13

Real Time Facial Feature Points Tracking with Pyramidal Lucas-Kanade Algorithm

F. Abdat, C. Maaoui and A. Pruski

Laboratoire d'Automatique humaine et de Sciences Comportementales, Université de Metz France

1. Intoduction

Facial expression tracking is a fundamental problem in computer vision due to its important role in a variety of applications including facial expression recognition, classification, and detection of emotional states, among others H. Xiaolei (2004). Research on face tracking has been intensified due to its wide range of applications in psychological facial expression analysis and human computer interaction. Recent advances in face video processing and compression have made face-to-face communication be practical in real world applications. However, higher bandwidth is still highly demanded due to the increasing intensive communication. 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 variety of imaging conditions (e.g., skin color, pose change, self-occlusion and multiple non-rigid features deformation) K. Ki-Sang (2007).

Our study aims to develop an automatic facial expression recognition system. This system analysis the movement of the eyebrows, lips and eyes from video sequences, to determine whether a person is happy, sad, disgust or fear.

In this paper, we concentrate our work on facial feature tracking. Our real time facial features tracking system is outlined in figure 1, which is constituted of two important modules:

1. Extract features in facial image, using a geometrical model and gradient projection Abdat et al. (2008).
2. Facial feature points tracking with optical flow (pyramidal Lucas-Kanade algorithm) Bouguet (2000).

The organization of this paper is as follows: in section 2, we will present a face detection algorithm with HAAR-like features. Facial feature points extraction with a geometrical model and gradient projection will be described in section 3. The tracking of facial feature points with Pyramidal Lucas-Kanade will be presented in section 4. Finally the concluding remarks will be given in section 5.

2. Face detection

Face detection is the first step in our facial expression recognition system, which consist to delimit the face area with a rectangle. For this, we have used a modified Viola & Jones's face detector based on the Haar-like features Viola & Jones (2001).

Source: Human-Robot Interaction, Book edited by: Daisuke Chugo,
ISBN 978-953-307-051-3, pp. 288, February 2010, INTECH, Croatia, downloaded from SCIYO.COM

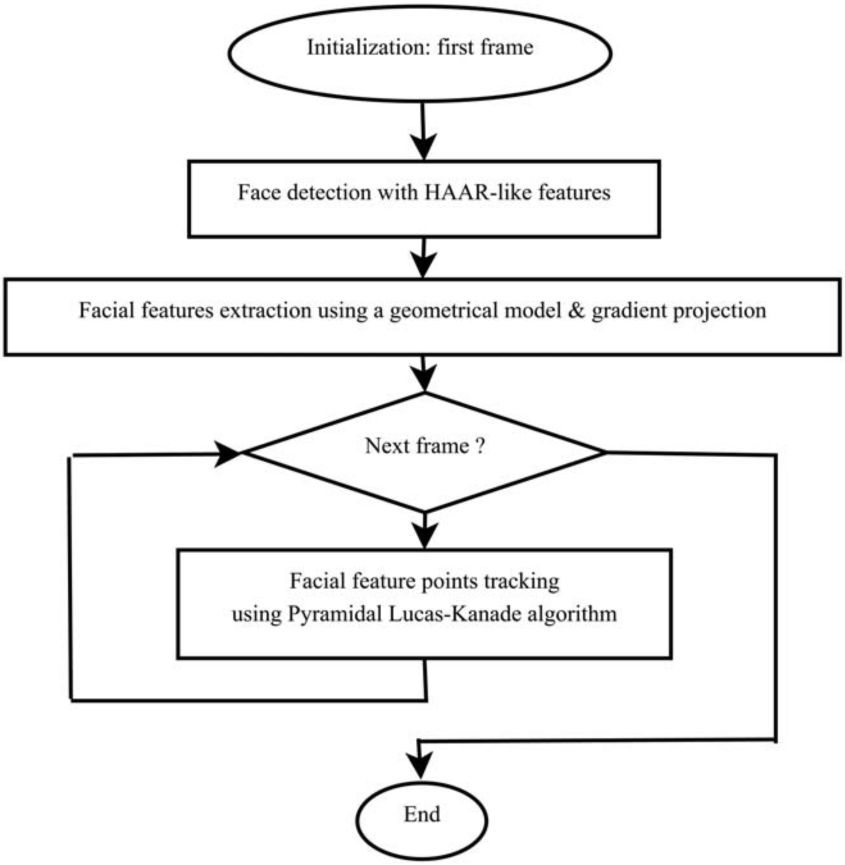

Initialization: first frame
Face detection with HAAR-like features
Facial features extraction using a geometrical model & gradient projection
Next frame ?
Facial feature points tracking
using Pyramidal Lucas-Kanade algorithm
End

Fig. 1. Real time facial feature points tracking system.

