Real-time Interpretation of Hand Motions using a Sparse Bayesian Classifier on Motion Gradient Orientation Images

Shu-Fai Wong and Roberto Cipolla
Department of Engineering
University of Cambridge
Cambridge, UK, CB2 1PZ
{sfw26,cipolla}@eng.cam.ac.uk

Abstract

An approach to recognise 10 elementary gestures is proposed and it can be applied to sign language recognition. In this work, a motion gradient orientation image is extracted directly from a raw video input and transformed to a motion feature vector. This feature vector is then classified into one of the 10 elementary gestures by a sparse Bayesian classifier. A training set of 628 samples and a testing set of over 1000 samples have been obtained to evaluate the proposed method. A real-time system was built and trained with the training set. From the experiment, the reported classification accuracy is 90% and the system can run in around 25 frames per second. Compared with other recently proposed methods that involve the use of hand tracking, the system can work reliably in real-time without relying on accurate tracking, and give a probabilistic output that is useful in complex motion analysis.

1 Introduction

Sign language systems are well known for their structured sets of gestures and they can provide a test bed for gesture recognition algorithms. Research in sign language recognition is therefore useful in building an interface between deaf and hearing people, and also in developing novel human-computer interfaces.

Sign language recognition is a complex problem, which requires a *divide-and-conquer* approach. Complex sign recognition can be considered as recognition of a sequence of primitive movements. It is, however, usually difficult to recognise primitive movements from raw images. This is mainly because getting motion information from raw images usually involves *target detection* and *visual tracking* that are also complex problems in computer vision. Recent works such as [4, 6] rely heavily on an accurate tracking result and thus they may fail if the tracking result is not reliable. To perform accurate tracking, *assumptions* that are usually *violated* in real world situation are often used, e.g., assuming non-skin colour background when a colour-based tracker is used, or assuming non-cluttered background if an edge-based tracker is used. Hence, the *applicability* of the system is often compromised.

BMVC 2005 doi:10.5244/C.19.41

In this paper, an approach to recognise 10 elementary gestures which *does not* rely on *tracking* is proposed. Gesture recognition is done by exploiting *motion gradient orientation (MGO) images* to form motion features and using a *sparse Bayesian classifier* to map the features into their corresponding classes. The present research has three main contributions. Firstly, by using the motion feature, which is derived from the MGO image, the key motion information can be encoded *without any accurate visual tracking*. This motion information can be used directly in motion classification and can yield a fairly high classification result. Secondly, due to the use of the Bayesian classifier, the final outcome is a *probabilistic value* instead of a simple true or false answer. The implication is that the final probabilistic results can be used in other *high-level inference processes* that need to maintain multiple hypotheses. Thirdly, the classifier maintains a *sparse model*, which facilitates an efficient use of computational resources and a *real-time performance*.

1.1 Previous work

Sign language recognition problem was first tackled by borrowing techniques from speech recognition. One of these techniques, Hidden Markov Models (HMMs), have been widely used in this problem in the past decade. In [7], HMMs were directly applied to solve the problem and their extensions such as parallel HMMs [10] and self-organizing HMMs [2] were also proposed to improve the performance. Recently, the use of HMMs has been criticised. One major criticism is that HMMs require *large training sets* (e.g. [4]) and this will inhibit the growth of the vocabulary. In addition, recent HMMs designs analyse *each sign as a whole* without breaking it down into corresponding components (e.g. [6]), making the model more complicated and reducing its extensibility.

A few suggestions have been proposed to address the mentioned weaknesses of HMMs. Bowden et al. [4] proposed the use of a "two stage classification" that does not involve the use of HMMs. In their work, motion data was firstly transformed into *a linguistic feature vector* which encodes a relative position, a relative motion and shape of hands (Stage I Classification), and the feature vector was then classified into its corresponding sign by *a bank of Markov chains* (Stage II Classification). As one may expect, the overall classification result is determined by the accuracy of the Stage I Classification. In their work, hard-coded prototypical motion patterns are used in the Stage I Classification to perform hand motion classification and this may reduce the reliability of the system. In [6], Derpanis et al. did not use HMMs in their work and they mainly focused on the classification of relative motion of hands from motion data (similar to Stage I Classification in [4]). They exploited manually defined functions to map *motion data in time series format* into the corresponding *motion type* robustly. Their work, however, relied heavily on the accuracy of the tracking result as shown in their experiments (97% overall accuracy with manually segmented output vs. 86% overall accuracy with automatic tracking output).

