

Bitte beachten Sie, dass der neue Aufbau gemäss der Anleitung zur Erstellung eines Web-Abstracts und der Vorlage des Web-Abstracts eingehalten werden soll.

Finger Tracking for High End Interface

Themenbereiche:	Bildverarbeitung, Range Sensor, Tracking
Studierender:	Mario Bucher, Lucas Schnüriger
Dozent:	Prof. Dr. Thomas Koller
Experte:	Ueli Amstutz, dipl. Ing. HTL, Opacc Software AG, Rothenburg
Wirtschaftspartner:	ABUSIZZ GmbH, Zermatt
Keywords:	Orbbec Astra, 3D-Sensor, OpenCV, Tiefenbilder, Fingergesten

1. Aufgabenstellung

Die Schweizer Firma ABUSIZZ stellt interaktive Holztische her, bei denen die Benutzeroberfläche von oben herab auf die Tischfläche projiziert wird. Mit Fingergesten kann mit dem System interagiert werden. Das bisherige System erfüllt die Anforderungen an Latenz nicht mehr und ein neuer Sensor zusammen mit neuen Algorithmen sollen Besserung bringen. Es soll ein Trackingsystem entwickelt werden, dass mittels eines 3D Sensors die Finger erkennt und Events zur Interaktion an eine Applikation weiterleitet. Dabei sollen verschiedene Ansätze zur Erkennung von Fingergesten erarbeitet und miteinander verglichen werden.

2. Ergebnisse

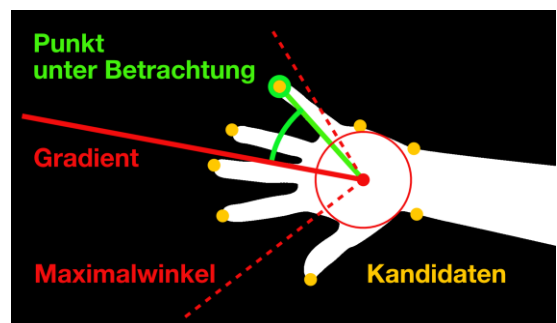
Im Rahmen der Bachelor-Diplomarbeit (BDA) wurden folgende Ergebnisse erarbeitet:

- **Finger Tracking Bibliothek:** Eine statische C++ Bibliothek mit allen entwickelten Algorithmen. Sie ist plattform- und sensorunabhängig implementiert. Die Methoden der Bibliothek können in einer anderen Applikation für einen konkreten Anwendungsfall benutzt werden.
- **Beispielapplikation:** Eine C++ Applikation, welche die Verwendung der Bibliothek und die Anbindung an Umsysteme demonstriert. Sie besteht aus den für den Anwendungsfall der ABUSIZZ optimalsten Methoden und ermöglicht eine Interaktion mit dem bestehenden System des Wirtschaftspartners.
- **Testapplikation:** Eine C++ Applikation zum Testen und Vergleichen der entwickelten Algorithmen. Sie dient als Grundlage für den Entscheid der optimalen Methoden für die Beispielapplikation.

In der Beispielapplikation wurde anhand der Resultate der Testapplikation die für die Anwendung der ABUSIZZ optimalsten Verfahren zusammengesetzt.

Die Applikation funktioniert mit dem zur Verfügung gestellten Sensor Astra S der Firma Orbbec und dient als Prototyp um Taps auf einem Tisch zu erkennen und auf einer Windows Oberfläche auszuführen.

Dabei wird das Tiefenbild vom Sensor für die Weiterverarbeitung zunächst in eine Matrix umgewandelt. Die Segmentierung der Hand basiert auf der Tatsache, dass der Hintergrund vom Bild in diesem Anwendungsfall statisch ist. Deshalb wird zur Segmentierung dem Bild ein (beim Start der Applikation erstelltes) Hintergrundmodell subtrahiert. Daraus resultiert eine Binärmaske der Hand. Eine konvexe Hülle um die Hand extrahiert mögliche Punkte an den Fingerspitzen. Mit Hilfe der Ausrichtung der Hand und eines definierten Winkels wird bestimmt, bei welchen Kandidaten es sich tatsächlich um Fingerendpunkte handelt. Nun wird die Distanz zum Sensor eines Fingerendpunktes mit einem vorgeetzten Referenzpunkt vom Tisch verglichen. Unterschreitet die Differenz dieser Punkte einen bestimmten Wert, interpretiert das Programm dies als Tap und führt einen Klick-Event an den entsprechenden Koordinaten aus.



Eine Konfigurationsdatei ermöglicht die zahlreichen Parameter für die Beispielapplikation zu modifizieren. So kann je nach Praxisanforderungen ein Fokus auf hohe Sensitivität (möglichst gute Erkennung aller Taps) oder Spezifität (möglichst keine fälschlicherweise erkannten Taps) eingestellt werden.

Je nach Konfiguration resultiert die Beispielapplikation in einer Sensitivität von 76-84 % und einer Spezifität von 76-96 %. Es ist also möglich praktisch alle falschen Taps zu eliminieren.

Aufgrund der Aufgabenstellung lag der Fokus bei der Lösungsentwicklung auf dem schnellen Erkennen von Taps. Messungen zeigen, dass das Programm rund 2 Millisekunden benötigt bis es von einem Bild einen Tap erkannt hat. Da der Sensor ungefähr alle 30ms ein neues Bild liefert, kann jedes Frame verarbeitet werden, was maximale Reaktionsgeschwindigkeit sicherstellt.

