

# **JONGLIEREN MIT DER KINECT**

**EIN SOFTWAREPROJEKT IM**

## **PROJEKT BILDVERARBEITUNG**

PROJEKTBERICHT

**ROLF BOOMGAARDEN**  
**FLORIAN LETSCH**  
**THIEMO GRIES**

**29. MÄRZ 2014**

UNTER AUFSICHT VON: **BENJAMIN SEPPKE**  
**ARBEITSBEREICH KOGNITIVE SYSTEME**  
**FACHBEREICH INFORMATIK, UNIVERSITÄT HAMBURG**

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Motivation</b>	<b>3</b>
<b>3</b>	<b>Zielsetzung</b>	<b>3</b>
<b>4</b>	<b>Möglichkeiten der Kinect</b>	<b>4</b>
<b>5</b>	<b>Recherche: Ein jonglierender Roboter</b>	<b>4</b>
<b>6</b>	<b>Lösungsidee</b>	<b>4</b>
<b>7</b>	<b>Umsetzung</b>	<b>4</b>
7.1	Programmfluss . . . . .	4
7.1.1	Schritt 1 . . . . .	4
7.1.2	Schritt 2 . . . . .	4
7.1.3	Schritt 3 . . . . .	4
7.1.4	Schritt 4 . . . . .	4
7.2	Erläuterung verwendeter Bildverarbeitungsverfahren . . . . .	5
7.2.1	Kalman Filter . . . . .	5
7.3	Herausforderungen . . . . .	5
7.4	Bewertung der Umsetzung . . . . .	5
<b>8</b>	<b>Anwendungsmöglichkeiten</b>	<b>5</b>
<b>9</b>	<b>Fazit</b>	<b>5</b>
	<b>Quellen</b>	<b>6</b>

# 1 Einleitung

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed aliquam, ligula vitae condimentum malesuada, turpis nisi placerat eros, vel facilisis mi neque quis nulla. Aenean eleifend risus id dolor ultricies scelerisque. Phasellus venenatis libero enim, vel lacinia massa interdum nec. Quisque a euismod ligula. In eget mattis orci. Integer vitae enim ac nisl scelerisque luctus ut et nibh. Quisque ut odio ultrices, consequat mi vel, accumsan metus. Donec faucibus, nulla vel mattis euismod, felis leo accumsan tortor, et congue turpis leo et elit. Proin gravida mollis facilisis. In enim nisi, pellentesque id tincidunt a, accumsan eget elit. Aliquam erat volutpat. In quam ante, accumsan eu est a, molestie euismod neque. Proin porta rhoncus nisl sed dignissim. Aliquam lacinia sed libero et eleifend. Ut placerat tortor eget augue pellentesque rutrum.

## 2 Motivation

Mit den technischen Möglichkeiten eines Tiefen- und Bilddaten liefernden Systems (konkret: Microsoft Kinect) soll in dieser Arbeit versucht werden, das Wurfmuster eines mit Bällen jonglierenden Akteurs zu analysieren.

Ein Jongleur wirft Jonglierbälle in einem Muster, das möglichst gleichmäßig ist. So ist der Höhepunkt der Flugbahn idealerweise konstant auf der gleichen Höhe. Zum Analysieren des Jongliermusters wäre dies also bereits ein erstes Kriterium, die *Güte eines Jongliermusters* automatisiert zu bewerten.

Denkbar sind auch weitere Anwendungen, wie etwa das automatische Zählen von erfolgreich gefangenen Würfeln. Eine computergesteuerte Erfassung der insgesamten Wurfbzahl ist ein einfaches Kriterium für eine *Leistungsbewertung des jonglierenden Benutzers*.

Die genaue Anwendung ist jedoch nicht Ziel dieser Arbeit. Stattdessen verfahren wir in einem bottom-up Herangehen, um von den rohen Bild- und Tiefendaten der Kinect ausgehend Informationen über sich im Bild befindliche Objekte (Jonglierbälle) zu erfassen und deren Bewegung zu erkennen. Das Ergebnis ist dann ein Fundament, auf dessen Grundlage konkrete Anwendungen entwickelt werden können.

## 3 Zielsetzung

Am Ende dieser Arbeit soll eine Anwendung stehen, die mit Hilfe der Kinect Daten über die Flugbahnen dreier jonglierter Bälle liefert.

Ein Akteur befindet sich hierbei im Bildzentrum in einem wohl definierten Abstand zur Kinect. Es werden drei matte Bälle beliebiger Farbe jongliert. Um das Ergebnis unabhängig von der Szenenbeleuchtung zu halten, sollen die Tiefendaten ausreichend Information für das eindeutige Identifizieren der Bälle liefern.