This paper extends their work to allow reliable hand motion classification without relying on tracking. In the following sections, the description of the proposed method and the implementation details will be presented.

2 Overview

In [6], Derpanis et al. introduced the idea of breaking down signs into *constituent primitive movements* with the aid of *linguistic information*. They generalised a set of 14 primitive movements and also a set of basic hand shapes using Stokoe's phonemic analysis of American Sign Language (ASL) [8]. Any sign in ASL can be represented by a combination of the primitive movements and the corresponding hand shape. On the other hand, sign language recognition can be done by recognising the hand shape, the primitive movements and the corresponding sequence. This recognition process can be considered as the Stage II Classification in [4]. Since the overall recognition accuracy is determined by the accuracy of recognising hand shape and hand motion, it is worth investigating how we could obtain an accurate result in the Stage I Classification. It is especially important to study hand motion recognition, which involves large inter- and intra-personal variation.

2.1 Problem Definition

In this paper, we will focus on how to obtain an accurate *hand motion classification* result *without* assuming *accurate tracking*. Instead of having 14 *primitive movements* as in Derpanis et al.'s work, we only have 10: (1) upward, (2) downward, (3) rightward, (4) leftward, (5) toward signer, (6) away signer, (7) nod, (8) supinate, (9) pronate, and (10) circular. In other words, the primitive movements that are a combination of two or more constituent primitives (e.g. 'up and down' equals to 'upward' plus 'downward') are removed. Figure 1 illustrates the 10 primitives, together with the hand shapes that are used in the experiment. The research problem in this paper is to classify a given video sequence of hand motion into one of the ten primitives.

2.2 Proposed Algorithm

The input of the system is a *video sequence* of a *single gesture* signed by a signer. The output of the system is a *classification result*. The tasks involved can be categorised into 3 main stages as shown below (the implementation details will be described in Section 3):

Feature Extraction Stage:

1. Obtain a *motion gradient orientation (MGO) image* (O) from the given *image sequence*, $V = \{I_0, I_1, ..., I_t\}$.

2. Obtain new basis functions by applying Principal Components Analysis (PCA) on the MGO images (O) obtained from all training examples (given N image sequences $\{V_1, V_2, ..., V_N\}$). Once the new basis functions are obtained, the MGO image (O) can be transformed into a *new motion feature vector* ($\mathbf{x}$) of a lower dimension.

Learning Stage:

1. Convert all examples (N image sequence $\{V_1, V_2, ..., V_N\}$) into their corresponding motion feature vectors ($\{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}\}$) using the above feature extraction scheme.

2. Train a *sparse Bayesian classifier* given the feature vectors ($\{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}\}$) and their corresponding class labels ($\{t_1, t_2, ..., t_N\}$).

Testing Stage:

1. Convert the given testing datum (V) into its corresponding motion feature vector ($\mathbf{x}$) using the above feature extraction scheme.

2. Obtain a classification result (t) from the Bayesian classifier given the motion feature vector ($\mathbf{x}$).

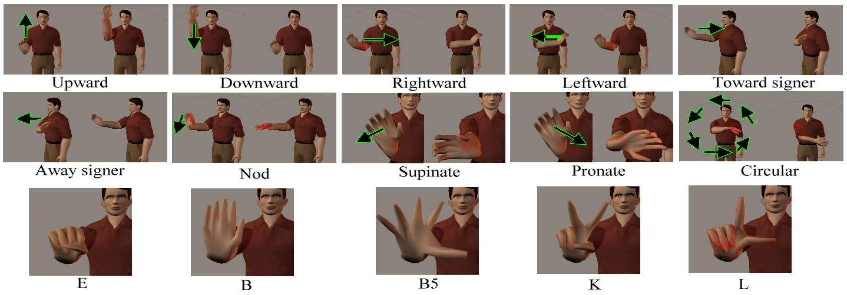

Figure 1: This figure illustrates the 10 primitive movements that are classified by the proposed system and the 5 basic hand shapes tested in the experiment.

