MVA2007 IAPR Conference on Machine Vision Applications, May 16-18, 2007, Tokyo, JAPAN

Sign Language Recognition Using Boosted Volumetric Features

Helen Cooper Richard Bowden
CVSSP, SEPS, University of Surrey
Guildford, UK
{H.M.Cooper, R.Bowden}@Surrey.ac.uk

Abstract

This paper proposes a method for sign language recognition that bypasses the need for tracking by classifying the motion directly. The method uses the natural extension of haar like features into the temporal domain, computed efficiently using an integral volume. These volumetric features are assembled into spatio-temporal classifiers using boosting. Results are presented for a fast feature extraction method and 2 different types of boosting. These configurations have been tested on a data set consisting of both seen and unseen signers performing 5 signs producing competitive results.

1 Introduction

The objective of this research is to produce a signer independent, environment invariant method of recognising motion. While the specific interests of this research lie in sign recognition the objectives are equally applicable to gesture. Sign Language (SL), being as complex as any spoken language, has many thousands of signs each differing from the next by minor changes in hand motion, shape or position. It's grammar includes the modification of signs to indicate an adverb modifying a verb and the concept of placement where objects or people are given a spatial position and then referred to later. The inter-signer differences are large, not just between different 'accents' and colloquialisms but also between new and old signers. The more fluent a signer, the faster and smaller they sign and the more co-articulation effects (where the end of one sign merges into the beginning of the next therefore blurring the boundary) become apparent. This, coupled with the complexities of sign grammar make true Sign Language Recognition (SLR) an intricate challenge.
This paper investigates a method of motion classification which identifies both the type of motion and its relative position to the signer in one step, eliminating the need for separate tracking and classification. The concepts of the integral volume, the features associated with it and boosting as a method for training the required classifiers are detailed. Then specifics of the implementation regarding numerical precision over long sequences and tractable memory requirements are discussed and a novel solution introduced. Finally the results of the experiments are presented.

2 Background

Many of the solutions to SLR that have been proposed use data gloves to acquire a definite position and trajectory of the hands [1] which are cumbersome to the user. The majority of current research focuses on tracking based solutions such as Staner and Pentland [2] who used colour to segment the hands for ease of tracking. More recently the focus has shifted to looking at sign language specifics as a way to aid classification. Vogler and Metaxas [3] used parallel HMMs on both hand shape and motion while Kadir et al [4] take this further by combining head, hand and torso position as well as hand shape to create a system that can be trained on five or fewer examples. Recently, non-tracking based research has begun to emerge, Zahedi et al [5] apply skin segmentation combined with 5 types of differencing to each frame in a sequence which are then down sampled to get features whereas Wong and Cippola [6] use PCA on motion gradient images of a sequence to obtain their features.

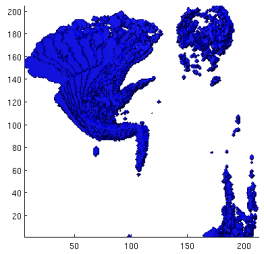

Figure 1: Example of signing motion over time.

3 Methodology

If a video stream of someone waving is examined and the footage processed using a frame differencing algorithm it can be seen how the hands move through time, this is shown in figure 1. As can be seen, there is a definite shape to the motion. By stacking each frame together as a volume, a sign or gesture can be thought of as a trajectory or subspace within this larger volume. By learning the shape and position of this subspace relative to the signer, a motion detector can be constructed. This approach approximates the subspace using a linear combination of simple block features learnt through boosting. In the remainder of this section we first discuss the boosting algorithms investigated before moving on to the concepts of the integral volume and how it can be used in SLR. Then some of the implementation issues that arise when considering the Integral Volume as a usable real time solution are discussed.