A statistical model of the face is trained. This model is made of a cascade of boosted tree classifiers. The cascade is trained on face and non-face examples of fixed size 24x24. Face detection is done using a retinal approach. A 24x24 sliding window scans the image and each sub-image is classified as face or non-face. To deal with face size the cascade is scaled with a factor of 1.2 by scaling the coordinates of all rectangles of Haar-like features.

2.1 Haar-like features

The pixel value inform us only about luminance and color of a given point. It is therefore more interest to find a detectors based on more global characteristics of the object. This is the case of HAAR descriptors, where the functions allow the knowledge of the contrasts difference between several rectangular regions in image. They encode the existing contrasts in a face and their spatial relationships.

Figure 2 represents the shapes of the used features. Actually, hundreds of features are used as these shapes are applied at different position in the 24x24 retina; a feature is defined by its shape (including its size depending on a scale factor that define the expected face size) and its location.

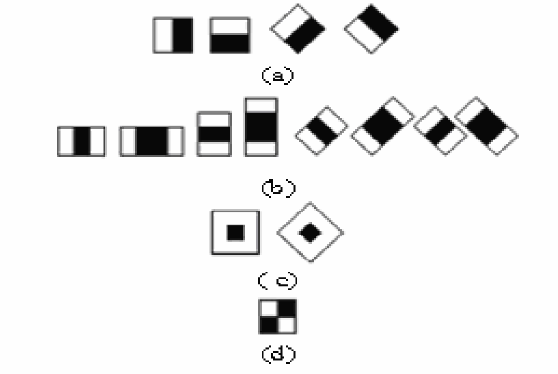


Fig. 2. Haar-like feature extended set.

A feature's value is the weighted sum of pixels over the whole area added to the weighted sum over the dark rectangle R. Belaroussi & Milgram (2006). Absolute value of black area weight is inversely proportional to its area as shown in Figure 3.

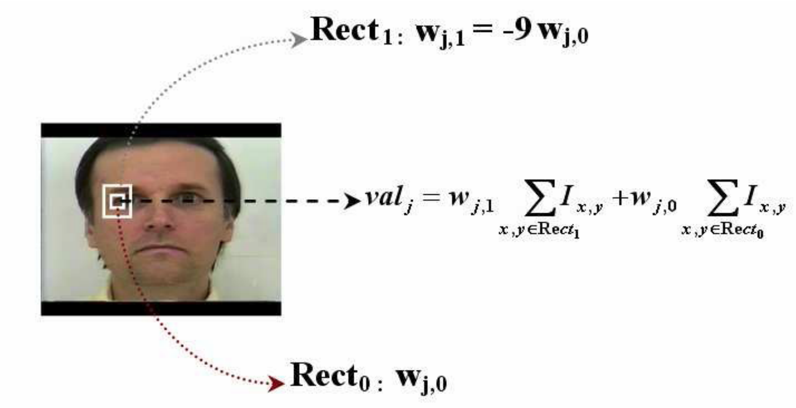

$\mathbf{Rect}_1 : \mathbf{w}_{j,1} = -9\,\mathbf{w}_{j,0}$
$val_j = w_{j,1} \sum_{x,y \in Rect_1} I_{x,y} + w_{j,0} \sum_{x,y \in Rect_0} I_{x,y}$
$\mathbf{Rect}_0 : \mathbf{w}_{j,0}$

Fig. 3. Shape and location in the search window of the feature *j*.

2.2 Cascade classifier

A simple decision tree classifier, referred to as "weak" classifier, processes the feature value. A complex classifier

$$F_k = sign(\sum_{i=1}^{n}(c_i f_i)) \tag{1}$$

is iteratively computed as a weighted sum of weak classifiers using a boosting procedure. At each iteration, a weak classifier parameters are trained and a weight c_j is assigned to the weak classifier relatively to its error on the training set. The trained weak classifier is then added to the sum and the training samples weights are updated in order to emphasize the misclassified ones. Finally, an intentional cascade is implemented: it is a cascade of boosted classifiers with increasing complexity. As shown in Figure 4, the simplest classifiers comes first and is intended to reject majority of sub-window before calling more complex classifiers P. Viola & M. Jones (2001).

