Machine Vision and Applications (2005) 16(5): 312–320
DOI 10.1007/s00138-005-0003-1

ORIGINAL PAPER

Eun-Jung Holden · Gareth Lee · Robyn Owens

Australian sign language recognition

Received: 13 September 2004 / Accepted: 24 August 2005 / Published online: 25 November 2005

Abstract This paper presents an automatic Australian sign language (Auslan) recognition system, which tracks multiple target objects (the face and hands) throughout an image sequence and extracts features for the recognition of sign phrases. Tracking is performed using correspondences of simple geometrical features between the target objects within the current and the previous frames. In signing, the face and a hand of a signer often overlap, thus the system needs to segment these for the purpose of feature extraction. Our system deals with the occlusion of the face and a hand by detecting the contour of the foreground moving object using a combination of motion cues and the snake algorithm. To represent signs, features that are invariant to scaling, 2D rotations and signing speed are used for recognition. The features represent the relative geometrical positioning and shapes of the target objects, as well as their directions of motion. These are used to recognise Auslan phrases using Hidden Markov Models. Experiments were conducted using 163 test sign phrases with varying grammatical formations. Using a known grammar, the system achieved over 97% recognition rate on a sentence level and 99% success rate at a word level.

Keywords Visual tracking · Vision system · Target detection · Human recognition

1 Introduction

Deaf communities in Australia communicate with each other by using a sign language called Auslan. While Auslan is different from the American Sign Language (ASL) or any other, all sign languages share the use of a combination of hand shapes, locations and motion as well as facial expressions.

Automatic recognition of a sign language requires the tracking of three target objects, namely the face and the two hands, and the extraction of features which are then classified as signs. Tracking is a difficult task since the face and hands are of the same colour and often overlap from a viewing point or touch. For example, "thank you" is signed in Auslan by having a straightened right hand tapping the chin once, then moving the hand forward as partially shown in Fig. 4. Thus the identification and the segmentation of occluded objects are necessary for the purpose of feature extraction. Features specify signs using the global representation that deals with motion trajectories and coarse shapes of the hands, or the local representation that deals with the characteristics of the fine hand shapes. These features are then classified as signs in the recognition process. Usually the signs in the vocabulary are modeled through training within the selected feature space, and used for classification.

We have developed an automatic Auslan recognition system using the global sign representation. The system tracks unadorned hands and the face in image sequences captured from a single colour camera, and recognises Auslan phrases using Hidden Markov Models (HMMs). It deals with occlusions of the face and a hand by tracking the contour of the foreground moving object. We have devised a set of global features that are invariant to scaling, 2D rotations and signing speed to represent signs for recognition.

A real-time ASL recognition system developed by Starner and Pentland [1] used coloured gloves to track and identify left and right hands. They extracted global features that represent positions, angle of axis of least inertia, and eccentricity of the bounding ellipse of two hands. Using an HMM recogniser with a known grammar, they achieved a 99.2% accuracy at the word level for 99 test sequences. Their feature space is dependent on the user's physical characteristics and the viewing distance, since the absolute positions and shapes of the target objects are used.

E.-J. Holden (✉) · R. Owens
School of Computer Science & Software Engineering,
The University of Western Australia,
35 Stirling Highway, Crawley, WA 6009, Australia
E-mail: {eunjung, robyn}@csse.uwa.edu.au

G. Lee
School of Engineering Science, Murdoch University,
Rockingham, WA 6168, Australia
E-mail: gareth.lee@murdoch.edu.au

Some systems, in contrast, deal with occlusions of unadorned hands and the face using a combination of image cues such as colour and motion. Yang and Ahuja [2] utilised these multiple cues to detect motion trajectory of ASL signs. They firstly perform motion detection by analysing each pair of successive frames for multiscale segmentation, matching regions of all scales across frames, and computing affine transforms for each matched region pair. Secondly colour segmentation is used to detect skin regions. Then thirdly, the head and palm regions are identified using the shape and size of skin regions in motion. Finally, sign motion trajectory is generated by concatenating affine transformations of the detected skin regions' motion. They classify these motion trajectories using a time delayed neural network, and recognise 40 ASL gestures with a 96% success rate. Their technique potentially has a high computational cost when false skin regions are detected, because all pairs of skin objects in successive frames are considered for the calculation of affine transforms.

Imagawa et al. [3] also integrated colour segmentation and temporal motion to track hands overlapping the face for their Japanese sign recognition system. The colour segmentation process uses a combination of colour look-up table and histogram backprojection to extract skin-colour regions and to enhance the contrast of the extracted regions. The motion tracking process uses the temporal difference images to detect moving hands which are then tracked using a Kalman filter.

