MULTI-KERNEL OBJECT TRACKING

Fatih Porikli, Oncel Tuzel

Mitsubishi Electric Research Labs, Cambridge, MA 02139

ABSTRACT

In this paper, we present an object tracking algorithm for the low-frame-rate video in which objects have fast motion. The conventional mean-shift tracking fails in case the relocation of an object is large and its regions between the consecutive frames do not overlap. We provide a solution to this problem by using multiple kernels centered at the high motion areas. In addition, we improve the convergence properties of the mean-shift by integrating two likelihood terms, background and template similarities, in the iterative update mechanism. Our simulations prove the effectiveness of the proposed method.

1. INTRODUCTION

Object tracking has two main tasks; detection of a new moving object, and finding the locations of the previous objects in the current frame. For stationary camera setups, detection of the new objects can be done by background subtraction, i.e. comparing the current frame with a reference model of the stationary scene. Wren [1] introduced a single unimodal, zero-mean, Gaussian noise process to describe the uninteresting variability, which corresponds to the reference background, in the scene. Earlier methods proposed to use Kalman filters to make a prediction of background pixel intensities. A Wiener filter is used by Toyama [2] to make a linear prediction of the pixel intensity values, given the pixel histories. Stauffer [3] suggested to represent the background with a mixture of Gaussian models. Elgammal [4] proposed a non-parametric approach where probabilistic kernels are used to model the density at a particular pixel.

The second task of object tracking, finding the previously detected objects in the current frame, can be done by a popular forward-tracking technique, the mean-shift tracking. The original mean-shift method is a non-parametric density gradient estimator. It is basically an iterative expectation maximization clustering algorithm executed within the local search regions. Comaniciu [5] has adapted the original mean-shift for tracking of manually initialized objects. The mean-shift tracker provides accurate localization and it is computationally feasible. However, it strictly depends on the assumption that object regions overlap between the consecutive frames. We proposed an automatic object tracking technique [6] that integrates a multi-modal background generation algorithm into a single-kernel mean-shift method for stationary camera setups. Here, a kernel represents the support region for the mean-shift searches, in other words, it is a window enclosing the previous location of the object.

Increasingly, object tracking systems are assembled from a large number of cameras. It is desired to achieve real-time tracking performance while keeping the hardware costs on an economical (often minimum) level. Therefore, it becomes necessary to process the vast amount of constantly streaming multiple channels of data on a single CPU at the same time. Unfortunately, most existing tracking approaches presume they can consume all the available processing power for a single sequence. Object tracking of multiple video sequences under the constricted computational power is still presents a major challenge.

One solution to enable processing of multiple video sequences on the same CPU is to sample every input video such that the number of frames per second is decreased proportional to the number of sequences. However, due to the decrease of the frame rate, the tracking algorithm receives video frames at a lower temporal resolution, which causes the objects to appear reciprocally much faster than to the original sequence. As a result, little or no overlap of object regions between the consecutive frames exist.

To understand the effect of using the low frame rate sequences on the performance of the single-kernel mean-shift tracking, we marked the ground truth (boundaries and trajectories of the moving objects) for benchmark test sequences, and evaluated the accuracy of our mean-shift based tracking method. The test sequences consist of more than 50,000 frames and depict both indoors and outdoors scenarios, partial and full occlusions, various object types such as pedestrians, vehicles, bicycles, etc. We measured the difference between the extracted trajectories and the ground truth. We also imposed a penalty term to incorporate other tracking failures such as wrong object initialization, object deletion, identity mismatches etc. The evaluation results are presented in Fig. 1. It is observed that the performance of the tracking method degrades as the frame rate gets lower values. The decline in performance is more apparent for the sequences that contain fast moving objects. In low frame

0-7803-9332-5/05/$20.00 ©2005 IEEE

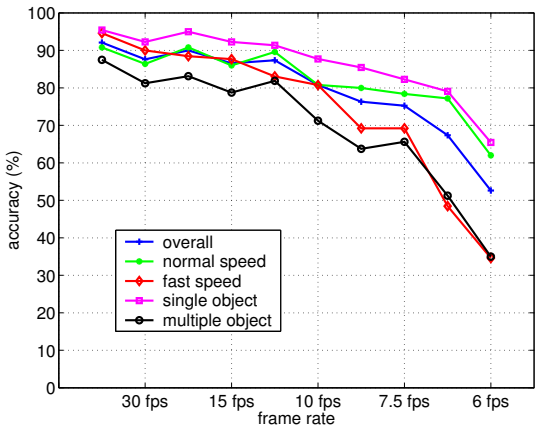

Fig. 1. Low-frame-rate video tracking require handling of fast moving objects.

