

OpenCV und Smart Kameras Stand-alone Bildverarbeitung quellenoffen programmieren

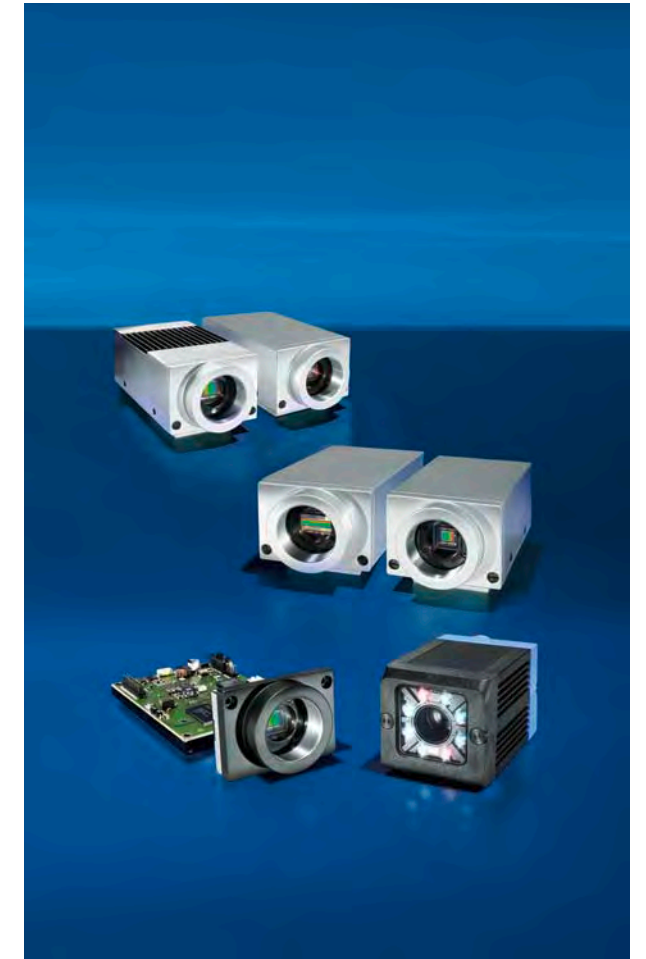
Michael Engel

Vision Components, Ettlingen



Inhaltsverzeichnis

- Kurzvorstellung der Vision Components GmbH
- Was ist eine Smart Kamera
- Möglichkeiten der Programmierung von Smart Kameras
- Was ist OpenCV?
- Wichtige Datenstrukturen von OpenCV
- Das OpenCV Interface
- Übersicht über die OpenCV – Funktionen
- OpenCV Beispielprogramme mit Bildern



Smart Kameras von Vision Components decken die ganze Bandbreite von Bildverarbeitungsapplikationen ab.

Vorstellung der Vision Components GmbH

- VC ist innovativer Hersteller intelligenter Kameras für die industrielle Bildverarbeitung
- weltweit erste industrielle Smart Kamera 1996
- führender Hersteller von frei programmierbaren Smart Kameras mit weltweitem Distributionsnetzwerk
- Weltweit mehr als 75000 Systeme im Einsatz
- Geschäftsführer Michael Engel ist einer der Pioniere der industriellen Bildverarbeitung mit über 25 Jahren Erfahrung in diesem Bereich
- Besuchen Sie uns auf der Vision in Halle 4, Stand 4D31 !



Was ist eine Smart Kamera?

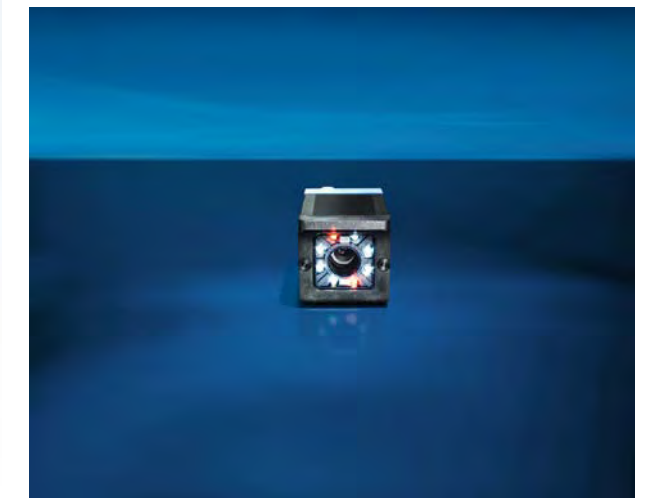
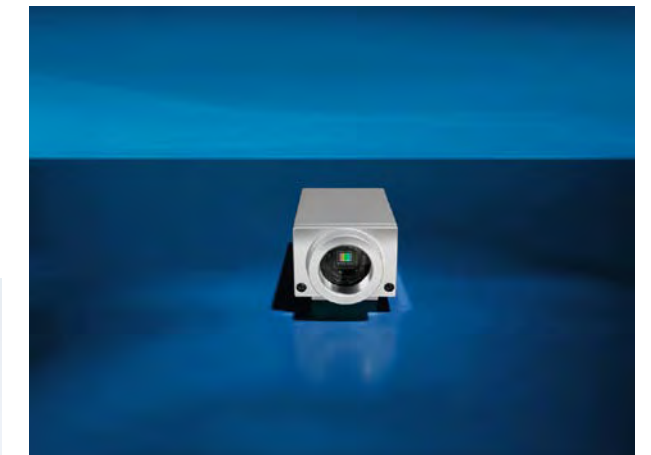
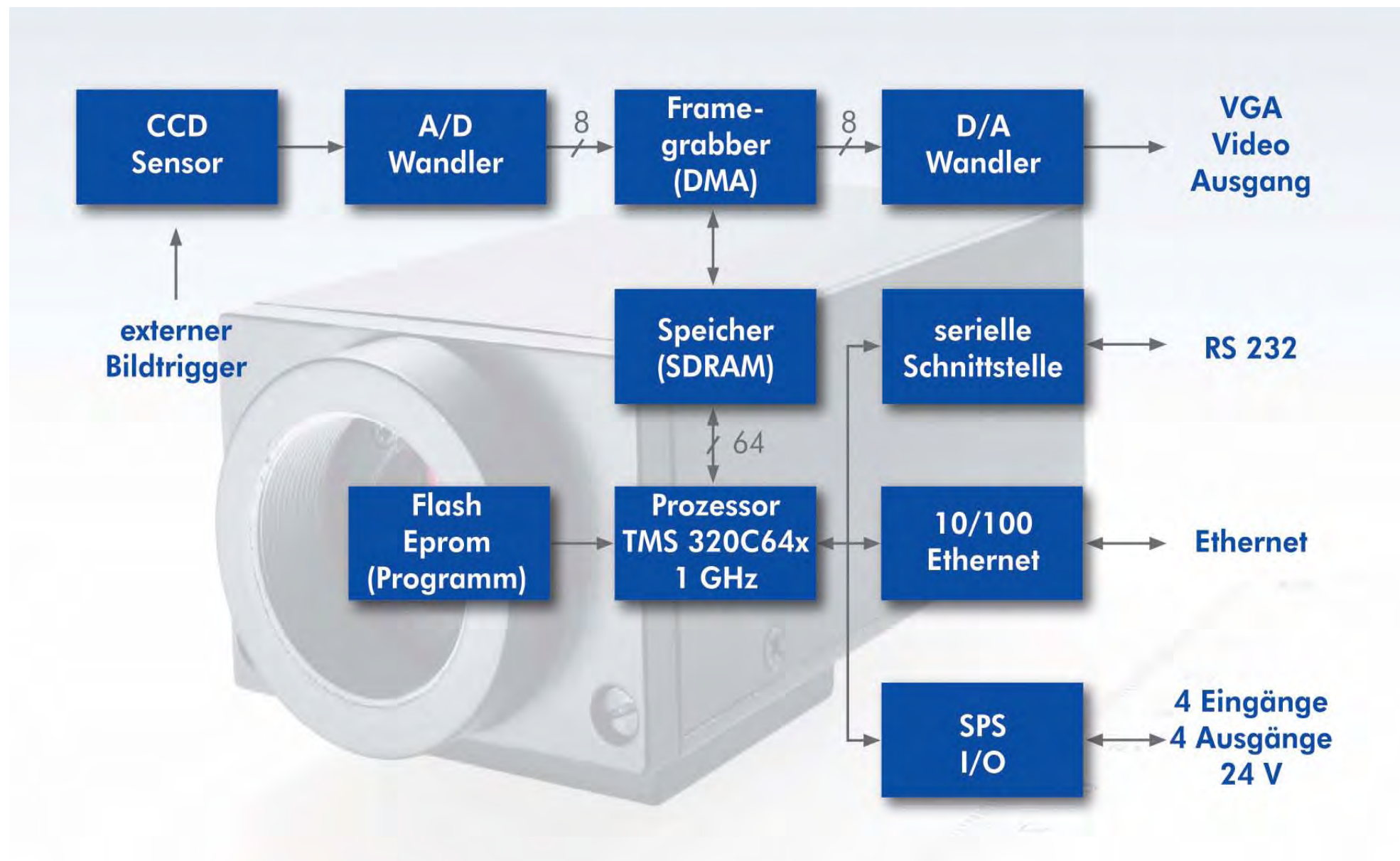
Eine intelligente Kamera ist ein komplettes **Bildverarbeitungssystem** mit Bildsensor in einem kompakten Gehäuse.