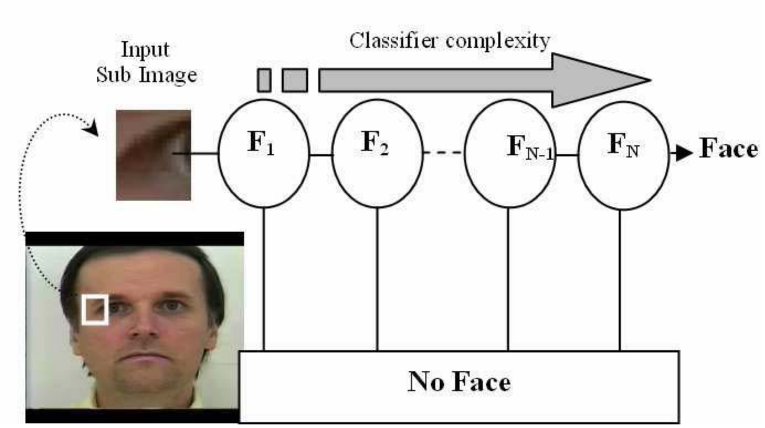


Fig. 4. Cascade of boosted classifiers.

The real-time implementation of this detector using our database, shows that the detector is fast (~10 frames per second) and robust to illumination conditions (Figure 5). However, the detector work hardly when face pose is too slanted. Figure 6 illustrates the limitation of this detector where the bowed face was not detected.

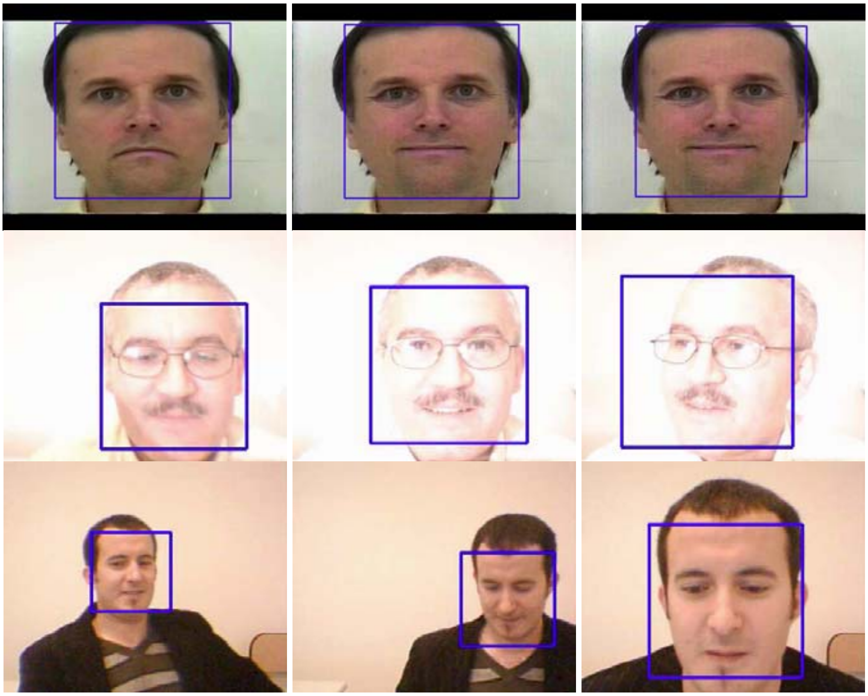


Fig. 5. Face detector.

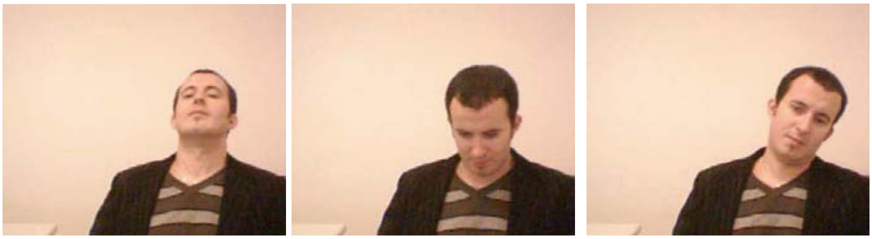


Fig. 6. Limits of the face detector.