While the above-mentioned systems [2, 3] used the colour and motion cues sequentially, Akyol and Alvarado [4] combined them into a single probability map to detect the signer's hands. Their technique uses Bayes' classification technique to generate a colour probability map, and uses motion history images to generate a motion probability map. Then these two probability maps are combined to detect the signer's hands. Tracking of these objects and recognition of signs are yet to be implemented as their aim is to build a sign recognition system for a mobile communication device for deaf people.

Tanibata and Shimada [5] used a combination of the colour cue and template matching technique for their Japanese Sign Language (JSL) recognition system. Skin colour detection is used to locate the hands and face, and the elbow is tracked throughout the image sequence to locate the wrists. When an occlusion of hands and face occurs, template matching is applied to locate and to separate the occluded objects. The texture templates of the face and hands, prior to the occlusion are used for the template matching. By rotating and translating the template, templates are matched within an expected region. They then extract hand features such as hand direction and the number of fingers to recognise 65 JSL words using HMMs. Applying template matching in varying angles and positions has a high computational cost and lacks reliability when the appearances change because of occlusions.

Another local feature extraction technique is developed by Imagawa et al. [6] who used an appearance-based eigen method to recognise signs even in two-handed and hand-to-hand contact cases. Using a clustering technique, they generate clusters of hand shapes on an eigen space, which are then used for classification. Signs comprising one-handed, two-handed, and hand-to-hand contact cases are used for experiment and they achieved 93% recognition of 160 words. The problem of using such an appearance-based recogniser is that the hand shapes and orientations appearing in a sign may vary involuntarily amongst the signers and amongst the utterances of a single signer. Thus they require a large set of training data accommodating these variations.

More recently, Bowden et al. [7] developed a British sign language recognition system that extracts a feature set describing the location, motion and shape of the hands based on sign linguistics, Recognition is performed using Markov chains combined with Independent Component Analysis. The use of high level linguisitic features minimalised the training effort for the recogniser. They achieved a recognition rate of 97.67% for a lexicon of 43 words using single instance training.

1.1 The proposed technique

We have developed the Auslan recognition system that has three components. The first is the tracking module that identifies the face and the hands while dealing with partial occlusions [8]. The second is the feature extraction process that extracts features that are invariant to scale, 2D rotation and signing speed by using relative geometrical positioning and shapes of the target objects, as well as their moving directions. The last is the recognition module which uses HMMs combined with a grammar to recognise colloquial Auslan phrases. Experiments are conducted using 163 test sign phrases of varying grammatical formations. The system achieved over 97% recognition at the sentence level using a known grammar and a 99% success rate at the word level.

The target objects are tracked using their geometrical characteristics of position and shape. When the face and a hand are occluded, we use the snake algorithm combined with motion information to find the contour of the foreground moving object. The active contour model or snake [9] is a well-known contour detection technique using a parameterised energy minimising spline that converges to an object contour within an image. A problem with the snake is that a high-level process must place the initial snake points close to the feature of interest because the snake will converge to the closest contour. In signing sequences, contour neighbourhoods of the foreground moving object change as it moves across the background object that contains similar contour features. An added complexity is that the hand object has many edge features within the object itself such as the finger joints. This can cause the snake to gradually draw onto a false contour such as the palm. Tracking the object contour in a cluttered background often relies on knowledge-based techniques [10] where viewing hand shapes must be known. In signing, it is difficult to build such a knowledge-base of hand shapes due to individually varying shapes for a particular sign, and shape changes between signs.

Our segmentation algorithm finds the contour of the foreground moving object, then segments the merged object in order to extract the corresponding geometric features. To achieve this, two types of motion information are combined with the snake algorithm to track the moving contour. One is an optical flow algorithm to initialise the snake location, where the shape of the initial snake is the bounding ellipse of the object in the previous frame. The other is temporal variance to draw the snake onto the moving object contour.

As features for recognition, we have devised a set of scale and rotation invariant features consisting of the geometrical relationship between the two hands and their temporal moving directions. Use of global features such as the position and coarse shape changes of the hands over time are sensitive to the physical characteristics of a signer, such as the arm length, hand size, or the viewing distance, as well as the signing speed. Instead, we use the angle between two hands with respect to the head, their moving directions, roundedness, and size ratio, to form a set of global features that are invariant to scaling and 2D rotations.

We used HMMs for classification of the signs to recognise colloquial Auslan phrases. The framework we employ is analogous to that used to recognise discrete verbal utterances. Originally developed by the speech recognition community, HMM technology has been widely used in gesture recognition systems in recent years [1]. The HMM provides a statistical model of development of the features associated with each phrase over time. The model parameters are estimated from a corpus of training examples and can subsequently be used to recognise a previously unseen example. Each model tracks the features over time using a sequence of continuous density distributions and is therefore robust to variation in the formation of signs and rate of signing. Models are constructed which correspond to individual signs and these are chained together, satisfying the rules of a predetermined grammar, to allow recognition of entire phrases.

