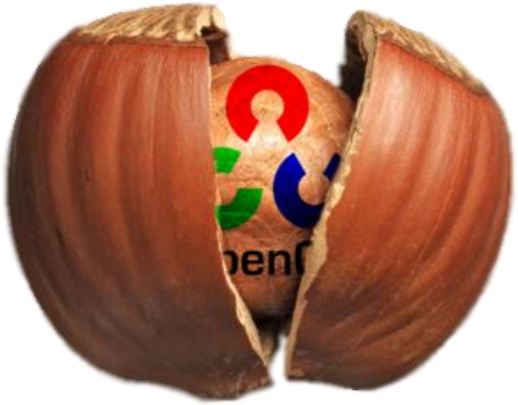
RWTH AACHEN UNIVERSITY

OpenCV in a Nutshell

Seminarvortrag im Studiengang Scientific Programming



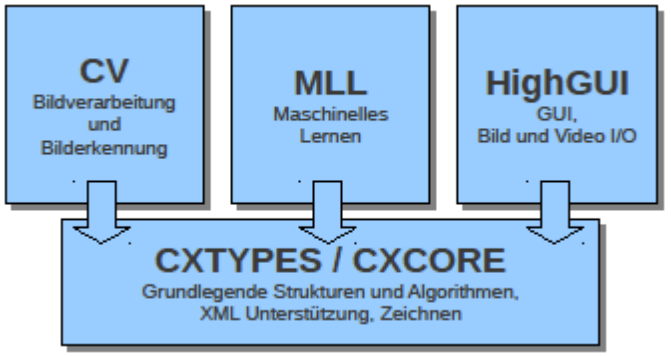
Inhalt

- Was ist OpenCV?
- Beispielcode
- Datenstruktur: IplImage
- Bildmanipulationen / Filter
- Bildtransformationen
- Bildanalyse
- Ausblick

- **Was ist OpenCV?**
  - OpenCV steht für: open source computer vision library
  - BSD-Lizenz
  - Frei für die private, akademische und kommerzielle Benutzung
- **Was Kann OpenCV?**
  - Bibliothek zur Bildverarbeitung
  - ca. 500 Funktionen
  - Portabel (write once, compile everywhere)
- **Warum OpenCV?**
  - Versuch der Standardisierung der Bildverarbeitung
  - Weite Verbreitung bei Firmen (Intel, Microsoft, Sony, Google, …) und Forschungseinrichtungen (Stanford, MIT, Cambridge, …)

Headerdateien

- **cv.h**
  - Bildverarbeitungsalgotithmen
- **highgui.h**
  - Ein- und Ausgabe
- **cxtxpes.h, cxcore.h**
  - Grundlegende Datentypen
  - Bereits in den anderen Headern eingebunden
- **ml.h**
- **cvaux.h**



Beispielcode

```
int main(int argc, char** argv){
    CvCapture *cap = cvCreateCameraCapture(0);
    IplImage* img = cvQueryFrame(cap);

    int key = -1;
    while(key < 0){
        key = processImage(img);
        img = cvQueryFrame(cap);
    }
    cvReleaseCapture(&cap);
}
```

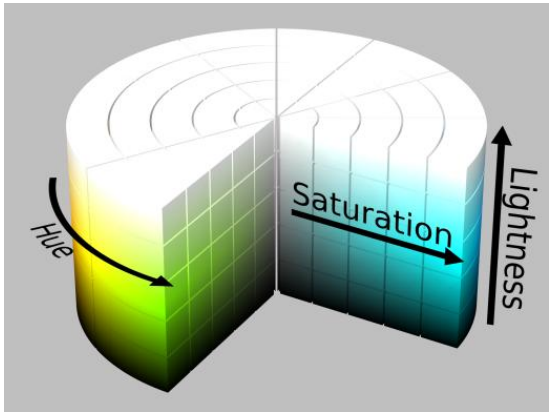
Beispielcode (2)

