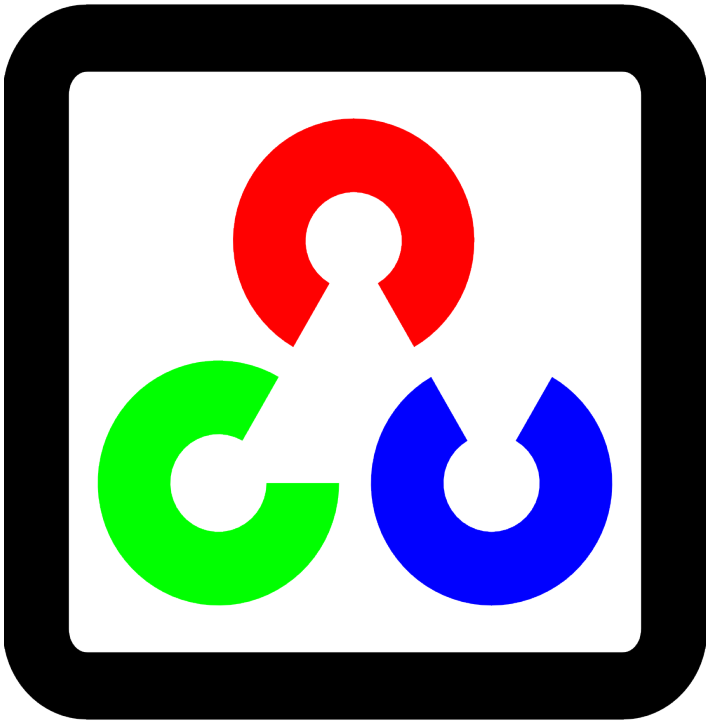
CVVisual



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“

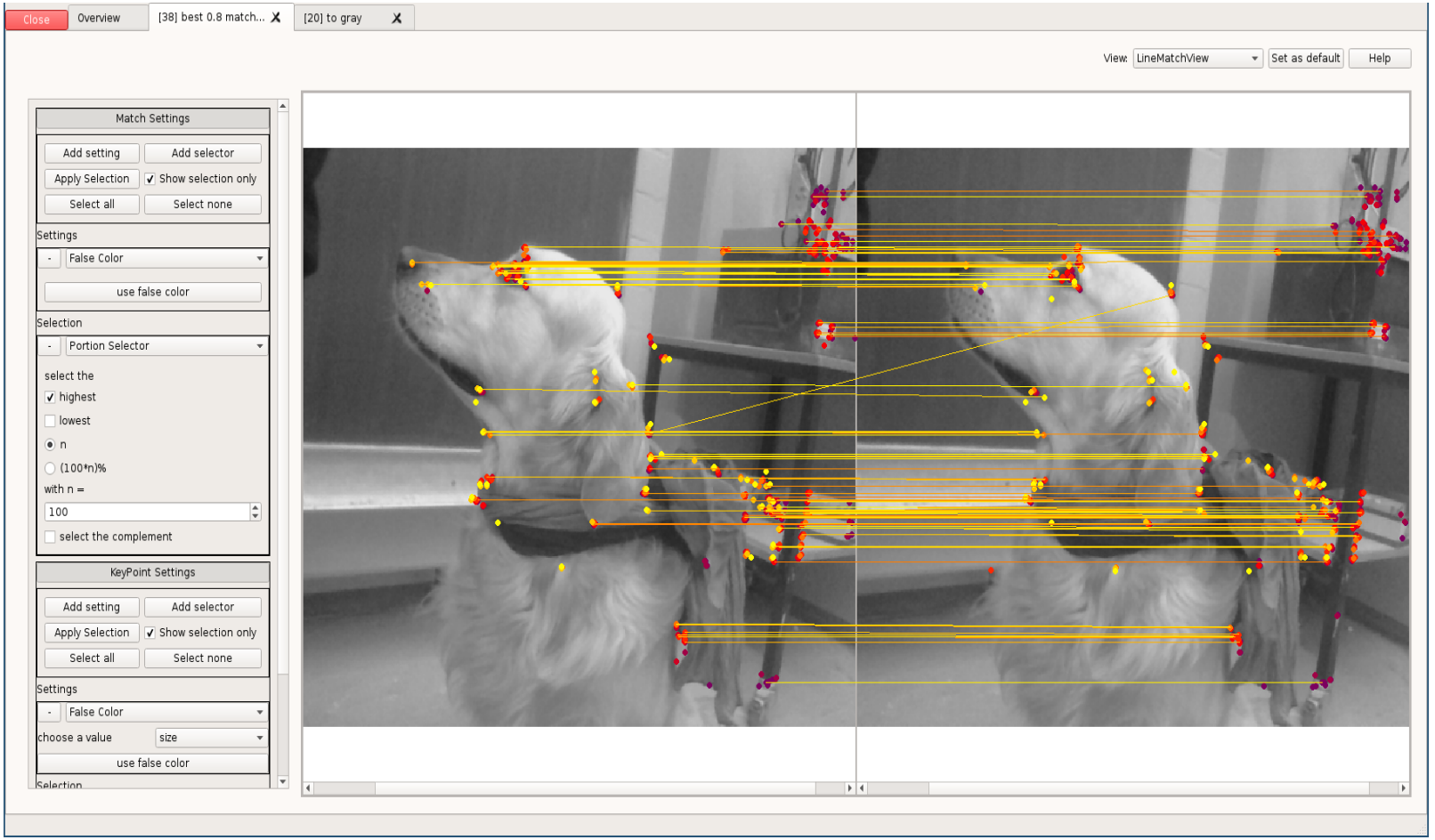
```
#ifdef DEBUG
        Mat img_matches;
        drawMatches( img_1, keypoints_1, img_2, keypoints_2,
                     good_matches, img_matches, Scalar::all(-1), Scalar::all(-1),
