




# Die Augen der vienna cubes

Computer Vision der fuballspielenden Roboter



vienna cubes






# Inhalte

- Das Team
- Die Aufgabe
- Die Hardware
- How to ...?
- Die Software




vienna cubes



# Das Team

```

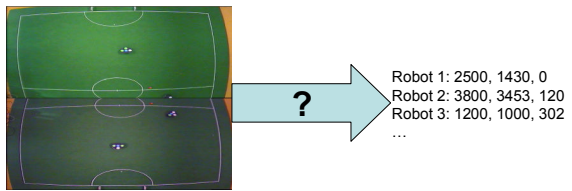
    graph TD
      AH[Alexander Hofmann  
Projektleiter] --- PV[Peter Voith  
Teamleitung]
      PV --- WH[Walter Hammel  
Mechanik]
      PV --- ML[Manfred Leonhartsberger  
Embedded Systems]
      PV --- MJ[Matthias Jakob  
Vision]
      PV --- DS[Dietmar Schreiner  
Knstliche Intelligenz]
      ML --- AP[Andreas Puhm]
      MJ --- WS[Walter Scherer]
      DS --- APK[Anton Pirker  
Stephan Ptschner  
Florian Demmer]
      AP --- J1[Juniors]
      WS --- J2[Junior:  
Timo Hunger]
      APK --- J3[Juniors]
  
```




vienna cubes

# Die Aufgabe



Robot 1: 2500, 1430, 0  
 Robot 2: 3800, 3453, 120  
 Robot 3: 1200, 1000, 302  
 ...



vienna cubes






# Die Hardware

## Hohe Auflsung



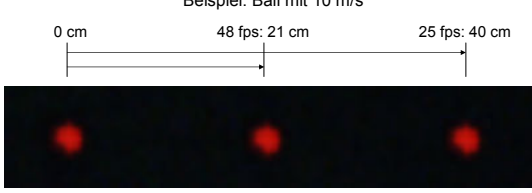


vienna cubes

# Die Hardware

## Hohe Framerate

Beispiel: Ball mit 10 m/s

vienna cubes

**Die Hardware**

Niedrige Latenzzeiten

The diagram illustrates the hardware setup for low latency. A camera mounted on a robot is connected to a frame grabber via CameraLink. The frame grabber is connected to a PC via PCI. The PC contains CPU(s).

vienna cubes

**Die Hardware - Kameras**

**2x JAI A70-CL**

- 1/2" progressive scan color CCD
- 8 / 10 bit CameraLink output
- 60 fps @ 768 x 576 pixel

vienna cubes

**Die Hardware – Framegrabber**

**SiliconSoftware MicroEnable 3 XXL**

- 64 bit / 66 MHz PCI Framegrabber
- MXC2S-600 microEnable mit Xilinx Spartan XC2S-600 Koprozessor
- Echtzeit-Konvertierung von 2 CameraLink Bayer-Kameras auf RGB-Format

vienna cubes

**How to ... ?**

Walter Scherer

vienna cubes

**Themengebiete**

- Kamera- / Framegrabberkonfiguration
- Spielfeldkonfiguration
- Suche nach interessanten Pixeln
- Pattern Recognition
- Probability Analysis

vienna cubes

**Themengebiete**

- Kamera- / Framegrabberkonfiguration
- Spielfeldkonfiguration
- Suche nach interessanten Pixeln
- Pattern Recognition
- Probability Analysis

vienna cubes

## cubes Kamerakonfiguration FACHHOCHSCHULE TECHNIKUM WIEN

- externes Tool von JAI
  - Änderung von Parametern im laufenden Betrieb möglich
    - Gamma Correction
    - Shutter Speed
    - u.v.m.
- alternativ: über eigene Kommandos direkt über CameraLink zur Kamera



vienna cubes

## cubes Framegrabber Config FACHHOCHSCHULE TECHNIKUM WIEN

- eigene SDK
  - wird im Programm über Funktionen verwendet
  - Beispiel: Allokieren eines Speicherblocks für den Framebuffer



vienna cubes

## cubes Themengebiete FACHHOCHSCHULE TECHNIKUM WIEN

- Kamera- / Framegrabberkonfiguration
- Spielfeldkonfiguration
- Suche nach interessanten Pixeln
- Pattern Recognition
- Probability Analysis



vienna cubes

## cubes Spielfeldkonfiguration FACHHOCHSCHULE TECHNIKUM WIEN

### Das Problem

- Schatten
- Unterschiedliche Helligkeiten
- Wo sind die Linien?



