

CVVisual

Andreas Clara Erich Florian Johannes Nikolai
Raphael

20. Juni 2014

Inhaltsverzeichnis

0.1	Gliederung	2
1	Einführung in OpenCV	2
1.1	Überblick	2
1.2	Matrizen	2
1.3	Filter	3
1.4	Matches	3
2	Motivation	3
2.1	Debuggen von OpenCV	3
2.2	Ziele	3
3	Anwenderfeatures	3
3.1	Verwendung	3
3.2	Übersicht	4
3.3	Übersicht	4
3.4	Übersicht	4
3.5	Übersicht	4
3.6	Übersicht	4
3.7	Filter	4
3.8	Filter	5
3.9	Matches	5
3.10	Matches	5
4	GUI-Demo	7
5	Dokumentation	7
5.1	Tutorials, Beispiele	7
5.2	Kurzdokumentation	7
5.3	Referenz:	7

6	Architektur	7
6.1	Entwurf	7
6.2	Signals/Slots & Templates	9
6.3	RegisterHelper	9
6.4	AutoFilterWidget	9
6.5	ZoomableImage	9
6.6	MatchScene	9
6.7	Match/KeyPointSetting	9
6.8	Views	9
7	API	10
7.1	Anwender API	10
7.2	Interne API	10
8	Ausblick	10
8.1	Rezeption	10
8.2	Rezeption	11
8.3	Links	11

0.1 Gliederung

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

1 Einführung in OpenCV

1.1 Überblick

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

1.2 Matrizen

- Bild = mehrdimensionale Matrix

- 3. Dimension = Channel

Bsp. BGR-Bild: 1. Channel blau, 2. Channel grün usw.

