT(g).

D.2.3 Cyclic versus Non-Cyclic Computations

In the definition of the correlation and the convolution, equations D.13 and D.16, you will notice that the index of, for example, $g(k+i)$ will run out-of-bound for high k's and i's. There are two solutions for this problem:

Cyclic computation:

In this case it is assumed that the source data will repeat itself indefinitely to the right and left. I.e. the cyclic extension of the data is

$$g(N) = g(0), g(N+1) = g(1), g(N+2) = g(2), \ldots$$

$$F_k = R_k + iI_k = A_k \bullet e^{i\,P_k} = A_k \bullet (\cos P_k + i\ \sin P_k) \tag{D.18}$$

and similarly to the 'left' in case of negative indices for computing the convolution. The advantage of using the cyclic extension of the data is that the computation of the correlation or convolution by using the Fourier transform can be done without further modifications, since the FFT is, by definition, also cyclic, i.e. it also assumes that the data will repeat itself in every dimension.

Note that in the cyclic case the result is symmetric because a displacement vector of k is the same as a displacement vector of N-k. Therefore it is sufficient to look at e.g. the two top quadrants of the image (in the 2D-case), because the other two quadrants do not contain any additional information.

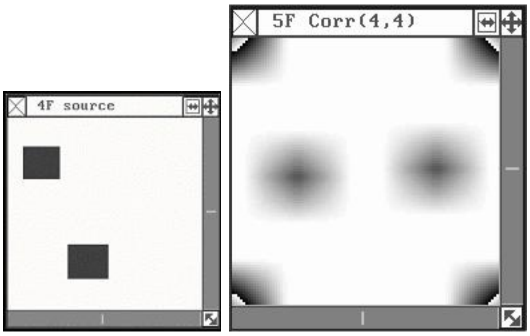


Figure D.3: Cyclic autocorrelation

Non-Cyclic computation:

In this case it is assumed that the source data will be 0 outside the specified data range 0-N-1. I.e. the extension of the data is

$$g(N) = 0, g(N+1) = 0, g(N+2) = 0, \ldots$$

and similarly to the 'left' in case of negative indices for computing the convolution. Another way of understanding this case is that in the equations D.13 and D.16 we are only computing the sum for those data values which actually exist. If any index is outside the bounds, it will not be added to the sum.

This case is most often physically much more meaningful, because there is usually no reason to assume that the data will repeat itself with just exactly that period N we choose to sample the data with.

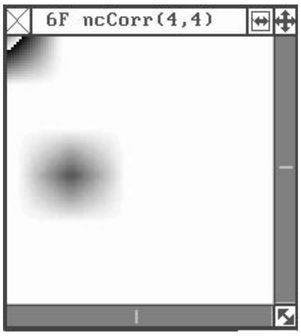


Figure D.4: Non-Cyclic autocorrelation

Now the question is, if we can compute the non-cyclic correlation and convolution still using the FFT. The answer is YES as long as we use a certain trick of zero-filling:

The trick is to double the source image and set all new pixels to zero. Then the FFT is computed just like in the cyclic case. We will end up with a resulting image twice the size as in the cyclic case, which we then cut in half. Since we doubled the N (and M in the 2D-case) we need to divide the image by the factor of 2 (1D) or 4 (2D).

Note that the result of the non-cyclic correlation is not symmetric.

This procedure is done automatically by function non-cyclic correlation and non-cyclic convolution. Since the FFTs and the IFFT is computed on an image twice the size, it will take longer than in the cyclic case. Since a number of intermediate temporary buffers are needed, make sure that you have not used up all your memory yet. You might have to clean-up your buffers (set them to zero size) before using these functions.

In some cases it is advantageous to fill only one image with zeros. For example, doing autocorrelation on parts of an image the best results are achieved, when a certain sub-image is picked out and stored in a separate buffer, and then an area twice as big symmetricallly surrounding that sub-image is taken as the counterpart for the correlation and stored in another buffer. The buffer with the sub-image itself is then doubled in size and filled with zeros. Finally the correlation is performed.

It is fairly simple to do those steps inside the program. Picking out an area of interest is done with menu data compression (compression factor = 1). Changing buffer sizes is done by clicking on the active buffer to go to the buffer parameter menu. When enlarging a buffer, the newly created areas are automatically set to 0.

