

# CVVisual

Ein Debug-Framework für OpenCV

Andreas Clara Erich Florian Johannes Nikolai  
Raphael

20. Juni 2014

# Gliederung

- Einführung in OpenCV
- Motivation
- Anwenderfeatures
- Gui-Demo
- Dokumentation
- Architektur
- API
- Ausblick

# Einführung in OpenCV

# Überblick

- Bildverarbeitung
- weite Verbreitung
- Matrizen als Grundlage
- Filter + Matches (und vieles mehr!)

# Matrizen

- Bild = mehrdimensionale Matrix
- 3. Dimension = Channel

# Filter

- 2D-Bilder
- Berechnung auf Umgebung jedes Pixels
- Bsp: dilate, erode, Sobel

# Matches

- FeatureDetector → Keypoints = charakteristische Punkte
- Match = Paar aus Keypoints

# Motivation



# Debuggen von OpenCV

Systematisches Debugging statt „Random Code“

# Ziele

- Visualisierung von Matrizen, Filtereffekten und Matches

## Anwenderfeatures

## Verwendung

```
std::string imgIdString{"imgRead"};
imgIDString += toString(imgId);
cvv::showImage(imgRead, CVVISUAL_LOCATION, imgIdString);

// convert to grayscale:
cv::Mat imgGray;
cv::cvtColor(imgRead, imgGray, CV_BGR2GRAY);
cvv::debugFilter(imgRead, imgGray, CVVISUAL_LOCATION,
                 "to gray", "SingleFilterView");
```

# Übersicht

Übersicht über alle Aufrufe

# Übersicht

Filterbar

# Übersicht

Sortierbar

# Übersicht

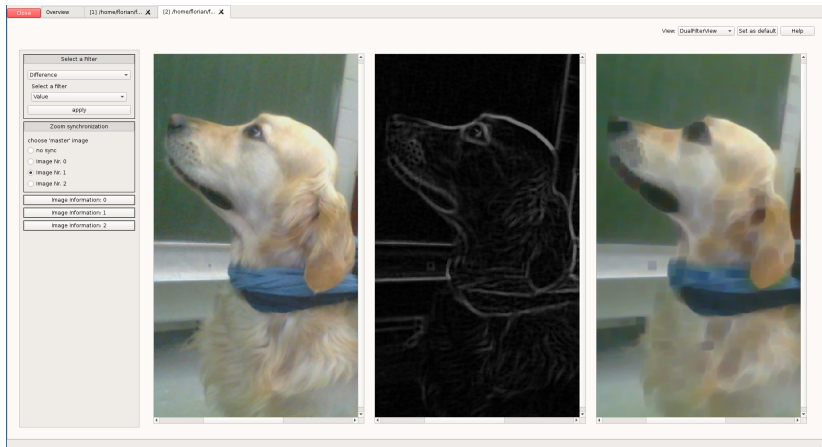
Gruppierbar



# Übersicht

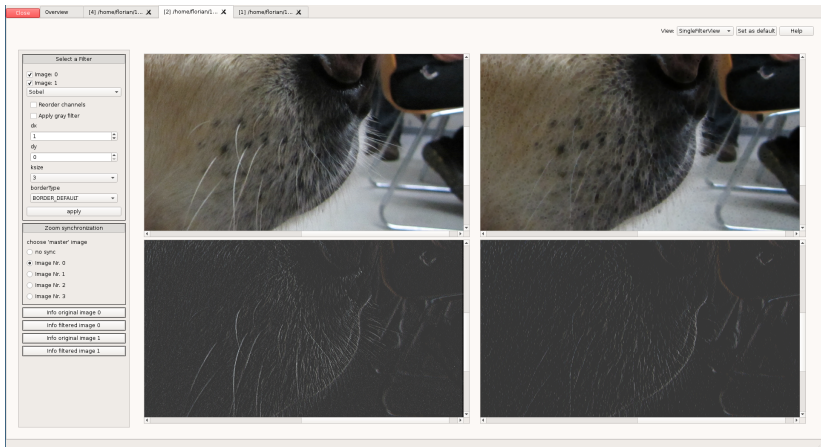
# Filter

- 2 Bilder  $\rightarrow$  1 Bild
- Differenzbilder, Overlay, geänderte Pixel für Filter



