

INTRODUCTION

We present a system that performs skin segmentation, hand and face tracking along with gesture recognition. The system consists of two main components namely segmentation and tracking followed by hand shape classification and gesture recognition as shown in Figure 1.

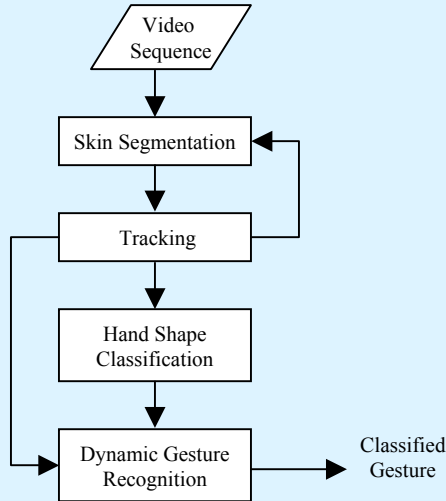


Figure 1 – System Architecture.

SEGMENTATION & TRACKING

Improving sign language SL recognition in natural conversation is our main motivation. This requires using skin detection techniques and handling occlusion between skin objects to keep track of the status of the occluded parts. We aim to present a unified system for segmentation and tracking of face and hands in a sign language recognition using a single camera. Unlike much related work that uses colour gloves, we detect skin by combining 3 useful features: colour, motion and position. These features together, represent the skin colour pixels that are more likely to be foreground pixels and are within a predicted position range. Also, unlike many other works that avoid occlusions, we handle occlusion between any of the skin objects using a Kalman filter based algorithm.

Skin Segmentation

Segmenting skin pixels is done inside small search windows which are defined using the predicted position of the skin objects. The next sections discuss the 3 features we used to segment skin pixels.

Colour information

In order to collect candidate skin pixels, we use two simple classifiers. First, a general skin model (colour range) P_g is applied on small search windows around the predicted positions of skin objects. As the fixed colour range can miss some skin pixels, we propose another colour distance metric $\text{dist}(C_{skin}, X_{ij})$ to take advantage of the prior knowledge of the last segmented object. This metric is the Euclidean distance between the average skin color C_{skin} in the previously segmented skin object and the current pixel X_{ij} in the search window. Finally, we normalize the values of the metric P_{col} .

Motion information

Movement information takes two steps. Firstly, we detect motion using frame differencing where the absolute difference image $D_f(x,y)$ is then normalized. The second step assigns a probability value $P_m(x,y)$ to each pixel in the search window to represent how likely it belongs to a skin object.

Position information

Assuming the the movement is sufficiently small between successive frames, we use a kalman filter model to predict the new position in the next frame. A binary mask of the object from the previous frame is translated to be centred on the new predicted centre and a search window is constructed around the bounding box of the mask. Then we compute a distance transform between all pixels in the search window and pixels of the mask. The values are then normalized to represent high probability P_{pos} values for near pixels and low probability for far pixels.

Information combination

After collecting the colour, motion and position features, we combine them logically using an abstract fusion formula to obtain a binary decision image.

Tracking and Occlusion Detection

Occlusion detection

We use the Kalman filter to predict the position of the bounding boxes around each of the current skin objects. If two boxes will overlap in the next frame, an alarm is raised. If in the next frame, the number of skin objects decreased and an alarm was raised, we conclude that occlusion has happened otherwise this means that one or more objects are hiding out of the camera view.

Tracking

After segmentation is applied inside each search window, connected regions are labelled after removing noisy small regions. Using the number of detected skin objects and the occlusion alarms as discussed above, we maintain a high-level understanding of the occlusion status. In our system, we deal with 7 occlusion cases that can occur between the face and the two hands. The final step is the blob matching where the previous frame blobs are matched against the new frame blobs using the knowledge of the high-level occlusions status. The matching is done using the distance between the previous objects centres and the new objects centres. Figure. 2 shows some examples of the segmentation and tracking of three frames.

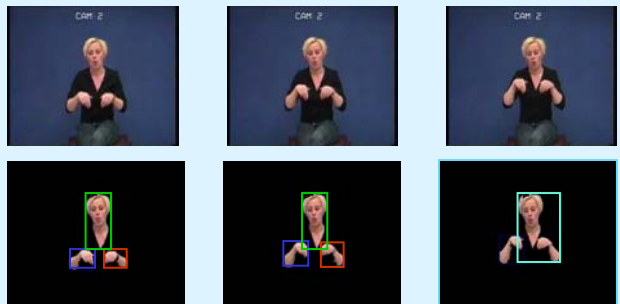


Figure 2 – Tracking samples including occlusion detection.

