

# Automatic Recognition of Colloquial Australian Sign Language

Eun-Jung Holden, Gareth Lee, and Robyn Owens

School of Computer Science & Software Engineering  
The University of Western Australia  
35 Stirling Highway, Crawley, W.A. 6009, Australia.  
{eunjung, gel, robyn}@csse.uwa.edu.au

## 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.*

## 1. Introduction

Auslan is the sign language used by the deaf communities in Australia. 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 touching. 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 Figure 2. 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 & Pentland [9] 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 a 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.

More recently, Bowden et al. [1] have developed a sign language recognition system that uses high level descriptors of the positions and movements of the hands, along with the classified hand shapes. Using a Markov chain classifier

combined with independent component analysis, the system recognises the temporal transition of individual signs. They achieved a 97% recognition rate for a lexicon of 43 words using only single instance training. The strength of this system is its accuracy using minimal training data.

While the above mentioned systems do not deal with occlusions of unadorned hands and the face, others have attempted to solve this problem using a combination of image cues such as colour and motion. Yang & Ahuja [13], for example, utilised skin colour detection and affine transforms of the skin regions in motion to detect the motion trajectory of ASL signs. Using a time delayed neural network, they recognised 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.

A local feature extraction technique is employed to detect hand shapes in sign language recognition. Imagawa et al. [3] used an appearance-based eigen method to detect hand shapes. Using a clustering technique, they generate clusters of hand shapes on an eigenspace, which are then used for classification. Signs comprising one-handed, two-handed, and hand-to-hand contact are used and they achieved a 93% recognition of 160 words. The difficulty of using an appearance-based recogniser is the collection of training data to accommodate the variations of hand shapes amongst the signers and amongst utterances of a single signer.

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 [2]. Occlusion is handled using colour and motion cues as well as a contour tracking technique. 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.

## 2. Tracking the face and hands

The tracking process consists of: skin colour detection which finds the locations of all skin blobs; the correspondence algorithm that identifies the skin colour blobs as the face and hands; and the segmentation technique that separates the occluded objects. This process is detailed in [2], thus only briefly explained in this paper.

### 2.1. Skin colour detection

Our skin colour detection uses principle component analysis (PCA) of the RGB colour space [7]. The skin model forms a cluster of the sample population in the PCA colour space. Thus given a colour component, the Mahalanobis distance measures the Euclidean distance from the population. These distances are thresholded to detect skin regions within an image and simple morphological operations are applied to the output of this process to clean the contour of the skin area.

### 2.2. The correspondence algorithm

The correspondence algorithm identifies the detected skin objects using their shapes and locations in the image sequence. Euclidean distances between the previous 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 of the bounding ellipse respectively, the likelihood of the object  $M$  being the head,  $P(H^t = M)$  is defined by applying Gaussian distributions over the Euclidean distances between their object components. Thus their likelihood is defined as

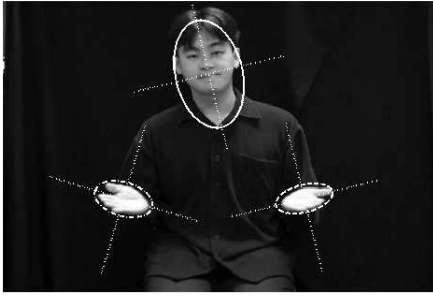
$$P(H^t = M) = \alpha_1 p_1 + \alpha_2 p_2 + \alpha_3 p_3$$

where for  $i = 1, 2, 3$ ,  $p_i = e^{-\|m_i - h_i^{t-1}\|^2 / (2\sigma_i^2)}$ ,  $\alpha_i$  represents the significance of the corresponding component and  $\sum_{i=1}^3 \alpha_i = 1$ . A result of the identification process is shown in Figure 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 separate it from the background object using a combination of motion cues and the snake algorithm.

The active contour model or snake [5] 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. 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



**Figure 1. The identification algorithm finds the corresponding target objects that are the face, right hand and left hand. The bounding ellipse of the face is shown in solid line, the right hand in dash-dot line, and the left hand in dashed line. Two dotted lines on each ellipse illustrate the corresponding major and minor axes.**

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 of Lukas & Kandade [6]. 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 with the object edge strength.

Temporal variance information specifies, for each pixel within the skin detected regions, the variance of temporal intensity change within a small neighbourhood of the pixel. Given two sequential 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:

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

where for each pixel location  $(x, y)$ , a small pixel neighbourhood of  $R(x, y)$  is used to calculate the variance. Thus temporal variances represent the motion between two sequential frames while eliminating camera or quantization noise.

