# Application of Video Processing Methods for Linguistic Research

**Przemyslaw Lenkiewicz[1], Peter Wittenburg[1], Binyam Gebrekidan Gebre[1], Anna Lenkiewicz[1], Oliver Schreer[2], Stefano Masneri[2]**

[1]Max Planck Institute for Psycholinguistics, Wundtlaan 1, 6525 XD Nijmegen, The Netherlands
[2]Fraunhofer-Heinrich Hertz Institute, Einsteinufer 37, 10587 Berlin, Germany
{Przemek.Lenkiewicz, Peter.Wittenburg, BinyamGebrekidan.Gebre, Anna.Lenkiewicz}@mpi.nl
{Oliver.Schreer, Stefano.Masneri}@hhi.fraunhofer.de

**Abstract**

Evolution and changes of all modern languages is a well-known fact. However, recently it is reaching dynamics never seen before, which results in loss of the vast amount of information encoded in every language. In order to preserve such heritage, properly annotated recordings of world languages are necessary. Since creating those annotations is a very laborious task, reaching times 100 longer than the length of the annotated media, innovative video processing algorithms are needed, in order to improve the efficiency and quality of annotation process.

**Keywords:** Language preservation, video processing, automated annotation

## 1. Introduction

Languages and cultures have always been evolving due to many well-understood historical factors. However, in recent decades the dynamics of those changes got an enormous speedup due to globalization. As a consequence UNESCO has reported that currently one language becomes extinct every two weeks and even major languages are changing. Similarly to biology, we can see a huge decrease in linguistic diversity (Crystal, 2000). Also the cultures are changing rapidly, identity building for young people becomes very difficult and the stability of societies is affected. We are deemed to loosing part of our cultural heritage since every language can be seen as a unique result of evolution resulting in rather different language systems. We also risk losing much of our knowledge about environment, species etc. since this is to a large extent encoded in the semantics of a given language.

During the last decades we recognize an increasing awareness about these threats resulting in a number of world-wide initiatives to document, archive and revitalize languages (DOBES[1], HRELP[2], PARADISEC[3]). It is well understood now, that we have the obligation to preserve our material and knowledge about languages for future generations, since they may want to understand their roots. Also we may wonder future generations may want to return to proper linguistic constructions that are currently blurring or which we currently are losing.

During the last decade also the awareness has grown that making recordings alone is not sufficient to guarantee that future generations will indeed be able to access the data. Recordings without appropriate annotations and metadata can be completely useless for anybody that has no knowledge about their creation and purpose. Therefore, at the Max Planck Institute for Psycholinguistics (MPI) an extensive language archive has been created with the aim of assuring long term preservation of audio and video recordings related to world languages.

Manual annotation of all the recordings in the MPI archive would be an impossible task. Therefore a significant role in the archiving tools will be played by the automated annotation algorithms, which are developed as part of the AVATecH project (Wittenburg *et al.*, 2010). Their role is twofold: 1) they would allow a dramatic decrease of time necessary to perform this task, which is normally very laborious; 2) automation of some parts of the process can greatly increase the uniformity of the annotations created worldwide by different researchers, which would contribute to consistency of the available language data. In this paper we describe in detail the algorithms that operate on video recordings and present the initial results that we could obtain with them.

## 2. Video Analysis Algorithms

The main principle that led the development of video analysis algorithm was to reduce the time needed to perform the annotation process and, when possible, make it completely automatic. The creation of robust and efficient algorithms was mandatory, due to the huge size of the video database of the MPI and the great diversity of the content. These two constraints were the main guideline in the creation of new algorithms and in the adaptation of existing ones to this specific problem. All the algorithms are designed to work without user interaction (except for the initial setup of some parameters) to allow batch processing on multiple videos. The implementation is done using a highly modular structure, so that future automatic annotators can be easily integrated in the current framework, using as input the results provided by the previous detectors.

### 2.1. Shot/cut detector and keyframes extractor

A shot is defined as the set of video frames that were continuously recorded with a single camera operation and represent therefore the basic unit of a video. All further described algorithms provide result for given shot and therefore rely on the results of the shotcut detection. The

[1] http://www.mpi.nl/dobes
[2] http://www.hrelp.org/
[3] http://www.paradisec.org.au/

tool developed in (Petersohn, 2004) was used as shot and sub-shot boundary detector. Sub-shots are defined as a sequence of consecutive frames showing one event or part thereof taken by a single camera act in one setting with only a small change in visual content. The detection of sub-shots proved to be useful for the development of the other detectors. The program processes standard definition videos at about 130 frames per second, on a Pc with Intel Xeon, 2.53GHz.