                     vector<char>(), DrawMatchesFlags::NOT_DRAW_SINGLE_POINTS );
        imshow("good matches", img_matches);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

versus

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

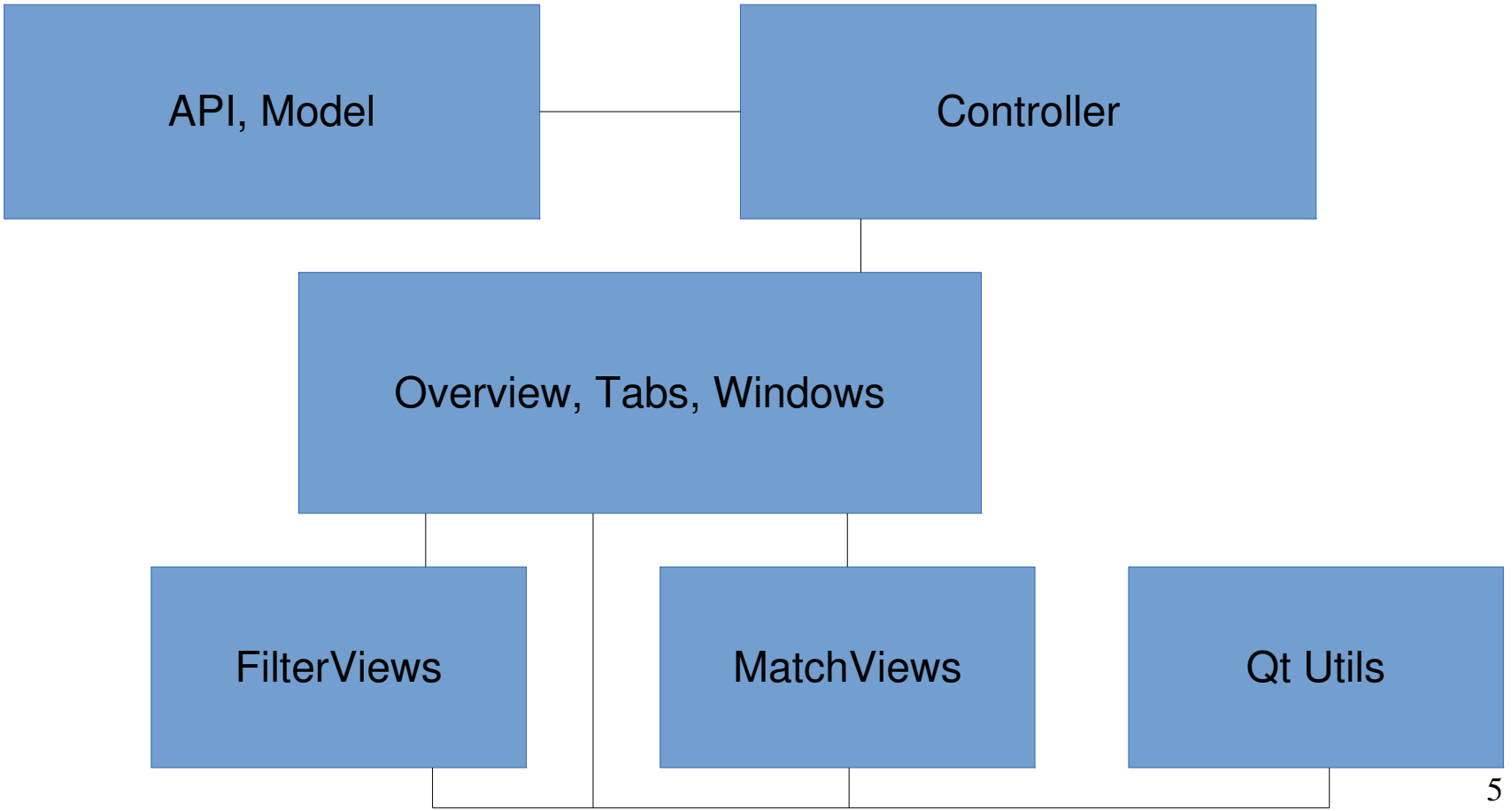
Motivation

Visualisierung von Matritzen, Filtereffekten und Matches.