3.1 Boosting

Boosting provides a way of building a strong classifier that performs well through a simple selection process. An iterative algorithm, boosting first selects the best weak classifier from a set compiled of all available features (each with an optimum response threshold). It then applies a weighting to each training example. Reducing the weight-

ing of examples classified in the last pass and increasing the weighting of those not classified boosts the importance of examples which prove challenging to classify, encouraging the next iteration to concentrate more on the challenging examples with the heaviest weightings. Two different boosting algorithms are detailed, AdaBoost [7]and AdaPlusBoost which is a novel hybrid of Ada boost and RealBoost [8]. RealBoost differs from AdaBoost in two distinct ways, firstly the weak classifiers no longer return a binary classification but instead return a likelihood ratio based on the training examples and secondly, the selection of weak classifiers is not based on their performance in isolation but on the performance of the strong classifier that would be created were they to be added to the current solution. While RealBoost offers advantages in reduced classifier sizes with fewer redundant weak classifiers, it requires a more complete data set than AdaBoost in order to build up probability distributions represented as histograms. This can be partially overcome using Parzan windowing, however, this is only a solution for a near complete data set. AdaPlusBoost was devised to combine some of the advantages offered by RealBoost with AdaBoost's robust attitude to a sparse training set. It uses the same weak classifiers as AdaBoost but takes from RealBoost the idea that the best weak classifier to choose is the best in combination rather than the best in isolation. This results in a more optimum strong classifier which can be trained on relatively small data sets as will be seen.

3.2 Integral Volumes and Volumetric Features

The integral image was introduced to the image processing community by Viola and Jones [7]. Similar to summed area tables in texture mapping [9], it gives a fast way to calculate block features in an image. This is done by creating an intermediate image where any point (x,y) contains a value equal to the sum of all the pixels to the upper left of itself. Using this integral image one only needs to perform an addition and two subtractions to get the summation of any block area in the image.
To extend these block features into the temporal domain and allow for the efficient computation of volumetric features an integral volume was used [10] [11]. If the frames of a video are stacked one behind the other in temporal order, to create a volume, any point in the integral volume (IV) will contain the sum of all points to its upper left plus those before it. This is shown in figure 2 and by equation 1. Where V is the volume or video to be converted and (x, y, z) & (x', y', z') are points referenced to the top front left corner $(0, 0, 0)$.

$$IV(x', y', z') = \sum_{x=0}^{x'} \sum_{y=0}^{y'} \sum_{z=0}^{z'} V(x, y, z) \qquad (1)$$

To calculate a volumetric summation, four subtractions and three additions are required, again regardless of the block size. The features used are compiled of 2 volumetric blocks, whose parameters are selected relative to the signers face position and scale which allows a smaller feature set to be used. This was done by finding the signer's face in the image and then restricting the spatial search for motion

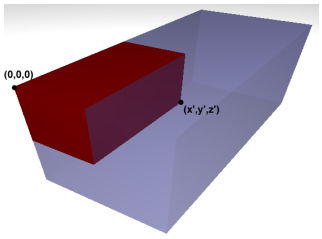

Figure 2: The value of the integral volume point (x,y,z) is the sum of all the points in the current and previous frames to the upper left of itself.

to an area around it. Furthermore if the size of the signer can be determined then a likely scale for the motion can be approximated. To this end the Viola Jones [7] face detector available in the OpenCV library [12] was used to find the face of the signer. From this, a position and relative size of the signer was determined.
The features themselves are based on the original block features used by Viola and Jones [7]. They have been extended into the temporal domain as shown in figure 3. Each feature returns a value which is the the difference between the values of the two volumes.