3. Facial feature extraction

After face detection in the first frame, the next step is to extract necessary information about the facial expression presented in the image sequence. When facial muscles contract, the transformation of the corresponding skin areas attached to the muscles produces changes in the appearance of facial features and results in a certain type of visual effect. The movements of facial points (eyebrows, eyes, and mouth) have a strong relation to the information about the shown facial expression. Therefore, many approaches greatly depend on the tracking of permanent facial features (eyebrows, eyes, mouth, and furrows that have become permanent with age) and/or transitional facial features (facial lines and furrows that are not present at a neutral state). In fact, the extraction of facial features is a very challenging task. Facial features cannot always be obtained reliably because of the quality of images, illumination, and some other disturbing factors. Furthermore, it usually takes a lot of computations to extract precise facial features.

3.1 Facial features localization

For facial features localization using the geometric face model, we have used the following stages as in Abdat et al. (2008):

1. Eyes axis location is determined by the maximum of the projection curve which has a high gradient. First we calculate the gradient of the image I:

$$\nabla I_x = \frac{\delta I}{\delta x}\vec{i} \tag{2}$$

   ∇I_x corresponds to the differences in the x (column) direction. The spacing between points in each direction is assumed to be one. Computing the absolute gradient value in each line given by:

$$HI_x(x) = \sum_{y=1}^{n} \nabla I_x(x,y) \tag{3}$$

   Then, we find the maximum value which corresponds to the line contains eyes. This line corresponds to many transitions: skin to sclera, sclera to iris, iris to pupil and the same thing for the other side (high gradient).
2. Median axis location is a vertical line which devises the frontal face in two equal sides. In other words, it is the line passed by the nose. To determine the median axis, we take the median of the bounding box of the face.

3. Mouth axis location is determined as the same way of eyes axis. For the localization of this axis, we look for the maximum value of the projection curve in the low part of the bounding box from eye axis.

Once the eyes and mouth axis are located, we use the geometric face model Shih & Chuang (2004) which suppose that:

- The vertical distance between two-eyes and the center of mouth is D.
- The vertical distance between two-eyes and the center of the nostrils is 0.6D.
- The width of the mouth is D.
- The width of nose is 0.8D.
- The vertical distance between eyes and eyebrows is 0.4D.

Figure 7 shows the results of the facial feature localization for a video sequence and for a real time acquisition. The eyes and the mouth are well located by rectangular windows.