Here it is shown how to do this using CL:

```
 // optimal non-cyclic AUTOCORRELATION of part of an image
// it is assumed that the relevant image is in buffer 1

SetRect( 1, 100, 100, 139, 119 );
    // rectangle 1:  40 x 20 pixels in size = sub-image
    // (if you let the user specify a rectangle, you
    // must make sure that there is enough space around
    // the selected rectangle)

SetRect( 2, 80, 90, 159, 129 );
    // rectangle 2:  80 x 40 pixels in size

SetBufferSize( 2, 80, 40, 1 );
    // make space for sub-image
```

```
B[2] = 0; // fill with zeros

SetBufferSize( 3, 80, 40, 1 )
    // make space for counter-image
B[3] = 0; // fill with zeros

MoveRectangle( 1, 1, 2, 20, 10 );
    // copy sub-image to center of buffer 2 !!

MoveRectangle( 1, 2, 3, 0, 0 )
    // copy surrounding area

FFT( 2, 4, 2 );     // 2D-FFT of sub-image
FFT( 3, 5, 2 );     // 2D-FFT of surrounding area
MultiplyFFT( 4, 5, 6, 3+2 ); // 2D-complex mult.  A * (B*)
IFFT( 6, 7, 2 );     // inverse FFT = correlation
SetBufferSize( 7, 40, 20, 1 ); // throw away symmetric parts
B[7] = B[7] / 4;     // renormalize
```

D.2.4 Literature

Adrian RJ, Christensen KT, Liu ZC (2000) “Analysis and interpretation of instantaneous turbulent velocity fields” , Exp. in Fluids, 29/3, p. 275-290

Ernst, Dr. E., “Einführung in die digitale Bildverarbeitung” , Franzis Verlag

Jähne, B., “Digitale Bildverarbeitung” , Springer-Verlag

Jaroslavskij, L.P., “Einführung in die Digitale Bildverarbeitung” , Hüthig Verlag

Blahut, R.E., “Fast algorithms for digital precessing. Reading, Mass.” , Addison-Wesley

D.3 Nonlinear Filter

The non-linear filters do not work with the simple matrix equation as the linear filters do. Each non-linear filter has its specific advantages, but on the other hand they are non-linear operations, i.e. the average intensity of the image may change.

For the three filters **Median, Erosion and Dilatation** a $k \times k$ area (k= 3,5,7,9 or 11) is taken and all elements are sorted according to their intensities. Then the elements are stored back following a special rule for each filter.

Median Filter

The **middle element** of the sorted $k \times k$ area is stored back as the new center pixel. It eliminates high frequency noise. As an advantage compared to many linear filters it does not eliminate sharp edges.

Erosion Filter

The **element before the middle element** of the sorted $k \times k$ area is stored back as the new center pixel. The effect is a kind of erosion. Valleys become deeper and finally, after having applied the filter several times, the image will be eroded away towards the background level.

Dilatation Filter

It does exactly the opposite of the Erosion filter. The **next element after the middle element** in the sorted $k \times k$ area is stored back as the new center pixel. The effect is that edges become sharper and plateaus flatten.

Concentration Filter

Concentrates intensities on the "local maximum" pixel. First of all the filter searches for pixels whose intensity is a local maximum. The intensity In addition has to be above a background level (set with the parameter **noise-level**). If found, it will move the intensities of the surrounding pixels toward the center to concentrate all power in the middle.

Intensity normalization (image-avg)/rms

The Image normalization function subtracts the local average (with the filter length given) and divides the result by the local intensity standard deviation. The result is an image with local mean intensity zero and a fixed dynamic range everywhere.

D.3.1 The Math behind NonlinearFilter

Sliding Average Filter:

This filter computes a local average over a specified scale length by computing the average according to the following equation applied 4-times, by going from left to right, from right to left, from top to bottom and from bottom to top through the image:

S^{avg} (0) = I (0)

and

S^{avg} (i) = (n-1)/n * S^{avg} (i-1) + 1/n * I (i) for $i \geq 1$

where I(i) is the pixel intensity at a certain pixel i and S^{avg} (i) is the computed average at that pixel and and S^{avg} (i-1) is the computed average at the previous pixel. N is the scale length (must be $n \geq 2$), which corresponds about to the distance over which the average is computed.

