Second Joint IEEE International Workshop on Video Surveillance and Performance Evaluation of Tracking and Surveillance
In conjunction with the Tenth IEEE International Conference on Computer Vision, Beijing, China, October 15-21, 2005.

Online Learning of Region Confidences for Object Tracking

Datong Chen and Jie Yang

School of Computer Science
Carnegie Mellon University
{datong, yang+}@cs.cmu.edu

ABSTRACT

This paper presents an online learning method for object tracking. Motivated by the attention shifting among local regions of a human vision system during tracking, we propose to allow different regions of an object to have different confidences. The confidence of each region is learned online to reflect the discriminative power of the region in feature space and the probability of occlusion. The distribution of region confidences is employed to guide a tracking algorithm to find correspondences in adjacent frames of video images. Only high confidence regions are tracked instead of the entire object. We demonstrate feasibility of the proposed method in video surveillance applications. The method can be combined with many other existing tracking systems to enhance robustness of these systems.

1. Introduction

Object tracking is one of the prerequisites for analyzing and understanding video data, and has been an active research topic in the computer vision community over the last two decades. The aim of tracking is to automatically find the same object in an adjacent frame from a video sequence once it is initialized. The previous research on object tracking falls into three different categories: appearance modeling, motion modeling, and searching methods.

Appearance modeling represents an object in an image by certain features from video. Different color models, e.g., color histograms, are commonly used to characterize moving objects in images. Online selection of color spaces has been studied in object tracking [1]. Appearance can also be simplified by using contours in tracked objects that can be easily segmented from their backgrounds [2]. On the other hand, pixel values are widely used to represent background objects, such as in inter-frame differencing scheme, Gaussian mixture model over time [3], adaptive filter methods [4], minimal and maximal intensity value methods [5], PDE level set [6], Hidden Markov models (HMMs) [7], and kernel density estimation techniques [8].

A motion model predicts an object's location in a new frame using its history. Linear models impose constraints that an object can only has translational or affine motions [9]. Non-linear models impose less constraint on motion than linear models, but they are more difficult to estimate and are more sensitive to noise [10].

Searching methods use various strategies to find an object, within an area predicted by a motion model, that is most similar to the appearance of the tracked object in an adjacent frame of a video sequence. Apart from the location, a searching algorithm may also search for the most proper scale of the tracking target. Many efficient searching algorithms, such as mean-shift [11], have been developed to search local best matching of a rigid or non-rigid object. A Kalman filter tracks objects using both foreground and background motion models [13]. Particle filtering is superior to Kalman filtering using non-parametric density estimation and multiple hypotheses [14]. These three approaches, however, only address the problem of "how to track an object," under the assumption that what to track is known.

In this paper, we are mainly interested in investigating another important problem: "what should be tracked?" in a tracking process. In other words, our goal is to improve object tracking by tracking selected local regions instead of tracking the entire object.

Occlusions and complex backgrounds (backgrounds with similar colors as foreground objects) are two major challenges for tracking an object in video. Many tracking errors are encountered when an object is tracked in front of a background that looks similar to the object in terms of tracking features. For example, in Figure 1 (row 1) the person in the blue shirt and white pants looks very similar to the background with blue windows and white wall if the tracking algorithm depends on color information. Tracking errors may also occur because of occlusion. Even partial occlusions will sometimes dramatically change the appearance of a tracked target in a feature space, and therefore increase the probability of tracking failure. However, a human vision system does not have any problem performing perfect tracking in these cases. In practice, very often a partially occluded object still contains enough discriminative tracking information in its non-occluded regions relative to its surroundings. Recent psychology research results [15] suggest the same, showing that a human vision system does shift its attention among different local regions during the observation of a moving object. Unlike the global focus of the attention process, which has been applied to model human attention [19] and motion grouping [20], this subtle attention shifting implies that different local regions of an object should have dynamically different confidences during a tracking task.

In this research, we present an approach for learning the confidences of local regions of an object online during tracking. We propose associating different local regions of an object with different confidences on the basis of their discriminative powers from their background and probabilities of being occluded. To this end, object appearances are first accumulated using a layered representation. We then partition the object area indicated by its associated layer into overlapping regions with the same size. The confidences of these regions are learned online using the distinguish power on the basis of matching distances and a Gaussian occlusion model, based on the discriminative powers and the occlusion probabilities. The discriminative power of a region can be exploited between the region and its local background [1]. The occlusion probability of a region is estimated from the distances between front layers. We further propose three alternative bias models to integrate multiple region confidences into a state-of-the-art tracking algorithm. Using these models, we illustrate the advantages of using region confidences against occlusions and a complex background.

