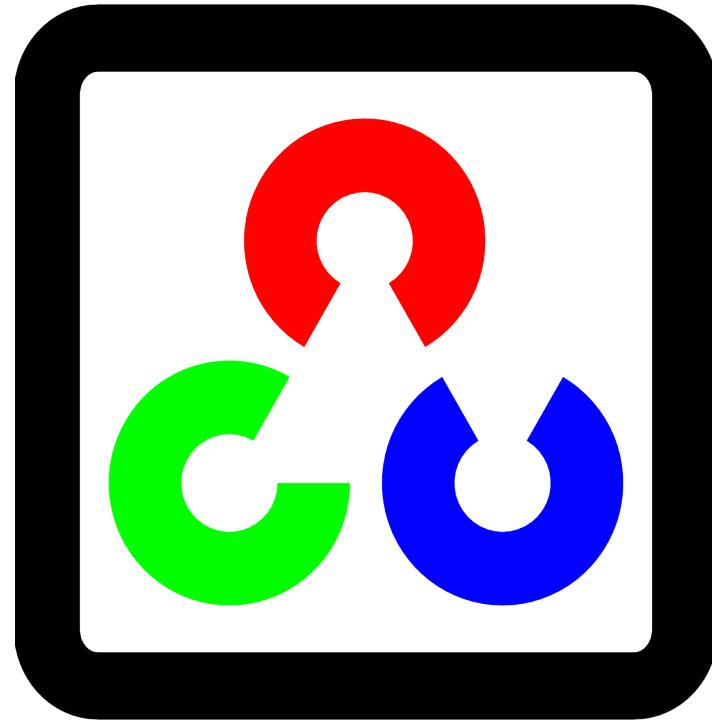


# CVVisual



Johannes Bechberger, Erich Bretnütz,  
Nikolai Gaßner, Raphael Grimm,  
Clara Scherer, Florian Weber

# Gliederung

- Motivation
- Entwurf
- Implementierung
- Qualitätssicherung
- Features
- Demonstration
- Ausblick

# Motivation

Systematisches Debugging statt „Random Code“

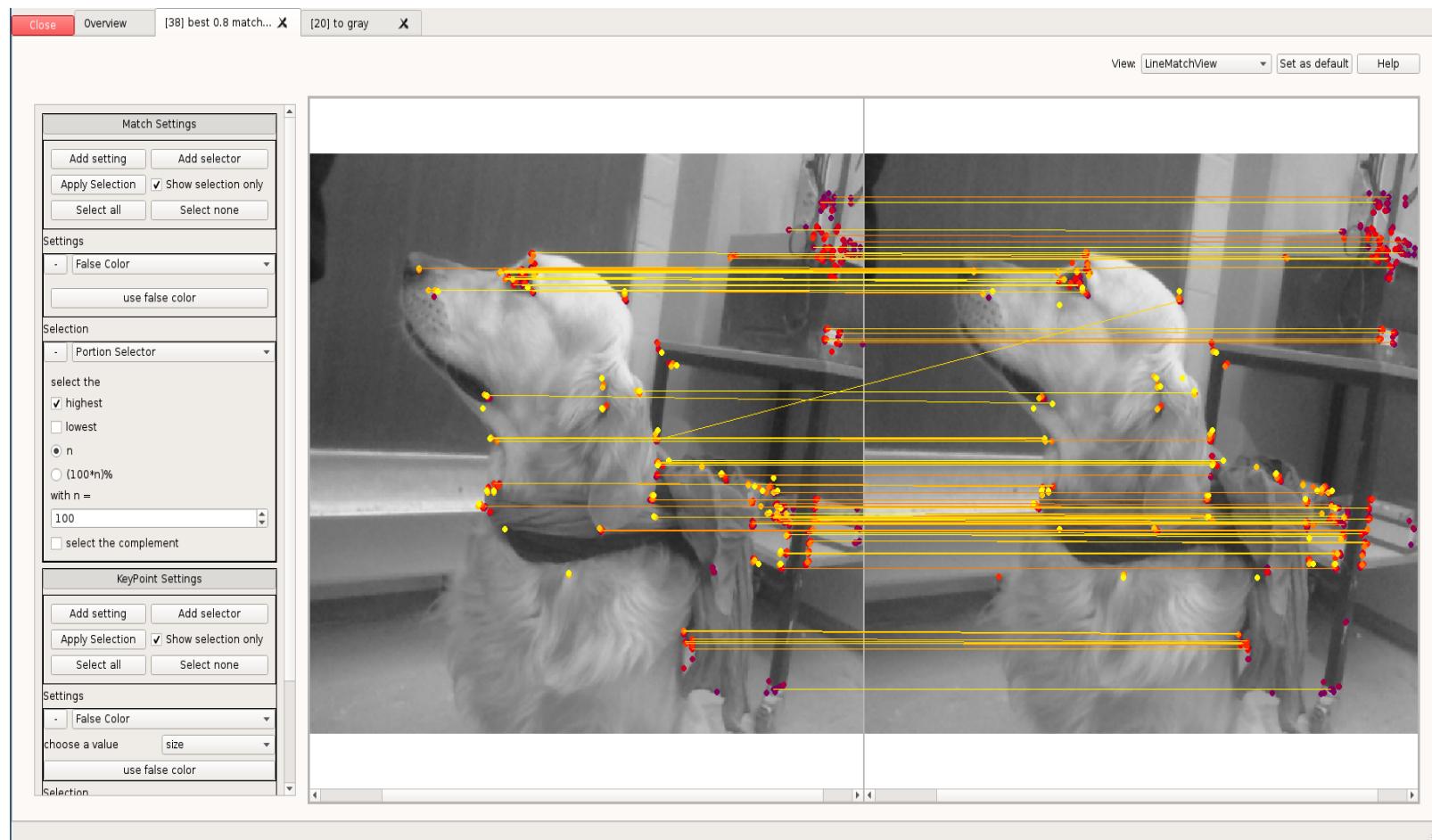
```
#ifdef DEBUG
    Mat img_matches;
    drawMatches( img_1, keypoints_1, img_2, keypoints_2,
                 good_matches, img_matches, Scalar::all(-1), Scalar::all(-1),
                 vector<char>(), DrawMatchesFlags::NOT_DRAW_SINGLE_POINTS );
    imshow("good matches", img_matches);
#endif
```