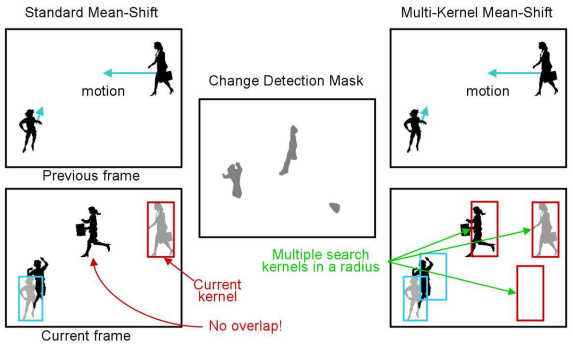

Fig. 2. We iterate the mean-shift in multiple kernels.

rate data, object movements are usually large and unpredictable, therefore a single mean shift window centered at the previous location of the target may not enclose the object in the current frame.

To overcome the above problems, we extend the mean-shift such that the iterative shifts are executed not only within a single kernel but in multiple kernels that are initialized at high motion areas of the scene and the previous location of the object, as explained in the next section.

2. MULTI-KERNEL MEAN-SHIFT

We estimate a statistical background model constructed by multiple layers using a Bayesian update mechanism, and we compare the current frame with the estimated background models to determine the foreground pixels in the current frame. The background generation is capable of adapting its learning coefficients with respect to the amount of the illumination change. We measure the distance between the pixel color and the corresponding models to obtain a distance map. A foreground mask is computed by thresholding the distance map using the pixel color variation, thus, the threshold is adaptive to each pixel, and varies in time. We keep track of two object sets. Objects that are not tracked for enough number of frames are marked as possible objects. We use the connected components to initialize objects. After estimating the location of each object, we match them with the connected components. An object is deleted if it is not matched with any of the connected components during a certain number of the subsequent frames. New objects are initialized accordingly from the connected components that are not matched with any of the current objects.

The current objects are tracked by multiple mean-shift kernels (as illustrated in Fig 2), and their shapes are adjusted by an in/out classifier using the distance map. We describe the details of the multi-kernel mean-shift tracking algorithm in the following sections.

2.1. Selection of Kernels

We apply a spatial clustering to the distance map. For each pixel, we weight the distance map value with respect to the distance between the pixel and the location of the kernel in the previous frame. This transformation assigns higher a likelihood to the pixels that are closer to the previous location of the object. Next, we find the peaks within the distance map. We choose the center-of-mass of the region of the object in the previous frame as an additional peak. We merge the peaks that are close to each other using the object size as a template. Then, we select recursively the pixels by starting from the pixel that has a maximum score, until a maximum kernel number is achieved or no possible kernel locations remains, by removing a region proportional to the object size at each iteration. Therefore, there is at least one peak, and depending on the amount of motion observed in the scene, there may be multiple peaks for each moving object. The maximum number is determined from the number of the current objects. We also use a weighting term to the initial set of kernel locations based on a pathway likelihood map that maintains the location history of previously tracked objects. We increase the value of the pixel in the pathway likelihood map if the object kernel corresponds to the pixel. We keep updating the pathway likelihood map for each frame in the video. Thus, after objects have been tracked in a large number of frames, the pathway likelihood map indicates likely locations of objects.