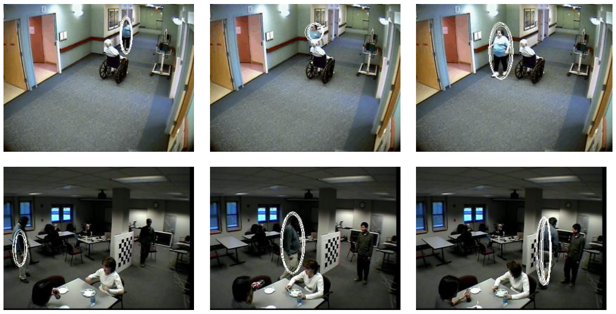


Figure 1 Illustrated high confidence regions in object tracking with occlusions and complex backgrounds. The first row illustrates that the center of the high confidence regions shifts up when the bottom of the object is occluded. The second row shows the changes of high confidence regions due to the background similarity and potential occlusion.

2. A Method of Exploiting Region Confidences

Intuitively, we believe that a tracking system may pay more attention to non-occluded regions or outstanding regions to track an object against occlusion and a complex background. Recent psychology results [15] have provided some scientific evidence for this process in a human vision system. Figure 1 illustrates two examples of the shift to regions with high confidence during tracking. The first example shows that the center of high confidence shifts up when the bottom of the object is occluded. In the second example, tracking attention is paid firstly to the middle of the object because the upper and lower parts of the object have similar color to the surrounding background. The confidence of the upper body increases as the object moving away from the blue windows. The last image shows that a potential occlusion from an object (person) on the right reduces the confidence of the regions on the right side. Although the mechanism of this function in a human vision system is not completely known to psychologists, as computer scientists we are curious to know if a functionally similar mechanism can be implemented in a computer vision system to improve object tracking.

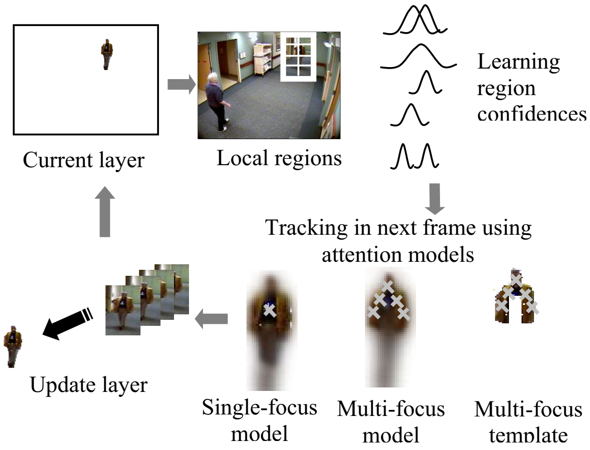


Figure 2 An overview of the proposed approach.

Figure 2 shows the diagram of an approach to characterize the role of region confidences in a tracking process. In the proposed approach, we first use a layer representation to remember the appearance of an entire object. There are two types of layers, front layer and background layer, which are updated using tracking results from every frame.

Then, an object is tracked by focusing on some high confidence regions instead of the entire object. The confidences of the regions are learned from the tracking results in the previous frame. Given location and scale of an object in the previous frame, we partition the object and its surroundings with a sliding window into several overlapping regions. The confidence of each region is then computed according to its discriminative power and probability of being occluded.

After obtaining the high confidence regions of an object, we map them to its latest layer model and crop their appearances. To combine multiple high confidence regions, we propose three alternative spatial bias models.

Each of these spatial bias models can be integrated into in a traditional tracking algorithm, for example the mean shift algorithm.

The location and the scale of the entire object are then predicted by using the relative positions of the tracked regions in its layer model.

3. Layered representation

In order to keep updating appearances of an object in video frames, we build a layer for each object in the foreground and a layer for the background. Layered representation of a video was proposed by Wang et al. [16] and extended by many other researchers, including Darrell et al. [17] and Tao et al. [18]. In this paper, a layer model of an object at time *t* is defined as $L_t = \left(PO_t, A_t, D_t\right)$, where PO_t is denoted as the location of an object in frame *t*. We assign this value as the geometric center of the object. A_t indicates the appearance of the object, and D_t encodes the depth of the layer in our representation. The background layer has a depth $D_t = 0$. The depth of a front layer is always greater than 1 but may vary during a tracking process.

