Stereo Vision Toolkit

v1.3

User Guide

Revision date: 03/07/2020

Contact Details

Correspondence Address:	I3D Robotics Ltd Pipers Business Centre 220 Vale Road Tonbridge Kent TN9 1SP UK
Email:	info@i3drobotics.com
Phone:	+44 (0)1732 373020 (09:00 – 17:00 UK)
Fax:	+44 (0)1732 373001

Revision date: 03/07/2020

About this Manual

Who is this for?

This manual is intended for a user with I3D Robotics Stereo Camera. It will help the operator understand how to get the most out of this new product using the Stereo Vision Toolkit

What is in it?

This manual provides the operator with instructions on how to install and use this software package.

Installation	Instructions on install the software package
How do I use it?	How to use the stereo vision toolkit
FAQ	Frequency ask questions. Will help will assist with commonly known problems.

Disclaimer

THE INFORMATION CONTAINED HEREIN IS PROVIDED "AS IS" WITHOUT WARRANTY, CONDITION OR REPRESENTATION OF ANY KIND, EITHER EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING BUT NOT LIMITED TO, ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL I3D ROBOTICS BE LIABLE FOR ANY LOSS OR DAMAGE, WHETHER DIRECT, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL OR OTHERWISE HOWSOEVER CAUSED WHETHER ARISING IN CONTRACT TORT OR OTHERWISE, ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE INFORMATION PROVIDED HEREIN.

Revision date: 03/07/2020

Contents

1 Introduction

Software compatibility
Windows 7+ (64bit) **cross-platform support coming soon**

1.1 System requirements

The table below outlines the minimal requirements of the connected computer running the user software.

Processor	Intel i3/5/7 or similar
RAM	4GB
Hard disk space	500MB
Graphics card	CUDA compatible card recommended, but not required.
Connectivity	USB3

2 Installation

2.1 What should I have?

- Stereo camera system
- USB3 Cable / GigE ethernet cable depending on camera system
- USB drive provided by I3DR containing software, documentation and factory calibration
- If using GigE stereo camera then a network inject is required to directly connect GigE the machine. Alternatively plug the GigE camera directly into your router or network switch for the network you are connected to.

2.2 Using installer from USB key

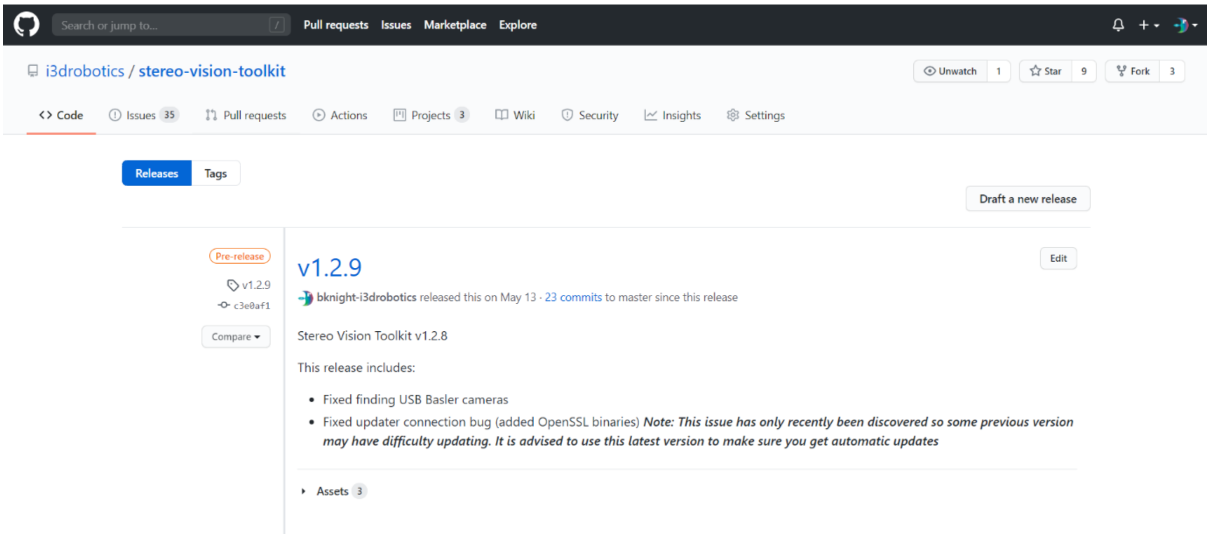
An installer for the toolkit software should be provided on the USB drive from I3DR. Look for a file with the name 'StereoVisionToolkit-{VERSION}-Win64.exe'. Once you have located this file, follow the **Installation process (2.4)**.

Alternatively, you can download the latest release from our GitHub repository. See **Using installer from** GitHub Releases **(2.3)**.

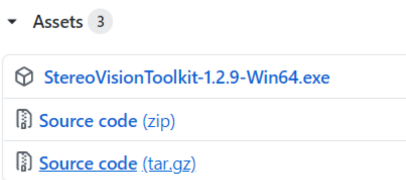
2.3 Using installer from GitHub Releases

You can also find the latest release of the Stereo Vision Toolkit on the GitHub repository using the following link www.github.com/i3drobotics/stereo-vision-toolkit/releases.

You will be presented with the following page:



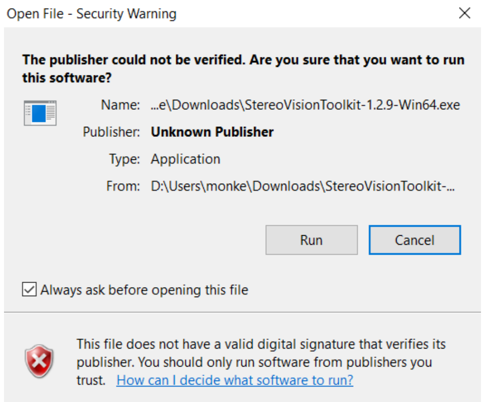
The latest release will be at the top of the page. Click 'Assets' to view the assets of this release, this includes a windows installer as well as the full source code for building the application for yourself.

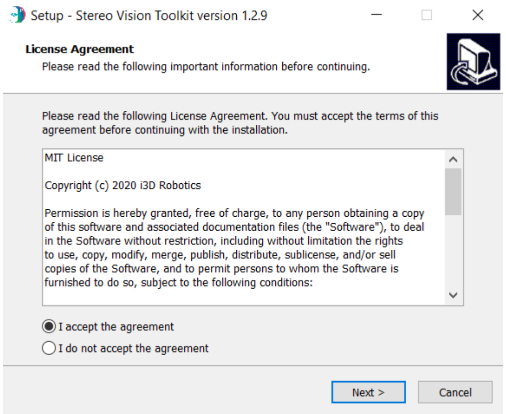


Download the installer by clicking on the 'StereoVisionToolkit-{VERSION}-Win64.exe'. This will start the download, which usually places the file in the 'Downloads' folder of the PC.

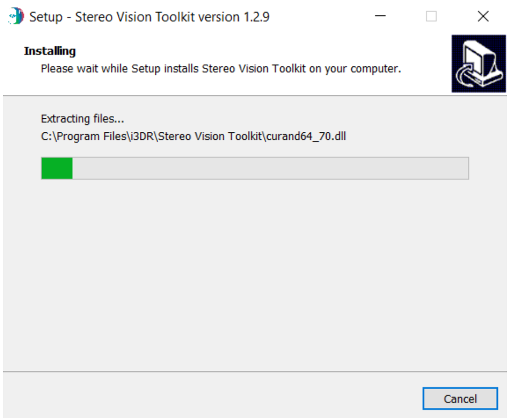
2.4 Installation process

Double click the installer file to start the installation process. You may see a security message from windows, this is perfectly normal. Select 'Run' to continue to open the installer. You may also be prompted to allow the install from the UAC in windows. This is an extra layer of security to make sure you intended to install new software. You should select 'Yes' to continue to run the installer.

