HHI Video Recognizer

Contents

1 Introduction

In this document the main features and functionalities of the ***HHI-VideoRecognizer*** are described. It contains a description of the provided annotations and offers a step by-step approach on how to install and use the software package.

This recognizer performs automatic analysis of human motion and provides a large variety of annotations. It is currently developed for single person and two person scenarios. Some of the video analysis algorithms are constrained to single person scenario or other constraints such as frontal view.

The analysis tool is based on skin colour in order to detect and track robustly the hands and the face of persons. An automatic skin colour estimation module is included in the recognizer, but we also provide a separate tool, which allows the setting of skin colour parameters in an interactive manner based on a graphical user interface. This tool is described at the end of this document.

The video recognizer can be run as a separate tool from command line or as part of the annotation tool ELAN. Details on how the tool is applied can be found in the following sections.

The video recognizer works with all the most common video codecs (such as *mpeg1-2*, *x264*, *DivX* …) and formats (*mpg*, *mpeg*, *avi*, *mp4*). A detailed list of supported codecs is available on FFmpeg website. However for some videos, due to wrong encoding options, problems may occur. In this case, a conversion to another standard video format or a re-encoding of the video is recommended. The video analysis tool is designed for being independent on the resolution. The most common resolutions such as CIF, SD video and HD video have been tested and the results should be almost independent of the resolution. However, better resolution will provide more accurate results but is computationally more expensive. The performance of the video recognizer is approximately real-time (i.e. one hour video requires one hour of processing) for SD resolution (720x576 pixels), while it processes 12 frames per seconds (i.e. a 10-minutes video requires 20 minutes of processing) in case of videos with higher resolution (1280x720 pixels). The effective performance strongly depends on the PC where the application is running, the number of people in the video, some of the parameters used (such as background detection) and of course the video resolution and frame rate.

2 Automatic annotations

The following annotations rely on an accurate estimation of skin colour and a robust and reliable detection, tracking and labelling of hands and the face. In order to achieve this, a lot of pre- and post-processing is done, in order to analyse the scene correctly. The aim is to cover as many as possible scenarios happening in the field of language and gesture research. The annotations described as follows are currently implemented both for a single person and a multi-person (usually 2 people) scenario. The single person scenario is the one which gives more robust and accurate results, though.

The next sub-sections describe the annotation results based on correctly tracked and labelled moving hands. The annotations created are based on the Module 1 and Module 2 of the Neuroges Coding Manual. A more detailed description of Neuroges and its use within ELAN can be found in (Lausberg & Sloetjes, 2009).

2.1 Hand motion analysis - Activation

The hand motion analysis is performed on a temporal segment in which a hand is moving a significant amount. The threshold that defines the required motion can be set by the user (see section 4). Three different values (“low”, “normal”, “high”) can be chosen, which define the necessary amount of motion to decide for the begin and end of a movement. To correctly determine the beginning and end of a movement the definition of *rest position* is required. For rest position we mean the place the hands stay when no arm or hand muscles are activated. The rest position is also the region where every movement starts and ends: in many cases during an on-going movement the hand is completely still, in an anti-gravity position, but that movement will end only when the hand goes back to its rest position.

For each person in the video, the movements of both hands are stored. The software distinguishes between two different types of movements: the ones that involve a change of the position of the hand (saved in the "Activation" annotation) and the ones where the position in space of the hand does not change, albeit there is some kind of motion (moving the finger, closing the hand, etc.). These are the so-called *intrinsic movements* and they are saved in the annotation "Activation_Intrinsic". The detection of intrinsic movements is performed only when the hands are inside their rest position.

In Figure 1, a typical result is shown. The temporal segments assign when a motion has been identified. The annotations are created for both left and right hand, and the movements are labelled differently, depending on the type of motion. The annotations "Activation" and "Activation_Intrinsic" are mutually exclusive, so there is no overlap possible between the two.

