ACCV2002: The 5th Asian Conference on Computer Vision, 23–25 January 2002, Melbourne, Australia.

Automatic Facial Point Detection

Eun-Jung Holden, Robyn Owens
Department of Computer Science & Software Engineering
The University of Western Australia
35 Stirling Highway, Crawley, W.A. 6009, Australia.
E-mail: {eunjung, robyn}@cs.uwa.edu.au

Abstract

A technique that automatically locates specific facial points on an unknown face image is presented. This technique uses log Gabor wavelets to represent each facial point by embedding local surrounding features. Using a sample log Gabor response of a facial point, locations on the unknown face image that appear similar to the sample facial point are detected. We identify facial points by employing geometric relationships between the facial points that are invariant to 2D rotations and scaling. An experiment is reported that uses three sample face images to determine the log Gabor representation of each facial point, and the technique is tested on an unknown face image. It demonstrates that the log Gabor representation is effective in finding similar features by successfully detecting and identifying all of the seven specified facial points.

1 Introduction

Automatic detection of facial features is important in computer vision applications such as face recognition, facial emotion recognition, and visual speech recognition (or lipreading). It requires some prior knowledge of the features, such as the shape of the eyes or mouth. Then given a new facial image, these feature locations are detected.

This research is focused on developing an automatic lipreading technique as part of a larger project to build a robust audio-visual speech recognition system. In speech recognition, the visual channel provides an additional or an alternative modality to the audio channel and is used to improve the recognition rate in a noisy environment. From a sequence of speech images, an automatic lipreading system extracts visual features that represent the changes of the mouth shape [3] or/and the inner mouth appearance [9] [2]. We have previously developed two novel techniques for automatic lipreading. One involves tracking the lip contour [1] and extracting the inner mouth appearance by using Cepstral analysis and Higher order Local AutoCorrelation (HLAC) feature extraction [5]. The other involves accommodating for 3D head movement to correct mouth dimensions detected from the image if those are affected by the 3D head rotations [4]. Both of these systems track the locations of specific facial points, namely the outer corners of the eyes, one nostril, and the corners and outer mid points of the lips throughout the image sequence. Our head and lip trackers require the automatic location of these facial points to initialise the systems for tracking.

This paper presents a facial feature detection technique that specifically identifies the facial points required for our lip and head trackers. For this purpose, we adapted the face recognition technique of Okada et al. [8]. They use the Gabor wavelet responses to represent the facial points, and elastic bunch graph matching to connect these points to represent a face. A facial point such as the centre of an eye, nose tip or a mouth corner is represented by a data set called a *jet*, which consists of a 40 dimensional Gabor wavelet response (which is complex valued), which describes the local features of the area surrounding the corresponding facial point. Given a sample jet, a specific facial point can be detected in an unknown face image by using two specific similarity functions that use the amplitude and phase of the wavelet responses embedded in the jet.

We adapted this technique for the detection of seven facial points, namely the outer eye corners, one nostril, and the outer corners and top and bottom mid points of the lips, since these points are required for our lip and head trackers. Our system, however, uses log Gabor wavelets instead of the Gabor wavelets, and represents each facial point with a jet of 24 dimensional log Gabor wavelet responses. A simpler similarity function is used by using the real and imaginary components of the wavelet responses. Instead of the graph matching technique, we solve the feature correspondence problem by employing geometric cross-ratios, which are invariant to 2D rotations and scaling. Our preliminary investigation shows that an average sample jet extracted from three subjects for each of the required facial points can successfully detect the corresponding facial points for a fourth unknown face.

This paper firstly explains the requirements of the facial points detection for our head and lip trackers. Secondly, it describes the feature representation technique that generates the jet by processing log Gabor wavelets on the face image. Then thirdly, it explains the detection algorithm that uses a similarity function to detect facial points as well as a correspondence algorithm to ensure that the facial feature locations are acceptable to form a face. Lastly, our experiment is reported and future development is discussed.

2 Hypothesis

The 3D head tracker [5] detects 3D head orientations from 2D images. It tracks three facial points, which are the outer corners of both eyes and one nostril on a 2D image sequence by using a 2D template matching algorithm, and calculates 3D head orientations using a 3D model-based tracking algorithm. For 2D tracking, an initial template for each of the facial points is sampled from the first image of the sequence. This sample template is updated throughout the sequence during tracking. Currently, the location of the initial sample templates are manually selected. Thus for automation, the head tracker requires automatic detection of these facial points. Figure 1 shows the head tracker. Note that the figure shows that the mouth corners and top and bottom of the lips are also tracked, but the head orientation detection only requires two eye corners and one nostril position.

