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 + Demo
- Ausblick (?)

Einführung in OpenCV

Überblick

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

Matrizen

- Bild = mehrdimensionale Matrix
- 3. Dimension = Channel //Bsp. BGR-Bild: 1. Channel blau, 2. Channel grün usw.

Filter

- 2D-Bilder
- Berechnung auf Umgebung jedes Pixels
- Bsp: dilate, erode, Sobel //erode -> kleine Details weg

Matches

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

Motivation

Debuggen von OpenCV

Systematisches Debugging statt „Random Code“ //Hinweis auf showMatches/showKeypoints

Ziele

- Visualisierung von Matritzen, Filtereffekten und Matches

Anwenderfeatures

Verwendung

Übersicht

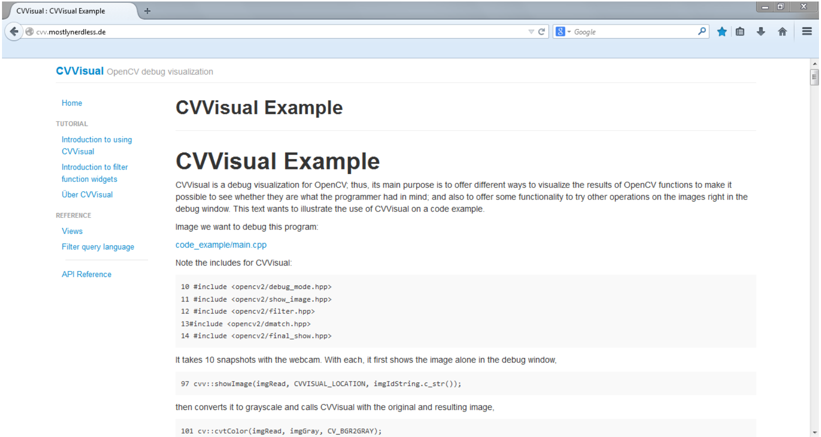
Filter

Matches

GUI-Demo

Dokumentation

Tutorials, Beispiele


CVisual : CVVisual Example
cvv.mostlynerdless.de
Google
CVVisual OpenCV debug visualization
Home
TUTORIAL
Introduction to using CVVisual
Introduction to filter function widgets
Über CVVisual
REFERENCE
Views
Filter query language
API Reference

CVVisual Example

CVVisual Example

CVVisual is a debug visualization for OpenCV; thus, its main purpose is to offer different ways to visualize the results of OpenCV functions to make it possible to see whether they are what the programmer had in mind; and also to offer some functionality to try other operations on the images right in the debug window. This text wants to illustrate the use of CVVisual on a code example.

Image we want to debug this program:

code_example/main.cpp

Note the includes for CVVisual:

```
#include <opencv2/debug_mode.hpp>
#include <opencv2/show_image.hpp>
#include <opencv2/filter.hpp>
#include <opencv2/dmatch.hpp>
#include <opencv2/final_show.hpp>
```

It takes 10 snapshots with the webcam. With each, it first shows the image alone in the debug window,

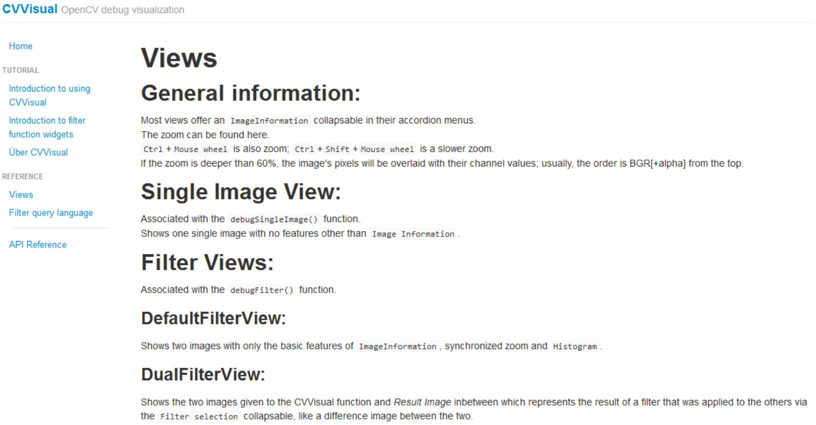
```
cvv::showImage(imgRead, CVVISUAL_LOCATION, imgIdString.c_str());
```