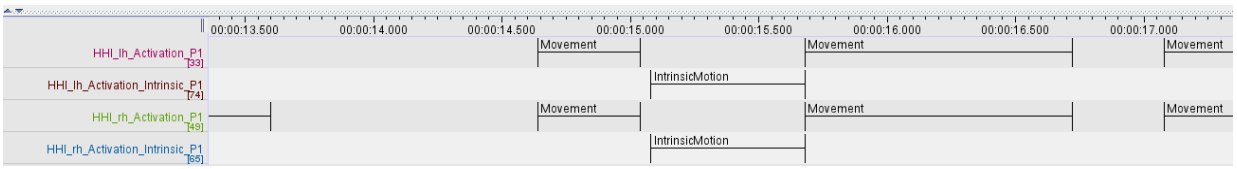

Figure 1: Example for annotation of hand motion analysis

The tracker should automatically recognize if only the hands of the person are detected ("long-sleeves scenario" or if the arms are visible as well ("short-sleeves scenario"). In the latter case, the recognizer will try to distinguish arm from hand in order to better track the position of the hand along the timeline.

2.2 Positional analysis of hand motion - Focus

For each detected movement, an analysis of its position in space is performed. The software analyses, for the whole length of the movement the position of the hand relative to the position of the head. Based on that information, the movement is divided into one or more sub-segments which describe if the hand position is either *on body* (acting on the body surface) or *in space* (acting in space without touching something). The Neuroges coding manual specifies four more possible positions (*within body*, *on person*, *on separate object*, *on attached object*), but they have not been implemented yet.

In Figure 2, a typical result for the positional analysis of person 1 is shown.

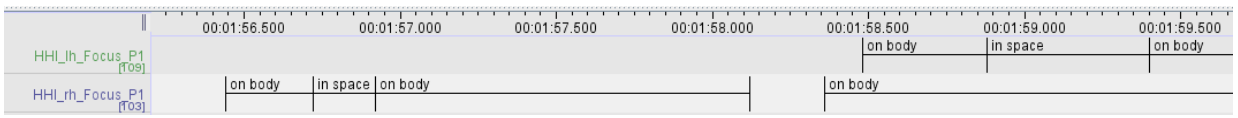

Figure 2: Example for positional information in temporal segments, where hand motion has been detected

2.2.1 Positional analysis of hand motion – Location

If the flag "*use_mc_neill_output*" is set to *true*, then a new annotation based on the positions of the hands is created. In contrast to the relational analysis, an assignment of the absolute position of the hands is of interest. Hence, for the case of a frontal view of the person the

commonly accepted definition of the gesture space by McNeill has been used for annotation (see Figure 3).

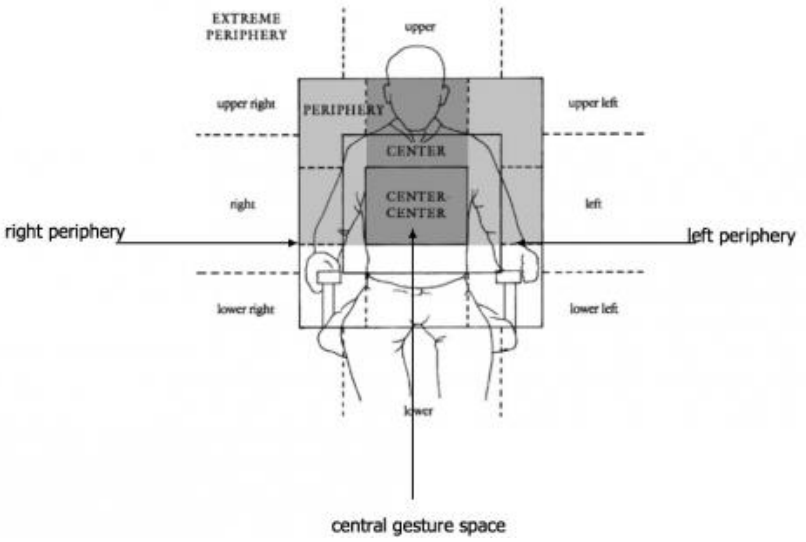

Figure 3: Gesture space definition according to McNeill (1992)

As the above definition of the gesture space is one out of many possibilities, the implemented definition of the gesture space has been simplified to some extent. The following annotations are possible in the current version of the video recognizer as shown in Figure 4 overlaid on the definition by McNeill.

The position as well as the size of the CENTER_CENTER space is defined based on the distance between the eyes, which is continuously estimated. The outer CENTER space is twice the dimension of the CENTER_CENTER space.