2 Tracking the face and hands

The tracking process consists of skin colour detection, which finds the locations of all skin blobs and the correspondence algorithm that identifies the skin colour blobs as the face and hands.

2.1 Skin colour detection

Skin coloured objects are detected from a colour image, using principle component analysis (PCA) of the RGB colour space [11], which was previously used for face detection [12]. This colour space can be derived from the RGB colour space by using the following:

$$(R, G, B) \rightarrow (a, b, c),$$

where $a = (R + G + B)/3$, $b = (R - B)$, and $c = (2G - R - B)/2$.

The skin model is represented by its average colour component, $m = (\bar{a}, \bar{b}, \bar{c})$, and a covariance matrix of the skin colour component,

$$C = \begin{bmatrix} C_{aa} & C_{ab} & C_{ac} \\ C_{ba} & C_{bb} & C_{bc} \\ C_{ca} & C_{cb} & C_{cc} \end{bmatrix},$$

where $C_{ij} = (1/n)\sum_1^n (i - \bar{i})(j - \bar{j})$, $i, j \in [a, b, c]$, and n is the number of the skin colour samples. The parameters m and C have been derived from a database of sample skin images.

The skin model forms a cluster of the sample population on the PCA colour space. Thus given a colour component, p, the Mahalanobis distance [13], D, measures the distance of p from the skin model, using

$$D = (p - m)C^{-1}(p - m)^{\mathrm{T}}.$$

The Mahalanobis distances are measured for the input image and thresholded. Simple morphological operations are applied to the output of this process to clean the contour of the skin area.

2.2 Identification

The correspondence algorithm identifies the detected skin objects using previous shapes and locations of the face and hands in the image sequence.

2.2.1 Object representation

We represent an object by $M = (m_1, m_2, m_3)$, where m_1 is position (x, y), m_2 is the object size (number of pixels), and m_3 is the eccentricity of the bounding ellipse. Given a binary image of an object, the eccentricity of the bounding ellipse is calculated as the ratio of the square roots of the eigenvalues that correspond to the matrix

$$\begin{pmatrix} a & b/2 \\ b/2 & c \end{pmatrix},$$

where a, b, and c are defined as

$$a = (\int\int (x')^2 dx'dy')/N,$$
$$b = (2\int\int x'y'dx'dy')/N,$$
$$c = (\int\int (y')^2 dx'dy')/N.$$

Note that (x', y') are the normalised coordinates of (x, y) relative to the object centroid, and N is the size of the object (that is, the number of pixels in the object).

The eigenvectors (v_1, v_2) correspond to the major and minor axes of the bounding ellipse, and the eigenvalues (e_1, e_2) represent the corresponding variances (σ_1^2, σ_2^2) of the shape distribution over the major and minor axes. Thus, the eccentricity of the bounding ellipse, or roundedness, is defined by the ratio of the standard deviations, that is σ_2/σ_1.

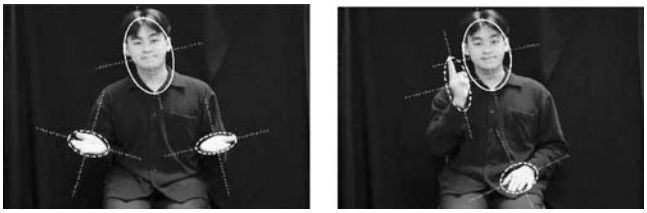

Fig. 1 The identification algorithm finds the corresponding target objects that are the face, right hand, and the left hand. Bounding ellipse of the face is shown in solid line, the right hand in dashdot line, and the left hand in dashed line. Two dotted lines on each ellipse illustrate the corresponding major and minor axes

(a) (b)

Fig. 2 **a** Occlusion of the face and the right hand is shown as one merged object as a result of skin colour detection. **b** Segmentation algorithm separates the hand and the face to extract their corresponding geometric features

2.2.2 The correspondence algorithm

Euclidean distances between the previous elliptical shapes and locations of the face and hands, and that of the detected skin coloured objects within the current frame, are used to determine their correspondence. For example, given a skin coloured object $M = (m_1, m_2, m_3)$ and the head object detected in the previous frame $H^{t-1} = (h_1^{t-1}, h_2^{t-1}, h_3^{t-1})$, where the object components represent the image location, size, and roundedness respectively, the likelihood of the object M being the head, $P(H^t = M)$ is defined as a function of the Euclidean distances between their object components. Thus their likelihood is defined as

$$P(H^t = M) = \alpha_1 p_1 + \alpha_2 p_2 + \alpha_3 p_3$$

where for $i = 1, 2, 3$, $p_i = e^{-\|m_i - h_i^{t-1}\|^2/(2\sigma_i^2)}$, α_i represents the significance of the corresponding component and $\sum_{i=1}^{3} \alpha_i = 1$. Given the image size 384×288, our implementation uses $[\alpha_1, \alpha_2, \alpha_3] = [0.5, 0.2, 0.3]$ and $[\sigma_1, \sigma_2, \sigma_3] = [100, 200, 0.1]$.