3.1. Kernel density estimation of a layer model

Traditionally, the appearance of a layer at time *t* should consist of only the object segmented from background and occlusions. However, a perfect segmentation is difficult to obtain with occlusions. We use the kernel density estimation (KDE) proposed by Elgammal [8] to avoid the difficulty of segmentations in layer modeling.

Given a set of appearances $A = \left(A_{t_1}, A_{t_2}, ..., A_{t_n}\right)$ of a layer extracted with rectangular windows from *n* frames, we can normalize the size of each appearance and represent it as $\overline{A} = \left(\overline{A}_{t_1}, \overline{A}_{t_2}, ..., \overline{A}_{t_n}\right)$. Let $A_t\left(x\right)$ be a pixel value at a location *x* in the rectangle appearance patch of A_t. Given the observed pixel value $A_t\left(x\right)$ in a tracking candidate window A_t (can also be normalized to $\overline{A}_t$), we can estimate the probability of this observation as:

$$\mathrm{P}\left(\overline{A_t}\left(x\right)\right) = \frac{1}{n}\sum_{i=1}^{n}\alpha_i K\left(\overline{A_t}\left(x\right), \overline{A_{t_i}}\left(x\right)\right), \qquad (1)$$

where K is a kernel function defined as a Gaussian function:

$$K(x_1, x_2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left\|x_1 - x_2\right\|^2}{2\sigma^2}}. \qquad (2)$$

The constant σ is the bandwidth. Using the color values of a pixel, the probability can be estimated as:

$$\mathrm{P}\left(\overline{A_t}\left(x\right)\right) = \frac{1}{n}\sum_{i=1}^{n}\alpha_i \prod_{j\in\left(R,G,B\right)} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(\overline{A_t}\left(x\right)^j - \overline{A_{t_i}}\left(x\right)^j\right)^2}{2\sigma^2}} ,(3)$$

where α_i, *i= 1,2....n*, are the weights associated with appearance samples in the model *A*, which can be computed as:

$$\alpha_i = \frac{1}{\left|\overline{A}\right|} \sum_{x\in \overline{A}_{t_i}} \mathrm{P}\left(\overline{A}_{t_i}\left(x\right)\right). \qquad (4)$$

3.2. Layer update

Using the KDE technique, the layer update adapts the appearance of model *A* using new samples. The model *A* needs to be initialized with only one example at the beginning of a tracking process. Appearances of the tracked object are added into *A* until the number of samples reaches a pre-specified value *n*. We use *n*=25 in this paper.

4. Region confidence learning

Let us denote D_θ as the matching distance between the appearance of an object (layer *L*) and a hypnosis image area θ in an object tracking algorithm. The matching distance is usually defined in a feature space against some types of variance in the object appearance during the tracking. For example, the color histogram is robust to non-rigid object motion. To apply the matching distance on local regions of the layer *L*, we divide a layer into *m* overlapping regions as shown in Figure 3. For each region *r*, we denote C_r as its confidence, $\sum_r C_r = 1$. We assume the matching distance of the entire area is a linear combination of its local regions:

$$D_\theta = \sum_r C_r\, D_\theta^r\,, \qquad (5)$$

where D_θ^r is defined as the matching distance of the region *r*. We further define the correct location of the object as θ_0. Since an optimal set of region confidences should make the matching distance at the correct location D_{θ_0} outstanding from the other locations, we maximize the sum of the square of the differences of the matching distances from locations θ s to the θ_0 with respect to the confidence set $C = \{C_r\}$:

$$\arg\max_\theta \sum \left(D_\theta - D_{\theta_0}\right)^2 = \arg\max_C \sum_\theta \left(\sum_r C_r \left(D_\theta^r - D_{\theta_0}^r\right)\right)^2 .(6)$$

Obviously, we should only choose one region in the ideal case to obtain the best tracking result. In practice, we do not know the correct location θ_0 in a new video frame before tracking. However, the values D_θ^r and $D_{\theta_0}^r$ can be estimated from the previous tracking result, when appearances of the object and the background do not dramatically change.