Given a thresholded variance image and an initial elliptical snake, the snake implementation of Williams & Shah [11] is adapted for our contour tracking. The snake algorithm uses an energy minimisation technique, and itera-

tively draws the initial spline to the closest object edges whilst maintaining its curvature (smoothness) and continuity (equidistance between neighbouring snake points). During 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  $s$  toward the object contour by finding the location within the scan line that minimises the overall energy term which is defined as:

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

where the parameters  $\alpha$ ,  $\beta$  and  $\gamma$  are used to control the relative importance of each term.

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

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

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

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

where  $\Delta x_i$  is  $x_i - x_{i-1}$  and  $\Delta y_i$  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 Equation 1, with the edge detected image  $W$ . Gaussian smoothing is applied to both  $V$  and  $W$ , and the image energy is defined as

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

Therefore, by combining the temporal variance with the edge strength, our snake algorithm effectively finds the contour of the moving object. Figure 2 shows in each column, the segmentation results of an example image frame.

## 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 [9], 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, 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.

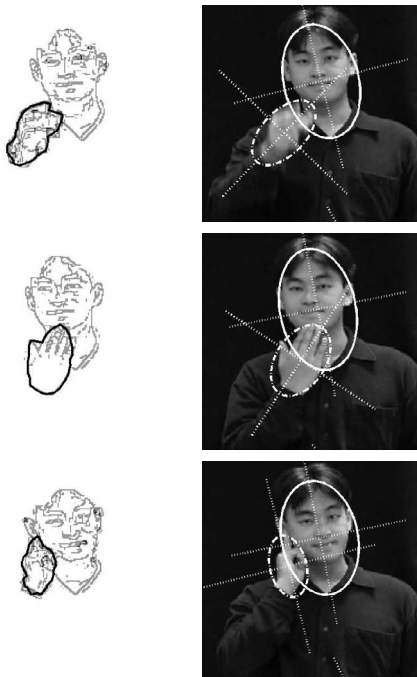


Figure 2. Each row shows the snake tracking results of an image frame. The first column shows image energies as pixel intensities and the snake tracking result in thick line. The second column shows the segmentation results. Using snake tracking, two merged objects are identified and their elliptical features are extracted for recognition. For clarity, the face regions have been cropped in the figure.

Figure 3 illustrates the positional relationship between the three target objects. The angle between the two arm vectors  $\vec{F}_t R_t$  and  $\vec{F}_t L_t$  is called  $\theta_1$ , 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  $\theta_2$  and  $\theta_3$  where each represents the angle between the hand velocity vector from the previous to the current frame ( $\vec{R}_{t-1} R_t$  or  $\vec{L}_{t-1} L_t$ ), 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. To avoid the discontinuity at  $360^\circ$ , 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,  $D_{R_t}$  and  $D_{L_t}$ , and the ratio between their areas  $S_{R_t}$  and  $S_{L_t}$ . These shape features are invariant to varying hand sizes and camera distances.

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

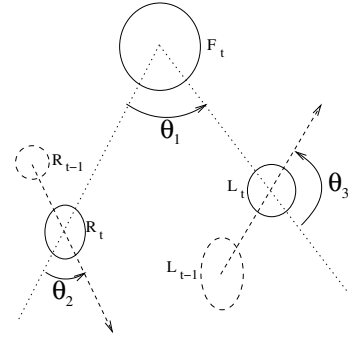


Figure 3. The feature set uses geometric properties of the current positions of the target objects and their previous temporal changes. Centre positions of the face, right hand, and left hand at time  $t$  are labeled as  $F_t$ ,  $R_t$  and  $L_t$  respectively. Hand positions in the previous frame 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 in the figure. The  $\theta_1$  is the angle between  $\vec{F}_t R_t$  and  $\vec{F}_t L_t$ ,  $\theta_2$  is between  $\vec{F}_t R_t$  and  $\vec{R}_{t-1} R_t$ , and  $\theta_3$  is between  $\vec{F}_t L_t$  and  $\vec{L}_{t-1} L_t$

## 5. Recognition

The extracted features are recognised using a set of continuous density HMMs [8]. 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 a 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 [12]. The HTK (Version 3.0) provides a number of default implementations of the algorithms needed to implement HMMs.

## 5.1. HMMs

A thorough treatment of the operation of HMMs can be found elsewhere [8], however a broad description of training and testing the pattern recogniser is as follows.

Each of the training examples was initially linearly segmented against the models and subsequently the Viterbi algorithm [8] was executed to provide initial estimates for the parameters of the HMMs. This also provides an initial segmentation of the utterances against the word models allowing the training algorithm to determine which vectors within the utterance would be used to train any particular HMM state.