## 4 Möglichkeiten der Kinect

Die Kinect ist eine von Microsoft zur Spielekonsole Xbox 360 vertriebene Erweiterung, die den Spieler mit einem RGB- und einem Tiefensensor erfasst und diese beiden Datenströme an die Konsole liefert. Da die Kinect über einen USB-Anschluss verfügt, kann sie an konventionellen Rechnern angeschlossen und betrieben werden. Eine quelloffene Implementierung zur Unterstützung der Kinect ist das `freenect` Projekt, das für Linux, Windows und MacOS zur Verfügung steht und im Rahmen dieser Arbeit als Bibliothek für Python verwendet wurde. **FIXME:** Quellen.

Die Videoquelle der Kinect liefert standardmäßig 30 (JA?) Bilder pro Sekunde mit einer Auflösung von 640x480 und 8 bit Farbtiefe. Die Tiefendaten stammen von einer Infrarot-Kamera und liefern bei gleicher Bildfrequenz 2048 verschiedene Tiefenwerte (11bit). Aus der Funktionsweise der Infrarotkamera ergibt sich, dass die Kinect in Umgebungen starker Infrarotstrahlung (beispielsweise im Tageslicht) nur beschränkt einsatzfähig ist.

**FIXME:** Erklären wie das mit diesem projizierten Punktemuster funktioniert.

## 5 Recherche: Ein jonglierender Roboter

(paper...)

## 6 Lösungsidee

image Processing Pipeline

## 7 Umsetzung

### 7.1 Programmfluss

#### 7.1.1 Schritt 1

#### 7.1.2 Schritt 2

#### 7.1.3 Schritt 3

#### 7.1.4 Schritt 4

verschiedene Ansätze...

## 7.2 Erläuterung verwendeter Bildverarbeitungsverfahren

### 7.2.1 Kalman Filter

**FIXME:** alles überarbeiten, mehr, nochmal nachlesen, wie's wirklich ist, Schaubilder

Der Kalman Filter kann sich bewegende Objekte beobachten und Schätzungen zur aktuellen oder auch zukünftigen Position machen.

In diesem Projekt wird er dazu verwendet, die Bälle zu beobachten und die weitere Flugbahn zu bestimmen. Damit kann eine voraussichtliche Flugbahn in die Ausgabe gezeichnet werden, aber auch in der internen Verfolgung und Zuordnung der Bälle spielen die Ergebnisse eine wichtige Rolle.

Der Kalman Filter besteht aus zwei Funktionen, einem `predict` und einem `update`. Im `predict` wird mit Informationen zur Art der Bewegung und mit mindestens einer Ortsinformation zum Zeitpunkt  $t$  eine Schätzung zum Ort im Zeitpunkt  $t+1$  gemacht. Ein `predict` kann beliebig oft hintereinander ausgeführt werden.

Im `update` wird die aktuell gespeicherte Ortsinformation mit extern gewonnenen Daten aktualisiert. Die hier gemessene Abweichung zwischen Schätzung und tatsächlichen Daten kann natürlich auch zur weiteren Schätzung eingebracht werden. Ein `update` wird nicht mehrmals hintereinander ausgeführt.

## 7.3 Herausforderungen

## 7.4 Bewertung der Umsetzung

Effizient. Speedup-Möglichkeiten

## 8 Anwendungsmöglichkeiten

Projekte nehmen als Grundlage für einfache Programme / Spiele.

- Objekte zählen
- Würfe zählen
- ... ?

## 9 Fazit

## Quellen

- [1] Paul Viola, Michael Jones,  
*Robust Real-time Object Detection*  
Vancouver, Canada, 13.07.2001.  
[http://research.microsoft.com/en-us/um/people/viola/Pubs/Detect/violaJones\\_IJCV.pdf](http://research.microsoft.com/en-us/um/people/viola/Pubs/Detect/violaJones_IJCV.pdf)
  
- [2] Ole Helvig Jensen,  
*Implementing the Viola-Jones Face Detection Algorithm*  
IMM-M.Sc.: ISBN 87-643-0008-0      ISSN 1601-233X  
Technical University of Denmark, Informatics and Mathematical Modelling  
Kongens Lyngby, Denmark, 2008.  
[http://www.imm.dtu.dk/English/Research/Image\\_Analysis\\_and\\_Computer\\_Graphics/Publications.aspx?lg=showcommon&id=223656](http://www.imm.dtu.dk/English/Research/Image_Analysis_and_Computer_Graphics/Publications.aspx?lg=showcommon&id=223656)