3 Technical Details

3.1 Extraction of Motion Gradient Orientation Images

Motion gradient orientation (MGO) was proposed by Bradski and Davis [5] to explicitly encode changes in an image introduced by motion events. The MGO is computed from a motion history image (MHI) and a motion energy image (MEI). While a MHI encodes how the motion occurred, a MEI encodes where the motion occurred, the MGO therefore is a concatenation representation of motion (*where and how it occurred*).

Given a video sequence, $V = \{I_0, I_1, ..., I_t\}$ where I_t is an image captured at time t (and the precise form is $I(x,y,t)$), a binary mask $D(x,y,t)$ can be obtained from *image differencing*[1] and this mask can be used to indicate *regions of motion* at time t.

The motion history image (MHI) can be defined as [3]:

$$H_\tau(x,y,t) = \begin{cases} \tau & if \qquad D(x,y,t) = 1 \\ max(0, H_\tau(x,y,t-1) - 1) & otherwise. \end{cases} \tag{1}$$

In our system, τ is set to 255, which is the highest intensity of a greyscale image, and the most *recent motion* will generate the *brightest* MHI pixels. Thus, the intensity in a MHI encodes *how the motion has occurred*.

In addition to the MHI, the MEI is obtained by thresholding the MHI above zero [3] to represent where the motion has occurred. By performing morphological openings on the MEI to remove any noise introduced by irrelevant motion, the refined MEI encodes the *location of the motion region*. By finding the largest rectangle that covers all the motion pixels in the refined MEI, we can define a region of interest (ROI) that contains the relevant motion.

Within the ROI, motion gradient orientation (MGO) can be defined as [5]:

$$\phi(x,y) = arctan\frac{F_y(x,y)}{F_x(x,y)} \tag{2}$$

[1] The threshold is set to 30 in this system.

where $F_x(x,y)$ and $F_y(x,y)$ are the spatial derivatives along x and y direction of the MHI.

To obtain the final MGO image value ($O(x,y)$), $\phi(x,y)$ is rescaled[2] so that the range of intensity values is between 0 and 255. The intensity value indicates the *orientation of the motion changes*. Furthermore, since the size of the ROIs may vary from one video to the other, the final MGO images are resized to the corresponding refined images with standard size (which is 200×200 pixels in the proposed system). Typical MGO images corresponding to the 10 primitive movements used are illustrated in Figure 2.

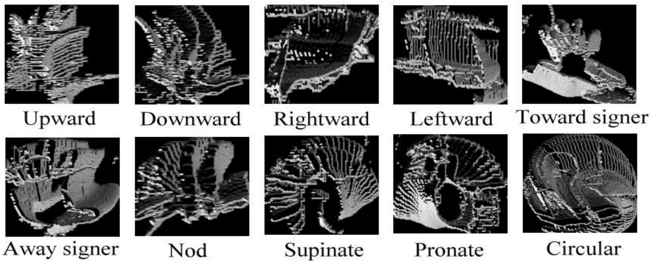

Figure 2: This figure illustrates the MGO images corresponding to the 10 primitive movements classified by the proposed system.

3.2 Dimension Reduction by PCA

In order to reduce the number of training samples, dimension reduction is done on the MGO images that potentially can be very large in size (200×200 in the proposed system). Principal Component Analysis (PCA) is used in the proposed system to reduce the dimension of the MGO images. By performing PCA on all the training MGO images, the eigenvalues indicate that the first 17 components provide an adequate summary of all the images, which account for 90% of the variation. Thus, the first 17 eigenvectors are chosen as the new basis functions for converting any new incoming MGO image (O) into a new feature vector. Finally, normalization is done to give a *final feature vector* ($\mathbf{x}$) with *zero mean* and *standard deviation of one* in all dimensions.

