Multi-Feature Fusion Based on Particle Filter Algorithm for Moving Target Tracking

Meng-Xin LI, Gao-Ling SU, Ying ZHANG, Chong LI
School of Information and Control Engineering, Shenyang Jianzhu University, Shenyang, Liaoning, China
Tel.: 15040143907, fax: 024-24692899
E-mail: sugaoling2010@126.com

Received: 11 June 2014 /Accepted: 29 August 2014 /Published: 30 September 2014

Abstract: Robust moving target tracking has become an important topic in the field of computer vision. The fusion of feature such as color, texture, edge strength and motion has proved to be a promising approach to robust visual tracking in situations where no single feature is suitable. Due to the poor tracking performance in complicated scenarios, a new visual target tracking scheme is proposed. This algorithm exploits the fusion feature of color feature and local binary pattern (LBP) texture feature under the framework of particle filter to improve the robustness of target tracking. In order to get the accurate color model of target, the multi-part color histogram with spatial information is introduced. Comparing with the particle filter tracking algorithm based on single feature, experimental results show that the proposed method is more robust in terms of pose changes and illumination changes, especially in scenarios where the target object contains cluttered background with similar color distributions. Copyright © 2014 IFSA Publishing, S. L.

Keywords: Target tracking, Particle filtering, Multi-part color representation, LBP texture features.

1. Introduction

This template provides authors with most of the formatting specifications needed for preparing their articles. Target tracking, an important task of computer vision, is widely used in human motion recognition, video surveillance, video retrieval aspects, virtual reality, human-computer interfaces, etc. The main task of target tracking is to track and capture target trajectory and size changes, whose essence is to estimate the true position of the target. Particle filtering, a time series filter for estimating a state using probabilistic approach, can be effectively used for a variety of non-linear, non-Gaussian system.

One of the important parts for target tracking is object’s model describing its appearance such as contours, appearance template, or other features. Target tracking methods based on multi-feature fusion have been proposed by other researchers in recent years [1-3]. Color distributions are robust to partial occlusion, rotation, scale invariant and have a low computational cost. However, histogram has no spatial information. In order to solve this, multi-part histogram which has color histograms divided into sub-regions has been proposed [4-6]. These results in them show that multi-part histogram is more robust than single-part histogram because of using object’s spatial information. A multi-part representation is used in [4] to track ice-hockey players, dividing the rectangular box which bounds the target into two non-overlapping areas, generally corresponding to the shirt and trousers of each player. This solution is effective for the specific application. However, it is
Particle filters (PF) is a Bayesian sequential estimation method that appropriates the posterior density of state vector by using a weighted particles set [7]. Particle filter is known to be suitable of estimating signals with nonlinear and non-Gaussian distributions. In the particle filter, the posterior state vector is estimated by:

\[ p(s_k | z_k) = \frac{1}{N} \sum_{i=1}^{N} \omega_i \delta(s_k - s_k^i) \]

where \( s_k \) is the state vector at time \( k \), \( z_{1:k} \) is the observations (image pixels with the bounding box) up to \( k \). Using a weighted sum of randomly generated samples or particles drawn from a proposal distribution, the posterior pdf estimate can be approximated as:

\[ p(s_k | z_{1:k}) \approx \frac{1}{N} \sum_{i=1}^{N} \omega_i \delta(s_k - s_k^i) \]

where \( s_k^i \) is the \( i \)-th particle, \( \omega_i \) is the weight, and \( N \) is the total number of particles. It is worth noting that estimating the posterior probability of state vector is in fact, under the formulation of visual object tracking, also interconnected with, or related to choosing the best image content within the box. In the context of visual tracking, the importance proposal density \( q(s_k^i | s_{k-1}^i, z_k) \) is usually chosen as the state transition probability \( p(s_k^i | s_{k-1}^i) \), thus particle weights are proportional to the likelihood, \( \omega_i = \omega_{i-1} p(z_k | s_k^i) \) and the posterior vector is updated by weighted particles as:

\[ s_k = \sum_{i=1}^{N} \omega_i s_k^i \]

3. Target Features Extraction

While the particle filter is a powerful tool for robust tracking, its success depends largely on the choice of appropriate features to track. In this paper, we use color and texture information. In order to get the spatial information of color histogram, multi-part color histogram is exploited. To overcome the shortage of tracking algorithm based on color feature only, the LBP texture histogram is introduced.