Amongst all skin coloured objects, the object with the highest likelihood is chosen as the target object. A result of the identification process is shown in Fig. 1.

3 Segmentation of occluded objects

Occlusion is detected by observing the number of skin coloured objects and their locations merging closer throughout the sequence. Once objects are merged, we track the contour of the foreground moving object and segment it from the background object using the snake algorithm.

A hand is a non-rigid object with many edge features within, such as the finger joints and finger nails. Also when two skin colour objects overlap, edge features on the contour of the foreground object are hard to detect because of their similar pixel intensities. Such problems make the conventional snake technique of drawing the snake onto the object edge difficult. We deal with these problems by combining temporal motion information through a two-step process. The first is to initialise the snake location for each frame using an optical flow algorithm. The shape of the initial snake is defined by the bounding ellipse of the moving object in the previous frame, in order to avoid the snake gradually moving towards a false inner contour within the object such as the palm of the hand. The second is to draw the snake onto the contour of the moving object by combining temporal variance information [14] with the object edge strength.

Figure 2 shows the skin detection process generating two merged objects as a single object in (a), and the result of the segmentation process viewing the bounding ellipses of two separate objects in (b). This two-step segmentation process is outlined next.

3.1 Optical flow snake initialisation

In snake tracking, it is important that the snake is initialised close to the surface of the object contour. We model the initial snake, for each frame, using the bounding ellipse of the object in the previous frame, where snake points are equally distributed along the boundary. The location of the initial snake is determined by a well-known, gradient-based optical flow method of Lukas and Kanade [15].

Within a small neighbourhood of a pixel, the Lucas and Kanade algorithm computes the velocity or displacement of the feature containing distinct horizontal and vertical gradients, using spatio-temporal derivatives of image intensity. In signing sequences, a contour neighbourhood of the foreground moving object may view the sudden appearance of similar gradient features from within the background object as it moves across. For example, when a hand moves across the face, the mouth and nose may appear in proximity to the moving hand contour. This may confuse the measurement of optical flow for the hand contour features. Thus we measure, for each snake point, the optical flow of the contour feature. Then the major flow vector that has the largest Euclidean norm is chosen for the translation of the elliptical snake that bounds the moving object in the previous frame.

Feature displacement within a small spatial neighbourhood, Ω, is characterised by a constant velocity $v = (v_x, v_y)^{\mathrm{T}}$. Given first-order derivative of $I(\mathbf{x}, t)$, $\nabla I(\mathbf{x}) = (I_x(\mathbf{x}, t), I_y(\mathbf{x}, t))^{\mathrm{T}}$, where $\mathbf{x} = (x, y) \in \Omega$ at time t, and the partial time derivatives of $I_{\mathbf{x}}$ is $I_t(\mathbf{x}, t)$, then the flow vector v is computed using

$$Av = B,$$

Fig. 3 A close-up view of the snake initialisation shows the previous snake in thin line, the optical flow calculated for each snake point, and the initialised snake for the current frame in thick line

where n is the number of pixels in Ω, and

$A = [\nabla I(\mathbf{x}_1), \ldots, \nabla I(\mathbf{x}_n)]^{\mathrm{T}}$,

$B = -(I_t(\mathbf{x_1}), \ldots, I_t(\mathbf{x_n}))^{\mathrm{T}}$.

In our implementation, equal importance is placed on all pixels within Ω, and a spatio-temporal Gaussian filter is applied to attenuate temporal aliasing and quantization effects in the input images. An example of optical flow calculation for the elliptical snake is shown in Fig. 3.

3.2 Motion snake

Once the initial location of the elliptical snake is determined for the current frame, the contour of the moving object is detected using the snake algorithm of Williams and Shah [16]. In signing sequences, edge features often appear close to the moving object contour such as the edge features in the background object or features within the moving object. To deal with this, we employ temporal variance information [14] combined with the object edge strength to draw the snake onto the contour of the moving object.

3.2.1 Temporal variance image

Intensity change from one image to the next is caused not only by object motion but also by camera or quantization noise. While both the background and the foreground of the image are affected by the noise, the object motion should generate greater intensity change, especially around the edges of the contour, than the noise. Motion between two sequential frames is represented, for each pixel within the skin detected regions, by the variance of temporal intensity change within a small neighbourhood of the pixel. Given greyscale images, $I(t-1)$ and $I(t)$ of the image sequence, the absolute pixel difference image, $R = |I(t) - I(t-1)|$ is generated. Then, the variance image V at time t is defined as

$$V(x, y) = \mathrm{var}(R(x - n : x + n, y - n : y + n)), \qquad (1)$$

where for each pixel location (x, y), a small pixel neighbourhood of $R(x, y)$ is used to calculate the variance. The variance image is then thresholded to separate the noise from the object motion.

