FACIAL FEATURES DETECTION

Surbhi
Mtech Student, Computer department, SBSCET, Ferozpur
Vishal Arora, Assistant Professor, Computer department
SBSCET, Ferozpur

ABSTRACT
Computer has been widely deployed to our daily lives, but human computer interaction still lacks intuition. Researchers intend to resolve these shortcomings by augmenting traditional systems with human like interaction mechanism. Dedicated hardware often infers the emotional state from human body measures. These have been a considerable amount of research done into the detection and implicit communication channels, including the study of facial feature extraction. [5] The original ASM method developed by Cootes et al. highly relies on the initialization and the representation of the local structure of the facial features in the image. We use color information to improve the ASM approach for facial feature extraction.
Image is converted into 256 X 256 pixel mask using affine transformation's Translation and Rotation Any skin color if found in the image is caught and turned black. A larger area of connected region (skin color) is found that is probable to become a face. The image is converted into a 2 color image (binary image), white representing the normal skin part, black defines the dark edges found on the face image. Using the binary images, we know the edges of facial feature (eyes, lips) now. These are used to extract out only the facial area. Based on the facial image block, eyes and lips are detected in the image. Using Flood Fill technique, each among the three features – left eye, right eye, lips, a shape define by some coordinates is derived out.
Key words: Affine transformation, Thresholding, floodfill algorithm, bezier curve

Introduction

Facial feature extraction consists in localizing the most characteristic face components (eyes, nose, mouth, etc.) within images that depict human faces. This step is essential for the initialization of many face processing techniques like face tracking, facial expression recognition or face recognition. Among these, face recognition is a lively research area where it has been made a great effort in the last years to design and compare different techniques. We intend to present an automatic method for facial feature extraction that we use for the initialization of our face recognition technique. In our notion, to extract the facial components equals to locate certain characteristic points, e.g. the center and the corners of the eyes, the nose tip, etc. Particular emphasis will be given to the localization of the most representative facial features, namely the eyes, and the locations of the other features will be derived from them.

Image Pre-processing

Affine transformations

Image is converted into 256 X 256 pixel mask using affine transformation's Translation and Rotation. An AffineTransform object (see definition below) is contained in the Graphics2D object as part of its state. This AffineTransform object defines how to convert coordinates from user space to device-dependent coordinates in Device Space The AffineTransform class represents a 2D Affine transform that performs a linear mapping from 2D coordinates to other 2D coordinates that preserves the "straightness" and "parallelness" of lines. Affine transformations can be constructed using sequences of translations, scales, flips, rotations, and shears. The coordinate system transformation described by AffineTransform have two very important properties:

- Straight lines remain straight
- Parallel lines remain parallel

An AffineTransform is a linear transform, so the transformation can be expressed in the matrix notation of linear algebra. An arbitraryAffineTransform can be mathematically expressed by six numbers in a matrix like this:

sx shx tx
shy sy ty

To represent affine transformations with matrices, we can use homogeneous coordinates. This means representing a 2-vector (*x*, *y*) as a 3-vector (*x*, *y*, 1), and similarly for higher dimensions. Using this system, translation can be expressed with matrix multiplication. The functional form *x*' = *x* + *tx*; *y*' = *y* + *ty* becomes:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

All ordinary linear transformations are included in the set of affine transformations, and can be described as a simplified form of affine transformations. So, any linear transformation can be also represented by a general transformation matrix. Second one is obtained by expanding the corresponding linear transformation matrix by one row and column, then fill the extra space with zeroes except for the lower-right corner, which must be 1. For example, the ***clockwise rotation matrix*** from *above* becomes:

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Using transformation matrices containing homogeneous coordinates, translations can be seamlessly intermixed with all other types of transformations. When using affine transformations, the homogeneous component of a coordinate vector (normally called *w*) will never be altered. One can therefore safely assume that it is always 1 and ignore it. However, this is not true when using perspective projections. An Affine transformation is a geometrical transformation which is known to preserve the parallelism of lines but not lengths and angles. Affine transforms are usually represented using Homogeneous coordinates: given a point (x,y) in the traditional plane, its canonical Homogenous coordinate is (x,y,1). The Affine transforms are represented in Homogeneous coordinates because the transformation of point A by any Affine transformation can be expressed by the multiplication of a 3x3 Matrix and a 3x1 Point vector.
To compose two Affine transforms, all you need to do is to multiply their matrices to get the matrix representing the resulting Affine transform. Given Affines A and B represented the matrices MA and MB, the Affine C = AoB is represented by the matrix MC = MA x MB.
Those arrays of six doubles can be easily generated with the art_affine_shear, art_affine_scale, art_affine_translate and art_affine_identity functions which generate the affines corresponding to