Ein optionaler Debug-Modus ermöglicht die Ausgabe von Informationen beim Ausführen der Applikation. Der Umfang der Informationen lässt sich über mehrere Stufen, beginnend von einfachen Konsolenausgaben bis hin zur Ausgabe von Bildern der Kamera angereichert mit relevanten Extraktionsmerkmalen, wählen.

3. Lösungskonzept

In zweiwöchigen Problemlösungszyklen wurde iterativ-inkrementell an verschiedenen Verfahren und Algorithmen zur Erkennung von Fingergesten gearbeitet.

Damit aus einem Bild eine Information wie beispielsweise ein Tippen eines Fingers gewonnen werden kann, muss das Bild mehrere Schritte durchlaufen. Dieser Ablauf wird als Bildverarbeitungsprozess definiert. Er diente als Basis für die Lösungsentwicklung in diesem Projekt. Pro Verarbeitungsschritt wurden entsprechende Methoden implementiert. Drängten sich für eine Aufgabe mehrere Varianten auf, wurden auch mehrere Varianten implementiert und diese anschliessend miteinander verglichen. Die nachfolgende Grafik zeigt, in welchem Verarbeitungsschritt welche Verfahren umgesetzt wurden. Verfahren auf der selben Ebene sind jeweils austauschbar, bzw. nur eines von ihnen muss ausgeführt werden.

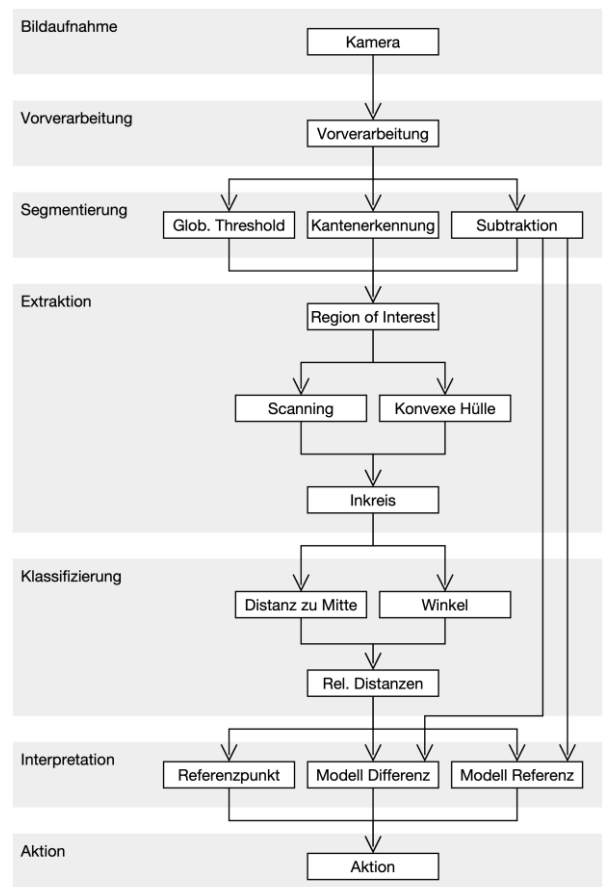
4. Spezielle Herausforderungen

Die Dokumentation der AstraSDK zum Lesen der Daten vom Sensor war sehr spärlich. Auch aufgrund fehlender Erfahrungen im Projektteam mit C++ dauerte es lange bis erste Bilder ausgewertet werden konnten. Trotz intensiver Einarbeit in die Programmiersprache, machten sich die mangelnden C++ Kenntnisse auch im weiteren Projektverlauf bemerkbar. Obwohl eine Lösung zumeist immer gefunden wurde, hätten einige Probleme wohl schneller gelöst werden können.

Durch die Architektur sind die einzelnen Verarbeitungsschritte stark voneinander abhängig. Dies bereitete beim Abstimmen der unzähligen Parameter aufeinander Schwierigkeiten.

5. Ausblick

Beim Projekt lag der Fokus zunächst auf dem Erkennen eines einfachen Antippens des Tisches mit einem einzelnen Finger. Aus zeitlichen Gründen konnten keine weiteren Gesten implementiert werden. Dennoch wurde bei der Lösungsentwicklung auch beachtet, dass mögliche weiteren Gesten, beispielsweise mit mehreren Fingern,



umgesetzt werden könnten. Die Finger Tracking Bibliothek könnte nun mit zusätzlichen Gesten erweitert werden.

Zum Testen der Applikation mit dem Umsystem von ABUSIZZ wurde eine simple Kalibrierung umgesetzt. Für eine genauere Interaktion muss diese Kalibrierung aber noch verbessert werden. Am benutzerfreundlichsten wäre eine automatisierte Kalibrierung beim Start der Applikation.

Damit die Segmentierung mit der Hintergrundsubtraktion auch bei längerer Laufzeit zufriedenstellende Resultate liefert, sollte das Hintergrundmodell laufend aktualisiert werden. So könnten kleine aber stetige Änderungen im Hintergrund abgefangen werden.

Des Weiteren könnten für einzelne Algorithmen gezielt weitere Alternativen untersucht und verglichen werden. Ebenso liesse sich nun mit einem höher frequentierten Sensor prüfen, was die Geschwindigkeit für eine Auswirkung auf das Benutzererlebnis hat.