ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 3, March 2017

Detection and Tracking of Facial Features in Video Sequences

[1]B. Sandhya, [2]V. Sri Lakshmi Priyanka, [3]CH. Kavya
[1] Student, [2] Student, [3] Faculty SPMVV

Abstract: -- **In this paper, we present face detection and tracking algorithm in real time camera input environment. The entire face tracking algorithm is divided into two modules. The first module is face detection and second is face tracking. To detect the face in the image, Haar based algorithm is used. On the face image, Shi and Thomasi algorithm is used to extract feature points and Pyramidal Lucas-Kanade algorithm is used to track those detected features. Results on the real time indicate that the proposed algorithm can accurately extract facial features points. The algorithm is applied on the real time camera input and under real time environmental conditions.**

Index Terms — **Face detection, face tracking, optical flow method, and Pyramidal Lucas kanade algorithm.**

I. INTRODUCTION

Facial Features tracking is a fundamental problem in computer vision due to its wide range of applications in psychosomatic facial expression investigation and human computer interfaces. Modern developments in face video processing and compression have made face-to face communication be applied in real world applications. And after decades, robust and realistic real time face tracking still poses a big challenge. The difficulty lies in a number of issues including the real time face feature tracking under a wide variety of imaging environments (e.g., skin color, pose change, self-occlusion and multiple non-rigid features deformation). In this paper, we focus our work on facial feature tracking. To identify the face in the image, we have used a face detector based on the Haar-like topographies. This face detector is fast and robust to any brightness situation. For feature point extraction, we have used Shi and Tomasi method . In order to track the facial feature points, Pyramidal Lucas-Kanade Feature Tracker algorithm is used. Pyramidal Lucas Kanade algorithm is the commanding optical flow algorithm used in tracking. It tracks preliminary from maximum level of an image pyramid and working down to lower levels. Tracking over image pyramids permits large motions to be caught by local windows.

II. PROPOSED ALGORITHM

The algorithm used to track the face is given below:

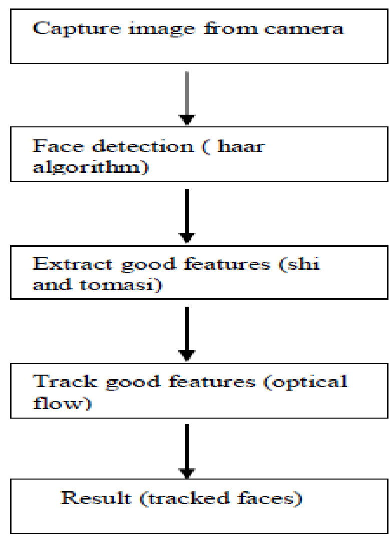

Figure 1. Proposed algorithm to track the face

3. Motion Detection Based on Frame Difference Method Human body motion analysis has been an interesting research for its various applications, such as physical performance, evaluation, medical diagnostics, virtual reality. At present methods used in moving object detection are mainly the frame subtraction method, the background subtraction method and the optical flow method. Optical flow method is to calculate the image optical flow field, and do cluster processing according to the optical flow distribution characteristics of image.

This method can get the complete movement information and detect the moving object from the background better. The background subtraction method is to use the difference method of the current image and background image to detect moving objects, with simple algorithm, but very sensitive to the changes in the external environment and has poor anti-interference ability. In the frame subtraction method the presence of moving objects is determined by calculating the difference between two consecutive images. Any motion detection system based on background subtraction needs to handle a number of critical situations such as:

- Image noise, due to a poor quality image source.
- Gradual variations of the lighting conditions in the scene.
- Small movements of non-static objects such as tree branches and bushes.
- Blowing in the wind.
- Shadow regions are projected by foreground objects and are detected as moving objects.

Detection of moving object from a sequence of frames captured from a static camera is widely performed by frame difference method. The objective of the approach is to detect the moving objects from the difference between the existing frame and the reference frame. The frame difference method is the common method of motion detection. This method adopts pixel-based difference to find the moving object.