Eine Smart Kamera enthält

- einen hochwertigen CCD Sensor
- Framegrabber
- leistungsfähiger Prozessor (DSP)
- SPS I/Os (24V)
- serielle / Ethernetschnittstelle
- schneller Triggereingang für Bildtriggerung
- SVGA-Videoausgang zum Anschluss eines TFT-Monitors

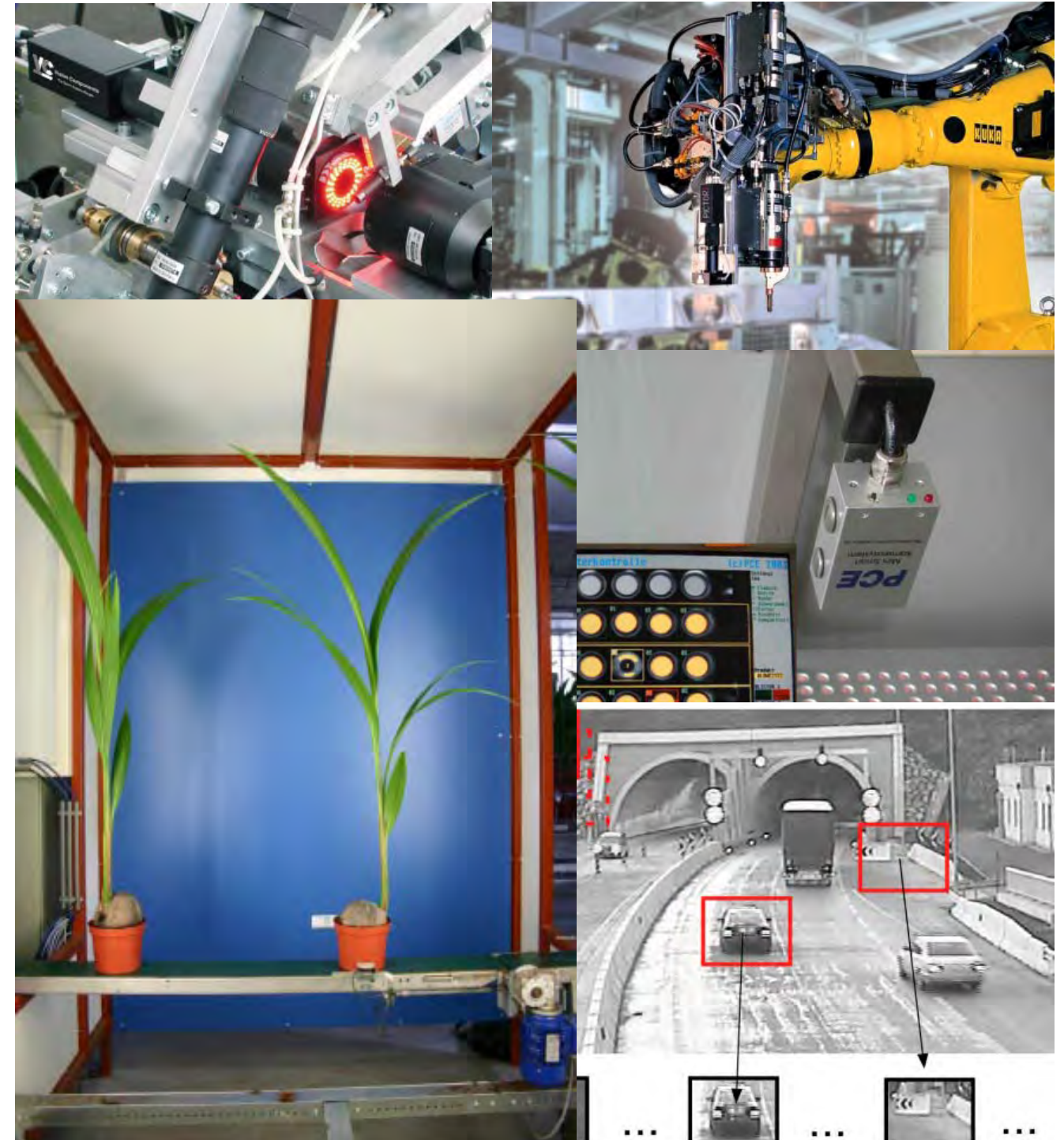


Aufbau einer Smart Kamera



Anwendungen

- industrielle Fertigungsprozesse
- Maschinenbau
- Messtechnik
- Druckindustrie
- Qualitätssicherung
- 1D und 2D Codes lesen
- Roboteranwendungen
- Lebensmittelindustrie
- Sicherheitstechnik