then converts it to grayscale and calls CVVisual with the original and resulting image,

```
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:

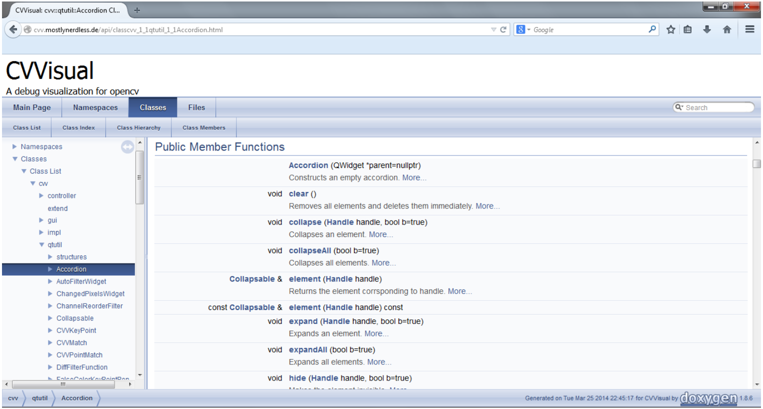
Shows two images with only the basic features of `ImageInformation`, synchronized zoom and `Histogram`.

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



CVVisual

A debug visualization for opencv

Public Member 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...
Collapsable &	**element** (**Handle** handle) Returns the element corrsponding to handle. More...
const **Collapsable** &	**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...

Generated on Tue Mar 25 2014 22:45:17 for CVVisual by doxygen 1.8.6

Arichtektur

API

Anwender API

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

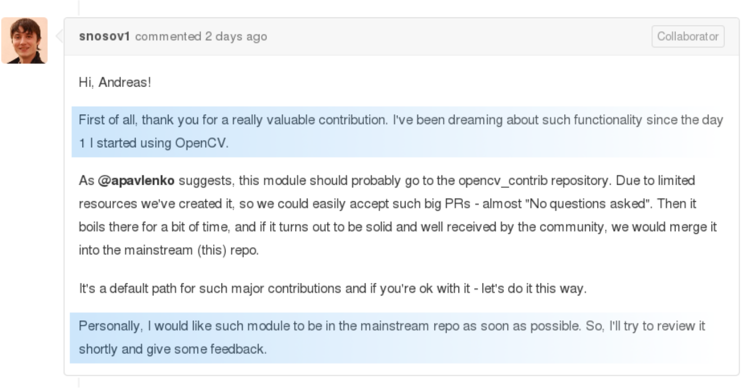
Interne API

- Leichtes, zentralisiertes Hinzufügen von Visualisierungen, Filtern, . . .

Ausblick

Rezeption

Projekt schien von der OpenCV-Community wohlwollend aufgenommen zu werden



snosov1 commented 2 days ago

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.

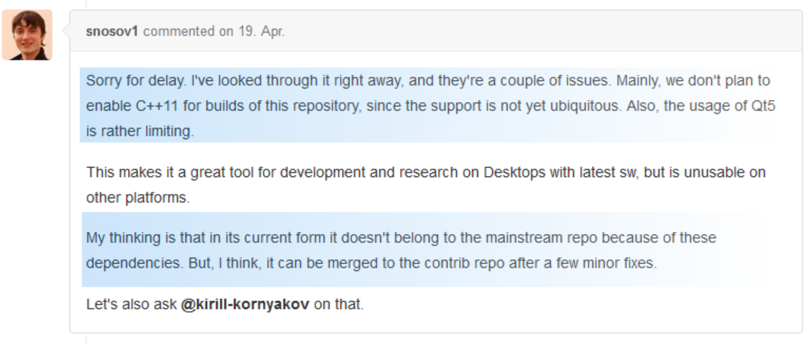
As **@apavlenko** suggests, this module should probably go to the opencv_contrib repository. Due to limited resources we've created it, so we could easily accept such big PRs - almost "No questions asked". Then it boils there for a bit of time, and if it turns out to be solid and well received by the 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/`