4. Face detection and Face tracking

For face detection, we have used Viola & Jones's face detector based on the Haar-like features. Paul Viola and Michael Jones, describes a visual object detection framework that is capable of processing images extremely rapidly while achieving high detection rates. There are three key contributions. The first contribution is a new a technique for computing a rich set of image features using the integral image. The second is a learning algorithm, based on AdaBoost, which selects a small number of critical visual features and yields extremely efficient classifiers. The third contribution is a method for combining classifiers in a "cascade" which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions.

4.1 Face Detection

Face detection is the task of determining the locations of human faces in digital images and it is the first step in any face processing application. The goal of face detection is to locate the facial features of a human. Face tracking systems assume that the initial location of the face is known . The human face is a dynamic object and has a high degree of variability in its appearance, which makes face detection an easy problem which humans can do effortlessly but a difficult problem in computer vision.

4.2 Face Tracking

Face tracking has become one of the most challenging problems of the object tracking field, owing to the large variability of faces and facial expressions that exist and the number of context where they can be found. Face tracking in a image sequence is a major research topic in computer vision and it has wide application in the areas such as intelligence visual surveillance, human-computer interaction, video coding, teleconference, face animation etc. Tracking requires a method for face detection which determines the image location of a face in the first frame of a video sequence.

Face tracking is a computer technique to keep the trace of similar detected faces within a video sequence. This is not an easy task because a face may vary its pose and appearance along the time, which means that what is being tracked will not look the same in every frame. Face tracking is done in two methods. The first method tracks the face by detecting the face region in each and every frame. The second method detects the face region in the first frame and the detected face region is tracked in the consecutive frames by classifying the face region from non-face regions. Face tracking is a challenging task due to factors such as background, illumination, noise, geometry (size, position, orientation and pose) and structural components as discussed below:

- Illumination variations: Illumination variations often lead to a loss of track. For example, if the camera is situated inside a car trying to track the driver's head, and suddenly if he drives into a tunnel, the illumination conditions would change a lot.
- Face size: The size of the face in an image depends on the distance of the subject with respect to the camera and the actual face size of the subject. It is difficult to control the position of the subject with respect to the camera. There sizing operation is commonly used in the literature to get a face image of fixed size, and it will change the appearance of the face image.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 3, March 2017

- Pose variations: In some situations the face that is being tracked is not frontal. These pose changes make the work more challenging. Hence some face detection techniques do not work when pose variation occurs. For example, when the subject turns his face to the right or left, the whole pattern that is being tracked changes, and hence, the tracker has to deal with it and follow a new pattern.
- Presence or absence of structural components: Facial features such as beards, mustaches, and glasses may or may not be present and there is a great deal of variability among these components including shape, colour and size.
- Facial deformations: Tracking faces through changes of expressions is another challenging problem.
- Occlusion and clutter: As with most tracking problems, occlusion and clutter affect the performance of face trackers. A simple approach to recover from a loss of track is to compute the mean face that has been tracked so far. With this technique, once the occlusion disappears the tracker would be able to find the face again using the mean face.
- Aging: The face tracking system should be designed to operate over a longer period of time. The changes in the facial appearance of an individual over a period of time must be addressed.

4.3 Integral image

Rectangular features can be computed very rapidly using an intermediate representation for the image called the integral image. The integral image at location (x, y) contains the sum of the pixels above and to the left of (x, y), inclusive:

$$ii(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i(x', y')$$

Where i(x, y) is the original image and ii(x, y) is the integral image. Using the following pair of recurrences:

$$s(x, y) = s(x,y-1)+ i(x, y) \qquad (1)$$

$$ii(x, y) = ii(x-1,y) +s(x, y) \qquad (2)$$

Where s(x, y) is the cumulative row sum, s (x, -1) = 0 and ii (-1, y) = 0 the integral image can be computed in one pass over the original image.

Using the integral image any rectangular sum can be computed in four array references (Figure 2). Clearly the difference between two rectangular sums can be computed in eight references. Since the two rectangle features defined above involve adjacent rectangular sums they can be computed in six array references, eight in the case of the three-rectangle features, and nine for four-rectangle features.