3.2.2 Snake tracking

Given a thresholded variance image and an initial elliptical snake, the snake algorithm detects the moving object contour. The snake algorithm uses an energy minimisation technique and iteratively draws the initial spline to the closest object edges whilst maintaining its curvature (smoothness) and continuity (equidistance between neighbouring snake points). In an iteration, a scan line is generated for each snake point along the normal of the spline surface. Then the snake algorithm moves the snake point towards the object contour, by finding the location within the scan line that minimises the overall energy term which is defined as

$$E = \int (\alpha(s)E_{\mathrm{cont}} + \beta(s)E_{\mathrm{curve}} + \gamma(s)E_{\mathrm{image}})ds,$$

where the parameters α, β, and γ are used to control the relative importance of each term.

Given n snake points in a single frame, $s_1 \ldots s_n$, where $s_i = (x_i, y_i)$, the continuity term is defined as

$$E_{\mathrm{cont}} = \bar{d} - d_i,$$

where $d_i = |s_i - s_{i-1}|$ and $\bar{d}$ is the average of d_i. This term ensures that the snake points will not be drawn together along the snake contour but will remain approximately equidistant.

The curvature term is defined as

$$E_{\mathrm{curv}} = \left[\frac{\Delta x_i}{d_i} - \frac{\Delta x_{i+1}}{d_{i+1}}\right]^2 + \left[\frac{\Delta y_i}{d_i} - \frac{\Delta y_{i+1}}{d_{i+1}}\right]^2,$$

where Δx_i is $x_i - x_{i-1}$ and Δy_i is $y_i - y_{i-1}$. The curvature energy controls the smoothness of the spline curvature.

For image energy, we combine the temporal variance image V as previously defined in Eq. 1, with the edge detected image W. Gaussian smoothing is applied to both V and W and the image energy is defined as

$$E_{\mathrm{image}} = 0.6(1 - W(x_i, y_i)) + 0.4(1 - V(x_i, y_i)).$$

Figure 4 shows, in each column, the segmentation results of an example image frame. Therefore, by combining the temporal variance with the edge strength, our snake algorithm effectively finds the contour of the moving object.

4 Feature extraction

Once three target objects are identified, we extract features for recognition. Absolute positions, roundedness, and areas of the detected hand blobs have been used by other sign recognition systems [1], but these are sensitive to the physical characteristics of a signer such as the arm length and the hand size, as well as the viewing distance from a camera. The use of temporal changes of these features, in contrast, will result in sensitivity to the speed of signing. Thus we have devised a set of features that are invariant to scaling,

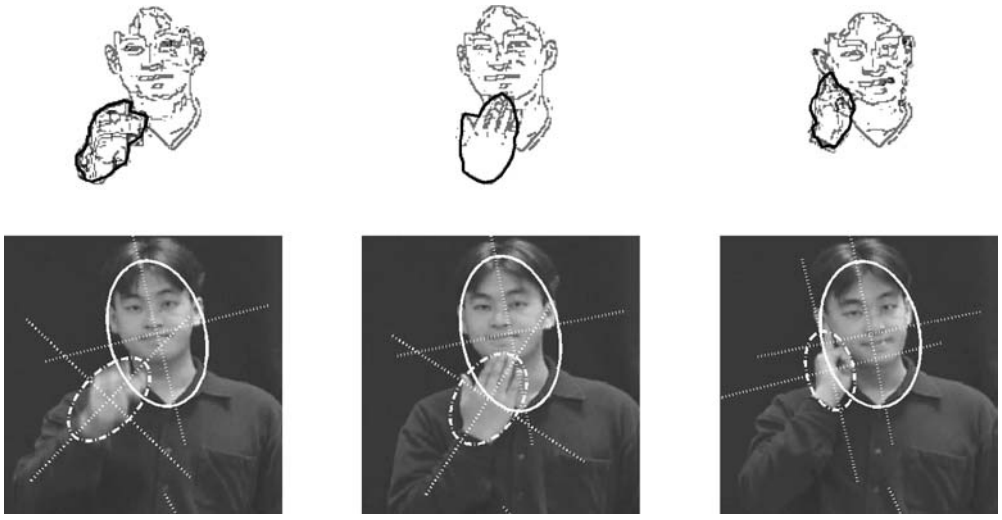

Fig. 4 Each column shows the snake tracking results of an image frame. The first row shows image energies as pixel intensities and the snake tracking result in thick line. The second row shows the segmentation results. Using snake tracking, two merged objects are identified and their elliptical features are extracted for recognition. For clarity, the face regions have been cropped

2D rotations, and signing speed. The features use relative geometric properties of the target objects, specifically positions, and shapes as well as the directions of the movement of the hands.

