MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Object Tracking in Low-Frame-Rate Video

Fatih Porikli, Oncel Tuzel

TR2005-013 March 2005

Abstract

In this paper, we present an object detection and tracking algorithm for low-frame-rate applications. We extend the standard mean-shift technique such that is is not limited within a single kernel but uses multiple kernels centered around high motion areas obtained by change detection. We also improve the convergence properties of the mean-shift by integrating two additional likelihood terms using object templates. Our simulations prove the effectiveness of the proposed method both under heavy occlusions and low frame rates down to 1-fps.

SPIE Image and Video Communications and Processing

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Copyright © Mitsubishi Electric Research Laboratories, Inc., 2005
201 Broadway, Cambridge, Massachusetts 02139

Object Tracking in Low-Frame-Rate Video

Fatih Porikli*, Oncel Tuzel

Mitsubishi Electric Research Laboratories, Cambridge, USA

ABSTRACT

In this paper, we present an object detection and tracking algorithm for low-frame-rate applications. We extend the standard mean-shift technique such that it is not limited within a single kernel but uses multiple kernels centered around high motion areas obtained by change detection. We also improve the convergence properties of the mean-shift by integrating two additional likelihood terms using object templates. Our simulations prove the effectiveness of the proposed method both under heavy occlusion and low frame rates down to 1-fps.

Keywords: Object Tracking, Mean-Shift, Surveillance Systems

1. INTRODUCTION

Video surveillance systems are designed to provide effective solutions to the demanding problem of monitoring secure environments by controlling user access and movement, detecting suspicious events, and enabling retrieval of vital information from huge amounts of recorded content. Typically, such systems consist of a large number of cameras and sensors. The multiplicity of components often brings financial and networking considerations together as much as it increases the computational load required to crunch the constantly streaming data. As a result, current products often need to improve processing power and minimize system costs by simultaneously handling multiple cameras on a single CPU. Since most robust tracking algorithms assume high-end processors to detect objects in real-time (or in a substantially short time), it becomes a challenge to scale such algorithms into a shared, thus, constricted computational power of the CPU.

Another issue arises due to the bandwidth and storage limits of such systems. Not always it is feasible to transmit or record all the available video in full temporal and spatial resolution. For instance, it is not possible to push more than a couple of color video streams through a conventional wired data channel without compressing the video first. It is almost prohitibitively expensive to record all video sequences for a long time period.

To facilitate sharing of available resources, we propose to subsample multiple video sequences in time and then send them to the processing unit. A simplified flow-diagram of the system is shown in Fig. 1. Due to subsampling, the object tracking algorithm receives video frames at a lower temporal resolution, which makes the moving objects to appear reciprocally faster in comparison to the original sequence. In subsampled data, rarely there will be an overlap of object regions between the consecutive frames. Since the most object tracking approaches make moderate motion assumptions, they eventually fail if the objects are moving fast. To improve the shortcomings of the existing approaches, we develop a low-frame-rate (LFR) tracking method based on the multi-kernel mean-shift technique. Our extensive simulations prove the robustness and effectiveness of the proposed LFR tracker.

In the next section, we give an overview of the tracking algorithms. In section 3, we introduce the LFR tracker and present the details of the multi-kernel method. In the last section, we provide sample simulation results.

2. BACKGROUND ON TRACKING

The most common approach for discriminating a moving object for stationary camera setup is background subtraction. That is, a reference image of the static scene is constructed, and the current frame is compared with this reference to detect the changed regions. The resulting difference image is thresholded to extract the moving regions. Although this task looks like fairly simple, in real world applications this approach rarely works. Usually background is never static and varies by time due to several reasons. The most important factors are lighting changes, moving regions and camera noise. Moreover in many of the applications, it is desirable to model the different possible appearances of the background such as shadows.

