Presentation skills automatic real-time feedback mechanisms

Date: November 10, 2014

Project members:
A.P. van Ree (10547320)
G. Linders (10527605)
T.M. Meijers (10647023)

1 Goals of the project

Goal as given in the project description:

"The goal is to develop a system able to give feedback usable for performance improvement of novice speakers on verbal and nonverbal skills. Through the interpretation of one or all of the following components: pose, gesture, prosody, intonation, pitch and speech rate performance grades should be given related to a predefined rubric and metrics (comparable with text editor generated readability statistics)."

To achieve the goal mentioned above, the project is split up in different parts. The projects end goal is to deliver a piece of software that can autonomously give a user (e.g. a student) feedback on his or her presentation skills.

To design such a system the research phase focussed a lot on what is important to know and to do during presentations for novices. The literary research was split up between audio and visual. The goal is to find out based on which actions a person giving a presentation can be easily given feedback, and if these actions are good or bad during a presentation.

Once this is known, the next goal is to implement smaller experiments to see wether it can be realized to implement these checks through algorithm. Our personal goal is to focus on the most important aspects which are large movements, gestures and vocal use.

Once these experiments are done and it is clear what can be used, the end goal is to bring these different experiments together in a piece of software that could work in conjunction with equipment like a Microsoft Kinect.

2 Work done towards goal

At the start of the project the goals we had in mind were a lot harder to achieve than the basics of what is important during a presentation and doing simple experiments to check if this can be achieved.

We wanted to start these experiments with a Kinect right away, and in the end wasted a lot of time on installing drivers for the Kinect, finding libraries to work with and trying different operating systems (Microsoft, Linux). We have since switched to experimenting with simple devices, such as using a webcam to take a series of pictures and analyze these for movement.

We had chosen to work with OpenCV, due to it being the largest opensource computer vision library. First we decided to work with Python due to the NumPy library, which makes working with arrays (images) a lot easier. The problem was however, these Python is a lot less well documented. This made us switch to Java as this language has better documented libraries for the kinect itself and OpenCV.

2.1 Visual

We have done research in what the most important non-verbal aspects are in giving a presentation. From different books and papers about presentation skills we have made a selection about what we will try to give feedback on. These aspects are:

- Posture:
- Gestures
- Eye contact

Another important aspect which we have left out are facial expressions. This is a very difficult and time-consuming task and is not the main focus of our research. Posture and gestures can be measured using various motion detecting algorithms.

We have implemented such an algorithm which detects motion. As we assume that only one person is in front of the camera in front of a static background, the only noise we will have to take in account is is light (e.g. reflections and flickering). The implemented algorithm filters out this noise, and uses a three frames to further filter noise. With this we can obtain an image which shows only the parts which show movement.

2.2 Audio

The easiest way to analyse sound is using the *Fast Fourier Transform* (FFT). This algorithm gives us the frequencies of the speakers voice, and helps us to determine whether a speaker has a lively voice. As research, we have found some papers that are about the analysis of sound when presenting. The most important is the standard of someone's voice frequency and its mean. With this a certain variable can be calculated to give a indication as to how a presenter is speaking. If this variable is too low, then a speaker should get a low rating on their speaking skills.

Other things that are important in these researches are tempo, volume and intonation. As for the first two; these can easily be measured. The third is a bit harder to do.

There are a lot of open source programs usable for FFT. We have found several programs in Java

that can find the FFT in a .WAVE file. We have been tryin to find the one that suits our research best and trying to understand how the program actually works.
This turns out to be harder than it seems, because to understand the code, we need to understand the exact math behind the algoritm, which is quite hard. We need this understanding to know how to use the output and do calculations with it.

3 Plans to achieve goal

3.1 Visual

As discussed in the section above we need to split up the visual part in multiple sub experiments. We have to focus on gestures, so detecting hand movements is an essential part of this. Furthermore gesture can be detected with the algorithm that is already implemented, but we have to find a way to only detect movement in ones gesture, and filter out smaller movements (hand gestures etc.). For the third point, eye contact, we want to experiment with an eye detection algorithm, and if we can detect movement of the eyes position it is clear that the user is not constantly focusing on one point.

Having these three subjects well implemented gives us a basis on which we can give a user feedback on his or her presentation skills. If this is done we have to assign a certain rating to the values distilled out of these three subjects.

3.2 Audio

In the coming two months, we need to find one FFT-code in Java that works the best for us. When we have this, we can start analyzing the monotony of a presenters voice.

Antoher thing to do is to find a way to calculate someones speech tempo. A way to do this is to count the amount of syllables per time rate, and then give this a rating. Which pace is too slow or too fast, and what is nice to listen to.

Lastly, there is the volume of a speakers voice. We read in a lot of presenting guides that a presenter can rarely talk too loud. We simply have to measure the decibels, and find a threshold that indicates when a person is talking too quietly.

3.3 Bringing the two together

The splitting up into two sections makes the implementing easier in the beginning. As we go on we want to try and find a way to combine the two together, and give a presenter an overall grade on their presentation skills. All of the smaller experiments should lead to us gaining more knowlegde about the overall project, little by little.