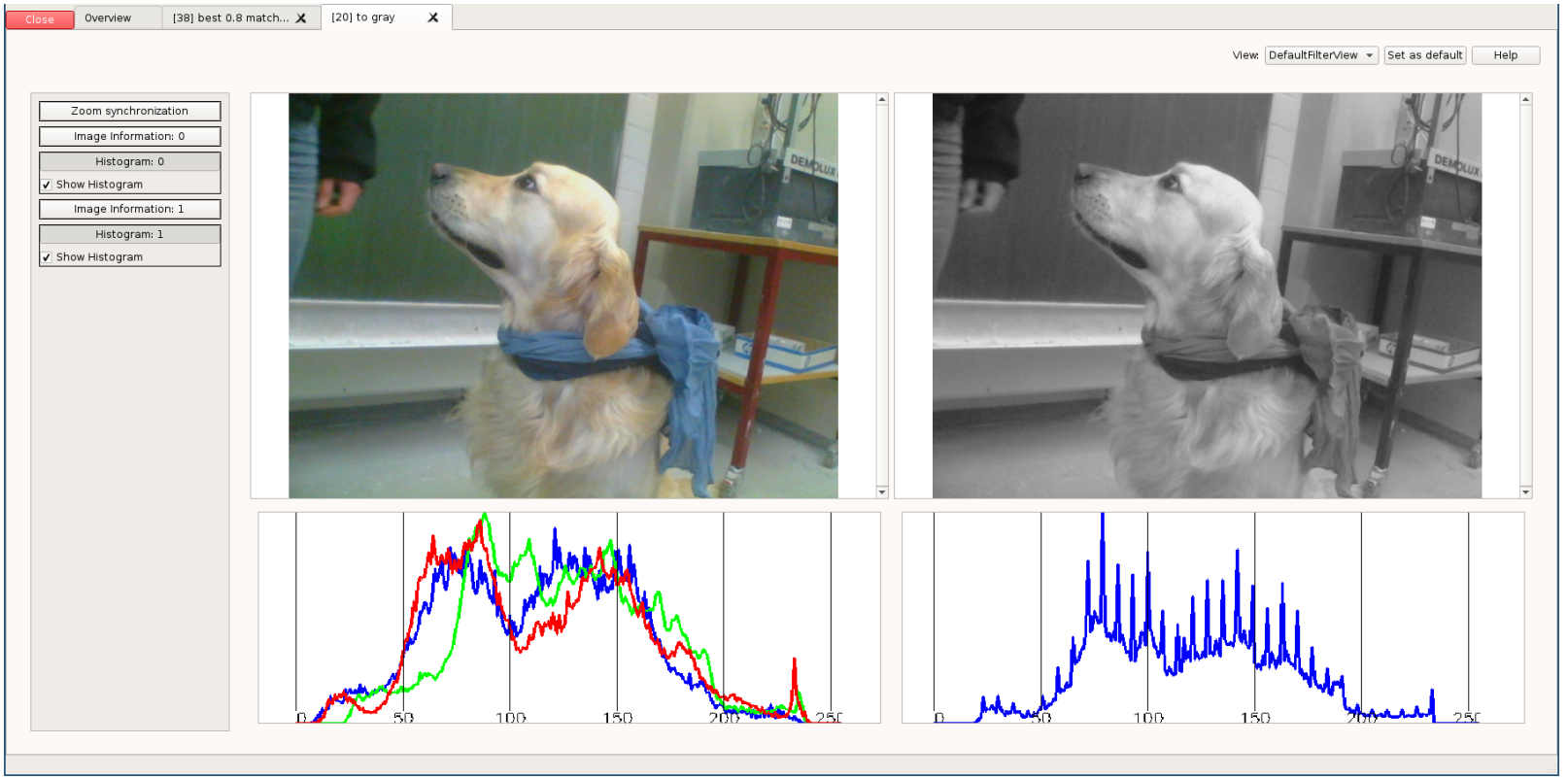
Entwurf

Trennung in API, Datenhaltung, Visualisierung



Entwurf

Erweiterbarkeit durch Modularisierung



Implementierung

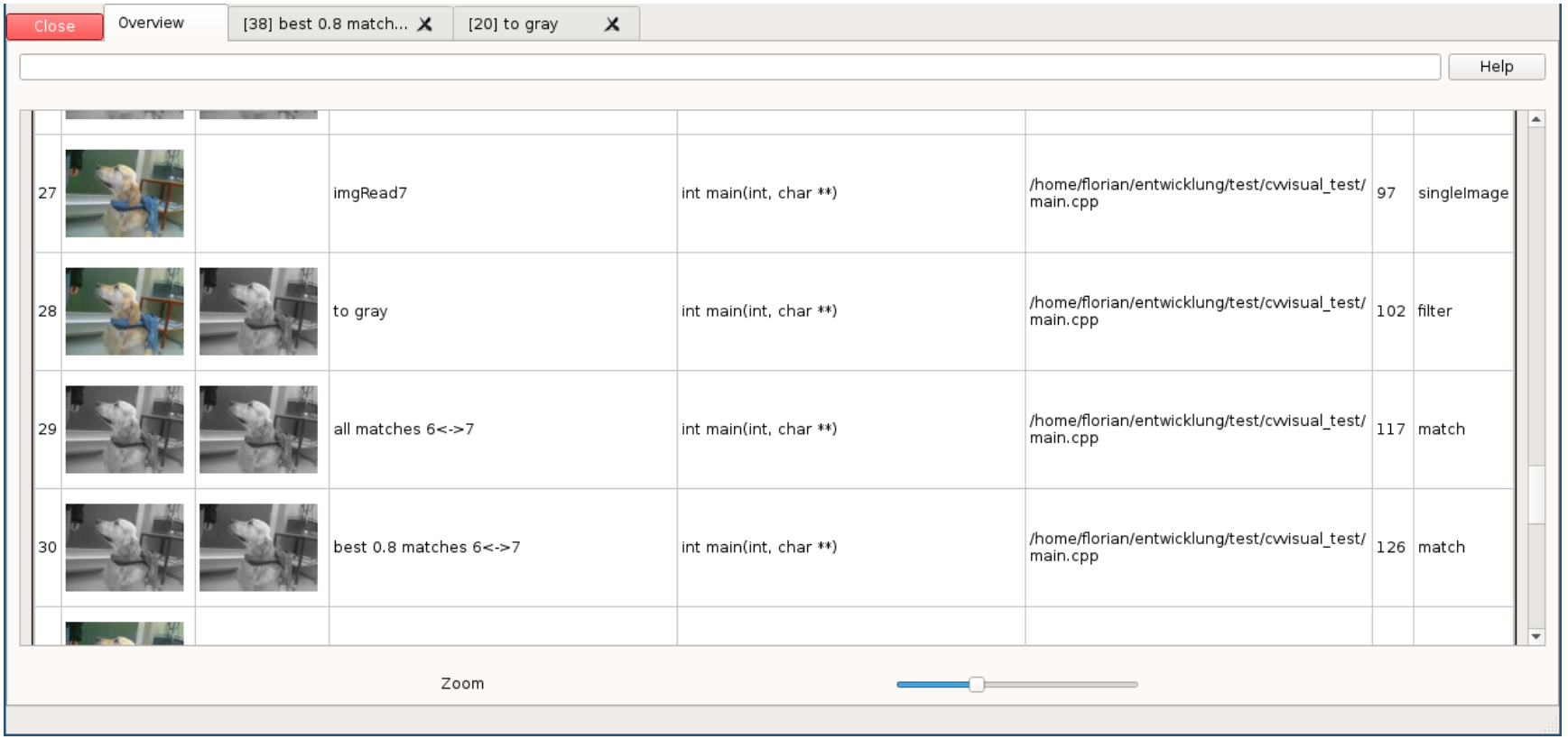
- C++98-kompatible Funktionen-basierte API
- Optional: Kommentar für Aufruf

```
