AGRICAREERNET
NETZWERK FÜR AGRARKARRIEREN

ZERTIFIKATSKURS FÜR
BILDGEBENDE SYSTEME IN AGRAR- UND LEBENSMITTELTECHNIK

Software für Bildgebende Sensortechnologien

Andreas Linz

HOCHSCHULE OSNABRÜCK
UNIVERSITY OF APPLIED SCIENCES

AUFSTIEG DURCH BILDUNG
OFFENE HOCHSCHULEN

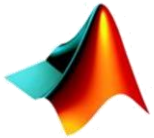
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

Übersicht:

- **Kommerzielle Software:**
  - MATLAB (MathWorks) → Image Processing Toolbox, Computer Vision Toolbox
  - LabVIEW (National Instruments) → NI Vision Builder for Automated Inspection
  - HALCON (MVTec) → Standardsoftware für industrielle Bildverarbeitung
- **Open Source Software:**
  - ImageJ
  - OpenCV
  - R → Programmiersprache (Interpreter) für Statistik und Bildverarbeitung
  - ROS → Middleware und Simulationsumgebung mit OpenCV, Gazebo oder V-Rep
- **Kommerzielle & Open Source Software:**
  - MATLAB (MathWorks) → Robotics Toolbox & ROS
  - Software Toolkits

Software Fokus

Statistik & Bildverarbeitung



MATLAB

ImageJ
Image Processing and Analysis in Java

Bildverarbeitung & Machine Learning

OpenCV

HALCON
a product of MVTec

Simulation

ROS

v-rep

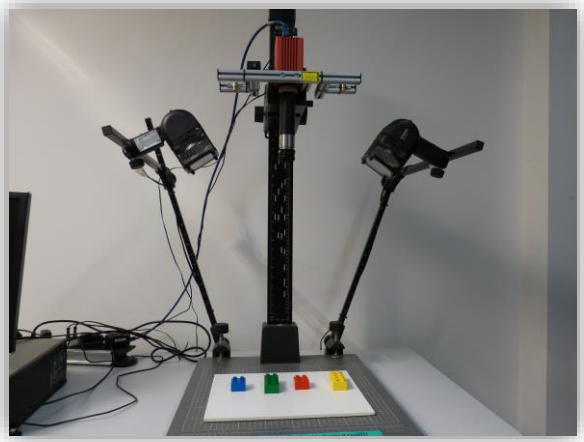
GAZEBO

MATLAB (MathWorks) → Image Processing Toolbox, Computer Vision Toolbox

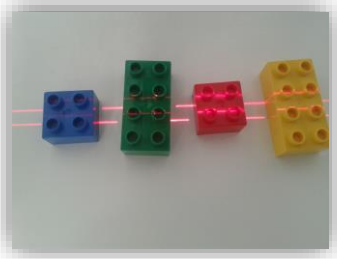
- MATLAB ist ein Programm der Firma MathWorks für numerische Berechnungen auf Grundlage von Matrizen.
- Ergebnisse können grafisch dargestellt werden z.B. in Diagrammen
- Das Grundprogramm kann mit verschiedenen Toolboxen erweitert werden
  - Image Acquisition Toolbox → Anbindung von Kameras
  - Image Processing Toolbox → Bildverarbeitungsalgorithmen
  - Computer Vision System Toolbox → Verarbeiten von Videosequenzen, 3D-Kameras Kalibrierung, …

MATLAB (MathWorks) → Image Processing Toolbox, Computer Vision Toolbox

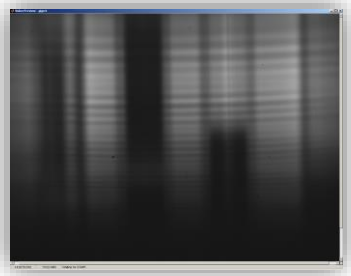
Beispiel: Spektralmessung (Hyperspectral) mit einer GigE-Kamera