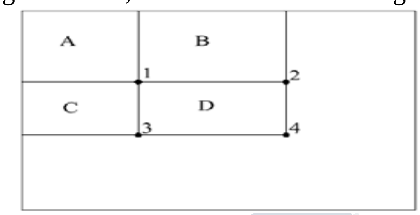

Figure 2. Sum of pixel values within "D".

The Above Figure 2 showing that the sum of the pixels within rectangle D can be computed with four array references. The value of the integral image at location 1 is the sum of the pixels in rectangle A. The value at location 2 is A + B, at location 3 is A + C, and at location 4 is A + B + C + D. The sum within D can be computed as 4 + 1 - (2 + 3).

5. Facial feature extraction

For facial feature points extraction we have used Shi and Tomasi method. This method is based on the general assumption that the luminance intensity does not change for image acquisition. To select interest points, a neighborhood N of nxn pixels is selected around each pixel in the image. The derivatives Dx and Dy are calculated with a Sobel operator for all pixels in the block N. For each pixel the minimum eigenvalue λ is calculated for matrix A where

And Σ is performed over the neighborhood of N. The pixels with the highest values of λ are then selected by thresholding.

The next step is rejecting the corners with the minimal Eigen value less than some threshold. Finally, a test is made, all the found corners are distanced enough from one another by getting two strongest features and checking that the distance between the points is satisfactory. If not, the point is rejected.

5.1 Feature selection

The main purpose of using features rather than the pixels directly is that features can act to encode ad-hoc domain knowledge that is difficult to learn using a finite quantity of training data. Also the the feature-based system operates much faster than a pixel-based system. The simple features used are reminiscent of Haar basis functions which have been used by Papageorgiou et al.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 3, March 2017

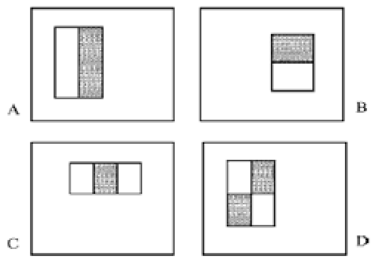

Figure 3. Rectangle features shown relative to the enclosing detection window.

The sum of the pixels which lie within the white rectangles are subtracted from the sum of pixels in the grey rectangles. Two-rectangle features are shown in (A) and (B). Figure (C) shows a three-rectangle feature, and (D) a four-rectangle feature.

6. Algorithms

6.1 Optical flow methods

There are different methods that add some constraint to the problem, in order to estimate the optical flow. Some of them are:

1. Block based methods: Minimizing sum of squared differences or sum of absolute differences, or maximizing normalized cross-correlation.

2. Discrete optimization methods: The whole space is quantized, and every pixel is labeled, such that the corresponding deformation minimizes the distance between the source and the target image. The optimal solution is often computed through min-cut max-flow algorithms or linear programming.

3. Differential methods: The differential methods of optical flow estimation, based on partial spatial and temporal derivatives of the image signal, as following:

- Lucas kanade method: Dividing image into patches and computing a single optical flow on each of them
- Horn schunck method: Optimizing a function-define the notion of similarity in a 2D neighborhood based on residuals from the brightness constancy constraint, and a particular regularization term expressing the expected smoothness of the flow field.
- Buxton buxton method: Based on a model recovered from the motion of edges in image sequences.
- General variation methods: Range of the extensions or modifications of Horn-schunck, using other data terms and other smoothness terms.

Here we are going to explain Lucas kanade method.

6.2 Lucas kanade method

The Lucas Kanade (LK) algorithm, as originally proposed in 1981, was an attempt to produce dense results. Yet because the method is easily applied to a subset of the points in the input image, it has become an important sparse technique. The LK algorithm can be applied in a sparse context because it relies only on local information that is derived from some small window surrounding each of the points of interest. The disadvantage of using small local windows in Lucas-Kanade is that large motions can move points outside of the local window and thus become impossible for the algorithm to find. This problem led to development of the “pyramidal” Lucas Kanade algorithm, which tracks starting from highest level of an image pyramid (lowest detail) and working down to lower levels (finer detail). Tracking over image pyramids allows large motions to be caught by local windows. The basic idea of the LK algorithm rests on three assumptions:

1. Brightness constancy: A pixel of an object in an image does not change in appearance as it (possibly) moves from frame to frame. For grayscale image, this means we assume that the brightness of a pixel does not change as is tracked from frame to frame.