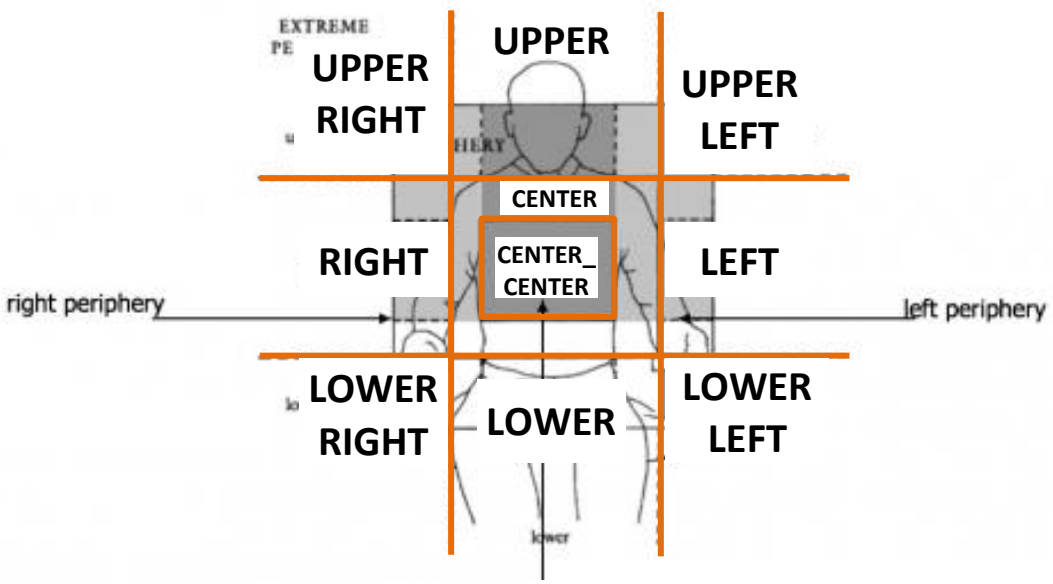

Figure 4: Revised gesture space definition in the HHI video recognizer

In Figure 5Figure 7, a typical result for hand location analysis of left and right hand of person 1 is shown.

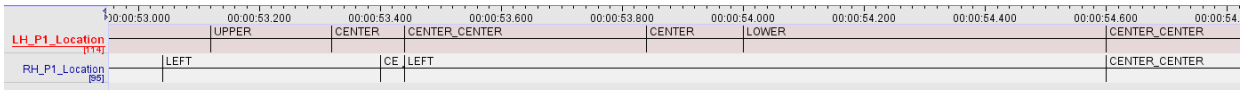

Figure 5: Example for annotation results from hand location analysis

2.3 Relational analysis of body parts - Contact

The same information used to perform the positional analysis of hands is used to perform the analysis of the relative motion of the hands. When both the hands are moving, a new annotation ("Contact") specifies whether the two hands are *in touch* or they *act apart*. The Neuroges coding manual distinguishes two possible cases when the hands are touching: *act on each other* and *act as a unit*, but this distinction has not been implemented so far.

In Figure 6, a typical result for relational analysis of left/right hand and head of person 1 is shown.

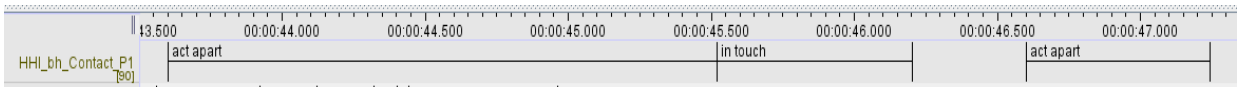

Figure 6: Example for annotation results from relational analysis

2.4 Structure of the movement analysis - Structure

The Neuroges coding manual classifies each movement as belonging to one of five categories: *phasic*, *repetitive*, *irregular*, *shift*, *aborted*. The recognizer currently distinguishes only the first three types of movements. The distinction is performed based on the directional analysis of the movement. Each movement is divided into one or more sub-segments describing the direction of the motion (the possibilities are UP, DOWN, LEFT, RIGHT or SMALL_MOTION, if the amount of motion is small and without a clear direction). Base on this sub-segmentation a movement is labelled as *phasic* if there is a path (for example UP->DOWN) traversed at most once; as *repetitive* if there is a path traversed at least twice (for example LEFT->RIGHT->LEFT->RIGHT); otherwise as *irregular*. Currently all the movements belonging to the "Annotation_intrinsic" annotation are marked as *irregular*. It is important to note that the definitions used for *phasic*, *repetitive* and *irregular* are different from the ones used in the Neuroges Coding Manual. One difference that is worth mention is that, according to our definition, all the intrinsic movements (the ones in the Activation_Intrinsic annotation) are labelled as *irregular*, since they do not have clear spatial direction, but they may be of different type, according to the Neuroges definition.