Verschiedene Möglichkeiten der Programmierung von Smart Kameras

- Parametrierung mittels Universalsoftware (auf PC, dann Download)
- Parametrierung mit Menüsteuerung direkt auf der Smart Kamera
- Programmierung in C / C++ mit Cross-Compiler auf PC
- Verwendung von Funktionsbibliotheken:

VCLIB	Vision Components
HALCON	MVTec
OpenCV	open-source



Eine Vielzahl von Anwendungsprogrammen und Softwarebibliotheken ist auf den Smart Kameras von Vision Components einsetzbar

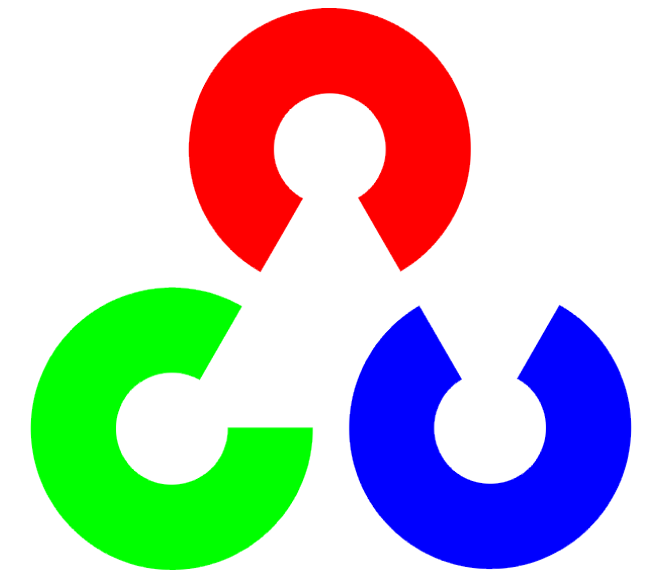
Was ist OpenCV?

OpenCV wurde ursprünglich von Intel® entwickelt für die Echtzeit-Bildverarbeitung mit folgender Zielrichtung:

- Mensch-Maschine Schnittstelle
- Objekterkennung
- Gesichts- und Gestenerkennung
- Bewegungsverfolgung (Tracking)
- Mobile Roboter
- Stereo-Bildverarbeitung

Lauffähig auf verschiedenen Plattformen: PC Windows / Linux, embedded Systeme, Smart Kameras (VCRT)

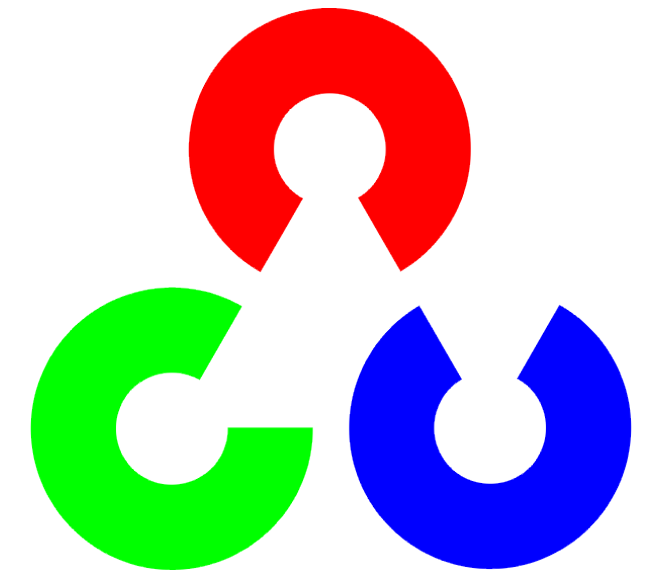
Lizenz: BSD Open Source – kostenloser Source Code, aber kein Support
Buch: Learning OpenCV, Computer Vision with the OpenCV Library



Datenstrukturen von OpenCV

Vergleich der image-Deskriptoren OpenCV und VCLIB

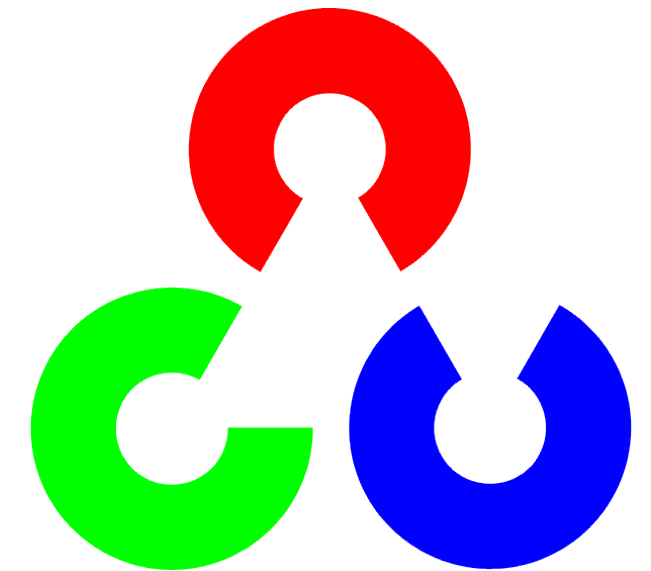
openCV <code>IplImage</code>	VCLIB <code>image</code>	Bemerkung
---	<code>type=</code> <code>IMAGE_GREY</code> <code>IMAGE_BAYER</code> <code>IMAGE_RGB</code> <code>IMAGE_CBCR444</code> <code>IMAGE_CBCR411</code> <code>IMAGE_YUVNORM</code> <code>IMAGE_IHS</code> <code>IMAGERGBO</code> <code>IMAGE_VECTOR</code> <code>IMAGE_GREY16</code> <code>IMAGE_GREY32</code>	<code>type</code> enthält Informationen über Anzahl der <i>channels</i> , das Speicherformat und die Pixeltiefe
<code>imageDataOrigin</code> <code>width</code> <code>height</code> ---	<code>st</code> <code>dx</code> <code>dy</code> <code>pitch</code>	Startadresse Bildbreite Bildhöhe openCV: <code>pitch = width</code>



Datenstrukturen von OpenCV

Vergleich der image-Deskriptoren OpenCV und VCLIB

openCV <code>IplImage</code>	VCLIB <code>image</code>	Bemerkung
<code>depth=</code>	---	Anzahl der Bits pro Pixel und Zahlenformat
<code>IPL_DEPTH_8U</code>		
<code>IPL_DEPTH_8S</code>		
<code>IPL_DEPTH_16S</code>		
<code>IPL_DEPTH_32S</code>		
<code>IPL_DEPTH_32F</code>		
<code>nChannels</code>	---	Anzahl der <i>channels</i>
<code>dataOrder</code>	---	VCLIB: verwendet immer <i>planes</i>



