

---

# **Indoor use of Location-based Augmented Reality**

**Setrak Michaelis**

**Cocoa-Heads Siegen**

---

Universität Siegen

Fachbereich 12 - Elektrotechnik und Informatik  
7. März 2012

---

# Übersicht

---

## Einleitung

- Zielsetzung
- Lösungsweg

## OpenCV

- Was ist OpenCV
- Machine learning

## XML Generierung

- Verfahren
  - Tools
-

# Übersicht

---

## Messungen

- Messdaten

## Augmented Reality

- Realisierung

## Fazit

---

# Einleitung

---

## Zielsetzung

- visuelle Erkennung von Markern
  - möglichst schnell
  - auf "älteren" Geräten lauffähig
  - Apple-Market kompatibel
  - Definition "Marker"
-

# Marker

---



# Einleitung

---

## Lösungsweg

- Barcodes / QR-Codes
    - Problem: zu klein / unschöne Optik
  - Template matching
    - Problem: Betrachtungswinkel / Helligkeit
  - Face recognition mit OpenCV
    - freie Wahl der Bilder
    - Erkennungstärke einstellbar
-

# OpenCV

---

## Was ist OpenCV?

- eine freie Programmbibliothek mit Algorithmen für die Bildverarbeitung und maschinelles Sehen.
  - anfangs C++ inzwischen auch in Python und (Java)
  - Stärken sind Geschwindigkeit und die große Menge an verschiedenen Algorithmen z.B.
    - Filter (Sobel, Canny, Gauß)
    - Machine learning
    - allgemeine Bildbearbeitungsfunktionen
    - etc.
  - Windows, Mac, Linux
-

# OpenCV

---

## Machine learning

- Unter Machine learning versteht man der Maschine etwas "beizubringen"
  - realisiert durch die traincascade-Methode von OpenCV
    - genaue Funktionsweise nicht bekannt und auch nicht dokumentiert
    - Ausgabeformat ist eine XML-Datei
    - Mit der XML-Datei wird das Objekt erkannt
-



# XML Generierung

```
4 <stageType>BOOST</stageType>
5 <featureType>LBP</featureType>
6 <height>24</height>
7 <width>24</width>
8 <stageParams>
9   <boostType>GAB</boostType>
10   <minHitRate>9.9000000953674316e-001</minHitRate>
11   <maxFalseAlarm>3.0000001192092896e-001</maxFalseAlarm>
12   <weightTrimRate>9.4999999999999996e-001</weightTrimRate>
13   <maxDepth>1</maxDepth>
14   <maxWeakCount>100</maxWeakCount></stageParams>
15 <featureParams>
16   <maxCatCount>256</maxCatCount></featureParams>
17 <stageNum>2</stageNum>
18 <stages>
19   <!-- stage 0 -->
20   <_>
21     <maxWeakCount>1</maxWeakCount>
22     <stageThreshold>1.</stageThreshold>
23     <weakClassifiers>
24       <_>
25         <internalNodes>
26           0 -1 1 -2097153 -2105857 -719847425 -10486273 -69215489
27           -69477425 -373564417 -5</internalNodes>
28         <leafValues>
29           -1. 1.</leafValues></_></weakClassifiers></_>
--
```

# XML Generierung

---

## Verfahren

- Anforderung:
  - Positiv Samples
  - Negativ Samples
  - .txt Dateien die den Ort der Samples enthält
  - viel Zeit...

# XML Generierung

---

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Seto>cd Desktop\Bilderkennung

C:\Users\Seto\Desktop\Bilderkennung>opencv_createsamples.exe
Usage: opencv_createsamples.exe
    [-info <collection_file_name>]
    [-img <image_file_name>]
    [-vec <vec_file_name>]
    [-bg <background_file_name>]
    [-num <number_of_samples = 1000>]
    [-bgcolor <background_color = 0>]
    [-inv] [-randinv] [-bgthresh <background_color_threshold = 80>]
    [-maxidev <max_intensity_deviation = 40>]
    [-maxxangle <max_x_rotation_angle = 1.100000>]
    [-maxyangle <max_y_rotation_angle = 1.100000>]
    [-maxzangle <max_z_rotation_angle = 0.500000>]
    [-show [<scale = 4.000000>]]
    [-w <sample_width = 24>]
    [-h <sample_height = 24>]

