Real-time elliptical head contour detection under arbitrary pose and wide distance range

Wei Zou *, Yuan Li, Kui Yuan, De Xu

Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 11 July 2008
Accepted 21 January 2009
Available online 5 February 2009

Keywords:
Elliptical head contour
Quadrant arc
Real time
Arbitrary head pose
Wide distance range
Gradient
Ellipse fitting
Robust

ABSTRACT

In this paper, we propose a real time elliptical head contour detection method based on quadrant arcs, which is efficient and robust to arbitrary head pose and wide distance range. First, the moving object area is detected according to background model which is built on three color channels. Then, all the valid elliptical arcs are extracted out from connected edges, and classified into four kinds of quadrant arc sets according to their gradient information. Finally, the arcs lying in different sets are combined to fit out the elliptical head contour based on the least square method. Experimental results confirm the robustness and the accuracy of this method under arbitrary head pose and wider distance range, as well as the real time property, strong robustness to long or short hair and with or without hat.

1. Introduction

Automatic detection of human head is an active research area in machine vision and a major step in many applications such as intelligent man machine interface, video surveillance, driver behavior analysis, robotic human tracking, human behavior analysis, and so on. Head detection can be regarded as an extension of face detection, which has been researched for decades and some relevant fruits have been utilized in face recognition, human identification/identity verification, security control, and so on [1,2]. Purpose of this paper is to construct a human head contour detection mechanism for triggering the covertly human-tracking process of mobile robot, namely, the robot controls itself to run into the human's blind field according to his/her line of sight during tracking. In this application, head detection is more difficult than face detection because of following reasons: (1) head may be under arbitrary pose, so the detection method must be able to extract invariable features to this. Most information provided implicitly or explicitly by eyes, nose, mouth, and so on, which play important role in face detection, will not suit to the head detection any more when human backs to camera; (2) in most applications of face detection, in order to achieve better effectiveness, the distance between human and camera can be subjectively adjusted to a certain value according to the requirement of the detection method. While for the head detection in our application, the method should be adaptive to a wide range of distance; (3) this kind of application has a high requirement to real time property, so some complex algorithm with large computational quantity is not suitable any more.

Common methods of head detection mainly include those based on color model, based on template matching and based on contour detection. Methods based on color model [3–6] detect out pixels or regions belonging to head area by using skin and hair color model. This category of method has the advantages of simplicity and good real time property, but they usually demand face's appearing in the image entirely or partially. When human backs to camera, the hair color tends to be confused with other objects under complex ground, which results into the unfeasibility of this method in our application. Methods based on template matching [7–10] first build the head's template model off-line manually or automatically, which may be fixed or deformable. Then based on these templates, the head area can be searched out by some similarity measurement and dynamic warping technologies. This kind of method is robust to complexity background and to arbitrary head pose, but the searching process is time-consuming. Methods based on contour detection [4,11–13] approximate the head contour into some geometric curves such as ellipse, gauss curve and so on. Parameters describing these curves can be obtained by curve fitting technologies according to the gradient information on head edges. This category of methods can achieve the head contour's analytic solution, which results into its popularity in head detection. Among all the applicable curves, for the sake of the most similarity to head shape and in order to achieve the closure area of
head, ellipse is utilized most widely. In this paper, we try to detect out the complete elliptical head contour by some fast algorithm.

Ellipse detection is an important task in computer vision. Least square method has been commonly employed for this purpose. A major disadvantage of this method is that it is susceptible to the influences of outliers, noise and extraneous pixels. In order to overcome this problem, the standard Hough transform (HT) and its variants, such as combinatorial Hough transform (CHT) [14,15], the randomized Hough transform (RHT) [16], the probabilistic Hough transform (PHT) [17], the dynamic generalized Hough transform (DGHT) [18], the iterative randomized Hough transform (IRHT) [19], the random sample consensus algorithm (RANSAC) [20] and so on, try to seek the estimation of ellipse parameters by voting and clustering in parameter spaces. This family of methods can detect ellipse under various situations; however, for the high dimension (5-D) of parameter space, they are usually inefficient for the practical applications with high requirement to real time property.

In order to speed up the process, some geometric properties of ellipses are combined into HT. In [21], 5-D RHT is reduced into 3-D RHT by polar and pole definition of conics and by application of gradient direction information on edge points. Based on the property that the line across the pole and the midpoint of polar chord must pass through the ellipse center, [22] applies 2-D RHT to achieve the ellipse center. Ref. [23] integrates edge directional properties to decompose the HT parameters space and make a full use of candidate point pairs to achieve computational efficiency. Although these methods improve efficiency to some extent, it is still not useful in many real-time applications, e.g., our robotic human tracking and video surveillance systems. In [24], an ellipse verification algorithm based on valid elliptical arcs is introduced. In this algorithm, the endpoints and subtended angles of elliptical arcs are detected based on pixel connectivity, and iterative fitting is executed to fit out valid ellipse. This method is more efficient than those based HT technologies, but the speed is still too low to be applied in our application. Another drawback of this method is its weak robustness to discretization errors when only local arc segments on the ellipse are utilized, which maybe always occurs in our application.