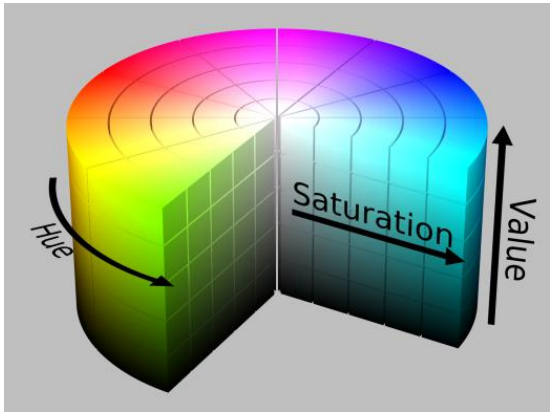
```
int processImage(IplImage* img)
{
  int key = cvWaitKey(50);
  CvRect center =  cvRect(3*img->width/8, 3*img->height/8, img->width/8, img->height/8);
  CvRect target = center;
  static IplImage *tpl = cvCreateImage(cvSize(center.width, center.height), img->depth, img->nChannels);
  static IplImage *dst = cvCreateImage(cvSize(img->width-tpl->width+1, img->height-tpl->height+1), IPL_DEPTH_32F, 1);

  switch(key){
  case -1: break;
  case 27:
     cvReleaseImage(&tpl);
     cvReleaseImage(&dst);
     return 1;
  case 32:
     cvSetImageROI(img, center);
     cvCopy(img, tpl);
     cvResetImageROI(img);
     break;
  default:
     printf("KeyPressed %d \n", key);
     break;
}

  cvMatchTemplate(img, tpl, dst, CV_TM_SQDIFF_NORMED);
```

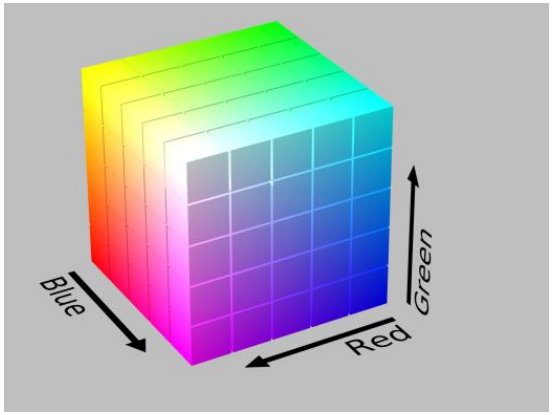
Beispielcode (3)

```
  CvPoint min_loc, max_loc;
  double min_val, max_val;
  cvMinMaxLoc(dst, &min_val, &max_val, &min_loc, &max_loc);
  target.x = min_loc.x;
  target.y = min_loc.y;

  if(min_val < 0.1){
    cvSetImageROI(img, target);
    cvAddWeighted(img, 0.25, tpl, 0.75, 1.0, tpl);
    cvResetImageROI(img);
  }else{
    cvSetImageROI(img, target);
    cvAddWeighted(img, 0.05, tpl, 0.95, 1.0, tpl);
    cvResetImageROI(img);
  }

  cvRectangleR(img, target, CV_RGB(255,0,0), 1);
  cvRectangleR(img,center, CV_RGB(0,255,0), 1);
  cvShowImage("Input",img);
  cvShowImage("Output",dst);
  cvShowImage("Template", tpl);

  return -1;
}
```

- **Grundlegende Datenstruktur**
- **Kann Bildinformationen in verschiedenen Farbräumen speichern.**
  - Räume: RGB, HSL, HSV, Grauwerte, …
  - Datentypen: char (signed und unsigned), int und float
- **Wird intern als Matrix von Pixeln behandelt**
  - Standard Matrix-Operationen (Addieren, Multiplizieren, …) verfügbar
  - Pixel sind i.d.R. jedoch nicht im mathematischen sinne Skalar.







Quelle: Wikipedia, HSL and HSV / RGB color model

IplImage (2)