3.3 Classification by a Sparse Bayesian Classifier

Variation in MGO images due to *inter- and intra-personal variation* can be huge and thus a powerful classifier is needed. Firstly, different signers will sign the same gesture in different ways and thus generating different MGO images (see Figure 3). Furthermore, MGO images corresponding to the same gesture signed by the same person can have a significant variation due to difference in hand used (left vs. right) and hand shape (see Figure 4(a)). This variation may cause a difficulty in classification of a gesture, and thus a powerful classifier, a sparse Bayesian classifier (or a Relevance Vector Machine (RVM) [9] for classification), is used in this system.

Compared with other state-of-the-art classifiers such as Support Vector Machine (SVM), the RVM classifier performs equally well (see Section 4.3 and 4.4). In addition, the final

[2] Orientation value ‘0’ is adjusted to 255 in order to avoid the confusion between pixels indicating this orientation and pixels representing the static background.

outcome of the RVM classifier is a *probabilistic value* instead of a simple true or false answer. Furthermore, *sparsity of the model* stored by the RVM classifier ensures fast and efficient classification process, which also implies that it can be implemented in computing devices with limited memory storage such as Pocket PC or Smartphone. With these advantages, application of RVM techniques on *regression* has been exploited in various vision-based applications such as face tracking [11] and body tracking [1].

Although the RVM classifier is supposed to be a binary classifier, it can also be extended to a *multi-class classifier*. In the proposed system, the "one-versus-others" method is used. Since we have a total of 10 classes of motion, 10 independent RVM classifiers are constructed and each of them is trained to separate one class of data from all others. After all the classifiers are trained, the system can be tested by feeding a sample to all the classifiers. In practice, suppose this sample belongs to class *i*, the classifier which is trained to separate class *i* data from the others will give the largest output.

Given a training set $\{\mathbf{x_n}, t_n\}_{n=1}^{N}$, the problem of *learning a binary classifier* can be expressed as that of learning a function f so that the input feature $\mathbf{x_n}$ will map onto their correct classification label t_n and the probability of $\mathbf{x_n}$ is classified as the target class (where $t_n = 1$) equals to $\sigma(y_n) = 1/(1+e^{-y_n})$ where $y_n = f(\mathbf{x_n})$.

Typically, the function f can be written as a sparse model where ($M \ll N$) [9]:

$$f(\mathbf{x_n}) = \sum_{m=1}^{M} \omega_m \phi_m(\mathbf{x_n}) + \omega_0 \tag{3}$$

where $\omega = (\omega_0, ...\omega_M)^T$ are the weights and $\phi_m(\mathbf{x_n}) = K(\mathbf{x_n}, \mathbf{x_m})$ with $K(\cdot,\cdot)$ a positive definite kernel function (where Gaussian Kernel with width 1 is used in the proposed system) and $\mathbf{x_m}$ an example (or a relevance vector) from the training set. Under the RVM framework where hyperparameters $\alpha = \{\alpha_0, ..., \alpha_M\}$ are introduced, learning f from the training data means inferring ω from the data $\mathbf{t} = \{t_1, ..., t_N\}$ such that the posterior probability over the weights, $p(\omega \mid \mathbf{t}, \alpha)$, is maximised. Due to space limitation, the learning algorithm will not be addressed here. Readers may refer to [9] for more details.

4 Experimental Results and Discussion

4.1 Experimental Setup

The proposed method was *implemented* using unoptimised C++ code and the OpenCV library. All the experiments described were executed on a P4 2.4GHz computer with 1G memory. Both training and testing data are video *captured* under arbitrary room conditions (with various backgrounds and lighting). The video was captured by a webcam with a resolution of 320×240 pixels at 15 frames per second. In each video clip, the signer attempts to sign one of the ten primitive movements as described in Section 2. On average, each movement, which is manually segmented, lasts between 2 and 5 seconds. Three sets of video data together with the corresponding MGO images are illustrated in Figure 3.