In this paper, we propose a real time elliptical head contour detection method based on quadrant arcs, which is efficient for arbitrary head pose and wide distance range. All the valid elliptical arcs are determined and classified into four kinds of quadrant arc sets according to their gradient information, and arcs lying in different sets are combined to fit out the elliptical head contour based on the least square method. Although the introduction to this method mainly focuses on static background in this paper, the idea can be implanted into dynamic scenes by simple modification and extension.

The paper is organized as follows. Section 2 outlines the framework of the elliptical head contour detection method proposed in this paper. The moving object area detection technology based on background model of three color channels is presented in Section 3. In Section 4, the definition of quadrant arc, how to extract out different quadrant arcs and how to combine quadrant arcs to construct the fit points set are introduced. Section 5 simply describes the ellipse fitting technology based on least square method. Experimental results of detection and simple extension to tracking are presented and compared to other methods in Section 6, and concluding remarks in Section 7.

2. Framework

The framework and processing flow of the elliptical head contour detection method proposed in this paper are shown in Fig. 1. The main steps of this method include:

1. Edge detection. In this step, edges in the image are detected out and described in single pixel width by Canny edge detector.

2. Connected edge search. All the connected edge pixels are searched out from the edge image, and arranged together in sequence according to their adjacency relation, which will result into several order pixel sets called connected edges.

3. Quadrant arcs determination and abstraction. In this step, each connected edge is split into line segments or arcs. According to its gradient information and shape, each arc is distributed into four class arc sets: 1st quadrant arcs, 2nd quadrant arcs, 3rd quadrant arcs or 4th quadrant arcs.

4. Quadrant arcs combination. One arc is selected out from each quadrant arc set according to some specified rules. All the pixels on the selected arcs constitute the candidate fitted points set, which is utilized to fit out ellipse.

5. Fitting ellipse. The ellipse and its parameters which have minimum errors for the candidate fitted points are fitted out by least square method.

6. Head Contour verification. According to human head’s characteristics, the most potential ellipse is determined as the elliptical head contour.

For static background situations such as video surveillance or the initial stage of robot human tracking, multiple video frames can be grabbed to establish the background model, and the moving object area can be checked out from the original image based on this background model. Processing on the sub-image corresponding to the moving object area, the above elliptical head contour detection steps can become so fast that the real-time requirement can be met to a large extent. As for the problem of robot–human tracking, once the head contour is detected out under static background, for the sake of the movement of robot, the background model can not be applicable any more, so the motion estimation and prediction technologies can be utilized to produce the moving object area, just as the dot line components shown in Fig. 1. This
paper mainly focuses on the introduction of the elliptical head contour detection under static background, which is shown in Fig. 1 as actual line.

3. Background model and moving object area detection

Suppose there are K frame static background images without containing any moving object: \( I_n(x, y) = [R_n(x, y), G_n(x, y), B_n(x, y)]^T, \ 1 \leq j \leq K, \) where \( R, G, \) and \( B \) are three primary colors of each pixel; \( (x, y) \in N^2, \ 0 \leq x < W, \ 0 \leq y < H \) is the pixels coordinates in image planar coordinate system; and \( W \) and \( H \) are the width and height of the image in pixel, respectively. In order to improve the robustness to illumination, the three primary colors are normalized firstly:

\[
\begin{bmatrix}
    R_n(x, y) \\
    G_n(x, y) \\
    B_n(x, y)
\end{bmatrix} = \frac{1}{R_n(x, y) + G_n(x, y) + B_n(x, y)} \begin{bmatrix}
    R_n(x, y) \\
    G_n(x, y) \\
    B_n(x, y)
\end{bmatrix}
\]

(1)

Then, for each normalized color channel \( t \in \{r, g, b\} \), compute the mean value \( \mu_t(x, y) \) and variance \( \sigma_t^2(x, y) \) statistically based on all the \( K \) frame static background images:

\[
\begin{align*}
    \mu_t(x, y) &= \frac{1}{K} \sum_{j=1}^{K} t_n(x, y) \\
    \sigma_t^2(x, y) &= \frac{1}{K} \sum_{j=1}^{K} [t_n(x, y) - \mu_t(x, y)]^2
\end{align*}
\]

(2)

Suppose an arbitrary pixel's normalized color vector is \( [r(x, y), g(x, y), b(x, y)]^T \), then the background model which is used to measure the pixel's possibility belonging to the background can be described as a probability vector \( P_n(x, y) \) based on the above parameters and Gaussian function:

\[
P_n(x, y) = [p_{rn}(x, y), p_{gn}(x, y), p_{bn}(x, y)]^T
\]