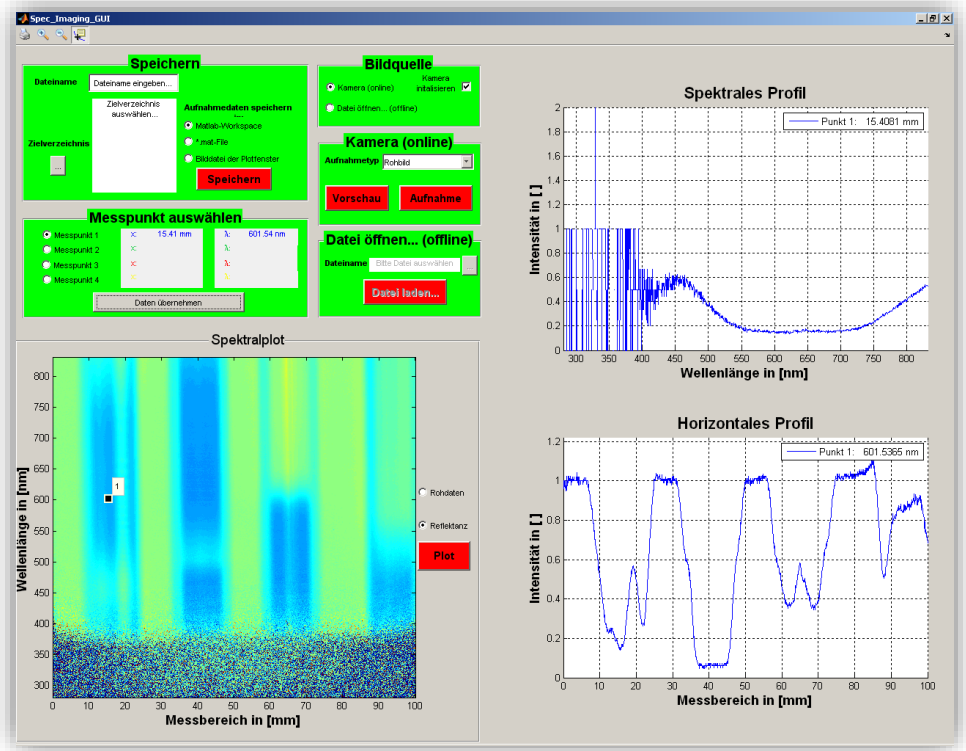
Versuchsaufbau: Hyperspectral Imaging



Messobjekte



Hyperspectral Image



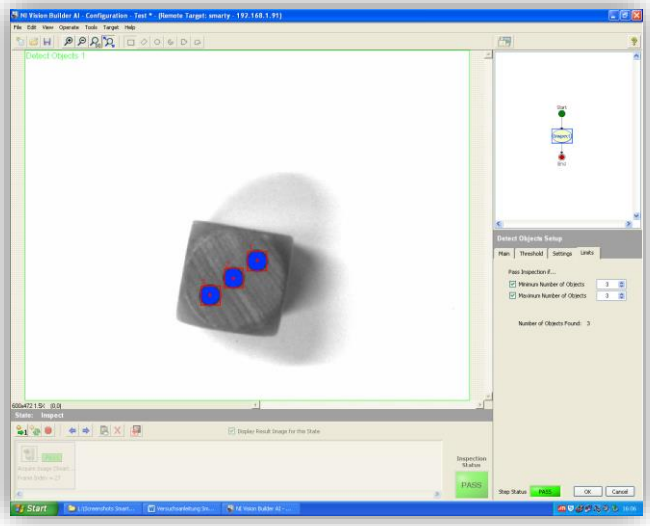
Aufnahme und Auswertung mit MATLAB

LabVIEW (National Instruments) → NI Vision Builder for Automated Inspection

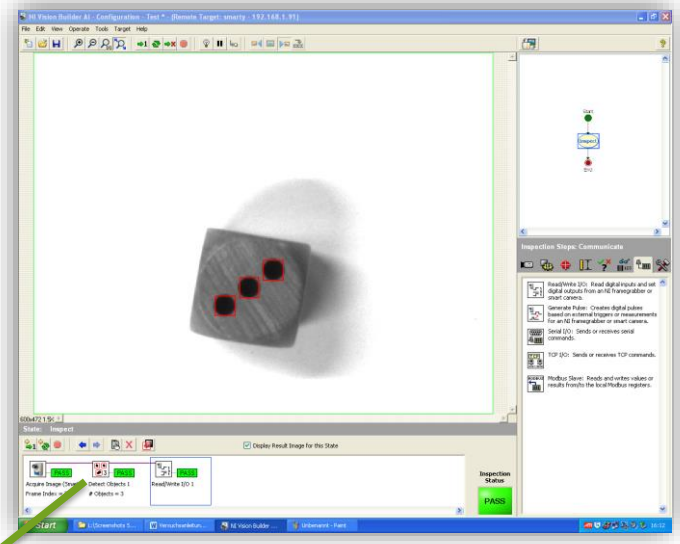
- Der „Vision Builder“ von National Instruments basiert auf einer für LabVIEW typischen Datenflussprogrammierung
- Es können Blöcke mit z.B. folgenden Funktionen eingefügt werden:
  - *Acquire Image Block* → Bilderfassung von einer Kamera oder von der Festplatte
  - *Detect Objects* → Objekterkennung
  - *Threshold* → Definieren eines Schwellwerte (Grauwertes) zur Segmentierung von Objekten
  - …
- Die meisten Blöcke basieren auf Standard Bildverarbeitungs-algorithmen

LabVIEW (National Instruments) → NI Vision Builder for Automated Inspection

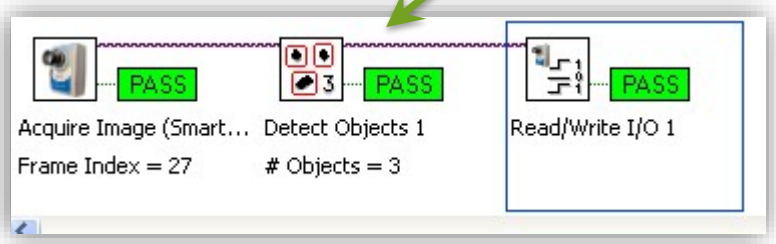
Beispiel: Smart Cam NI 1742



Threshold



Detect Objects



Flussdiagramm

HALCON a product of MVTec → Standardsoftware für industrielle Bildverarbeitung

- MVTec HALCON ist eine Sammlung von Bildverarbeitungs-algorithmen die mit einer Skriptsprache programmiert werden. Alternative werden eine GUI und eine API für verschiedene Programmiersprachen, wie z.B. C#, angeboten.

- Da es eine kommerzielle Software ist, gibt es regelmäßige Updates einen Support und aktualisierte Anpassung für viele verschiedene Plattformen. Als Beispiel kann HALCON mit der Embedded-Vision-Plattform auf einen Raspberry Pi portiert werden.