In the learning stage, video data on a single signer was used to train the classifier. The goal of the training stage is to train the classifier so that it can correctly recognise the *primitive movements*, irrespective of the *hand (left or right) being used* and the *shape of the hand*. Therefore, the signer was asked to generate the primitives (or gestures) using different hand shapes with each hand. For hand shape, 5 hand shapes that correspond to

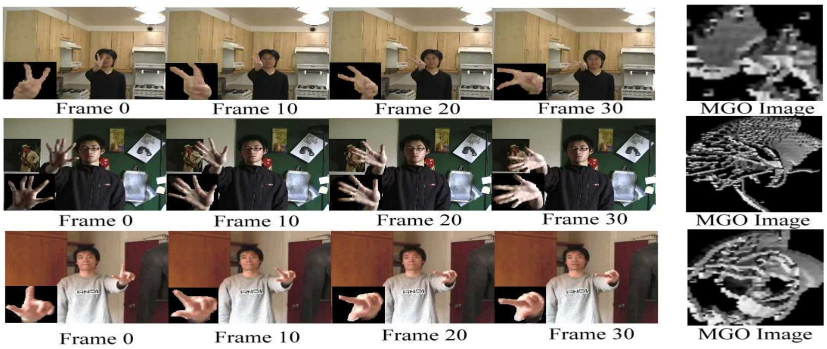

Figure 3: Each row shows 4 sample images from a typical set of testing video together with the corresponding MGO image. The enlarged hand shape is shown in the left-bottom corner of each image. In all 3 cases, the signers attempt to sign the ’Supinate’ gesture.

characters ‘E’, ‘B’, ‘B5’, ‘K’ and ‘L’ in ASL were selected to be signed (see Figure 1 for these hand shapes). These 5 hand shapes were selected because people can easily tell them apart through visual observations (for example, ‘E’ and ‘A’ differ slightly in the position of the thumb and thus only ‘E’ was chosen). Thus, the total possible number of training cases is 10 (number of primitive movements) $\times$ 2 (left or right hand) $\times$ 5 (number of hand shapes) $=$ 100. The signer was also asked to sign the same combination for at least 5 times. In total, the training set consisted of 628 gestures.

In the testing stage, video data from 5 different signers was used to test the classifier. Similar to the setup for capturing training data, the signers were asked to generate all classes of gestures using different hand shape and different hand. The hand shape used was the same as those used in capturing of training data. The signers were also asked to sign each combination at least twice. The testing set consisted of 1025 gestures.

4.2 Evaluation of the proposed method

Given the size of the training set is 628 and using the “one-versus-others” classification scheme, each RVM classifier maintains a *sparse model* with around 30 relevance vectors eventually (and at most 40). Assuming static background[3], the classifier can recognise primitive movements from the proposed motion features accurately under a *wide range of backgrounds* (e.g. cluttered background, and background with skin colour). The overall *accuracy* on 1025 test cases is 89.7%. Among these 1025 test cases, 73 of them cannot be mapped into any class (i.e. 7.1% of test cases). The *confusion matrix* is shown in Table 1.

In terms of *execution speed*, the extraction of the motion features (which includes the generation of the MGO image from the captured images and the generation of the final feature vector through PCA projection) takes 34.3 ms on average while the RVM classification step takes 7.2 ms on average. The average frame rate is 24.1 frames per second (fps)). That is to say, the system can run in *real-time*.

[3] This assumption is valid in most signing environments (e.g. those shown in television).

