
Status Report

PR Multimediale Systeme

SS 2016 – Gruppe [M27]

Document Control

Contributors

Person	Role	Company	Contribution
Bernhard Rieder	Developer	UNIVIE	Status Report
Michael Pichler	Developer	UNIVIE	Status Report

Revision History

Issue	Author	Date	Description
V0.1	Michael Pichler	11-May-2016	1 st Progress, Outlook
V0.2	Michael Pichler	11-May-2016	1 st Risk Analysis
V1.0	Michael Pichler	12-May-2016	2 st Risk Analysis, 1 st Feedback (showcase)
V1.1	Bernhard Rieder	13-May-2016	Final review and additional changes
V2.0	Michael Pichler	13-May-2016	Final status report

Table of Contents

1	Progress	3
1.1	Summary.....	3
1.2	Status per Milestone	4
1.3	Activities per Person	4
2	Risk Analysis	5
3	Outlook	5
3.1	Milestones and Schedules	5
3.2	Planned Effort per Person	6
4	Feedback.....	6

1 Progress

1.1 Summary

Der Fortschritt des Projektes wird anhand des Meilensteinplans vom Dokument "Requirements and Design" bewertet. Da bis zum Zeitpunkt 13.05.2016 nur Meilenstein 1 aktiv ist, wird hier kurz eine Übersicht über den Fortschritt des Meilenstein 1 und des Projektes gegeben.

Meilenstein 1:

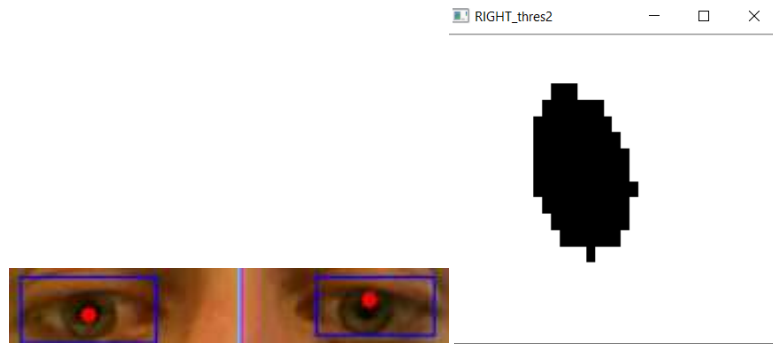
Meilenstein 1 ist einer der aufwändigsten Schritte des Projektes, da hier das Grundgerüst für die nachfolgenden Meilensteine erstellt wird. Das Ziel am Anfang ist, das Projekt zu erstellen. Dabei haben wir uns für die IDE Visual Studio 2015 entschieden und die Programmiersprache C++. Der Grund für Visual Studio ist jener, dass die IDE eine sehr gute Integration für Teamarbeiten besitzt und somit der Ressourcen-Austausch zwischen dem Team erleichtert wird. C++ wird verwendet, da wir die optimale Leistung aus unserem GazeTracker herausbekommen möchten und mit ihr die bestmögliche Leistung in Verbindung mit OpenCV erreicht werden kann.

Als erster Schritt wurde das Projekt erstellen und alle Abhängigkeiten von OpenCV eingebunden. Die verwendete Version von OpenCV ist 3.1.0. Wichtig ist hierbei, dass EntwicklerInnen, welche das Projekt verwenden möchten, eine Umgebungsvariable „OPENCV_DIR“ erstellt hat, die in „\opencv\build\x64\vc14“ zeigt. Zusätzlich muss auch die PATH Systemvariable mit „%OPENCV_DIR%\bin“ gesetzt werden. Diese Instruktionen sind auch in der README.md Datei zu finden.

Der zweite Schritt lag beim Zugriff auf die Web-Kamera. Dabei stellt OpenCV eine VideoCapture Klasse zur Verfügung, welche dies ermöglicht. Durch den Zugriff auf die Kamera bekommt man jede Sekunde Bilder – sogenannte Frames - zur Verfügung gestellt, mit denen alle weiteren Verarbeitungsschritte durchgeführt werden. Falls kein Zugriff zur Kamera besteht, wird das Programm mit einer Fehlermeldung abgebrochen.