- Die Unterstützung von den meisten Kameratypen, Framegrabbern und Schnittstellentechnologien ist gewährleistet.

HALCON a product of MVTec & JAI (Multispektralkamera) → NDVI

Programmfenster - main () - Hauptthread: 1756

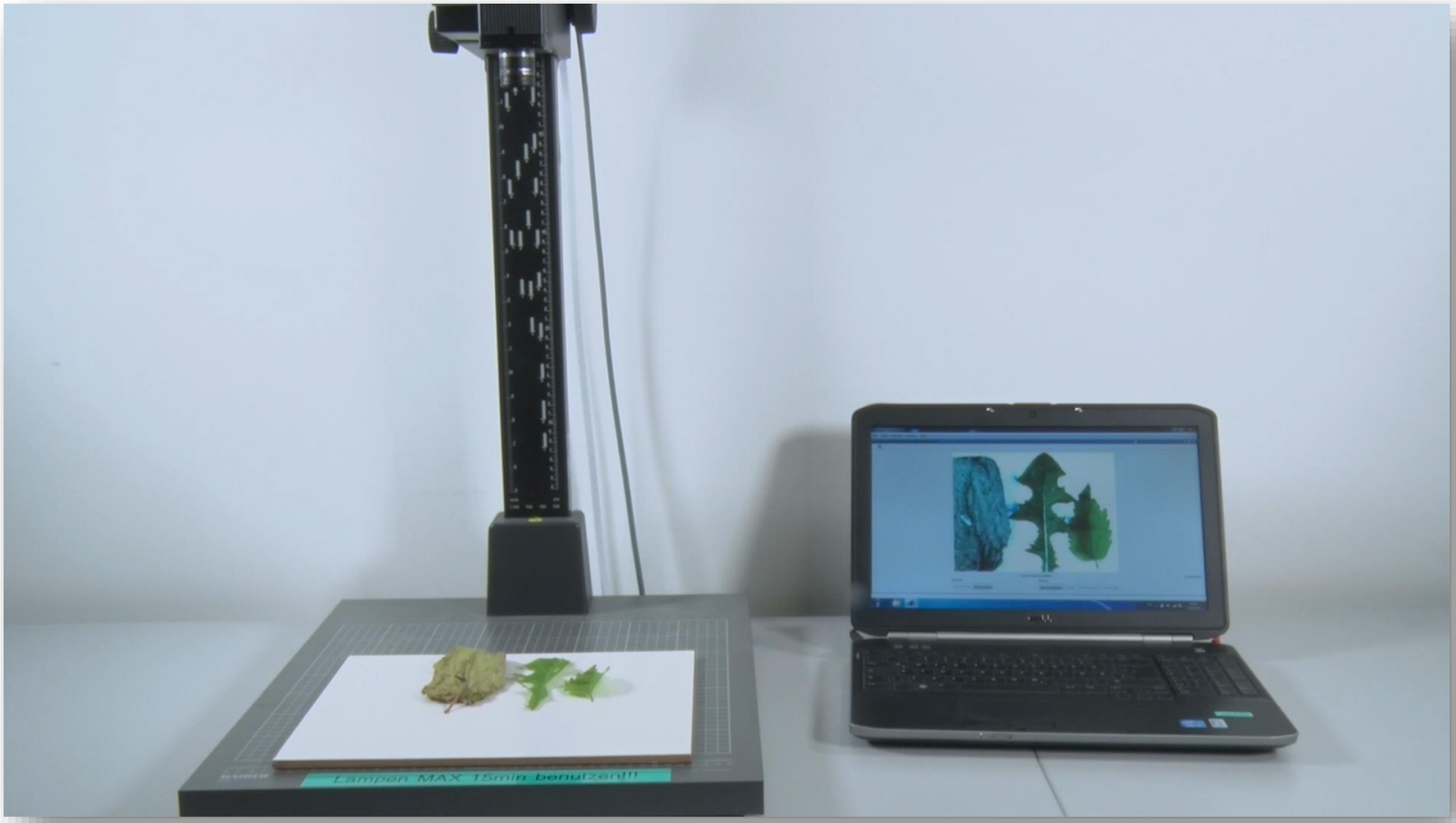
main (:::)

```
* Image Acquisition 01: Code generated by Image Acquisition 01
open_framegrabber ('GigEVision', 0, 0, 0, 0, 0, 0, 'default', -1, 'default', -1, 'false', 'default', '000cdf049429_JAILtdJa
grab_image_start (AcqHandle01, -1)

* Image Acquisition 03: Code generated by Image Acquisition 03
open_framegrabber ('GigEVision', 0, 0, 0, 0, 0, 0, 'default', -1, 'default', -1, 'false', 'default', '000cdf04a429_JAILtdJa
grab_image_start (AcqHandle03, -1)

* Bildfenster erzeugen

dev_open_window (0, 0, 1024, 768, 'white', WindowHandle)
dev_set_window (WindowHandle)
dev_update_window ('off')
Button := 1
while (Button ==1)
    * Bild von Kamera holen
    grab_image_async (Image_RGB, AcqHandle01, -1)
    grab_image_async (Image_NIR, AcqHandle03, -1)
    * Bild drehen
*    rotate_image(Image_RGB, Image_RGB, 180, 'constant')
*    rotate_image(Image_NIR, Image_NIR, 180, 'constant')
    * Teilbilder RGB erzeugen
    decompose3 (Image_RGB, Image_R, Image_G, Image_B)

    * Datentypen ändern wg. Genauigkeit
    convert_image_type(Image_R, Image_R_Real, 'real')

    convert_image_type(Image_NIR, Image_NIR_Real, 'real')
```