(3)

where

\[
p_{tn}(x, y) = \frac{1}{\sqrt{2\pi} \sigma_t} \exp \left( -\frac{(t(x, y) - \mu_t(x, y))^2}{2\sigma_t^2} \right), \ t \in \{r, g, b\}
\]

(4)

In the stage of on-line head contour detection, every pixel in original image is normalized by Eq. (1) and its memberships to background are estimated by model (3) and (4). For determination the pixel's attribution to background or foreground, the basic idea of D-S evidence theory is applied to fuse out the final probability: considering background and foreground as frame of discriminate, and the three components of vector \( P_n(x, y) \) as basic probability assignments for the singleton set of background coming from three evidences, according to Dempster combination rule, the final probability can be determined as:

\[
P_{bg}(x, y) = \frac{P_{rb}(x, y)P_{gb}(x, y)P_{bb}(x, y)}{P_{rb}(x, y)P_{gb}(x, y)P_{bb}(x, y) + (1 - P_{rb}(x, y))(1 - P_{gb}(x, y))(1 - P_{bb}(x, y))}
\]

(5)

Comparing to minimum and multiply operators, the above fusion method has the merit of stronger robust property to noise and illumination by considering three evidences comprehensively, as well as the result's tendency to bipolar of the interval \([0, 1]\), which facilitates the binarization of the origin image. For every pixel in the image, if its \( p_{bg} \) is greater than some threshold \( T_{bg} \), consider it as a point in background and set its value 0, else a foreground point and its value 1. After the above thresholding process and denoised by morphological methods, a binary image corresponding to the origin image can be obtained, which can be utilized to check out the moving object by searching the largest connected area of pixels 1. Because the main purpose of this step is to extract out area completely including the moving object without special requirement for accuracy, a resample of the origin image with ratio \( 3 \times 3 \) is used for computation reduction and enhancement of real-time property. An example of moving object area detection is shown in Fig. 2, where the threshold value \( T_{bg} = 0.3 \).

4. Quadrant arcs determination and combination

4.1. Connected edge search

Suppose the sub-image in origin image corresponding to the object area is \( I_{obj}(x, y) \), \( 0 \leq x < W_{obj} \), \( 0 \leq y < H_{obj} \), where \( W_{obj} \) and \( H_{obj} \) are width and height of the sub-image, respectively. Apply Canny operator on it and obtain the single-pixel-width edge image \( EdgeM \), in which \( EdgeM(x, y) = 1 \) for edge points and \( EdgeM(x, y) = 0 \) for others. Then, the connected edges can be searched out by following steps:

1. Clear global buffer \( LineBufGlobal \) and local buffer \( LineBufLocal \); define MaskLabel as a label array with same dimensions as \( I_{obj} \), and set all of its elements 0.
2. Scan each pixel of \( I_{obj} \) in turn from top to bottom and from left to right. If current pixel \((i, j)\) satisfies \((EdgeM(i, j) = 1) \land (MaskLabel(i, j) = 0)\), add it into \( LineBufLocal \) and set MaskLabel\((i, j) = 1\), goto (3).
3. Suppose the last and the second last pixels in \( LineBufLocal \) are \((i_s, j_s)\) and \((i_s, j_s)\), respectively. Starting from \((i_s, j_s)\), scan the 8-neighbor pixels of \((i_s, j_s)\) clockwise. If current pixel \((i_s, j_s)\) satisfies \((EdgeM(i_s, j_s) = 1) \land (MaskLabel(i_s, j_s) = 0)\), then append it at the end of \( LineBufLocal \), set MaskLabel\((i_s, j_s) = 1\), stop scanning and continue (3). If all the 8-neighbor pixels can't meet the above condition, goto (4).
4. Continue searching in a similar way to step (3), starting from the first pixel of \( LineBufLocal \) and in counterclockwise. The pixels searched out are inserted into the buffer as the first pixel sequentially.
5. If there are more than 10 pixels in \( LineBufLocal \), add the corresponding connected edge into \( LineBufGlobal \), clear \( LineBufLocal \) and goto (2).

After scanning all the pixels in sub-image \( I_{obj} \), the connected edges with single-pixel-width and length greater than certain threshold (10 pixels in this paper) are saved in \( LineBufGlobal \). A main merit of this searching process is that it only needs scanning the image one time for extracting all the connected edges, which results in its higher efficiency than other methods, such as the classical two-step labeling algorithm, etc.