C:\Users\Seto\Desktop\Bilderkennung>
```

# XML Generierung

---

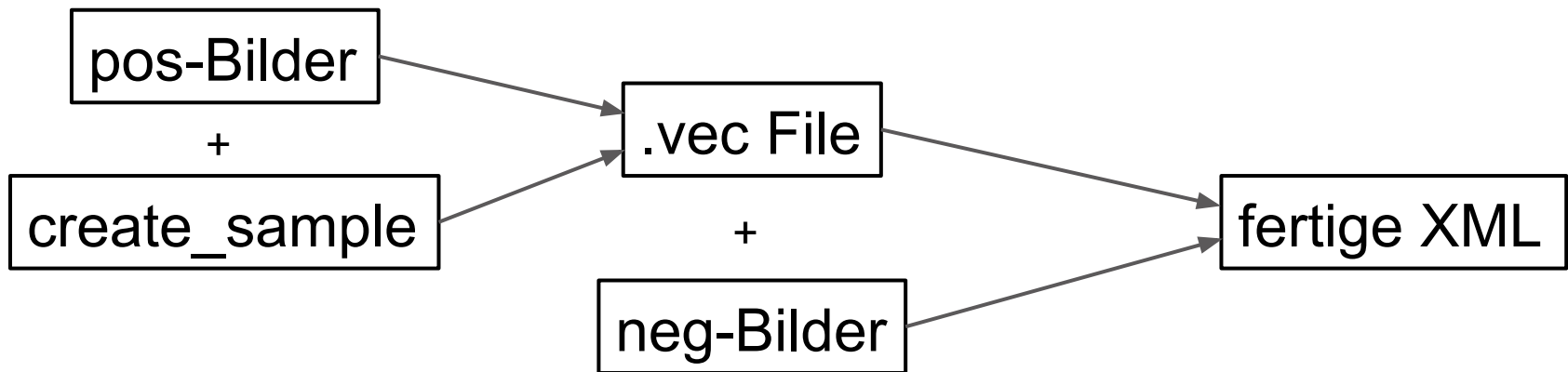
```
C:\Users\Seto\Desktop\Bilderkennung>opencv_traincascade.exe
Usage: opencv_traincascade.exe
  -data <cascade_dir_name>
  -vec <vec_file_name>
  -bg <background_file_name>
  [-numPos <number_of_positive_samples = 2000>]
  [-numNeg <number_of_negative_samples = 1000>]
  [-numStages <number_of_stages = 20>]
  [-precalcValBufSize <precalculated_vals_buffer_size_in_Mb = 256>]
  [-precalcIdxBufSize <precalculated_idx_buffer_size_in_Mb = 256>]
  [-baseFormatSave]
--cascadeParams--
  [-stageType <BOOST(default)>]
  [-featureType <{HAAR(default), LBP}>]
  [-w <sampleWidth = 24>]
  [-h <sampleHeight = 24>]
--boostParams--
  [-bt <{DAB, RAB, LB, GAB(default)}>]
  [-minHitRate <min_hit_rate> = 0.995]
  [-maxFalseAlarmRate <max_false_alarm_rate = 0.5>]
  [-weightTrimRate <weight_trim_rate = 0.95>]
  [-maxDepth <max_depth_of_weak_tree = 1>]
  [-maxWeakCount <max_weak_tree_count = 100>]
--haarFeatureParams--
  [-mode <BASIC(default) | CORE | ALL]
--lbpFeatureParams--