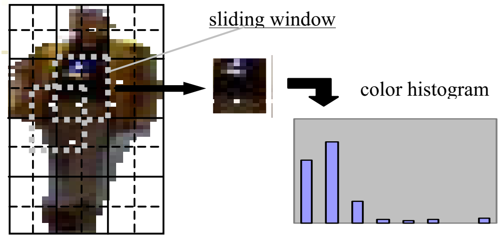


Figure 3 Overlapped regions of an object located by its associated layer.

In detail, given the tracked location of the object in a previous frame, we are able to compute D_{θ}^{r} and $D_{\theta_0}^{r}$ using the layer model and obtain the region confidences. These confidences are very similar to the corresponding confidences in the current frame, though they are not exactly the same. In our experiments, regions that contain small, non-rigid motion appear slightly different, as they are in the layer model. Occlusions may also change the appearances of regions dramatically. It is not very robust if we only keep one region as in an ideal case. It is sometimes even harder to find an outstanding region in an object layer, especially when local regions are too small. The confidences therefore need to be estimated in a more robust way. To this end, we consider that a region's confidence C_r is related to predicted discriminative power and the predicted probability of being occluded.

From Eq (7), the discriminative power of a local region is expected to have a high value if the sum of the square of the differences of the matching distances between the hypnosis locations θs and the true locations θ_0 is high. We thus define a function *f* that select regions with the highest predicted discriminative power of the region *t* using the D_{θ}^{r} and $D_{\theta_0}^{r}$ from the object layer and latest frame. Let us define f as the selective function based on the discriminative power of the local regions. The method of learning the function *f* is discussed in the Section 4.1. Apart from the predicted discriminative power, we also need to consider occlusions that may introduce appearance changes when regions in a layer model are used to predict the object appearance in a new frame. Occlusions may completely change the appearance of some regions. Let us denote the probability of region *r* being occluded O_r. We use O_r as a weight in estimating the region confidence. Therefore, the confidence can be estimated as:

$$C_r = \alpha f_r \left(1 - \beta O_r\right), \qquad (7)$$

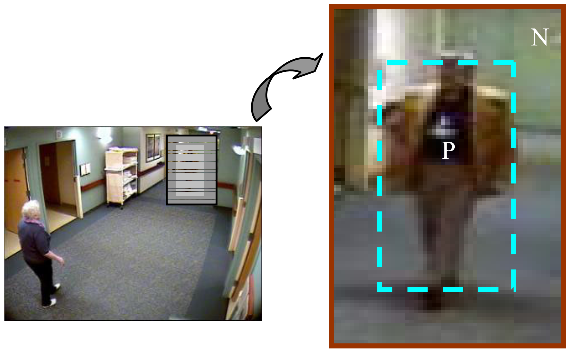


Figure 4 Sampling the regions around a moving front layer. The regions containing in the layer are labeled as positive examples (*P*) and the regions surround the layer are labeled as negative examples (*N*).

where α is a normalizer to keep the sum of the confidences equal to 1. The parameter β is a scalar to amplify the probability O_r into the range of [0, 1].

4.1. Learning the discriminative power

Although selecting only one region is not always practical, we still expect to select as few regions as possible to avoid unnecessary computation. Therefore, the discriminative power function *f* should be able to select as few regions as possible while keeping the discriminative power strong and robust enough to track an object in a new frame.

We first limit outside area *N* to the surroundings of the object, which is usually the same as the searching range of a tracking algorithm. As we partition an object area (area *P*) into local regions, we may accidentally partition some background area (area *N*) into local regions as shown in **Figure 4**. The discriminative power of a candidate region *r* in the *P* area is the matching distances from this region to every the regions in the *N* area, which is defined as the weighted sum of the matching distances:

$$Dist(r) = \frac{\sum_{nei \in N} \left\| nei - r \right\| D(nei, r)}{\sum_{nei \in N} \left\| nei - r \right\|}, \qquad (8)$$

where the matching distance function *D* is the same as the color histogram matching distance used in Eq. (5) and the distance $\left\| nei - r \right\|$ is denoted as the distance between the centers of the two regions.

To select regions using the *Dist* values, we model the distribution of the *Dist* values of all regions in the *P* area as Gaussian:

$$P(Dist(r)) \sim G(\mu, \sigma).$$

The regions with high confidences are then selected the threshold:

$$\tau = \mu + \sigma,$$

which means that the selective function f is defined as:

$$f_r = f(r) = \begin{cases} 1 & Dist(r) > \tau \\ 0 & otherwise \end{cases} \quad (9)$$

Various numbers of regions can be selected from different appearances of objects.