*fatih@merl.com, phone: 1.617.621.7586

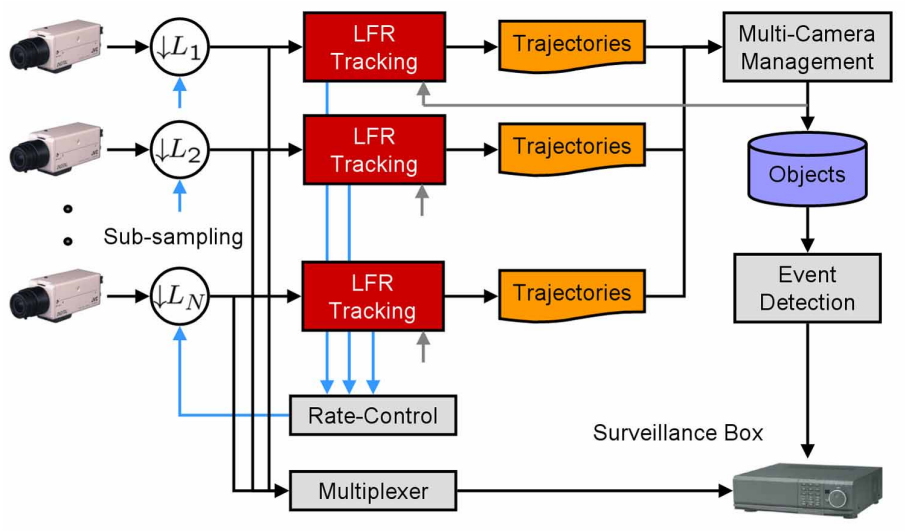

Figure 1. Multiple video sequences are subsampled in time to achieve processing them on a single processor.

Existing background subtraction methods can be classified as either single-layer or multi-layer approaches. Single-layer methods construct a model for the color distribution of each pixel based on the past observations. A simple approach assumes that the past observations fits into a certain function such as a uniform distribution, and estimates a mean value by averaging past values to determine the current value of the background. However, preset models and moving average operations generally cause so called ghost regions in the background that neither have the true background color nor the foreground object color. Wren[10] introduced a single unimodal, zero-mean, Gaussian noise process to describe the uninteresting variability in the scene. The background is updated with the current frame according to a preset weight, which acts as a learning factor; it adjusts how fast the background should be blended to the new frame. However, such a blending is sensitive to the selection of the learning factor. Depending its value, either the foreground objects may prematurely blended into the background, or the model becomes unresponsive to the observations.

To overcome these problems, adaptive background models became more popular. Earlier adaptive methods use simple adaptive filters to make a prediction of background pixel intensities. In Koller's tracker,[6] Kalman filtering is used to model background dynamics. Similarly Wiener filter is used by Toyama[9] to make a linear prediction of the pixel intensity values, given the pixel histories. The various parameters of the filter such as the transition matrix, the process noise covariance and the measurement noise covariance may change at each time step but are generally assumed to be constant. By using larger covariance values, the background adapt quicker to the illumination changes, however, it becomes more sensitive to the noise and moving objects in the scene. One drawback of the Kalman filter is its inability to represent multiple modalities, i.e. a background region depicts a swaying tree. Stauffer and Grimson [8] suggested to model the background with a mixture of Gaussian models. Rather than explicitly modeling the values of all the pixels as one particular type of distribution, the background is constructed by a pixel-wise mixture of Gaussian distributions to support multiple backgrounds. Stauffer's background update method make use of an expectation maximization (EM) based framework, and contains two significant parameters; a learning constant and a parameter that controls the proportion of the data that should be accounted for by the background. A similar competitive multi-modal background algorithm[7] was presented by Porikli. Elgammal[4] proposed a non-parametric approach where the use of Gaussian kernels for modeling the density at a particular pixel was proposed. The mixture methods are adaptable to illumination changes and they do not cause ghost regions. Robustness and speed are the two major bottlenecks of the existing approaches. Besides, accurate object segmentation and tracking under the constraint of low computational complexity still presents a challenge.