C:\Users\Seto\Desktop\Bilderkennung>
```

# XML Generierung

---

## Erstellung:





# XML Generierung

---

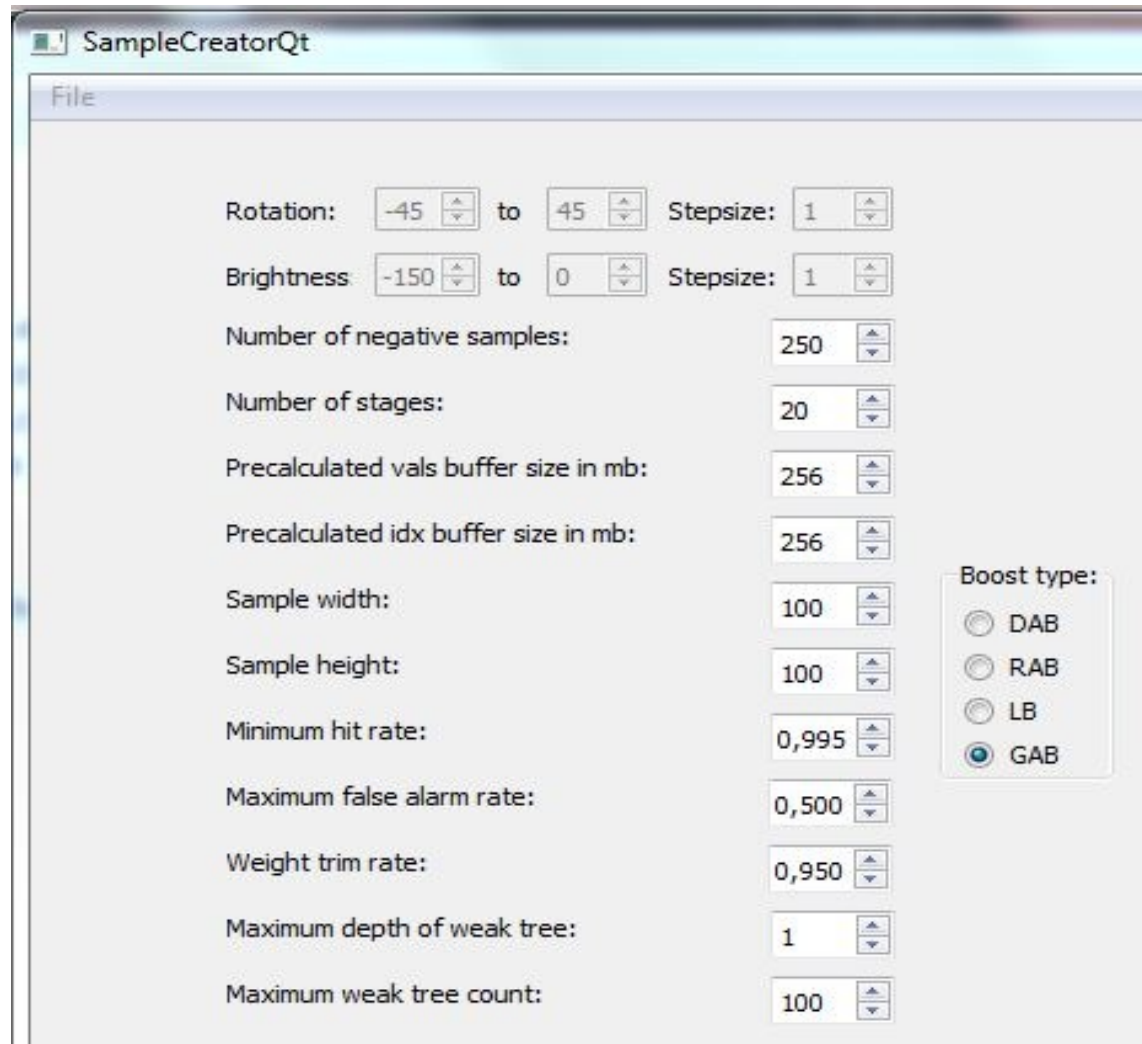
## Tools:

- Tools sollen das Sammeln von pos-Bildern erleichtern
    - es gibt Tools um Videos frame by frame zu durchsuchen
    - in der PG erstelltes Tool das Helligkeit und Rotation ändert und Bilder erstellt
    - etc. (Google hilft)
-



# XML Generierung

---



SampleCreatorQt

File

Rotation: -45 to 45 Stepsize: 1

Brightness: -150 to 0 Stepsize: 1

Number of negative samples: 250

Number of stages: 20

Precalculated vals buffer size in mb: 256

Precalculated idx buffer size in mb: 256

Sample width: 100

Sample height: 100

Minimum hit rate: 0,995

Maximum false alarm rate: 0,500

Weight trim rate: 0,950

Maximum depth of weak tree: 1

Maximum weak tree count: 100

Boost type:

- ☐ DAB
- ☐ RAB
- ☐ LB
- ☒ GAB



Rotation: -45 to 45 Stepsize: 1

Brightness: -150 to 0 Stepsize: 1

Number of negative samples: 250

Number of stages: 20

Precalculated vals buffer size in mb: 256

Precalculated idx buffer size in mb: 256

Sample width: 100

Sample height: 100

Minimum hit rate: 0,995

Maximum false alarm rate: 0,500

Weight trim rate: 0,950

Maximum depth of weak tree: 1

Maximum weak tree count: 100

Boost type:

☐ DAB

☐ RAB

☐ LB

☒ GAB

os-cillation  
development and engineering

Image successfully loaded.

Create XML



# Messungen

---

## Anzahl Bilder zu Frames:

Anzahl der XML-Files	Frames
1	ca. 8,5
2	ca. 6
3	ca. 5
4	ca. 4
5	ca. 3 - 4

---

# Messungen

---

## Messdaten:

Bildgröße (cm)	Erkennungsdistans (m)
7,8 x 7,8	ca. 0,73
15,5 x 15,5	ca. 1,6
31 x 31	ca. 3,03 bis 3,10

---

# Augmented Reality

---

## Realisierung

- Idee: Kompass orientiere Texteinblendung
  - um Tiefe zu vermitteln Schriftgröße anpassen
  - Pfeile als Richtungweiser
  - Schilder
-

# Fazit

---

- OpenCV ist sehr vielseitig und interessant
  - Dokumentation ist recht gut, doch leider sehr lückenhaft, vorallem im Bereich Machine learning
  - zurzeit sind die Handys noch "zu langsam"
-

# Vielen Dank

---

Fragen?

---