Danach war die Herausforderung die Pupillen zu erkennen. Um den Bereich der Pupillen zu erkennen wurde als Erstes eine Gesichtserkennung und danach eine Augenerkennung angewandt. Dafür bietet OpenCV die Klasse CascadeClassifier, welche mithilfe von HAAR oder LSB Classifier die definierten Bereiche erkennt. Es wurde entschieden, in diesem Projekt mit 3 unterschiedlichen Classifier zu arbeiten. Alle verwendeten HAAR Classifier für dieses Projekt, werden standardmäßig von OpenCV zur Verfügung gestellt und wurden nicht modifiziert. Hierbei wurde für das Gesicht die Datei „haarcascade_frontalface_alt.xml“, und für die Augen jeweils „haarcascade_lefteye_2splits.xml“ und „haarcascade_righteye_2splits.xml“ verwendet. Der Grund für jedes Auge einen eigenen Classifier zu verwendet ist, dass die Erkennung stabiler läuft und die Zuweisung der Daten für die spätere Analyse vereinfacht wird.

Der schwierigste Teil war das genaue Erkennen der Pupillen in einem spezialisierten Bild, indem nur noch die Augen zu erkennen sind. Dazu wurde das Ergebnis, welches von der Augenerkennung geliefert wird verwendet und der Ausschnitt verkleinert, damit störende Faktoren wie Augenbrauen abgeschnitten werden. Wir haben uns dazu entschieden, dass wir die Erkennung der Pupillen mittels Kreiserkennung (OpenCV Funktion: HoughCircles) durchführen. Dazu muss jedoch das Bild zuerst verschiedene Schritte durchlaufen, damit die Kreiserkennung erleichtert wird. Um dies zu ermöglichen wird das Rauschen herausgefiltert und in eine Binary (Schwarz/Weiß) Darstellung umgewandelt. Dabei wird die Pupille in Schwarz und der Rest wird in Weiß dargestellt. Der weiße Teil bildet damit den uninteressanten Bereich für die Kreiserkennung.



Dass Problem bei der Pupillenerkennung ist der Einsatz von verschiedenen Verarbeitungsprozessen, die jeweils andere Parameter benötigen. Dadurch funktioniert die Erkennung nicht immer Zuverlässig, sondern hängt stark von den Lichtverhältnissen der Umgebung ab. Daher ist die Voraussetzung für den Einsatz des GazeTrackers ein gut beleuchteter Raum.

Außerdem wird im Hintergrund mit der implementierten Klasse ScreenCapture der gesamte Bildschirm aufgezeichnet. Diese Aufzeichnung wird später für die Analyse verwendet.

1.2 Status per Milestone

Meilenstein 1:

Der Meilenstein 1 wurde zeitgerecht begonnen und ist beinahe zu 100% beendet. Der Zugriff der Kamera funktioniert einwandfrei, was als Voraussetzung des ganzen Projektes ist. Die Gesichts- und Augenerkennung wurde implementiert und funktioniert schon relativ gut. Die Pupillen werden auch schon erkannt und die Position ermittelt. Hier gibt es noch Verbesserungspotenzial in Hinsicht auf verschiedene Lichtverhältnisse und Position (Da diese noch zu stark hin und her springt).

Das einzige, was noch nicht funktioniert ist die Erkennung, in welche Richtung die Person gerade blickt. Dieser Punkt wird somit in den Meilenstein 2 verschoben, da es hier hauptsächlich darum geht, den GazeTracker zu kalibrieren und den Auftreffpunkt des Blickes zu ermitteln.

1.3 Activities per Person

1.3.1 Report on working hours per person

In diesem Projekt wurde viel Wert gelegt, dass jede Person des Teams in die einzelnen Prozesse miteingegliedert wird und miteinander den Fortschritt vorantreibt. Sprich es gibt keine konkrete Rollenaufteilung, welche Aufgabe welche Person zu erledigen hat. Es wird auf die Erfahrungen von anderen Projekten von der Universität zugegriffen und das Management verbessert. Die Kommunikation hat über sozialen Medien und auf der Universität stattgefunden.