4.2. Occlusion probability

The occlusion probability of a region should be higher if it is close to another front layer. Formally, if two layers L^1 and L^2 are close enough, which means the distance between the two locations $\left\| L^1 - L^2 \right\| < \varepsilon$, we define the occlusion probability of a region r in the L^1 using the nearest Gaussian distance:

$$O_r = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{\|nei - r\|^2}{2\sigma^2}} , \quad (10)$$

where *nei* is the closest region in the layer L^2 to the region *r*. The distance $\|nei - r\|$ is denoted as the distance between the centers of the two regions. The variance σ is related to the predicted moving speed of the object. Now, we can compute the confidence of each region using Eq. (7).

5. Tracking objects using region confidences

In order to illustrate the proposed idea without losing generality, we integrate the proposed online learning mechanism into the mean-shift algorithm [11]. That is, we track objects using spatial-biased color histogram as the appearance feature for each object. The bias is computed on the basis of the region confidences. Given a layer *L* and its region confidences C_t the tracking of the layer *L* in a subsequent frame can be described as the following three iterative steps:

Table 1 Tracking algorithm with region confidence

1. Compute a bias for each position in the layer *L* using a specific bias model;
2. Compute spatial-biased color histogram of the latest appearance of the layer *L* using the biases;
3. Track the layer *L* using spatial-biased color histogram in the new frame;

Traditional color histogram-based algorithms, as we used in the Section 4, consider each pixel as the same weight or radial weights. The spatial-biased color histogram calculates the color distribution of a region with different bias at each pixel. We model this bias over an object area as the combination of the learned regions confidences. Three different bias models are presented here.

5.1. Single focus model (SFM)

A human vision system tends to have one focus at a time. Accordingly, we would like our model to focus on only one point at a time. The bias reduces as distance increases from the focus. Let $Loc(t)$ be denoted as the relative center location of the region *T* of a front layer *L*. We define the focus *F*(*L*) as the average location of the centers of all local regions weighted by their confidences:

$$F(L) = \frac{1}{\sum\limits_{t_i \in L, i=1}^{|P|} C_{t_i}} \sum_{t_i \in L, i=1}^{|P|} C_{t_i} Loc(t_i) \cdot \quad (11)$$

We model the weights of relative positions in the layer as a Gaussian distribution in which higher weights are given to the position close to the focus. For each position *x* in the layer *L*, the bias of *x* is taken from:

$$w(x) \sim N\left(F(L), \sigma_s\right). \quad (12)$$

The single focus model is very robust to occlusions because only one local part of the object needs to be tracked. However, the focus may sometimes fall into a region that has very low confidence due to the averaging step. This usually happens when many high-confidence regions surround a region with low confidence. To address this problem, we propose a multi-focus model.

5.2. Multi-focus model (MFM)

The multi-focus model works like the vision systems of bees or other insects, which have many simple eyes working together. We create a multi-focus model using a Gaussian mixture model, in which each local region with non-zero attention is modeled as a Gaussian distribution. Following the definitions in the last section, the weight of a position *x* in the layer *L* can be computed as:

$$w(x) = \frac{1}{\sqrt{2\pi\sigma_m{}^2} \sum\limits_{t_i \in L, i=1}^{|P|} C_{t_i}} \sum_{t_i \in L, i=1}^{|P|} C_{t_i} e^{-\frac{\|x - Loc(t_i)\|^2}{2\sigma_m^2}} . \quad (13)$$

5.3. Multi-focus template (MFT)

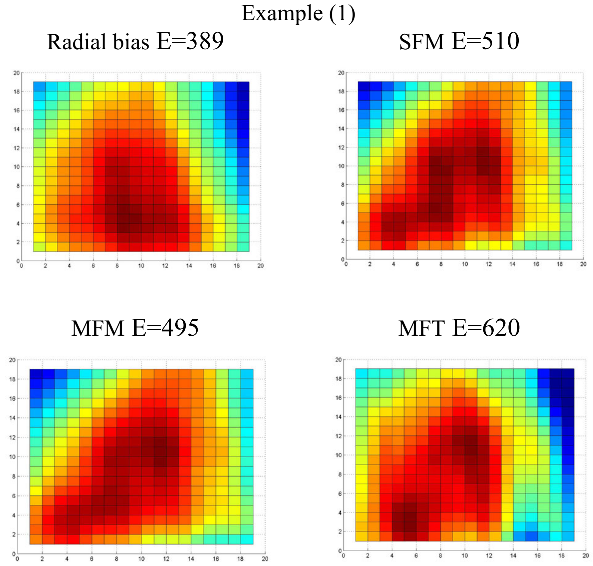
The third focus model we propose is a template. We simply put a threshold δ on the attention of the local regions, and give each pixel in threshold regions an equal weight. Formally, the bias of a position *x* in the layer *L* is:

$$w(x) = \begin{cases} \eta & x \in t \,\&\, C_t > \delta \\ 0 & otherwise \end{cases} , \quad (14)$$

where η takes the value that keeps the sum of the bias of all positions into 1.