HALCON a product of MVTec & JAI (Multispektralkamera) → NDVI

```
    * rot + nir berechnen
    add_image(Image_NIR_Real, Image_R_Real, Image_NDVI_nenner, 1, 0)

    * nir-rot berechnen
    sub_image(Image_NIR_Real, Image_R_Real, Image_NDVI_zaehler, 1, 0)

    dev_update_window ('on')
    * division diff durch summe teilen
    div_image(Image_NDVI_zaehler, Image_NDVI_nenner, Image_NDVI, 128, 128)
    dev_update_window ('off')
    * Schwelle setzen
*    threshold(Image_NDVI, Image_CHLOPHY, 130, 255)
    * anzeigen

*    dev_display(Image_CHLOPHY)


    if (Button!=1)
        close_framegrabber (AcqHandle01)
        close_framegrabber (AcqHandle03)
    endif
endwhile
* Kameras schließen
close_framegrabber (AcqHandle01)
close_framegrabber (AcqHandle03)
```

HALCON a product of MVTec & JAI (Multispektralkamera) → NDVI



ImageJ
Image Processing and Analysis in Java

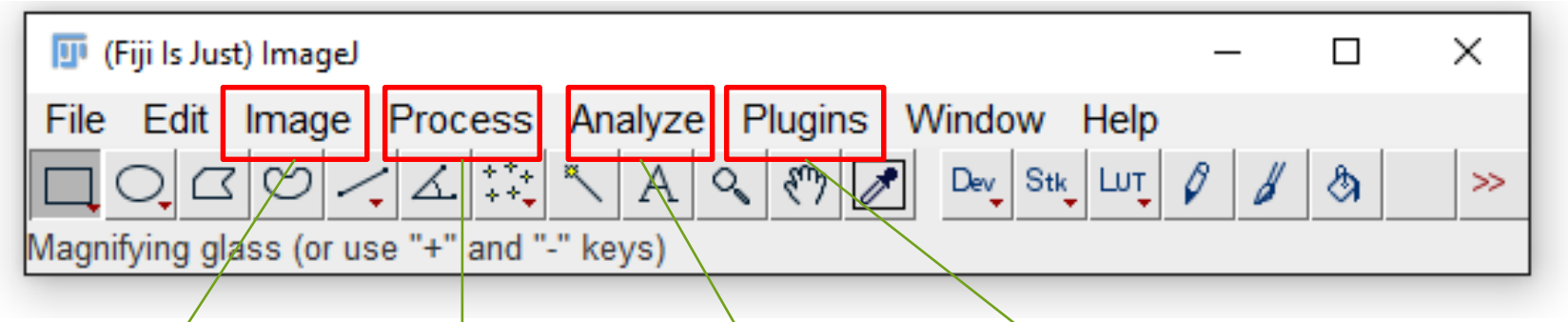
- ImageJ ist ein freies auf Java basierendes Bildverarbeitungsprogramm und dadurch weitgehend Plattformunabhängig. Die Möglichkeit Plug-ins zu schreiben und andere Plug-ins (über Hundert) zu nutzen, macht das Programm äußerst flexibel.

- Hauptnutzungsbereich ist die medizinische und wissenschaftliche Bildanalyse

- Beispiele für Plug-ins:

  - Twain Scan → Hierüber können Bilder direkt von einem Scanner oder von einer USB-Kamera eingelesen werden
  - 16-bit Histogram → Erzeugen von Histogrammen
  - HSB Stack Splitter → Transformation eines RGB-Bildes in den HSB-Farbraum mit Hue, Saturation und Brightness
  - …

ImageJ
Image Processing and Analysis in Java



Konvertierung und Modifizierung

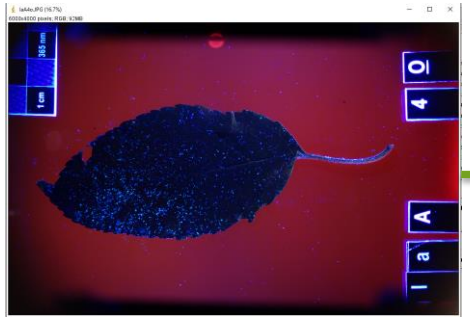
<u>Bildverarbeitung:</u>
- Punktoperationen
- Filter
- Arithmetische Operationen

<u>Bildanalyse:</u>
- Histogramm
- Statistische Messungen
- Particle Analyzer

<u>Erweiterungen:</u>
- Plugins
- Macros
- Find Commands

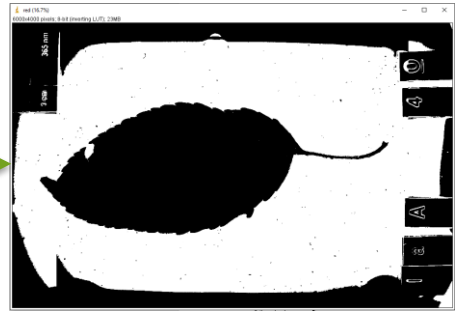
ImageJ
Image Processing and Analysis in Java

Beispiel: Blattparameter eines mit Pflanzenschutzmittel behandelten Blattes ermitteln