vienna cubes

## cubes Spielfeldkonfiguration FACHHOCHSCHULE TECHNIKUM WIEN

### Eine Lösung (1)

- Schatten / Helligkeiten
  - Abweichungswerte von Default Pixel feststellen...
  - ...und bei der Erkennung der Pixel berücksichtigen
  - Annahme 1: Keine Veränderung der Lichtsituation während eines Spieles
  - Annahme 2: Kamera liefert "stabile" Werte



vienna cubes

## cubes Spielfeldkonfiguration FACHHOCHSCHULE TECHNIKUM WIEN

### Eine Lösung (2)

- Wo sind die Linien?
  - Linien = Lange weiße Streifen im Bild bei fixen Koordinaten
  - Elfmeterpunkte = Kleine weiße Punkte bei fixen Koordinaten
  - Annahme: Position von Kamera zu Feld wird nicht verschoben.



vienna cubes

## cubes **Themengebiete**



- Kamera- / Framegrabberkonfiguration
- Spielfeldkonfiguration
- Suche nach interessanten Pixeln
- Pattern Recognition
- Probability Analysis



vienna cubes

## cubes **Feature Pixelsuche**



- Manuelles Durchiterieren durch das Bild
  - Pointerarithmetik
- Verwenden von Intel IPP oder OpenCV Library Funktionen
  - Umfangreiche Sammlung von u.a. Bildverarbeitungsfunktionen
- Verwenden von anderen Algorithmen



vienna cubes

## cubes **Linear Scan**



- Bild Zeile für Zeile, Pixel für Pixel durchsuchen
- Pixelwerte mit gewünschten Werten vergleichen
- ⇒ Menge an Feature Pixels



vienna cubes

## cubes **Monte Carlo Sampling**



- Suche "auf gut Glück"
  1. Nimm irgendein noch nicht überprüftes Pixel aus dem Bild
  2. Vergleiche es mit den gewünschten Wertebereichen
  3. Bei Treffer: Weiter bei 5.
  4. Bei fehlender Übereinstimmung: Weiter bei 1.



vienna cubes

## cubes **Monte Carlo Sampling**



- Suche "auf gut Glück"
  5. Merke dir Position des Pixels
  6. Suche die Umgebung des Pixels nach weiteren interessanten Pixeln ab
  7. Bei Treffer: Wahrscheinlichkeit groß, dass ein Punkt gefunden wurde (z.B. Ball)
  8. Bei fehlender Übereinstimmung: Nimm Pixel in die Pixelliste auf und gehe zu 1.



vienna cubes

## cubes **Monte Carlo Sampling**



- teilweise schnellere Entdeckung von interessanten Pixeln
- aber: Implementierung komplizierter, mehr Speicheraufwand
- Einsatz bei den vienna cubes?
  - Geschwindigkeitsfrage



vienna cubes

- Kamera- / Framegrabberkonfiguration
- Spielfeldkonfiguration
- Suche nach interessanten Pixeln
- Pattern Recognition
- Probability Analysis



- “Pattern Recognition is the scientific discipline dealing with methods for object description and classification.”  
– J.P. Marques de Sá, Pattern Recognition
- Grundkonzept: Similarity
- Beispiel: Roter Apfel und grünliche Orange



- Kamera- / Framegrabberkonfiguration
- Spielfeldkonfiguration
- Suche nach interessanten Pixeln
- Pattern Recognition
- Probability Analysis