The number of videos in the database is so big and increasing at such a fast pace that often the researchers don't even have the chance to watch the video to decide whether it is worth annotating it. That's why one of the first requests from the linguist was to realize a tool that, even if it doesn't help in the creation of new annotations, allows them to browse easily and quickly the content of a video. The key frames extraction tool takes as input the information provided by the shotcut detector and extracts an image each time a sub-shot is detected. Using a standard configuration the processing speed is 5 to 10 times faster than real-time.

## 2.3. Global motion detection

Another useful feature that can provide useful information to the researchers is the detection of motion in a video. Accurate motion analysis allows distinguishing between different types of video content and it can be used to segment a video in order to select only the parts which are relevant for the researchers. For example, the presence of zooms and motion inside of a scene are usually the most interesting, while shots containing just panning and a low amount of internal motion are usually of little interest and can be usually discarded without further analysis. The algorithm developed performs a frame-based analysis and detects when global motion (pan, tilt, zoom in or zoom out) occur inside a shot. For each frame in the video a motion vector map is computed using the Hybrid Recursive Matching (HRM) algorithm (Atzpadin *et al.* 2004). The motion map represents the motion of a grid of pixels inside the frame. For each vector both the absolute value (i.e. the speed, calculated as L2 norm) and the phase (i.e. the orientation) are then computed. To detect the direction of global motion an 8-bins histogram of the phase of the motion vectors is also computed. Frames are considered candidates for global motion analysis when there are more than $\mu 1$ of motion vectors with absolute value above threshold $\mu 2$. These thresholds can be decided by the researchers but the standard implementation uses $\mu 1 = 5$ and $\mu 2 = 40\%$ of the total number of motion vectors in a frame.

A pan or tilt motion is then detected if one of the bins of the phase histogram contains more than half of the motion vectors, while zoom is detected when the 2 biggest bins in the phase histogram contain less than half of the motion vector and the variance of the absolute value of the motion vectors in the biggest bin is above a specified threshold. The approach used for zoom detection is similar to (Dumitraş and Haskell, 2004) and is based on the idea that when a zoom happens the majority of motion vectors point to (or come from) the center of the frame, with phases that range evenly between $[0, 2\pi]$ and absolute values that decrease nearing the center of the image. If no global motion is detected for a particular frame but there is nonetheless a significant amount of motion in the image the frame is then marked as having motion inside the scene. The program runs at about 30 frames per second on standard definition video and computing motion vectors on grids of size 8x8 pixels.

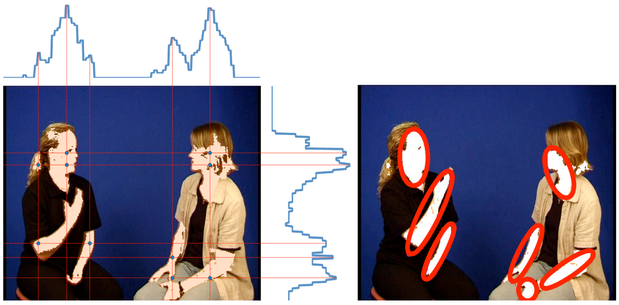

Fig. 1. Histogram of skin color points (left) and example of ellipses approximating the skin areas in given image (right).

## 2.4. Skin color estimation

Due to the peculiarities of the dataset in the underlying application scenario, there isn't a unique set of skin color parameters which can achieve good results in the entire dataset and therefore typical approaches that make use of a training set to collect the parameters for skin detection on the entire dataset cannot be applied (Terrillon *et al.*, 2000; Vezhnevets *et al., 2003*). The algorithm created uses both the temporal information provided by the change between one frame and the next and the spatial information provided by the fact that skin color pixels tend to cluster in well defined regions. This skin color estimator does not need a training dataset but rather estimates the YUV ranges identifying skin color for each frame in each video. The algorithm works in two steps: at first it uses a change detection tool to select the most suitable frames for skin color estimation, and then it applies an iterative clustering algorithm to select the range in the YUV domain that best represents skin color.

