

# Visual Detection – Entwurf:

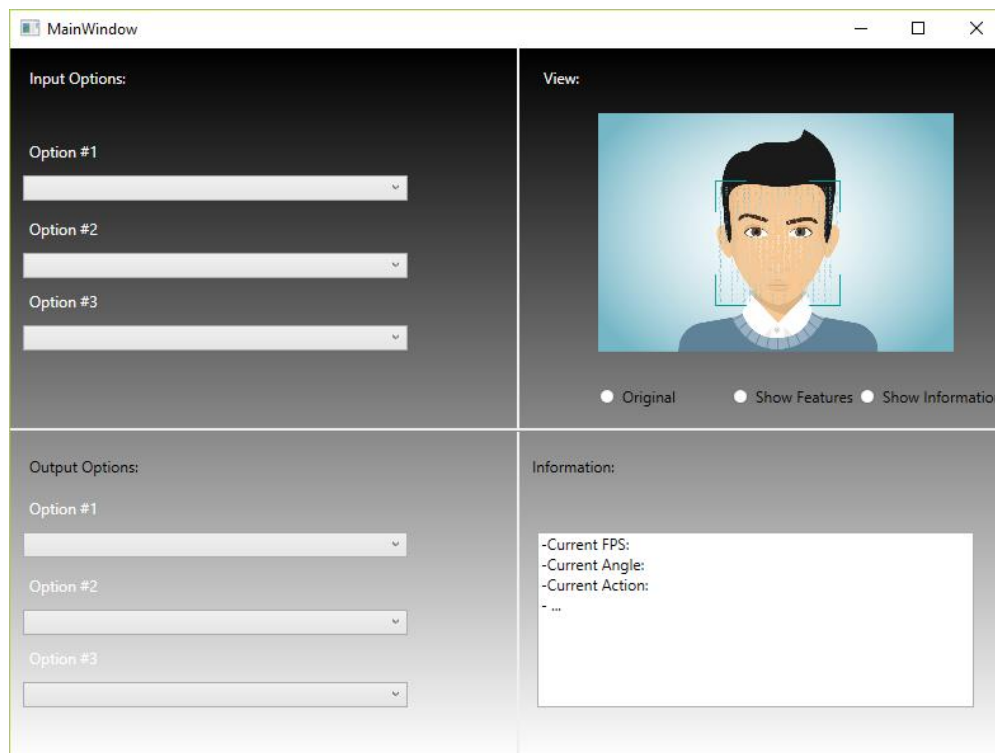
## 1. Kurzbeschreibung

Es wird eine Software geschrieben mit der Kopfbewegungen des Nutzers, die mit einer Kamera aufgenommen werden, detektiert und somit verschiedene Aktionen getriggert werden können. Die Software besteht aus einer kompakten GUI, welche dem User eine Anzeige des aktuell bearbeiteten Kamerabildes anzeigt sowie verschiedene Optionen und Informationen über den aktuellen Programmzustand zur Verfügung stellt.

## 2. Verwendete Programmiersprache, Library,...

Die Software wird in einem WPF-Projekt in C# geschrieben. Zur Verwendung kommt dabei EMGU CV, ein OpenCV Wrapper für C#.

## 3. Programmbeschreibung:



### 3.1 Input Options:

Hier kommen alle Optionen für die Detektierung und Verarbeitung des Bildes. Zum Beispiel sind hier unter anderem folgende Einstellungen geplant:

- Skalierung/Auflösung: Falls das Kamerabild auf eine gewünschte Auflösung skaliert werden soll, bevor es verarbeitet wird (Performancefrage)
- Auswahl des Algorithmus zur Detektierung (zb. SIFT, SURF...)
- Kamera (falls mehrere zur Verfügung stehen)
- Einstellungen zur Verfeinerung der Detektierung (Kantenschärfe, Kontrast,...)

### 3.2 View:

Hier sollte das aktuelle Kamerabild zu sehen sein. Dazu gibt es mehrere Auswahlmöglichkeiten, wie zum Beispiel:

- Originalbild ohne Bearbeitung
- Bild mit eingezeichneten gefundenen Features
- Bild nach verschiedenen Bearbeitungsschritten.

### 3.3 Output Options:

Hier soll der User folgende Einstellungen tätigen können:

- Trigger (Kopf nach links/rechts neigen, etc. )
- Sensibilitäten (Wie viel Grad muss der Kopf geneigt sein, um Trigger auszulösen / wie lange)
- Aktionen, die nach dem Triggerevent ausgelöst werden:
  - Windows-Aktionen
  - Applikationen öffnen
  - Simulierter Tastendruck oder Kombination von Tastenanschlägen, bzw. Sequenz von Tastenanschlägen.

### 3.4 Information:

Hier sollen aktuelle Statuszustände des Programmes angezeigt werden (in Form einer Konsole):

- Performanceinformation (Bilder/Sekunde, Zahl der Threads/CPU-Kerne,...)
- Gefundene Features
- Ausgelöste Trigger/Aktionen
- Fehlermeldungen

## 4. Zeitplanung, Meilensteine

1. Framework bzw. Programmskelett: Ende KW38
2. Implementierung der Grundfunktion des Programmes (1 Algorithmus zur Auswahl, Triggerevent anzeigen): Ende KW40
3. Einbau der Trigger-Aktionen, Performance-Verbesserungen, Bug-Fixes und generelle Fertigstellung KW43
4. Fertigstellung der Dokumentation und Abschluss: KW 45.