2.2. Object Model

Object model is a nonparametric color template. Template is a $(W \times H) \times D$ matrix whose elements are 3D color samples from the object, where W and H are the width

and height of the template respectively and D is the size of the history window. Let $\mathbf{z}_1$ be the estimated location of the target in current frame. We refer to the pixels inside the estimated target box as $(\mathbf{x}_i, \mathbf{u}_i)_{i=1}^N$, where $\mathbf{x}_i$ is the 2D coordinate in the image coordinate system and $\mathbf{u}_i$ is the 3D color vector. Corresponding sample points in the template are represented as $(\mathbf{y}_j, \mathbf{v}_{jk})_{j=1}^M$, where $\mathbf{y}_j$ is the 2D coordinate in the template coordinate system and $\mathbf{v}_{jk}$ is the 3D color values $\{\mathbf{v}_{jk}\}_{k=1..D}$. During tracking, we replace the oldest sample of each pixel of the template with one corresponding pixel from the image.

2.3. Background Information

Although color histogram based mean shift algorithm is efficient and robust for nonrigid object tracking, if tracked object color information is similar with the background, tracking performance reduces. We propose to use background information to improve the tracking performance.

Let **p(z)** be the color histogram of candidate centered at location **z** and **b(z)** be the background color histogram at the same location. We construct background color histogram using only the confident layers of the background. Again 2D Gaussian kernel is used to assign smaller weights to pixels farther away from the center.

Bhattacharya coefficient $\rho(\mathbf{p(z)}, \mathbf{q}) = \sum_{s=1}^{m} \sqrt{q_s p_s(\mathbf{z})}$, measures the similarity between the target histogram and histogram of the proposed location **z** in the current frame. We integrate the background information and define the new similarity function as:

$$\eta(\mathbf{z}) = \sum_{s=1}^{m} \sqrt{p_s(\mathbf{z})} \left(\alpha_f \sqrt{q_s} - \alpha_b \sqrt{b_s(\mathbf{z})} \right) \qquad (1)$$

where α_f and α_b are the mixing coefficients for foreground and background. Besides maximizing the target similarity, we penalize the similarity among the current and background image histograms. Let $\mathbf{z}_0$ be the initial location where we start search for the target location. Using Taylor expansion around the values of $p_s(\mathbf{z}_0)$ and $b_s(\mathbf{z}_0)$,putting constant terms inside Q_2, and using definition of **p(z)** and **b(z)**, the similarity function is rewritten as:

$$\eta(\mathbf{z}) \approx Q_2 + Q_3 \sum_{i=1}^{N} w_i k_N \left(\left\| \frac{\mathbf{z} - \mathbf{x}_i}{h} \right\|^2 \right) \qquad (2)$$

$$w_i = \sum_{s=1}^{m} \frac{\alpha_f \sqrt{q_s} - \alpha_b \sqrt{b_s(\mathbf{z}_0)}}{2\sqrt{p_s(\mathbf{z}_0)}} \delta[\hat{m}_f(\mathbf{x}_i) - s] - \sum_{s=1}^{m} \frac{\alpha_b \sqrt{p_s(\mathbf{z}_0)}}{2\sqrt{b_s(\mathbf{z}_0)}} \delta[\hat{m}_b(\mathbf{x}_i) - s]. \qquad (3)$$

where $\hat{m}_f()$ and $\hat{m}_b()$ maps a pixel in observed and background images, to the corresponding color bin in quantized color space. The spatial bandwidth h is equal to the half size of the candidate box along each dimension. The second term in (2) is equal to the kernel density estimation with data weighted by w_i. Mode of this distribution can be found by mean shift algorithm. Mean shift vector at location $\mathbf{z}_0$ becomes:

$$m(\mathbf{z}_0) = \frac{\sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{z}_0) w_i g_N(\| \frac{\mathbf{z}_0 - \mathbf{x}_i}{h} \|^2)}{\sum_{i=1}^{n} w_i g_N(\| \frac{\mathbf{z}_0 - \mathbf{x}_i}{h} \|^2)}. \qquad (4)$$

where $g_N(\mathbf{x}^*) = -k'_N(\mathbf{x}^*)$.

Fig. 3. Tracking samples of Multi-Kernel tracking at 6-fps temporal frame rate, that 4 out of 5 frames are dropped out from the original 30-fps video.