the given transformations. It is possible to composite Affine transformation's matrices with art_affine_multiply and to invert an Affine transformation: art_affine_invert. Finally, I to apply an Affine transformation to a point, you can use art_affine_point ***In other words, given the values for an x and y coordinate in user space, these multiplicative factors are used calculate new values for those coordinates for use in device space, thereby accomplishing scaling, shear, and translation.***

Scaling Scaling is perhaps the easiest of the four types of transforms to understand. This simply means that if a point is located at a horizontal coordinate value of x in user space, it will be located at sx*x in device space, where sx is a numeric multiplier that can be either positive or negative.

Translation The purpose of translation is to move the origin of the coordinate system in device space. For example, the default position of the origin is the upper left-hand corner of the component on which the graphic is being displayed. Assume that the component is a Frame object that is four inches on each side. You might like for the origin to be in the center of the Frame instead of at the top left-hand corner. You could accomplish this by translating the origin by two inches in both the horizontal and vertical directions. Another use of translation (in combination with scaling) is to flip the default positive direction of the vertical axis so that increasing positive values go up instead of down, which is the default. I will leave the implementation of this as an exercise for the student.

Shear I like the way that Jonathan Knudsen describes shear in his book entitled Java 2D Graphics. He describes it something like this. Take a stack of paper (like you might place in your laser printer) and draw a picture on the side of the stack. Then deform the stack so that the side of the stack on which you drew the picture takes the form of a parallelogram. Be especially careful to keep the opposite sides parallel. Then look at your picture. It will have been subjected to shear in one dimension.

Rotation Rotation is also fairly easy to visualize (but the combination of rotation and shear can be very difficult to visualize). To visualize rotation, draw a picture on a piece of paper. Use a single tack to attach it to a bulletin board. Then rotate the paper a few degrees around the tack. You will have rotated the picture around the coordinates of the tack by the specified number of degrees.

Thresholding

This method is relatively simple, does not require much specific knowledge of the image. The basic idea here is that,

- If the average value is greater than the threshold value, color output is white (1)
- If the average value is small than the threshold value, color output is black (0)

Threshold value defined here is 110

We took Threshold value as 110 because skin color is always very much light as compare to the color of hairs or any darkest region around the face of a person and this skin color take overall value of R, G and B components as 110 according to the Drawing class for colors in .net. for this reason for matching with skin color we took threshold as 110 as it is near to this value in case of skin color

IV. Connected Regions

We used the concept of region growing technique of segmentation here, what happens in this case we select a seed pixel and according to its criteria we found the neighbor pixels which full fill the same criteria and if we have such neighbor pixels then we consider those pixels are connected to seed pixels and with the similar manner we proceed with region growing algorithm and find the maximum connected area.

V. Binary Image

In this case all the pixels which comes under the maximum connected regions got the gray scale value as 255 and those which are not under maximum connected region got gray scale value as 0. These pixels are selected according to skin color segmentation.

VI. Face Area

The connected pixels form the face area. To check the face area we compare the width and height of the image. If the largest connected region's height & width is *larger or equal than 50* and the ratio of height / width is between 2 : 1, then it may be a face.

VII. Eye Lip Detection

Eye Detection Coordinate regions decide or locate the position of Eyes and lips. we convert the RGB face to the binary face. We consider the face width by W. Scan from the W/4 to (W-W/4) to find the middle position of the two eyes. The highest white continuous pixel along the height between both is the middle position of the two eyes. Then we scan the starting high or upper position of the two eyebrows by searching vertical. For left eye, we search w/8 to mid and for right eye we search mid to w – w/8. Here w is the width of the image and mid is the middle position of the two eyes.

For left eye, we search from the mid/4 to mid - mid/4 width. And for right eye, we search mid + (w-mid)/ 4 to mid+3*(w- mid)/ 4 width from image lower end to starting position of the eyebrow.

