Object Tracking Using Adaptive Object Color Modeling

Alireza Asvadi, MohammadReza Karami-Mollaie, Yasser Baleghi
1 Babol University of Technology
2 Babol University of Technology, mkarami@nit.ac.ir
3 Babol University of Technology, y.baleghi@nit.ac.ir

Abstract: This paper presents an object tracking method based on the object and background color. A three dimensional feature vector is used to describe color feature (R, G, B) of each pixel. A quantized histogram of that feature value of pixels on the selected object and a quantized histogram of pixels from the neighborhood surrounding the object are developed. The RGB based joint probability density function of the selected object region and that of a neighborhood surrounding the object are used to separate object from background. An adaptive color model of separated object is developed to detect object and successfully resolve the problems caused by the camera movement and illumination change, rotation, shape deformation and 3D transformation of the target object.

Keywords: Object Tracking, RGB based joint probability density function, Adaptive RGB Color Modeling, Mean-Shift.

1. Introduction

Tracking is a basic task for several applications of computer vision, e.g., video monitoring systems, traffic monitoring, automated surveillance, and so on. The objective of object tracking is to faithfully locate the targets in successive video frames. The major challenges encountered in visual tracking are, cluttered background, noise, change in illumination, occlusion and scale/appearance change of the objects. There exist many object tracking algorithms in the literature. These algorithms can be divided into three categories [1]: Point Tracking [2, 3], Kernel tracking (e.g. Mean shift [4], KLT tracker [5]), silhouette tracking (e.g. Variational methods [6], condensation algorithm [7]). These algorithms mainly differ in the way they use image features and model motion, appearance and shape of the object.

Here a fast algorithm of tracking is proposed that uses RGB based joint probability density function of the selected object region and that of a neighborhood surrounding the object to separate object from background in the first frame. Next color features of the object are used for modeling object color. Then object color model is employed to separate object from background in other frames meanwhile mean-shift procedure is used to track object location. Adaptive color modeling developed to overcome object appearance change during tracking.

The rest of the paper is organized as follows: Section 2 describes the proposed algorithm that presents object selection, feature extraction, Object–background separation, object color modeling and updating and Object localizer. The experimental results are shown in Section 3, and the conclusions are given in Section 4.

2. Proposed Algorithm

Fig. 1 shows the schematic diagram of the proposed method. Each block is described in the following sections. In object tracking, first one needs to develop a model for the object of interest from the given initial video frame. The objective of the object model is to accurately identify the object from the background. Next, the object localizer estimates the target location in the subsequent frames using the object model.

![Fig. 1: System overview](image-url)
2.1. **Object Selection**

Initially the object of interest is selected manually by the user input by drawing a rectangle around the object of interest. To detect the target object correctly, a background color near the target object is used. Object boundary and outer boundary are shown in Fig. 2. The outer rectangle is chosen such that the number of background pixels in the region surrounding the object is approximately the same as the number of pixels within the object rectangle. Equation (1) is used to define the width of the region surrounding the object rectangle:

\[
d = \frac{w+h}{4} \sqrt{2} - 1
\]  

(1)

Where \(w\) and \(h\) are the width and height of the object window and \(d\) is the width of the region surrounding the object rectangle [8].

2.2. **Feature Extraction**

In the proposed tracker the features used for modeling the object are quantized pixel color based features, which correspond to the values in quantized RGB color spaces. The quantized pixel-based features are extracted for the object pixels and surrounding background pixels. Fig. 3 shows quantized R, G, B color spaces. In the presented work 4-bit encoding of R, G and B channels is chosen that result the total histogram size of 16*16*16 = 4096. This decreasing in color depth and histogram size is performed for efficiency, and for defeating the curse of dimensionality. In order to represent the target appearance, quantized R, G, B pixel values of separated object from next section is used.