*versus*

```
cvv::debugMatches(img1, img2, keypoints_1, keypoints_2, good_matches);
```

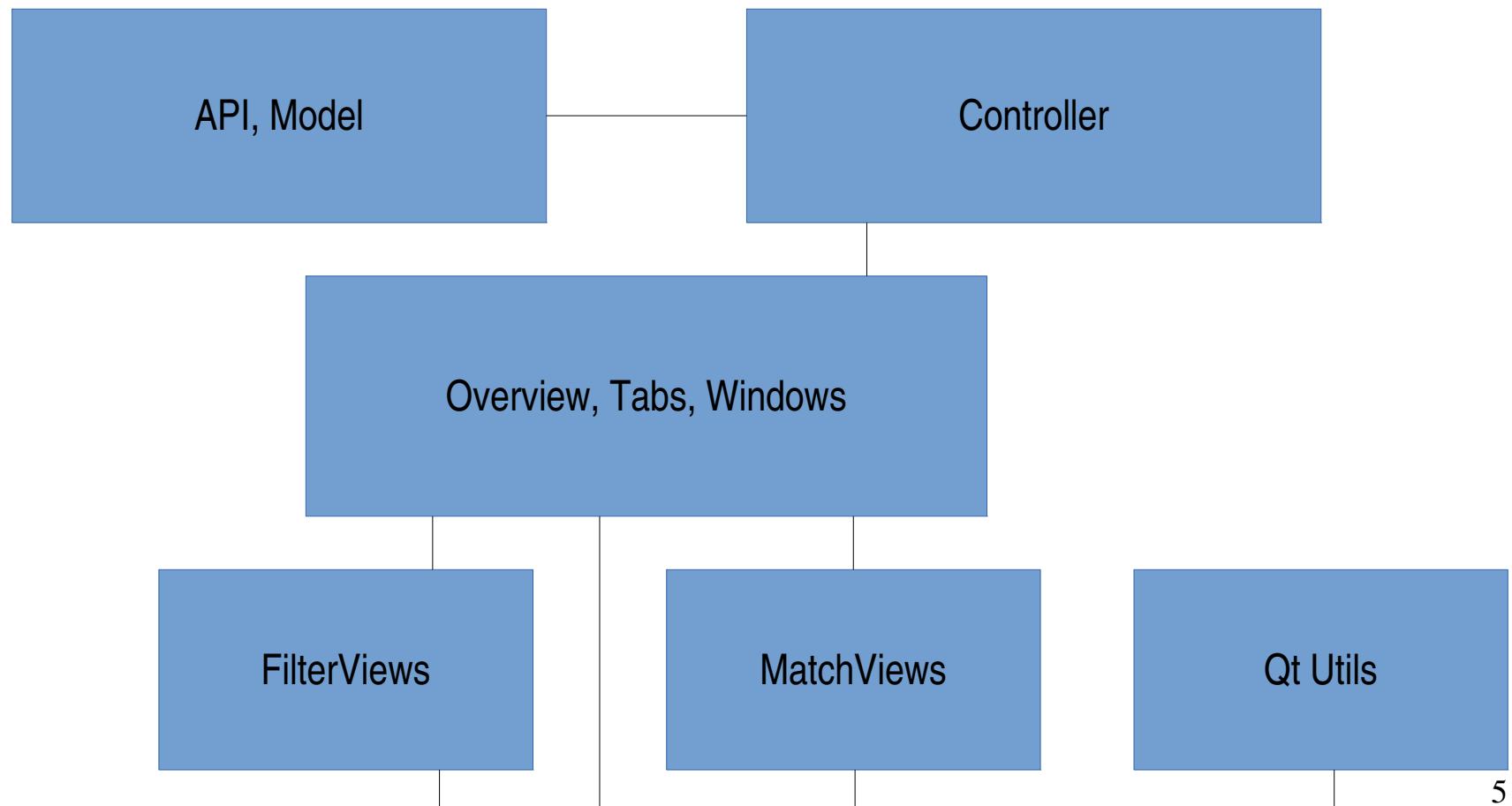
# Motivation

Visualisierung von Matrizen, Filtereffekten und Matches.