The effect is a sliding average, where each new pixel is added with a small weighting factor. This type of filter is much faster than any Fourier type of average filter or a linear $n \times n$ matrix filter, since very few multiplications are necessary for each pixel, independant on the scale length. There is a small edge effect at the image borders, but by far not as severe as when using a FFT-Filter. The only drawback is that the image intensities are shifted a bit in the direction of processing (an intensity peak becomes smoothed and the peak center is shifted by n/2). But this is compensated by going not only from left to right, but also from right to left.

The exact equations for the 4 passes are decribed below:

- Going from left to right: For all rows y do:
  S^{avg} (x,y) = (n-1)/n * S^{avg} (x-1,y) + 1/n * I (x,y)
- Going from right to left: For all rows y do:
  S^{avg} (x,y) = (n-1)/n * S^{avg} (x+1,y) + 1/n * S^{avg} (x,y)

- Going from top to bottom: For all columns x do:
  S^{avg} (x,y) = (n-1)/n * S^{avg} (x,y-1) + 1/n * S^{avg} (x,y)

- Going from bottom to top: For all columns x do:
  S^{avg} (x,y) = (n-1)/n * S^{avg} (x,y+1) + 1/n * S^{avg} (x,y)

For example, if the scale length is 10 pixel, then

S^{avg} (i) = 0.9 * S^{avg} (i-1) + 0.1 * **I** (i)

Sliding Minimum Filter:

This filter computes a local minimum over a specified scale length. The filter has the same principle as the sliding average, but with slightly different equations:

With m = n / 2 half of scale length, do

Step 1: going from left to right: For all rows y do:

fmin = S^{avg} (0,y) = **I** (0,y)

and

if (**I** (x,y) $>$ fmin)

fmin = (m-1)/m * fmin + 1/m * **I** (x,y)

else

fmin = **I** (x,y)

S^{avg} (x,y) = (n-1)/n * S^{avg} (x-1,y) + 1/n * fmin

where fmin is an intermediate variable for storing a sliding minimum over half the scale length, which again is smoothed over the full scale length.

Step 2-4: same as for sliding average

Note that this filter can still produce sometimes pixel intensities that are slightly above the original values. The strict sliding minimum filter explained below is strictly below the original intensities, but is not as smooth as this filter.

Sliding Maximum Filter:

Same as sliding minimum, only '$<$' instead of '$>$' in the above equation in step 1.

Strict Sliding Minimum Filter:

Same as sliding minimum, except that none of the resulting pixel intensities will be below the intensity of the original pixel at a certain position. The equations are slightly different:

- Going from left to right: For all rows y do:

  fmin = S^{avg} (0,y) = I (0,y)

  and

  if (I (x,y) $>$ fmin)

  fmin = (m-1)/m * fmin + 1/m * I (x,y)

  else

  fmin = I (x,y)

  S^{avg} (x,y) = min(fmin, (n-1)/n * S^{avg} (x-1,y) + 1/n * fmin)

- Going from right to left: For all rows y do:

  S^{avg} (x,y) = min(S^{avg} (x,y) ,
  (n-1)/n * S^{avg} (x+1,y) + 1/n * S^{avg} (x,y))

- Going from top to bottom: For all columns x do:

  S^{avg} (x,y) = min(S^{avg} (x,y) ,
  (n-1)/n * S^{avg} (x,y-1) + 1/n * S^{avg} (x,y))

- Going from bottom to top: For all columns x do:

  S^{avg} (x,y) = min(S^{avg} (x,y),
  (n-1)/n * S^{avg} (x,y+1) + 1/n * S^{avg} (x,y))

Strict Sliding Maximum Filter:

Same as strict sliding minimum, only 'max' instead of 'min', and '<' instead of '>' in the above equation in step 1.

Sliding Average Filter (gaussian profile)

The algorithm follows the same pattern as the sliding average filter described above: making horizontal passes left to right and back again and averaging it with the vertical passes top to bottom and back again.