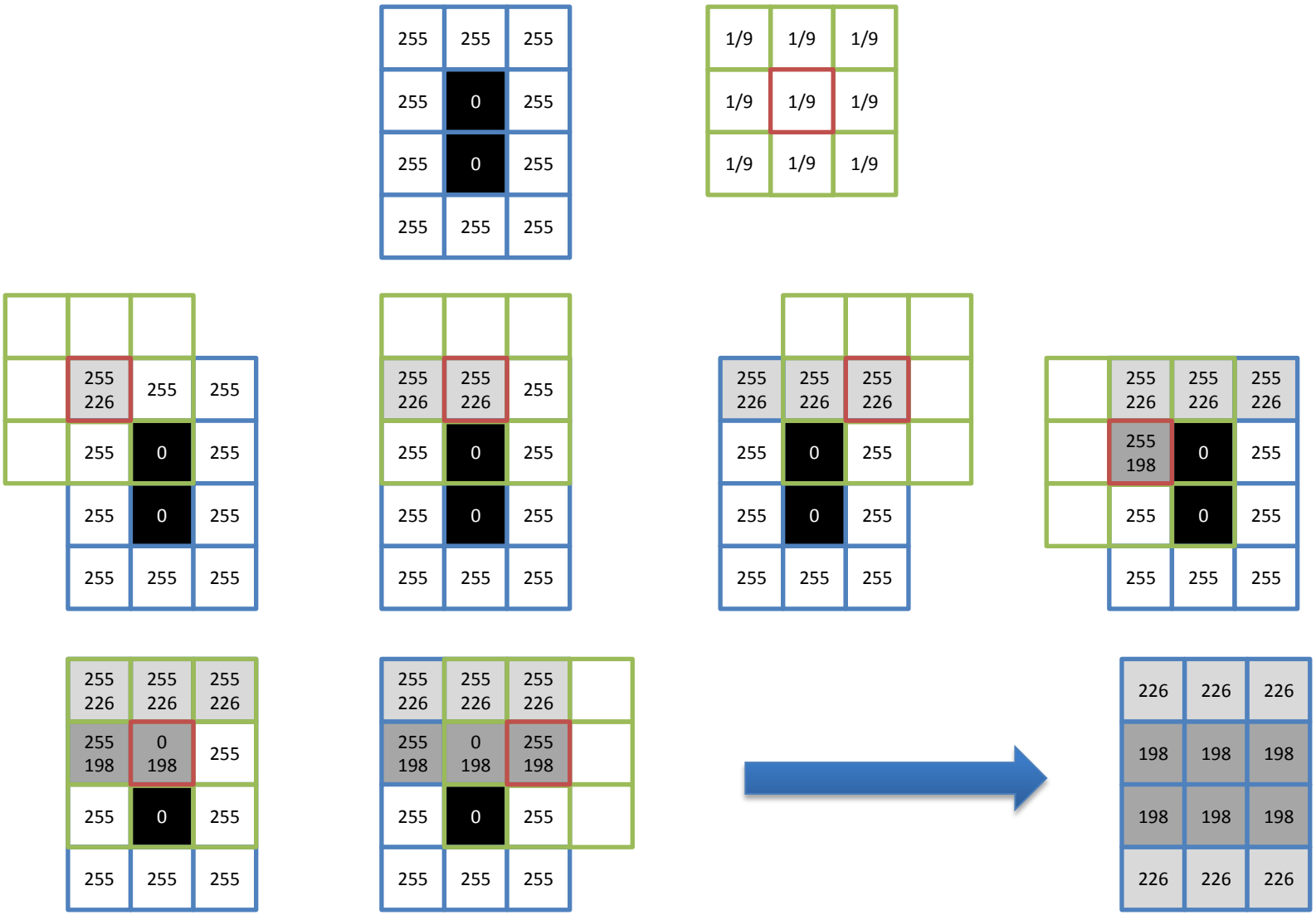
- **Initialisieren:**
  - `IplImage* cvLoadImage(const char*)`
  - `IplImage* cvCreateImage(CvSize, int, int)`
- **Darstellen**
  - `cvShowImage(const char*, IplImage*)`
  - `cvSaveImage(const char*, IplImage*)`
- **Speicher Freigeben**
  - `void cvReleaseImage(IplImage**)`

- **Abbildung von einem Ur-Bild I auf ein Ziel-Bild E**
  - Ein Pixel im Ziel-Bild wird aus einem Pixel und seiner Umgebung im Ur-Bild berechnet.
  - Die Umgebung nennt sich Kernel, die Position des Ur-Pixels im Kernel nennt sich Anker.

$$E(x, y) = F(\{I(i, j) \mid i, j \in K(x, y)\})$$

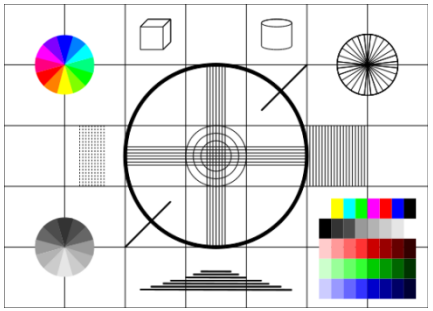
- **Aufgabe: Hervorheben von Merkmalen im Urbild**
  - Kanten hervorheben
  - Bildstörungen beseitigen
  - Flächen vereinheitlichen