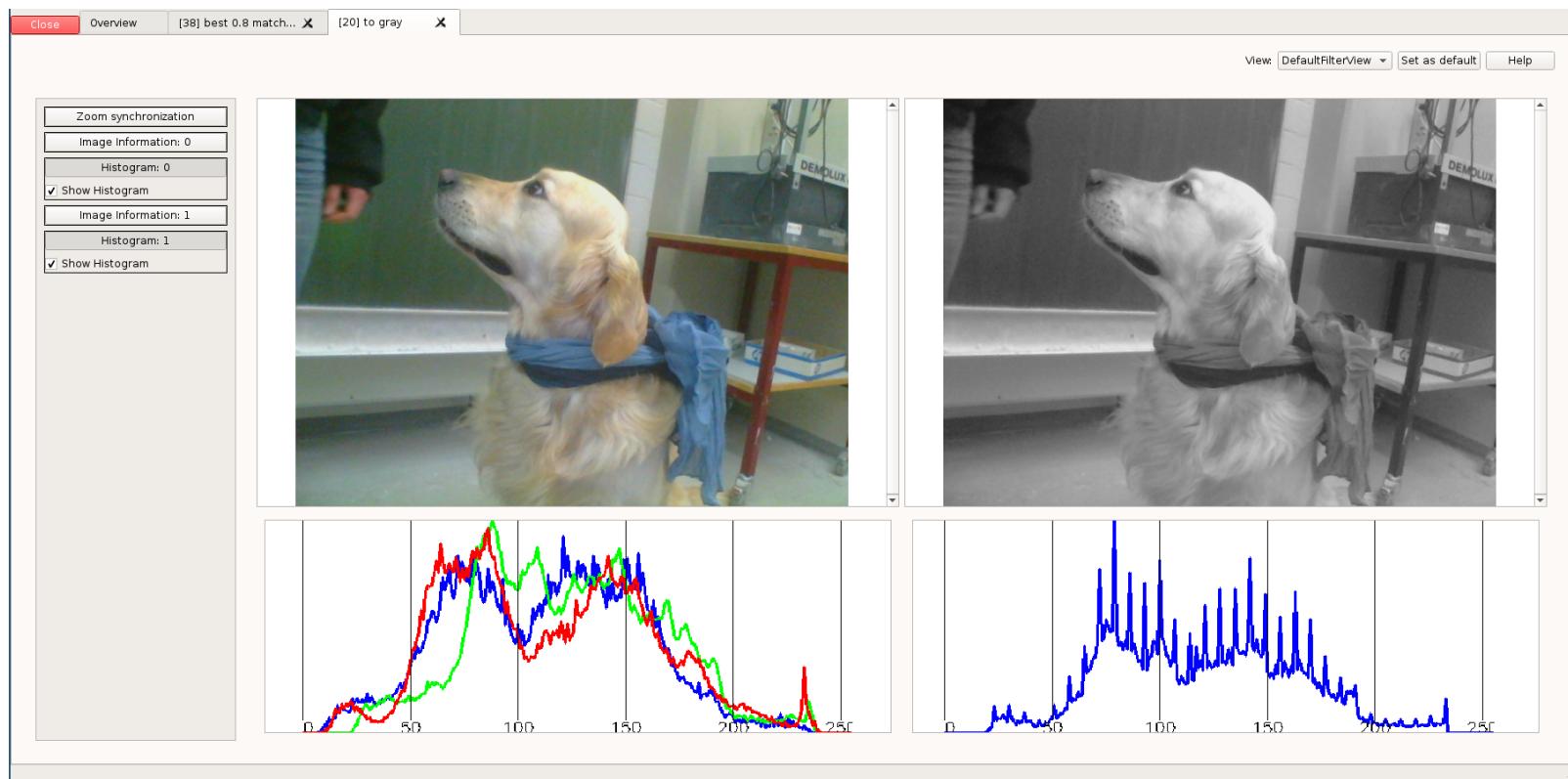
# Entwurf

Trennung in API, Datenhaltung, Visualisierung



# Entwurf

## Erweiterbarkeit durch Modularisierung



# Implementierung

- C++98-kompatible Funktionen-basierte API
- Optional: Kommentar für Aufruf

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

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

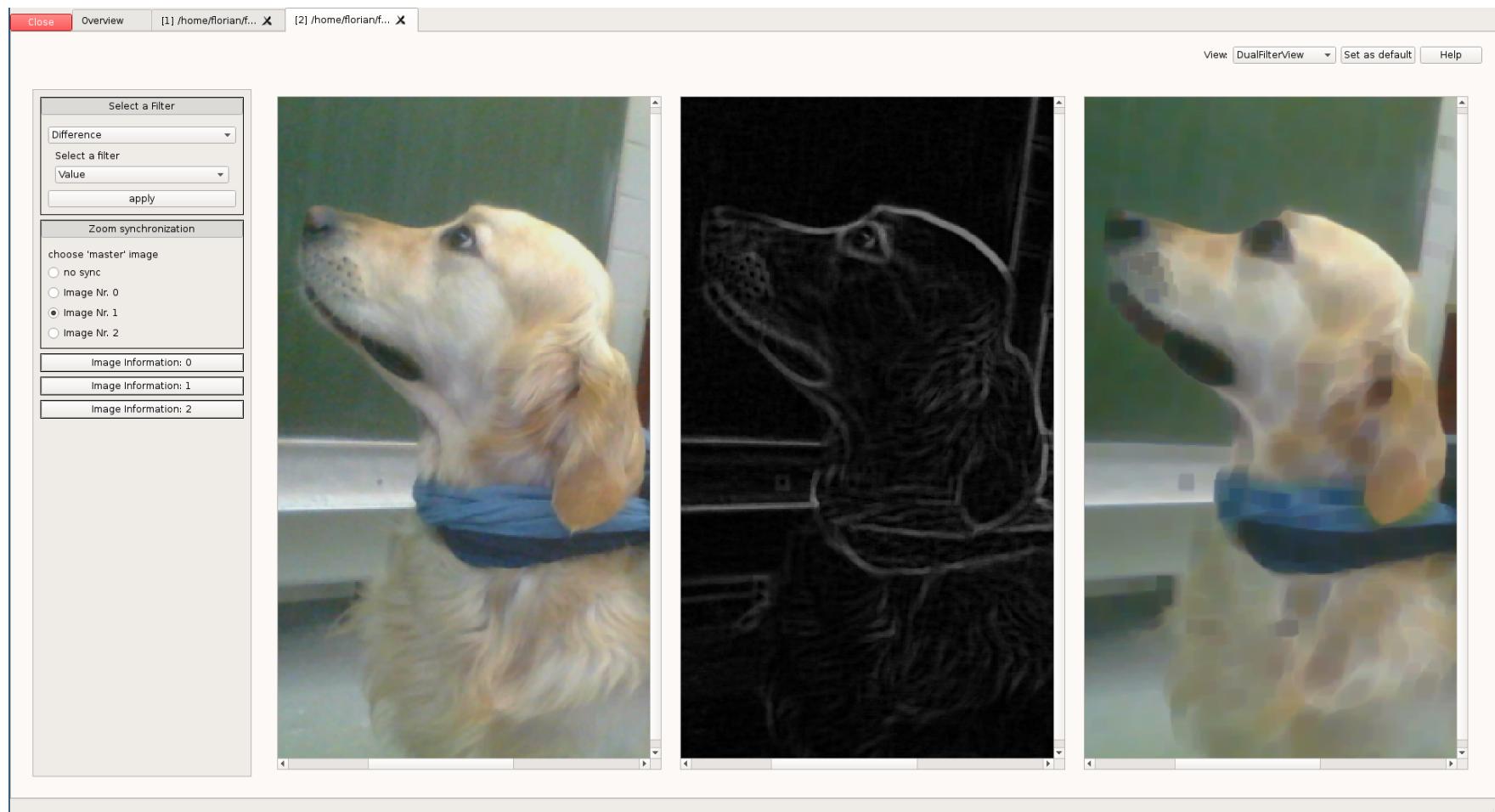
# Implementierung

- Übersicht über alle Aufrufe
- Filter- und sortier- und gruppierbar

27			imgRead7	int main(int, char **)	/home/florian/entwicklung/test/cvvisual_test/main.cpp	97	singleImage	
28			to gray	int main(int, char **)	/home/florian/entwicklung/test/cvvisual_test/main.cpp	102	filter	
29			all matches 6<->7	int main(int, char **)	/home/florian/entwicklung/test/cvvisual_test/main.cpp	117	match	
30			best 0.8 matches 6<->7	int main(int, char **)	/home/florian/entwicklung/test/cvvisual_test/main.cpp	126	match	
								