- Wie wahrscheinlich ist es, dass ein Roboter, der vor 16 ms noch rechts oben war, jetzt links unten steht?  
– so gut wie unmöglich  
– ⇒ Einsatz von Area of Interest-Scanning (AOI Scanning)



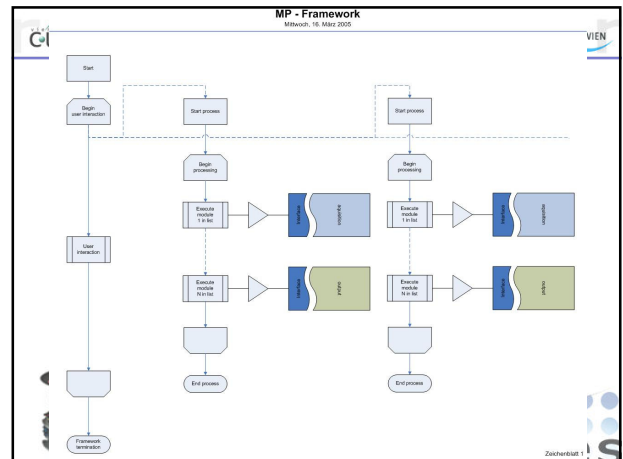
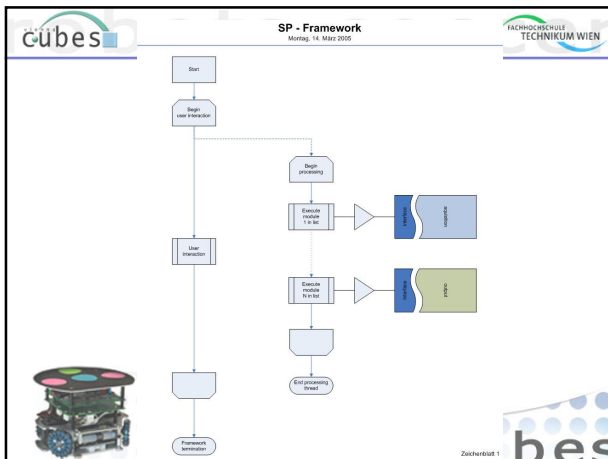
- Betrachte nicht jeden Frame einzeln, sondern auch die Vorgängerbilder  
– D.h. war im Vorgängerbild der Ball an Position (413,214) zu finden, dann wird er im nächsten Bild auch in diesem Bereich zu finden sein.  
– ⇒ Geschwindigkeitsgewinn  
– alle 5 Sekunden: Gesamtscan, um neu hinzugekommene Objekte zu erkennen



## Softwaredesign, verwendete Tools, ...

Matthias Jakob





**Betriebssystem**

- **Microsoft Windows XP**
- Framegrabbersoftware für Windows ausgereifter
- Mehrere Teammember können mit Windows besser umgehen
- Annehmbare IDE (MS VS.NET 2003)
- Restliche benötigte Software wäre auch für Linux verfügbar

**Entwicklertools**

- **Microsoft Visual Studio .NET 2003 Professional**
  - Bequeme Programmierung
- **Intel C++ Compiler 8.1**
  - Enorme Performanceoptimierungen

**Libraries**

- **Intel Integrated Performance Primitives**
  - „Intel® Integrated Performance Primitives (Intel® IPP) offers highly optimized functions for audio, video, graphics, imaging, speech, cryptography, speech recognition, and signal processing. Intel IPP delivers maximum performance in the least amount of time.“
- **Intel OpenCV**
  - „This library is mainly aimed at real time computer vision. Some example areas would be Human-Computer Interaction (HCI); Object Identification, Segmentation and Recognition; Face Recognition; Gesture Recognition; Motion Tracking, Ego Motion, Motion Understanding; Structure From Motion (SFM); and Mobile Robotics. „
- **Boost C++ Libraries**
  - „A further goal is to establish "existing practice" and provide reference implementations so that Boost libraries are suitable for eventual standardization. „

**Vielen Dank für Ihre Aufmerksamkeit**