Das OpenCV Interface

4 zusätzliche Funktionen:

```
int init_VC_openCV_interface()
```

Initialisierung von OpenCV und Interfacefunktionen, VCRT ANSI I/O

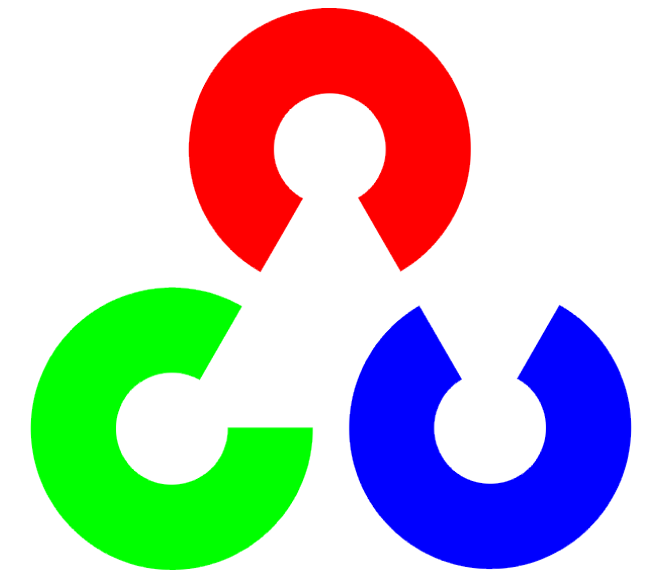
```
void cvSetImageStart(IplImage *img, char *start)
```

Zuweisung der Startadresse eines Bildspeichers (z.B. aktueller Kamerabildspeicher zu einem OpenCV Image-Deskriptor

```
I32 copyVC(IplImage *src, image *dst)
```

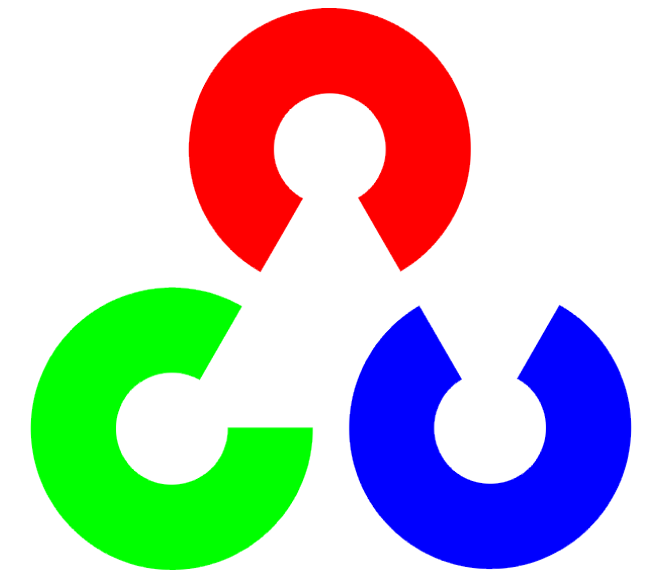
```
I32 copyCV(image *src, IplImage *dst)
```

Bildkopierfunktionen zwischen OpenCV und VCLIB-Bildspeichern, nur nötig, bei unterschiedlicher Speicher methode (z.B. bei Farbbildern)



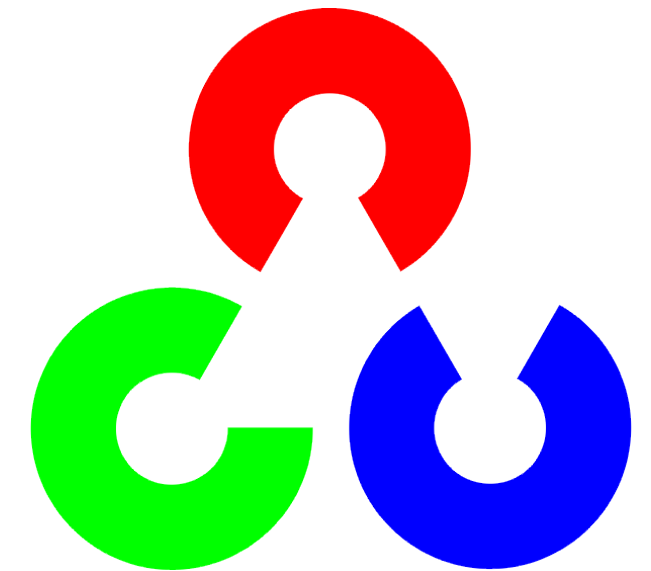
Wichtige OpenCV-Funktionen

CXCORE Funktion	VCLIB Funktion	Bemerkung
cvCreateImage	ImageAllocate	erzeugt <i>header</i> und alloziert Daten
cvCopy	copy	kopiert Bilder
cvSet	set	setzt Bild auf konstanten Wert
cvLUT	look	Lookuptabellen-Transformation
cvAdd, cvAddS	add2	Addition zweier Bilder
cvTransform	affine_image_transform	affine Bildtransformation
cvPerspectiveTransform	---	perspektivische Transformation
cvCalcPCA	---	P roincipal C omponent A nalysis
cvDFT	---	diskrete Fouriertransformation
cvDCT	---	diskrete Kosinustransformation
cvPutText	chprint	Textausgabe



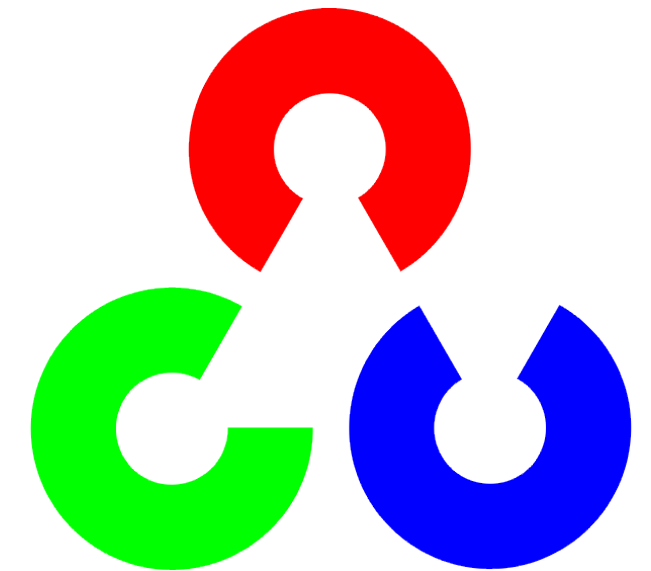
Wichtige OpenCV-Funktionen

OpenCV Funktion	VCLIB Funktion	Bemerkung
cvKMeans2	---	k-means Klassifikator
cvSobel	sobel	Sobeloperator
cvLaplace	laplace	Laplaceoperator
cvCanny	edge	Canny Kantenoperator
cvCornerHarris	---	Harris Corner-Detektor
cvLogPolar	---	Log-Polartransformation
cvErode	maxMxN	Erosion mit bel. Strukturelement
cvDilate	minMxN	Dilatation mit bel. Strukturelement
cvMorphologyEx	---	spezielle Morphologiefunktion
cvSmooth	gauss	Tiefpassfilter
cvFilter2D	ff3	beliebiger 2D-Filter
cvCvtColor	RGB_IHS, etc.	Farbraumtransformation
cvThreshold	binarize	Binarisierung mit festem Schwellwert
cvAdaptiveThreshold	maskx	Binarisierung mit var. Schwellw.
cvPyrDown	pyramid	Bildpyramiden
cvPyrUp	zoomup	Zoomfunktion
cvFloodFill	---	Bildkomponente fluten
cvFindContours	contour8	Konturbestimmung