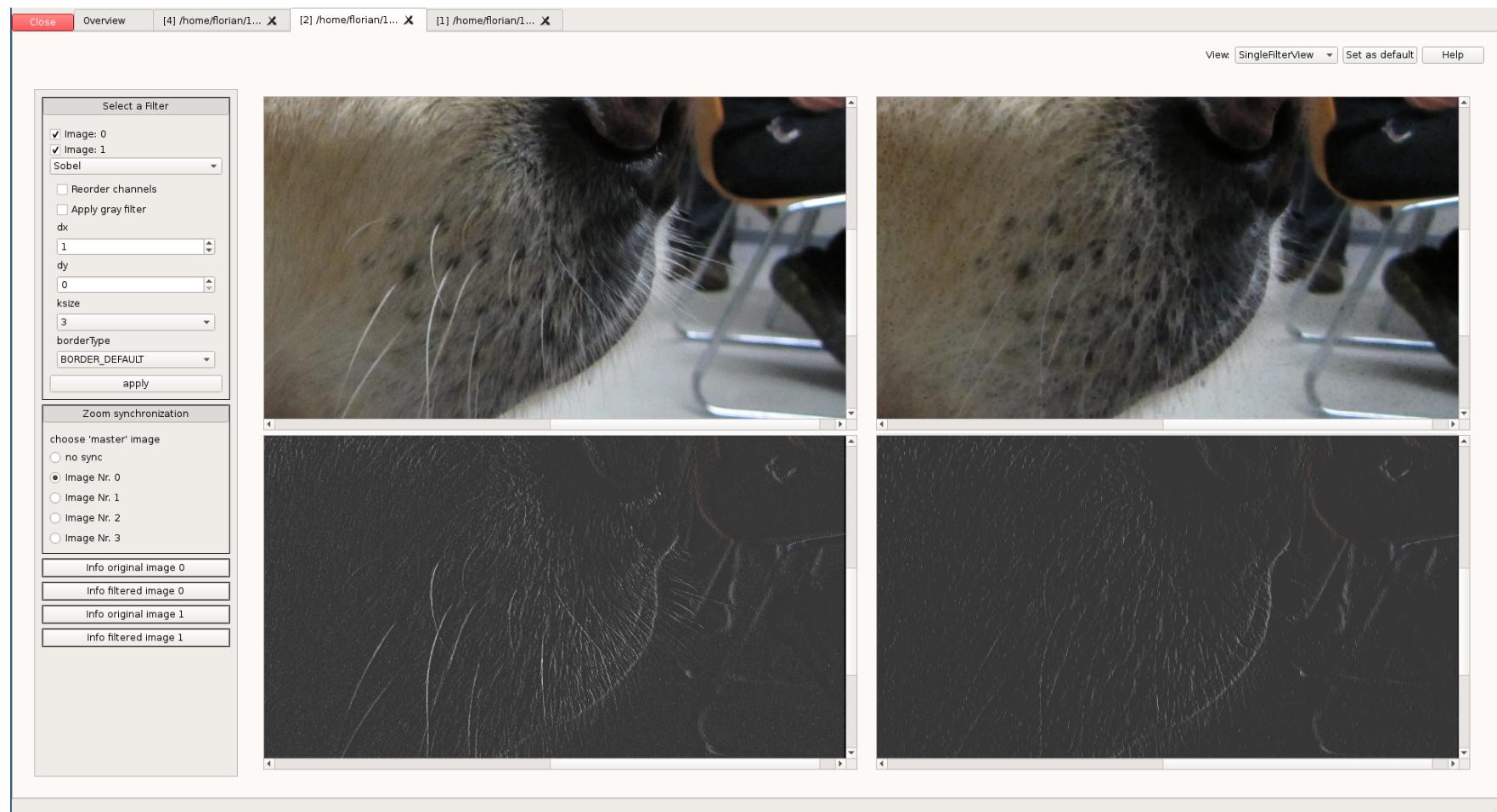
# Implementierung

## Differenzbilder, Overlay, geänderte Pixel für Filter