Previously, we developed a multi-modal background generation and mean-shift tracking algorithm[7] for full frame rate tracking. To understand the effects of the lower frame rates on the accuracy of the object tracking, we manually marked the boundaries and trajectories of the moving objects for more than 40 sequences that contain segments of approximately 1000 frames (40,000) frames. This segments include outdoors and indoors scenarios, lighting changes, severe occlusion,

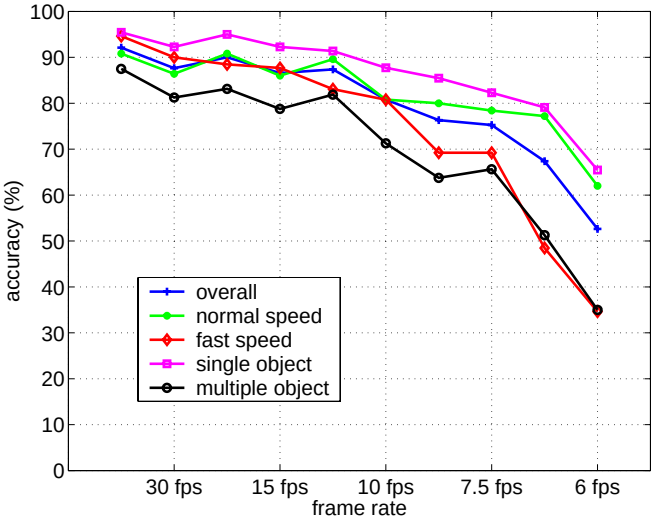

Figure 2. Low-frame-rates require a tracking algorithm capable of handling faster objects.

and multiple objects, e.g. pedestrians, vehicles, bicycles, etc. We run the tracker without any fine tuning at different frame rates. The results are evaluated by computing the distance between the detected trajectories and the ground truth. We also imposed a subjective penalty term to incorporate other tracking mistakes such as identity switches, wrong object initialization, deletion, etc. The evaluation results are given in Fig. 2. As expected, the accuracy degraded with the lower frame-rates.

3. LOW-FRAME-RATE TRACKING

We use a multi-modal background generation method[7] that updates the reference image according to the observed illumination changes. Then, we evaluate whether the current frame pixels fit to their existing background models, and assign the pixels that significantly diverge from the models as the foreground pixels. To initialize object regions, we remove small regions of pixels by morphology, and determine connected regions, which then will be grouped into the separate objects.

Low frame rate tracking algorithm keeps track of two object sets. Objects that are not tracked for enough number of frames are marked as possible objects. They may correspond to either noise in the scene or future tracked objects. After tracking a possible object for enough number of frames, it is removed from the possible object set and inserted into the tracked object set. We use the properties of the connected regions to determine the object regions. If the inner boxes of two connected regions are overlapping they are assigned to the same object. If their outer boxes are overlaying and the overlapping area is comparable to the area of the regions, they are assigned to the same object too. Then, for each group of merged regions, a possible object is set.

3.1. Multi-Kernel Mean-Shift

Mean shift algorithm is used to track objects in consecutive frames. Mean shift algorithm is a robust clustering technique which does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters.[2] The algorithm starts on the data points and at each iteration moves in the gradient direction. Iterations end when point converges to a local mode of the distribution. It is proved that convergence to a local mode of the distribution is guaranteed when mean shift iteration is started at a data point.

Mean shift tracking requires significant overlap on the target kernels in consequent frames. In low frame rate data, target movements are usually large and unpredictable so single mean shift window centered at the previous location of the target is not enough. To overcome this problem, besides the previous location of the target, multiple mean shift windows, so called as kernels, are initialized at high motion areas of the scene. Object template likelihood scores are computed at the converged points and maximum scored location is chosen as the target location as illustrated in Fig 3.