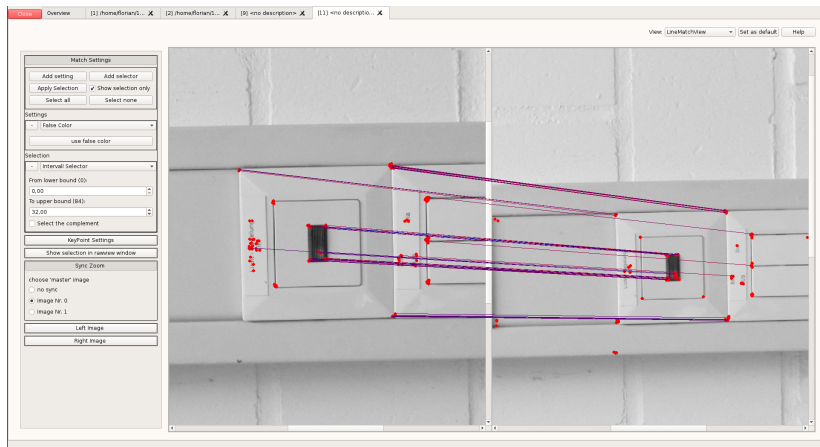
# Filter

- 1 Bilder  $\rightarrow$  1 Bild
- Nachträgliche Anwendung weiterer Filter



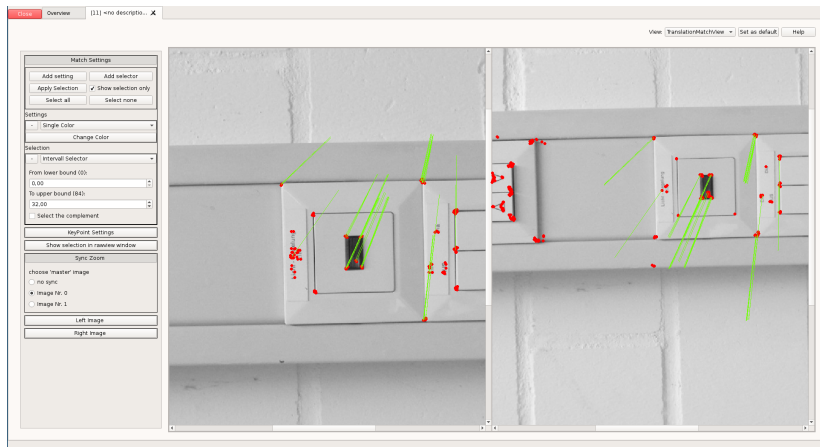
# Matches

- Anzeigen / Filtern von Keypoints / Matches
- Anzeige der Verbindungen von Keypoints



# Matches

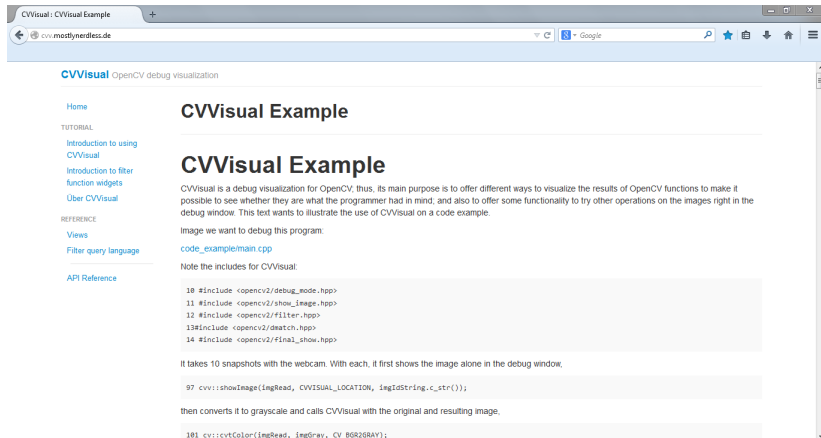
- Anzeigen / Filtern von Keypoints / Matches
- Anzeige der Translation von Keypoints