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");
```

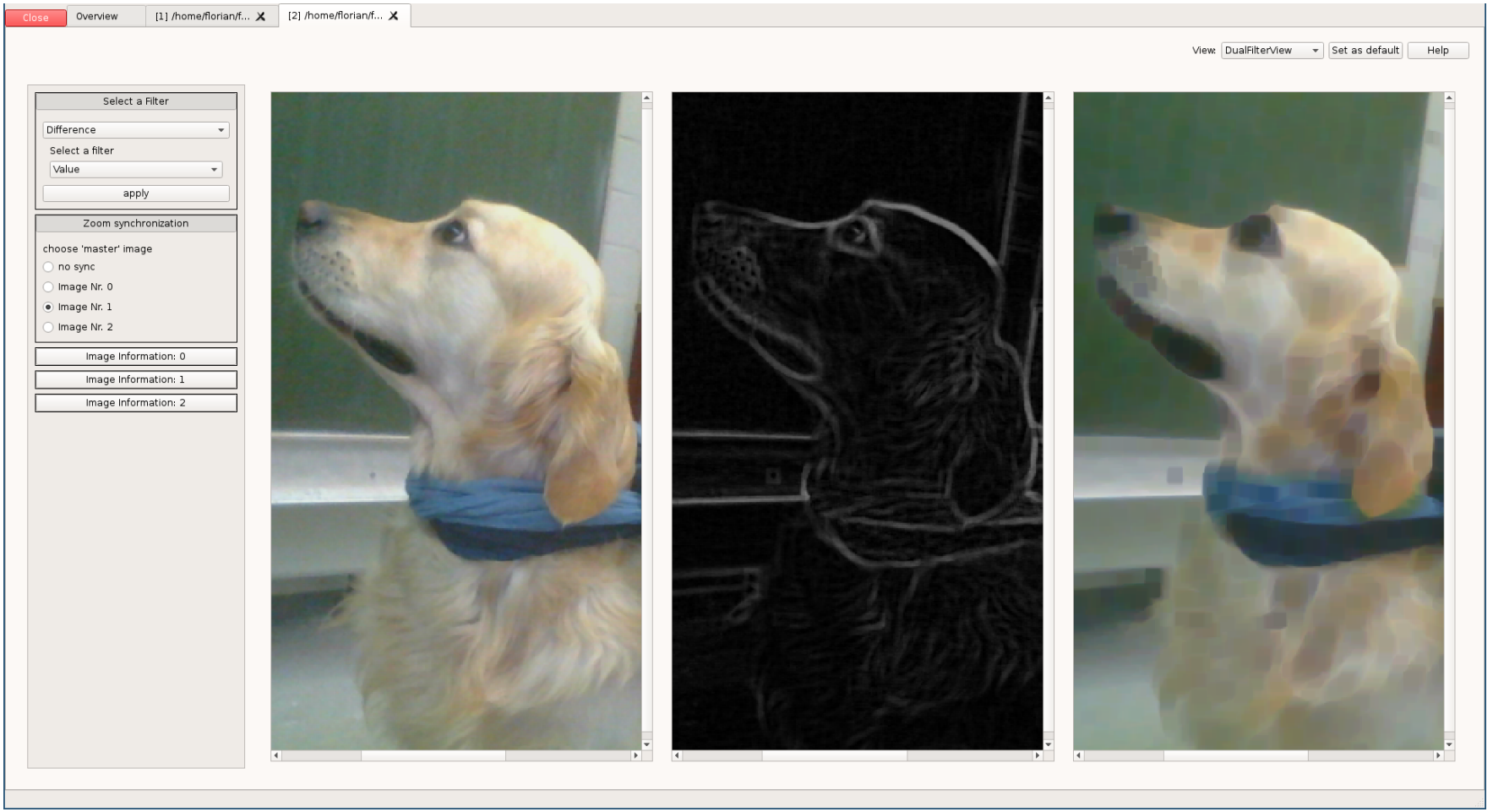
Implementierung

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



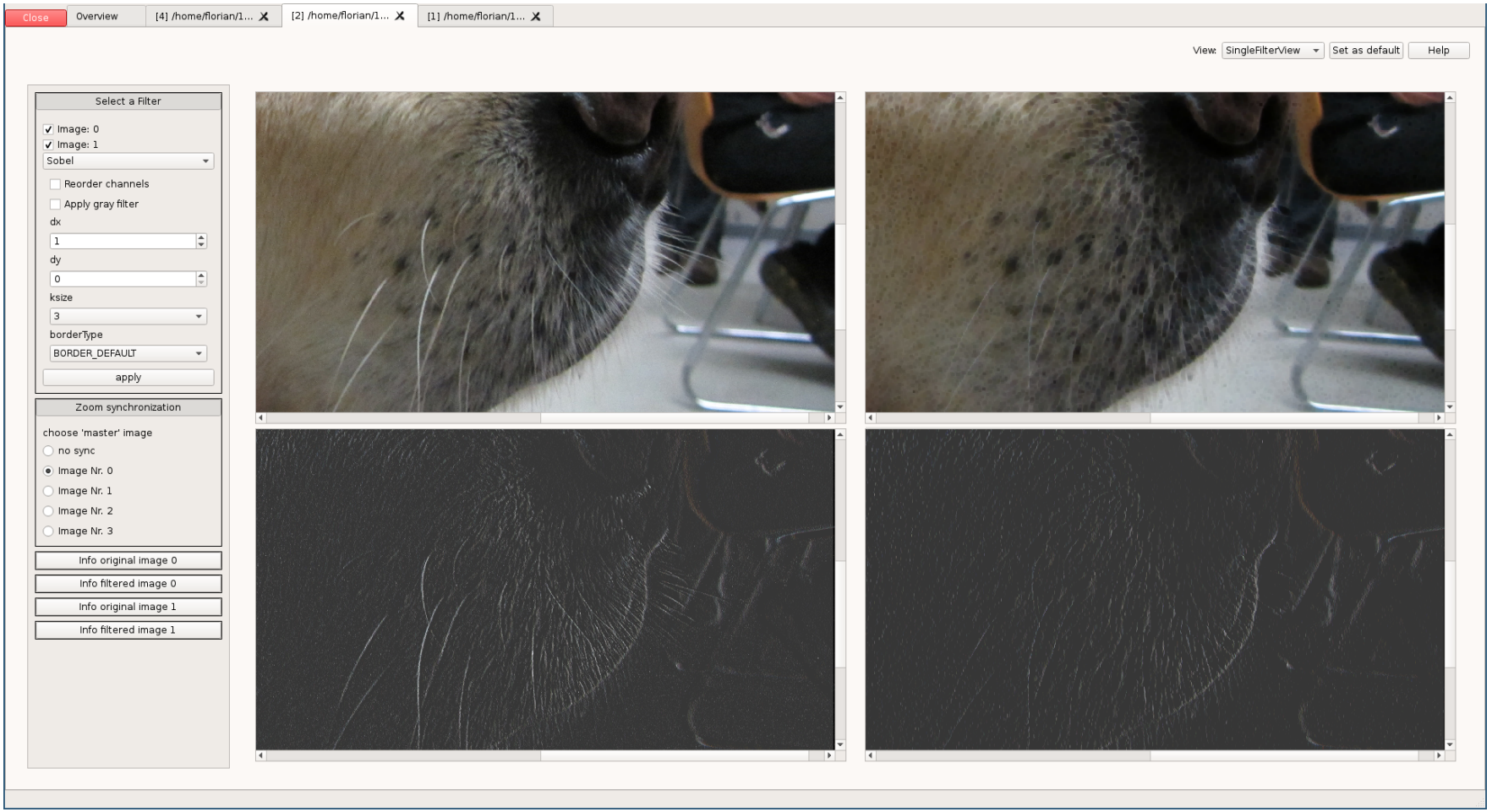
Implementierung

Differenzbilder, Overlay, geänderte Pixel für Filter



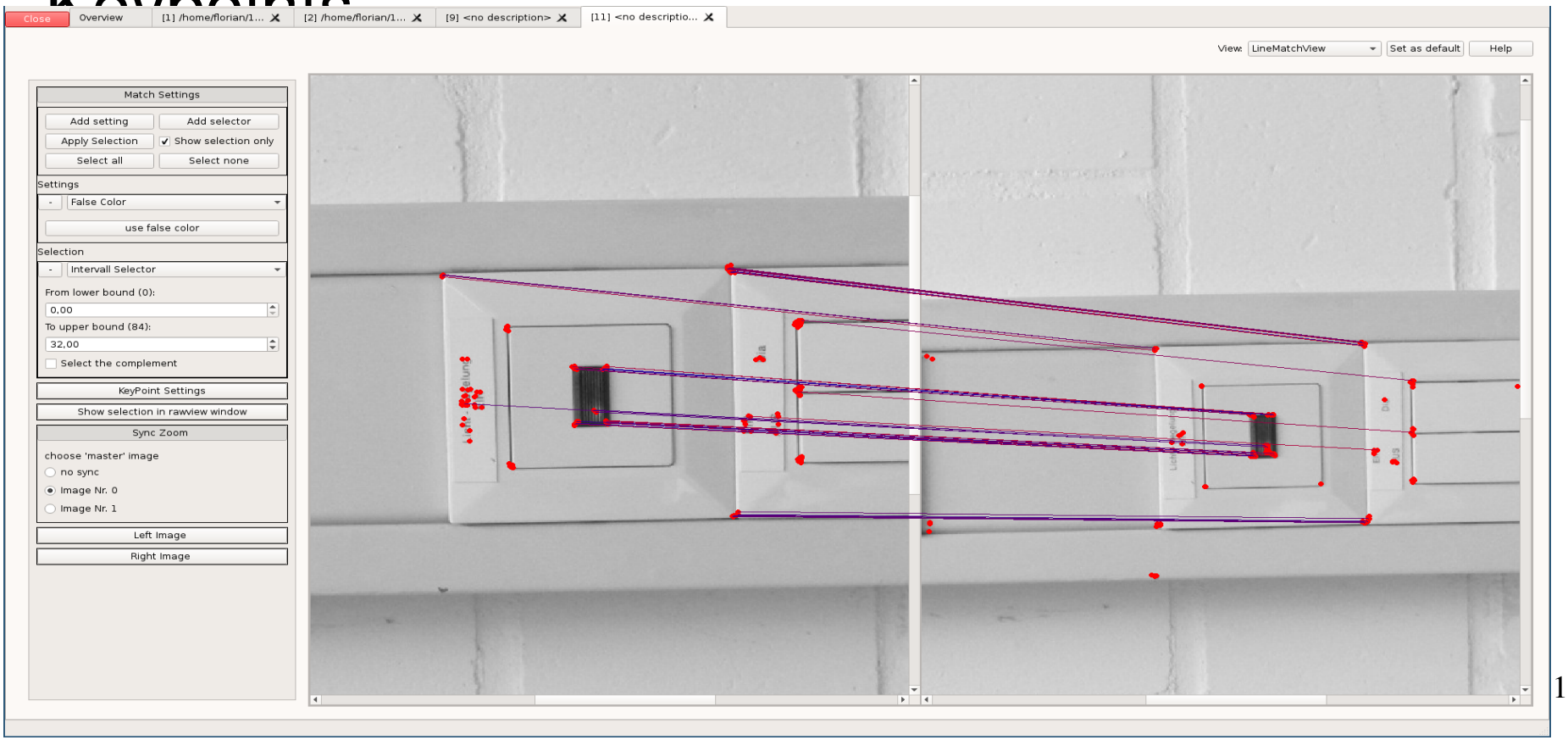