The idea of the change detection step is to apply a change detection algorithm to the luminance component of consecutive frames of the video and to obtain then a binary image (the change image) that is set to one for pixels where the difference in value between the frames is above a certain threshold. A 2D histogram of the change image is then computed and its bins are grouped into clusters. Ideally, each one of these clusters represents a body part moving in the current frame. Information regarding size, position, compactness is recorded for each cluster found in the histogram. After that all this information is passed to a cost function, which assigns a score to the current frame based on the properties of the clusters. The higher the score, the higher the probability that arms and heads are not overlaying, making the subsequent skin color estimation possible. The three frames obtaining the highest score are then selected to perform the second step of the algorithm, the iterative skin color estimation. To successfully estimate the skin color the algorithm retrieves six parameters, the mean value and the size of the range for the luminance component Y and the color components U and V. These parameters define a subset (a parallelepiped) in the YUV color space, and all the pixels in the image inside this subset are marked as skin.

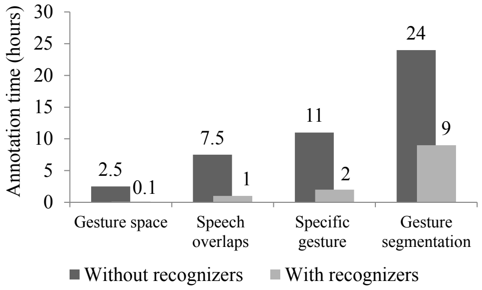

Fig. 2. Comparison of annotation time for different tasks.

As a first step the algorithm segments the selected frames, marking the pixels in the image if they are within a specific range in both the U and V components. In this step only the mean values change, while the ranges are fixed (20 for U and 30 for V). In this way for each UV interval under consideration a corresponding binary image is obtained and, a cluster analysis is performed to decide which range is the most likely ones to represent human skin color. The decision of the best color range is based on the number of clusters retrieved, their size, their compactness (defined as the ratio between the number of segmented pixels and the area of the ellipse that best approximates the shape of the segmented skin region) and their position with respect to the position of the clusters found analyzing the change image. The approach of segmenting the image many times, varying the color parameters is therefore repeated three more times, until all the six parameters are estimated. First, the algorithm segments the image varying the ranges of the U and V component but keeping fixed the values of their mean. After that the mean values for the color components are fine-tuned. Finally, the segmentation is performed one last time, varying the mean value of the luminance component while keeping its range fixed.

## 2.5. Hands and heads tracking

The algorithm works at first by segmenting the image in skin vs. non-skin pixels, using the information provided by the skin color estimator. The subsequent step in the detection process involves the search of seed points where the hands and heads regions most likely occur. Histograms along the horizontal and vertical directions compute the number of pixels with luminance and color values within the desired interval; the pixels where a maximum occur in both the directions are selected as seed points (Fig. 1 left). A region growing algorithm is then applied to the seed points in order to cluster together all the skin pixels in the neighborhood. Each region is approximated by an ellipse, characterized by the position of the center, its orientation and the length of its axes and for tracking purposes each of them is assigned a label (Fig. 1 right). The tracking is performed by analyzing the change in position and orientation of the ellipses along the timeline, assigning labels based on position of the regions in the current and previous frames. The tool is still in a development stage, current work focuses on improving the tracking when the hands and the head join or overlap, detecting the number of people in the video, distinguish left and right arm, separate the hand from the arm.

## 3. Results

We considered the measure of effectiveness of our solutions as the difference between the time necessary to create annotation to given media manually and with our algorithms. This value is not easily calculated, as the time necessary for annotating a time unit of media depends on factors like: the purpose of the recording and contents of the media; what exactly from the contents needs to be analyzed and annotated; the person performing the annotation process and their expertise. Also the level of applicability of our methods can be different for different scenarios, therefore resulting in different amount of help they can offer.

In order to estimate the usefulness of our methods we have created a scenario in which researchers had to perform a number of annotation tasks, to answer different linguistic research questions. The tasks have been chosen to represent a very common set of actions taken by a researcher that is annotating his recordings and included: 1) marking the size of the gesture space; 2) marking where speech overlaps with gesturing; 3) marking specific gesture or behavior through the entire recording, like nodding, raising your arms from rest to the level of the body, etc.; 4) marking when gesturing action happens and segmenting them into stroke, hold and retreat. These tasks have been first performed manually by several researchers and the time necessary to carry them out was measured and averaged. In order to measure the possible decrease of the time with the use of our video processing algorithms, the following steps have been taken:

Using the hands and head tracking method the positions of the hands have been marked in every frame of the recording and their x and y coordinates have been saved together with the time information. Then, for each of the tasks the appropriate simple algorithms have been written as extensions to the ELAN annotation tool. These extensions, called recognizers, are able to operate on the media from ELAN and information from all the previously described video processing algorithms. Using the information about the hands' position in time, it would be possible to extract the following information:

The size of the gesture space, using the maximum and minimum x and y coordinates of the hands. Furthermore, recognizers offer the possibility to get more complex values than by user observation, like for example: the size of the entire gesturing space is 300 by 400 pixels, but 90% of all gesturing happens in a 150 by 150 pixels subspace.