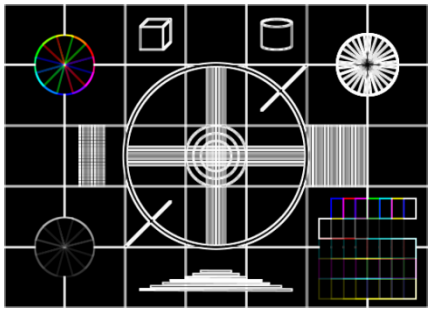
Bildmanipulationen / Filter (2)



Bildmanipulationen / Filter (3)

- **Kanten hervorheben:** `cvLaplace`





- **Bildstörungen verringern:** `cvSmooth`

- **Flächen vereinheitlichen:** `cvDelate`, `cvErode`

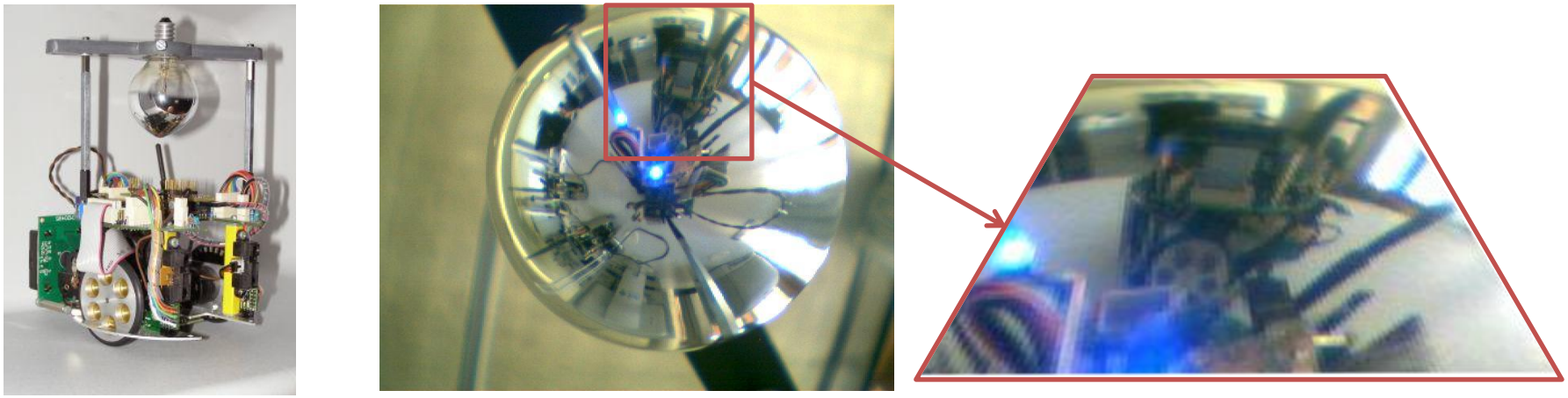
- **Globale Änderungen an den Eigenschafen des Bildes**
  - Farbraum
  - Breite
  - Höhe
  - Geometrie

- **Aufgabe: Verbesserung der Bildanalyse in Genauigkeit oder Zeitaufwand**
  - Halbierung der Breite und Höhe geht Quadratisch in die Fläche und damit in den Rechenaufwand ein.
  - Anpassung des Aufgenommenen Bildes an die Geometrie der Kameralinse
  - Andere Farbräume bieten andere Informationen

Bildtransformationen (2)

