Multi-Cue-Based Face and Facial Feature Detection on Video Segments

PENG ZhenYun (彭振云), AI HaiZhou (艾海舟), Hong Wei (洪 微), LIANG LuHong (梁路宏) and XU GuangYou (徐光佑)

Department of Computer Science and Technology, Tsinghua University, Beijing 100084, P.R. China

E-mail: cpeng@eastbird.com; ahz@mail.tsinghua.edu.cn

Received March 16, 2001; revised April 12, 2002.

Abstract An approach is presented to detect faces and facial features on a video segment based on multi-cues, including gray-level distribution, color, motion, templates, algebraic features and so on. Faces are first detected across the frames by using color segmentation, template matching and artificial neural network. A PCA-based (Principal Component Analysis) feature detector for still images is then used to detect facial features on each single frame until the resulting features of three adjacent frames, named as base frames, are consistent with each other. The features of frames neighboring the base frames are first detected by the still-image feature detector, then verified and corrected according to the smoothness constraint and the planar surface motion constraint. Experiments have been performed on video segments captured under different environments, and the presented method is proved to be robust and accurate over variable poses, ages and illumination conditions.

Keywords multi-cue, facial feature detection, face detection, video segment, motion constraint

1 Introduction

Locating face regions and facial features on video segments plays a very important role in the applications of content-based video retrieval and video compression. However, most of the existing facial feature detection methods are performed on single-frame images and are not suitable to video segments. Besides, because one single frame cannot provide enough information about changes of poses, lighting conditions, background and expressions, these feature detectors may miss features or extract false ones. The key issue is that once the features are detected, no enough evidence can be taken from one single image to verify the results.

The presented approach aims to robustly detect face regions and facial features on video segments. To ensure the detection to be accurate and robust, a cross-verification mechanism is established based on multi-cues, including gray-level distribution, color, motion, templates, algebraic features and so on.

Faces are first detected by multi-means including skin-color-based segmentation, template matching and artificial neural network (ANN). On the resulted face regions, a PCA-based feature detector for still images is then used to detect facial features on each single frame until the resulted features of three adjacent frames do not change significantly. These three frames are defined as base frames whose features are considered to be correct. Given a base frame, the features of its neighbor frames are first estimated according to the known features, then detected by the still-image feature detector, finally verified and corrected by using the smoothness constraint and the planar surface motion constraint. Experiments have been performed on video segments captured under different environments. It is proved that the presented method is robust and efficient over variable poses, ages and illumination conditions.

2 Face Detection

As presented in [6], the face detection algorithm consists of two parts (see Fig.1): color segmentation for color images and template-based face detection for gray-level images.

2.1 Skin Color Segmentation

In skin color segmentation, a qualitative 3D mo-
del in HSI color space is used, in which hue (H) and saturation (S) are two chromatic coordinates with intensity (I) as the third dimension. In practice, I, S and H are quantized into 16, 256 and 256 levels respectively. Further for the sake of speed a lookup table of $16 \times 256 \times 256$ is established based on the results from training chromatic face samples where 1 and 0 indicate skin and non-skin respectively. After skin color classification, pixels of skin color will be grouped according to color uniformity and pixel connectivity, and then a rectangular region bounding the merged area will be ascertained according to color and position nearness and heuristic knowledge about face shape and size. The rectangular box bounding the face region will provide a good estimation of the face size and then save the computation time during the follow-up template-based face detection.

2.2 Template-Based Detector

As illustrated in Fig.1, skin color segmentation or the gray-level image template matching is performed to detect face. Two types of templates, eye-pair and face itself, are used one by one in searching for candidate faces. To verify the results from the initial face detection two multi-layer perceptrons (MLPs) are used independently so as to exclude most of the false alarms.

2.2.1 Template Generation

A $20 \times 20$ average face is derived from a set of mug-shot photographs containing frontal upright faces, of which each face is manually cut out and then is transformed to a normalized scale and a gray distribution with identical average and variance. To emphasize the importance of the eye feature, the region of two eyes of the average face is cut out as the eye-pair template (see Fig.2).

To detect faces in different poses rotating in image plane, six groups of templates corresponding to different rotation angles are generated via affine transform. They account for a pose variation from $-30^\circ$ to $30^\circ$ with a $10^\circ$ step that covers most of face poses contained in normal images.