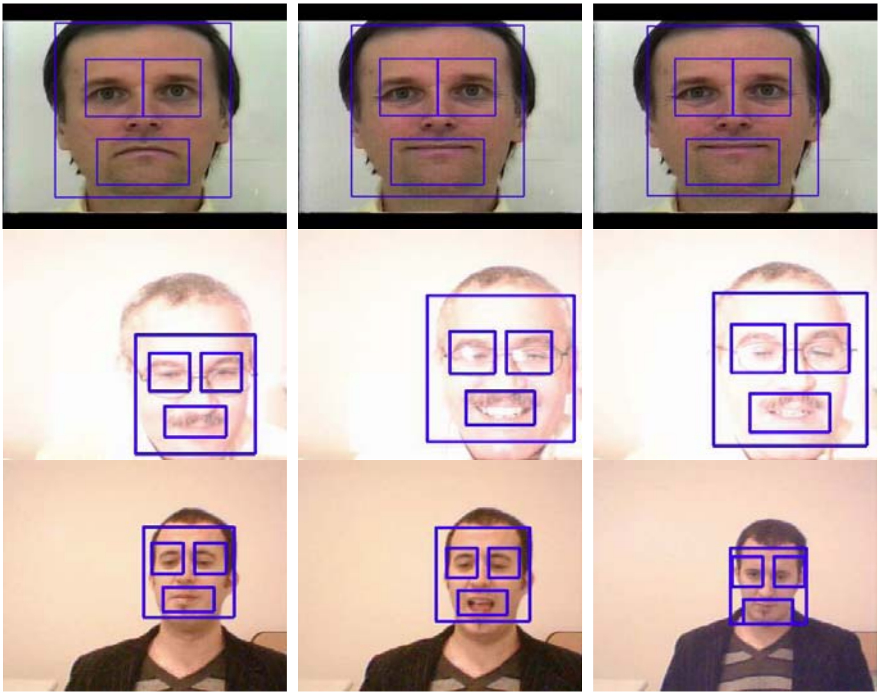


Fig. 7. Facial feature localization

3.2 Facial features points selection

The detected rectangles in the previous step do not give accurate information on facial features. To describe the movement of these features, we detected interest points. As first step, we used the uniform distribution, which consists on sampling the points of the rectangles in the directions of x and y with a step one-fifth $\frac{1}{5}$ of the rectangles size.

Figure 8 illustrates three refined rectangles, while the feature points are uniformly distributed in each rectangle. This selection of feature points is used in Shih & Chuang (2004). After this extraction step, the facial feature points will be tracked using an algorithm of optical flow which is pyramidal Lucas-Kanade tracker.

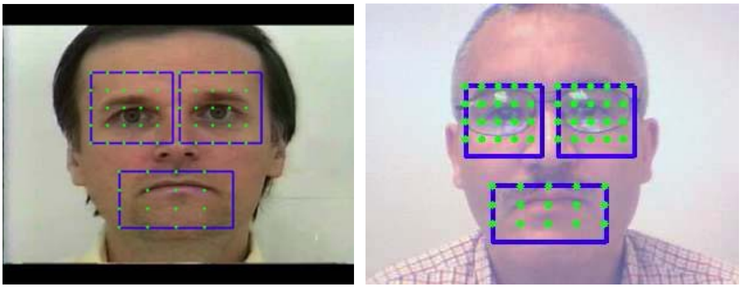


Fig. 8. Uniform distribution selection from the bounding box of facial feature.

4. Optical flow tracking

Optical flow is defined as an apparent motion of image brightness. Let *I*(*x*,*y*, *t*) be the image brightness that changes in time to provide an image sequence. Two main assumptions can be made Su & Hsieh (2007):

1. Brightness *I*(*x*,*y*, *t*) smoothly depends on coordinates *x*, *y* in greater part of the image.
2. Brightness of every point of a moving or static object does not change in time.

Let some object in the image, or some point of an object, move and after time *dt* the object displacement is (*dx*,*dy*). Using Taylor series for brightness *I*(*x*,*y*, *t*), we obtain:

$$I(x+dx,y+dy,t+dt) = I(x,y,t) + \frac{\delta I}{\delta x}dx + \frac{\delta I}{\delta y}dy + \frac{\delta I}{\delta t}dt + ... \tag{4}$$

where ". . ." are higher order terms.
Then, according to assumption 2:

$$I(x+dx,y+dy,t+dt) = I(x,y,t) \tag{5}$$

and

$$\frac{\delta I}{\delta x}dx + \frac{\delta I}{\delta y}dy + \frac{\delta I}{\delta t}dt + ... = 0 \tag{6}$$

Dividing equation 6 by *dt* gives:

$$\frac{\delta I}{\delta t} = \frac{\delta I}{\delta x}\frac{\delta x}{\delta t} + \frac{\delta I}{\delta y}\frac{\delta y}{\delta t} \tag{7}$$

Usually, equation 7 called optical flow constraint equation, where:

$$\tfrac{\delta x}{\delta t} = u \text{ et } \tfrac{\delta y}{\delta t} = v$$

are components of optical flow field $\vec{U}$ in *x* and *y* coordinates respectively.
Calculate optical flow returns to calculate for each point in the image the following equation:

$$\frac{\delta I}{\delta t} = u \cdot \frac{\delta I}{\delta x} + v \cdot \frac{\delta I}{\delta y} \tag{8}$$

However, the equation 8 can not determine with a single way the optical flow. The indetermination of optical flow due to the absence of global constraint in the precedent equations, only gradients which are local measures are taken into account. Lucas and Kanade have added new constraints to ensure the uniqueness of the solution. The method of Lucas and Kanade consists to find $\vec{U}$ applying a calculation of least squares to minimize constraint. They define a pre-neighborliness, and they optimize $\vec{U}$ in order to give a solution of the following system for *n* points:

$$\begin{bmatrix} \frac{\delta I}{\delta x}(p_1) & \frac{\delta I}{\delta y}(p_1) \\ \vdots & \vdots \\ \frac{\delta I}{\delta x}(p_i) & \frac{\delta I}{\delta y}(p_i) \\ \vdots & \vdots \\ \frac{\delta I}{\delta x}(p_n) & \frac{\delta I}{\delta y}(p_n) \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\frac{\delta I}{\delta t}(p_1) \\ \vdots \\ -\frac{\delta I}{\delta t}(p_i) \\ \vdots \\ -\frac{\delta I}{\delta t}(p_i) \end{bmatrix} \tag{9}$$