4.2. Definition and properties of quadrant arcs

For the sake of background's complexity, variation of illumination, arbitrary head pose and so on, the head contour may be split into several arcs and contained in different connected edges which are saved in \( LineBufGlobal \). In order to obtain effective points set for head elliptical contour fitting, it is necessary to extract out valid elliptical arcs from all these connected edges. At the same time, the distribution of fit points or arcs on ellipse may play more important role in the process of ellipse fitting for the sake of discretization errors. An example about this is shown in Fig. 3. Points on the bold arc in Fig. 3a and five points marked “*” in Fig. 3b are respectively used to fit out the objective ellipse shown in Fig. 3a. It can be seen in this figure that: although much more points are utilized in Fig. 3a than in Fig. 3b, due to the fact that all of these points are distributed on a local arc of the objective ellipse, and for the
In this paper, we present an ellipse fitting method which is based on quadrant arcs. By combining different quadrant arcs, the above principle rules for fit point selection can be concluded as:

1. select fit points which lying on different part of the ellipse preferentially;
2. select more fit points for fitting accuracy.

According to Theorem 1, 1st quadrant arc and 3rd quadrant arc can be discriminated from 2nd quadrant arc and 4th quadrant arc, but the gradient direction angle difference between 1st quadrant arc and 3rd quadrant arc, and the gradient direction angle difference between 2nd quadrant arc and 4th quadrant arc are not embodied for the sake of the uncertainty of gray variation in different directions.

Theorem 2. Suppose \( P_4P_6 \) is an arbitrary arc on an ellipse with two endpoints \( P_6 \) and \( P_4 \), is an arbitrary point on this arc. \( \theta(P) \in [-180, 180] \) is the angle between X axis and the arc’s normal line on \( P \), then if \( \theta(P) \) decreases or increases continuously and monotonically when moves along \( P_4P_6 \), and

1. if for all \( P \), \( \theta(P) \in [-180, -90] \) or \( \theta(P) \in [90, 180] \), arc \( P_4P_6 \) must be a 1st quadrant arc or 3rd quadrant arc;
2. if for all \( P \), or \( \theta(P) \in [-90, 0] \), arc \( P_4P_6 \) must be a 2nd quadrant arc or 4th quadrant arc.

According to Theorem 1, 1st quadrant arc and 3rd quadrant arc can be discriminated from 2nd quadrant arc and 4th quadrant arc, but the gradient direction angle difference between 1st quadrant arc and 3rd quadrant arc, and the gradient direction angle difference between 2nd quadrant arc and 4th quadrant arc are not embodied for the sake of the uncertainty of gray variation in different directions.

Theorem 2. Suppose \( P_4P_6 \) is an arbitrary quadrant arc on an ellipse with two endpoints \( P_6 \) and \( P_4 \), is the midpoint of the chord \( P_4P_6 \) with coordinate \( (x_m, y_m) \), is the intersecting point of the perpendicular bisector line of \( P_4P_6 \) with arc \( P_4P_6 \) whose coordinate is \( (x_m, y_m) \), then just as shown in Fig. 5, the following statements must be true:

1. if \( P_4P_6 \) is a 1st quadrant arc or 2nd quadrant arc, \( y_m > y_6 \);
2. if \( P_4P_6 \) is a 3rd quadrant arc or 4th quadrant arc, \( y_m < y_6 \).

By utilizing the above two theorems synthetically, an arbitrary arc on an ellipse can be partitioned and determined into different quadrant arcs.

4.3. Quadrant arcs determination

According to the above definition and theorems, different quadrant arcs can be detected out from all the connected edges saved in LineBufGlobal. For each connected edge in, suppose its length (number of pixels contained in this edge) is \( N \), then define an \( N \)-dimensions label array \( F[N] \), and the detection and determination of different quadrant arcs can be executed as following steps:

...
(1) Gradient direction angle computation.

For each point on the connected edge, the direction of its normal line is determined by gradient information provided by the sub-image $I_{obj}$. Suppose the corresponding pixel in sub-image $I_{obj}$ to a point on the connected edge is $(i, j)$, the gradient components on this points along X-axis and Y-axis of image coordinate system can be computed out by Sobel Operator centered at $(i, j)$, which are denoted as $G_x$ and $G_y$, respectively. Then, the gradient direction angle $\theta$ of this point can be determined as:

$$\theta = \left\{ \begin{array}{ll}
\arctan(G_y/G_x) & \text{if } G_x > 0 \\
\pi + \arctan(G_y/G_x) & \text{if } G_x < 0 \\
\arctan(G_y/G_x) - \pi & \text{if } G_x = 0 \text{ and } G_y > 0 \\
\arctan(G_y/G_x) + \pi & \text{if } G_x = 0 \text{ and } G_y < 0
\end{array} \right. \quad (6)$$

and furthermore, for the convenience of threshold value selection, the angle value is mapped into $[-180, 180]$ from $[-\pi, \pi]$ by conversion between radian and degree.

(2) First determination of quadrant arc.