2.2.2 Neural Network Classifier

In the matching procedure two MLPs are used. Each MLP contains 374 input nodes corresponding to a $20 \times 20$ searched window (several corner points are ignored). One MLP has 20 hidden nodes and the other has 24 hidden nodes, which are completely linked to all input nodes and only one output node. The output node has a value in $[0,1]$, where 1 represents face and 0 non-face. In these MLPs Sigmoid nonlinear function is used.

The MLPs are trained independently by BP algorithm with the same face samples (about 5,000 face samples transformed from about 700 hand-cut-out faces by slightly rotations and reflection) and different non-face samples collected by two independent modified bootstrap procedures in which the initial non-face samples are false alarms generated by a pure template matching-based detector. Finally those two MLPs use about 5,000 non-face samples each.

2.2.3 Matching and Arbtrating Strategy

The matching algorithm for gray-level image is composed of the following procedures:

1. Given an image, initialize a face candidate list.
2. At each point of the image, firstly the corresponding searched window is normalized in gray distribution, then eye-pair templates of various poses are matched via the correlation coefficient, of which top 3 matches over a predetermined eye-pair threshold are further matched with corresponding face templates. Then the maximum one from those over a predetermined threshold for the face matching will be selected.
3. De-rotate the selected region as a candidate, then feed it into the MLP classifier. If the output exceeds a
threshold, compare the candidate with those in the face
candidate list, if there is no overlapping region found,
put it in the list, otherwise the bigger one will replace
the smaller one.
4. Sub-sampling the image by a ratio of 1.2, repeat
from Step 2, until a specified size is reached.

In practice, two MLPs are used in Step 3 to
generate two different face candidate lists. Since
these MLPs are trained using the same face sam-
ple and different non-face samples, they have sim-
ilar response to face regions and significantly differ-
ent responses to non-face regions. Therefore, an ar-
britiation procedure between those two lists is used
to exclude most of the false alarms after searching.

3 PCA-Based Facial Feature Detection on
Still Images

3.1 Representation of Eyes in Eigen-Eye
Space

As stated in [7], eyes in the training set and
the target image are first calibrated using the ho-
logenous transform so that all eye regions under
consideration have identical position and size.

Then eigen vectors with a total of \( l \)
\((u_1, u_2, \ldots, u_l)\) are calculated from the training set
\( \{i_1, i_2, \ldots, i_m\} \) based on the Singular Vector De-
composition (SVD) theorem.

Let \( p \in \mathbb{R}^n \) be the normalized input eye pair of
size \( w \times h \). It can be projected onto the eigen-eye
space as:

\[
p = \sum_{i=1}^{l} c_i u_i = U(c_1, c_2, \ldots, c_l)^T
\]

Because \( U \) is orthogonal, we have,

\[
(c_1, c_2, \ldots, c_l)^T = U^T p
\]

Therefore, \( p \) is mapped to the eigen-eye space as
\( p' = \sum_{i=1}^{l} c_i u_i \). The representation error is mea-
sured by the correlation between \( p \) and \( p' \):

\[
\delta(p, p') = \frac{E(pp') - E(p)E(p')}{\sigma(p)\sigma(p')}
\]

3.2 Eyes Detection

By using the Hough-transform-based eyes de-
tection algorithm we presented in [8] that \( k \) can-
date irises \( C_1, C_2, \ldots, C_k \) can be first found in the
AOI (Area of Interest) under consideration. Let
\( C_1, C_2, \ldots, C_k \) be the nodes of a complete graph \( G \).

A benefit function \( BF(i, j) \) for the edge between \( C_i \)
and \( C_j \) is defined as follows:

\[
B(i, j) = \left( k_1 \delta(p_{ij}, p'_{ij}) + k_2 \gamma(p_{ij}, p'_{ij}) \right) * D(i, j) * A(i, j)
\]

where \( k_1, k_2 \in [0, 1], \ k_1 + k_2 = 1 \); \( \gamma(p_{ij}, p'_{ij}) \) is
the similarity and symmetry measurement of \( C_i 
\) and \( C_j \); \( p_{ij} \) is the eye area with irises being \( C_i 
\) and \( C_j \), and \( p'_{ij} \) is the projection of \( p_{ij} \) on the eigen-
eye space. \( \delta(p_{ij}, p'_{ij}) \) is defined by (3); \( DC(i, j) \)
and \( AC(i, j) \) are the distance and tilt constraints
to eye pairs respectively.