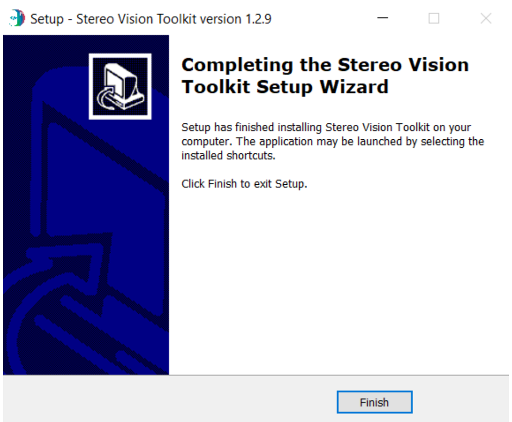




At this point the installer will be running, accept the license agreement using the checkbox labelled 'I accept the agreement' then press 'Next >'. You will be given the option to create a Desktop shortcut. This is useful for finding the Stereo Vision Toolkit however is completely optional. Make your choice and select 'Next >'. Now click 'Install' to start the installation process.



The installer will start installing and show the progress via the green progress bar. A few windows may pop up when the camera drivers are installed, this is expected behaviour.



When the installation process is complete the screen on the left will be shown. The Stereo Vision Toolkit is now installed and read to use. Exit the installer by pressing 'Finish'.

3 License

3.1 Why is a license required?

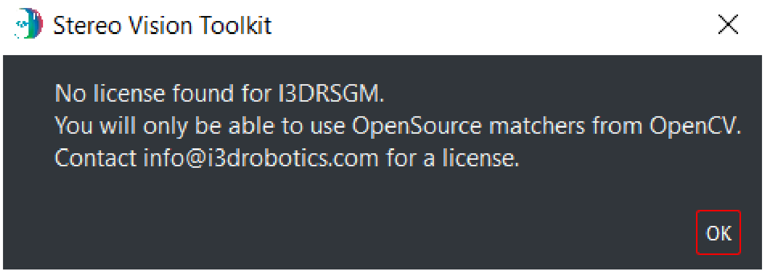
I3DR's has developed a high speed and effective stereo matcher that outperforms the open source matchers such as OpenCV BM and SGBM. To protect our IP the Stereo Vision Toolkit will only allow access to this matcher with a valid license. The Open Source elements of the Stereo Vision Toolkit are freely available and will not be affected if you do not purchase a license however to get the best results we advise using our I3DRSGM.

3.2 Where to find my license

Your license file should be provided on the USB key along with the software and calibration files. The license file is a file with the extension '.lic' and has key license information that is read by the Stereo Vision Toolkit to check the current machine has permission to run I3DRSGM. If you have lost your license or would like to request one please contact us info@i3drobotics.com.

3.3 How to use license

When requesting a license, you provide us with some minor information on your PC that is used to identify it and check the correct machine is using the license. This information is stored inside the license file. To let the Stereo Vision Toolkit read this information, this file must be copied into the location where the software was installed. By default, this location is 'C:\Program Files\i3DR\StereoVisionToolkit\'. To check the license is accepted run the Stereo Vision Toolkit. If the license is invalid, then a message will be displayed stating:



However, if the license is accepted then I3DRSGM will be available to you in the matchers list after you connect a camera.

3.4 Future improvements to licensing

We are currently in development of a new licensing system that will allow for online licensing. This will avoid the need to provide us with PC information and allow multiple machines to use the same license. If this sounds of interest to you, please contact us and we will inform you of the progress towards this and timelines to making this available.

4 Operation

4.1 Using the Stereo Vision Toolkit

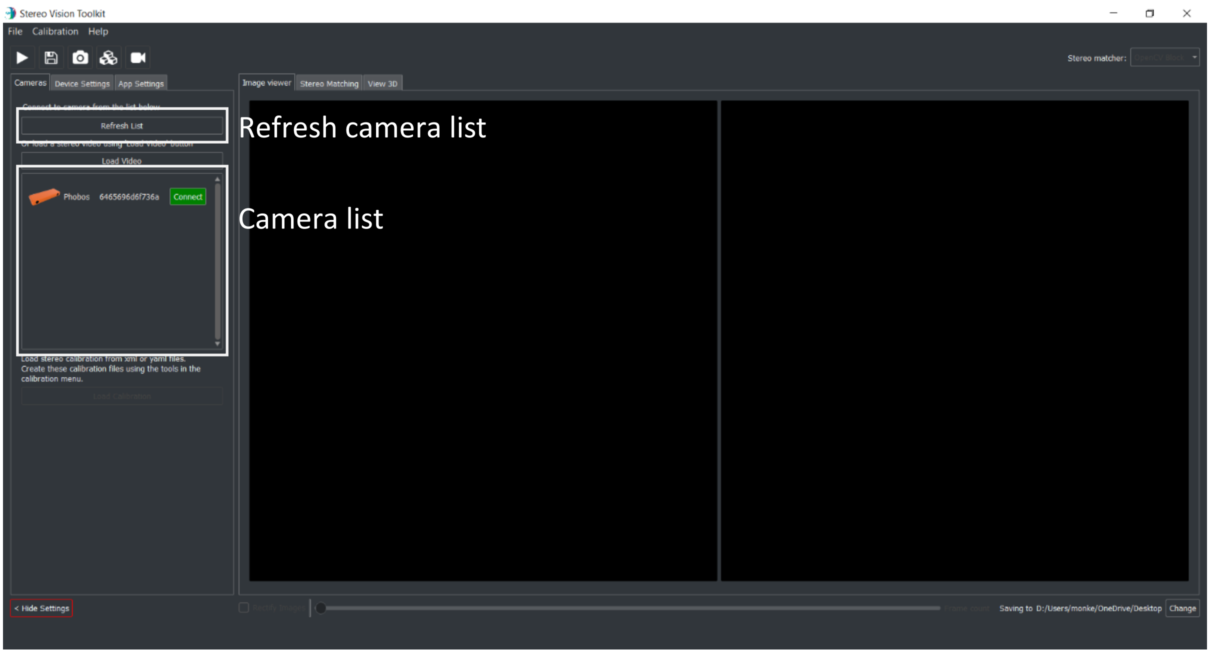
We provide an open source GUI tool to experiment with your camera. The tool can be used to acquire images, perform stereo image matching, and view live 3D point clouds. For most applications, we would expect that customers will integrate the camera using their own software (e.g. using the OpenCV library), but our toolkit is designed for rapid application prototyping.

The software installer is provided on a USB stick, or may be downloaded from our Github repository (https://github.com/i3drobotics/stereo-vision-toolkit) either a windows installer from the releases section, or as source which you can compile yourself. If you wish to compile from source, we provide library dependencies built using Visual Studio 2017 x64 in the repository.

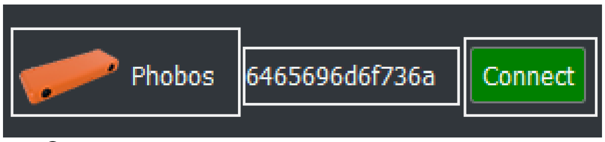
4.2 Connecting to camera

First, make sure your stereo camera is plugged in. If using USB preferably USB3 to achieve the best framerate. If using GigE, then either connected to the machine running the software directly or into the same network.