In Figure 7, a typical result for structure analysis of left and right hand of person 1 is shown.

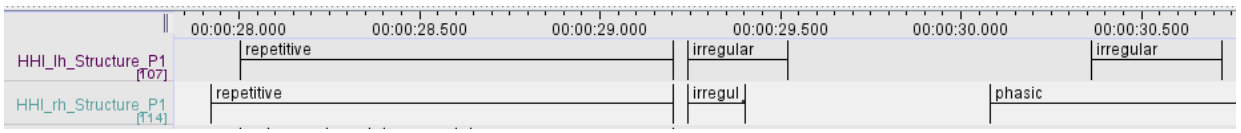

Figure 7: Example for annotation results from movement structure

2.5 Output format

The results are stored in an xml file whose name is specified by the user, either in a parameter file or directly from ELAN. The xml file uses the new multi-tier format supported by ELAN from version 4.5. Each tier contains one of the annotations described above. A naming convention is used to describe the annotation: At first there is one prefix, **HHI**, that indicates that the annotation has been created via the recognizer; then another prefix, **lh**, **rh** or **bh**, is used to specify if the annotation relates to left or right hand, or both; after that there is the specific name of the annotation (**Activation**, **Contact**, **Focus** or **Structure**); finally, the postfix **P1** or **P2**, to specify if the annotation refers to the first person or the second. In case there is more than one person in the video, person **P1** is the one on the left and **P2** is the one on the right.

3 Installation

The installation is performed by installer software. This installer takes care about the correct placement of executable, libraries and other files. If default installation options are chosen, the programs will be installed in `C:\Program Files\Fraunhofer HHI-VideoRecognizer\` and desktop shortcuts will be created. The installer will also take care of the installation of CodeMeterRuntime (required for the execution of protected software) and, if it's not already installed, the Microsoft VC++ redistributable. After the install procedure is completed the software is then ready to run. The first time the program runs, though, the USB dongle has to install the required driver (the procedure is automatic) before the program can be launched.

4 Executing the video recognizer

The HHI video recognizer can be run in two ways, either from command line or within ELAN.

4.1 Running from command line

The syntax is:

HHI-VideoRecognizer.exe –i parameters.xml

If no parameter file is provided, an alternate syntax is:

HHI-Video-Recognizer.exe –c [param1 value1 param2 value2]

In this case, all the mandatory parameters must be provided. The command

HHI-VideoRecognizer.exe –h

Provides some more information:

INFO: ***HHI-VideoRecognizer.exe***
INFO: Usage:
INFO: ***HHI-VideoRecognizer.exe --param-file filename.xml [--param-change name1 value1 name2 value2 ...]***
INFO: -i is the same as --param-file
INFO: -c is the same as --param-change

INFO: --param-file - uses stdin as input
INFO: --help or -h for info
INFO: Input parameter file is not necessary, if all mandatory parameters are specified with --param- change option.
INFO: If both --param-file and --param-change are used, the xml parameter file will be overwritten, when possible.
INFO: If "-i -" is used, then pipes are used (i.e.: HHI-VideoRecognizer.exe -i - < ListOfParam.xml)

HHI-VideoRecognizer application

This program reads the input video and the input xml containing the skin colour parameters. The skin colour parameter estimation is an important part of the overall analysis. It can be done manually by using the application **HHI-SkinColourEstimator**. The skin colour parameters are stored in a specific xml-file with a filename set by the user. If there are no skin colour parameters available, the ***HHI-VideoRecognizer*** performs automatic skin colour estimation, using the face pixels as training samples.

The ***HHI-VideoRecognizer*** detects and tracks arms, hands and heads in the video. Based on this information several annotations are performed on higher semantic level as described in section 2. The results are stored in a single xml output file containing all the different annotations. An output video can be produced as well containing some visualisation of detection and tracking.