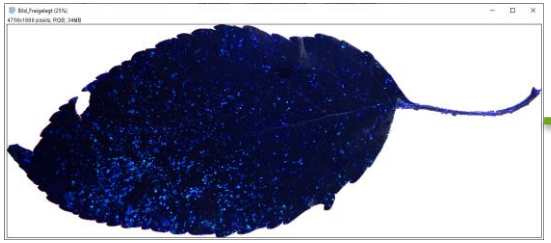


Originalbild mit fluoreszierenden Tropfen

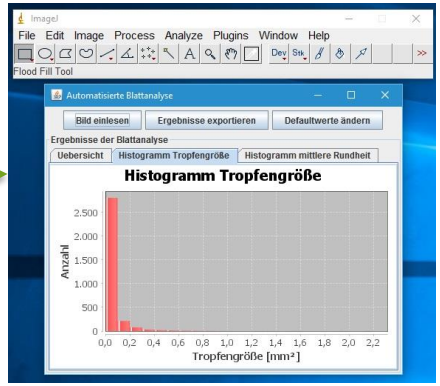
Grauwertbild



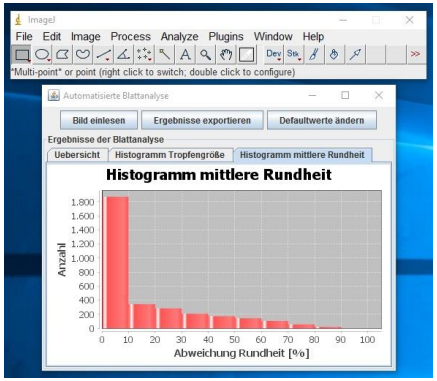
Binärbild (Schwellwert)



Segmentiertes Blatt



Histogramm mit Tropfengröße



Histogramm mittlere Rundheit

- OpenCV ist eine freie Programmbibliothek für Bildverarbeitung, die ursprünglich von Intel entwickelt wurde. Nachdem sie lange von der „*Willow Garage*“ gepflegt wurde ist sie indirekt über die Firma *Itseez* wieder zu Intel zurückgekehrt.

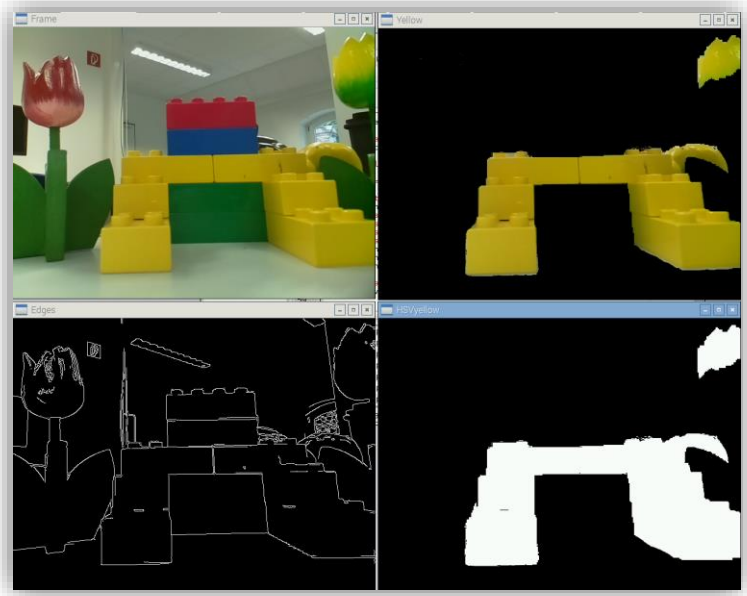
- Die Bibliothek ist für verschiedene Programmiersprachen verfügbar:
  - C & C++
  - Java
  - Python

- Und läuft unter vielen Betriebssystemen:
  - Windows
  - Linux
  - Mac OS
  - iOS & Android

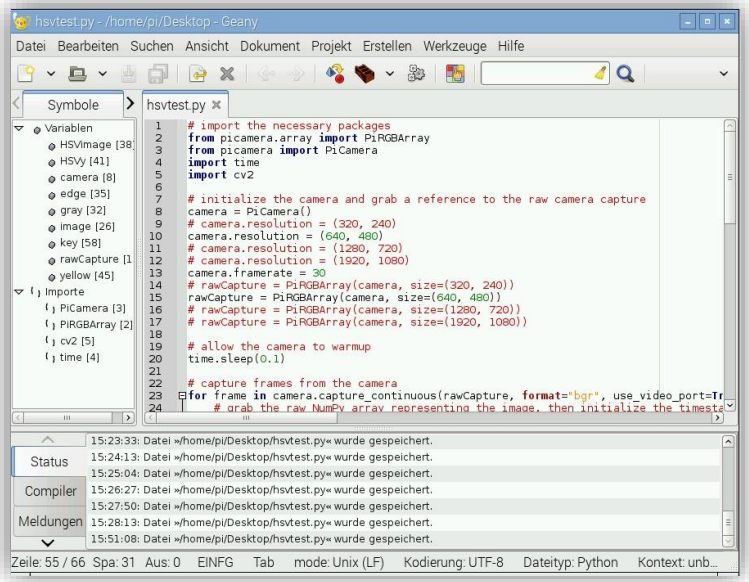
OpenCV → Raspberry Pi

Beispiel: Raspberry Pi als Smart Cam
Betriebssystem Linux,
Programmiersprache Python

Raspberry Pi mit Kamera



Farb- und Kantenselektierung (Canny)



Programmcode in Python

- R ist eine Programmiersprache die ursprünglich für statistische Berechnungen konzipiert wurde. Zwischenzeitlich gibt es viele online Pakete, die Bibliotheken für unterschiedlichste Anwendungen enthalten. So auch „EBImage“ von *Bioconductor*.