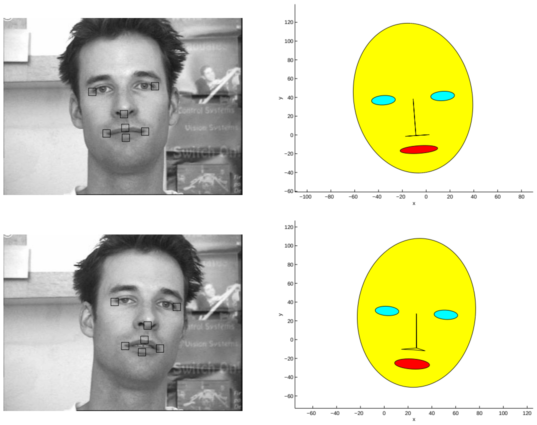

Figure 1. The head tracker tracks facial feature points on a 2D image sequence, and detects the 3D orientations of the head as illustrated by the model.

The lip tracker [1] tracks the outer lip contours by using a modified snake algorithm that employs a combination of 2D template matching and snakes. Given the initial locations for the mouth corners and top and bottom of the lips, the lip tracker samples the neighbourhood templates around the mouth, then tracks the outer lip contours throughout the sequence using the template matching snake. Currently, the initial locations of the four facial points around the mouth are manually selected from the first frame of the image sequence, similar to the head tracker. Figure 2 shows the mouth tracker.

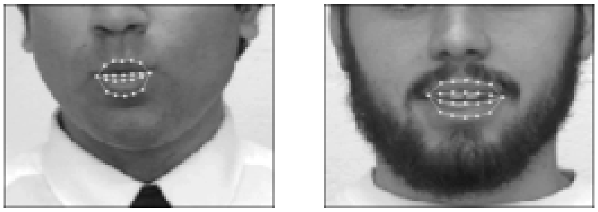

Figure 2. The lip tracker requires initial selection of top, bottom, left and right corner points of the mouth. It then tracks the lip contours using pattern matching snakes.

Both the head and lip tracker assume that in the first frame of the image sequence, the speaker faces the camera directly with a closed mouth position. Together, these trackers use the seven facial points shown in Figure 3.

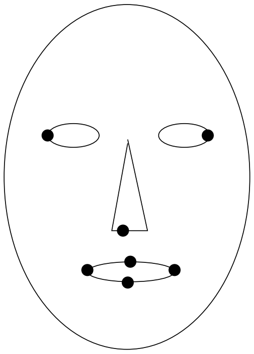

Figure 3. Seven facial points need to be identified for our system, namely, left and right outer eye corners, one nostril, the top and bottom of the lips, and the left and right corners of the mouth.

3 Feature Representation

A facial point is represented by its log Gabor wavelet responses. The log Gabor filters are Gabor filters constructed on the logarithmic frequency scale. The wavelet filters are applied in the frequency domain of the face image, and the filter responses represent the local features surrounding a facial point at different scales.

Kovesi [6] has constructed the 2D log Gabor filter in the frequency domain by using two filter components, namely the radial filter component and the angular filter component.

The radial filter has the transfer function:

$$G(\omega) = e^{\frac{-(\log(\omega/\omega_0)}{2\log(\kappa/\omega_0)}},$$

where ω_0 represents the filter's centre frequency, and κ controls the bandwidth of the filter in the radial direction.

The angular filter has the Gaussian transfer function:

$$G(\theta) = e^{\frac{-(\theta-\theta_0)^2}{2s\Delta\theta^2}},$$

where θ_0 represents the orientation angle of the filter, and s is a scaling factor, and $\Delta\theta$ is the orientation spacing between the filters.

The log Gabor filter is obtained by multiplying its radial and angular components together. Figure 4 shows three even and odd symmetric log Gabor filters with different bandwidths all tuned to the same centre frequency. Each even and odd symmetric pair of log Gabor filters comprise a complex log Gabor filter at one scale.