1.3 Filter

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

erode -> kleine Details weg

1.4 Matches

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

2 Motivation

2.1 Debuggen von OpenCV

Systematisches Debugging statt „Random Code“

Hinweis auf showMatches/showKeypoints

2.2 Ziele

- Visualisierung von Matritzen, Filtereffekten und Matches

3 Anwenderfeatures

3.1 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");
```

3.2 Übersicht

Übersicht über alle Aufrufe

3.3 Übersicht

Filterbar

3.4 Übersicht

Sortierbar

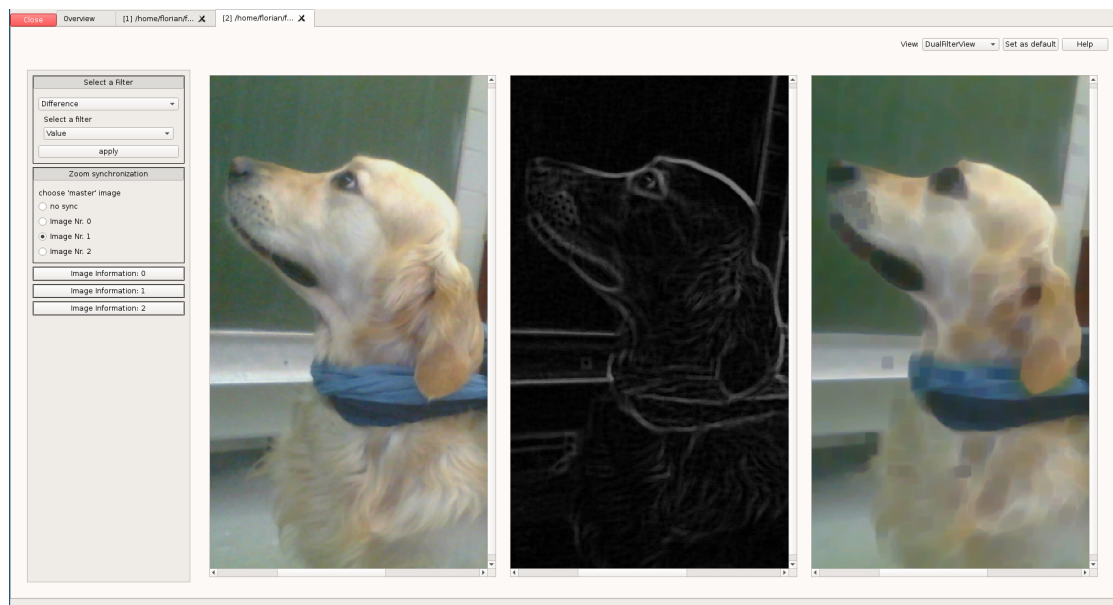
3.5 Übersicht

Gruppierbar

3.6 Übersicht

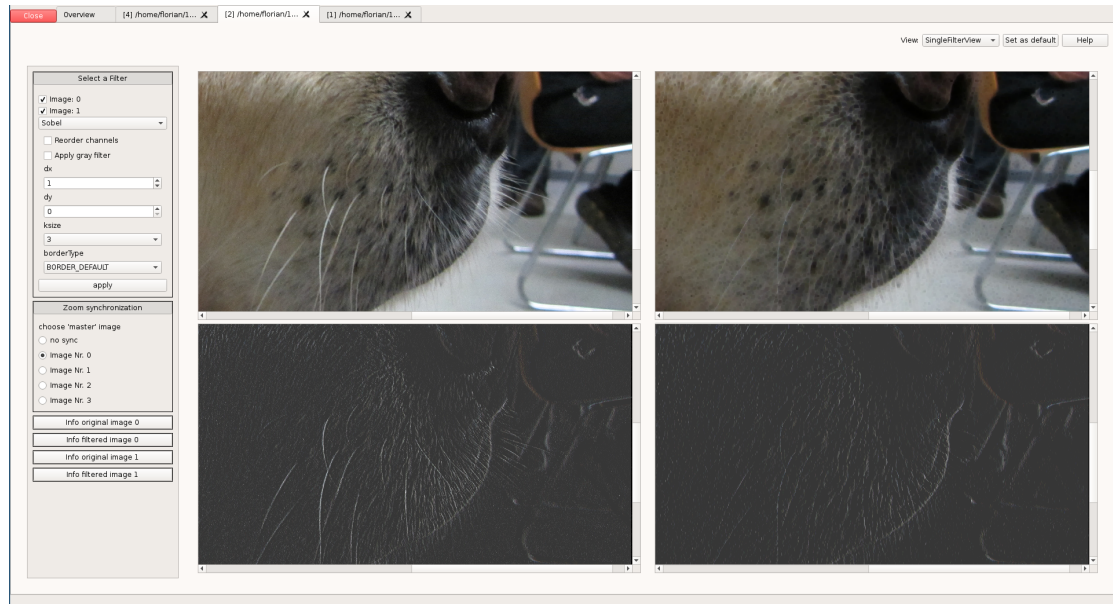
3.7 Filter

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



3.8 Filter

- 1 Bilder → 1 Bild
- Nachträgliche Anwendung weiterer Filter

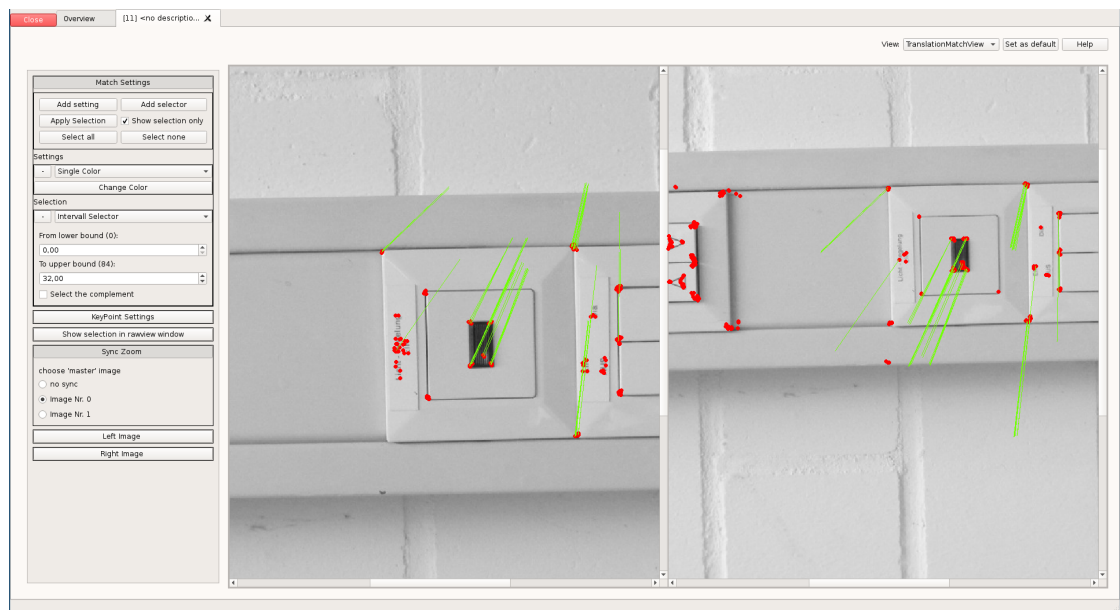
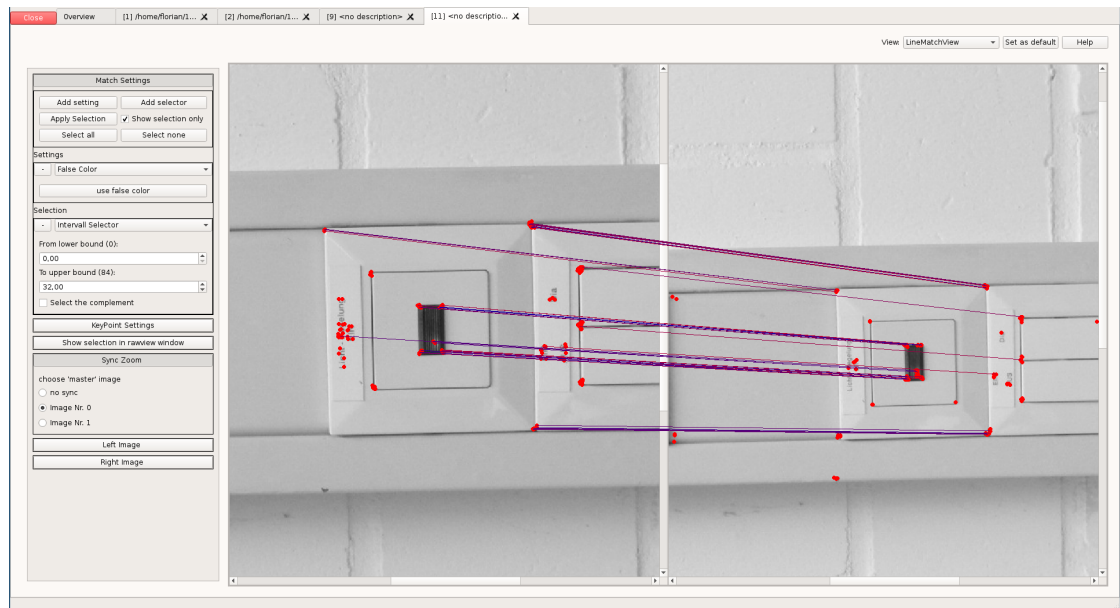


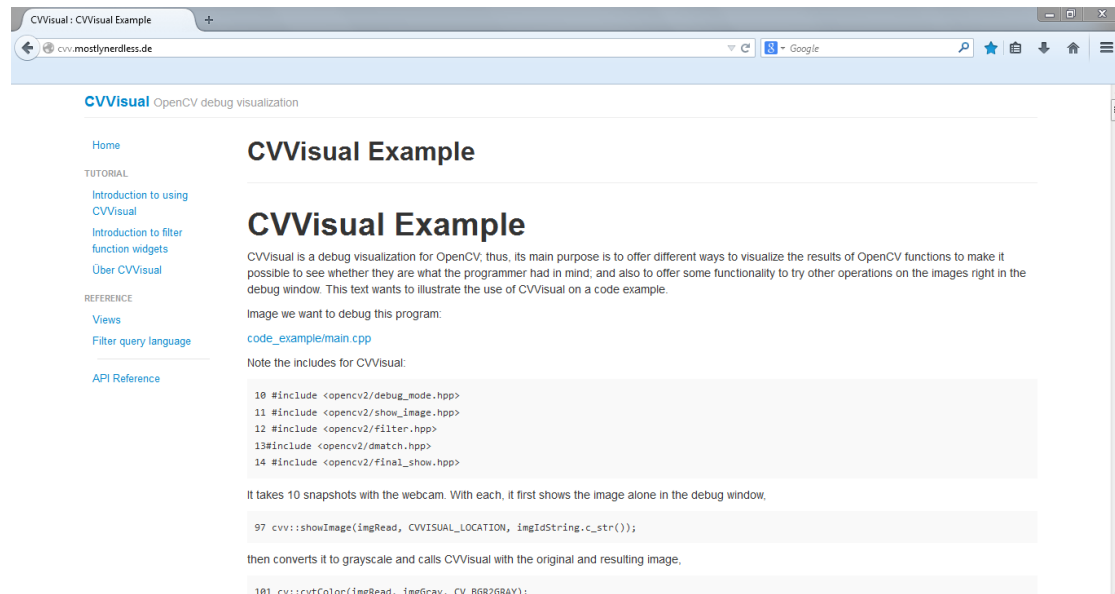
3.9 Matches

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

3.10 Matches

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





