



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b> .....	<b>3</b>
<b>2</b>	<b>Allgemeine Informationen</b> .....	<b>3</b>
2.1	Systemvoraussetzungen, Plattformen .....	3
2.1.1	Hier steht eine Überschrift dritter Ebene .....	3
2.2	Noch eine Überschrift zweiter Ebene .....	4

## 1 Einleitung

OpenCV ist eine Open-Source Bibliothek, welche Funktionen für den Bereich Computer Vision enthalten. Das Softwarepaket wurde von Intel<sup>®</sup> für Linux und Windows OS entwickelt und im Jahr 2000 veröffentlicht. Sämtliche Dateien, die für den Einsatz der Bibliothek notwendig sind, können unter <http://www.intel.com/research/mrl/research/opencv> frei heruntergeladen werden. Sämtliche Routinen der Bibliothek sind in ANSI C geschrieben und selbstverständlich im Quelltext vorhanden. Einer Newsgroup kann unter <http://groups.yahoo.com/groups/OpenCV> beigetreten werden. [1]

- Zielsetzung dieser Bibliothek ist die Grundlage für einen breiten Einsatz von Personal Computern im Bereich der Computer Vision zu schaffen. Die Bibliothek beinhaltet
- se und der 3D-Rekonstruktion vorhanden, so dass beispielsweise die Umwelterfassung
- eines Roboters durch „hintereinanderschalten“ von Routinen aus der OpenCV Biblio
- thek modelliert werden kann. Selbstverständlich können die Routinen dieser Bibliothek aber auch für nicht-Echtzeitanwendungen eingesetzt werden.

Ziel dieser Ausarbeitung ist es, einen Überblick über die Funktionalität von OpenCV zu geben und in Form eines Tutorials die Verwendung der Bibliothek zu demonstrieren. Abschließend wird ein potentiell Anwendungsbeispiel im medizinischen Bereich vorgestellt.

## 2 Allgemeine Informationen

### 2.1 Systemvoraussetzungen, Plattformen

#### 2.1.1 Hier steht eine Überschrift dritter Ebene

OpenCV wurde für Personal-Computer entwickelt, deren Prozessortechnik auf der Intel-Architektur basiert (Pentium<sup>®</sup> MMX, Pentium<sup>®</sup> Pro, Pentium<sup>®</sup> III, Pentium<sup>®</sup> 4). Die implementierten Funktionen verwenden zum Großteil dieselben Strukturen wie die Image Processing Library (IPL), bei der mehr als die Hälfte der Funktionen<sup>1</sup> auf Assembler-Ebene optimiert wurden.

Zur Zeit werden die Betriebssysteme Microsoft Windows 95, Windows 98, Windows 2000, Windows NT sowie Linux unterstützt. Neben dem Betriebssystem wird auf jeden Fall ein C++ Compiler benötigt. Um sämtliche Funktionen und Demos benutzen zu können muss bei Windows OS zusätzlich DirectX 8.x, ActiveTCL 8.3.3 sowie die IPL 2.2+ installiert sein, bei

---

<sup>1</sup> Daran, dass sich bei Bildfolgen die Position und Intensität der Hintergrundpixel nicht verändern, solange sie nicht vom bewegten Objekt überdeckt werden. Ist das Objekt identifiziert kann dessen Bewegung analysiert werden. Hierbei können Motion Templates verwendet werden. Diese erzeugen Bilder, die es ermöglichen festzustellen, in welche Richtung und was für eine Art der Bewegung stattgefunden hat.

Linux wird TCL 8.3.3 + BWidgets, Video4Linux und die IPL 2.2+ benötigt. Auf einem Windows System benötigt OpenCV ca. 110 MByte Festplattenplatz. Für eine Installation der IPL werden weitere 50 MByte benötigt.

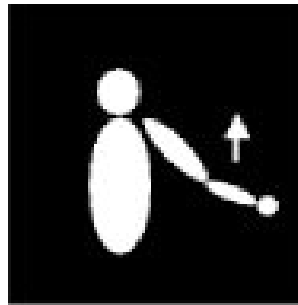


Fig. 1. Motion Tracking 1

Für das Tracking z. B. eines Gesichts einer Person können die Funktionen des CamShift-Algorithmus verwendet werden.

## 2.2 Noch eine Überschrift zweiter Ebene

Dieser erstellt aus der Verteilung der Farbwerte und der bisherigen Bewegung des Objekts, ein Fenster, in dem sich das Objekt beim nächsten Trackvorgang mit der größten Wahrscheinlichkeit aufhalten wird. Active Contours, besser bekannt unter dem Begriff Snakes, sind ebenfalls im Funktionsumfang. Mittels Energie-minimierender Strukturen werden auch hier Konturen herausgefiltert. Eine andere Möglichkeit für ein Tracking stellt die Berechnung des optischen Flusses (Optical Flow) dar. Dieser benutzt für seine Berechnungen die Bildkoordinaten, die Intensität und den Aufnahmezeitpunkt. Mit geeigneten Annahmen, z. B. dass sich die Intensität der Pixel eines ruhenden oder eines sich bewegenden Objektes im Verlauf der Zeit nicht ändert, kommt man nun zum gewünschten Ergebnis. Eine weitere Möglichkeit, die in OpenCV berücksichtigt wurde, ist die Anwendung von Estimators. Ein Programm, das auf diese Weise arbeitet, besteht aus einem Modell der Szene, das für jeden Zeitpunkt alle Variablen und Parameter vollständig erfasst, und einem Estimator. Im Fall von OpenCV wurden Kalman und ConDensation implementiert.

## **Literaturverzeichnis**

[1] Hier stehen meine Literaturangaben, auf die ich im Text verweise