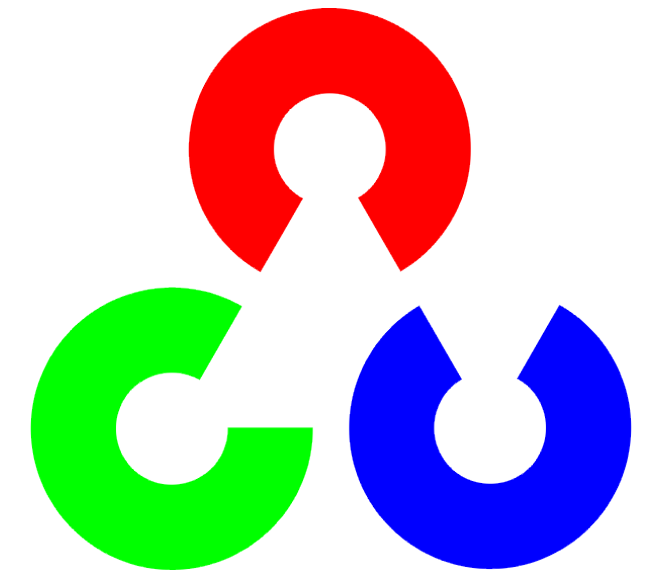
Wichtige OpenCV-Funktionen

OpenCV Funktion	VCLIB Funktion	Bemerkung
cvPyrMeanShift	---	MeanShift Algorithmus (Gesichtserkennung)
cvWatershed	---	Watershed Segmentierung
cvMoments	rlc_moments	Berechnung der Momente
cvGetHuMoments	mom_calc_phi1/2	Berechnet 7 Hu-Invarianten
cvHoughLines2	FindHoughLine	Linienfinder nach Hough
cvHoughCircles	FindHoughCircle	Kreisfinder nach Hough
cvDistTransform	---	Distanztransformation
cvHistogram	histo, color_histo	Histogramm
cvCalcBackProject	color_classify	Rückprojektion
cvMatchTemplate	vc_corr2	Mustererkennung, drehlageabhängig
cvFitEllipse2	bestcircle	Kreis- / Ellipsenausgleich
cvFitLine	bestline	Geradenausgleich
cvConvexHull2	---	konvexe Hülle
cvCalcOpticalFlowLK	---	Objektverfolgung (Tracking)
cvHaarDetectObjects	---	Objekterkennung nach Haar
cvFindHomography	---	Berechnung d. perspektivischen Transformation
cvCalibrateCamera2	---	Berechnung von intrinsischen / extrinsische Parametern



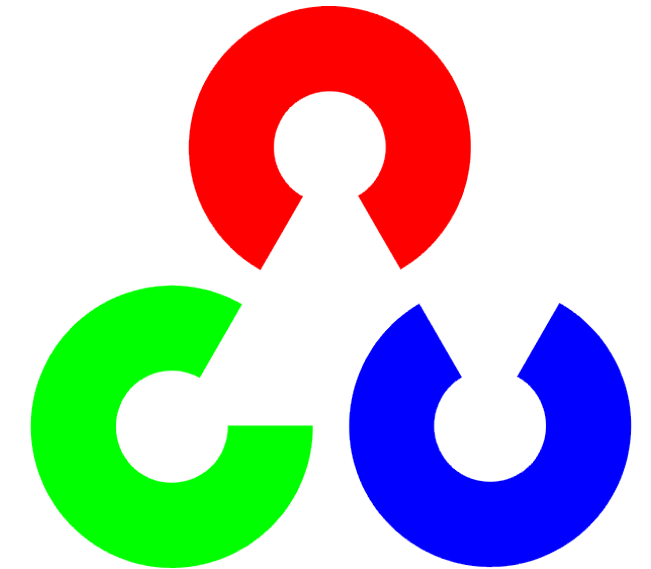
Wichtige OpenCV-Funktionen

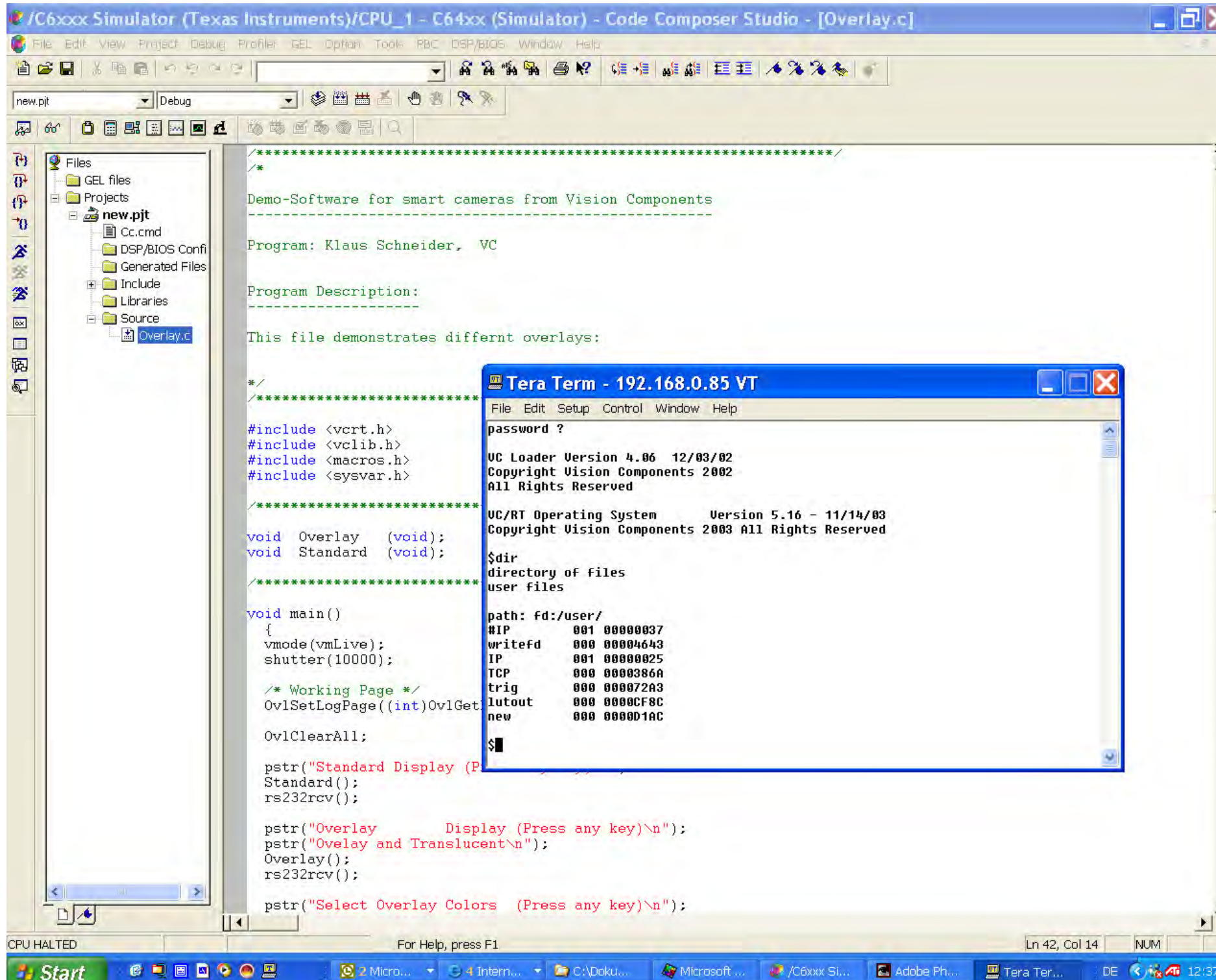
OpenCV Funktion	VCLIB Funktion	Bemerkung
cvUndistort2	---	Ausgleich der Linsenverzerrungen
cvFindFundamentalMat	---	Berechnung der Fundamentalmatrix aus korrespondierenden Punkten zweier Bilder
cvComputeCorrespond Epilines	---	Berechnung der korrespondierende Epilinen in einem Stereo-Paar
cvConvertPoints Homogenous	---	Bildpunktkonvertierung in/aus homogenen Bildkoordinaten



