Towards Automatic Gesture Stroke Detection

Binyam Gebrekidan Gebre, Peter Wittenburg, Przemyslaw Lenkiewicz

Max Planck Institute for Psycholinguistics
Wundtlaan 1, 6525 XD Nijmegen, The Netherlands
{BinyamGebrekidan.Gebre, Peter.Wittenburg,Przemyslaw.Lenkiewicz}@mpi.nl

Abstract

Automatic annotation of gesture strokes is important for many gesture and sign language researchers. The unpredictable diversity of human gestures and video recording conditions require that we adopt a more adaptive case-by-case annotation model. In this paper, we present a work-in progress annotation model that allows a user to a) track hands/face b) extract features c) distinguish strokes from non-strokes. The hands/face tracking is done with color matching algorithms and is initialized by the user. The initialization process is supported with immediate visual feedback. Sliders are also provided to support a user-friendly adjustment of skin color ranges. After successful initialization, features related to positions, orientations and speeds of tracked hands/face are extracted using unique identifiable features (corners) from a window of frames and are used for training a learning algorithm. Our preliminary results for stroke detection under non-ideal video conditions are promising and show the potential applicability of our methodology.

Keywords: Gesture, stroke, annotation

1. Introduction

Many gesture and sign language researchers manually annotate recorded videos to systematically study patterns that are relevant to their hypotheses. The set of patterns and the set of annotations that go along with the patterns vary continuously and unpredictably in accordance with researchers needs. This fact makes any attempt at developing non-adaptive or non-learning general-purpose automatic annotation methods less effective. The trend in the literature has been to develop models that are designed to be trained once by developers and used thereafter by average users. This approach has three main disadvantages. First, it is impossible to know beforehand all the patterns that could be of interest to all researchers. Second, it is practically impossible to find enough training examples for all patterns. Third, it is currently impossible to learn a model that is robust across all video quality recording variations.

To overcome the three problems, this paper proposes a case-by-case user-controlled annotation model. The main philosophy for this kind of model is that a model designed to give the best average performance in a variety of scenarios is usually less accurate for a particular scene than a model tailored to the characteristics of that scene. This approach is also grounded in the *No Free Lunch* theorems, which establish that for any algorithm, any elevated performance over one class of problems is offset by performance over another class(Wolpert and Macready, 1997).

We apply this approach to the problem of gesture stroke detection. To be more precise, we develop a stroke detection model that takes intuitive input from the user for a given video and we apply standard algorithms optimized to the characteristics of the video.

2. Gesture stroke

Gesture stroke is the most important message-carrying phase of the series of body movements that constitute a gesture (or the phrases in a gesture). The body movements usually include hand and face movements. The relevant questions for automatic stroke recognition are *a*) what is a gesture? *b*) where does a gesture start and end? *c*) what are the phases in the gesture? *d*) which one is the stroke?

Figure 1 shows the different phases in a gesture unit. The figure shows that a gesture unit consists of one or more gesture phrases and each gesture phrase consists of phases that are called preparation, pre-stroke hold, stroke, post-stroke hold and retraction. Except for strokes, which are mostly obligatory, the rest of the phases in a gesture phrase are optional.

For the purpose of this paper, any hand/face movement is classified into two classes: strokes and non-strokes. The non-stroke gesture phases include the preparation, the hold, the retraction and any other body movements excluding the strokes. As strokes are mostly the only meaningful phases in a gesture, it is important to distinguish them from other types of movements (here called non-strokes). It is important to notice that we are not identifying the meanings of the strokes. We are only trying to locate their presence.

3. Methodology

Our approach to determining gesture strokes involves four steps: *a*) detect face and hands for every person *b*) track them *c*) extract features *d*) distinguish strokes from non-strokes. Each step is solved by different algorithms. Detection of faces and hands is carried out by two features: corners and colors. The two features have been selected because they are usually stable from frame to frame for a given video.

Corners are shown to be good features for tracking(Shi and Tomasi, 1993). A given point in a homogenous image cannot be identified whether or not it has moved in the subsequent frame. Similarly, a given point along an edge cannot be identified whether or not it has moved along that edge. However, the motion of a corner can conveniently be computed and identified. A corner has the property that it is different from its surrounding points. This makes it identifiable and a good feature for tracking.