- **Farbraumumwandlung:** `cvCvtColor`

- **Kamerageometrie:** `cvAffineTransform`, `cvPerspectiveTransform`

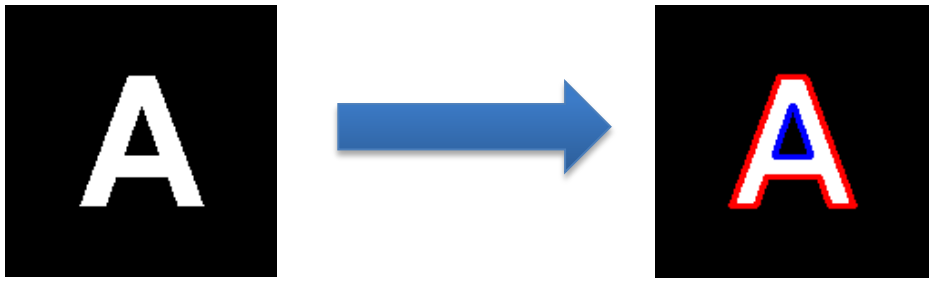


- **Finden von Punkten (Punktmengen) mit bestimmten Merkmalen**
  - Vergleich mit Vorlagen direkt (Pixelweise) oder über Formen bzw. Farben (Histogramme)
  - Finden zusammengehöriger Bildteile als Fläche oder Kontur

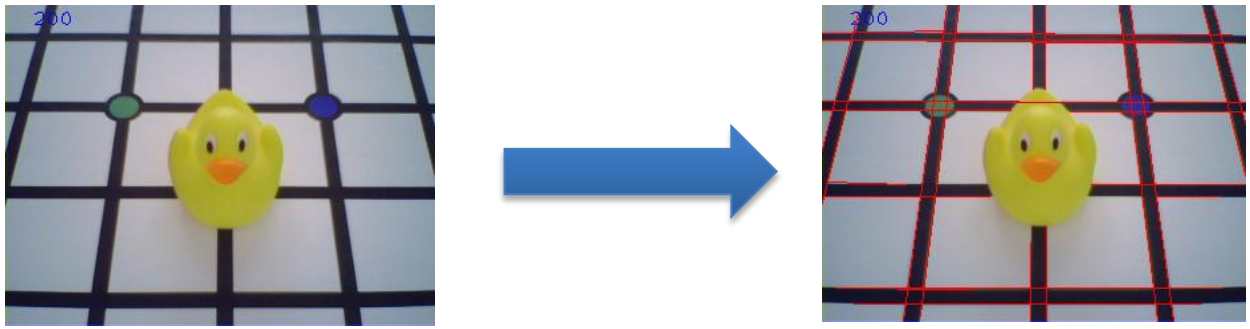
- **Aufgabe: Identifizieren von Objekten in Bildern**
  - Konturen
  - Linien
  - Segmente
  - Ähnlichkeit

Bildanalyse (2)

- **Konturen:** `cvFindContours`



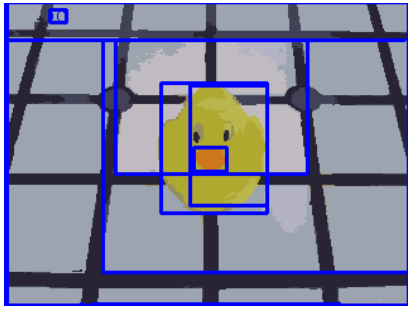
- **Linien:** `cvHoughLines2`



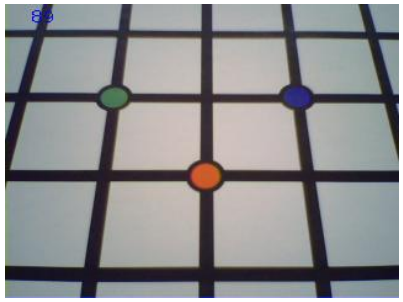
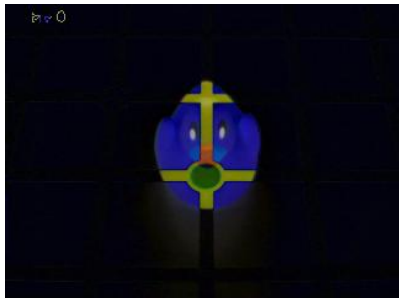
Segmente