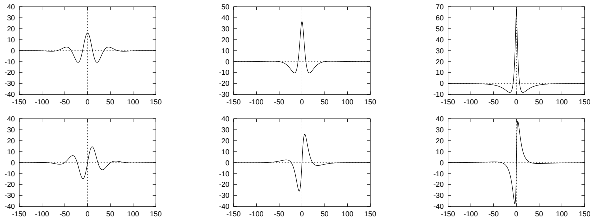

Figure 4. Three quadrature pairs of log Gabor wavelets all tuned to the same frequency, but having bandwidths of 1, 2 and 3 octaves respectively.

In order to represent a facial feature point on an image, we use the complex log Gabor filter in 6 orientations with 4 resolution levels differing by 0.2 octaves, thus generating a filter response data set, the jet, which has a total of 24 complex values.

4 Feature Detection

The feature detection process requires prior knowledge of a sample jet for each of the seven facial points, as well as sample geometric relationships between the seven facial points. Given an unknown face image, the feature detection process finds all possible matched locations for each facial point in the image. For each facial point, there may exist more than one matched location because some facial locations, for example, the left eye corner and the left mouth corner, have similar surrounding features. We solve this correspondence problem by finding the combination of facial points that closely match the sample geometric relationships of the facial points.

4.1 Detecting possible locations

All possible locations for the seven facial points are detected by using the processes shown in Figure 5.

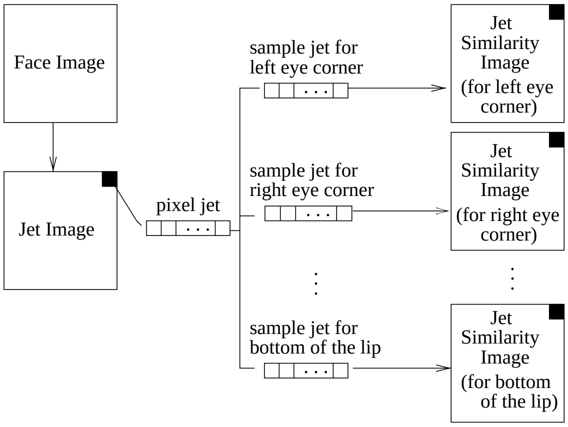
Face Image
Jet Image
pixel jet
sample jet for
left eye corner
sample jet for
right eye corner
sample jet for
bottom of the lip
Jet Similarity Image (for left eye corner)
Jet Similarity Image (for right eye corner)
Jet Similarity Image (for bottom of the lip)

Figure 5. Construction of the jet difference images.

The face image is converted into the jet space by processing each pixel with the 24 log Gabor filters covering six directions and four scales, generating a jet that contains 24 complex valued responses. A jet is produced for each pixel, and represents the features surrounding that pixel. The array of jets corresponding to each pixel value will be referred to as a *jet image*.

The distance D between two jets is calculated by using the Euclidean distance within the complex plane. A jet J is defined by $J = [j_1, j_2, \ldots, j_{24}]$ where each $j_i = (j_{i1}, j_{i2})$. Note that j_{i1} is the real part and j_{i2} is the imaginary part of a log Gabor wavelet response. Given another jet K defined by $K = [k_1, k_2, \ldots, k_{24}]$, we define the distance D between J and K as the sum of the Euclidean distances of the wavelet responses for the corresponding log Gabor filters between the sample jet and the jet image pixel. That is,

$$D = \sum_{i=1}^{24} ||j_i - k_i||.$$

The similarity between the pixel jet of a jet image and the sample jet is calculated by using the following steps:

- Calculate the distance between the sample jet and each pixel jet of the jet image.
- Normalise the jet distances using the maximum and minimum distance calculated within the image, so that the distances lie between 0 and 1.
- Subtract the normalised distances from 1, so that similarity of 0 indicates the minimum similarity within the jet image to the sample and a similarity of 1 indicates the maximum similarity. By arranging these similarities as pixel values, we generate a *jet similarity image* of a specific facial point.

Possible locations for each of the seven facial points are detected using the following steps:

- Using the sample jet of the corresponding facial point, calculate the similarity image of the facial point.
- Threshold the jet similarity image to select the regions of high similarity to the sample jet.
- Within each selected region, find the pixel location of the highest similarity. This is repeated for all regions.

4.2 Correspondence Algorithm

Among the possible locations for all facial points, the correspondence algorithm selects one set of facial points that represents a face. This is achieved by using geometric relationships of the relative angles and Euclidean distances between the 2D facial points, defined by the geometric cross-ratio [7].