- EBImage enthält Standard Manipulationsalgorithmen für die Bildverarbeitung. Eigentlich konzipiert zur Segmentierung von Zellen, können die Funktionen auch auf andere Strukturen wie zum Beispiel Blätter angewendet werden.

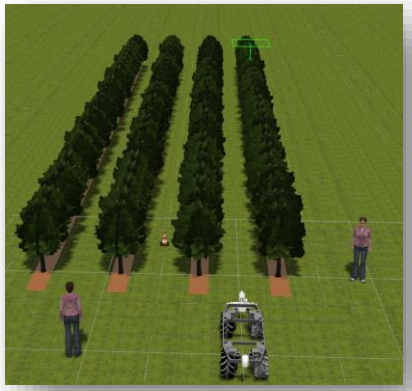
- Unter Anwendungen andere Bibliotheken wie z.B. Signalverarbeitung, Machine Learning und Statistical Modelling können Prozesse automatisiert werden um einen hohen „Throughput“ zu erreichen.

ROS Middleware und Simulationsumgebung mit OpenCV

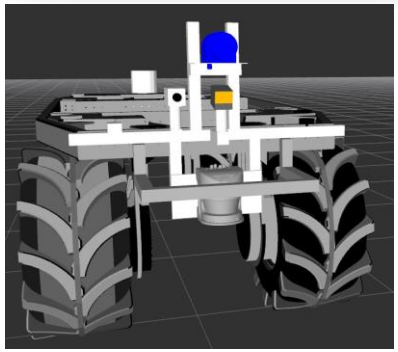
- ROS (Robot Operating System) ist ein Open Source Framework das als Grundfunktion einen Datenaustauch zwischen Prozessen auf Basis von Socket-Verbindungen zur Verfügung stellt.
- Die Datenpakete (Messages) zum Austausch sind standardisiert, können aber auch frei definiert werden.
- Die Software wurde lange von der „Willow Garage“ (siehe OpenCV) gepflegt, und hat somit OpenCV schon fest integriert. Erzeugte und verarbeitete Bilder werden über spezielle Messages (Imagetransport) ausgetauscht.
- Unterstütze Programmiersprachen sind C++ und Python.
- Des weiteren wird die 3D-Simulationsumgebung „Gazebo“ direkt unterstützt

ROS → Middleware und Simulationsumgebung mit OpenCV

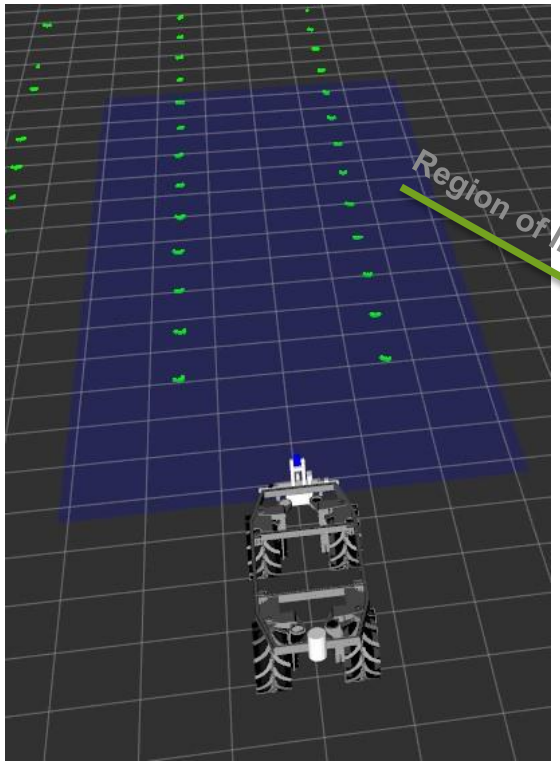
Beispiel: elWObot (Reihenerkennung) → Laserscanner & Hough-Transformation