Then we find the right side of the left eye by searching black pixel horizontally from the mid position to the starting position of black pixels in between the upper position and lower position of the left eye.

Lip Detection

We determine the lip box and consider that lip must be inside the lip box.

- We determine the distance between the forehead and eyes. Then we add the distance with the lower height of the eye to determine the upper height of the box which will contain the lip.
- Now, the starting point of the box will be the ¼ position of the left eye box and ending point will be the ¾ position of the right eye box. And the ending height of the box will be the lower end of the face image.
- So, this box will contain only lip and may some part of the nose. Then we will cut the RGB image according the box.

VIII. Bezier Curves

Flood fill algorithm is used here after detection of facial features, to find the exact shape of the Bezier curve from eyes and lips. Steps performed by this algorithm are:

- It first detects the eyes and lips which area already detected in previous phase.
- Then fill algorithm fills the shape with a single color black.(the inner area of shape is given value - 0)
- Outer most pixels are still the same, but pixels inside them are given a value of 1 (white).

This procedure derives a Bezier curve as output.

Literature survey

As one of the most successful applications of image analysis and understanding, face recognition has recently received significant attention, especially during the past several years. At least two reasons account for this trend: the first is the wide range of commercial and law enforcement applications, and the second is the availability of feasible technologies after 30 years of research. Even though current machine recognition systems have reached a certain level of maturity, their success is limited by the conditions imposed by many real applications. For example, recognition of face images acquired in an outdoor environment with changes in illumination and/or pose remains a largely unsolved problem. In other words, current systems are still far away from the capability of the human perception system. The main conclusion of this review of face and feature detection is that is very difficult to compare algorithms due to the lack of common test sets. This is especially true for feature based face detection and feature localisation methods. For template based face detection and multi-view face detection the CMU data sets provide common data to compare results. Therefore whole face template methods are more thoroughly tested and appear to provide more robust results, compared to feature based detection, especially for low resolution upright faces. Therefore the main approach to face detection in this thesis is template based. Face detection has been one of the most studied topics in the computer vision literature. In this technical report, we survey the recent advances in face detection for the past decade. The seminal Viola-Jones face detector is first reviewed. We then survey the various techniques according to how they extract features and what learning algorithms are adopted. It is our hope that by reviewing the many existing algorithms, we will see even better algorithms developed to solve this fundamental computer vision problem. To be able to develop and test robust affective multimodal systems, researchers need access to novel databases containing representative samples of human multi-modal expressive behavior. The creation of such databases requires a major effort in the definition of representative behaviors, the choice of expressive modalities, and the collection and labeling of large amount of data. At present, public databases only exist for single expressive modalities such as facial expression analysis. There also exist a number of gesture databases of static and dynamic hand postures and dynamic hand gestures. However, there is not a readily available database combining affective face and body information in a genuine bimodal manner. Accordingly, in this study, a bimodal database presented, this is recorded by two high resolution cameras simultaneously for use in automatic analysis of human nonverbal affective behavior. [15]

Conclusion

In this paper we have concluded that image is converted into 256 X 256 pixel mask using affine transformation’s Translation and Rotation.Any skin color if found in the image is caught and turned black. A larger area of connected region (skin color) is found that is probable to become a face. The image is converted into a 2 color image (binary image), white representing the normal skin part, black defines the dark edges found on the face image. Using the binary images, we know the edges of facial feature (eyes, lips) now. These are used to extract out only the facial area. Based on the facial image block, eyes and lips are detected in the image.Using Flood Fill technique, each among the three features – left eye, right eye, lips, a shape define by some coordinates is derived out. This is called a Bezier curve.