Using the seven facial points, two sets of five points are formed. The first set consists of two eye corners, one nostril and two mouth corners, and the second set consists of the nostril, two mouth corners, and top and bottom of the mouth. For each set, we calculate the angle cross-ratio, and the Euclidean distance cross-ratio. Figure 6 illustrates the angle and the distance cross-ratios for the first set of five points.

The geometric cross-ratios are invariant to 2D rotations and scaling. For our lip and head tracker, a speaker is required to face the camera directly with a closed mouth before proceeding to speech. Thus these geometric cross-ratios are adequate for our system to determine the correspondence.

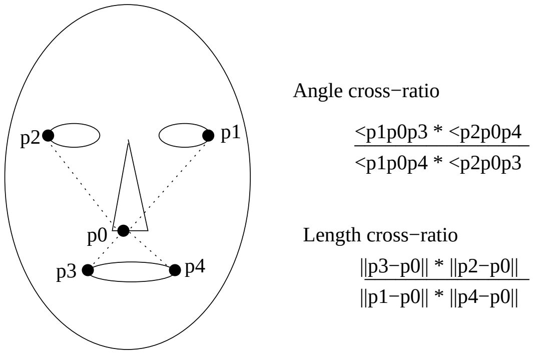

Figure 6. Geometric cross-ratios of a set of facial points.

Given the sample geometric cross-ratios, we search all combinations of the possible facial points to find the combination whose cross-ratio is nearest to the sample.

5 Experiment

An experiment is reported that uses one face image of each of four subjects. The images of three subjects are used to collect the sample jets and sample geometric cross-ratios, and the face image of the fourth subject is used for testing. These face images are extracted from our speech sequence database, that are recorded in a set environment with uncluttered background. The speakers were asked to face the camera directly and were sitting in a fixed position away from the camera.

The technique is implemented using Matlab 6.0 under Linux operating system on a Pentium2 PC.

5.1 Sample Collection

Figure 7 shows the sample face images. Three subjects consist of two male and one female with varying complexion and facial features.

From the sample face images, seven facial points are manually selected and for each of the facial points, a sample jet is generated by averaging the jets which were calculated from each of the three sample faces. The original images are in colour, of size 288x384, but are converted into grey scale images and rescaled to be 300x300 images for the processing of log Gabor wavelets. The log Gabor filters use the minimum wavelength of 15 pixels, using 6 orientations and 4 resolution scales increasing the wavelength by 0.2 octaves.

Figure 7. Three faces are used as samples. From these images, the sample jets are constructed and the sample cross-ratios are calculated.

Using the manually selected facial points, the sample geometric cross-ratios are determined by averaging the cross-ratios of the facial points of three sample faces.

5.2 Facial point detection

For a fair evaluation, we use an unknown face image for the system to identify the facial points. The test face image is shown in Figure 8. The figure also shows the jet similarity images for five of the facial points, namely the left and right outer eye corners, the nostril, and the left and right mouth corners. The superimposed markers indicate the possible locations of the corresponding facial point.

The correspondence algorithm then determines the most likely combination amongst the possible facial locations by using the average geometric cross-ratio properties of the sample data. All of the seven facial points are correctly identified. The result is shown in Figure 9.

6 Summary

We have successfully implemented a facial point identification technique. Log Gabor wavelets are used to represent a facial point with respect to its local surroundings. A simple similarity function between log Gabor wavelet responses was sufficient to find similar facial feature locations for the sample facial point within an unknown face image.

Further testing is required on more subjects with more variety of complexion and features prior to employing it into our lip and head tracker. Two improvements can be made to the system.

One is to detect the face region first so that the facial point detection can be made more efficiently within the face region only. This can be achieved by using the colour distribution within the image.