Parameter used by the detector:

- input_video: MANDATORY. Specifies the input video. User MUST specify the full path.
- input_xml: OPTIONAL. This parameter is an xml file that contains information about the start and end of the video, as well as the skin colour parameters.
- output_video: OPTIONAL. Specifies the name of the output video.
- output_video_resolution: OPTIONAL. Specifies the resolution of the output video, compared to that of the input. Available values are "*full*", "*half*" and "*quarter*". Available in ELAN in the "advanced" tab.
- output_xml: MANDATORY. Specifies the output xml file containing the annotations.
- logging_level: OPTIONAL. Specifies the amount of log messages during processing. Available values are "*low*", "*normal*" and "*verbose*".
- background_detection: OPTIONAL. If true, use background information to improve the detection and tracking of head and hands. Available in ELAN in the "advanced" tab.
- speed_threshold: OPTIONAL. Specifies the threshold on the hands' speed when detecting start and end of a movement. The three accepted values are "low", "normal", "high". Available in ELAN in the "advanced" tab.
- use_rest_position: OPTIONAL. If set to true, the detection of the end of a movement has to be inside the rest position. Rest position is automatically detected for each hand and represents the position in space where the hands lie and no muscle is activated. Available in ELAN in the "advanced" tab.
- use_mc_neill_output: OPTIONAL. If true, annotations based on McNeill's gesture space will be provided in the output xml. Available in ELAN in the "advanced" tab.

4.2 Running the recognizer within ELAN

The application can also run from within ELAN (since version 4.5.0). In order to do so, copy the provided folder "*hhi-hand-head-tracking*" inside the *"extensions*" folder, located where ELAN has been installed (usually "*C:\Program Files (x86)\ELAN 4.8.1*"). The folder contains the recognizer file ("*recognizer.cmdi*") which contains all the information required to run the software from ELAN. The file assumes that the recognizer has been installed in the default directory. If this is not the case, open the *recognizer.cmdi* with a text editor, and change the "*runWin"* attribute (by default "*C:\Program Files\Fraunhofer HHI-VideoRecognizer\bin\HHI-VideoRecognizer.exe -i -*") to the path of the hand head tracker executable. For example, if ´the recognizer has been installed in the directory "*D:\Apps*", the attribute value will be "*D:\Apps**HHI-VideoRecognizer\HHI-VideoRecognizer.exe* *-i -*")

Now, open ELAN and go to the tab "*Video recognizer"*, as shown in Figure 8, and select the recognizer *"Human motion analysis and annotation – local version"*.

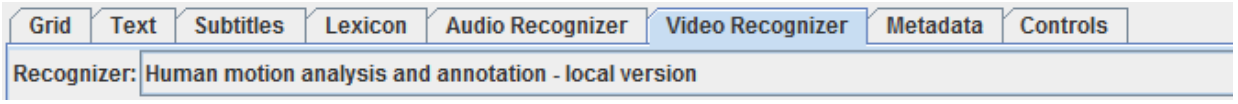

Figure 8: The tabs in ELAN

A list of parameters is then displayed:

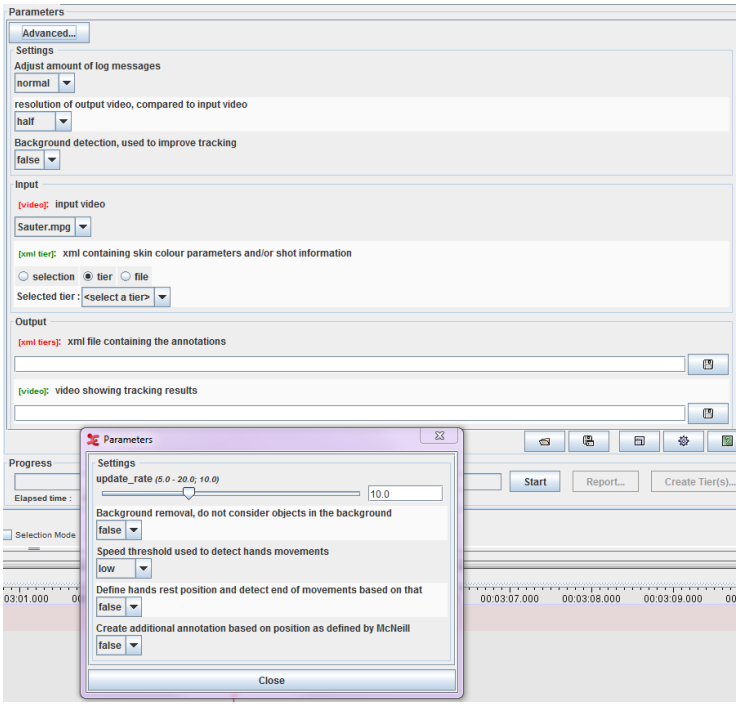

Figure 9: The list of parameters in ELAN

These are the same parameters as described before. The ones in RED are mandatory. The second parameter (*xml containing skin colour parameter and/or shot information*) is optional and can be created using the application ***HHI-SkinColourEstimator.exe*** (see below for details). The other parameters specify the amount of logging messages, the resolution of the output

video, the input and the output data. More advanced parameters can be set after clicking the "Advanced…" button. The list of parameters available in ELAN is shown in Figure 9.