Open the software, you will be presented with an interface which is mostly disabled until a camera is connected. The image below shows this view, on the left a list of cameras is shown. This will display all the camera currently available which will refresh every 5 seconds. You can manually refresh this list by pressing the 'refresh list' button.



The interface will show you the type of camera which can be identified by the icon shown. Deimos and Phobos will be displayed as icons for quick and easy identification. Also provided is the serial number of the camera. This should be identifiable on your camera via a sticker attached to the device.

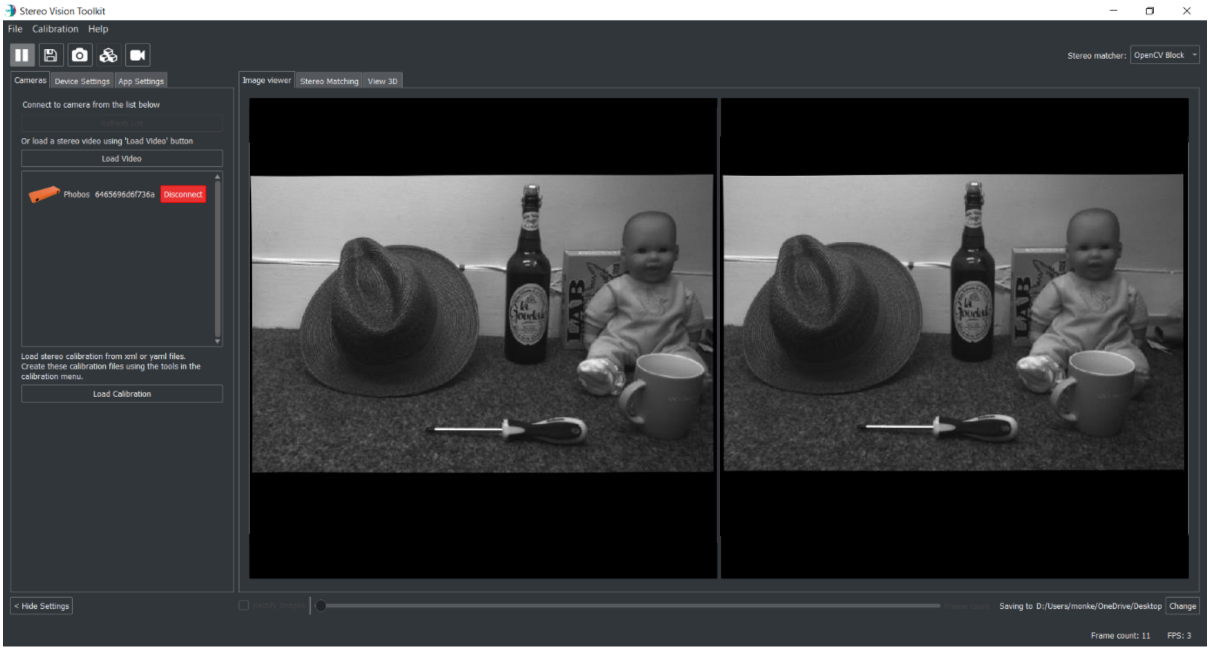


Connect to camera

Camera type

Camera serial

Connect your chosen camera in the list by clicking on the green 'connect' button. This will connect and initialise the stereo camera after a few seconds. Once connected the connect button will change red and read 'disconnect'. The camera will automatically start capturing and display the left and right images to the interface.

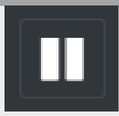
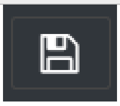
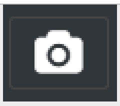
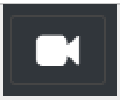


Now the camera is connected all the features of the Stereo Vision Toolkit are available to you.

4.3 Exploring the interface

The different features of the toolkit are arranged by tab. The first tab shows the stream from each camera. You can check the frame rate in the status bar at the bottom. Also, in the status bar, you can see the frame count[1] and the current output directory.

The toolbar provides you with the following options:

Command	Button	Description
Play/Pause		Start or stop the camera image stream
Save		Save the current frame to a user-specified folder
Snapshot		Grab a single frame and then pause
Enable matching		Toggle stereo matching
Record video		Start or stop recording a stereo video

At the top right, you can specify which matcher to use (I3DRSGM, OpenCV Block, or OpenCV SGBM). I3DRSGM will not be available if a valid license is not provided, this will cause the I3DRSGM option to be greyed out.

[1] This can be helpful as an indicator that the camera is still capturing images.

4.4 Camera settings

Camera settings will be show based on the camera used this is because not all cameras use all the same settings. The table below lists all the settings available.

Camera setting	Description
Exposure	Millisecond amount of exposure. Increase to let more light into the camera. This will increase motion blur. Try to set this as low as possible whilst still able to see the image as motion blur and over exposure cause problems for stereo matchers.
Gain	Gain filter applied to capture signal. Higher gain will brighten the image but will increase noise. It is advised to set this to 0 for best matching response as noise causes problems with stereo matchers. Auto gain will adjust the gain dynamically based on the scene.
FPS	Frames per second. When hardware trigger is off this will set the software frame capture rate. Turning trigger off will mean the left and right images may not be synchronised so this will reduce the effectiveness of the matchers. When hardware trigger is on this will update the camera controller trigger rate via serial interface.
HDR	HDR mode is useful for suppressing highlights in the scene, and does not affect the maximum framerate, especially in conjunction with auto-expose mode.
Binning	Down sample factor to apply to the image. This will increase capture rate as less data then needs to be transferred over the connection interface. This will also increase matching speed. However, increased binning will mean the matching is less accurate.
Packet delay	GigE only setting for adjusting the delay between right and left data packets. This is only required if on a slow network images are having trouble getting through.
Packet size	GigE only settings for adjust size of data packets. If network interface allows jumbo packets this should be set the match the largest jumbo frame available (e.g. 9000).

4.5 Device settings

Setting	Description
Swap LR	Swap the left and right image. Can be useful if the left and right cameras are the wrong way round. Usually no necessary to set this. Better to fix the hardware.
Downsample	Downsample factor to apply to image. Increasing this will speed up matching but reduce the accuracy. It is advised to set the binning on the camera settings to increase the overall capture and matching speed.

4.6 Calibration

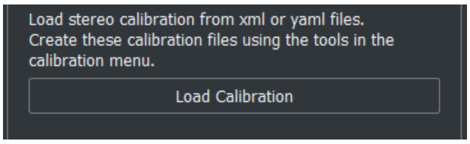
4.6.1 What is calibration

Calibration is required to reconstruct 3D images using any stereo camera. The purpose of camera calibration is to obtain the intrinsic parameters of each camera, which include the focal length of the lens, the intersection between the lens optical axis and the sensor plane (camera centre) and the characteristics of the lens' distortion. This is sufficient to 'undistort' a single camera – a simple way to think about it is that straight lines in the world become straight in the undistorted image. For stereo systems, an extrinsic calibration is also needed which describes the relative positions of the cameras with respect to each other. Combined with the intrinsic calibrations for each camera, a rectification transform is calculated which warps the raw images prior to stereo matching.

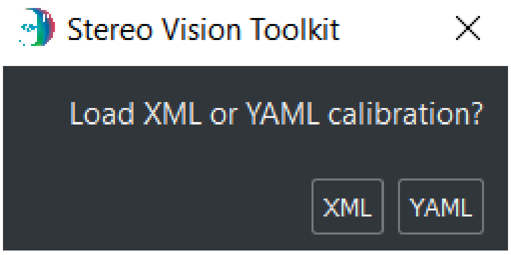
4.6.2 Load calibration files

Provided on your USB key should be calibration files. These will be in '.yaml' or '.xml' format. For increased performance it is advised to copy these from the USB key somewhere on your machine. Make a note of this location for loading within the Stereo Vision Toolkit.