Exemplary action of interests has been described and detected. Namely, the action of raising one's hands from the resting position has been described as the situation when the hands significantly increase their y coordinate in short time and when they reach values that are similar to the position of the head. This action is relatively simple to describe and detect, but manual annotation for such is nevertheless very time consuming. We are working currently on a more compound scheme for pattern description, which would allow detecting much more complex actions in the recordings.

The gesturing activity has been described and detected in the recording. The description was based on the three phases that normally take place in a gesture: the stroke, which was characterized by a fast acceleration of the hand; the hold, which was described as a period of time following the stroke, with very little hand movement; and finally the retreat, which was defined as a steady movement of the hand, after the hold phase. This description naturally doesn't cover all the situations of gesturing in human behavior and therefore it will not successfully detect all the gestures and all the phases in the recordings (manual corrections of the resulting annotation would be necessary).

The above described recognizers have been executed with the input being the video recording and the results of previously described hands and head tracking algorithm. As the result, appropriate annotation has been created. Afterwards, the annotations have been subjected to evaluation and corrections by a researcher.

Fig. 2 presents the time necessary to carry out all the mentioned tasks. When the task has been performed with the help of recognizers, the necessary time stands for correcting the results obtained from them in order to make them useful for the researcher. It is possible to see that all the tasks took much shorter time to perform with the help of recognizers. Marking the size of the gesture space has been performed very effectively and almost no feedback from the user was necessary. Detecting specific action and detecting overlaps of speech and gesturing took more of user feedback, requiring respectively 1 and 2 hours of corrections. Detecting and segmenting the gestures, being the most complex tasks, required significant amount of corrections. However, all test cases have proven to save a lot of time of the researchers.

## 4. Conclusions and future work

The specification and implementation of the above described video processing recognizers has been performed in a very close contact with linguist researchers and according to the needs they have specified. After testing the relative effectiveness of our methods and witnessing the dramatic decrease of time necessary for annotations, we can say that our goals have been chosen correctly and our methods have proven very useful. As our next steps we are planning to fully develop the possibility of detecting and tracking the hands in the videos, differentiate left from right one and also work together with linguists to develop new recognizers that would create new types of annotations, for different research questions. We believe this work would contribute significantly to the quality of linguistic data stored at the Language Archive of MPI and possibly in other locations.

## 5. Acknowledgement

AVATecH is a joint project of Max Planck and Fraunhofer, started in 2009 and funded by MPG and FhG Some of the research leading to these results has received funding from the European Commission's 7th Framework Program under grant agreement n° 238405 (CLARA).

## References

Atzpadin, N., Kauff, P., Schreer, O., (2004). Stereo Analysis by Hybrid Recursive Matching for Real-Time Immersive Video Conferencing. In: Transactions on Circuits and Systems for Video Technology, Special Issue on Immersive Telecommunications, 14 (3), pp. 321-334.

Crystal, D. (2000). *Language Death.* Cambridge University Press.

Dumitraş, A., Haskell, B.G. (2004). *A look ahead method for pan and zoom detection in video sequences using block-based motion vectors in polar coordinates. Proceedings of International Symposium on Circuits and systems, ISCAS 2004, Vancouver, Canada, May 2004.*

Petersohn, C. (2004). *Fraunhofer HHI at TRECVID 2004: Shot Boundary Detection System. TREC Video Retrieval Evaluation Online Proceedings, 2004.*

Terrillon, J.C., Shirazi, M.N., Fukamachi, H. and Akamatsu, S. (2000). Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. *Proceedings of the International Conference on face and gesture recognition 2000, Grenoble , France, 28 March 2000, pp. 54-61.*

Vezhnevets, V., Sazonov, V. and Andreeva, A., (2003). *A Survey on Pixel-Based Skin Color Detection Techniques. Proceedings of the GraphiCon 2003, pp. 85-92.*

Wittenburg, P., Auer, E., Sloetjes, H., Schreer, O., Masneri, S., Schneider, D. and Tschoepel, S. (2010). *Automatic annotation of media field recordings. 4th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, LaTeCH 2010, Lisbon, Portugal.*