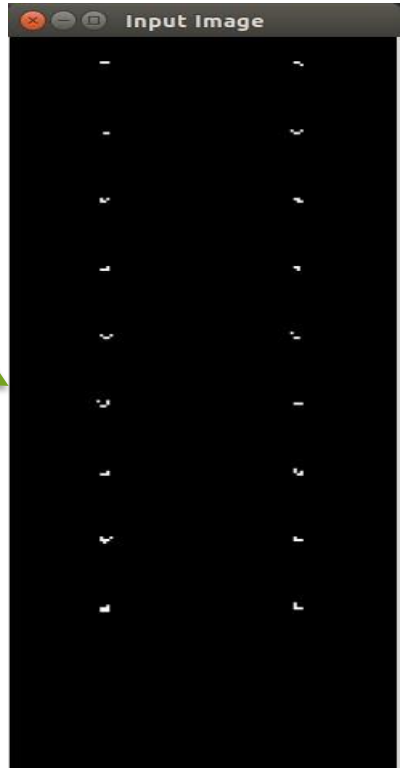
3D-Simulation Gazebo



elWObot mit Sensoren



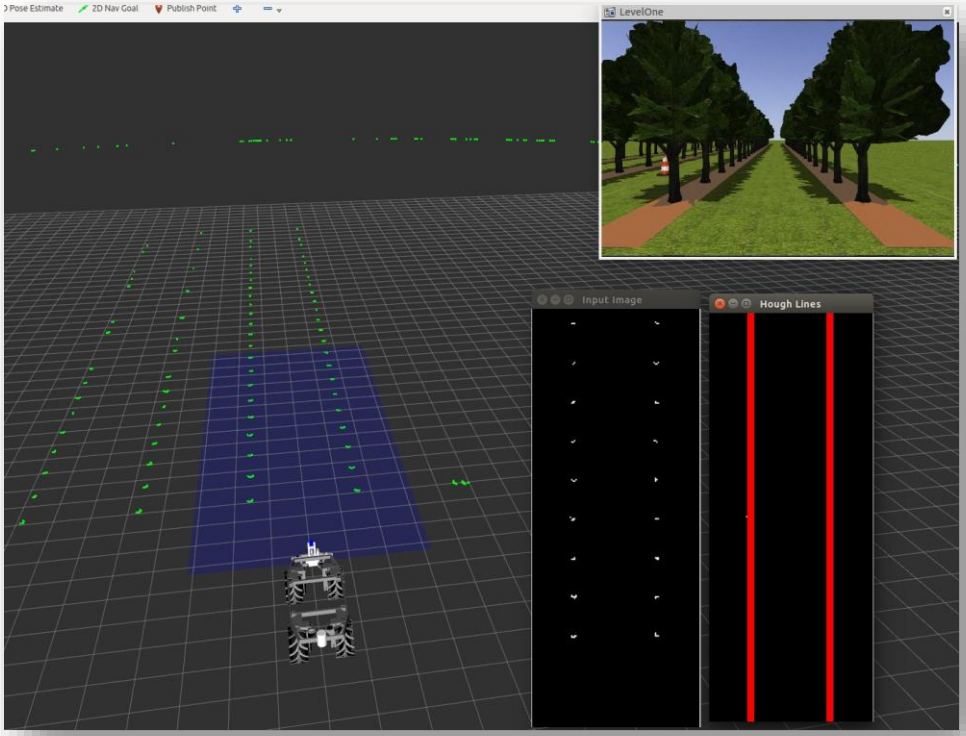
Sensordaten (Laserscanner) in Rviz



Binärbild → Scandaten ROI

ROS → Middleware und Simulationsumgebung mit OpenCV

Beispiel: elWObot (Reihenerkennung) → Laserscanner & Hough-Transformation

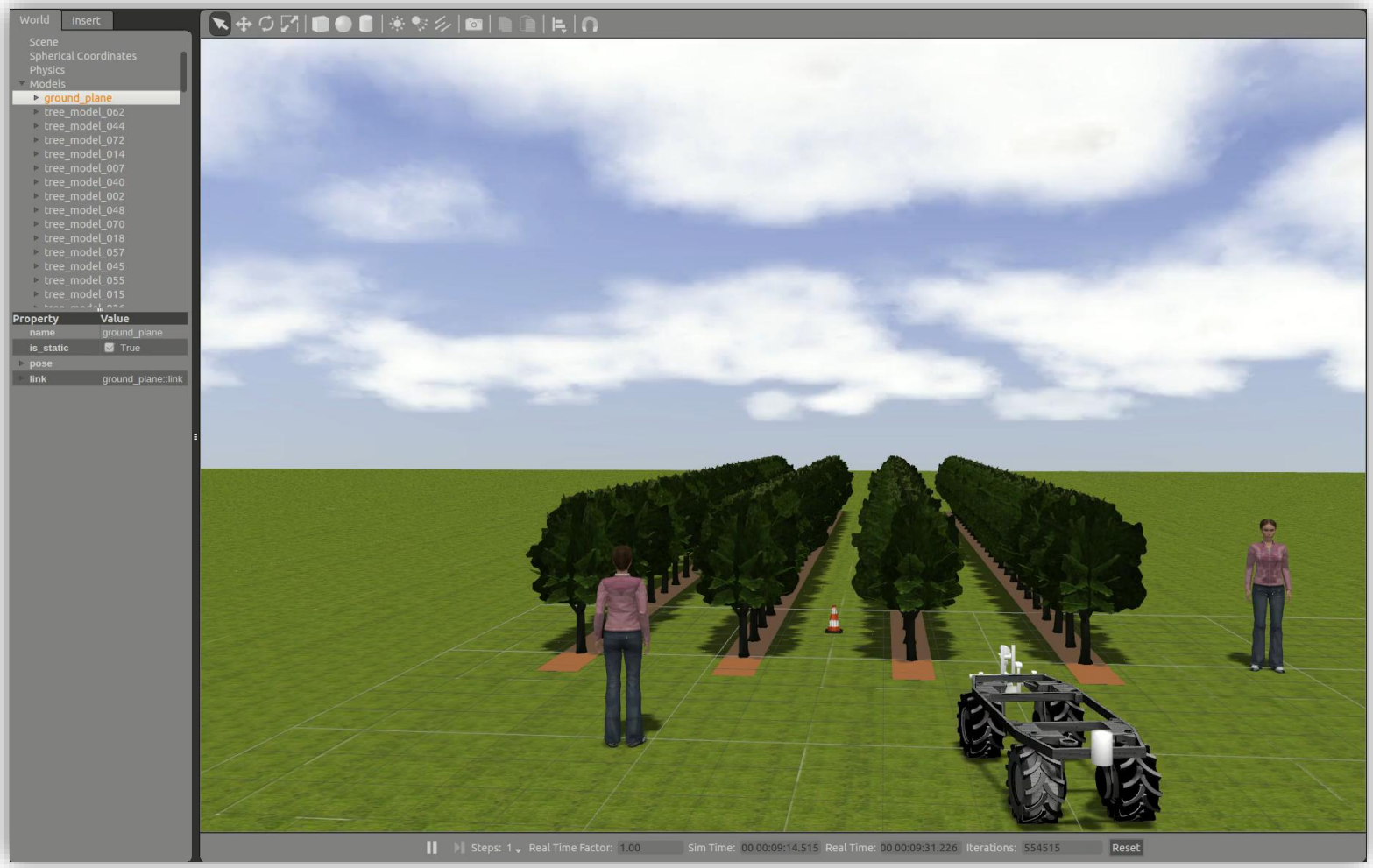