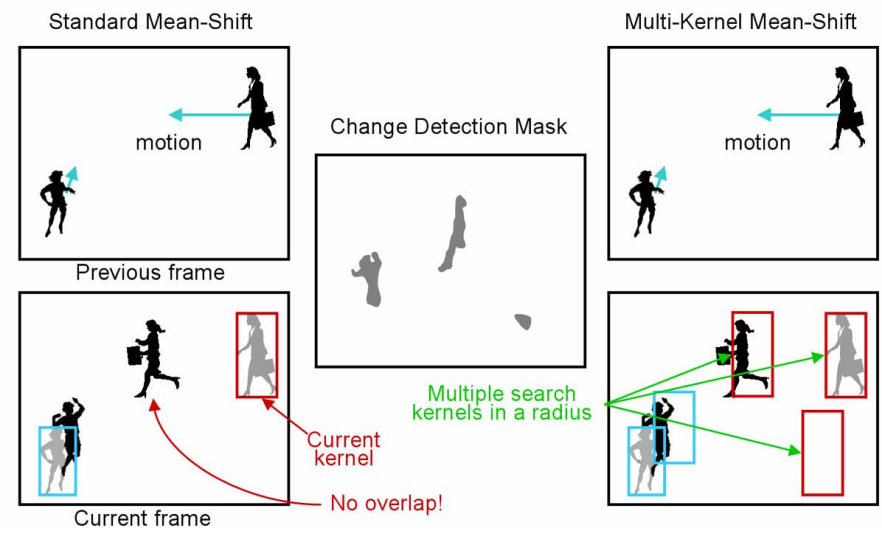

Figure 3. Instead of depending standard kernel, which is the object box in the previous frame, we iterate mean-shift in multiple kernels centered at high motion areas obtained by change detection.

Tracking of a single object can be summarized as follows.

Algorithm *Multi-Kernel Mean-Shift*
Input: Target at location $\mathbf{z}_0$ at previous frame and connected components centered at $\{\mathbf{c}_i\}_{i=1..l}$.

1. $L_{max} \leftarrow 0,\ i \leftarrow 0$
2. Initialize mean shift centered at previous target location
3. **while** $i \leq l$
4. Find mean shift vector $m(\mathbf{z}_0)$ using (9)
5. **while** $\eta(\mathbf{z}_0) < \eta(\mathbf{z}_0 + m(\mathbf{z}_0))$
6. $\mathbf{z}_0 \leftarrow \mathbf{z}_0 + m(\mathbf{z}_0))$
7. Find mean shift vector $m(\mathbf{z}_0)$ using (9)
8. Compute likelihood $L(\mathbf{z}_0)$ using (11)
9. **if** $L_{max} < L(\mathbf{z}_0)$
10. **then** $L_{max} \leftarrow L(\mathbf{z}_0)$, $\mathbf{z}_1 \leftarrow \mathbf{z}_0$
11. Initialize mean shift centered at i_{th} connected component ($\mathbf{z}_0 = \mathbf{c}_i$),
12. $i \leftarrow i + 1$

After estimating the location of each object, we match them with the connected components. The matching is performed based on the overlap of the object box with connected component pixels. An object is deleted if it is not matched with any connected component in subsequent frames. New objects are initialized for the connected components that are not matched with any already tracked object. If a currently tracked object is not merged with another one, its scale is updated using (12) and the template is updated. We describe the details of the multi-kernel mean-shift tracking algorithm in the following sections.

3.2. 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 $\{\mathbf{v}_{jk}\}_{k=1..D}$. Recall that index i inside the estimated target box maps

to index j in the template. This mapping is not one-to-one. Usually, size of the target box is much larger than the size of the template, so one pixel in the template maps to several pixels inside the target box. During tracking, we replace the oldest sample of each pixel of the template with one corresponding pixel from the image. We do not use mean of the several corresponding pixels to prevent blurring. Using foreground segmentation, template pixels which correspond to background pixels in the current frame are not updated.

3.3. Mean shift with 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 $\{q_s\}_{s=1..m}$ be the kernel weighted color histogram of the reference model. Reference model histogram is constructed using the nonparametric object template (Section 3.2):

$$q_s = Q_1 \sum_{j=1}^{M} \sum_{k=1}^{D} k_N \left(\left\| \frac{\mathbf{y}_j}{h_t} \right\|^2 \right) \delta(\hat{m}(\mathbf{v}_{jk}) - s) \tag{1}$$

where template bandwidth h_t is equal to half size of the template size (both horizontal an vertical) and k_N is the profile of the multivariate normal kernel:

$$k_N(\mathbf{x}^*) = (2\pi)^{-d/2} e^{-\frac{1}{2}\mathbf{x}^*} \tag{2}$$

for d dimensional space. Throughout the paper we use 2D kernel for spatial and 3D kernel for color space. Constant term Q_1 satisfies that $\sum_{s=1}^{m} q_s = 1$ and the function $\hat{m}$ maps a color value to the corresponding bin in quantized color space. Object template has a history information which makes the histogram more accurate in occlusions. 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:

$$\eta(\mathbf{z}) = \alpha_f \rho(\mathbf{p(z)}, \mathbf{q}) - \alpha_b \rho(\mathbf{p(z)}, \mathbf{b(z)}) \tag{3}$$