REFERENCES

[1]. RokGajsek, VitomirStruc, France Mihelic Multi-Model Emotion Recognition Using Canonical Correlations and Acoustic features in International Conference on Pattern Recognition, 2011

[2]. Matthias Wimmer, Christoph Mayer, Sylvia Pietzsch, and Bernd Radig ,Tailoring Model-based Techniques to Facial Expression Interpretation in First International Conference on Advances in Computer-Human Interaction 0-7695-30876-9/08 IEEE DOI 10.1109/ACHI.2008.7
[3]. George A. Tsihrintzis, Maria Virvou, EfthymiosAlepis, and Ioanna-OuraniaStathopoulou Towards Improving Visual-Facial Emotion Recognition through Use of Complementary Keyboard-Stroke Pattern Information in 978-0-7695-3099-4/08, 2008 IEEE DOI 10.1109/ITNG.2008.152
[4]. KiyhoshiNosu and TomoyaKurokawa, A Multi-Modal Emotion-Diagnosis System to Support e-Learning in Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC'06)0-7695-2616-0/0, 2006
[5]. HaticeGunes and Massimo Piccardi, A Bimodal Face and Body Gesture Database for Automatic Analysis of Human Nonverbal Affective Behavior in Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), 2006
[6]. Nugraha P. Utama, Atsushi TakemotoYasuharu Koike, Katsuki Nakamura, Phased processing of facial emotion: An ERP study in Neuroscience Research 64 (2009) 30–40. Journal homepage: www.elsevier.com/locate/neures
[7]. R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Vot- sis, S. Kollias,W. Fellenz, and J. Taylor. Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine*, 18 (1)(1):32 – 80, January 2001.
[8]. F. Eyben, M. Wllmer, and B. Schuller. openear - introducing the munich open-source emotion and affect recognition toolkit. In *Proc. of ACII 2009*, Amsterdam, pages 576–581., 2009.
[9]. T. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonicalcorrelations. *TPAMI*, 29(6):1005–1018, June 2007.
[10]. M. Mansoorizadeh and N. M. Charkari. Multimodal information fusion application to human emotion recognition from face and speech.Multi.Tools and App, 2009.
[11]. O. Martin, I. Kotsia, B. Macq, and I. Pitas. The enter-face’05 audio-visual emotion database.In *ICDEW ’06*,Washington, DC, USA, 2006.
[12]. M. Paleari, R. Benmokhtar, and B. Huet. Evidencetheory-based multimodal emotion recognition. In*MMM’09*, pages 435–446, Berlin, 2008.
[13]. J. Pittermann, A. Pittermann, and W. Minker. *Handling Emotions in Human-Comp. Dialog*.Springer,Dordrecht (The Netherlands), 2009.
[14]. B. Schuller. Speaker, noise, and acoustic space adaptation for emotion recognition in the automotive environment.In *Proc. 8th ITG conf. on Speech Comm.*, 2008.
[15]. B. Schuller, S. Steidl, and A. Batliner. The interspeech2009 emotion challenge.In *Proc. Interspeech 2009*.
[16]. N. Sebe, I. Cohen, and T. G. T. Huang. Multimodalapproaches for emotion recognition : A survey. In *Proc.of SPIE*, volume 5670, pages 56–67, January 2005.
[17]. P. Viola and M. Jones. Robust real-time face detection.*Int. J. of Comp. Vision*, 57(2):137 – 154, 2004.
[18]. Gibson. 1.1.. The Perception of the Visual World (Riverside Press, Cambridge. 1950)

[19]. Gibson.1.1.The Senses Considered as Perceptual Systems (Houghton-Mi Win, Boston. MA, 1966).
[20]. Gibson, J.J, On the analysis of change in theoptic array. Scandinavian I. Psychol. 18 (1977)161-163.

[21]. Nakayama. K. and laomis. J.M.. Optical velocity patterns. Velocity sensitive neurons and pace perception. Perception 3 (1974) 63-80.
[23] B.H. Sonka, Image Processing, Analysis, and Machine Vision [1999], pp. 757-759, 808.
[24]. Liyanage C. DE SILVA**, I** Tsutomu MIYASATO , Ryohei NAKATSU ,Facial Emotion Recognition using Multi-Model information in International conference on “Information, Communications and Signal Processing” ICICS’ 97 , Singapore 9-12 September, 1997.
[25]. O. Yamaguchi, K. Fukui, and K. Maeda. Face recognition using temporal image sequence. In *Proc. of AFGR*,pages 318–323, 1998.
[26]. Face Recognition: A Literature Survey W. ZHAO**,**R. CHELLAPPA**,**P. J. PHILLIPS AND A.ROSENFELD.
[27]. Automatic Detection of Facial Features in Grey Scale Images David Cristinacce Imaging Science and Biomedical Engineering,2004.