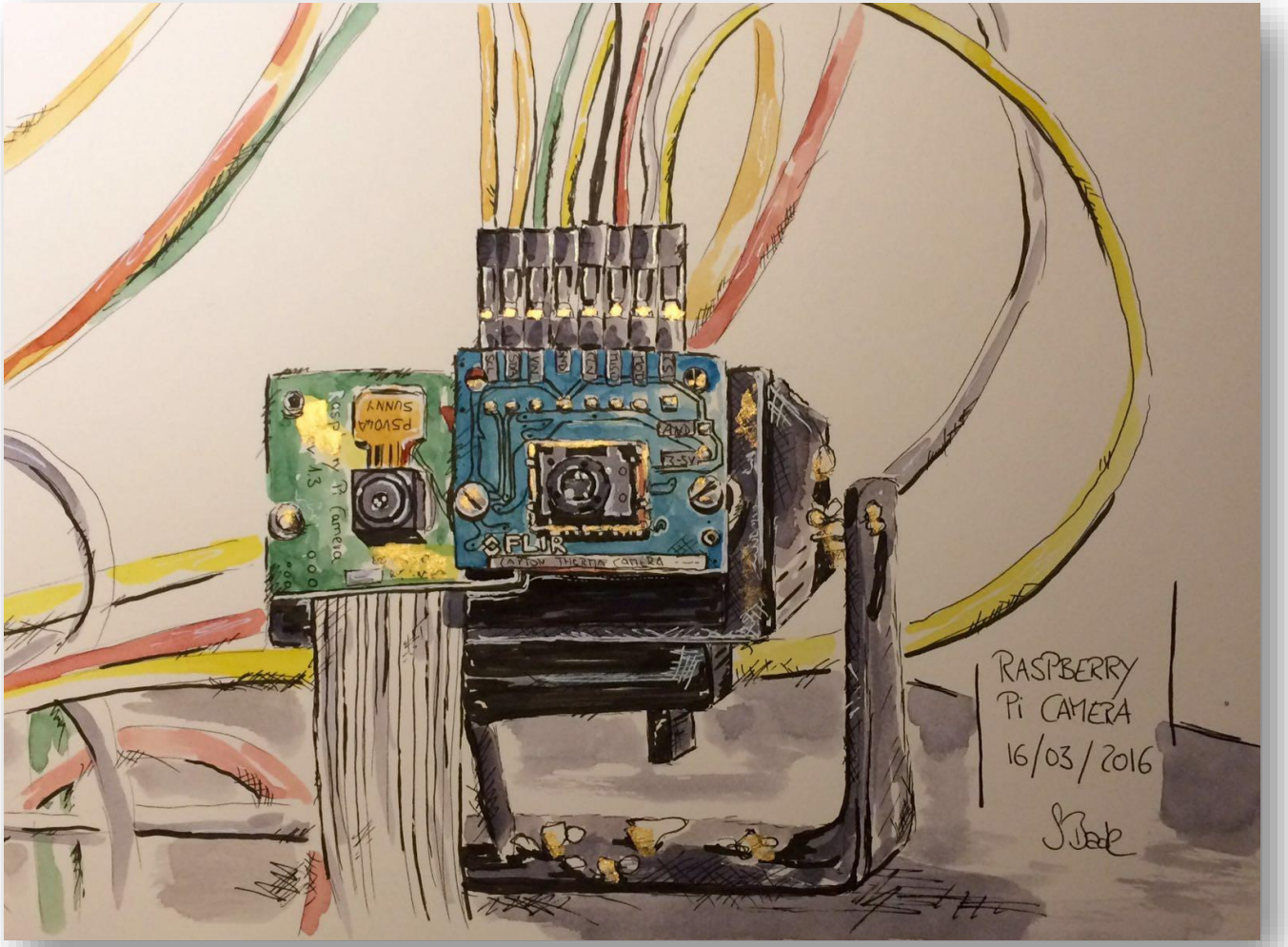
elWObot → Hough-Transformation in Rviz

```
// Now do Hough-Transformation
vector<Vec2f> lines;
HoughLines(cv_ptr->image, lines, cvht_rho, cvht_theta_rad, cvht_threshold, cvht_srn, cvht_stn);
```

OpenCV Funktion *HoughLines* (Programmiersprache C)

ROS → Middleware und Simulationsumgebung mit OpenCV




RASPBERRY
Pi CAMERA
16/03/2016