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. The location where the target is, should have a different appearance than the background. We use $\alpha_f = 1$ and $\alpha_b = 1/2$. The similarity function can be rewritten as:

$$\eta(\mathbf{z}) = \sum_{s=1}^{m} \sqrt{p_s(\mathbf{z})} \left(\alpha_f \sqrt{q_s} - \alpha_b \sqrt{b_s(\mathbf{z})} \right) \tag{4}$$

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 $b_s(\mathbf{z}_0)$

$$\begin{aligned} \eta(\mathbf{z}) \approx & \sum_{s=1}^{m} \sqrt{p_s(\mathbf{z}_0)} \left(\alpha_f \sqrt{q_s} - \alpha_b \sqrt{b_s(\mathbf{z}_0)} \right) + \sum_{s=1}^{m} \frac{\alpha_f \sqrt{q_s} - \alpha_b \sqrt{b_s(\mathbf{z}_0)}}{2\sqrt{p_s(\mathbf{z}_0)}} (p(\mathbf{z}) - p(\mathbf{z}_0)) \\ & - \sum_{s=1}^{m} \frac{\alpha_b \sqrt{p_s(\mathbf{z}_0)}}{2\sqrt{b_s(\mathbf{z}_0)}} (b(\mathbf{z}) - b(\mathbf{z}_0)). \end{aligned} \tag{5}$$

Putting constant terms inside Q_2 we obtain

$$\eta(\mathbf{z}) \approx Q_2 + \sum_{s=1}^{m} \frac{\alpha_f \sqrt{q_s} - \alpha_b \sqrt{b_s(\mathbf{z}_0)}}{2\sqrt{p_s(\mathbf{z}_0)}} p(\mathbf{z}) - \sum_{s=1}^{m} \frac{\alpha_b \sqrt{p_s(\mathbf{z}_0)}}{2\sqrt{b_s(\mathbf{z}_0)}} b(\mathbf{z}). \tag{6}$$

Using definition of **p(z)** and **b(z)**, the similarity function is rewritten as:

$$\eta(\mathbf{z}) \approx Q_2 + Q_3 \sum_{i=1}^{N} w_i k_N \left(\left\| \frac{\mathbf{z} - \mathbf{x}_i}{h} \right\|^2 \right) \tag{7}$$

$$w_i = \sum_{s=1}^{m} \frac{\alpha_f \sqrt{q_s} - \alpha_b \sqrt{b_s(\mathbf{z}_0)}}{2\sqrt{p_s(\mathbf{z}_0)}} \delta[\hat{m}_f(\mathbf{x}_i) - s] - \sum_{s=1}^{m} \frac{\alpha_b \sqrt{p_s(\mathbf{z}_0)}}{2\sqrt{b_s(\mathbf{z}_0)}} \delta[\hat{m}_b(\mathbf{x}_i) - s]. \tag{8}$$

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 (7) is equal to the kernel density estimation with data weighted by w_i. Mode of this distribution (maximum of similarity function (3)) can be found by mean shift algorithm. Recall that the weights w_i might be negative. Unlike,[11] we use zero instead of negative weights. Alternatively, Collins describes how negative weights can be used to construct mean shift vector.[1] Mean shift vector at location $\mathbf{z}_0$ becomes:

$$m(\mathbf{z}_0) = \frac{\sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{z}_0) w_i g_N(\| \frac{\mathbf{z}_0 - \mathbf{x}_i}{h} \|^2)}{\sum_{i=1}^{n} w_i g_N(\| \frac{\mathbf{z}_0 - \mathbf{x}_i}{h} \|^2)}. \tag{9}$$

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

3.4. 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:

$$l_j(\mathbf{u}_i) = \frac{1}{D h_c^3} \sum_{k=1}^{D} k_N \left(\left\| \frac{\mathbf{u}_i - \mathbf{v}_{jk}}{h_c} \right\|^2 \right). \tag{10}$$