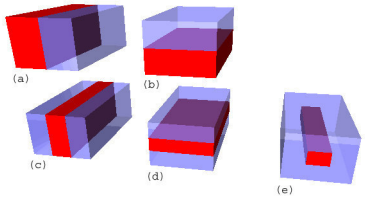

Figure 3: Some of the features used for the weak classifiers. A weak classifier gives a response (R_x) by subtracting the volumetric sum of the solid area$((x_s, y_s, z_s)(x'_s, y'_s, z'_s))$ from that of the translucent area$((x_t, y_t, z_t)(x'_t, y'_t, z'_t) - (x_s, y_s, z_s)(x'_s, y'_s, z'_s))$ and applying the optimally chosen threshold T_{wc}. Where (x, y, z) refers to the top left point of a sub-volume and (x', y', z') the bottom right point. See equation 2

$$V_s = \sum_{i=x_s}^{x'_s} \sum_{j=y_s}^{y'_s} \sum_{k=z_s}^{z'_s} V(i, j, k)$$

$$V_t = \sum_{i=x_t}^{x'_t} \sum_{j=y_t}^{y'_t} \sum_{k=z_t}^{z'_t} V(i, j, k)$$

$$R_x = \left\{ \begin{matrix} 1 \; if \; (V_t - 2V_s) \geq T_{wc} \\ 0 \; if \; (V_t - 2V_s) < T_{wc} \end{matrix} \right\} \qquad (2)$$

3.2.1 Image Buffer

One of the main implementation issues with the integral volume is maintaining numerical precision over long sequences, this is not an issue when working with short isolated sequences of signs but a serious consideration when considering a real-time detection application. This problem has been overcome using a double image buffer system that updates two image buffers simultaneously. While one buffer is being restarted the other is used to perform the calculations. A pictorial representation is shown in figure 4,

as can be seen the buffers are offset by half their size, this should be at least the same as the maximum size of the classifiers used. This can be done since the integral space time is relative and does not need to be continuous over all of the image stream but only over the length of the classifier.

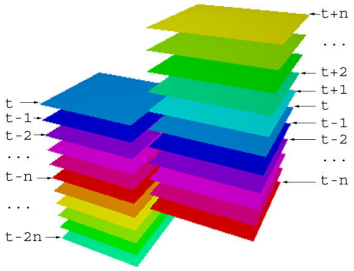

Figure 4: The dual image buffers work out of step with each other, when the first is half full the second starts filling from zero, when the first buffer is full the window switches into the second buffer and the first buffer is restarted. This way, at any frame there is always a full window of history yet the total value of any point is limited.

3.2.2 Pre-Computed Values

When training a system that requires many video clips it is inevitable that there will be computational considerations. Consider a sign video clip of 320 by 240 pixels and average length 50 frames, to store the full version would require a minimum space of 230.55Kb and recomputing the integral volume each time it was used. To store the integral video would require a minimum of 614.8Kb (assuming a 64bit integer is used per pixel) with a reduced complexity of only needing to compute the volumetric differences in each iteration. Alternatively, to store the response for each feature/example combination would only require around 400kb per example (assuming 50,000 features) and computation is reduced to one look-up each time a feature is tested. In the case of this research the dominating factor was speed, with a feature set of over a million candidate weak classifiers to choose from and using pre-computed values the system trains an average classifier in around 2 hours on a parallel implementation across 96 processors.

3.3 Feature Extraction

Frame differencing was used to extract features for the integral volume by subtracting pixel colour intensities between consecutive frames to show temporal change. Before being subjected to a morphological opening to remove noise the resultant differenced frame was converted to greyscale. Two options were then investigated, creating an integral volume from this greyscale difference image or applying a threshold to the frame to produce a binary image which could then be used in the integral volume. The processed volume (V_p) which comes from the original video volume (V_o) and is converted into the Integral Volume (IV) is given by the equation shown in 3. If a threshold (T) is applied, then the processed volume becomes as shown in equation 4

$$V_p(x,y,z) = |V_o(x,y,z) - V_o(x,y,z-1)| \quad (3)$$

$$V_p(x,y,z) = \left\{ \begin{array}{l} 1 \; if \; |V_o(x,y,z) - V_o(x,y,z-1)| \geq T \\ 0 \; if \; |V_o(x,y,z) - V_o(x,y,z-1)| < T \end{array} \right\} \quad (4)$$

Through experimental verification it can be shown that T should be kept low (around 10) so as to remove false motion (such as signal noise or compression artefacts) whilst still capturing all the actual motion.