4.1 Discussion

After feature points extraction using the uniform distribution, we have used pyramidal Lucas-Kanade algorithm to track those points as shown in Figure 9. This algorithm has less computation. So, it is adapted for real time application. A motion, caused by a real moving-face, should be highly correlated in space and time domains. In other words, a moving-face in a video sequence should be seen as the conjunction of several smoothed and coherent observations over time.

Tracking a set of interest points is based on valuation techniques of movement between two consecutive images. To obtain a reliable tracking, it is important that these issues be discriminating in the image. For example, a point in the midst of a region of a uniform image may not be identified precisely because all the neighboring pixels are similar. Also, an interest point is normally a point which has a position in the image with strong bi-directional changes. The points tracking consist to identify a set of *N* interest points in order to model the interest region, and compute a location of each item according to calculations of optical flow.

Figure 9, shows an example of points tracking. These points are selected using the uniform distribution. It can be noted that from the second image, points began to disperse in arbitrary manner diverging from the correct position.

With the uniform distribution, we have got a bad results because these points haven't a strong bidirectional variation. In order to resolve this problem, we search for the strong points in the image, for this reason, we have look for good features to track of Shi & Tomasi (1994).

4.2 Good features to track of Shi and Thomasi:

In order to compare the obtained results using uniform distribution, we have used the method of Shi and Thomasi for interest points extraction. This method is based on the general assumption that the luminance intensity does not change for image acquisition.

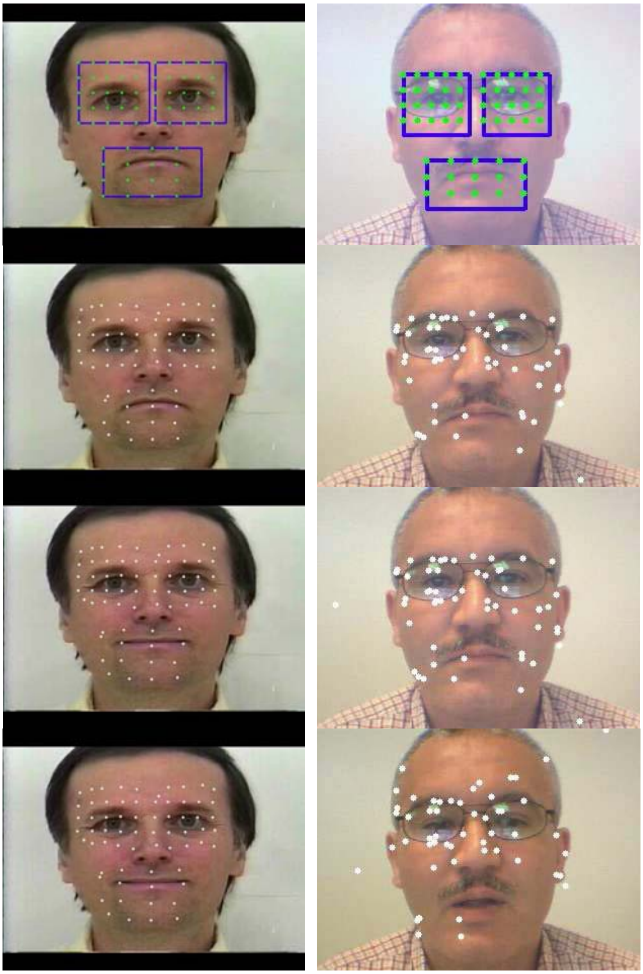


Fig. 9. Tracking of the uniform distribution for video sequence.

To select interest points, a neighbourhood *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

$$A = \begin{bmatrix} \sum D_{x_{i,j}}^2 & \sum D_{x_{i,j}} \sum D_{y_{i,j}} \\ \sum D_{x_{i,j}} \sum D_{y_{i,j}} & \sum D_{y_{i,j}}^2 \end{bmatrix} \tag{10}$$

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 eigenvalue 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. For further details see Shi & Tomasi (1994).