But in contrast of using a sliding value and the current pixel only, the gaussian profile filter uses the last 3 pixels in an iterative convolution to create an approximation to a gaussian bell 2D filter. Below the horizontal pass is described exemplarily :

- set first three destination pixels $I_d(0,y)$, $I_d(1,y)$ and $I_d(2,y)$ to the average intensity of the three source pixels $(Is(0,y)+Is(1,y)+Is(2,y))/3$
- then starting from $x = 4$ calculate the next pixels $I_d(x,y)$ iteratively using the pixel $I_s(x,y)$ and previous destination pixels $I_d(x-1,y)$, $I_d(x-2,y)$ and $I_d(x-3,y)$:

  $$I_d(x,y) = B \times I_s(x,y) + \frac{B_1 \times I_d(x-1,y) + B_2 \times I_d(x-2,y) + B_3 \times I_d(x-3,y)}{B_0}$$

with the following constants

$$\begin{aligned}
\sigma &= filter\ length/2.0 \\
Q &= 0.98711 * \sigma - 0.96330 \quad for \quad \sigma \geq 2.5 \\
& \quad 3.97156 - 4.14554 * \sqrt{1 - 0.26891 * \sigma} \quad for \quad \sigma < 2.5 \\
B_0 &= 1.57825 + 2.44413 * Q + 1.4281 * Q^2 + 0.422205 * Q^3 \\
B_1 &= 2.44413 * Q + 2.85619 * Q^2 + 1.26661 * Q^3 \\
B_2 &= -1.4281 * Q^2 - 1.26661 * Q^3 \\
B_3 &= 0.422205 * Q^3 \\
B &= 1 - (B_1 + B_2 + B_3)/B_0
\end{aligned}$$

The same procedure is done backwards again. The vertical pass is equivalent with x and y exchanged moving from top to bottom and back again working on the horizontally smoothed image.

Please note that the filter length of this filter is defined as a float value (rather than an integer value of the *sliding average* filter) and hence a finer adjustment of the smoothing level is possible.

Reference: *A.Lukin* "*Tips and Tricks: Fast Image Filtering Algorithms*" in *Proceedings of GraphiCon'2007, Moscow, Russia, June 2007, pp. 186-189*

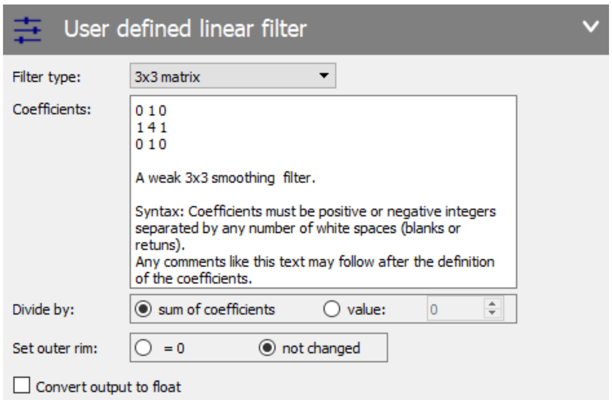
Filtering horizontally

Same as the above described modes, but the filters are only applied horizontally, i.e. only performing steps 1 and 2 in the above equations. If one would like the sliding filter applied only vertically, one needs to rotate the buffer by 90° , apply the filter horizontally, and rotate back 90° .

D.4 Operation for Linear Filters

The dialog for linear filters can be found in the processing operation. Here the user can choose between predefined filters, such as **Smoothing**, **Gaussian Smoothing**, **Sobel**, **Laplace**, **Compass**, **Sharpening**, and **User filters** that can be changed by the user.

D.4.1 User defined linear filters



Select a **filter type** from the list. The other items display the filter matrix and some other settings, see sections below.

Filter Matrix

Usually the filter operates as a NxN-matrix, which is placed over the image. For the selected filter the matrix is always displayed at the bottom of the dialog box. The new value of a pixel (P'ij) is calculated from a **linear combination of the surrounding pixel values** and the center pixel ($P_{i,j}$) itself. The result is normalized by a "divisor".