Figure 5 illustrates the positional relationship between the three target objects. The angle between the two arm vectors $\overrightarrow{F_tR_t}$ and $\overrightarrow{F_tL_t}$ is called θ_1, representing the degree of spreading of two hands regardless of the arm length of the signer. The moving directions of the right and left hands are determined by the angles θ_2 and θ_3 where each represents the angle between the hand velocity vector from the previous to the current frame ($\overrightarrow{R_{t-1}R_t}$ or $\overrightarrow{L_{t-1}L_t}$), and the corresponding arm vector. These angles define the velocity directions with respect to their arm vectors, thus are invariant to 2D rotations and the signing speed. They are within the range of [0°, 360°], and to avoid the discontinuity at 360°, we use *sine* and *cosine* values of these angles as features. The features also use coarse shape descriptions of the hands such as the roundedness of each hand, that is the eccentricity of the bounding ellipse previously specified in Sect. 2.2.1, D_{R_t} and D_{L_t}, and the ratio between their areas S_{R_t} and S_{L_t}. These shape features are relatively invariant to varying hand sizes and camera distances.

Thus the feature set comprises $\{\cos\theta_1, \sin\theta_1, \cos\theta_2, \sin\theta_2, \cos\theta_3, \sin\theta_3, D_{R_t}, D_{L_t}, S_{R_t}/S_{L_t}\}$.

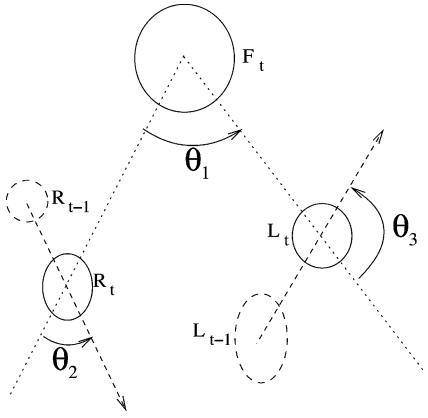

Fig. 5 The feature set uses geometric properties of the current positions of the target objects and their previous temporal changes. Centre positions of the face, right hand, and left hand at time t are labeled as F_t, R_t, and L_t respectively. Hand positions in the previous frame at time $t-1$ are labeled as R_{t-1} and L_{t-1}. Their positioning is defined by the angles between the vectors as shown. The variable θ_1 is the angle between $\overrightarrow{F_tR_t}$ and $\overrightarrow{F_tL_t}$, θ_2 is the anglebetween $\overrightarrow{F_tR_t}$ and $\overrightarrow{R_{t-1}R_t}$, and θ_3 is the angle between $\overrightarrow{F_tL_t}$ and $\overrightarrow{L_{t-1}L_t}$

5 Pattern recognition

The extracted features are recognised using a set of continuous density HMMs [17]. HMMs have been widely used to recognise sequences of feature vectors emanating from non-stationary stochastic processes, such as the neurological processes which generate verbal or signed utterances. The parameters for an HMM are estimated from a set of training utterances for each word in the vocabulary. When combined with a grammar, which describes all the allowed sequences of words in an utterance, the models can recognise any test utterance to find the most likely sequence of words.

Our HMM recogniser is constructed by using the Hidden Markov Toolkit (HTK), which has been widely used by the speech recognition community [18]. The HTK (Version 3.0) provides a number of default implementations of the algorithms needed to implement HMMs.

5.1 Hidden Markov models

The operation of HMMs is described in detail elsewhere [17], but a broad description of training and testing the pattern recogniser is as follows.

The algorithms process each training utterance U by decomposing it into a time-sequence of feature vectors (or observations). Each of the utterances is initially linearly segmented against the models and subsequently the Viterbi algorithm [17] is executed to provide initial estimates for the parameters of the HMMs. This algorithm is a form of dynamic programming which builds a trellis of possible alignments of the observations against the states of the HMM. It determines a map showing which state each observation vector most probably matches. Once this map is determined it can be inverted to determine the subset of observation vectors which are associated with a specific HMM state. The observation vectors within the subset can then be used to improve the estimate of the mean and covariance parameters associated with a state.

After this initialisation stage the Baum–Welch [17] algorithm is repeatedly executed; at each iteration it refines both the transition probabilities and also the mean and covariance matrices associated with each of the HMM state distributions. In our experiment each HMM state corresponded to a single 10-dimensional multivariate Gaussian distribution and all non-diagonal covariance elements were held at zero. Baum–Welch is a form of dynamic programming but, unlike the Viterbi algorithm, it does not make a categorical decision of which state a specific observation matches. However, similarly to the Viterbi algorithm, Baum–Welch has two phases of operation. In phase 1, a probabilistic mapping is established between observations and states, whereas in phase 2, the mapping is used to improve the parameter estimates of the states. Consequently, the Baum–Welch algorithm is repeatedly iterated between the two phases for each word model until the average probability of the training examples converges to a final value.