Übersicht über die Demoprogramme für OpenCV

vccolor.c	Farbraumtransformation HSV, Farbhistogramm
vcdist.c	Distanztransformation
vcedge.c	Canny Kantenerkennung
vcface.c	Gesichtserkennung mit Haartransformation
vcharris.c	Harris Corner-Detektor
vchist.c	Helligkeits- und Kontrastregelung mit Histogramm
vchough.c	Hough-Transformation für Linien
vckm2.c	kMeans2 Klassifikator
vclk.c	Lukas-Kanade Tracker mit Optical Flow
vcmatch.c	Pattern Matching, drehlageabhängig
vcshift.c	CAMSHIFT Demo mit Farbhistogramm, Rückprojektion





```

/C6xxx Simulator (Texas Instruments)/CPU_1 - C64xx (Simulator) - Code Composer Studio - [Overlay.c]
File Edit View Project Debug Profiler GEL Option Tools PBC DSP/BIOS Window Help
new.pjt Debug
Files
  GEL files
  Projects
  new.pjt
    Cc.cmd
    DSP/BIOS Confi
    Generated Files
    Include
    Libraries
    Source
      Overlay.c

/*****
/*
Demo-Software for smart cameras from Vision Components
-----
Program: Klaus Schneider, VC
Program Description:
-----
This file demonstrates differnt overlays:
*/
/*****
#include <vcrt.h>
#include <vclib.h>
#include <macros.h>
#include <sysvar.h>
/*****
void Overlay (void);
void Standard (void);
/*****
void main()
{
  vmode(vmLive);
  shutter(10000);
  /* Working Page */
  OvlSetLogPage((int)OvlGet

  OvlClearAll;

  pstr("Standard Display (P
  Standard();
  rs232rcv();

  pstr("Overlay      Display (Press any key)\n");
  pstr("Ovelay and Translucent\n");
  Overlay();
  rs232rcv();

  pstr("Select Overlay Colors  (Press any key)\n");
}

```

```

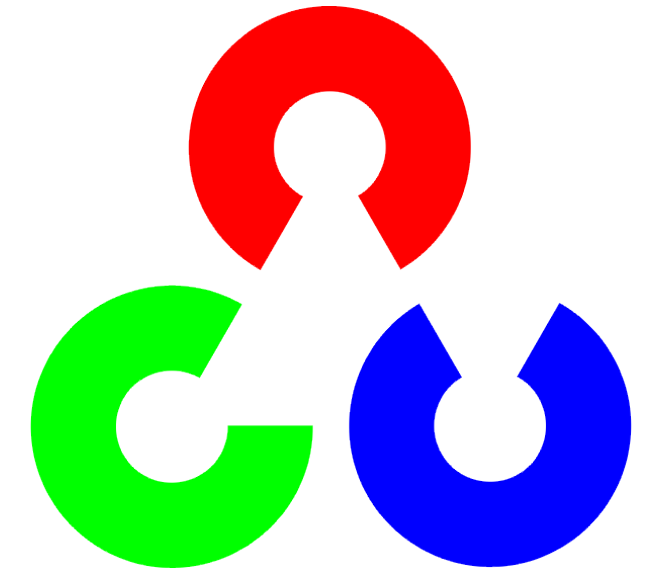
Tera Term - 192.168.0.85 VT
File Edit Setup Control Window Help
password ?
VC Loader Version 4.06 12/03/02
Copyright Vision Components 2002
All Rights Reserved
VC/RT Operating System Version 5.16 - 11/14/03
Copyright Vision Components 2003 All Rights Reserved
$dir
directory of files
user files
path: fd:/user/
#IP      001 00000037
writefd  000 00004643
IP       001 00000025
TCP      000 0000386A
trig     000 000072A3
lutout   000 0000CF8C
new      000 0000D1AC
$