To load a calibration file to enable stereo matching abilities connect a camera to the Stereo Vision Toolkit as described in **Connecting to camera (4.2)**. Once connected the 'Load Calibration' button will be available. Click this and select the folder that holds the calibration files.



A message box will ask if the calibration files being loaded are in XML or YAML format. Select which you have using the buttons (XML or YAML). It is likely you have been given both, in which case select YAML as these loads faster.



Once the calibration files have been loaded this will automatically start rectifying the images. You will see this as a slight skew to the images as the edges. This calibration will need to be re-loaded using the 'Load Calibration' button whenever you disconnect and re-connect from the camera.

4.6.3 Re-calibrate camera

Your camera is provided with a (tested) factory calibration which should remain stable in transit. However, you may wish to perform your own calibration and sooner or later you will probably need to. If you adjust the focus of the lenses yourself, or if for some reason you find that the images are failing to match (even with the projector on) then you can try a re-calibration.

First, print off a calibration target: https://docs.opencv.org/2.4/_downloads/pattern.png

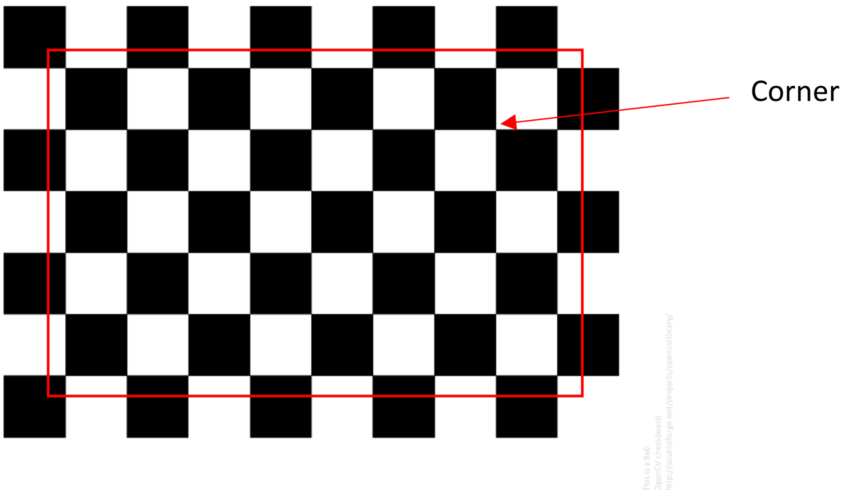


Figure 1 A standard 'chessboard' calibration target. We refer to the calibration points as corners, the points where the square vertices meet. The red box highlights the 'active' area of the pattern, which should be visible in all images.

You should print with as high a resolution as your printer will allow. An A4 target is the minimum recommended size, you may have better results with an A3 or even A2 target. Ensure that the aspect ratio of the print is fixed, if you have rectangles rather than squares, the calibration will be poor. If using a poster printer, be aware that glossy finishes usually suffer from specular reflections from room lighting.

Fix the printed calibration to a stiff board or surface, such as a sheet of wood or aluminium. The surface should be as close to planar as possible. Avoid taping the edges of the target, as over time paper tends to warp and the target will no longer be flat. A better solution is to glue the target to the board.

Measure the width of one of the calibration squares as accurately as you can – ideally using the software you printed the target with, if possible (print 1:1). The calibration algorithm attempts to locate 'corners' – points where the corners of the squares meet (see Figure above).

Next, specify a convenient save directory and capture some stereo image pairs of the target in different positions and orientations. It is important that you include images of the pattern tilted, and at various points and scales in the field of view. If your images are too similar, your calibration will be poor. This is important to accurately determine the distortion parameters for the lenses.

A typical selection of orientations might be:

1. Face on, close to camera
2. Face on, far from camera
3. Left image edge, tilted at various angles
4. Right image edge, tilted at various angles
5. Top image edge, tilted at various angles
6. Bottom image edge, tilted at various angles

Some example images (left only) are shown below:

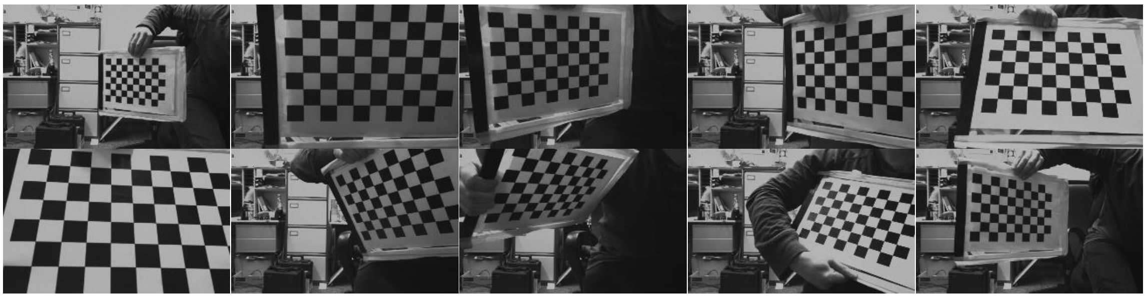
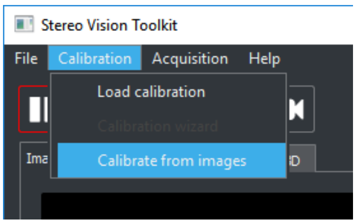


Figure 2 An example set of calibration images, using an A4 target printed on a board.

Capturing 10 calibration pairs should be enough for many cases. Make sure the pattern is fully visible in both cameras. Once you are done, open the calibrate from images tool:



Follow the instructions in the dialog window, providing the location of the left and right calibration images. The default image file mask should work. Make sure you set the correct pattern size (the image above is an 6 x 9) and the correct square size in millimetres.

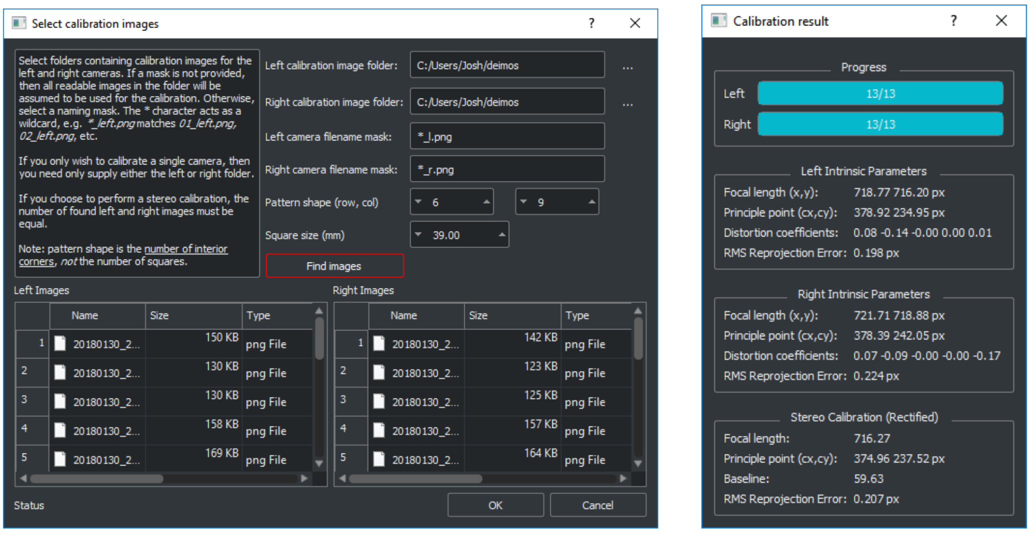


Figure 3 Left: the calibrate-from-images tool. Right: the calibration result using the calibration images in Figure 1. Note the good stereo RMS reprojection error of 0.2 px.