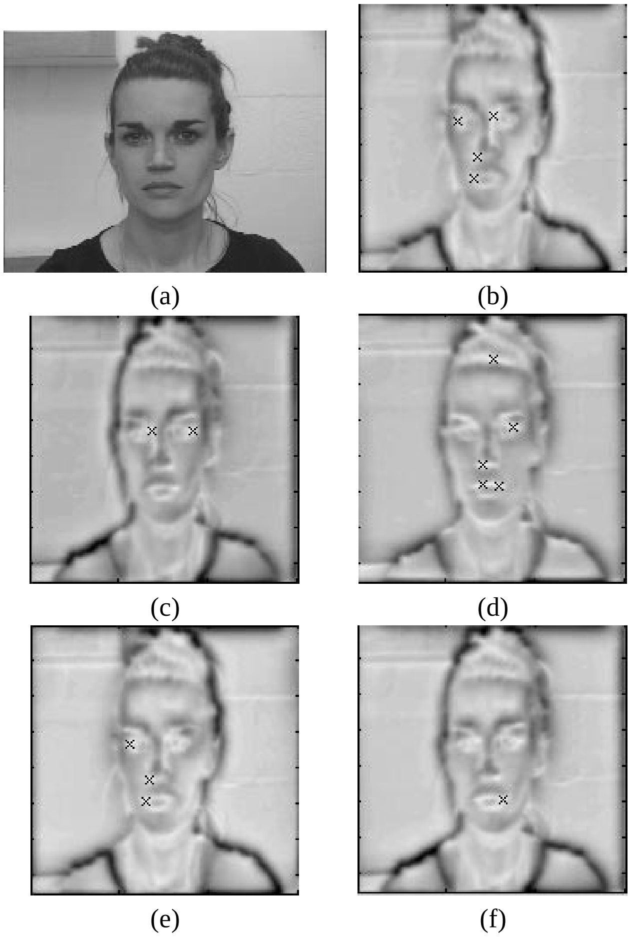

Figure 8. The testing face image is shown in (a). The other images show the jet similarity images and the superimposed markers indicate the detected possible locations of the corresponding facial point: (b) and (c) are for the left and the right eye corner facial points; (d) are for the nostril facial point; (e) and (f) are for the left ane right mouth corner facial points, respectively. Note that the face image of 288x384 is converted into a 300x300 image for log Gabor wavelet processing. Thus the jet similarity images are of size 300x300.

The other is to investigate methods to deal with the different scale of face sizes as well as the facial orientations that may affect the similarity measurements of two jets. Our current system does not require this because the speech sequence is recorded in a fixed environment and having the speaker face the camera directly. However, to improve its practicality, scaling and 3D head orientation must be considered.

Acknowledgement We would like to thank Peter Kovesi for making his effective log Gabor wavelet filters available, as well as for invaluable discussions. This work is supported by the Australian Research Council.

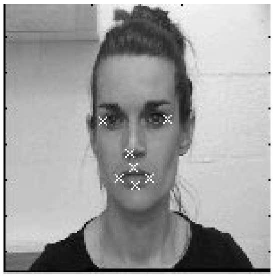

Figure 9. The result of the facial point detection. All of the seven facial points are correctly identified.

References

[1] M. Barnard, E. J. Holden, and R. Owens. Lip tracking using template matching snakes. Technical Report 01/01, Department of Computer Science and Software Engineering, The University of Western Australia, 2001.

[2] C. Bregler and S. M. Omohundro. Nonlinear manifold learning for visual speech recognition. In *Proc. of 5th International Conference on Computer Vision*, pages 494–499, 1995.

[3] G. I. Chiou and J. N. Hwang. Image sequence classification using a neural network based active contour model and a hidden markov model. In *Proc. of International Conference on Image Processing*, pages 926–930, 1994.

[4] E. J. Holden, G. Loy, and R. Owens. Accommodating for 3d head movement in visual lipreading. In *Proceedings of IASTED International Conference on Signal and Image Processing*, pages 166–171, 2000.

[5] E. J. Holden and R. Owens. Visual speech recognition using cepstral images. In *Proceedings of IASTED International Conference on Signal and Image Processing*, pages 331–336, 2000.

[6] P. Kovesi. *Invariant Measures of Image Features from Phase Information*. PhD Thesis, Department of Computer Science and Software Engineering, The University of Western Australia, 1996.

[7] J. Mundy and A. Zisserman. *Gometric Invariance in Computer Vision*. MIT press, 1992.

[8] K. Okada, J. Steffens, T. Maurer, H. H., E. E., N. H., and C. Malsberg. The bochum/usc face recognition system and how it fared in the feret phase iii test. In H. Wechler, P. Phillips, V. Bruce, F. Fogelman'Souve, and T. S. Huang, editors, *Face Recognition: From Theory to Applications*, pages 186–205. Springer-Verlag, 1998.

[9] O. Vanegas, A. Tanaka, K. Tokuda, and T. Kitamura. HMM-based visual speech recognition using intensity and location normalization. In *Proc. of Int. Conf. on Spoken Language Processing*, pages 289–292, 1998.