3.1. Multi-part Color Histogram

Color histograms have been widely used to represent, analyze, and characterize images. They allow for significant data reduction, and can be computational efficient. Moreover, color histograms are robust to noise and local image transformations. The color distributions are expressed by color histograms discretized \( m \) bins. We calculate color histograms in the HSV space to be more invariant against illumination changes. The saturation and the brightness value of each pixel of such a region are compared against two thresholds 0.1 and 0.2, respectively, to decide if the region is a “hue region” or a “value region”. If more than 3/4 of the pixels have saturation and brightness bigger than the thresholds, the region is taken as a “hue region”. In the converse case, the region is considered as a “value region”, and if less than 3/4 of the pixels are of one kind, then the region is abandoned. Color histograms are composed of a 1-dimensional \( V \) histogram with \( m_v \) and a 2-dimensional \( HS \) histogram with \( m_h m_s \) bins. Therefore, total bins of color histograms are \( m = m_h + m_s m_v \). Not all pixels in the region are equally important to describe the objects. In order to increase the reliability of the color distribution when boundary pixels belong to the background or get occluded, smaller weights are assigned to the pixels that are further away from the region center by employing the Epanechnikov kernel:

\[ k(r) = \begin{cases} 1-r^2, r < 1 \\ 0, \text{otherwise} \end{cases} \]

where \( r \) is the distance from the region center. The color distribution \( p(y) = \{p(y)^{(i)}\}_{i=1...m} \) at location is calculated as:
\[ p(y)^{(a)} = f \sum_{i=1}^{N} k \left( \frac{\| y - x_i \|}{a} \right) \delta[h(x_i) - u], \quad (5) \]

where \( N \) is the number of pixels in the region, \( \delta \) is the Kronecker delta function, \( h(x_i) \) assigns one of the \( m \)-bins to a given color at location \( x_i \), and the parameter \( a \) is used to adapt the size of the region. The normalization factor:

\[ f = \frac{1}{\sum_{i=1}^{m} k \left( \frac{\| y - x_i \|}{a} \right) \delta[h(x_i) - u]}, \quad (6) \]

ensures that \( \sum_{u=1}^{m} p(y)^{(u)} = 1 \).

To calculate the likelihood of a candidate \( a \) function which defines a distance between the model and the candidate is needed. A commonly used metric is based on the Bhattacharyya coefficient \([7]\).

The distance between two bins normalized histograms \( p^u(u=1,...,m) \) and \( q^u(u=1,...,m) \) is defined as:

\[ \rho = \sum_{u=1}^{m} \sqrt{p^u q^u}, \quad (7) \]

\[ d = \sqrt{1-\rho}, \quad (8) \]

When two normalized histograms are a perfect match, we obtain \( d=0 \).

In the target tracking domain, color histograms are a popular form of target representation, because of their independence from scaling and rotation, and robustness to partial occlusions \([8]\). Nevertheless, the robustness of such a model is weakened due to the lack of spatial information in challenging tasks, such as target rotation and anisotropic scaling. Therefore, we use a multi-part color histogram. This problem can be limited by computing more than one histogram on different parts of the target. Generally, a region to calculate histogram is represented as Fig. 1 (a). We choose a multi-part histogram divided into four sub-regions as Fig. 1 (b). Four parts are obtained from the ellipse partition created by the two axes to add spatial information necessary to recognize rotations of the target.

\[ \text{Fig. 1. Color histogram representation.} \]

Hence the distance between the multi-part model \( p= \{p_1, p_2, p_3, p_4\} \) and the candidate \( q= \{q_1, q_2, q_3, q_4\} \) with ellipse parameters \( y \), is calculated using the average of the Bhattacharyya distance as:

\[ d_{\text{multi}}[p, q] = \frac{1}{4} \sum_{i=1}^{4} d[p_i, q_i], \quad (9) \]

where \( p_i \) and \( q_i \) are the i-th sub-region histogram shown as Fig. 1 (b) respectively, and \( d[p_i, q_i] \) is calculated by equation (7) and (8).

3.2. LBP Texture Histogram

The local binary pattern operator is an image operator which transforms an image into an array or image of integer labels describing small-scale appearance of the image. These labels or their statistics, most commonly the histogram, are then used for further image analysis. The most widely used versions of the operator are designed for monochrome still images but it has been extended also for color images as well as videos and volumetric data. The LBP operator was originally designed for texture description. The local binary pattern (LBP) is a nonparametric operator which describes the local spatial structure of an image. Ojala et. al \([9]\) first introduced this operator and showed its high discriminative power for texture classification. The operator assigns a label to every pixel of an image by thresholding the 3 \( \times \) 3 neighborhood of each pixel with the center pixel value and considering the result as a binary number. Then, the histogram of the labels can be used as a texture descriptor. See Fig. 2 for an illustration of the basic LBP operator.