4.3 Detection of facial feature points using the Shi and Thomasi method:

Figure 10 shows the obtained results for feature points detection with the method of Shi and Thomasi (video sequence, real time acquisition) applied to the whole image. We can see a good tracking for these points in the remaining of the sequence, unlike the first method (uniform distribution), which prove that the Pyramidal Lucas-Kanade Feature Tracker need a strong points to be tracked.

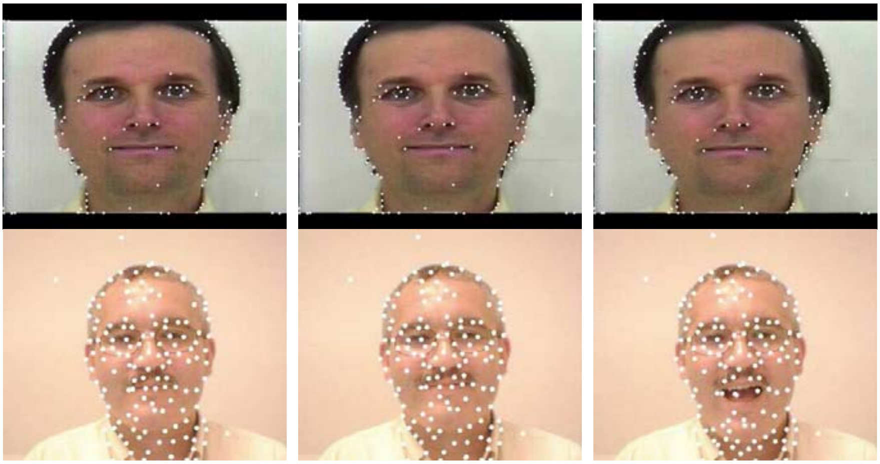


Fig. 10. Extraction of feature points in the first frame and feature points tracking using pyramidal Lucas-Kanade feature tracker in the remaining of the sequence.

The method of Shi and Thomasi ensures good detection of points that have strong gradient. This good detection leads to a good tracking of these points.

4.4 Detection of facial feature points in the bounding box:

In the previous section, we have presented a detection of interest points in the face, however, we need only the points which surround facial features such as eyes, eyebrows and mouth. For this reason, we will reject all the pixels outside the rectangle. Figure 11 shows interest region, which will be used for the detection of points with Shi and Thomasi method.

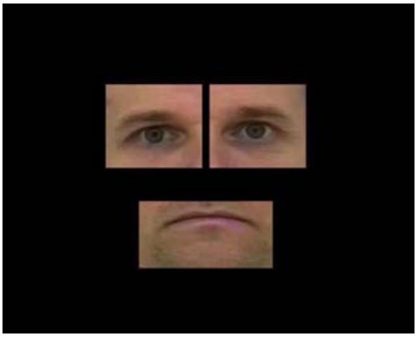
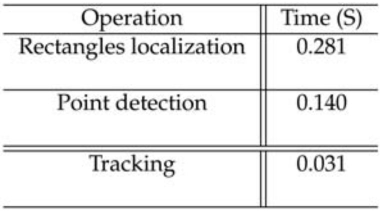


Fig. 11. The interest region feature point extraction in the first frame.

Figure 12 shows an example of points tracking in the bounding box. The tracking is very well; the first image presents the detection of points in the bounding boxes which delimit the facial features using Shi and Thomasi method. These detected points correspond to pixels with strong gradient. The following images present the 1st, 2nd, 22th and 46th frames in the first video sequence and the 1st, 2nd, 51th and 67th for the second sequence.
Our system is implemented in VC.NET on a pentium IV with $2GHz$ under windows XP. The table 1 presents the elapsed time for each step in our system. The size of the frame is $576 * 720$ and the video sequence format is $AVI - I420$.
For the first frame, the elapsed time for rectangle localization is $0.281S$ and the elapsed time for point detection is $1.40S$. For the remaining of the sequence, we can track the detected points for $0.031s$.



Operation	Time (S)
Rectangles localization	0.281
Point detection	0.140
Tracking	0.031

Table 1. Elapsed time for each step.