6. Experiments

We first compared the matching distance distribution with and without using region confidence. The matching distances are computed around the manually labeled ground truth. To show the values at all possible locations, we perform a full search, though in practice no tracking algorithm will really search so many locations. Figure 5 shows matching distance distributions computed for tracking three objects. We reverse the distance values so that the centers can be seen in 3D surfaces. In each example we extract 35 *P* regions and 18 *N* regions for confidence learning.

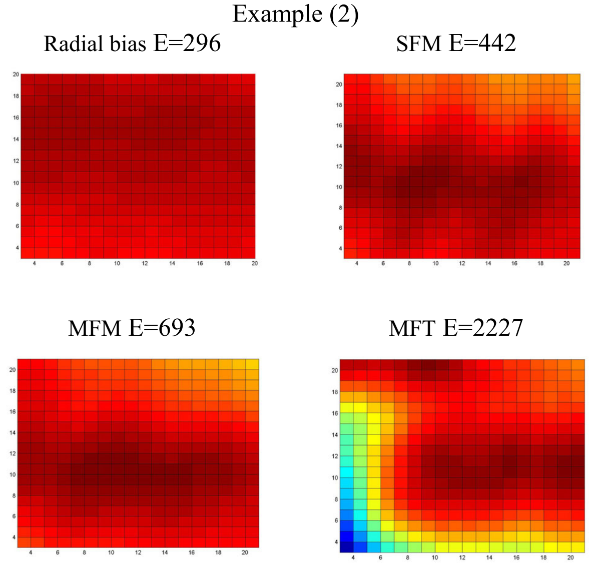

Example (1)

Radial bias E=389

SFM E=510

MFM E=495

MFT E=620

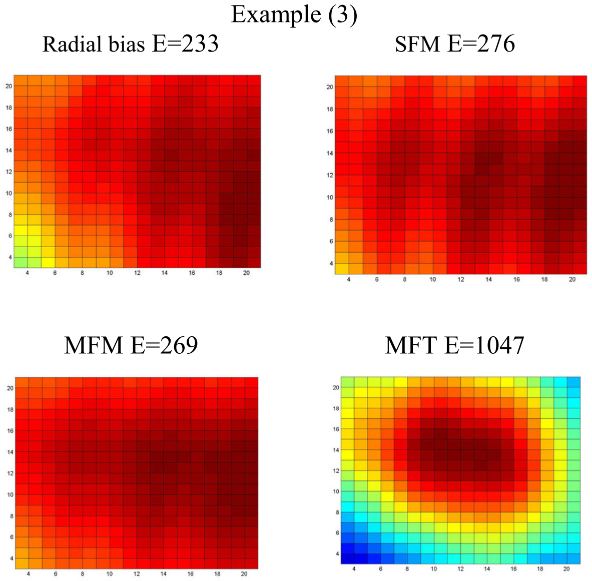

Example (2)

Radial bias E=296

SFM E=442

MFM E=693

MFT E=2227


Example (3)