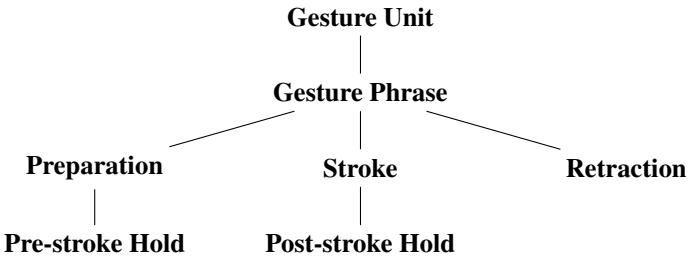

Figure 1: Gesture Phases
(Kendon, 1980; Kendon, 1972)

For a given application, not all corners in a video are equally important. For gesture analysis, the interesting corners are the ones resulting from the body movements, mainly from head and hand movements. In order to filter out the corners irrelevant to body movements, we mask out corners that do not have the skin color of human beings. The user selects a representative skin color region from the first frame of the video and can adjust the skin color ranges until the skin color regions are clearly separated from other regions.

The selected part of the video is converted from the RGB color space to HSV. A histogram of the H (Hue) channel with specified SV (Saturation and Value) ranges is used as a model of the skin color. The SV range values should not be the same for all videos. Changing their values for particular video recording conditions contributes to a more accurate model of the skin color. In our system, the SV adjustment is done using sliders and immediate visual feedback of the adjustment is shown, allowing the user to experiment before deciding on the most accurate adjustment.

The skin color model is used to calculate the probabilities of frame pixels of being skin color. A region of pixels with probabilities more than 0.3 is used for finding corners belonging to the hands and the head. Pixels with probabilities less than 0.3 are considered non-skin.

Given corner features from the regions of the skin in the video, the tracking is done with the pyramidal implementation of the Lucas Kanade algorithms(Bouguet, 1999; Bradski and Kaehler, 2008). This algorithm tries to find the displacement that minimizes the difference of the given interest point from two frames in a sequence. The Lucas Kanade algorithm works based on three assumptions: *1*) brightness constancy - a point in a given image does not change in appearance as it moves from frame to frame *2*) temporal persistence - the motion of a surface patch changes slowly in time *3*) spatial coherence - neighboring points in an image belong to the same surface, have similar motion, and project to nearby points on the image plane. The tracking of the selected features is done within a window of specified size. There is a trade-off with the choice of the window size. A small window size cannot capture large motions. A large window violates the spatial coherence assumption. The trade-off problem is solved by applying Lucas-Kanade algorithm over a pyramid of images(Bouguet, 1999). A pyramid of images is a collection of down-sampled images(Adelson et al., 1984) and in our case, it is used to detect larger motions.

With the corners, the skin color model and the Lukas-Kanade algorithm, we have a set of corners in every frame. At this stage, which corners belong to which body parts (i.e. left hand, right hand and head for every person) in the video is unknown. Assuming there is one person gesturing, the corners can have from one to three clusters corresponding to left hand, right hand, head and their different possible combinations (joining).

Values extracted from the number of corners, clusters and their dynamics across frames (context) are fed into a supervised learning algorithm with class labels 1 for frames inside a stroke and 0 for frames outside a stroke. The machine learning algorithm is designed to predict whether or not a given frame is inside a stroke.

4. Experiment data

To test the detection of strokes, we used a stroke and non-stroke annotated video data of about 3.6 minutes long. It has one young lady speaking and gesturing in a natural environment. The lady is facing the camera most of the time. Figure 3 shows a screen shot of the video and table 1 gives information of the resolution, frame rate and other elements of the the video. The video has 60 strokes each ranging from 6 to 32 frames in duration with mean 15 and standard deviation 6. This video has been taken from the MPI archive `http://corpus1.mpi.nl/ds/imdi_browser/`.

Feature	Value
Length in minutes	3.6
Video resolution	320x240
Frame rate(per second)	24
Total frames	5441
Stroke frames	988
Non-stroke frames	4453

Table 1: Video header information.

5. Feature extraction

The experiment data consists of X values of different features and y labels. For every frame of the video, there is a feature vector and a corresponding label. The label is binary. If the frame is part of a stroke phase, it is labeled as 1 and 0 otherwise. The feature vectors are extracted from a given frame and its neighbors(three preceding and three

following frames). Every frame has information related to 11 predefined positions, four orientations and velocity. The predefined positions are discretized regions in front of and to the sides of the gesturer. This gesture space classification is inspired by McNeill (McNeill, 1992) and is shown in figure 2. The four orientations represent the four quadrants of the Cartesian plane with origin represented by the initial positions of the corners being tracked.