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

$$L(\mathbf{z}) = \frac{1}{N} \sum_{i=1}^{N} l_j(\mathbf{u}_i) k_N \left(\left\| \frac{\mathbf{x}_i - \mathbf{z}}{h} \right\|^2 \right). \tag{11}$$

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

3.5. Scale Adaptation

Scale adaptation of the objects are performed using the foreground pixels. Let B be the box of the object centered at estimated location $\mathbf{z}_1$. We define a second box O around the object center which has twice area of B. We are trying to maximize

$$S = \sum_{\mathbf{x} \in B} \hat{c}(\mathbf{x}) + \sum_{\mathbf{x} \in O - B} (1 - \hat{c}(\mathbf{x})) \tag{12}$$

where $\hat{c}(\mathbf{x})$ is one if **x** is a foreground pixel and zero otherwise. At each frame, leaving O fixed we modify B $\pm 5\%$ in all dimensions and chose the scale which gives the best score.

Figure 4 shows a low frame rate tracking example (6 fps). Usually, there is no overlap in object boxes in two consecutive frames, that makes it impossible to track with original mean shift algorithm. The results show that, our object template likelihood function is very effective in resolving ambiguities caused by multiple objects in the scene. Moreover, fusion of background information makes significant improvements over the histogram based mean shift tracker.

We give a comparison of the original and proposed algorithms in Fig. 5 where the original video sampled at 1-fps temporal rate. Due to temporal sampling, there is no overlap between the consecutive object locations. As visible in the results,the original algorithm initializes objects, however it fails to find their position in the following frame. On the other hand, the multi-kernel method successfully tracked objects even if the relocation between the successive frames is very large.

We found that the multi-kernel method can also handle multiple object scenarios thanks to the competent template models. We observed that the template improve the tracking performance under heavy occlusion as well, e.g. an object temporarily disappearing behind a tree completely.

Figure 4. 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.

4. SUMMARY AND DISCUSSION

In this paper, we present an object tracking algorithm for low-frame-rate applications. We extend the standard mean-shift technique such that its is iterated within the multiple kernels centered around high motion areas obtained by the change detection. We also improve the convergence properties of the mean-shift by integrating two additional likelihood terms, namely object template vs. current image, and background image vs. current image likelihoods to the original current image vs. background image likelihood.

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, we can process multiple high temporal rate videos at the same time on a single processor by subsampling the input sequences. Our simulations show that the multi-kernel method performs superior at the full temporal resolution as well.

Acknowledgments

We would like to thank Professor Peter Meer at Rutgers University for his vision and valuable comments.