The 3x3 filter

$$\begin{pmatrix} a1 & a2 & a3 \\ b1 & b2 & b3 \\ c1 & c2 & c3 \end{pmatrix} / divisor$$

leads to the new value $P'_{i,j}$:

$$P'_{i,j} = (a_1\ P_{i-1,j-1} + a_2\ P_{i,j-1} + a_3\ P_{i+1,j-1} +$$
$$b_1\ P_{i-1,j} + b_2\ P_{i,j} + b_3\ P_{i+1,j} +$$
$$c_1\ P_{i-1,j+1} + c_2\ P_{i,j+1} + c_3\ P_{i+1,j+1}\ ) / \text{divisor} .$$

1D-Vector Filter

There are two predefined vector filters for smoothing over 9 pixel in one dimension (**horizontally or vertically**). Other 1-dimensional filters can be freely defined as **user filters**.

Vector filters are very useful for analyzing, e.g., 1-dimensional spectra, which have another information on the vertical axis of the image (e.g. space coordinate or time). Maybe, just the spatial axis should be smoothed, but the spectral resolution on the horizontal axis is kept as it is. In this case use the vertical smoothing filter.

D.4.2 User Filter

The user can change any of the settings of the predefined filters but not change the name and not add a new filter.

Choose the **type** of linear user filter.

Define the filter by entering the matrix or vector coefficients inside the corresponding field directly.

Two further actions (**divide by** and **set outer rim**) are defined at the bottom of the dialog box. They are performed before the result of the filter is stored.

The **description** entered here will appear below the list of linear filters, when the filter has been selected (see the page before).

Basically, there are three **types** of user filters.

- **NxN matrices**: Usual NxN-matrices (N= 3, 5, 7 or 9) that are applied in the same way as the predefined linear filters. They are defined by entering the matrix elements in the center field.

- **NxN separable matrices:** They work much faster than the usual matrices. If a matrix is separable (is the product of a vertical times a horizontal 1D-vector), several steps of the full matrices operations are redundant. Therefore it can be applied in two steps of vector filter operations: first a horizontal vector, second a vertical vector. These filters are also defined by entering the elements directly into the vectors.
- **1D vectors**: Vector filters in one dimension.

For **seperable matrices** and **vector filters** enter the **coefficients** into the vectors (horizontal and/or vertical). Enter **normal matrice** filters inside the matrice.

Negative or floating point values can only be stored in image buffers of the type FLOAT. So it might be necessary to **convert the output buffer** to a FLOAT buffer, in order not to lose significant digits of the data.

After performing the matrix (or vector) operation, the result buffer can be **divided by** a specified value.

- **sum of coefficients:** Choose this divisor in order not to change the average intensity of the image buffer.
- **value:** Any other divisor can also be defined.

At the **rim** of the image a filter will not work over the same amount of pixels as in the middle. Therefore the result might differ drastically at the rim, espacially if a gradient filter is used. Then it might be useful to set the whole **outer rim** equal to 0. In the default setting the rim of the result buffer is not changed.

Index

Glossary

Buffer
Collection of image or vector data and additional information like scales, see detailed description on page 93.

Recording
Set with acquired camera images and e.g. additional data from AD converter.

Scale
Mapping from pixel position of the buffer to real world coordinates or from image intensitiy (counts) to real world value. Scales are displayed in image and vector views and in the status bar for the active mouse position.

Set
DaVis uses **SETs** to store parameters and data files as *single objects*. For example all acquisition parameters and the acquired images are stored as a SET. Most dialogs are supporting the SET structure to make the usage of large numbers of files easier.

Stream Set
Result of highspeed recording with additional AD data. The Stream Set includes severel sets with different recording parameters.

Manual No 1003001 - November 2, 2022

FOCUS ON IMAGING

LaVisionUK Ltd
2 Minton Place / Victoria Road
Bicester, Oxon, OX26 6QB / UK
www.lavisionuk.com
Email: sales@lavision.com
Tel.: +44-(0)-870-997-6532
Fax: +44-(0)-870-762-6252

LaVision GmbH
Anna-Vandenhoeck-Ring 19
D-37081 Göttingen, Germany
www.lavision.com
Email: sales@lavision.com
Tel.: +49(0)551-9004-0
Fax: +49(0)551-9004-100

LaVision, Inc.
211 W. Michigan Ave., Suite 100
Ypsilanti, MI 48197, USA
www.lavisioninc.com
Email: sales@lavisioninc.com
Phone: +1(0)734-485-0913
Fax: +1(0)240-465-4306