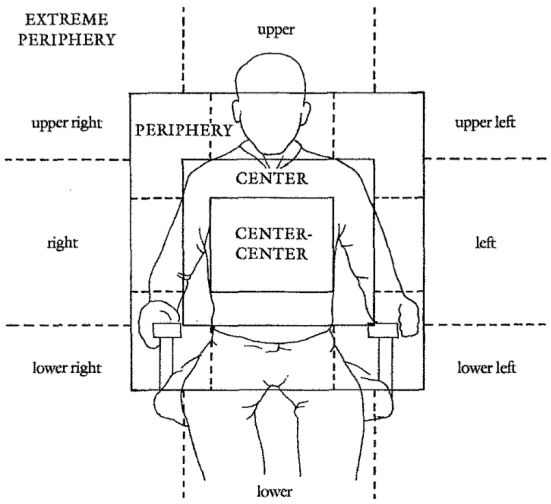

Figure 2: Typical gesture space of an adult speaker (McNeill, 1992)

6. Classification

To test the effect of the features in stroke detection, we implemented a regularized logistic regression classifier. Logistic regression is a discriminative linear classifier. It works by finding parameter values that minimize the cost function given in equation 1. In this equation, m is the number of examples in the training set, x_i is a feature vector and y_i is a $0-1$ label for example i. λ is a regularization parameter and n is the number of features. $h_\theta(x_i)$, shown in equation 2, is a sigmoid function that gives continuous values between 0 and 1 (i.e. probability). For classification, we have the freedom of choosing the threshold below/above which we decide to classify a given feature into one of the two classes. This freedom allows us to manage the trade-off between precision and recall as we wish. In our case, we set the threshold to 0.5. Using the gradient of the cost function and an optimization algorithm (fminunc), we find weights (parameter values) that minimize the cost function for the training data. We then use these weights for prediction.

$$J(\theta) = \left\{ \frac{-1}{m} \Big[\sum_{i=1}^{m} y_i \log h_\theta(x_i) + (1-y_i)\log(1-h_\theta(x_i)) \Big] + [\frac{\lambda}{2m} \sum_{j=1}^{n} {\theta_j}^2] \right\} \quad (1)$$

$$h_\theta(x_i) = \frac{1}{1+\exp^{-\theta^T x_i}} \quad (2)$$

7. Results

The accuracy of the system for face and hands detection based on initializations of regions of skin color on one frame in the video and applied to the subsequent frames in the same video depends on how distinct the skin color is as compared to the background and the clothes of the person. There is no ground truth data to give quantitative evaluation of the skin color detection algorithm. The qualitative evaluation is that the detection of skin color works good enough given that a) the user selects skin-region correctly and that b) they make correct adjustments using the sliders and the visual feedback.

The sliders on the system provide a mechanism to change the values of the ranges that specify the selected color region (in this case, skin color) and immediate feedback of the given action shows how good that action is. In our test video, this approach proved effective enough in identifying the skin color regions. However, if there are objects in the video where the color of the object is similar to the skin color, then the detection result includes also the object. This is not desirable and affects the detection process in a negative way. For example, in our test video, the color of the chair where the gesturer is sitting on is very similar to the skin color of the gesturer and is identified as skin color. Figure 4 shows a screen shot of the detected skin-region. The figure clearly shows that the system identified the chair as part of the skin color region. In the case where the camera is almost fixed, which is the case in our test video, the effect of the chair on the detection of corners belonging to the skin region(face and hands) can be ignored as the chair has few moving corners of its own.

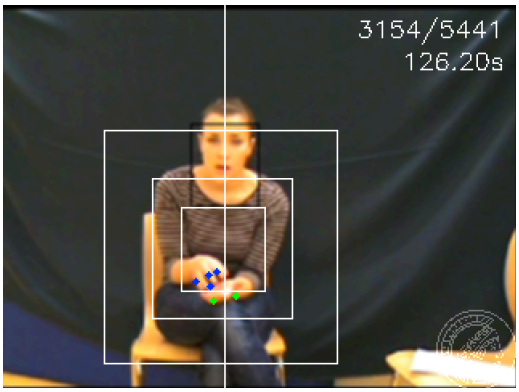

Figure 3: A screen shot of a video with corners. The corners are colored differently depending on their location in the gesture space. The rectangle in black is the region selected by the user as a model for the skin color.