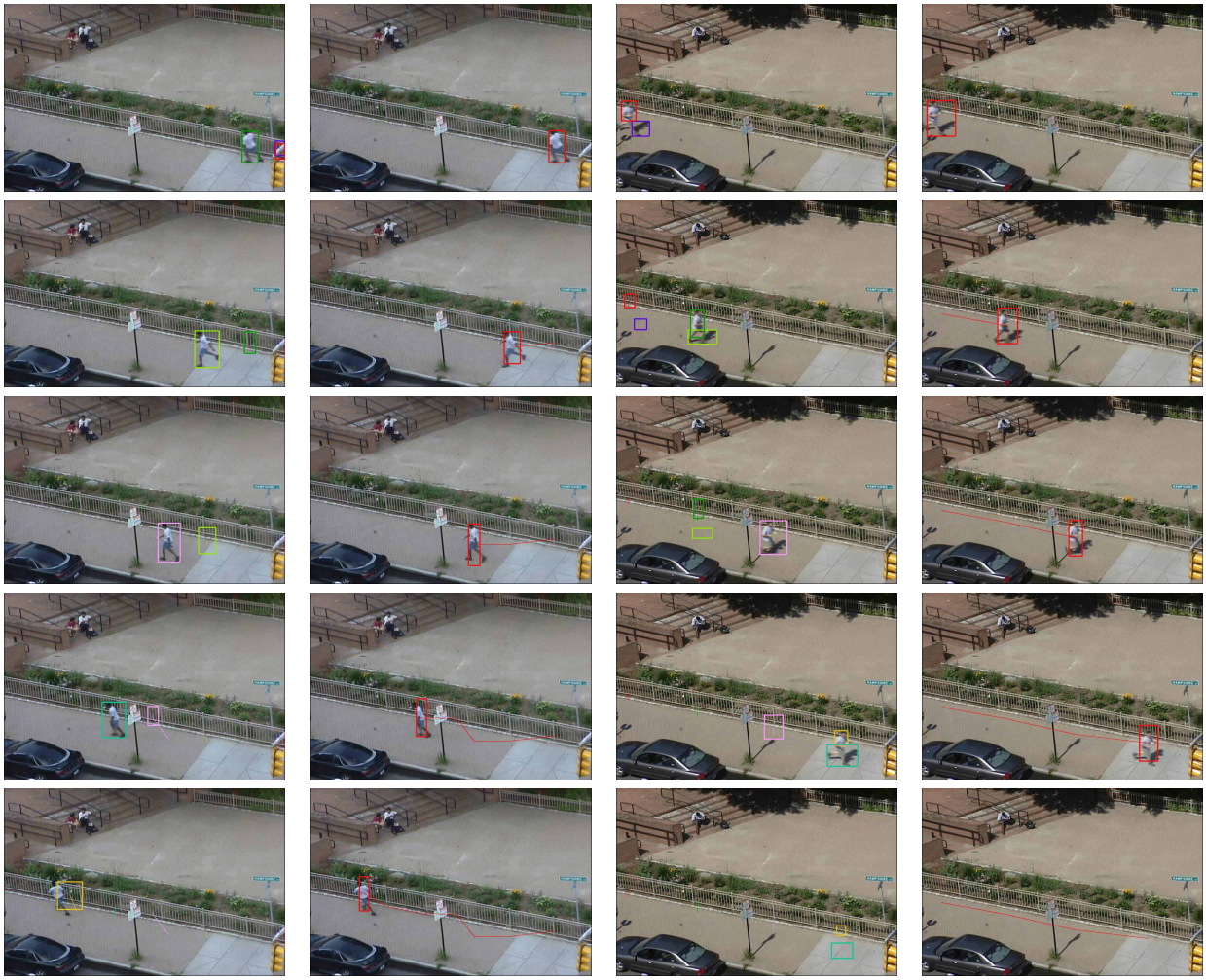

Figure 5. Tracking results for the subsampled input sequence at 1-fps temporal resolution, that 29 frames are dropped out of every 30 frames. **a,c:** Standard approach. **b,d:** Multi-Kernel tracking.

REFERENCES

1. R. Collins, "Mean-shift blob tracking through scale space," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Madison, WI, 2003, pp. 234–240.
2. D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, pp. 603–619, 2002.
3. D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, pp. 564–577, 2003.
4. A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. European Conf. on Computer Vision*, Dublin, Ireland, volume II, 2000, pp. 751–767.
5. K.-P. Karman and A. von Brandt, "Moving object recognition using an adaptive background memory," in Capellini, editor, *Time-varying Image Processing and Moving Object Recognition*, volume II, (Amsterdam, The Netherlands), Elsevier, 1990, pp. 297–307.
6. D. Koller, J. Weber, and J. Malik, "Robust multiple car tracking with occlusion reasoning," in *Proc. European Conf. on Computer Vision*, Stockholm, Sweden, 1994, pp. 189–196.
7. F. Porikli and O. Tuzel, "Human body tracking by adaptive background models and mean-shift analysis," in *Conference on Computer Vision Systems, Workshop on PETS*, IEEE, April 2003.

8. C. Stauffer and E. Grimson, “Adaptive background mixture models for real-time tracking,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Fort Collins, CO, volume II, 1999, pp. 246–252.
9. K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, “Wallflower: Principles and practice of background maintenance,” in *Proc. 7th Intl. Conf. on Computer Vision*, Kerkyra, Greece, 1999, pp. 255–261.
10. C. Wren, A. Azarbayejani, T. Darell, and A. Pentland, “Pfinder: Real-time tracking of the human body,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 780–785, 1997.
11. T. Zhao and R. Nevatia, “Tracking multiple humans in crowded environment,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Washington, DC, volume II, 2004, pp. 406 – 413.