Click “Find images” and ensure that the images you just captured appear in the table. There should be an equal number of left and right images.

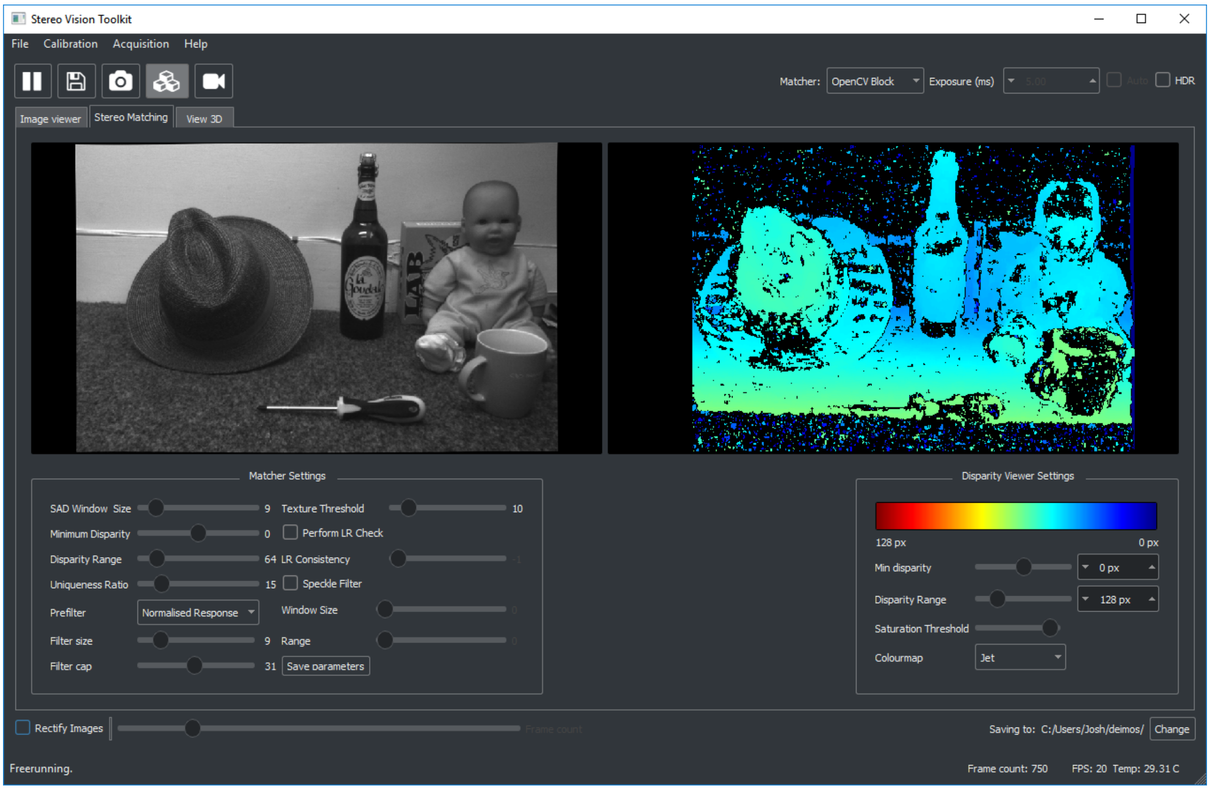
Click OK to perform the calibration. The calibration will take a few seconds to complete, depending on how many images you used. Once done, you will see a dialog with the result. A good calibration error (RMS reprojection error) is less than 0.25 px.

Finally, load your calibration files by selecting the folder that the calibration was saved to (currently the application directory). You can move the files somewhere else if you wish to, e.g. for backup or if using multiple Deimos cameras. Check the calibration works by seeing if you get sensible stereo matching results.

4.7 Matching images

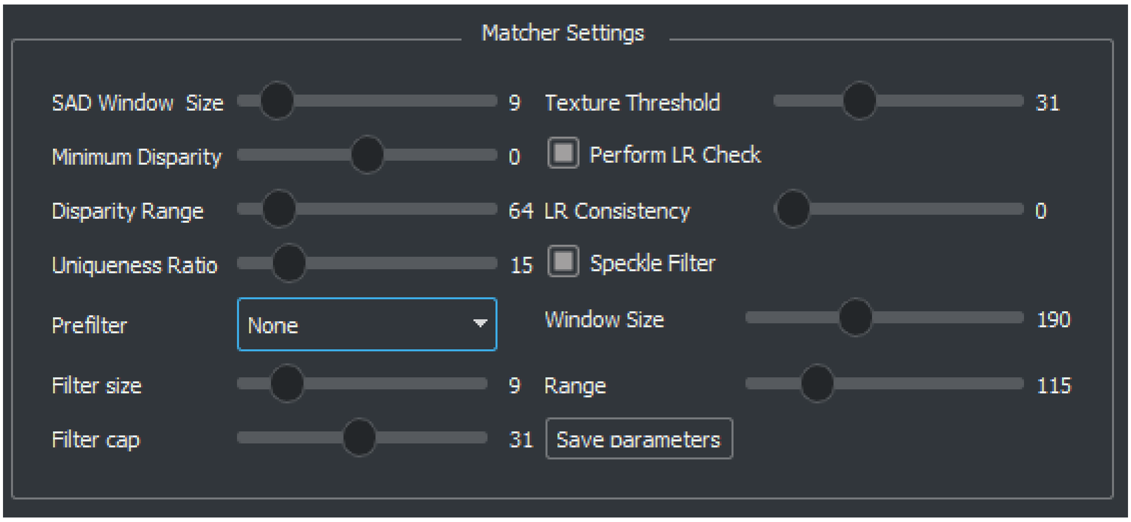
Once you have loaded in the correct calibration files for your camera, you can enable image rectification and start matching. Alternatively, if you are loading a video that has already been rectified, then you can enable matching regardless.

The matching window has three sections: the view from the left camera, the disparity map, and the parameters for the selected stereo matcher. Parameter changes will update in real time, allowing you to experiment to find out what works best for your scene.



Along with I3DRSGM, two open source matchers are provided currently, both from OpenCV. The first is a block matcher which local area matching[2]. This is a fast and simple method which can give good results, particularly if using the projector. However, it can struggle with repetitive objects. The second algorithm is the OpenCV implementation of Semi-Global Matching which is a more sophisticated algorithm and runs quite a lot slower than block matching.

[2] It compares a small patch in the left image and attempts to locate its match in the right image using correlation.



There are several common parameters (further description may be found in OpenCV's documentation [here](#))

Window size – sets the size of the comparison region. Bigger values will produce denser disparity maps at the expense of fine detail and edge sharpness.

Minimum disparity – set the minimum disparity to search from.

Disparity range – set the number of disparities to search over.

Uniqueness ratio – sets how confident the algorithm must be before a match is allowed. Set this to a high value to improve robustness at the expense of disparity map density.

LR Check – enable an approximate left-right consistency check, which can reduce noise in the disparity map.

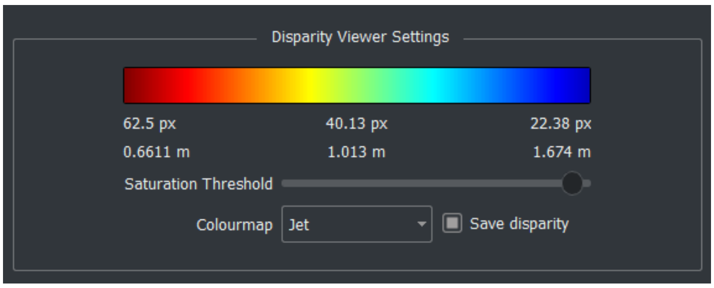
Texture threshold – minimum image texture to allow matching, can avoid spurious matches in featureless regions.