2. Temporal persistence or small movements: The image motion of a surface patch changes slowly in time. In practice, this means the temporal increments are fast enough relative to the scale of motion in the image that the object does not move much from frame to frame.

3. Spatial coherence: Neighboring points in a scene belong to the same surface, have similar motion, and project to nearby points on the image plane.

6.3 Pyramidal lucas-kanade feature tracker

Pyramidal lucas kanade algorithm is the powerful optical flow algorithm used in feature tracking. Consider an image point u = (ux, uy) on the first image I, the goal of feature tracking is to find the location v = u + d in next image J such as I(u) and J(v) are “similar". Displacement vector d is the image velocity at x which also known as optical flow at x. Because of the aperture problem, it is essential to sense. Let ωx and ωy are two integers, then d the vector that minimizes the residual function defined as follows:

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 3, March 2017

$$\epsilon(\mathrm{d}) = \epsilon(d_x, d_y) = \sum_{x=u_x-\omega_x}^{u_x+\omega_x} \sum_{y=u_y-\omega_y}^{u_y+\omega_y} \left(I(x,y) - J(x+d_x, y+d_y)\right)^2$$

Observe that following that definition, the similarity function is measured on a image neighborhood of size (2ωx + 1) x (2ωy +1). This neighborhood will be also called integration window. Typical values for ωx and ωy are 2,3,4,5,6,7 pixels.

7. Implementation and Results

The tracking results shown below are captured under occlusions, pose variations, scales, and illumination changes. Figure 4 illustrates face tracking results with different poses and under illumination changes. Figure 5 illustrates face tracking results where occlusions happen.

Figure 4. Face tracking results with different poses and under illumination changes

Figure 5. Face tracking results with occlusions

8. Conclusion and Future work

In this paper, we have presented a face tracking algorithm in real time camera input environment. To detect the face in the image, we have used a face detector based on the Haar-like features. This face detector is fast and robust to any illumination condition. For feature points extraction, we have used the algorithm of Shi and Tomasi to extract feature points. This method gives good results. To track the facial feature points, Pyramidal Lucas- Kanade Feature Tracker KLT algorithm is used. Using detected points with the algorithm of Shi and Tomasi, we have got good results in video sequence and in real time acquisition. The obtained results indicate that the proposed algorithm can accurately extract facial features points. The future work will include extracting feature points with some conditions to limit the number of feature points in bounding box and choose only the points which describe well the shape of the facial feature. This work will be used for real time facial expression recognition application. Further the work can be extended by detecting faces at different inclinations or slopes as Haar classifier has its own limitations.

REFERENCES

[1] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” Proceedings of the DARPA imaging understanding workshop, pp. 121–130, 1981.

[2] J. Quinlan, “Induction of decision trees,” Machine Learning, Journal, vol. 1, no. 1, pp. 81-106, 1986.

[3] M. Lades, J.C. Vorbrüggen, J. Buhmann , J. Lange,

C. Malsburg, R. Würtz., and W. Konen, “Distortion invariant object recognition in the dynamic link architecture,” IEEE Trans. Computer, vol. 42, no. 3, pp. 300-310, 1993.

[4] J. Shi and C. Tomasi, “Good Features to Track,” IEEE Conference on Computer Vision and Pattern Recognition, pp. 593-600, 1994.

[5] Y. Freund and R.E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” in European Conference on Computational Learning Theory, pp. 23–37, 1995.

[6] C. Papageorgiou, M. Oren, and T. Poggio, “A general framework for object detection,” In International Conference on Computer Vision, vol. 21, pp. 555–562, 1998.

[7] K. Sung and T. Poggio, “Example-based learning for view-based face detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 20, pp. 39–51, 1998.

[8] J.Y.Bouguet,“Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm,” Intel Corporation, Microprocessor Research Labs, pp. 1-9, 2000.

[9] P. Viola and M. Jones, “Robust Real-time Object Detection,” 2nd international workshop on statistical and computational theories of vision - modeling, learning, computing, and sampling, pp. 1-25, 2001.