The iris pair \( (C_i, C_r) \) is accepted as the correct
eye pair if the following condition holds.

\[
BF(i, r) = \max_{i, j=1, 2, \ldots, k} BF(i, j) \geq \delta_0
\]

where \( \delta_0 \) is the threshold of the eye fidelity. If \( \max_{i, j=1, 2, \ldots, k} BF(i, j) \geq \delta_0 \), the region under
consideration will be considered as a non-face region.

3.3 Mouth Detection

First, the mouth region can be roughly bounded
to the eye positions according to the anthropo-
metric measurements. As shown in Fig.3, suppose the
two irises to be \( C_i \) and \( C_r \), then the mouth is likely
to be within the parallelogram \( ABCD \).

\[\text{Fig.3. Binding the mouth.}\]

The integral projection method is very useful for locating the mouth corners. In our case, the
horizontal integral projection of \( I(x, y) \) in \( ABCD \)
is defined as

\[
H(y) = \sum_{x} I(x, y)
\]

Similarly, the vertical integral projection is defined
as
\[ V(x) = \sum_{y=AB(x)}^{DC(x)} I(x, y) \] (7)

Here \( y = AB(x) \) and \( y = DC(x) \) are the linear equations of lines \( AB \) and \( DC \) respectively, and \( y = BC(x) \) and \( x = AD(y) \) are those of lines \( BC \) and \( AD \). While \( H(y) \) is calculated on the source image, \( V(x) \) is obtained from the combination of the vertical gradient map and the source image.

The vertical position of the mouth is obtained by first looking for the local valleys of \( H(y) \) and then choosing the applausive one according to the intensity distributions around them. Say the obtained horizontal line is \( L \), then \( V(x) \) can be calculated on \( A_1B_1C_1D_1 \). By finding the two ramps of \( V(x) \), we can estimate the horizontal positions of the two mouth corners.

### 3.4 Nose Detection

The two nares are found using the method similar to mouth detection:

1) Nose region is first estimated according to the mouth region (see Fig.4);

![Fig.4. Binding the nose.](image)

2) The base line of the nose \( y = y_n \) is obtained by using the horizontal integral projection again;

3) The nares \( N_1(x_{n_1}, y_n) \) and \( N_3(x_{n_3}, y_n) \) are two points on the base line \( y = y_n \) which meet the following conditions:

\[ S(x_{n_1}) = \min_{x \in [x_3, x_m]} S(x) \] (8)

\[ S(x_{n_3}) = \min_{x \in [x_m, x_4]} S(x) \] (9)

where,

\[ S(x) = \sum_{(x,y) \in \text{Circle}(x,y_n,r_n)} I'(x, y) \] (10)

### 4 Facial Feature Detection on Video Segments

#### 4.1 Overview of the Cross-Verification Mechanism

The facial feature detection algorithm for still images can be used for detecting faces in image frames of video segment as well. However, by taking the advantage of the correlation between the video frames, the reliability and the accuracy can be improved further. In this case, the facial features detected by the algorithm for still images will provide the initial estimation, then the following algorithm will take over:

1) Find three adjacent frames such that the features detected using the above method do not change significantly over these 3 frames, which are named as base frames. The features of these frames are considered as correct and are called as base features.

2) Given a base frame and the corresponding base features, features of its neighbor frames are detected in the following way:

   (a) the features are first estimated on the target frame;

   (b) detection is only performed within the estimated regions;

   (c) the resulting features are verified using the motion smoothness constraint;

   (d) if the resulting features do not comply with the smoothness constraint, new features are estimated using the planar surface motion constraint.

#### 4.2 Motion Smoothness Constraint

Because the motion of the head can usually be ignored with respect to the distance between the head and the camera, distances among features (Fig.5) in two adjacent frames are constrained in a small range.

![Fig.5. Distance between features.](image)
4.3 Planar Surface Motion Constraint

If the resulting features do not follow the smoothness constraint, new features have to be estimated using the planar surface motion constraint. Even if the resulting features follow the smoothness constraint, these features might not be accurate, then we also need to use the planar surface motion constraint to locate the features accurately.