Speckle filter – a post-processing step to remove small outliers in the disparity map. Generally setting both range and window size to be high works well.

Block matching also has the option to pre-filter the image which can improve noise in the disparity map.

When you are happy with the parameters you have chosen, you can opt to save them as defaults for the next time you run the application.

Below the disparity viewer, the view settings will show what the colours represent. The image below shows how a lower disparity means a higher distance in meters. This can be useful to quickly estimate object distances. Saturation filtering can be useful if very bright lights are observed by the cameras.



If this is not visible you may need to hide the camera settings toolbar. This can be done by clicking 'Hide settings' in the bottom left.

The disparity viewer will show 3D represented as a coloured disparity map.

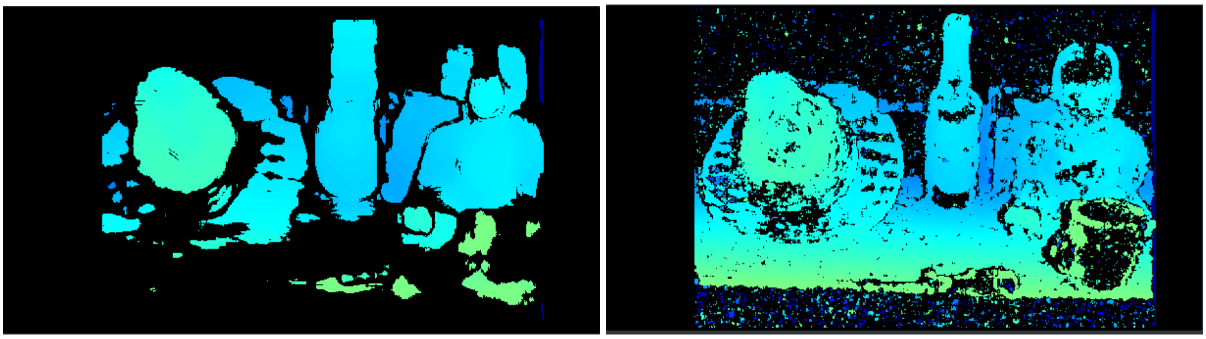


Figure 4 Left: disparity map with a SAD window size of 30 px, Right: disparity map with a SAD window size of 9 px

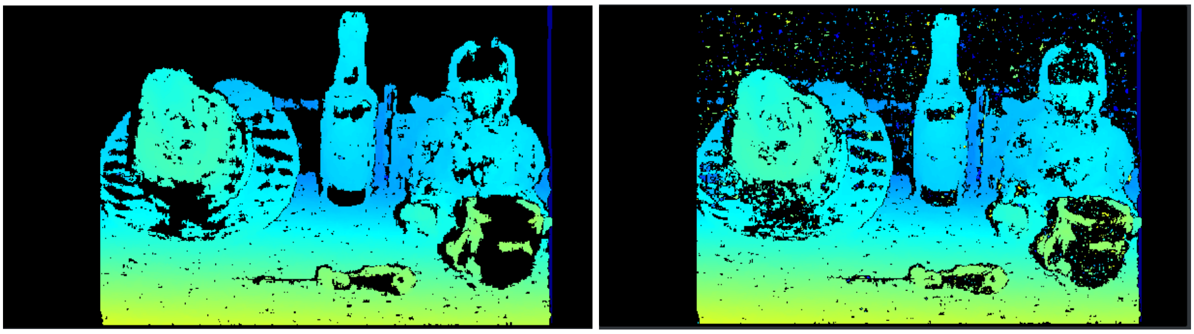


Figure 5 Left: Speckle filtering enabled, Right: no speckle filtering

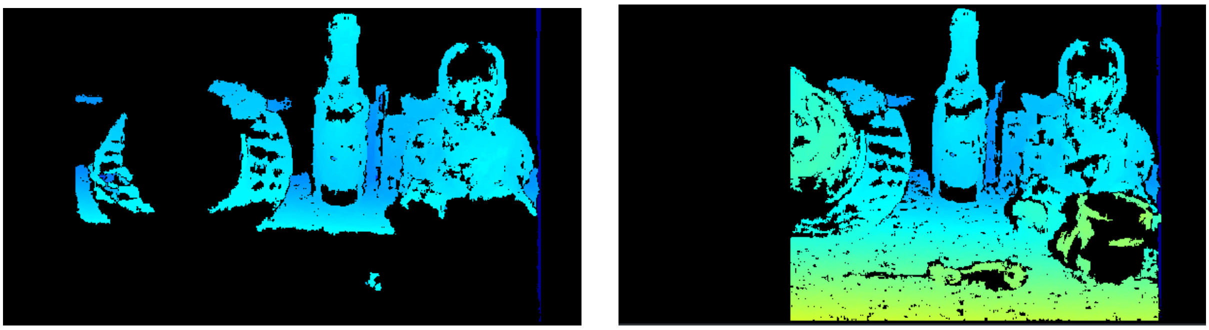


Figure 6 Left: A small disparity range of 32 px does not match the whole image. Right: A higher range of 192 px matches more of the image closer to the camera, but also reduces the region of the image that can be matched.

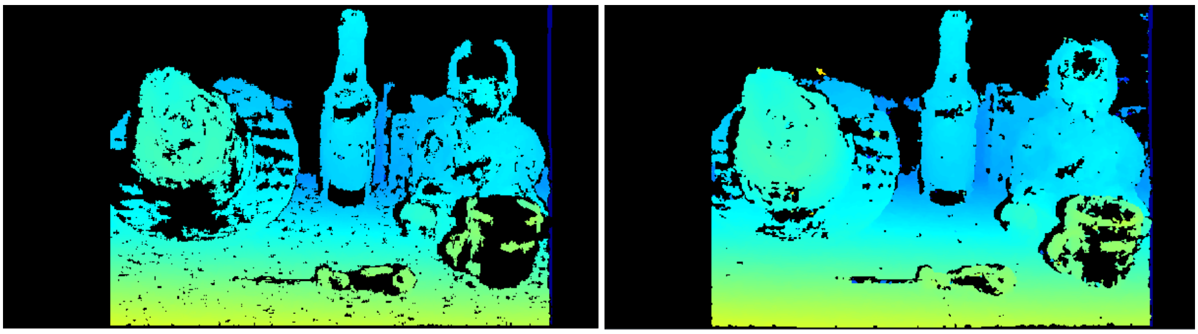
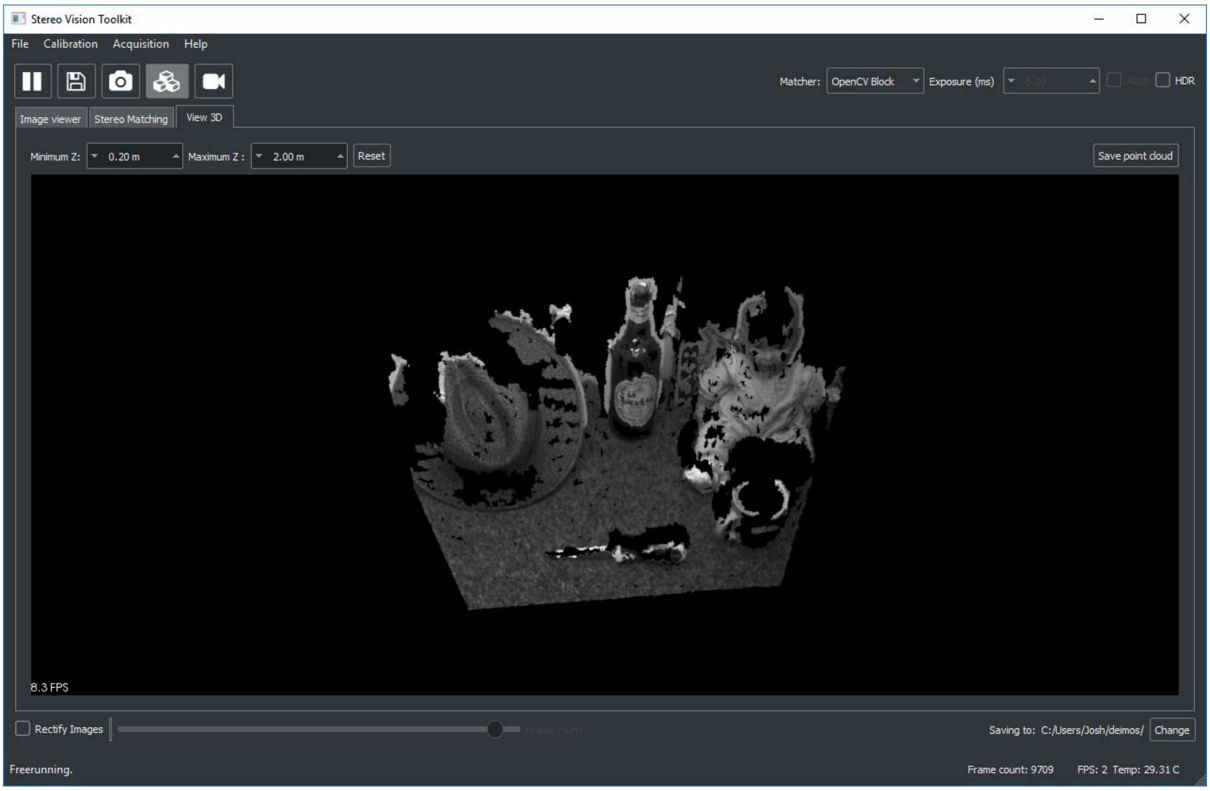