GESTURE RECOGNITION

Our current system can identify a vocabulary of 48 gestures. This vocabulary consists of 28 static gestures from the Irish Sign Language dictionary, 23 static finger spelling and 5 counting gestures. In addition it also comprises 20 dynamic gestures.

Static Hand Shape Classification

In hand shape recognition, transformation invariance is key for successful recognition. We have developed a system that is invariant to scale, skin colour and pixel translations along with small rotation and shape variations. Scale and Skin Colour invariance is dealt with using the pre-processing steps outlined in Figure 3.

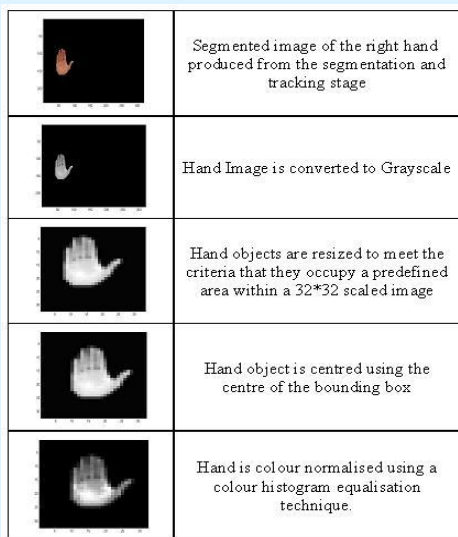


Figure 3 – Achieving Scale and Skin Colour invariance

Translation, rotation and small shape invariance is achieved by using a-priori knowledge to create a transformation subspace for each hand shape. Transformation subspaces are created by performing Principal Component Analysis (PCA) on images produced using computer animation.

These transformations relate to a non-linear transformation in high dimensional space. However if we approximate this hyper-plane of images using Principal Component Analysis (PCA), the dimensionality can be significantly reduced. PCA is a statistical tool that chooses a different set of axis to represent the data in fewer dimensions.

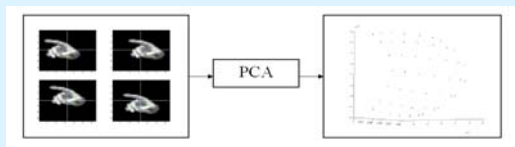


Figure 4 – Eigenspace Production

Figure 4 illustrates how an eigenspace is produced by performing PCA on a dataset that consists of translations of a particular hand shape. We have opted to use Poser Modelling software when training. Now all training images can be produced automatically. Some sample Poser images are shown in Figure 5.



Figure 5 – Sample Images Produced From Poser

Once the training images are acquired the system is constructed as displayed in Figure 6. A test image can be classified by finding the eigenspace with the minimum perpendicular distance to the image.

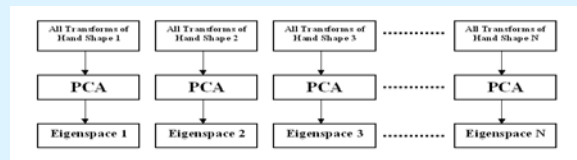


Figure 6 – Recognition Procedure

Instead of performing an exhaustive search on each subspace, we developed a hierarchical search tree that groups similar eigenspaces together. This is achieved by performing a fuzzy k-means algorithm on the origins of the eigenspaces. This allows us to reduce the search involved while retaining accurate search results.

Dynamic Gesture Recognition

Hidden Markov Models (HMMs) are used to recognise dynamic gestures. The feature vectors consists of two elements, both of which are positive integers. The first denotes the group to which the static hand shape has been classified. The second defines the position of the hand in the image and will be in the range 1-9. This range represents each of the 9 sections as shown in Figure 7. A dynamic gesture is then represented as a sequence of these feature vectors

A HMM is trained for each possible gesture using many different examples. A gesture is classified online, by manually identifying its start and stop point, then finding the HMM with the highest probability for the feature vector of the test sequence.

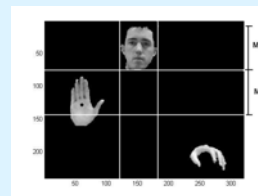


Figure 7 – Hand Position Classification

DISCUSSION

To date the system has only received a small number of informal user trials. However these trials have proved encouraging. The system operates in real time in an office environment. We only stipulate that users hands and face are the only skin coloured objects in the picture.

Some applications of our gesture recognition system include, Sign language recognition, robot manipulation, virtual reality and gaming. Almost all human computer interaction procedures could be improved by using gesture as a more natural way to interface with the computer.