2.3. **Object–Background separation**

The object–background separation is used for detecting the object. The quantized R, G, B histogram of the region within the inner (red) rectangle is used to obtain the quantized RGB based joint probability density function (pdf) of the object region and the quantized R, G, B histogram of the region between the outer (green) and inner (red) rectangles is used for obtaining the surrounding background pdf. The resulting log-likelihood ratio (LLR) of an object region / background region surrounding the object is used to determine object pixels. The log-likelihood of a pixel considered within the object’s bounding rectangle is obtained as:

\[
L_i = \log \frac{\max[H_o(i, \varepsilon)]}{\max[H_b(i, \varepsilon)]}
\]  

(2)

Where \(H_o(i)\) is a histogram of that feature value of pixels in the object rectangle and \(H_b(i)\) is a histogram for pixels from the region surrounding the object, wherein the presented work index \(i\) ranges from 1 to 4096 that is the number of histogram elements. \(\varepsilon\) is a small non-zero value to avoid numerical instability that prevents dividing by zero or taking the log of zero. Here \(\varepsilon\) is set to 0.01. If the log likelihood function from the previous step is used to detect object pixels, the result is a likelihood image where, object color contains positive values, background color contains negative values and color that are shared by both object and background tend towards zero. The binary mask for object obtained as:

\[
M_i = \begin{cases} 
1 & \text{if } L_i > \tau_0 \\
0 & \text{otherwise}
\end{cases}
\]  

(3)

Where \(\tau_0\) is the threshold to decide on the most reliable object pixels. In the presented work, the value of \(\tau_0\) set at 0.8, in order to obtain reliable object pixels. Once the object region is selected manually by the user in the first frame, the log-likelihood map and binary mask of the object are obtained in order using (2) and (3) [8, 9, 10]. Fig. 4 shows in order, object region, likelihood image resulted from the log likelihood function using Equation (2) and binary mask of the object obtained by Equation (3).
2.4. Object Color Modeling and Updating

In the first frame, object color model is developed by using the quantized RGB value of separated object resulted from Equation (3). Fig. 5 shows the reliable quantized RGB seeds of the separated object correspond to Fig. 4c. However, the quantized color spaces essentially tolerate the small variation of color and illumination but it is not robust for extensive change in object color or scene illumination. In order to have a faithful object tracking color model updating of object is necessary. Modeling object color in each frame is computationally expensive, here a criterion is defined to indicate in which frames object color adaptation is needed. Assume $S_o$ is the average R-G-B color of pixels within the separated object. Change in $S_o$ shows the need for object color adaption. In the presented work in every incoming frame after detecting object the average R-G-B color of pixels within the separated object is computed. When $S_o$ in current frame deviated more than 0.05*256 in comparing with the last frame (Here, deviation set at 15), it shows the object color is changing and the object color adapting performed.

2.5. Object Localizer

The object localization starts at the centroid of the detected binary object in the frame where it was previously tracked. In order to find the object pixels, the features are extracted from object rectangle and are tested with the object color model. For tracking object mean-shift procedure is used. The main idea behind the mean-shift procedure is to treat the points in the spatial space as a probability density function, where the densest regions in the spatial space correspond to the local maxima which is object location [4]. The displacement of the object is given by the shift in the centroid of the object pixels. In each iteration center of object rectangle shift to the centroid of detected binary object. The object rectangle is iteratively shifted and tested until object completely placed inside the rectangle (mean-shift convergence). Object centroid relocates at each iteration by using Equations (4) and (5):

$$X_{new} = \frac{\sum_{i=0}^{n} x_i}{n} \quad (4)$$
$$Y_{new} = \frac{\sum_{i=0}^{n} y_i}{n} \quad (5)$$

Where $x_i$ and $y_i$ shows location of the each detected object pixels of the object rectangle in frame coordinate; $X_{new}$ and $Y_{new}$ show the relocated object centroid in each iteration and $n$ is the number of the detected object pixels.

In presenting work, the centroid movement less than 5 pixels considered as a complete convergence. Fig. 6 shows mean-shift convergence procedure for a typical object.