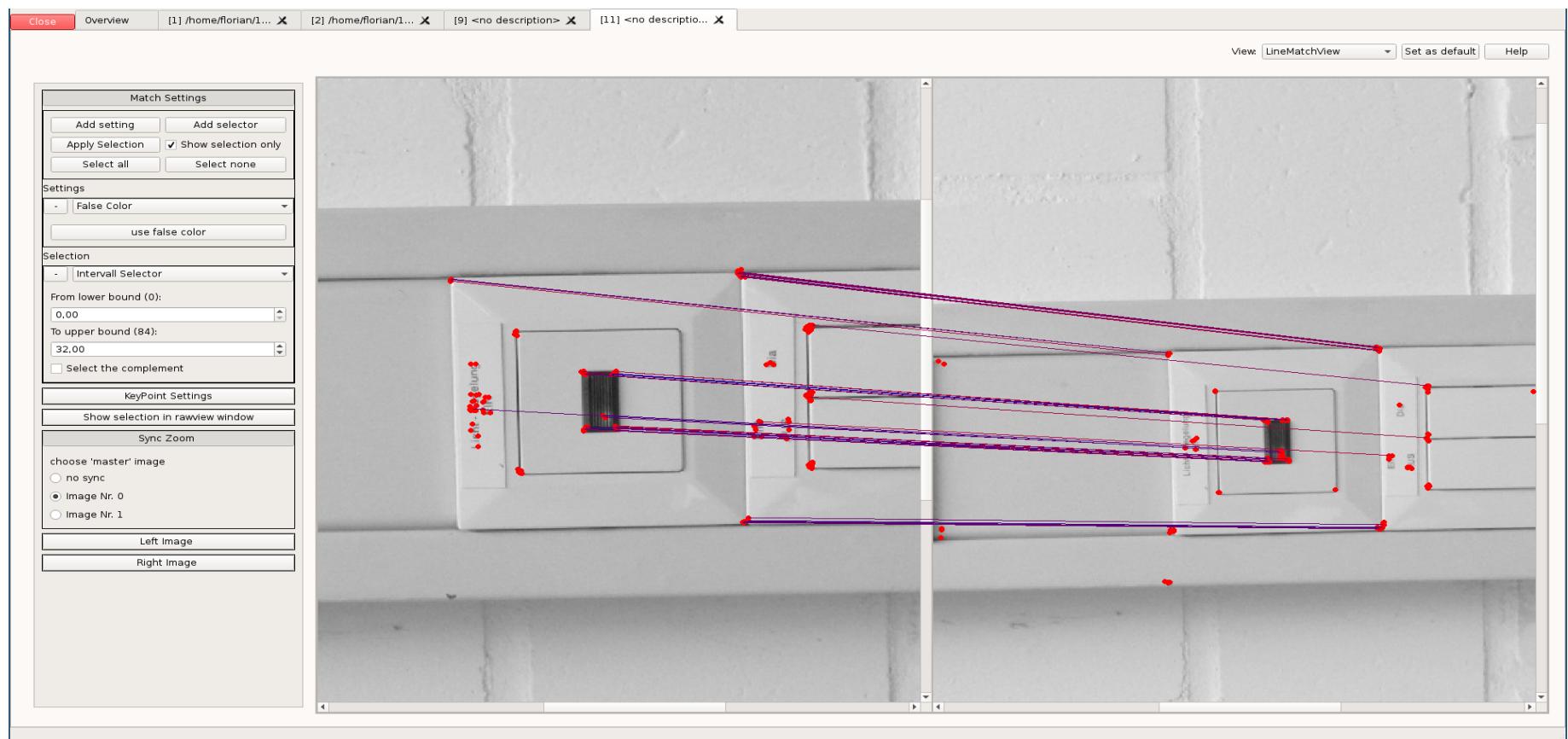
# Implementierung

## Nachträgliche Anwendung weiterer Filter



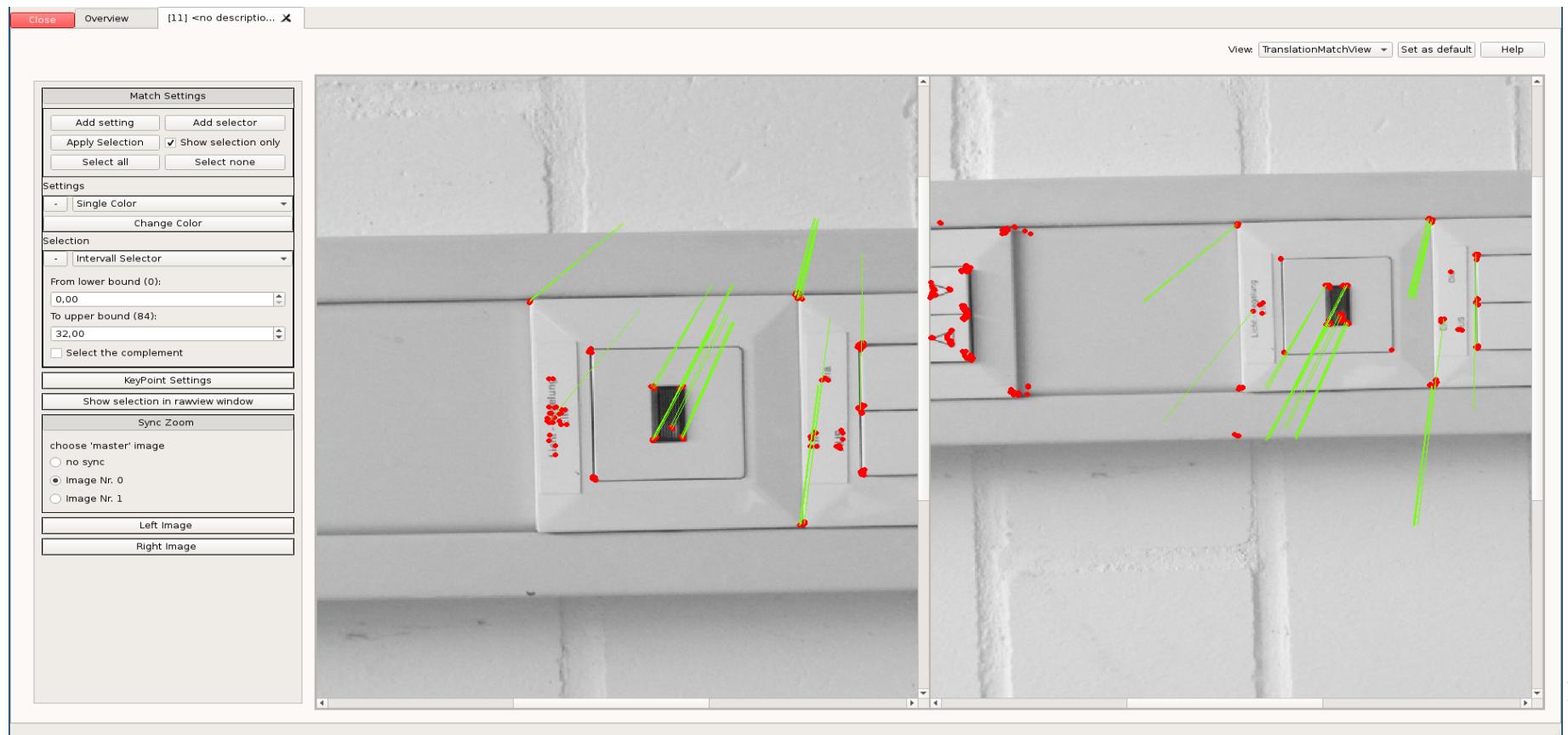
# Implementierung

- Anzeigen und Filtern von Keypoints und Matches
- Anzeige der Translation und Verbindungen von Keypoints