Suppose the gradient direction angle of $n$th point on a connected edge is $\theta_n$, then the angle's variation neighboring on this point can be measured as:

$$k_n = \min(|\theta_{n+2} - \theta_{n+1}|, |\theta_{n+1} - \theta_{n-2}|) \quad (7)$$

and the label array element corresponding to this point is determined as: if $k_n < T_n |F[n]| = 0$; else, $F[n]$ is set to be 1, 2, 3, 4 for $\theta_n \in [0, 90]$, $\theta_n \in (90, 180]$, $\theta_n \in (-180, -90)$ and $\theta_n \in [-90, 0)$, respectively, where $T_n$ is a given threshold value and in implementation $T_n = 4$.

The above assignment for the label array represents the initial judgment for a point which kind of quadrant arc it belongs to, while $|F[n]| = 0$ declares that the point may lie on a line segment. According to Theorem 1, there may be much confusion between 1st quadrant arc and 3rd quadrant arc, or 2nd quadrant arc and 4th quadrant arc, as well as some noising assignment for the sake that only local information is utilized around the corresponding point. Both of these two problems will be solved in the following steps.

(3) Assignment value filtering.

In order to remove or reduce noising assignment for the label array, a sliding window size of 5 is used to filter across $F[N]$. Suppose the index of the current element at the center of the sliding window is $n$ and the assignment value with most array elements in the window is $k$, then let $F[n] = k$, where $k \in \{0, 1, 2, 3, 4\}$.

(4) Quadrant arc extraction.

All points corresponding to continuous elements in $F[N]$ with same and non-zero value are extracted and saved into quadrant arc buffers $ArcBuf[n]$, where $n = 1, 2, 3, 4$ denotes the four categories of quadrant arcs and equals to the corresponding element value.

(4) Quadrant arc adjustment.

According to Theorem 2, the confusions between 1st quadrant arc and 3rd quadrant, and confusions between 2nd quadrant arc and 4th quadrant arc are clarified in this step. For each quadrant arc $A$ in $ArcBuf[n]$, suppose its two endpoints are $(x_n, y_n)$ and $(x_e, y_e)$, then the equation of the perpendicular bisector line of the line segment between these two points can be determined as:

$$L(x, y) = y - y_e + \frac{x_e - x_n}{2(y_e - y_n)}(x - x_n) = 0 \quad (8)$$

The cross point $P_M$ between line $L(x, y)$ and arc $A$ is determined as the point nearest to $L(x, y)$ on arc $A$, namely:

$$P_M = \arg \min_{P \in A} D(L, P) \quad (9)$$

where $D(L, P)$ is the distance from point $P$ to line $L$. Suppose the coordinate of $P_M$ is $(x_m, y_m)$, then the quadrant arc can be adjusted as following rules:

- if $y_m < (y_e + y_n)/2$ and $A$ is assigned as a 2nd quadrant arc in first determination of quadrant arc, it will be adjusted as 4th quadrant arc;
- if $y_m < (y_e + y_n)/2$ and $A$ is assigned as a 1st quadrant arc in first determination of quadrant arc, it will be adjusted as 3rd quadrant arc;
- if $y_m > (y_e + y_n)/2$ and $A$ is assigned as a 4th quadrant arc in first determination of quadrant arc, it will be adjusted as 2nd quadrant arc;
- if $y_m > (y_e + y_n)/2$ and $A$ is assigned as a 3rd quadrant arc in first determination of quadrant arc, it will be adjusted as 1st quadrant arc.

4.4. Combination of different quadrant arcs

The points set used for ellipse fitting can be formed by combination of different quadrant arcs. The combination process can be executed and tested by selection one from each quadrant arc set determinately or stochastically. In order to improve real-time property, some heuristic knowledge is utilized to guide the combination process in this paper, which is as followings:

(1) Select a 1st quadrant arc in $ArcBuf[1]$ whose one endpoint is nearest to the top of the sub-image $I_{obj}$ and by similar way select a 2nd quadrant arc in $ArcBuf[2]$ between these two arcs, select the one whose one endpoint is nearest to the sub-image's vertical center line as the candidate quadrant arc.

(2) Preferentially select the quadrant arc with short distance to the combined arcs, of course, this arc must belong to different quadrant arc categories from all the combined arcs. Suppose there have been $k$ combined arcs with endpoints $(x_i, y_i)$,
1 \leq i \leq 2K$, then the distance between these combined arcs and an arbitrary arc with endpoints $(x_a, y_b)$ and $(x_c, y_c)$ can be evaluated as:

\[
D = \min_i \left[ \min_j \sqrt{(x_i - x_b)^2 + (y_i - y_b)^2}, \min_j \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \right]
\]

(10)

According to the above principles, at least three different quadrant arcs are selected and combined to fit out an ellipse. The combination process is iterated on the four quadrant arc sets until the fitted ellipse satisfies the following conditions:

1. the ellipse falls into the sub-image $I_{obj}$ entirely;
2. according to the shape of human head, the ratio between ellipse's long axis and short axis must fall into range [1.0, 2.0];
3. the ellipse's center should fall into the sub-image's vertical strip ranging from $X_{center} - W_{obj}/5$ to $X_{center} + W_{obj}/5$, and the top point of the ellipse should fall into the sub-image's horizontal strip ranging from 0 to $H_{obj}/5$, where $X_{center}$ is horizontal coordinate of the sub-image's vertical center line. $W_{obj}$ and $H_{obj}$ are sub-image’s width and height, respectively.