4 Experimental results

4.1 Data Set

The training set consists of 5 repetitions of 5 different signs performed by 9 different people, it has a non-consistent cluttered background and was taken using a standard web cam. The test set was taken under similar conditions and consists of the same 5 signs repeated 5 times by 12 people, 3 of whom are not present in the training set (allowing testing on unseen data containing different subjects). The five signs in the set are the signs for 'sign', 'language', 'cat', 'hello' and 'friend'. The 12 people in the set were a mix of signers and non-signers and therefore contains considerable variation in signing across subjects.

4.2 Results

The system was trained and tested in the 4 different configurations, AdaBoost or AdaPlusBoost and $T = 10$ or no T. Fig 5 shows the first 20 weak classifiers chosen by the AdaBoost learning algorithm with T = 10 when classifying the sign for (a) 'hello' (a wave), and the sign for (b)'friend'(shaking hands). The face reference box is also shown, from this measurement all the classifier position and scales are calculated. The darker the classifier the earlier the weak classifier was chosen in the boosting algorithm. It can be seen that in (a) the initial classifiers are clustered around the moving arm and the side of the body where motion would be expected during a wave and for (b) the classifiers are concentrated around where the hands are shaking. ROCs for the tested configurations are shown in figure 6 and figure 7. The system was tested on two unseen test sets, the first containing just the known signers present in the training set and the second consisting of both the known and the unknown signers.

Figure 5: The first 20 classifiers chosen by the AdaBoost learning algorithm using $T = 10$ for the sign (a)'Hello' which is a wave and (b)'Friend' which is shaking hands with yourself.

5 Discussions

5.1 Boosting Algorithms

Comparing the two algorithms tested, AdaBoost and AdaPlusBoost, it can be seen that they both achieve sim-

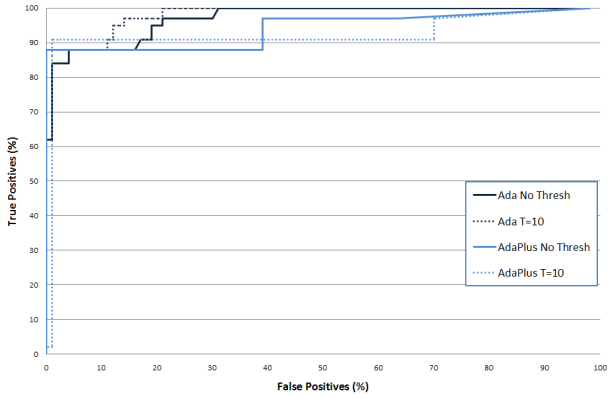

Figure 6: ROC Curves showing performance on known signers for the sign Hello

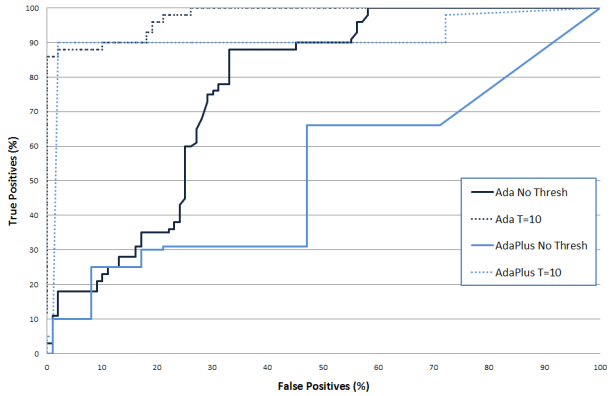

Figure 7: ROC Curves showing performance on a mixture of known and unknown signers for the sign Hello

ilar recognition rates. It should be noted, however, that AdaPlusBoost uses far fewer classifiers (often a factor of ten fewer than AdaBoost) to produce similar results. This means that the system would be much more suited to real-time applications if using the AdaPlusBoost classifiers, for while they take longer to train (due to re-calculating the strong classifier each iteration) they reduce the per frame time required for detection.