Implementierung

Nachträgliche Anwendung weiterer Filter



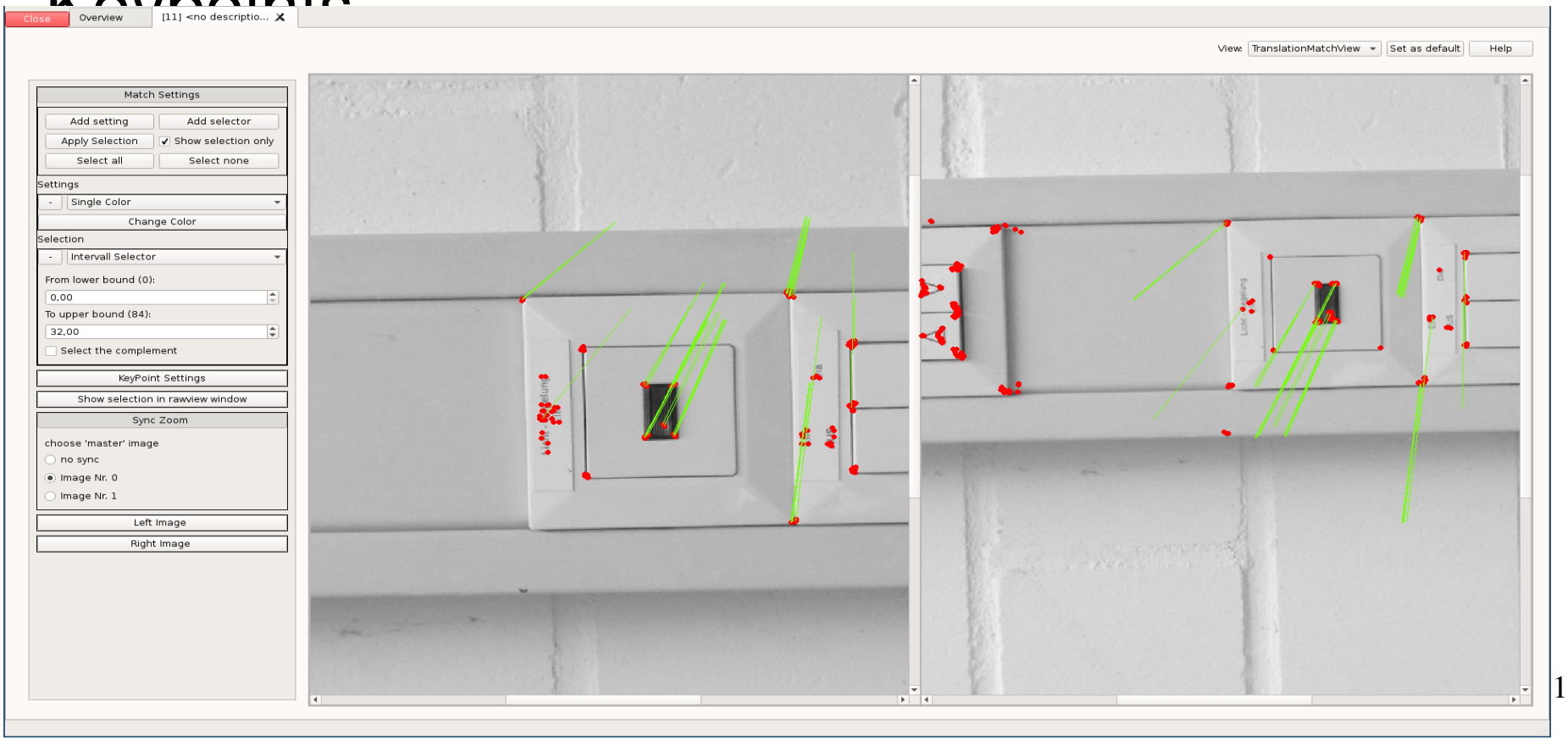
Implementierung

- 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

```
[       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
```

Qualitätssicherung

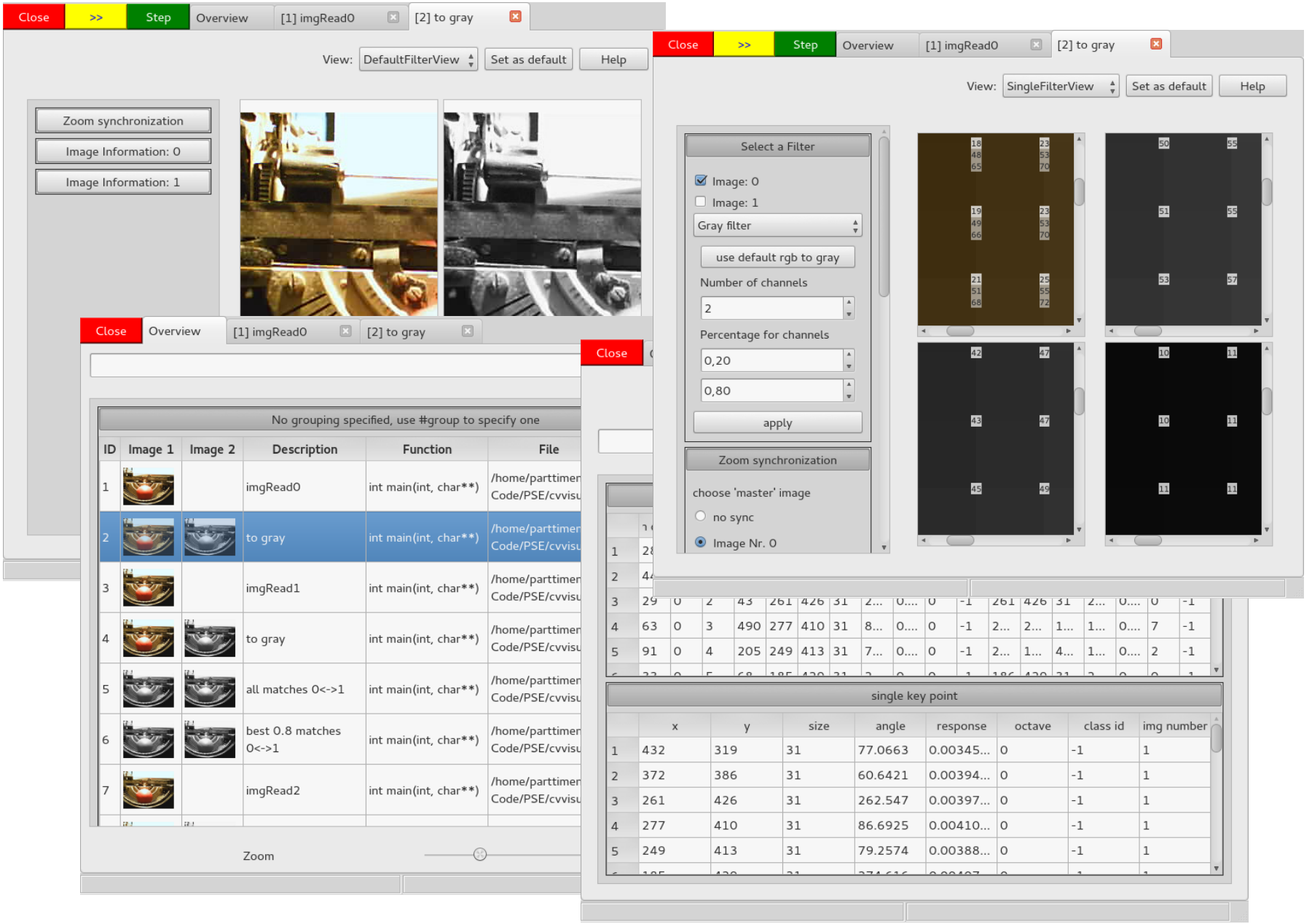
- Qt neigt zu willkürlichen Beschränkungen und Laufzeitfehlern
- Qt5 auf Ubuntu 12.04 ist fehlerhaft

```
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


Close
>>
Step
Overview
[1] imgRead0
[2] to gray
View:
DefaultFilterView
Set as default
Help
Zoom synchronization
Image Information: 0
Image Information: 1
SingleFilterView
Select a Filter
Image: 0
Image: 1
Gray filter
use default rgb to gray
Number of channels
2
Percentage for channels
0,20
0,80
apply
Zoom synchronization
choose 'master' image
no sync
Image Nr. 0
No grouping specified, use #group to specify one
ID
Image 1
Image 2
Description
Function
File
imgRead0
imgRead1
to gray
all matches 0<->1
best 0.8 matches 0<->1
imgRead2
int main(int, char**)
Zoom
single key point
x
y
size
angle
response
octave
class id
img number

Ausblick

Aufnahme in OpenCV schien 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:Itseez/opencv_contrib.git