Figure 7 Left: OpenCV Block Matcher Right: OpenCV SGB Matcher

For most applications you will probably need to tweak the matcher settings to get the best reconstruction for a specific scene and depending on the desired trade-off between point cloud sparsity and robustness.

4.8 Viewing point clouds

If the camera is properly calibrated, you can switch to the third tab to see the disparity map projected into 3D space. The point cloud is displayed with the image intensity overlaid on top of it.



You can zoom with the mouse wheel and rotate with the left mouse button. Holding shift and clicking pans the scene.

Set the min/max Z distance to limit the range of points shown.

You can save the current point cloud as a PLY file for viewing in other software, such as the freely available Meshlab or CloudCompare.

Click the reset button to refresh the view. Occasionally the point cloud will not appear immediately, and you may need to shift the view slightly to re-render it.

5 Updating Software

The Stereo Vision Toolkit will automatically check on start-up if a new version of the software is available. When an update is available a window will appear in front of the application window informing you of the update. To download the update, click the link provided and close the current running application. This link will open in your default browser and start download the new installer. Once complete run the installer to update the software. Installation steps are the same as listed in **Installation process (2.4)**

6 Building from source

As the Stereo Vision Toolkit is currently Windows only this section will assume a Windows x64 PC.

Required software:

- Git (Version Control)
- QT Creator (Tested with 4.12.3 and 4.5.1)
- Visual Studio (Tested with 2015 and 2017)
- [Optional] CMake

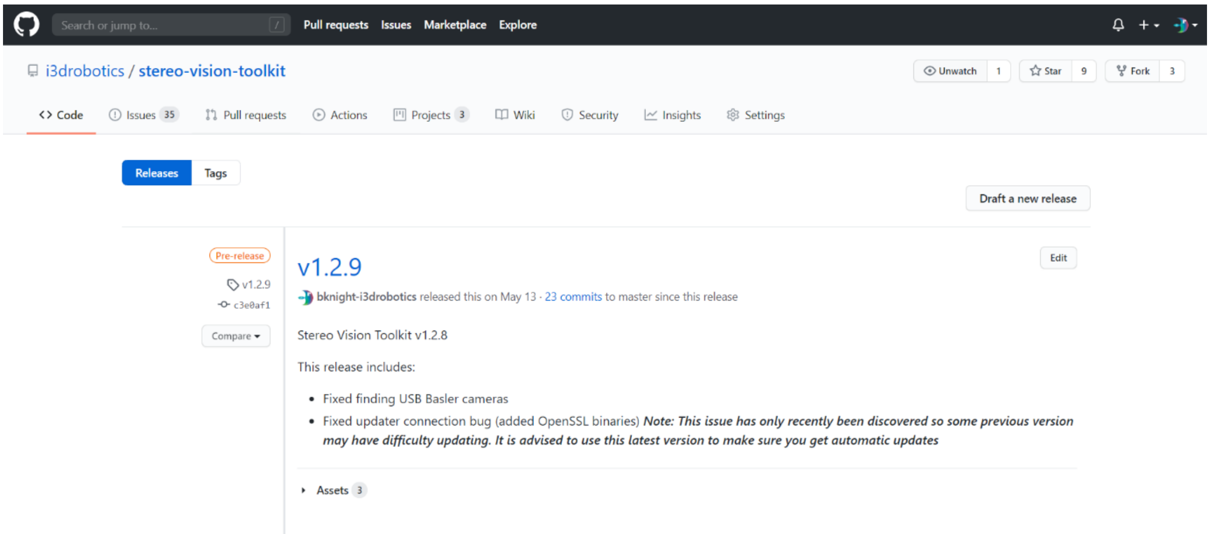
This guide will assume these are installed and setup on the machine building the software.

If you would like to use CMake rather than QT Creator to build, then a CMake file is provided however is not robustly tested and updated regularly so it is advised to use QT Creator.

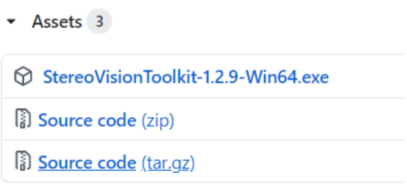
6.1 Source code

You can also find the latest release of the Stereo Vision Toolkit on the GitHub repository using the following link www.github.com/i3drobotics/stereo-vision-toolkit/releases.

You will be presented with the following page:



The latest release will be at the top of the page. Click 'Assets' to view the assets of this release, this includes a windows installer as well as the full source code for building the application for yourself.



To keep up to date with code changes it is best to clone this repository directly instead of downloading the source code zip, however, this source code zip is a stable release and the main repository could contain unstable changes.

If using git to clone the repository then open a terminal in the folder you would like the repository and enter the following command:

Git clone https://github.com/i3drobotics/stereo-vision-toolkit.git

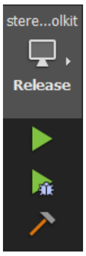
6.2 Dependencies

All 3rd party libraries required by the application are provided in the source code in the '3rdparty' folder. These are directly referenced in the qt 'stereo_vision_toolkit.pro' file. Should you want to use you own version of these libraries the qt 'stereo_vision_toolkit.pro' file would need manually updating. These libraries are built for Windows x64.

6.3 Build

This application can be built in QT Creator or using CMake. This option is provided so that installing QT Creator is not required. The QT libraries for building using CMake are provided in the repository.