The HMM parameters derived from the training phase are then used to recognise test utterances. Each of the test utterances was evaluated against the possible word sequences resulting from the grammar to find the most likely sequence using the Viterbi algorithm. The algorithm can find the best possible alignment of feature vectors against HMM states so as to maximise the probability of the utterance given the model, $P(U|S_i)$, where U is the sequence of observation vectors constituting the utterance and S_i is the sequence of HMMs corresponding to the ith phrase. The HMMs corresponding to individual words in the phrase can be "chained together", in accordance with the grammar, to form a sentence level HMM. Bayes' rule can be used to reverse the conditional probabilities, resulting in the probability of each sentence model given the utterance:

$$P(S_i \mid U) = \frac{P(U \mid S_i)P(S_i)}{P(U)}.$$

If the models have some a priori bias (i.e. it is known that some sentences are more likely to be spoken than others) then this can be reflected in the choice of $P(S_i)$ otherwise uniform probabilities should be chosen. A value of 1 can always be used for $P(U)$. Consequently, the sentence for which $P(S_i|U)$ is maximal can be chosen—this is the classification decision.

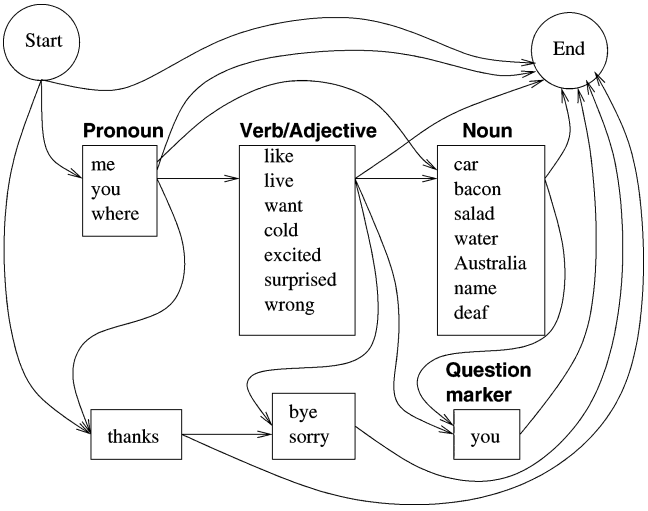

Fig. 6 The grammar structure used by the recogniser

5.2 Grammar

We created a grammar representing colloquial Auslan sentences as used by the deaf communities. The sentences consist of a sequence of pronouns, verbs, adjectives, or nouns formed in a legitimate order. The grammar of colloquial Auslan varies to some extent from the formal grammars of the language [19]. With the aim of building a practical application, we decided to accommodate colloquial grammar. We chose some useful sign phrases and requested our local deaf community to determine the grammatical formations of each phrase. On the basis of these colloquial grammars, a grammar graph was generated for the HMM recogniser. The graph for the recogniser is shown in Fig. 6.

5.3 Experimental results

Figure 6 shows the grammar structure used by the recogniser during an experiment. Note that some words are separated from the word groups to avoid feedback loops, thus allowing only forward paths in the network. The words "thanks", "bye", "sorry" are separated because these words are often used by themselves without forming connections to other word groups. Also a separate class of "you" is defined as it is often used at the end of phrases for questioning.

The grammar allows about 415 possible sentences to be constructed from 21 distinct words (but of these, many would be non-sensical). We were able to train and test the system using 379 utterances of 14 distinct sentences. Examples of these sentences include questions such as "where-live-you" (meaning "where do you live?") and "you-name-you" ("what is your name?"), simple pronoun-adjective phrases such as "me-cold"(I am cold), pronoun-verb-noun phrases such as "me-want-salad"("I want salad"),

and commonly used phrases such as "thanks-bye"("thank you, bye"). Each utterance resulted from processing a different video recording of a signer to produce a sequence of 10-dimensional feature vectors as described in previous sections. The 379 utterances were partitioned into two disjoint subsets: 216 training examples and 163 examples for testing. The training and test subsets each contained examples of all 14 sentences.

The recognition result shows that the system achieved 97% recognition at the sentence level, and 99% at the word level. Some of the failed cases were caused by coarticulation effects, where the hand motion that occurred from the end of a sign to the next is recognised as a sign.

6 Limitations and future developments

The proposed system has the following limitations that will be considered for future developments.

The tracking algorithm uses a combination of skin colour detection and a simple geometric correspondence algorithm. If skin coloured objects appear in complex backgrounds or the signer's clothing, with the similar shape and location of the target objects, the tracking may fail to determine the correspondence. The use of a prediction algorithm using spatio-temporal velocity may be difficult as a hand changes its direction suddenly causing the discontinued velocity. To deal with complex backgrounds, the system may require the tracking of elbows or other physiological landmarks.