# Implementierung

- Anzeigen und Filtern von Keypoints und Matches
- Anzeige der Translation und Verbindungen von Keypoints



# Umsetzung des Designs

- Qt erzwang die massenhafte Verwendung von „new“
- Threadsafety Qt-bedingt praktisch unmöglich
- Zusammenfassung aller Filter in einer API-Funktion
- Sonst wenig große Änderungen

# Qualitätssicherung

- Automatische Tests für utility-Klassen und Funktionen
- Manuelle Tests für GUI-lastige Dinge

```
[ RUN      ] IsAnyOfTest.TestIsAnyOfLongInt
[       OK ] IsAnyOfTest.VectorLongInt (0 ms)
[-----] 4 tests from IsAnyOfTest (1 ms total)

[-----] 1 test from ObserverPtrTest
[ RUN      ] ObserverPtrTest.ConstructionAssignment
[       OK ] ObserverPtrTest.ConstructionAssignment (0 ms)
[-----] 1 test from ObserverPtrTest (0 ms total)

[-----] 8 tests from ReferenceTest
[ RUN      ] ReferenceTest.Construction
[       OK ] ReferenceTest.Construction (0 ms)
```

# Qualitätssicherung

- Qt neigt zu willkürlichen Beschränkungen und Laufzeitfehlern
- Ubuntu's Qt5 ist fehlerhaft

```
int main(int argc, char** argv) {  
    »    new QApplication{argc, argv};  
    »    QWidget* window = new QWidget;  
    »    window->show();  
    »    QApplication::instance()->exec();  
    »    //delete QApplication::instance();  
}
```

# Features

- Minimaler Overhead im Release-Modus
  - Conditional Compilation
  - Aktivierbarkeit pro Translation-Unit
- C++98-kompatible API ohne Qt-Abhängigkeiten
- Zentrale Verwaltung aller Aufrufe
- Interne API zur leichten Erweiterung
- Unterstützung mehrerer Fenster

# Livedemo

Three windows illustrating a pipeline:

- Top Left Window:** Shows two images side-by-side. On the left is a color image of a mechanical assembly. On the right is a grayscale version of the same image. Below the images are three buttons: "Zoom synchronization", "Image Information: 0", and "Image Information: 1".
- Top Right Window:** Shows four grayscale images arranged in a 2x2 grid. Each image contains numerical values. A "Select a Filter" dialog is open, showing "Image: 0" checked, "Gray filter" selected, and "use default rgb to gray" button.
- Bottom Window:** A table showing operations and their details:
 

ID	Image 1	Image 2	Description	Function	File
1			imgRead0	int main(int, char**)	/home/partimer/Code/PSE/cvvisu
2			to gray	int main(int, char**)	/home/partimer/Code/PSE/cvvisu
3			imgRead1	int main(int, char**)	/home/partimer/Code/PSE/cvvisu
4			to gray	int main(int, char**)	/home/partimer/Code/PSE/cvvisu
5			all matches 0<->1	int main(int, char**)	/home/partimer/Code/PSE/cvvisu
6			best 0.8 matches 0<->1	int main(int, char**)	/home/partimer/Code/PSE/cvvisu
7			imgRead2	int main(int, char**)	/home/partimer/Code/PSE/cvvisu

 Below the table is a "Zoom" slider and a "single key point" table:
 

	x	y	size	angle	response	octave	class id	img number
1	432	319	31	77.0663	0.00345...	0	-1	1
2	372	386	31	60.6421	0.00394...	0	-1	1
3	261	426	31	262.547	0.00397...	0	-1	1
4	277	410	31	86.6925	0.00410...	0	-1	1
5	249	413	31	79.2574	0.00388...	0	-1	1

# Ausblick

## Aufnahme in OpenCV wahrscheinlich:



**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.

# Vielen Dank für Ihre Aufmerksamkeit!

## Fragen?

- Webseite: <http://cvv.mostlynerdless.de>
- Git:
  - Noch: `git@github.com:CVVisualPSETeam/CVVisual.git`
  - Hoffentlich bald: `git@github.com:ltseez/opencv_contrib.git`