5. Ellipse fitting

In this paper, the least square method is utilized to fit out the head elliptical contour based on the points set formed by different quadrant arcs combination. Suppose the fit point set is $P = \{(x_i, y_i) | 0 \leq i \leq N\}$ and ellipse equation with quadratic polynomial form is

\[
x^2 + k_0xy + k_1y^2 + k_2x + k_3y + k_4 = 0
\]

(11)

and the total error function is defined as

\[
G = \sum_{i=0}^{N} (x^2_i + k_0x_iy_i + k_1y_i^2 + k_2x_i + k_3y_i + k_4)^2
\]

(12)

Let $X = [k_0, k_1, k_2, k_3, k_4]^T$ and the derivations of the above function on these parameters equal to zero, then the final result can be rearranged into the form $AX = b$, where

\[
A = \begin{bmatrix}
\sum x_i^2 y_i & \sum x_i y_i^2 & \sum x_i^2 & \sum x_i y_i & \sum y_i \\
\sum x_i y_i^2 & \sum y_i^2 & \sum x_i & \sum y_i & n + 1
\end{bmatrix}
\]

(13)

\[
b = \begin{bmatrix}
-\sum x_i y_i \\
-\sum x_i^2 \\
-\sum x_i \\
-\sum y_i \\
N + 1
\end{bmatrix}
\]

(14)

The five unknown parameters can be obtained by solving the above equation. According to $k_0$, $k_1$, $k_2$, $k_3$ and $k_4$, the formulas for calculation of the five parameters in elliptical normalized form, namely the ellipse’s center $(x_c, y_c)$, semi-axis $(a, b)$ and the orientation $\theta$ of ellipse with to $X$ axis, are derived as:

\[
\begin{align*}
x_c &= \frac{2k_1 k_3 - k_0 k_4}{k_0^2 - 4k_4} \\
y_c &= \frac{2k_0 k_3 - k_1 k_4}{k_0^2 - 4k_4} \\
a^2 &= \frac{2k_0 \sin 2\theta + 2k_1 \cos 2\theta}{2k_0 \sin 2\theta + 2k_1 \cos 2\theta - m} \\
b^2 &= \frac{2k_1 \sin 2\theta + 2k_0 \cos 2\theta}{2k_1 \sin 2\theta + 2k_0 \cos 2\theta - m} \\
\theta &= \frac{1}{2} \arctan \left( \frac{k_0}{k_1} \right)
\end{align*}
\]

(15)

where $m = x_c^2 + k_1y_c^2 + k_3x_cy_c - k_4$.

6. Experiment results

In order to evaluate the performance of the elliptical head contour detection method proposed in this paper for arbitrary head pose and under complex background, a large quantity of video sequences are processed on-line. The experiment settings are: CCD camera
with specification SONY FCB-EX45AP; QP300 image grabber of Chinese Daheng Corporation; PM 1.6 GHz CPU and 512 M memory. The resolution of the original inputting image is $480 \times 360$.

6.1. Quadrant arcs extraction

In initial stage when there is not any moving object in the camera’s vision field, 200 frames images are captured and utilized to build the background model. Based on this background model, the moving object area in the original image can be detected out (some detection results of the moving object area are shown by red rectangles in Fig. 7 and Fig. 8). In sub-image corresponding to the moving object area, the connected edges and different quadrant arcs are detected out based on the method proposed in Section 4. Two examples are shown in Fig. 6, where the green lines denote the detected connected edges, and the 1st quadrant arcs, 2nd quadrant arcs, 3rd quadrant arcs and 4th quadrant arcs checked out from all the connected edges are shown in Fig. 6b–e with red color, respectively. It can be seen from this figure that by using of the method propose in this paper, the effective arcs externalizing the elliptical head contour’s appearance are not only extracted out from multiple complex edges, but also are partitioned into different categories of quadrant arcs according to their gradient information, which will enhance the system’s real time property and robustness to discretization errors to a large extent in ellipse fitting.

Fig. 7. Some examples of elliptical head contour detection under complex background.
6.2. Performance evaluation

Some examples of elliptical head contour detection under complex background are shown in Fig. 7, where three video sequences are included, the detected object sub-images are denoted by red rectangle and the head contours by green ellipse. In these video sequences, the subjects are a male with short hair, female with long hair and male with hat, respectively, and the longest distance between human and camera can reach about 9 m, just as shown in the third image of Fig. 7a, and the last image of Fig. 7b. From this figure, we can conclude that the method proposed in this paper not only can detect out the elliptical head contours under complex background, but also has strong adaptability to different subjects and better effectiveness within wider distance ranges.