The segmentation algorithm has been tested with moving foreground objects, but does not deal with the background object changing shape. This is important when dealing with occlusions of two hands where both of the foreground and background objects are changing shapes.

The current sign representation mainly deals with global motion in space with limited information on local motion of the hands. If two signs have the same trajectory, the system can only differentiate them if the hand shapes of both signs differ in their roundedness, or the signs differ in the size ratio of the right and left hands. For future developments, a better local shape representation technique needs to be incorporated into our sign representation to recognise fine motion of fingers and hands.

Another future research direction is to develop a two-way communication tool between English and Auslan in a practical application domain. We have also developed a real-time sign display system that generates Auslan signs using a 3D human model on computer graphics [20] which is shown in Fig. 7. We aim to combine the proposed sign recogniser and the sign display system to aid deaf people to communicate in places where sign language interpreters are not immediately available.

Fig. 7 A screetshot of our Auslan display system

Acknowledgements We thank Michael Arbib for his input to this project. This work is supported by the Australian Research Council.

References

1. Starner, T., Pentland, A.: Visual recognition of American sign language using hidden markov models. In: Proceedings of the International Workshop on Automatic Face- and Gesture-Recognition, pp. 189–194. Zurich, Switzerland (1995)
2. Yang, M.H., Ahuja, N.: Recognizing hand gestures using motion trajectories. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 466–472. Ft. Collins, CO, USA (1999)
3. Imagawa, K., Lu, S., Igi, S.: Color-based hand tracking system for sign language recognition. In: Proceedings of the IEEE 3rd International Conference on Automatic Face and Gesture Recognition, pp. 462–467. Nara, Japan (1998)
4. Akyol, S., Alvarado, P.: Finding relevant image content for mobile sign language recognition. In: Proceedings of the International Conference on Signal Processing, Pattern Recognition and Applications, pp. 48–52. Rhodes, Greece (2001)
5. Tanibata, N., Shimada, N.: Extraction of hand features for recognition of sign language words. In: Proceedings of the 15th International Conference on Vision Interface, pp. 391–398. Calgary, Canada (2002)
6. Imagawa, K., Matsuo, H., Taniguchi, R., Arita, D., Lu, S., Igi, S.: Recognition of local features for camera-based sign language recognition system. In: Proceedings of the International Conference on Pattern Recognition (ICPR), pp. 4849–4853. Barcelona, Spain (2000)
7. Bowden, R., Windridge, D., Kadir, T., Zinaerman, A., Brady, M.: A linguistic feature vector for the visual interpretation of sign language. In: Proceedings of the 8th European Conference on Computer Vision, vol. 2, pp. 391–401. Plague, Czech Republic (2004)
8. Holden, E., Owens, R.: Segmenting occluded objects using a motion snake. In: Proceedings of the 6th Asian Conference on Computer Vision, pp. 342–347. Jeju, Korea (2004)
9. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. In: Proceedings of the IEEE First International Conference on Computer Vision, pp. 259–269. London, UK (1987)
10. Blake, A., Isard, M.: Active Contours. Springer, Berlin Heidelberg New York (1998)

11. Ohta, Y., Kanade, T., Sakai, T.: Color information for region segmentation. Comput. Graph. Image Process. **13**, 222–241 (1980)
12. Hotta, K., Kurita, T., Mishima, T.: Scale invariant face detection method using higher-order local autocorrelation features extracted from log-polar image. In: Proceedings of the International Workshop on Automatic Face- and Gesture-Recognition, pp. 422–433. Nara, Japan (1998)
13. Manly, B.F.J.: Multivariate Statistical Methods: A Primer. Chapman and Hall, London (1986)
14. Habili, N., Lim, C.C., Moini, A.R.: Hand and face segmentation using motion and color cues in digital image sequences. In: Proceedings of the IEEE International Conference on Multimedia & Expo Lausanne, Switzerland (2002)
15. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the Image Understanding Workshop, pp. 121–130. Washington DC, USA (1981)
16. Williams, D.J., Shah, M.: A fast algorithm for active contours and curvature estimation. CVGIP: Image Understanding **55**(1), 14–26 (1991)
17. Rabiner, L.R., Juang, B.H.: An introduction to hidden Markov models. IEEE ASSP Mag. **3**(1), 4–16 (1986)
18. Woodland, P.C., Leggetter, C.J., Odell, J.J., Valtchev, V., Young, S.: The 1994 HTK large vocabulary speech recognition system. In: Proceedings of the ICASSP'95, Detroit, MI, pp. 73–76 (1995)
19. Johnston, T.A.: Auslan Dictionary: A Dictionary of the Sign Language of the Australian Deaf Community. Deafness Resources, Australia (1989)
20. Yeates, S., Holden, E., Owens, R.: An animated Auslan tuition system. Int. J. Mach. Graph. Vision **12**(2), 203–214 (2003)