\[ \text{Fig. 2. The basic LBP operator.} \]

In order to be able to deal with textures at different scales, the original LBP operator was extended to use a circular neighborhood of different radius size later \([10]\). Improved LBP operator defines the local neighborhood as a set of sampling points evenly spaced on a circle centered at the pixel to be labeled, which allows any radius and number of sampling points. Their \( LBP_{r,g} \) notation refers to \( P \) equally spaced pixels on a circle of radius \( R \). See Fig. 3 for an example of circular neighborhoods.
Fig. 3. The circular (8,1), (16,2) and (8,2) neighborhoods.

The operator can be expressed as follows:

\[ LBP_{p,R} = \sum_{p=0}^{P-1} 2^p (g_p - g_c), \]

where \( g_c \) corresponds to the grey value of the center pixel, to the values of the surrounding pixels, and function \( s(g_p - g_c) \) is defined as:

\[ s(g_p - g_c) = \begin{cases} 1 & \text{if } g_p \geq g_c, \\ 0 & \text{if } g_p < g_c. \end{cases} \]

4. Multi-Feature Fusion Based on Particle Filter

In this section, we discuss the state transition model and observation model of the target tracking system under the framework of particle filter at first. Then the fusion strategy of multi-feature is introduced. Finally, concrete steps of the algorithm are shown.

4.1. State Transition Model and Observation Model

The target regions are represented by ellipses, so that a sample is given as:

\[ s = \{x, y, \hat{x}, \hat{y}, H_x, H_y, \hat{H}_x, \hat{H}_y\}, \]

where \( x, y \) represent the location of the ellipse, \( \hat{x}, \hat{y} \) the motion, \( H_x, H_y \) the length of the half axes and \( \hat{H}_x, \hat{H}_y \) the corresponding changes. The sample set is propagated through the application of a dynamic model:

\[ s_i = As_{i-1} + w_{i-1}, \]

where \( A \) defines the deterministic and \( w_{i-1} \) the stochastic component. In our application we currently use a first order model for \( A \) describing an object moving with constant velocity for \( x, y, H_x \) and \( H_y \).

Expanding this model to second order is straightforward.

To weigh the sample set, the Bhattacharyya coefficient has to be computed between the target distribution and the distribution of the hypotheses. Each hypothetical region is specified by its state vector \( s^{(n)} \). Both the target \( q \) and the candidate histogram \( p_i(s) \), are calculated from Eq. 5 where the target is centered in the region and \( a = \sqrt{H_x^2 + H_y^2} \).

The observation probability of each sample:

\[ w_k = \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{d^2}{2\sigma^2} \right), \]

where is specified by a Gaussian with variance \( \sigma \).

Color Region Observation:

\[ w_k^c = \frac{1}{\sqrt{2\pi}\sigma_c} \exp \left( -\frac{d_i^2}{2\sigma_c^2} \right), \]

Textured Region Observation:

\[ w_k^l = \frac{1}{\sqrt{2\pi}\sigma_l} \exp \left( -\frac{d_i^2}{2\sigma_l^2} \right). \]

Next, to obtain the object’s state, we calculate the mean state of the sample set by Eq. 2.

4.2. Multi-feature Fusion Strategy

Common information fusion strategies mainly include multiplicative integration, weighted integration, maximum and minimum rules, etc. Multiplicative integration taking full advantage of a variety of information, is simple and easy to understand and implement. Moreover, it is proved as the best estimates under the assumption of independence according to principles of Bayesian filtering. This strategy is widely used in various fields of computer vision. The n clue joint likelihood model can be expressed as:

\[ p(z_1, \cdots, z_n | s) = \prod_{i=1}^{n} p(z_i | s), \]

where \( z_i \) is the observation under the i-th clue and independent, and is to be estimated state.

Color histogram is easy to be influenced by illumination changes and background confusion, while texture histogram is not sensitive to light variation. Therefore, we proposed target tracking algorithm based on color and texture features multiplicative fusion. In the tracking process, we
assume that the color and texture histograms are two separate clues, which each generate two independent observations. Hence, the observation equation of state $s_k$ is:

$$w'_k = w_k^S \cdot w_k^I,$$  \hspace{1cm} (18)

4.3. Concrete Steps of the Algorithm

In the framework of particle filter, the basic steps of target tracking algorithm as follows:

