SSI/ModelUI - A Tool for the Acquisition and Annotation of Human Generated Signals

Johannes Wagner[1], Elisabeth André[2], Michael Kugler, Daniel Leberle

[1 2] *Multimedia Concepts and their Applications, University Augsburg, 86159 Augsburg, Germany*

Email: [wagner,andre]@informatik.uni-augsburg.de

Introduction

Humans are used to express their needs and goals through various channels, such as speech, mimic, posture, *etc.* The recognition and understanding of such behaviour is a key requirement towards a more natural human-computer interaction (HCI) and believed to be an important part of next-generation user interfaces [1]. To achieve this, we need to give the computer access to human generated signals and provide adequate models to recognize and interpret the behavioural patterns. Ideally, processing should happen in (near) real-time in order to allow an application to show immediate reactions.

At our lab we have developed Smart Sensor Integration (SSI), a framework for multimodal signal processing in real-time. It allows the recording and processing of human generated signals in pipelines based on filter and feature extraction blocks. By connecting a pipeline with a classifier it becomes possible to set up an online recognition system. However, before a classifier becomes usable, a model has to be trained for the recognition problem. This is usually a time-consuming task and involves several steps, including sample recording, annotation and training. Since the careful accomplishment of each task is crucial for the success of a classifier, we have developed a tool called ModelUI, which supports these steps.

ModelUI is a graphical user interface (GUI) that runs on top of SSI in order to support the following tasks: 1. record and manage a large number of human generated signals, 2. observe how user behaviour is expressed in the signals, 3. evaluate how well different machine learning algorithms allow the recognition of the observed patterns, and 4. train a model that can be used with an online classifier to recognize behavioural patterns in real-time.

Smart Sensor Integration

In [2] we have introduced SSI as a general framework for the integration of multiple sensors into multimedia applications. In particular SSI supports the pattern recognition pipeline by offering tailored tools for data segmentation, feature extraction, and pattern recognition. In fact, SSI has been successfully applied in a number of projects under the grant of the European Union, which are concerned with the analysis and development of novel user interaction methods [3].

From our experiences in these projects we have learned that the performance of a model is greatly enhanced if trained on data collected in a context similar to the final application. This encouraged us to develop a GUI, which allows even unskilled users to create what we call ”personalized” models. In addition, we have included the possibility to play around with different machine learning algorithms and tune them until a suited configuration is found. In this way, SSI/ModelUI combines the functionality of an annotation tool, such as Anvil[1], and a machine-learning tool, such as WEKA[2], into a single application. The possibility to process live input from multiple sensors makes it particularly suited for the processing of human-generated signals.

Data Aquisition

SSI offers two ways for connecting a sensor device: a direct connection, which requires implementation of an abstract sensor class, or via socket communication. The first has been applied to a number of sensor devices, such as microphone, camera, or haptic and physiological sensors, which SSI supports by default, but can be applied to any new sensor device, too. The socket interface offers a convenient way to capture data from devices, which are connected to other machines in the network.

Once the connection to a sensor device has been established it is plugged to a pipeline, which stores the captured signal streams to disk. If necessary pre-processing filters can be interposed, e. g. to remove artefacts or apply compression to the streams. If multiple modalities are connected, SSI takes care that data streams are kept synchronized.

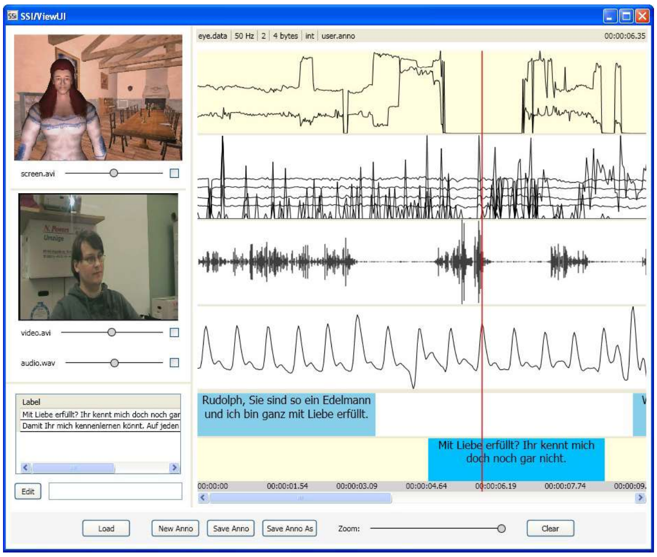
Recognition Pipeline

For each signal type, a number of possible recognition pipelines can be defined. Each pipeline consists of a series of cascaded filter/feature blocks, which transform the raw signal into the format required by the machine learning algorithm that is located at the end of the pipeline. Learning algorithms, which require the same or a similar format, can share a pipeline.