5.2 Feature Extraction Threshold

When working on purely known signers (those who appeared in the training data) little difference is made by using a threshold after the feature extraction stage. The advantages of the threshold are only fully noticeable when the system is tested on a mixture of both known and unknown signers. On this type of data the threshold reduces the disparity between different signers' skin colours and clothing colours making the system more robust to unseen signers.

6 Conclusion

In this paper a system has been discussed that can distinguish between various signs performed by any signer. Using the classifiers produced by AdaPlusBoost the system could be made to detect signs in a real-time situation. With the application of a threshold after the feature extraction phase there is no need for a new signer to re-train the classifiers. Together these points produce a signer-invariant and robust solution to recognise a selection of signs or gestures.

7 Future work

To extend this work a hand shape detector could be added to increase sign classification for signs which have similar motions. Also to reduce complexity, as the sign database increases it would be logical to work on the component parts of signs, the visemes, comparable to phonemes in speech.

References

[1] Gaolin Fang, Wen Gao, and Jiyong Ma. Signer-independent sign language recognition based on sofm/hmm. In *RATFG-RTS '01: Proceedings of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS'01)*, page 90, Washington, DC, USA, 2001. IEEE Computer Society.

[2] Thad Starner and Alex Pentland. Real-time american sign language recognition from video using hidden markov models. In *ISCV '95: Proceedings of the International Symposium on Computer Vision*, page 265, Washington, DC, USA, 1995.

[3] Christian Vogler and Dimitris N. Metaxas. Handshapes and movements: Multiple-channel american sign language recognition. In *Gesture-Based Communication in Human-Computer Interaction, 5th International Gesture Workshop*, pages 247–258, Genova, Italy, April 2003.

[4] Timor Kadir, Richard Bowden, Eng Jon Ong, and Andrew Zisserman. Minimal training, large lexicon, unconstrained sign language recognition. In *Proceedings of the British Machine Vision Conference*, volume 2, pages 939–948, Kingston UK, September 2004.

[5] Morteza Zahedi, Daniel Keysers, and Hermann Ney. Appearance-based recognition of words in american sign language. In *Second Iberian Conference in Pattern Recognition and Image Analysis*, volume 1, pages 511–519, June 2005.

[6] S.-F. Wong and R. Cipolla. Real-time interpretation of hand motions using a sparse bayesian classifier on motion gradient orientation images. In *Proceedings of the British Machine Vision Conference*, volume 1, pages 379–388, Oxford, UK, September 2005.

[7] Paul Viola and Michael Jones. *Robust Real-time Object Detection*. Second International Workshop on Statistical and Computational Theories Of Vision Modelling, Learning, Computing, and Sampling, 2001.

[8] Y. Freund and R. Schapire. *A decision-theoretic generalization of on-line learning and an application to boosting*. Journal of Computer and System Sciences, 1997.

[9] Frank Crow. *Summed-area Tables for Texture Mapping*. Association for Computing Machinery's Special Interest Group on Graphics and Interactive Techniques, 1984.

[10] Yan Ke, Rahul Sukthankar, and Martial Hebert. *Efficient Visual Event Detection Using Volumetric Features*. International Conference on Computer Vision, 2005.

[11] Takehito Ogata, William Christmas, Josef Kittler, and Seiji Ishikawa. Improving human activity detection by combining multi-dimensional motion descriptors with boosting. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, pages 295–298, Washington, DC, USA, 2006. IEEE Computer Society.

[12] OpenCV-User-Group. *OpenCV Library Wiki*. http://opencvlibrary.sourceforge.net, 2006.