Another group of examples of detection results are shown in Fig. 8. In these images, the subject waves his head in different direction and with large amplitude deliberately, which include yaw, pitch, roll or combination of them. From these images, it can be seen that the detection method based on the quadrant arcs is very robust to different head pose. In fact, as long as the majority of the head appears in the view field of the camera, the elliptical head contour can be detected out in most cases. When the head disappears or is occluded by other objects, just as shown by the last image of the second row in Fig. 8, the method becomes invalid, which can be solved by tracking technology furthermore.

In order to evaluate the performance of the method proposed in this paper under different conditions, we capture 3600 frame images of 5 persons (2 females with long hair, 2 males with short hair and 1 male with hat) at nine positions with distance to camera from 1 to 9 m at interval 1 m approximately. In these video sequences, the subject exhibits different head pose deliberately, including yaw, pitch, roll or arbitrary combination of them. Comparing to yaw, the amplitudes of roll and pitch own narrower variable range, which results into less impact on the detection of head ordinarily, except that it is deliberately to baffle the method (just as the cases shown in Fig. 8). So, at each position, 400 frames are roughly partitioned into four head pose ranges according to yaw angle, namely the angles between the person’s view line and the camera’s direction.

Each image is subjectively distributed into perfect detection, incomplete detection or false detection according to the detection effectiveness: if the detected ellipse is consistent to the head contour perfectly, it is regarded as perfect detection; if the detected ellipse falls into a sub-region of the head (as shown in Fig. 9a), or only a local arc of the ellipse is consistent with the head contour (as shown in Fig. 9b), it is regarded as an incomplete detection; if no ellipse is detected or the ellipse is wide of the mark with the head contour, it is regarded as a false detection. The statistical results are shown in Table 1. From this table, it can be seen that when the distance between the human and camera is less than 7 m, the average perfect detection rate can reach 93.5%; when the distance is farer than 7 m and less than 9 m, the average perfect detection rate reduces to 90.4%. When the person faces or backs to the camera, the perfect detection rates are higher than those when the person faces laterally, but under the latter condition and with farer distance to camera, the perfect detection rate can still reach
Among all the cases of not perfect detection, the incomplete detection takes high percent, which can be rectified by other technologies furthermore.

The detection method proposed in this paper has outstanding real time property, which can arrive 14 frames per second. Comparing to the ellipse detection method based on standard Hough transform and its variants, it is more suitable for the application with high requirement to processing speed, such as video surveillance or robotic human-tracking etc.

6.3. Extension to tracking and comparison with previous work

Although the method is proposed to solve the problem of elliptical head contour detection under static background, it can be easily extended to tracking under dynamic background without requirement to background modeling. Suppose the head contour parameters are \([x_c(k-1), y_c(k-1), a(k-1), b(k-1), \theta(k-1)]^T\) and \([x_c(k), y_c(k), a(k), b(k), \theta(k)]^T\) at times \(k-1\) and \(k\), respectively, then a candidate rectangle area can be determined to replace the moving object area to search for the elliptical head contour, whose center \((x_{rect}, y_{rect})\), width \(w_{rect}\) and height \(h_{rect}\) can be predicated as:

\[
\begin{align*}
   x_{rect} &= 2x_c(k) - x_c(k-1) \\
   y_{rect} &= 2y_c(k) - y_c(k-1) \\
   w_{rect} &= 2\sqrt{a(k)^2 \sin^2 \theta(k) + b(k)^2 \cos^2 \theta(k)} + \Delta w \\
   h_{rect} &= 2\sqrt{a(k)^2 \cos^2 \theta(k) + b(k)^2 \sin^2 \theta(k)} + \Delta h
\end{align*}
\]

where \(\Delta w\) and \(\Delta h\) are fixed as 8 pixels to ensure the majority of the head contour falls into this area. In this rectangle area, quadrant arcs are checked out and combined to fit out the elliptical head contour.

In order to demonstrate its efficiency, three video sequences (each contains 2000 frames) are applied and the results are compared with Birchfield’s tracking method [4], which is an elliptical head tracking method based on intensity gradients and color histograms, and exhibit higher performance in conditions of significant out-of-plane rotation, arbitrary camera motion, textured foregrounds and backgrounds than other methods. Some tracking results for each video sequence are shown in Figs. 10, 11 and 12, respectively. In the sequence shown in Fig. 10, the background is changed dynamically, while in Figs. 11 and 12, the tracking object is a woman with long hair and a man with a hat. The latter two sequences both involve the scenes containing multiple people.