4 GUI-Demo

5 Dokumentation

5.1 Tutorials, Beispiele

5.2 Kurzdokumentation

Wird von der Hilfefunktion des Programms benutzt.

5.3 Referenz:

- Mit Hilfe von Doxygen

6 Architektur

6.1 Entwurf

- Trennung in API, Datenhaltung, Visualisierung
Altes Entwurfs Bild

[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.

The screenshot shows a web browser window with the address bar displaying `cvv.mostlynerdless.de/api/classcv_1_1qtutil_1_1Accordion.html`. The page title is "CVVisual" and the subtitle is "A debug visualization for opencv". The navigation bar includes tabs for "Main Page", "Namespaces", "Classes", and "Files". Below the navigation bar, there are sub-tabs for "Class List", "Class Index", "Class Hierarchy", and "Class Members". The left sidebar shows a tree view of the class hierarchy, with "cvv" expanded and "qtutil" selected. The main content area displays the "Public Member Functions" for the `Accordion` class. The functions listed are:

- `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`: Returns the element corresponding to handle. More...
- `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)`: Hides the element. 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.

6.2 Signals/Slots & Templates

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

6.3 RegisterHelper

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

6.4 AutoFilterWidget

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

6.5 ZoomableImage

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

6.6 MatchScene

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

6.7 Match/KeyPointSetting

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

6.8 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)

7 API

7.1 Anwender API

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

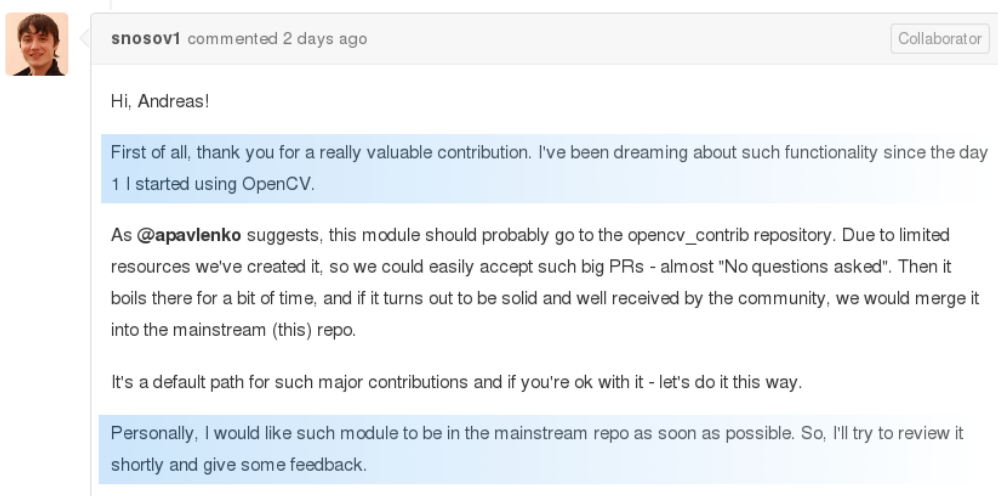
7.2 Interne API

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

8 Ausblick

8.1 Rezeption

Projekt schien von der OpenCV-Community wohlwollend aufgenommen zu werden



8.2 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.

8.3 Links

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