5. Conclusion and future works

In this paper, we have presented a face tracking algorithm in real time camera input environment, in order to use it in the facial expression recognition system. To detect the face in the image, we have used a modified face detector based on the Haar-like features. This face detector is fast and robust to illumination condition but hardly work when face pose is too slanted. For feature points extraction, we have used the algorithm of Shi and Thomasi to extract feature points. This method gives good results. To track the facial feature points, Pyramidal Lucas-Kanade Feature Tracker KLT algorithm is used. We have got a bad results with a uniform distribution of feature points which explain that this algorithm need a strong points. However, using detected points with the algorithm of Shi and Thomasi, we have got good results in video sequence and in real time acquisition. The obtained results indicate that the proposed algorithmcan 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.

6. References

Abdat, F., Maaoui, C. & Pruski, A. (2008). Real facial feature points tracking with pyramidal lucas-kanade algorithm, *IEEE RO-MAN08, The 17th International Symposium on Robot and Human Interactive Communication, Germany*.

Bouguet, J. (2000). Pyramidal implementation of the lucas kanade feature tracker, *Intel Corporation* Microprocessor Research Labs.

H. Xiaolei, Z. Song, W. Y. M. D.-S. D. (2004). A hierarchical framework for high resolution facial expression tracking, *3rd IEEE Workshop on articulated and non rigid motion ANM 2004*.

K. Ki-Sang, J. Dae-Sik, C. H.-I. (2007). Real time face tracking with pyramidal lucas-kanade feature tracker, *Computational science and its applications ICCSA 2007* 4705: 1074–1082.

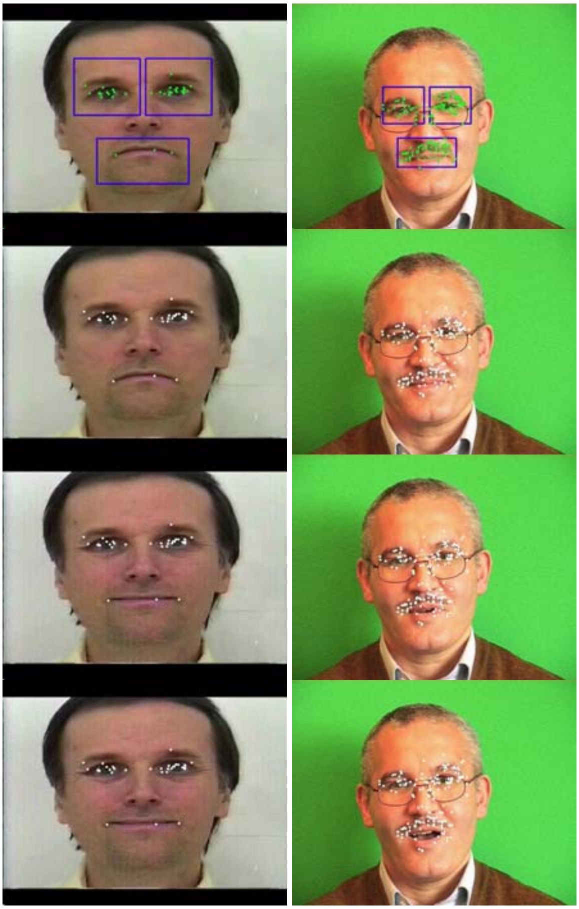


Fig. 12. Extraction of feature points in the bounding box for the first frame and Feature points tracking using pyramidal Lucas-Kanade in the remaining of the sequence.

P.Viola & M.Jones (2001). Rapid object detection using a boosted cascade of simple features, *Conference on CVPR 2001*.

R.Belaroussi & Milgram, M. (2006). Face tracking and facial features detection with a webcam, *CVMP 2006*.

Shi, J. & Tomasi, C. (1994). Good features to track, *IEEE Conf. Computer Vision and Pattern Recognition Seattle* CVPR'94.

Shih, F. & Chuang, C. (2004). Automatic extraction of head and face boundaries and facial features, *Information Sciences* 158: 117–130.

Su, M. & Hsieh, Y. (2007). A simple approach to facial expression recognition, *Proceeding WSEAS 2007 Australia*.

Viola, P. & Jones,M. (2001). Robust real-time object detection, 2nd international workshop on statistical and computational theories of vision - modeling, learning, computing, and sampling Vancouver, Canada.