5 Interactive skin colour estimation

If no input xml is specified, the program will automatically estimate the skin colour parameter before tracking hands and heads. In some cases though, especially in the case of low resolution, poor quality or non-uniform background the estimation can give poor results. In this case the user can manually estimate the parameters using the application ***HHI-SkinColourEstimator.exe***:

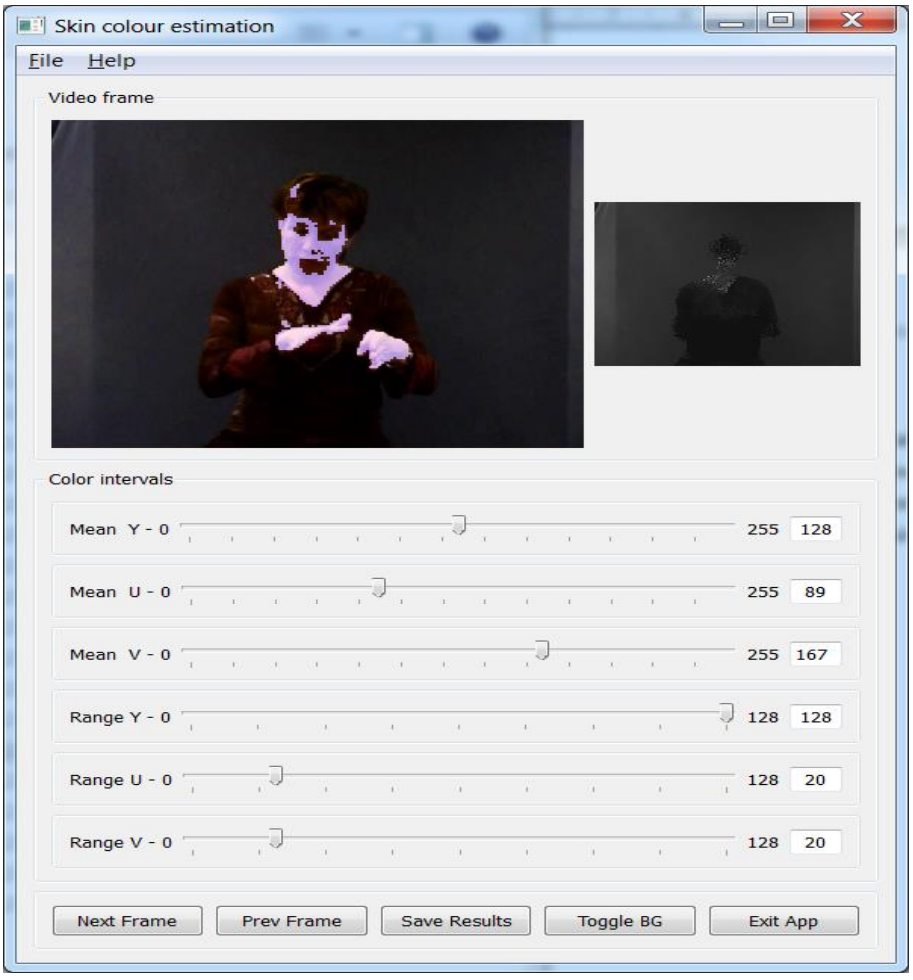

Figure 10: Screenshot of the GUI of HHI-SkinColourEstimator

The program graphically shows the segmentation in light violet (see Figure 10), moving the slider will modify the segmentation. The aim is to have light violet pixels only where the skin is. The "Next Frame" and "Previous Frame" buttons show the segmentation on different frames, the "Toggle BG" shows in black which pixels are marked as background (and won't then be considered by the **HHI-VideoRecognizer** application). Once the segmentation results are satisfactory, results can be saved on file with the "Save Results" button.

6 Bibliography

Lausberg, H., & Sloetjes, H. (2009). Coding gestural behavior with the NEUROGES--ELAN system. *Behavior Research Methods*, 41 (3), 841-849.

For further questions, please contact

Stefano.Masneri@hhi.fraunhofer.de or *Oliver.Schreer@hhi.fraunhofer.de*