## GUI-Demo

# Dokumentation

# Tutorials, Beispiele



The screenshot shows a web browser window with the address bar displaying "cvv.mostlynerdless.de". The page title is "CVVisual : CVVisual Example". The main content area has the heading "CVVisual Example" and a sub-heading "CVVisual OpenCV debug visualization". On the left, there is a navigation menu with links: "Home", "TUTORIAL" (with sub-links "Introduction to using CVVisual", "Introduction to filter function widgets", and "Über CVVisual"), "REFERENCE" (with sub-links "Views" and "Filter query language"), and "API Reference". The main text explains that CVVisual is a debug visualization for OpenCV, designed to help programmers visualize the results of OpenCV functions. It provides a code example for "code\_example/main.cpp" and shows the includes for CVVisual:

```
10 #include <opencv2/debug_mode.hpp>
11 #include <opencv2/show_image.hpp>
12 #include <opencv2/filter.hpp>
13 #include <opencv2/dmatch.hpp>
14 #include <opencv2/finai_show.hpp>
```

It then states that it takes 10 snapshots with the webcam and shows the image alone in the debug window. The code snippet for showing the image is:

```
97 cvv::showImage(imgRead, CVVISUAL_LOCATION, imgIdString.c_str());
```

Finally, it mentions that the image is converted to grayscale and calls CVVisual with the original and resulting image. The code snippet for this is:

```
101 cv::cvtColor(imgRead, imgGray, CV_BGR2GRAY);
```



# Kurzdokumentation

Wird von der Hilfefunktion des Programms benutzt.

**CVVisual** OpenCV debug visualization

[Home](#)

TUTORIAL

[Introduction to using  
CVVisual](#)

[Introduction to filter  
function widgets](#)

[Über CVVisual](#)

REFERENCE

[Views](#)

[Filter query language](#)

[API Reference](#)

## Views

### General information:

Most views offer an `ImageInformation` collapsable in their accordion menus.

The zoom can be found here.

`Ctrl+Mouse wheel` is also zoom; `Ctrl+Shift+Mouse wheel` is a slower zoom.

If the zoom is deeper than 60%, the image's pixels will be overlaid with their channel values; usually, the order is BGR[+alpha] from the top.

### Single Image View:

Associated with the `debugSingleImage()` function.

Shows one single image with no features other than `Image Information`.

### Filter Views:

Associated with the `debugFilter()` function.

### DefaultFilterView:

Shows two images with only the basic features of `ImageInformation`, synchronized zoom and `Histogram`.

### DualFilterView:

Shows the two images given to the CVVisual function and `Result Image` inbetween which represents the result of a filter that was applied to the others via the `Filter selection` collapsable, like a difference image between the two.

# Referenz:

- Mit Hilfe von Doxygen

The screenshot shows a web browser window displaying the CVVisual application. The address bar shows the URL: `cvv.mostlynerdless.de/api/classcvv_1_1qtutil_1_1Accordion.html`. The page title is "CVVisual" with the subtitle "A debug visualization for opencv".

The application interface includes a navigation menu on the left with tabs for "Main Page", "Namespaces", "Classes", and "Files". Under the "Classes" tab, a tree view shows the class hierarchy: `Namespaces` > `Classes` > `Class List` > `OW` > `controller` > `extend` > `gui` > `impl` > `qtutil` > `structures` > **`Accordion`**. The breadcrumb trail at the bottom shows: `cvv` > `qtutil` > `Accordion`.

The main content area is titled "Public Member Functions" and lists the following functions:

- Accordion** (QWidget \*parent=nullptr)  
Constructs an empty accordion. More...
- void clear ()**  
Removes all elements and deletes them immediately. More...
- void collapse (Handle handle, bool b=true)**  
Collapses an element. More...
- void collapseAll (bool b=true)**  
Collapses all elements. More...
- Collapsible & element (Handle handle)**  
Returns the element corresponding to handle. More...
- const Collapsible & element (Handle handle) const**
- void expand (Handle handle, bool b=true)**  
Expands an element. More...
- void expandAll (bool b=true)**  
Expands all elements. More...
- void hide (Handle handle, bool b=true)**  
Makes the element invisible. More...