After this initialisation stage the Baum-Welch [8] algorithm was 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. (Each state had a single 10 dimensional multivariate Gaussian distribution associated with diagonal covariance elements). The Baum-Welch algorithm was repeatedly iterated for each word model until the average probability of the training examples given the model ceased to improve.

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. Viterbi is a form of dynamic programming which finds 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 feature vectors corresponding to the utterance and  $S_i$  is the sequence of HMMs corresponding to the  $i^{th}$  sentence. (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 conditions, resulting in the probability of each sentence model given the utterance:

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

If the models have some a priori bias (ie. 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.

## 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 [4]. 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. Based on these colloquial grammars, a grammar graph was generated for the HMM recogniser. The graph for the recogniser as shown in Figure 4.

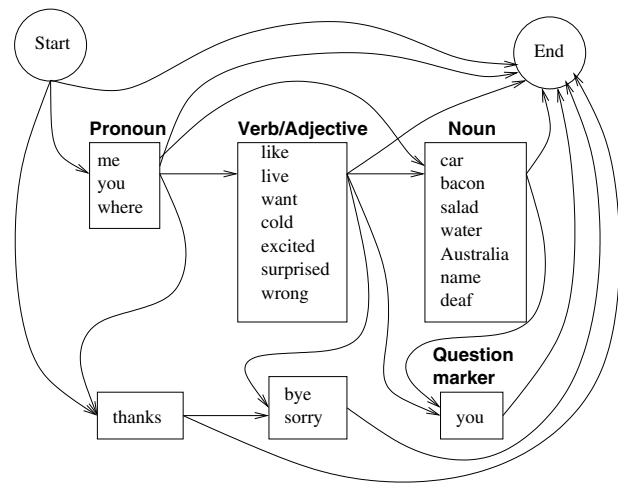


Figure 4. The grammar structure used by the recogniser.

## 5.3. Experimental results

Figure 4 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. Ongoing development

The proposed 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.

Our on-going project 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 [14] which is shown in Figure 5. 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.



**Figure 5. A screenshot of our Auslan display system.**

---

## Acknowledgments

We would like to thank Michael Arbib for his input to this project, and Jason Wong for his contribution. This work is supported by the Australian Research Council.

## References

- [1] R. Bowden, D. Windridge, T. Kadir, A. Zisserman, and M. Brady. A linguistic feature vector for the visual interpretation of sign language. In *The 8th European Conference on Computer Vision*, pages 391–401, 2004.
- [2] E. Holden and R. Owens. Segmenting occluded objects using a motion snake. In *The 6th Asian Conference on Computer Vision*, pages 342–347, 2004.
- [3] K. Imagawa, H. Matsuo, R. Taniguchi, D. Arita, S. Lu, and S. Igi. Recognition of local features for camera-based sign language recognition system. In *International Conference on Pattern Recognition (ICPR)*, pages 4849–4853, 2000.
- [4] T. A. Johnston. *Auslan Dictionary: A dictionary of the sign language of the Australian deaf community*. Deafness Resources, Australia, 1989.
- [5] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *IEEE First International Conference on Computer Vision*, pages 259–269, 1987.
- [6] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Image Understanding Workshop*, pages 121–130, 1981.
- [7] Y. Ohta, T. Kanade, and T. Sakai. Color information for region segmentation. *Computer Graphics and Image Processing*, 13:222–241, 1980.
- [8] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1):4–16, Jan. 1986.
- [9] T. Starner and A. Pentland. Visual recognition of american sign language using hidden markov models. In *International Workshop on Automatic Face- and Gesture-Recognition*, pages 189–194, 1995.
- [10] N. Tanibata and N. Shimada. Extraction of hand features for recognition of sign language words. In *The 15th International Conference on Vision Interface*, pages 391–398, 2002.
- [11] D. J. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *CVGIP: Image Understanding*, 55(1):14–26, 1991.
- [12] P. C. Woodland, C. J. Leggetter, J. J. Odell, V. Valtchev, and S. Young. The 1994 HTK large vocabulary speech recognition system. In *Proc. ICASSP '95*, pages 73–76, Detroit, MI, May 1995.
- [13] M. H. Yang and N. Ahuja. Recognizing hand gestures using motion trajectories. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 466–472, 1999.
- [14] S. Yeates, E. Holden, and R. Owens. An animated auslan tuition system. *International Journal of Machine Graphics and Vision*, 12(2):203–214, 2003.