```

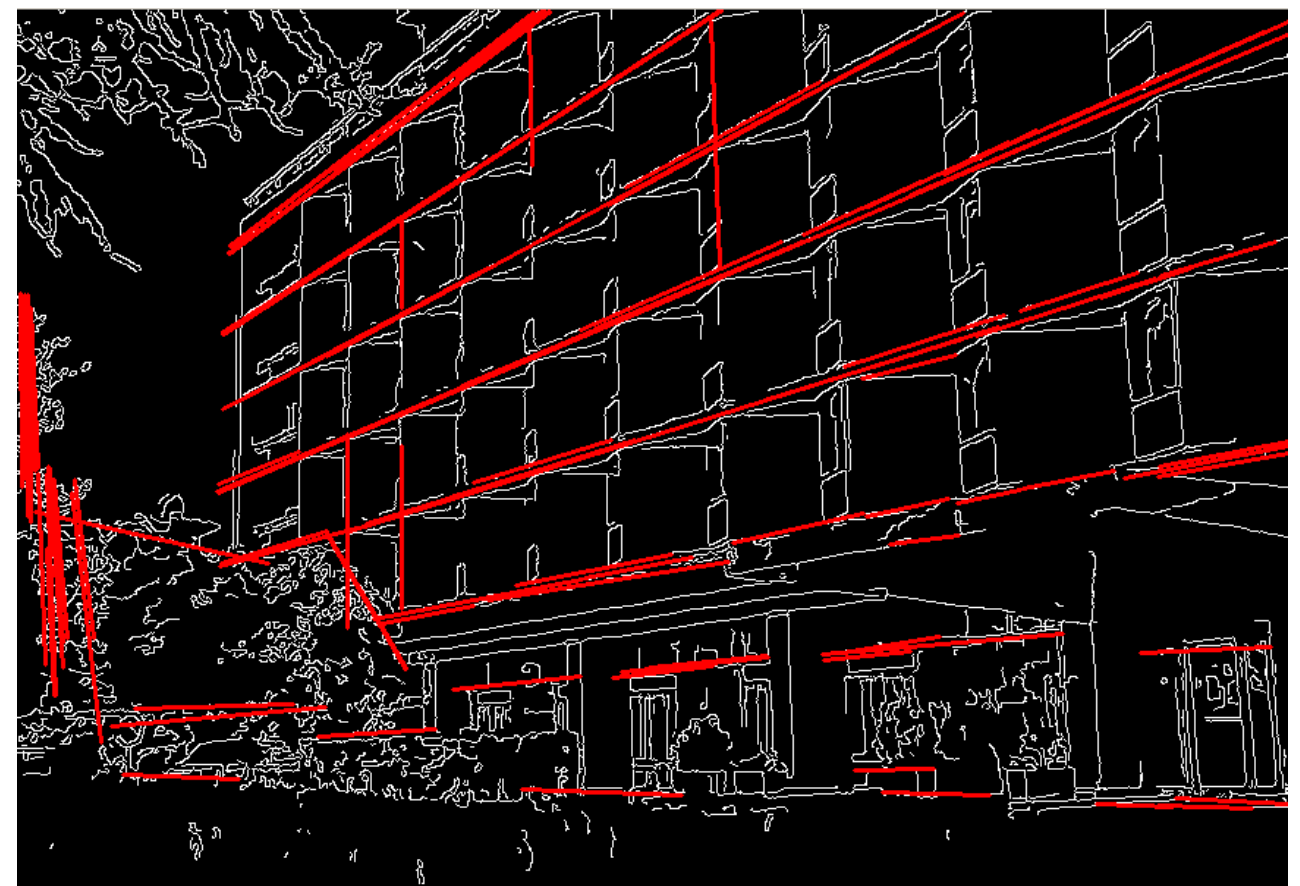


Typische Anwendungen von OpenCV

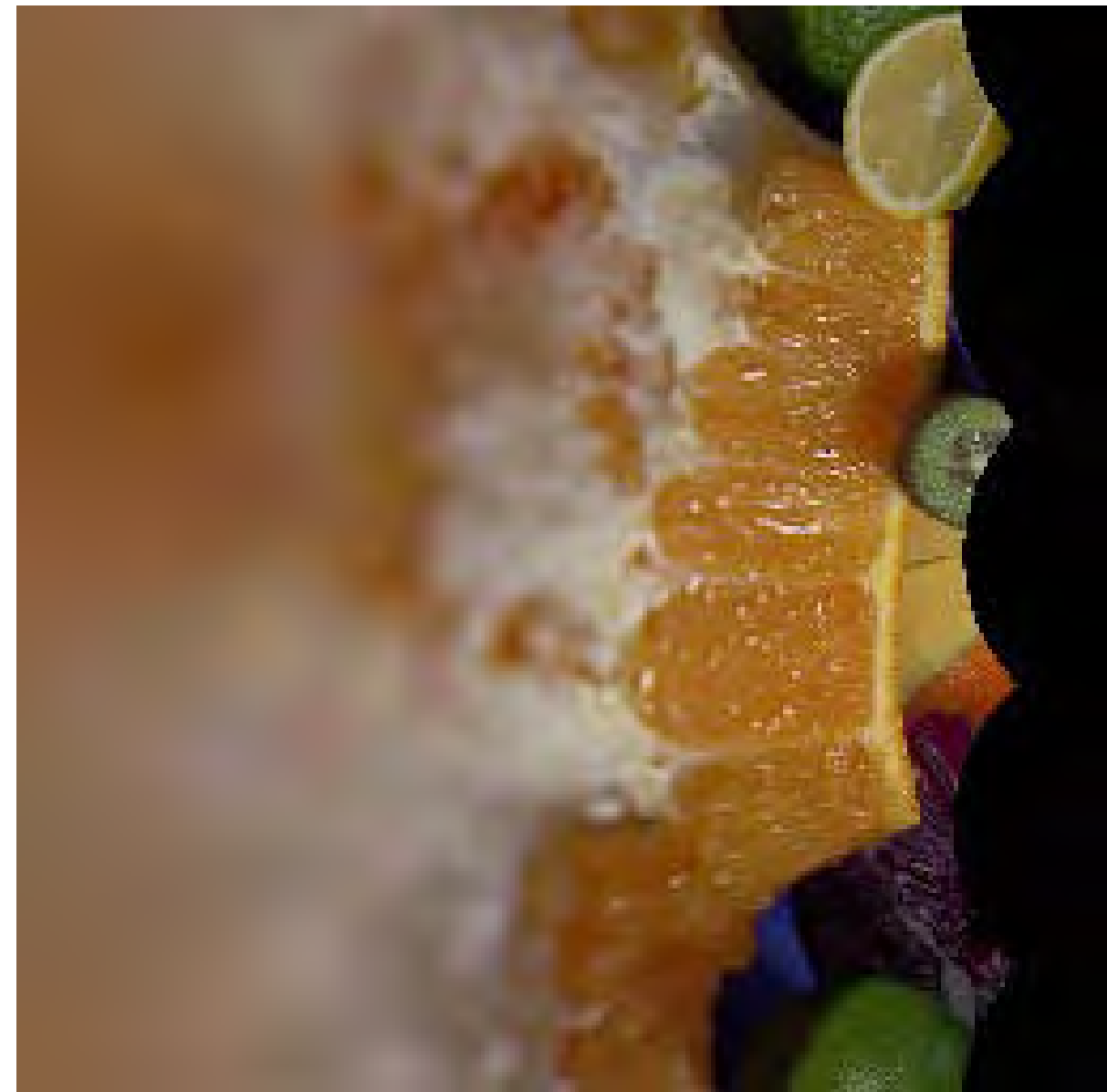
- Gesichtserkennung und –Verfolgung (vcshift.c)
- Gesichtserkennung (vcface.c)
- Verkehrsüberwachung, Optical Flow, Lukas-Kanade-Tracker (vclk.c)