The head is a rigid object and the head motion is generally rigid motion as far as there is no facial expression. Under the perspective projection, the feature points satisfy a parametric 2-D motion model. Suppose a feature point in the base frame is \( x = (x_1, x_2) \), the corresponding feature point in the target frame is \( x_0 = (x_{01}, x_{02}) \), and the estimated one in the target frame is \( x' = (x'_1, x'_2) \), then we have:

\[
\begin{align*}
  x'_1 &= r_{11}x_1 + r_{12}x_2 + r_{13} + T_1/X_3 \\
  x'_2 &= r_{31}x_1 + r_{32}x_2 + r_{33} + T_3/X_3
\end{align*}
\]

(11)

\[
\begin{align*}
  x'_2 &= \frac{a_4x_1 + a_5x_2 + a_3}{a_7x_1 + a_6x_2 + 1} \\
  x'_1 &= \frac{a_1x_1 + a_2x_2 + a_3}{a_7x_1 + a_6x_2 + 1}
\end{align*}
\]

(12)

Because the six feature points (2 irises, 2 nares and 2 mouth corners) can be considered to be lying on a planar surface, the mapping from the base frame to the target frame can be described as:

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  1 \\
  0 \\
  0 \\
  0
\end{bmatrix}
= \begin{bmatrix}
  x'_1 \\
  x'_2
\end{bmatrix}
\]

(15)

The coefficients \( a_1, \ldots, a_8 \) can be estimated by using the following equation:

\[
\begin{bmatrix}
  x_1 & x_2 & 1 & 0 & 0 & 0 & -x_1x'_1 - x_2x'_2 & A
\end{bmatrix}
= \begin{bmatrix}
  0 & 0 & 0 & x_1 & x_2 & 1 & -x_1x'_1 - x_2x'_2
\end{bmatrix}
\]

where

\[
A = [a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6 \ a_7 \ a_8]^T
\]

(16)

To solve (15), there must be at least 4 correspondent points. In the face feature case, there are 6 correspondent points between the base frame and the target frame. To select 4 correspondent points from 6 candidates, we have \( C_6^4 = 15 \) different combinations. In order to obtain the optimized combination, the vector \( A \) is calculated under each combination of 4 correspondent points, resulting 15 vectors, \( A_1, \ldots, A_{15} \).

For each of the 15 vectors, all the 6 feature points \( x' \) can be estimated from the base frame using (11) and (12). The optimized coefficient \( A_{opt} \) is obtained according to the following rule:

\[
A_{opt} = \{A | \text{min (Err}(A_i)\} \quad i = 1, \ldots, 15
\]

(17)

Here \( Err(A_i) \) is the error of this estimation, which is defined as:

\[
Err(A_i) = \max(|x^{ij}(A_i) - x^{ij}_0|) \quad j = 1, \ldots, 6
\]

(18)

After \( A_{opt} \) is obtained, more accurate feature points can be estimated by using the parametric 2-D motion model. In the example shown in Fig.6, the nares are initially located at the positions marked by the crosses, and then corrected to more accurate locations (indicated by diamonds) by using the planar surface motion constraint.

![Fig.6. An example of the estimated features.](image)

5 Experimental Results

30 video segments are tested for evaluating the performance of the presented algorithm. Each sequence contains 50 frames. The training set includes 80 sample images. By using the Principal Component Analysis Method, 55 eigen eye pairs are retained as the base vectors of the eigen-eye space. All tested images are not included in the training set. The test platform is a PIII/933 MHz computer with 256MB RAM under Windows 2000. The experimental results are summarized (see Table 1) as: (1) the average correct ratio of irises detection is 98.00%; (2) the average correct ratio of nares detection is 98.00%, (3) the average correct ratio of mouth corners detection is 98.49%, and (4) the average overall correct ratio is 96.36%. The average computation time is 10s/image. Fig.7 shows part of the experimental results of a video segment.

6 Conclusion

A proposal is presented to detect faces and facial
features on video segments by making use of multici cues and the inter-frame cross-verification mechanisms. Experiments show that the approach is relatively efficient in both accuracy and speed. Compared with existing methods, the key contribution of the proposal is that features are not only detected but also verified and estimated based on multi-cues, which are independent of each other.

References