3. Experimental Results

The proposed algorithm has been tested on several complex video sequences. Here, three cases presented. In the first case, the moving ball undergoes severe illumination change, over time. It is observed that the proposed tracker fails to track object when it does not use color adaption. However, by using color model updating it is able to successfully track the object when it goes under extensive illumination change. The tracking result is shown in Fig. 7. Fig. 7a-l shows the fast moving ball in six consecutive frames. The tracked object without using color adaption is shown into red solid rectangle boundary whereas tracked object by using color model updating method is shown into the green dashed rectangle boundary. Fig. 7g-l shows the tracking results in a binary format when color adaption is not used. Fig. 7m-r shows the tracking results in a binary format when color adaption is used. As it can be seen as long as the object and the background is distinguishable in quantized RGB color feature space, the proposed method will perform satisfactorily.

![Fig. 5: Object color model in quantized RGB color spaces](image5.png)

![Fig. 6: The iteratively shifting of object rectangle location until convergence.](image6.png)
Fig. 7. (a-f) The tracked object without using color adaption (red solid rectangle) and tracked object by using color model updating method (green dashed rectangle), (g-l) Tracking results without using color adaptation in binary format, (m-r) Tracking results in a binary format when color adaption is used.

Fig. 8a shows color of object in first frame and Fig. 8b show color of object at sixth frame. The difference in object color when it undergoes illumination change is expressive.

Table I shows the detailed performance of the proposed algorithm for the case presented in Fig. 7. The first column shows the frame number. The second column shows the number of mean shift iteration to find an object location for each frame and third column shows updating color model is performed or not. The average mean-shift iterations for different scenarios with tolerance less than 5 pixels for object centroid achieves about 3. As it can be seen color model updating done in frames 3, 4 and 5.

TABLE I: Frame numbers, number of mean-shift iterations at each frame and frame numbers which the object color is updated by proposed method corresponding to Fig. 7.

<table>
<thead>
<tr>
<th>Frame number</th>
<th>Mean-shift iteration</th>
<th>Model Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>No</td>
</tr>
</tbody>
</table>

In the second case, the objective is face tracking. The proposed tracker is illustrated to be able successfully track face and resolve the problems caused by rotation, shape deformation, 3D transformation and movement of the camera. Fig. 9a-f shows the tracking results. Fig. 9g-l shows the tracking results in binary format.

Fig. 10a-f shows PETS 2001 Scenario [11]. Here, the multi-color object walks such that his body undergoes partial occlusion, as well as, appearance changes, over time.

In the second case, the objective is face tracking. The proposed tracker is illustrated to be able successfully track face and resolve the problems caused by rotation, shape deformation, 3D transformation and movement of the camera. Fig. 9a-f shows the tracking results. Fig. 9g-l shows the tracking results in binary format.

Fig. 9: (a-f) Tracking result of proposed method for face tracking. (g-l) Tracking result in binary format.
It is observed that the proposed tracker is able to track the object. One advantage of the proposed method is that if an object is partly occluded, it will not adversely affect the performance of the tracker. Fig. 10g-l shows the tracking results in a binary format.

4. Conclusions

This paper presents an object tracking method based on the object color modeling. A three dimensional feature vector is used to describe color feature (R, G, B) of the object. The RGB based joint probability density function of the selected object region and that of a neighborhood surrounding the object are used to separate object from background in the initial frame. Adaptive object color modeling is developed to overcome object appearance change during tracking. Object localization is achieved by estimating object in each of the frames by using the mean shift procedure. The proposed tracker is illustrated to be able to track object with moving camera and successfully resolve the problems caused by the illumination change, rotation, shape deformation and 3D transformation of the target object. The proposed method incurs very low computational load after building object color model that makes it suitable for real-time applications. Texture-based feature can be utilized for better discrimination and tracking results by accepting computational cost.

References


Fig. 10: (a-l) Tracking result of proposed method for ‘PETS 2001’ sequence. (g-l) Tracking result in binary format.