- Farbbasiert: `cvPyramidSegmentation`



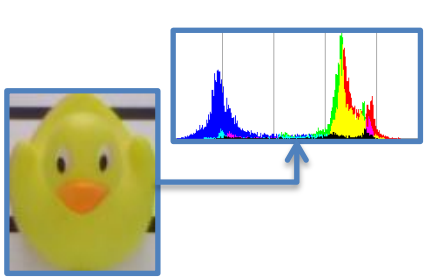


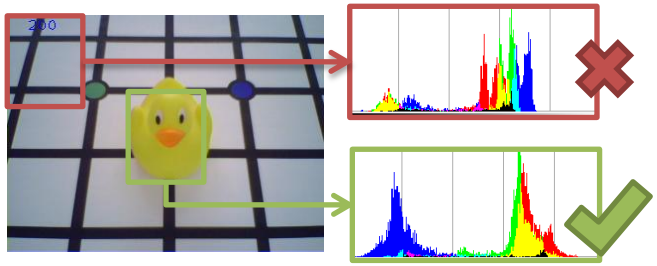
- Unterschiedsbasiert: `cvAbsDiff`

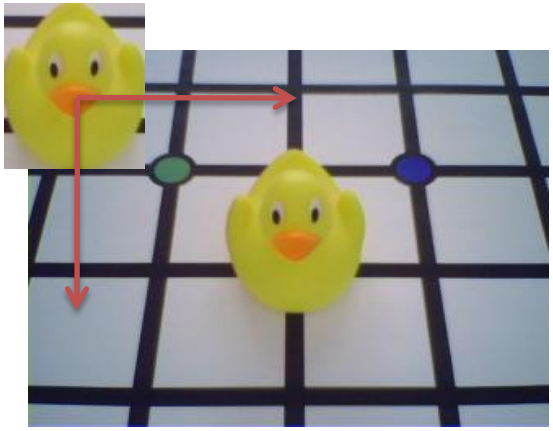
Ähnlichkeiten

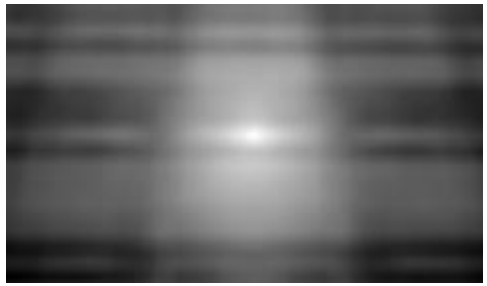
- Farbbasiert: `cvCompareHist`





- Pixelbasiert: `cvMatchTemplate`





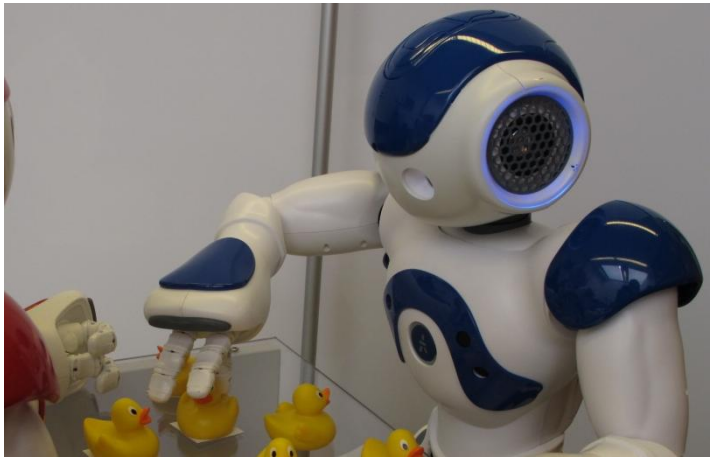
Ausblick

Augmented Reality

- Recognizr für Android

ARf, Virtual Pet für das iPhone

Robotik: NAO



Ausblick (2)

RWTH AACHEN UNIVERSITY

- **Bildverarbeitung**
  - Jähne, Prof. Dr. B.: Digitale Bildverarbeitung. 5. Auflage. Springer-Verlag Berlin, 2002. ISBN 3-540-51260-3
- **Bildverarbeitung mit OpenCV**
  - Bradski, Gary ; Kaehler, Adrian: Learning OpenCV. First. Sebastopool, CA : O'Reilly Media Inc., 2008 http://oreilly.com/catalog/9780596516130. ISBN 978-0-596-51613-0
- **OpenCV Referenz**
  - Willow Garage Inc. (Hrsg.): OpenCV Reference Manual v2.1. http://opencv.willowgarage.com/documentation/c/index.html
  - Bradski, Gary u. a. ; Willow Garage Inc. (Hrsg.): OpenCV Wiki. http://opencv.willowgarage.com/wiki/

Fragen?