The results associated with the classification of frames into part of a 'stroke' and part of 'non-stroke' are shown in figure 5. The horizontal axis of the graph represents the amount of data used for training and test. For example, 0.3 means 30% of the data is used for training and the remaining 70% for testing. The vertical axis represents performance levels ranging from 0 to 1 for accuracy, precision, recall and F1 measures. The average accuracy obtained is 88.06% and is measured as percentage of correct pre-

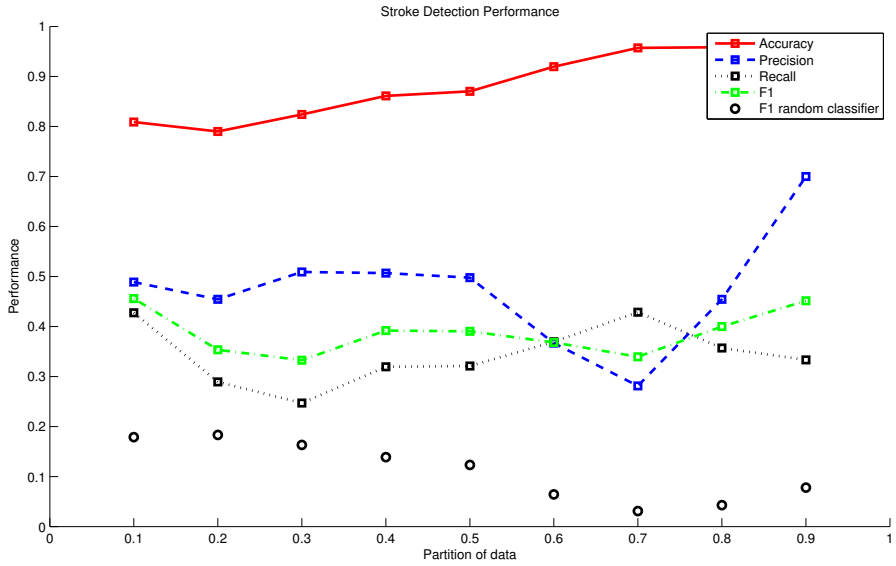

Figure 5: The graph shows accuracy, precision, recall and F1 results as the data for training and testing is partitioned. The bigger the partition number, the more data for training(the less data for testing) and vice versa.

Figure 4: A screen shot of a video showing detected skin regions. Very white regions correspond to high probabilities for skin regions. The chair is also part of the skin-region as it has almost the same color as the gesturer. See figure 3

dictions. The accuracy measure does not give a complete picture of the results because of the imbalances between the number of 'part-of-a-stroke' frames and 'part-of-a-non-stroke' frames. Frames belonging to strokes are only 988 out of 5441 (18.16%), which guarantees a baseline accuracy of 81.84%. The useful measures are precision, recall and their combined measure (F1). The average precision, recall and F1 measures are 47.24% 34.41%, 38.71% respectively. These measures show that the stroke detection performance is poor for practical and general purposes. However, the measures are much better than random stroke detection. The performance for a random classifier is consistently below the performance of our classifier as shown in figure 5. The average F1 measure for a random classifier is about 11.15% whereas for our logistic classifier it is 38.71%.

Evaluation for accuracy of frame boundaries for strokes and non-strokes should and cannot be as clear-cut as we presented it above. One or two frame misses are not bad given that even humans do not accurately mark the correct boundary anyway. However, we did not consider that observation in our evaluation results.

8. Conclusion

In this paper, we have put more emphasis on a more adaptive case-by-case annotation model based on the idea that with a little more input from users and facilitated by more user-friendly interfaces, annotation models can be more adaptive, more accurate and more robust.

We have tested our approach for the problem of hands/face tracking and automatic stroke detection. We have noticed that building a skin color model offline for all human skin colors will not only make the model more complex but also less accurate when applied on any particular video. However, models built online for a given video initialized by input from the user achieve higher performance at no more cost than the initialization.

We have also shown that unique features (i.e. corners) and their dynamics across frames can be indicative of the presence of strokes. In our future research, we will continue to to improve the stroke detection performance using more features and learning algorithms. If the problem of stroke detection is solved, the next stage would be to classify them according to their meanings.

9. Acknowledgements

The research leading to these results has received funding from the European Commissions 7th Framework Program under grant agreement n^o 238405 (CLARA). Special thanks go to Emanuela Campisi for providing the annotated data used in the experiments described in this paper.

10. References

E.H. Adelson, C.H. Anderson, J.R. Bergen, P.J. Burt, and J.M. Ogden. 1984. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41.

J.Y. Bouguet. 1999. Pyramidal implementation of the lucas-kanade feature tracker: Description of the algorithm, opencv documentation. *Santa Clara, CA: Intel Corp., Microprocessor Research Labs*.

G. Bradski and A. Kaehler. 2008. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media.

A. Kendon. 1972. Some relationships between body motion and speech. *Studies in dyadic communication*, 7:177.

A. Kendon. 1980. Gesticulation and speech: Two aspects of the process of utterance. *The relationship of verbal and nonverbal communication*, 25:207–227.

D. McNeill. 1992. Hand and mind: What gestures reveal about thought. *University Of Chicago Press, IL*.

J. Shi and C. Tomasi. 1993. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE.

D.H. Wolpert and W.G. Macready. 1997. No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1):67–82.