Meilenstein1 (100% = 100 PS)

PS = Personen Stunden

- Aufsetzen des Projektes und Zugriff auf die Kamera 20%
 - Bernhard: 10% (10 PS)
 - Michael: 10% (10 PS)
- Face and Eye Detection: 35%
 - Bernhard: 17,5% (17,5 PS)
 - Michael: 17,5% (17,5 PS)
- Pupil Detection: 45%
 - Bernhard: 22,5% (22,5 PS)
 - Michael: 22,5% (22,5 PS)

Statusreport (100% = 6 PS)

- Bernhard: 33,33% (2 PS)
- Michael: 66,66% (4 PS)

2 Risk Analysis

In jedem Projekt kann es vorkommen, dass zu einem Meilenstein oder Arbeitspunkt Risiken auftreten können die den weiteren Fortschritt verhindern. Das kann zur Folge haben, dass das ganze Projekt nochmals überarbeitet werden muss (Struktur, Bauteile von Programme neu schreiben, etc.). Außerdem können einzelne Komponenten vom Programm nicht richtig miteinander funktionieren, was jedoch erst zum Schluss getestet werden kann und damit ein hohes Risiko darstellt.

Das größte Risiko ist es, dass beim Testen die Augen nicht erkannt werden können, wenn schlechte Lichtverhältnisse vorhanden sind. Hier muss sich der User darum kümmern, dafür zu sorgen, ausreichende Beleuchtung zu besitzen. Wenn die Lichtverhältnisse nicht geändert werden können, kann der GazeTracker nicht verwendet werden.

Des Weiteren ist der GazeTracker für nur eine einzige Person gedacht, der an dem PC arbeitet. Wenn aber mehrere Augen erkannt werden kann man die richtige Zuordnung nicht mehr gewährleisten.

Es könnte das Risiko bestehen, dass die derzeitige Version vom GazeTracker zu ungenau funktioniert für den User. Sei es, dass die Erkennung zu sprunghaft ist (von den Pupillen) oder wegen verschiedene Lichtverhältnisse (im Laufe eines Tages). Daher könnte es sein, dass eventuell bessere Erkennungsalgorithmen zu implementieren sind, die auf die verschiedenen Probleme genauer eingehen (Template von Augen, etc.).

3 Outlook

3.1 Milestones and Schedules

1. Meilenstein

- Zugriff auf Webcam -> 18.04.2016 -> **erledigt**
- Face/Eye Recognition -> 01.05.2016 -> **erledigt**
- Ermitteln der Augen Position und Blickrichtung -> 13.05.2016 -> Ermitteln Augenposition ist **erledigt**, Blickrichtung zu M2 verschoben aufgrund der Zusammengehörigkeit

2. Meilenstein

- Berechnung des Auftrittpunktes auf dem Bildschirm -> 25.05.2016

3. Meilenstein

- Implementierung der Showcase für GazeTracker -> 17.06.2016

4. Meilenstein

- Fertigstellung des Projektes -> 01.07.2016



3.2 Planned Effort per Person

Abschnitt	Beginn	Ende	Dauer in h
Anforderungsanalyse	12.03.2016	08.04.2016	20
Meilenstein 1	09.04.2016	13.05.2016	100
Meilenstein 2	14.05.2016	25.05.2016	80
Meilenstein 3	26.05.2016	17.06.2016	50
Meilenstein 4	18.06.2016	01.07.2016	50

4 Feedback

Da in der ersten Abgabe nicht genau spezifiziert wurde, wie unser Showcase aussehen soll, wird in diesem Abschnitt genauer eingegangen.

Unser Tool soll hauptsächlich Analysieren, wo der User auf dem Bildschirm hinsieht. In unserem Showcase soll die Amazon Seite analysiert werden. Generell User, die etwas Bestimmtes kaufen möchten, gehen sehr zielstrebig vor und werden nicht viele andere Produkte näher betrachten. Unsere Zielgruppe sind jene, die keinen speziellen Kaufwunsch haben, sondern einfach erkundigen, welche neuen Produkte und Aktionen es gibt. Sprich er surfte ohne Ziel auf dieser Seite herum. Analysiert werden soll dann, wo der User den Bildschirm am meisten betrachtet hat und ob dabei irgendwelche Auffälligkeiten zu finden sind. (Hat User ein spezielles Produkt länger betrachtet? Wenn ja warum? Möglichkeiten: Produkt hat herausgestochen, User hat Interesse, etc.).