From these figures and the statistical results on the whole video sequence, it can be seen that the two methods can both acquire desirable tracking effects in most cases and both show high robustness to head pose, but in the following aspects, our method is superior to Birchfield’s method:

1. It is more applicable to wider distance range. When the distance between human and camera becomes longer, our method can still track out the elliptical contour perfectly, while the Birchfield’s method usually becomes instable, which can be seen in the 9th and 10th frame in Fig. 10, the

![Fig. 10. Tracking results of a video sequence with dynamic background.](image)
6th frame in Fig. 11 and the 2nd frame in Fig. 12. By statistic on 1312 frames among these video sequences where the distance is roughly greater than 6m, the perfect detection rate reaches 91.3% using our method, while Birchfield’s method is only 75.6%. This is mainly caused by the fact that the color information applied in matching process becomes less when the distance becomes larger, while the shape can still be maintained and has little influences to the abstraction of quadrant arcs.

Fig. 11. Tracking a woman in a video sequence.

(2) Whether the object’s hair is long or short has less influences on our method. In Fig. 11, the head contour of a woman with long hair is tracked. When majority of the object’s face appears in the field of view, both methods can acquire satisfactory tracking results. When the object faces against camera, our method shows superior performance to Birchfield’s method, which is illustrated by the 1st, 5th, 8th and 10th frames in this figure. This kind of difference is mainly caused by the fact that when the object faces to the camera, both
the color and gradient information can be abstracted out to constrain Birchfield's method to converge to the true head contour; while when against to the camera, face color information disappears and the maximum gradient value lies on the boundary of the hair area, which results Birchfield's method can only converge to fit the long hair area. For our method, because long hair has less influences to head shape (except that the object scatters her hair deliberately) and the shape can be maintained and observed when the relative pose between the camera and the object is changed, in most cases at least 3 kinds of quadrant arcs can be detected out and the ellipse can be fitted to the true head contour. Among 524 frames where the object is against camera fully or partially in this video, the perfect detection rate is only 41.2% for Birchfield's method, while our method is 79.3%.

(3) Our method is more robust to whether the object wears a hat or not, and to hat color. This is demonstrated by Fig. 12, which shows some tracking results of a video sequence containing a man from without to with a blue hat. For lack of hair color information, the ellipse always tries to fit out the face boundary using Birchfield's method, which results into incomplete coverage on the head as shown by frames from the 4th to 10th in Fig. 12. For no reliance on color information, our method can keep high performance no matter whether the object wears a hat or not, as long as the head shape can be kept by the hat. Exceptions are ones like clown caps. Among 1160 frames where object wears the blue hat in this video, the perfect detection rate is only 44.3% for Birchfield's method, while our method is 84.9%.

(4) It is faster than Birchfield's method. By our method, it only needs about 15 ms (not including the frame grabbing time) for per frame averagely, while the other is 50 ms. On the one hand, the difference in time consuming is caused by computation and matching of color histograms in Birchfield's method. On the other hand, Birchfield's method is mainly accomplished via hypothesize-and-test procedure, which results into the fact that it usually needs multiple attempts before the desirable contour is matched out.

(5) It can achieve satisfactory performance under partially occlusion conditions, which can be seen in the 4th frame in Fig. 10.

When extended to tracking, the method proposed in this paper can both be applied to scenes containing multiple people. Figs. 11 and 12 are two representative examples. But "head stealing" as shown in Fig. 13 always happens under this condition. In this figure, when the tracked object passes by before the first person, it has no influence to the tracking result, but when he passes by behind the second person, for the reason of complete occlusion, the second person steals the ellipse away and becomes the tracked object for left frames. This kind of confusion can be solved by addition of recognition module, or by multiple objects tracking technology. Both are beyond the scope of this paper.

7. Conclusion

In this paper, a real time elliptical head contour detection method based on quadrant arcs is introduced. The method utilizes background model to extract out the moving object area. Based on the sub-image corresponding to the object area, the connected edges and different quadrant arcs are detected, discriminated and combined to fit out the elliptical head contour. The ellipse fitting process is implemented by least square method. The merits of this method are: ① it is applicable to complex background; ② it can detect out the elliptical contour for arbitrary head pose with high detection rate and precision; ③ it can meet the requirement of wider distance ranges; ④ it has stronger real time property; ⑤ it is robust to long/short hair and with/without hat. So, the method is more suitable for the applications with high requirement to processing speed and robustness, such as video surveillance or robotic human-tracking etc.

In this paper, the proposed method is mainly focused on the detection of the elliptical head contour under static backgrounds. As demonstration to the performance of this method, a simple extension to tracking problem is discussed in the experiments. As a future work, the elliptical head contour tracking problem under containing multiple people scenes should be researched furthermore.

Acknowledgments

This work was supported in part by National Natural Science Foundation of PRC under Grant Nos. 60705026 and 60805038; also supported by National High Technology Development 863 Program of PRC under Grant Nos. 2006AA042258 and 2006AA040202.

References