	up	down	right	left	toward	away	nod	supinate	pronate	circular
up	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
down	0.00	**0.96**	0.00	0.00	0.00	*0.01*	*0.03*	0.00	0.00	0.00
right	0.00	0.00	**0.99**	0.00	0.00	0.00	0.00	0.00	*0.01*	0.00
left	0.00	0.00	*0.01*	**0.99**	0.00	0.00	0.00	0.00	0.00	0.00
toward	*0.03*	*0.01*	0.00	0.00	**0.96**	0.00	0.00	0.00	0.00	0.00
away	*0.01*	0.00	0.00	0.00	*0.01*	**0.98**	0.00	0.00	0.00	0.00
nod	0.00	*0.01*	0.00	0.00	0.00	0.00	**0.99**	0.00	0.00	0.00
supinate	*0.04*	*0.01*	0.00	0.00	0.00	0.00	*0.04*	**0.89**	*0.01*	*0.01*
pronate	*0.01*	*0.01*	0.00	0.00	*0.01*	0.00	0.00	0.00	**0.97**	0.00
circular	0.00	*0.02*	0.00	*0.01*	0.00	0.00	0.00	*0.07*	0.00	**0.90**

Table 1: Confusion Matrix that shows the motion label (vertical) versus the classification result (horizontal). Each cell (i,j) in the table shows the percentage of class i motion being recognised as class j. Thus, diagonal cells (i, i) show the percentage of the correctly recognised gestures while the remaining cells show the percentage of misclassification.

4.3 Comparison with SVM Implementation

In order to compare the performance of the RVM classifier on learning the motion features with that of the other state-of-the-art classifiers, a similar gesture recognition system was implemented using a SVM classifier. To facilitate the comparison, the SVM classifier was implemented using the same Kernel (Gaussian Kernel with kernel width 1) as the RVM classifier. By training the SVM classifier with the same training set of 628 samples using “one-versus-others” classification scheme, each SVM classifier maintain a model with around 300 support vectors eventually (and at most 400 support vectors). By using the same testing set of 1025 test cases, an overall accuracy of 79.3% is achieved. The percentage of test cases that cannot be mapped into any class is 20.3%. In terms of execution speed, the SVM classification takes 18.9 ms on average.

4.4 Discussion

The experimental result shows that the *classification accuracy* is fairly high and it is comparable to some recent works on recognition of primitive movements for sign language such as [6], which achieves an accuracy of 86.00% (using colour-based tracking). In addition, as a result of the use of the proposed motion feature, the proposed method does *not rely on accurate tracking* as most of the other works do. Since most of the tracking algorithms may need to assume a background without skin colour and non-cluttered background, this means the proposed method can be applied to *a wider range of environment*. Furthermore, tracking may incur an extra computational cost to the sign recognition system and this may cause the final system not to run in real-time. Instead of using visual tracking, the proposed method needs only the motion feature extraction module that consumes relatively low computational resources (takes 34.3 ms on average to complete) and thus the unoptimised code implementing the proposed method can *run in real-time*.

Compared with SVM implementation, the proposed method, which uses RVM classifier, performs relatively better in terms of generalisability and sparsity. In the experiment, SVM classifier achieves a lower accuracy mainly due to the large number of primitive movements (20.3% of all test cases) that cannot be mapped into any class. By taking a closer look at these unrecognisable movements, it is not difficult to observe that these movements cannot be found directly in the training set and they are usually generated with some side-effects (such as motion of head or other background motion). Typical MGO images and images contain some commonly occurred artefacts are illustrated in Figure 4. Due to the problem of overfitting in SVM, the SVM classifier is relatively weak in classifying these movements. In contrast, the experiment shows that the RVM classifier can still recognise most of these movements (usually movements with minor artefacts) because of its *better generalisability*. The relatively poor classification result of SVM may also be due to the non-probabilistic nature of its output. Under "one-versus-others" scheme, if all SVM classifiers give '0' response, the system will conclude that the input is not recognisable. In contrast, RVM classifiers give *probabilistic values as an output*, and the final decision is made based on these values. Thus, RVM classifiers seldom give unrecognisable results with the exception that all probabilistic values are too low. Finally, the experiment shows SVM classifier maintains a complex model consisted of around 300 support vectors and thus gives a relatively slow performance. In contrast, RVM classifier maintains a *sparse model* consisted of around 30 relevance vectors and thus gives a faster performance (7.2 ms in RVM classifier vs. 18.9 ms in SVM classifier).