2.4. Template Likelihood

The probability that a single pixel $(\mathbf{x}_i, \mathbf{u}_i)$ inside the candidate target box centered at **z** belongs to the object can be estimated with Parzen window estimator:

$$l_j(\mathbf{u}_i) = \frac{1}{D h_c^3} \sum_{k=1}^{D} k_N \left(\left\| \frac{\mathbf{u}_i - \mathbf{v}_{jk}}{h_c} \right\|^2 \right). \qquad (5)$$

Bandwidth of the 3D color kernel is selected as $h_c = 16$. The likelihood of an object being at location **z** is measured

$$L(\mathbf{z}) = \frac{1}{N} \sum_{i=1}^{N} l_j(\mathbf{u}_i) k_N \left(\left\| \frac{\mathbf{x}_i - \mathbf{z}}{h} \right\|^2 \right). \qquad (6)$$

The kernel k_N assigns smaller weights to samples farther from the center making the estimation more robust.

(a) (b)

Fig. 4. Tracking results for the subsampled input sequence at 1-fps temporal resolution, that 29 frames are dropped out of every 30 frames. **a:** Single-Kernel, **b:** Multi-Kernel.

2.5. Fusion

We combine the location estimations of multiple kernels to determine the new location of the object in the current frame. We determine a model likelihood score for each kernel by comparing the object model with the kernel centered at the estimated location. The model likelihood distance includes color and template distances. It is possible to choose the location with the highest score as the new location.

Alternatively, we can infer the location estimations as a set of given measurements for a random variable, and the value of the estimation as the corresponding model likelihood scores. There is a certain analogy between this approach and the particle filtering. Each kernel is regarded as a particle and fusion is interpreted as estimation of a posterior likelihood function for this random variable. The maximum of the posterior function indicates the new location of the object.

3. EXAMPLES

Figure 3 shows a low frame rate tracking example (6 fps). Almost all frames, no overlap of object regions in the consecutive frames exists, which makes it impossible to track objects using single-kernel mean shift method. As visible, the the multi-kernel approach can resolve the tracking ambiguities arises dou to the existence of multiple objects. We also observed that the presented template likelihood improves the fusion performance for occlusion.

We give a comparison of the original and proposed algorithms in Fig. 4 where the original video sampled at 1-fps temporal rate in this case. Due to temporal sampling, there is no overlap between the consecutive object locations. As visible in the results, the multi-kernel method can track objects accurately even if the relocation between the successive frames is very large, unlike the single-kernel method.

The computation load of finding an existing object in the current frame increases as much as the number of the multiple kernels. Note that, the load does not change with respect to the number of objects when it is compared to the single-kernel method since the single-kernel method is also applied separately to each object. To improve the computational complexity, we limit the proximity of multiple kernels within a range depending on the frame-rate, e.g. for higher frame rates we assign smaller neighborhoods. We also start spatial sub-sampling of object models as the number of kernels increases.

4. SUMMARY AND DISCUSSION

We present an object tracking algorithm for low-frame-rate applications. We assign multiple kernels centered around high motion areas. We also improve the convergence properties of the mean-shift by integrating two additional likelihood terms. Unlike the existing approaches, the proposed algorithm enables tracking of moving objects at lower temporal resolutions as much as 1-fps frame rate without sacrificing the robustness and accuracy. Therefore, it can process multiple videos at the same time on a single processor.

Note that, the low frame rate constraint corresponds to the fast motion of the moving objects. Thus, the proposed method is capable of tracking fast objects even in the original frame rates.

5. REFERENCES

[1] C. Wren, A. Azarbayejani, T. Darrell, A. Pentland “Pfinder: Real-time tracking of the human body”, *PAMI*, 19-7, 1997

[2] K. Toyama, J. Krumm, B. Brumitt, B. Meyers, “Wallflower: Principles of background maintenance”, *ICCV*, 1999

[3] C. Stauffer and W.Grimson, “Adaptive background mixture models for real-time tracking”, *CVPR*, 1999

[4] A. Elgammal, D. Harwood, and L. Davis, “Non-parametric model for background subtraction”, *ECCV*, 2000

[5] D. Comaniciu, V. Ramesh, and P. Meer, “Real-Time Tracking of Non-Rigid Objects using Mean Shift”, *CVPR*, 2000

[6] F. Porikli and O. Tuzel,“Human body tracking by adaptive background models and mean-shift analysis”, *ICVS, Workshop on PETS*, 2003