



Institut für Informatik
Lehrstuhl Human Centered Multimedia

Prof. Dr. Elisabeth André
Dipl. Inf. Ionut Damian
M. Sc. Tobias Baur
M. Sc. Dominik Schiller

Email: mi@hcm-lab.de



Multimodal Interaction, SS 2017

1. Übungsblatt: “von Mäusen und Kanten” (10 Punkte)

Abnahme am 10.05.2017

Code *nicht später als 10 Uhr* per Email an: mi@hcm-lab.de

Im Praktikum haben Sie Grundlegendes zur Signalverarbeitung und dem SSI (Social Signal Interpretation) Framework gelernt. Auf diesem Aufgabenblatt werden wir versuchen, dies praktisch anzuwenden.

Zur Abgabe: Bitte nur Solution/Projekt Dateien (d.h. *.sln/*.vcxproj/*.vcxproj.filters) und Header/Source Dateien (d.h. *.h/*.cpp) per Email schicken (die Größe des Anhangs sollte dann nur ein paar Kilobytes groß sein)!

Zur Bewertung: Falls die Abgaben Fehler enthalten, d.h. nicht das tun, was sie tun sollen, bitte dies unbedingt angeben. Für die maximale Punktzahl muss das Programm nicht nur das Richtige tun, sondern sollte sich auch korrekt beenden lassen, keinen offenen Speicher zurücklassen und der Code muss leicht verständlich sein.

Aufgabe 1: Hit the Edge (5 Punkte) _____

Implementieren Sie die Klasse `HitTheEdge`, die von `IConsumer` ableitet. `HitTheEdge` erwartet einen Stream mit 2-D Mauspositionen. Berechnen Sie aus diesem Stream wie oft der Nutzer die Bildkanten (oben, unten, links und rechts) innerhalb eines Zeitraums von 8 Sekunden trifft (z.B. x ist ungefähr 0). Die entstehenden Werte können Sie einfach auf der Konsole ausgeben.

Tipps:

1. falls Sie die Frame size auf 8 Sekunden (“8s”) in der Pipeline setzen, erhält `HitTheEdge` immer 8 Sekunden an Daten.
2. wenn bei dem Mouse-Sensor die Option `scale` auf `true` gesetzt ist, wird der Koordinatenbereich automatisch zwischen 0 und 1 skaliert.

Aufgabe 2: MouseFeatures (5 Punkte)

In dieser Aufgaben sollen Sie die Klasse `MouseFeatures` implementieren. Diese wird der `HitTheEdge` stark ähneln, jedoch soll sie nicht von `IConsumer` sondern von `IFeature` erben. Grund hierfür ist dass anders als bei `HitTheEdge`, die berechneten Werte auch als Stream weitergeleitet werden sollen.

Genauer müssen Sie folgende Features aus den Mauspositionen eines Zeitraums von 1 Sekunde berechnen:

- Frame Minimum, Maximum und Durchschnitt (pro Koordinate)
- Globales Minimum, Maximum und Durchschnitt (pro Koordinate)
- Länge der hinterlegten Strecke (Frame, 2D)
- Geschwindigkeit (Frame, 2D)
- Varianz (Frame, 2D)
- Standardabweichung (Frame, 2D)

Zusätzlich, soll die Klasse erlauben mit Hilfe von Optionen wählen zu können, welche Features berechnet werden sollen. Dies machen Sie am Besten indem Sie eine zusätzliche Klasse implementieren die von `OptionList` erbt. Zum Beispiel:

```
class Options : public OptionList {

public:
    Options () {
        min = true;
        max = false;
        addOption ("min", &min, 1, SSI_BOOL, "compute minimum");
        addOption ("max", &max, 1, SSI_BOOL, "compute maximum");
    }
    bool min, max;
};
```

Tipps:

1. die Gesamtdauer eines Frames kann folgenderweise berechnet werden: $d = \frac{1}{sr} \times num$
2. sie haben Zugriff aufs gesamte SSI repository, inklusive verschiedener `plugins` die Sie als Muster oder Inspiration benutzen können.