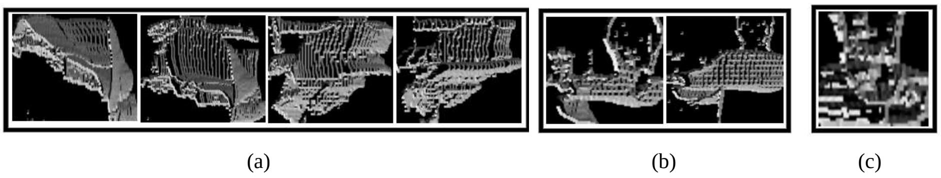

Figure 4: (a) shows typical MGO images of 'Leftward' gesture. The 2 Leftmost images show left hand motion while the rightmost images show right hand motion. (b) shows the MGO images of the same gesture with minor artefacts (mainly due to the head motion and body motion). (c) shows the MGO image of the same gesture with serious artefacts (mainly due to the wrong identification of the region of interest).

One major *limitation* of using RVM classifier is the *long training time*. In this experiment, it took around 2 hours to train the classifier using all the 628 training examples while SVM training took less than 10 seconds. Theoretically, the training time for RVM is proportional to the cube of the number of training samples and thus it may take extremely long time in training the classifier. Nonetheless, the number of training examples can be limited to achieve shorter training time while compromising the classification accuracy. A study was done to evaluate the drop in accuracy due to a reduction in the size of the training samples. The summary is illustrated in Table 2. It shows that as long as the number of training samples is larger than 400 which may take 40 minutes in training, the accuracy will be higher than 80%.

Number of training examples	100	200	300	400	500	600
Classification accuracy	64%	72%	79%	84%	88%	90%

Table 2: This table shows the relationship between the number of training samples and the classification accuracy

5 Conclusion

A new method is proposed to *recognise primitive movements* in this paper and it can be applied to *sign language recognition*. The proposed method performs better than recently used methods in three ways. Firstly, by extracting a motion gradient orientation image directly from images, a descriptive motion feature is formed *without depending* on any *tracking* algorithms. This means *computational overheads* due to tracking can be *reduced* and the *assumptions* tracking algorithms usually make can be *relaxed*. Secondly, by using a sparse Bayesian classifier that has relatively *better generalisability* and *sparsity*, the final classification result is comparable to other motion recognition methods and the result can be obtained with a *minimum amount of online computational resources*. Finally, the *probabilistic nature* of the Bayesian classifier implies that the proposed method can be applied in *complex motion analysis* that must maintain multiple hypotheses.

Acknowledgements. SW is funded by the Croucher Foundation Scholarships (H.K.).

References

[1] A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. In *CVPR*, pages II 882–888, 2004.

[2] B. Bauer and K.-F. Kraiss. Video-based sign recognition using self-organizing subunits. In *Proc. ICPR*, pages 282–296, 2002.

[3] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *PAMI*, 23(3):257–267, 2001.

[4] R. Bowden, D. Windridge, T. Kadir, A. Zisserman, and M. Brady. A linguistic feature vector for the visual interpretation of sign language. In *Proc. ECCV*, pages Vol I: 390–401, 2004.

[5] G. R. Bradski and J. W. Davis. Motion segmentation and pose recognition with motion history gradients. *Machine Vision and Applications*, 13(3):174–184, 2002.

[6] K. G. Derpanis, R. P. Wildes, and J. K. Tsotsos. Hand gesture recognition within a linguistics-based framework. In *Proc. ECCV*, pages 282–296, 2004.

[7] T. Starner, J. Weaver, and A.P. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *PAMI*, 20(12):1371–1375, December 1998.

[8] W. C. Stokoe, D. Casterline, and C. Croneberg. *A Dictionary of American Sign Language*. Linstok Press, Washington, DC, 1965.

[9] M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1:211–244, 2001.

[10] C. Vogler and D. Metaxas. A framework for recognizing the simultaneous aspects of american sign language. *CVIU*, 81(3):358–384, March 2001.

[11] O. Williams, A. Blake, and R. Cipolla. A sparse probabilistic learning algorithm for real-time tracking. In *Proc. ICCV*, page 353, 2003.