In SSI a pipeline can be used in two ways: online, i. e. with live input, or offline, i. e. from pre-recorded signals stored on disk. In the non-continuous case, where recognition is based on certain actions in the signal, e. g. when the user speaks or performs a gesture, we have to define when certain actions occur. In the offline case this is given by the annotation; in the online case, however, we use trigger to automatically detect interesting segments.

[1] http://www.anvil-software.de/

[2] http://www.cs.waikato.ac.nz/ml/weka/



Mit Liebe erfüllt? Ihr kennt mich doch noch gar

Damit Ihr mich kennenlernen könnt. Auf jeden

Rudolph, Sie sind so ein Edelmann und ich bin ganz mit Liebe erfüllt.

Mit Liebe erfüllt? Ihr kennt mich doch noch gar nicht.

Figure 1: Recorded sessions are visualized together with different annotation tracks that describe the observed user behaviour. The screenshot shows four signals (top down: eye gaze, head tracking, audio and blood volume pulse) and two annotation tracks (here: the transcription of the dialog between the virtual character and the user). On the left side videos of the application and the user are displayed.

Annotation Files

An annotation file contains a list of time intervals with according label information. Intervals with the same label belong to the same class. During training a model is built, which separates the feature space into regions representing the different classes within the annotation. Unseen samples are mapped on the feature space and assigned the class label that belongs to the region they land in.

ModelUI

The GUI called ModelUI assists a user to undertake the different tasks a pattern recognition problem is concerned with. However, the interface only manages and displays the data, while the actual processing is left to SSI. Hence, the interface is not restricted to a certain recognition problem.

Recording

Each recording is stored as a new session and inserted in the database of the current project. During a recording a stimuli can be presented to the user in form of a series of html pages, which may include textual instructions, but also images or videos. Additionally the screen (or parts of it) can be captured in order to create a video as reference during annotation.

During a recording the triggers in the pipelines are used to generate pre-annotations, i. e. to mark segments in the signals with activity. A detected segment receives the label of the current stimulus. These pre-annotations can extremely speed up the annotation process. Likewise, events generated by the application are also collected and inserted as additional annotation tracks.

Annotation

Recorded sessions can be visualized together with annotation tracks that describe the observed user behaviour. Figure 1 shows how signals and annotations are horizontally aligned along a common timeline. Videos are separately displayed. A recording can be played back from any position and annotation blocks can be added or adjusted using mouse and keyboard operations.

Training

Given a signal and an annotation track, features can be automatically extracted using the available pipelines. In this case, a feature vector (or - depending on the type of learning algorithm - a series of low-level descriptors) is calculated for each segment in the annotation track and stored to disk.

The train panel lists the extracted features together with suited learning algorithms. The user can select the whole feature set or any subset. Different evaluation methods are available to test the recognition accuracy of a model. When a model is trained, the number of classes is automatically derived from the set of distinctive labels. Finally, a model can be tested with live-input, too.

Conclusion

We have presented SSI, a framework for multimodal signal processing, and ModelUI, a GUI for data acquisition, annotation and model training. Throughout the paper, we have described why the tools are suited to analyze human generated signals and how they are used to train models that can automatically recognize behavioural patterns of human users.

SSI and ModelUI are freely available from:

`http://mm-werkstatt.informatik.uni-augsburg.de/ssi.html`

Acknowledgements

The work described in this paper is funded by the EU under research grants CALLAS (IST-34800), IRIS (Reference: 231824) and Metabo (Reference: 216270).

References

[1] M. Pantic, A. Nijholt, A. Pentland and T. S. Huanag, "Human-Centred Intelligent Human Computer Interaction (HCI^2): how far are we from attaining it?", in International Journal of Autonomous and Adaptive Communications Systems, 2008, pp. 168–187.

[2] J. Wagner, E. André and F. Jung, "Smart sensor integration: A framework for multimodal emotion recognition in real-time", in Affective Computing and Intelligent Interaction (ACII 2009), 2009.

[3] J. Wagner, F. Jung, J. Kim, E. André and T. Vogt, "The Smart Sensor Integration Framework and its Application in EU Projects", in Workshop on Bio-inspired Human-Machine Interfaces and Healthcare Applications (B-Interface 2010), 2010, pp. 13-21.