The footer of the page indicates it was "Generated on Tue Mar 25 2014 22:45:17 for CVVisual by **doxygen** 1.8.6".

# Architektur

# Entwurf

- Trennung in API, Datenhaltung, Visualisierung

## Signals/Slots & Templates

- Qt erlaubt keine Templateklassen mit Signals/Slots
- Signals/Slots in Objekte ausgelagert

# RegisterHelper

- Bietet Funktionalität zum Anmelden neuer Funktionen
- Auswahl erfolgt über eine Combobox
- Beim Anmelden wird ein Auswahl Name angegeben

# AutoFilterWidget

- ist Unterklasse von RegisterHelper
- Erlaubt Auswahl und Anwendung von Filtern
- Gibt Ergebnisse der Filter per Signal weiter (z.B. an ein ZoomableImage)

# ZoomableImage

- Eigentständige Umwandlung von `cv::Mat` in Qt Format
- Signal & Slot für Zoom Events
- Slot zum Bild wechseln
- `ZoomableImageOptionPanel` zeigt weiter Informationen/Optionen an



# MatchScene

- Enthält 2 ZoomableImages
- Enthält die KeyPoints/Matches als QGraphicsObjects
- Hat Probleme mit der Mausinteraktion von der Matches

# Match/KeyPointSetting

- Keine Auslagerung von Singals/Slots möglich
- Daher parallele Entwicklung von KeyPoint und MatchSetting
- Nur Selektierte KeyPoints/Matches werden angezeigt

# Views

- Visualisierung der unterschiedlichen Aufrufe
- Unterscheiden sich meist in unterschiedlichen Nutzen von QT Util Klassen
- Einzige Aufgabe Weiterleitung und Annahme der Selektion (beim Wechsel der Views)

# API

# Anwender API

- Triviale Benutzung auch in C++98
- Sehr klein und übersichtlich

# Interne API

- Erweiterung über Funktionen in `cvv::extend`
- Leichtes, zentralisiertes Hinzufügen von Visualisierungen, Filtern, Views,...

# Ausblick

# Rezeption

Projekt schien von der OpenCV-Community wohlwollend aufgenommen zu werden



snosov1 commented 2 days ago

Collaborator

Hi, Andreas!

First of all, thank you for a really valuable contribution. I've been dreaming about such functionality since the day 1 I started using OpenCV.

As @apavlenko suggests, this module should probably go to the opencv\_contrib repository. Due to limited resources we've created it, so we could easily accept such big PRs - almost "No questions asked". Then it boils there for a bit of time, and if it turns out to be solid and well received by the community, we would merge it into the mainstream (this) repo.

It's a default path for such major contributions and if you're ok with it - let's do it this way.

Personally, I would like such module to be in the mainstream repo as soon as possible. So, I'll try to review it shortly and give some feedback.



# Rezeption

Nach aktuellem Stand aber aufgrund C++11 und Qt5 keine Aufnahme ins Haupt-Repo



snosov1 commented on 19. Apr.

Sorry for delay. I've looked through it right away, and they're a couple of issues. Mainly, we don't plan to enable C++11 for builds of this repository, since the support is not yet ubiquitous. Also, the usage of Qt5 is rather limiting.

This makes it a great tool for development and research on Desktops with latest sw, but is unusable on other platforms.

My thinking is that in its current form it doesn't belong to the mainstream repo because of these dependencies. But, I think, it can be merged to the contrib repo after a few minor fixes.

Let's also ask [@kirill-kornyakov](#) on that.

## Links

- Github: <https://github.com/CVVisualPSETeam/CVVisual>
- Dokumentation: <https://cvv.mostlynerdless.de/>
- Doxygen: <https://cvv.mostlynerdless.de/api/>