

Dariusch Ochlast 2101626
Sven Blanck 2101594

AVPRG Dokumentation

Projekt: Hör dich fit

Beschreibung eines technischen Teilaspektes - Liegestütz-Zähler

Das Konzept unserer Anwendung soll den Nutzer dazu animieren sich mit körperlichen Bewegungen einen kleinen Bonus in Form von Musikspielzeit zu erspielen. Jede komplett ausgeführte Bewegung, wie z.B. Liegestützen, geben so weitere Sekunden, in denen die Musik weiterspielt.

Diese Funktion haben wir wie folgt umgesetzt:

Zu Beginn des Projektes dachten wir uns, dass wir die Applikation mit Hilfe der Microsoft Kinect umsetzen wollen. Nach Schwierigkeiten in der Erkennung des menschlichen Skeletts haben wir uns jedoch dagegen entschieden. Das Problem lag darin, dass sich die einzelnen Körperteile in der Liegestütz-Position zu sehr überlagert haben, um das Skelett ansatzweise zu erkennen. Nach Absprache mit Ihnen, Herr Plaß, haben wir unser Programm nun auf die ColorKeyer Applikation aufgebaut.

Zur Aufnahme des Videos benutzen wir die Webcam, welche am System angeschlossen ist.

Diese Kamera wird mit Hilfe der Klassen, der schon aus der Vorlesung bekannten Videoengine, aufgerufen. Es wird der colorKeyerHSV Prozessor ausgewählt, welcher die eigentliche Verarbeitungs-Instanz unserer Applikation darstellt.

Wir haben uns bewusst dafür entschieden die Farbe rot zu erkennen. Aufgrund des colorKeyer-Algorithmus können wir so die menschliche Haut ziemlich genau erkennen. Wir haben bei der Initialisierung der Applikation einige Werte von Werk aus gesetzt, um die Haut zu erkennen. Darunter fällt das die kleinen Regionen herausgefiltert werden, um nur die

Dariusch Ochlast 2101626
Sven Blanck 2101594

Region mit der größten Fläche, in unserem Idealfall den Kopf, zu erkennen.

Den Hue-Wert, also der Farbwinkel im HSV Modell, haben wir auf 0° gesetzt, was Rot entspricht.

Die Farbtoleranz haben wir auf 30° festgelegt, was bedeutet, dass der Algorithmus auch die Farbwerte von 330°-360° und 0° - 30° erkennt.

Die Transparenz Alpha haben wir mit 100% deklariert, da wir keine transparenten Farben erkennen wollten.

Den Weichzeichner median haben wir auf false gesetzt, weil wir nicht noch eine zusätzliche Last auf den Prozessor legen wollten, da wir mit Hilfe der maskSmallRegions Methode schon sehr gute Ergebnisse erzielt haben.

Opening haben wir auf false gesetzt, da wir dies nicht zwingend benötigten.

Der eigentliche Grund warum wir uns für festgesetzte Werte entschieden haben ist, dass wir dem Benutzer der Anwendung nicht mit Kontroll-Elementen überhäufen wollten. Für die eigentliche Entwicklung der Anwendung waren diese von Vorteil, um verschiedene Einstellungen auszuprobieren, jedoch sollte sich der Endbenutzer nicht mit ihnen in Kontakt treten sollen. Für diesen sollten die minimal Elemente zur erfolgreichen Nutzung des Programms ausreichen.

Kommen wir zur eigentlichen Zähl-Funktion:

Die Zähl-Funktion wurde von uns in der Methode checkPushupCounter() umgesetzt. Diese Funktion wird in der process Methode des colorKeyers aufgerufen, also zur Laufzeit ausgeführt.

Vom Prinzip her verarbeiten wir den Kamera-Input mit Hilfe des ColorKeyer Algorithmus und den gesetzten Variablen. Dann suchen wir uns die größte Region aus, in unserem Falle meist der Kopf der Person, und zählen, mit Hilfe der checkPushupCounter-Methode die Liegestützen. Wenn der Kopf erstmalig unterhalb der unteren Linie war beginnen wir den Zählprozess. Dabei muss der Kopf die obere Linie überschreiten, damit wir dem Benutzer die Spielzeit gutschreiben können.

Dariusch Ochlast 2101626
Sven Blanck 2101594

Zur einfacheren Bedienung kann der Nutzer bei der ersten Benutzung des Programms diese zwei Linien verschieben, um so die Höhe der Liegestütze mit der Entfernung von Kamera zum Kopf einzustellen.

```
void ColorKeyerHSV::checkPushupCounter(Mat& image, Point center){  
    if(center.y < image.rows/20*upperLine){  
        start = true;  
        if(wasDown){  
            pushups = pushups + 1;  
            wasDown= false;  
        }  
    }  
    if(start && center.y > image.rows/20*bottomLine){  
        wasDown = true;  
    }  
}
```

Code-Auszug 1: checkPushupConter Methode

In Code-Auszug 1 sehen wir unsere Zähl-Methode. Diese bekommt das verarbeitete Bild, nach dem maskieren, übergeben und einen Punkt center, der die Mitte der größten Region repräsentiert.

Mit Hilfe der von uns public gesetzten Variablen upperLine und bottomLine, welche die obere und untere Linie repräsentieren, können wir nun mit der y-Position des center-Punktes und der Bildhöhe und der auswählbaren y-Position der Linien wasDown auf false bzw. true setzen um dann bei einem true-Wert von wasDown den pushup-Counter zu erhöhen und wasDown wieder auf false zu setzen. Dies entspricht dem Tracking des Kopfes während einer Liegestütze.

Da wir nun die Erkennung der Liegestütze, den Counter und das verarbeitete Bild haben wandeln wir das zuvor in Graustufen konvertierte Bild zurück in den RGB-Farbraum, um die Linien direkt auf dem Input Kamera-Frame anzuzeigen.

Es war zunächst sehr schade, dass wir das Projekt mit der Kinect nicht weiterführen konnten, dennoch haben wir, nach unserer Meinung, mit dem ColorKeyer ein gutes Resultat erzielt.