Houghtransformation



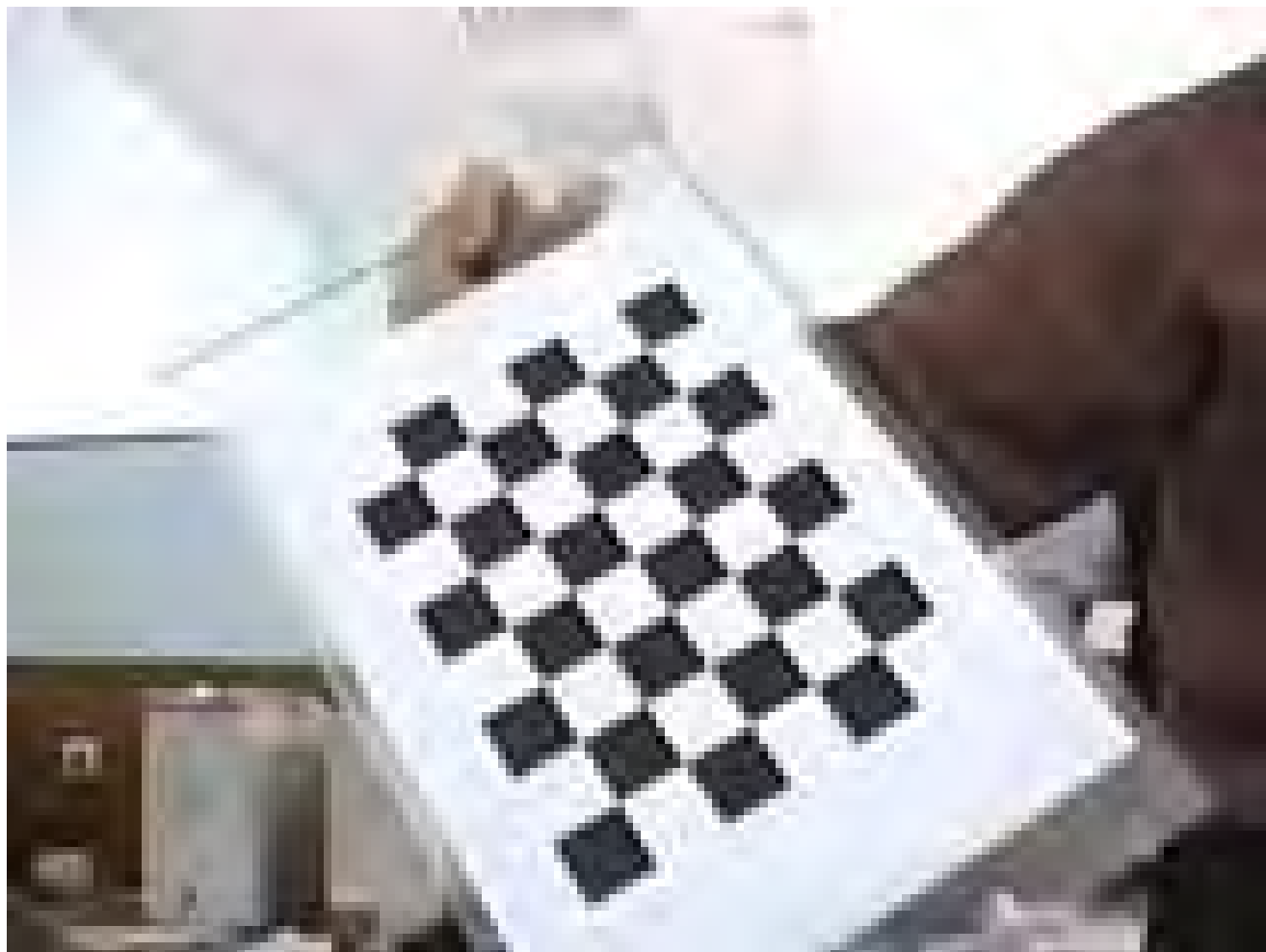
Log-Polartransformation



Camshift-Algorithmus



Kamerakalibrierung



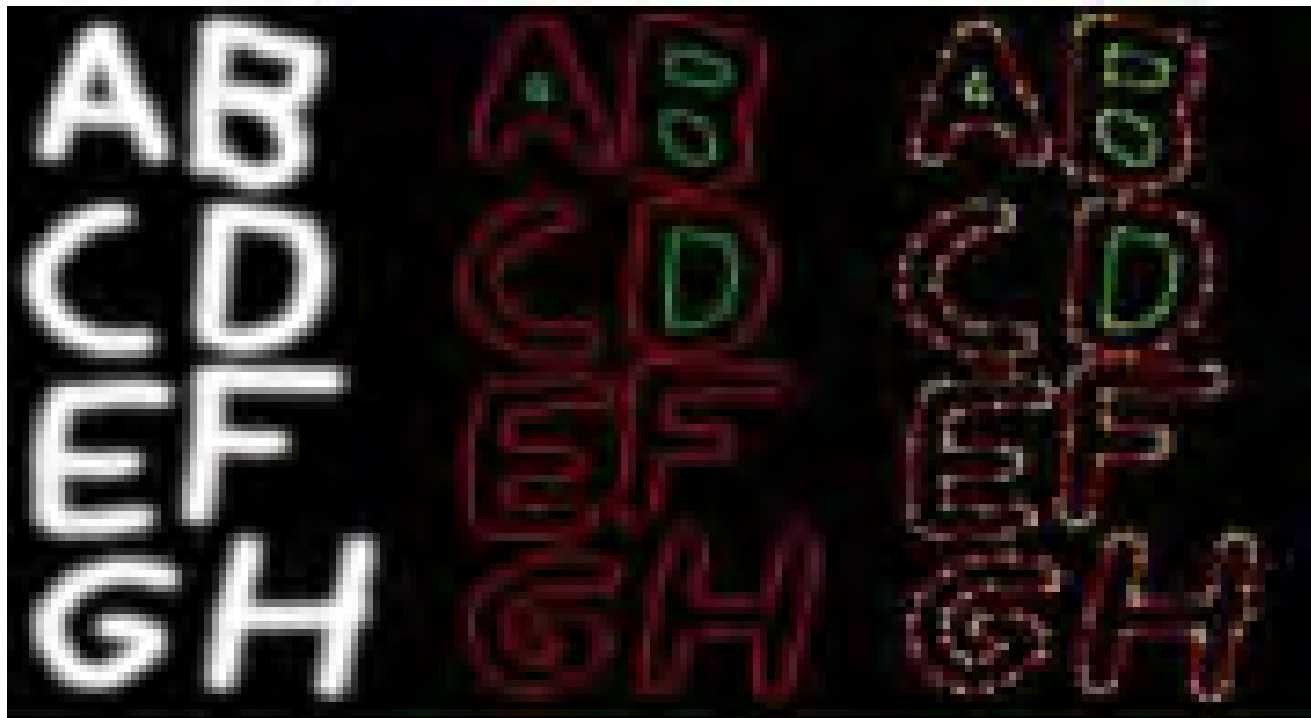
Stereokorrespondenz



Morphologie



Formanalyse mit OpenCV



Vielen Dank für Ihre Aufmerksamkeit!

Fragen?

Besuchen Sie uns auf der Vision in Halle 4, Stand 4D31 !