Step 1. Initializing
1) Select a target to track by elliptical region
2) Calculate sub-regions (in Fig. 1 (b) )color histograms $P_{c_1}, P_{c_2}, P_{c_3}$ and $P_{c_4}$ in the elliptical region, multi-part color histogram $c_p$.
3) Calculate texture histogram(LBP histogram) in the elliptical region $p_r = \{p_r^{(n)}, p_r^{(o)} p_r^{(a)}, p_r^{(a)}\}_{r=1,...,m}$

Step 2. Tracking
1) Propagate each sample from the sample set $s_i$ by equation (13)
2) Calculate each sample weight $w'_k$
   (a) calculate color observation probability $w_k^S$ of each sample $s_i^{(o)}$
   (b) Calculate texture observation probability $w_k^I$ of each sample $s_i^{(o)}$
3) Estimate the mean object state by equation 2
4) Select next new samples $\{(s_i^{(o)}, w_i^{(o)}), i=1,...,N\}$ from the sample set $\{(s_i^{(o)}, w_i^{(o)}), i=1,...,N\}$
5) Repeat (1) ~ (4) during tracking object

5. Experiments Results

To test the performance of proposed tracking scheme, three color videos have been used for our case study experiments. These videos contain moving target scenarios with a range of complexity, e.g. pose changes, illumination variations and similar color distributed background clutter. The image sequences have been downloaded from dataset (http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIAR DATA1/). For all videos, object is manually marked in the first frame.

The parameters are the same for all test video sequences. They are described as follows. The bins of $V$ histogram are $m_1=10$, the bins of $HS$ histogram are $m_2m_3=10 \times 10$. Total bins are $m=m_1+m_2m_3=110$. For particle filter in all the cases, the number of particles is $N=100$.

5.1. Case 1: Face Motion

The first set of experiments is to track moving face images. The tracking results are presented in Fig. 4. From left to right, the corresponding frame number is frame 20, 90, 115, 280 and 390 out of 500 frames. In this experiment, the tracked subject turns her head around which creates sudden color changes of the visible side of her head. At many frames the face is totally invisible, which can make color based particle filter tracking algorithm lose the target completely. The proposed scheme exploiting the fusion feature of multi-part color histogram and LBP histogram, which can extract details of the target template, successfully tracks the human face.

(a) Tracking results with color feature alone
(b) Tracking results with multiple feature fusion

Fig. 4. Comparison of the proposed algorithm and the other visual tracking.

5.2. Case 2: Face Motion

The video has been captured by the static camera and the selected target is a man who walks to meet someone from illuminated area to shaded area. The aim is to test the performance of proposed scheme in the scenarios of illumination variations. Fig. 5 shows some of the tracking results.

From left to right, the corresponding frame number is frame 35, 58, 64, 71 and 92 out of 123 frames. One can observe from figure that color based particle filter tracking algorithm fails to track target because of the weakness of illumination. In the contrast, the proposed scheme can track the target through the whole progress regardless of light influence. The reason for this is due to good complementarity of color feature and texture features. Hence, even the color characteristics of the light
changes, which generates with degradation, but the texture features can still play a role to avoid color degradation caused by track failure.

![Tracking results with color feature alone](image1)

![Tracking results with multiple feature fusion](image2)

Fig. 5. Comparison of the proposed algorithm and the other visual tracking.

5.3. Case 3: Face Motion

In this experiment, the selected target is intersected by a man having similar color distributions. The aim is to test whether the proposed scheme can track the target with similar background. Fig. 6 shows some of tracking results. From left to right, the corresponding frame number is frame 5, 15, 24, 30 and 34 out of 60 frames. One can observe from figure that color based particle filter tracking algorithm locks on the target with similar color in the background after intersection, resulting in loss of track. However, tracking algorithm with multiple features can avoid the influence of background confusion. The reason for this is that multi-part color histogram with spatial information introduced in the model avoids the lost track.

![Tracking results with color feature alone](image3)

![Tracking results with multiple feature fusion](image4)

Fig. 6. Comparison of the proposed algorithm and the other visual tracking.

7. Conclusions

This paper addresses the issue of tracking a visual target through complicated scenarios, where a target object may be influenced by pose changes, illumination variations and background confusion. A visual target tracking scheme is proposed that exploits the hybrid algorithm of the multi-part color representation and local binary pattern texture features under particle filtering framework to improve the robustness of target tracking. Comparing with the particle filter tracking algorithm based on color feature, experimental results show that the proposed method is more robust in terms of pose changes and illumination changes, especially in scenarios where the target object contains cluttered background with similar color distributions.

Future work includes utilizing multi-feature fusion based on particle filter algorithm to track the multiple targets in complicated scenarios.

References