6.3.1 QT Build



To build the software open the 'stereo_vision_toolkit.pro' file from the repository in QT Creator. Then select build (the hammer icon). Debug and Release dlls are configured for most libraries to enable full debugging of the application. Change to debug build by changing build configuration (computer screen icon).

Once build is complete the software can be run from the QT Creator by pressing the play button or running the exe directly from the build folder.

6.3.2 CMake Build

If you would like to use CMake rather than QT Creator to build, then a CMake file is provided however is not robustly tested and updated regularly so it is advised to use QT Creator.

Open the repository in the terminal and run the following command:

Cmake --build /build -G "Visual Studio 15 2017"

Note: Building QT Applications in CMake has a long compile time.

6.4 Building installer

Inno Setup is used to generate a windows installer from the build/release folder after the software has been built. Inno Setup can be download from the following link: https://jrsoftware.org/isinfo.php.

Once installed the Inno Setup script can be run to compile the installer by right clicking the 'installer/installer.iss' file in windows explorer and selecting 'compile'. This will build to installer in the folder 'installer/output/' with the name 'StereoVisionToolkit-{Version}-

Win64.exe'. This install is the file that is provided in the Releases section of the GitHub and provided to the user via USB key.

6.5 Building documentation

Code documentation is provided by using Doxygen. This uses code commenting to automatically generate class, function, and variable documentation. To re-generate documentation, run the following command:

Doxygen Doxyfile

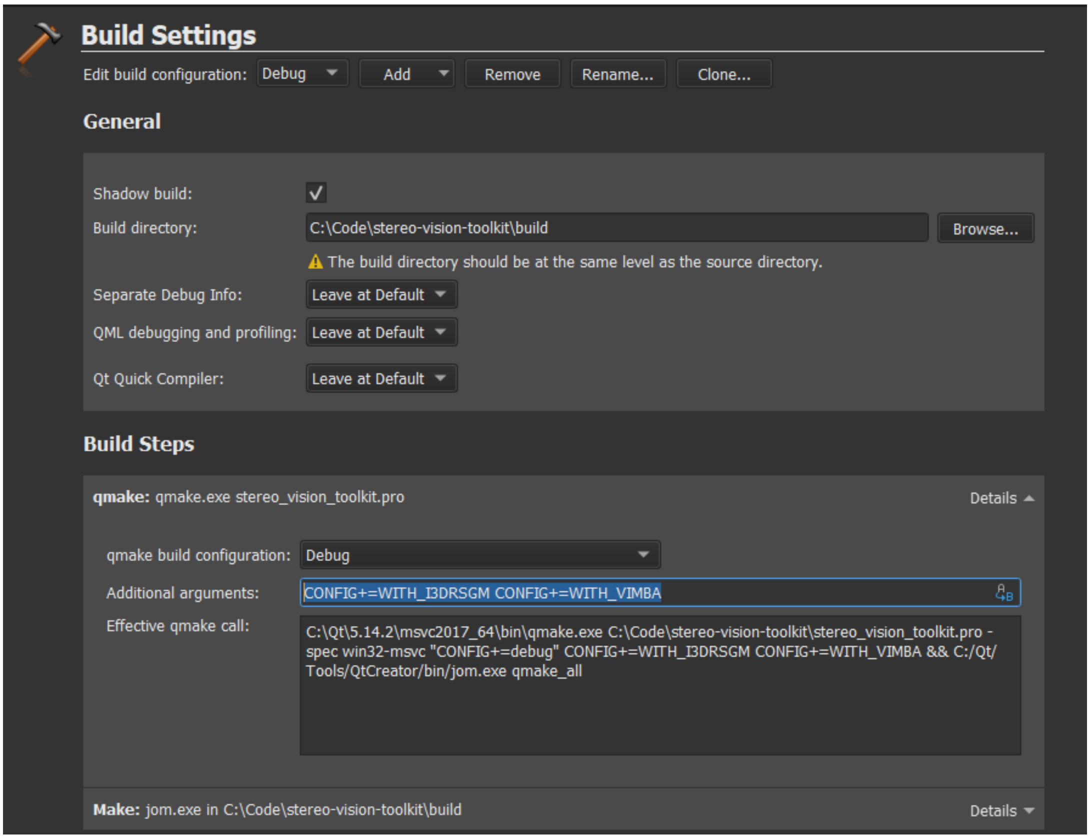
Auto generating of documentation is disabled to avoid large updates to the code on every build so documentation updates must be triggered manually.

6.5.1 Build options

For open source building of the repository no build options are required.

QT Build options can be access in the 'Build Settings' of QT Creator under the section

'Build Steps'->'Additional arguments'



The following options are available:

Command	Description
WITH_I3DRSGM	Only required for using I3DR's stereo matching algorithm in the software. Only available to I3DR developers as the required I3DRSGM private repository used is not publicly available to protect I3DR IP. Requires CUDA enabled GPU.
WITH_VIMBA	Only required while vimba cameras are being integrated. This should be omitted unless you are a developer.
WITH_CUDA	Optional if you have a CUDA enabled GPU. This will include CUDA OpenCV library files in the application.

7 Troubleshooting

7.1 Framerate is slower than expected

Check that you are not performing any graphics-heavy processing on the same machine. The stereo vision toolkit automatically performs rectification using your GPU (if available). Similarly, if you are doing any heavy computation work, this may slow down stereo matching.

7.2 I only get 30fps, not 60fps

This is typically because the camera is operating in USB2 mode. Please check that you are using a USB3 port, and that it has sufficient bandwidth. Alternatively try using a different USB cable.

7.3 Image matching is poor

It is possible you need to recalibrate the system. First, try to image a target that is amenable to matching, with lots of features. A blank wall with the projector turned on is a good option. If you still do not get good match results, then perform a recalibration and try again. Check that the rectification appears to be sensible – straight lines in the world should be straight in the images.

7.4 USB connector is not secure

This is a common problem for USB3 Micro-B connectors over time, particularly if they are removed and re-inserted frequently. We provide the Deimos system with tapped holes, usable with standard USB3 vision cables (with screw locks). You may find that another USB3 cable provides better latching.

7.5 I am having problems with the software

Please visit the GitHub repository and post an issue (preferred), or contact us for assistance.

7.6 I am getting high calibration errors (> 0.3 px)

- Make sure that you are in a brightly lit area, and lower the exposure time, to prevent motion blur
- Use HDR mode
- Capture more calibration images, making sure that the board is present at different distances and with a large variety of orientations relative to the sensor
- Check that the pattern size you provided is correct

7.7 I cannot see any 3D output

Sometimes you may need to move the view slightly (using your mouse) to re-render the point cloud. You need a valid calibration to project to 3D, as the camera projection matrix must be known.

7.8 Where can I find out more about stereo imaging?

The canonical reference textbook for stereo imaging geometry is Hartley and Zisserman, *Multiple View Geometry in Computer Vision* (OUP), however there is little information on matching. Unfortunately, there are few good books on stereo, in terms of a good algorithm comparison, so it's usually better to look at the relevant papers.

You can look at Scharstein and Szeliski's stereo taxonomy paper (https://doi.org/10.1109/SMBV.2001.988771) or the OpenCV calib3d documentation (https://docs.opencv.org/2.4/modules/calib3d/doc/calib3d.html). For information on the calibration routine used, take a look at Zhang's paper (https://www.microsoft.com/en-us/research/publication/a-flexible-new-technique-for-camera-calibration/).

You can also look at the Middlebury and KITTI stereo benchmarks for a more comprehensive algorithm comparison, though note that the test data has existed for a while and there is likely to be some bias towards matchers which work well on that data. For example, the top KITTI performer is a deep neural net trained on driving imagery, so it is not clear how well it would transfer to other domains.