Radial bias E=233

SFM E=276

MFM E=269

MFT E=1047

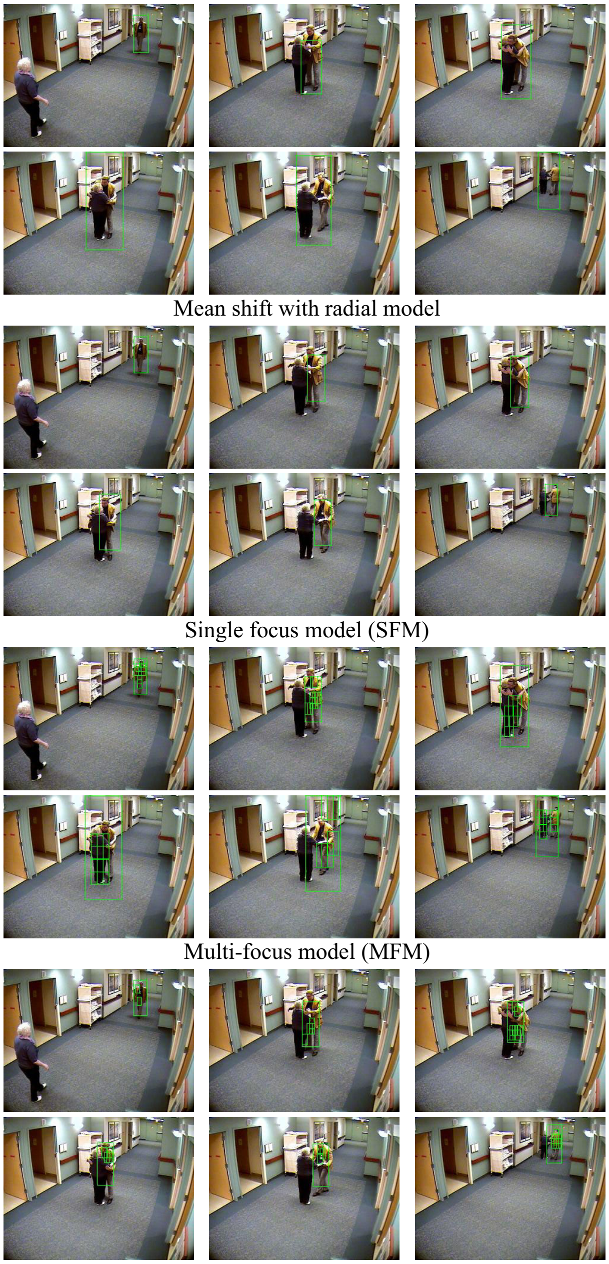
Figure 5 Three examples (in columns) of matching distance (reversed for display) distributions computed around the ground truth locations in tracking. In each example, we report the results by using different bias from left to right: conventional radial, single-focus model, multi-focus model, and multi-focus template. (Different examples may use different color ranges)

The evaluation value E is derived from the Eq. (6), with the biased color histogram.

$$E = \sum_{\theta} \left(D_{\theta} - D_{\theta_0} \right)^2 . \qquad (15)$$

In the first example, we are tracking a small object (30x12 pixels) that is occluded by another object. The MFT has the best chance to provide correct tracking according to the E value. However, the distributions of the four methods look very similar. The second example is obtained by tracking a bigger object (90x36 pixels) in front of a background that contains similar colors. The MFT shows not only the best E value but also a smooth distribution. The MFM also performs better than the other two models in this case. The object we tracked in the third example has similar size to the second one but with a rather simple background. The MFT provides the best E value and also a sharper distribution.

To demonstrate the advantages of the proposed tracking approach, we present 2 tracking examples with occlusions and complex backgrounds.



Mean shift with radial model

Single focus model (SFM)

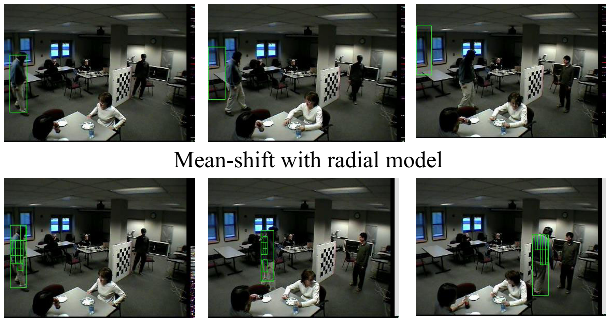
Multi-focus model (MFM)

Multi-focus template (MFT)

Figure 6 An example of a long period occlusion. The standard mean shift algorithm and MFM missed the accurate locations of the person and merged the two people together during the occlusion period. The SFM and MFT provided more robust tracking. We display the regions with non-zero confidences for the proposed models.

The first example presents the challenge of tracking a person occluded for a long period of time by another person in a corridor. Unlike a short-time occlusion caused by two objects moving across each other, the occlusion in this example is caused by a 540-frame-long, close body contact (such as hand shaking and hugging) within an 1140-frame-long interaction between the two people. The video has 30 frames per second with 360x240 pixels per frame. Motion models cannot easily predict the location of the person because he has three different motion patterns in this example: walking quickly to the other person, very small motion during the greeting, and walking back slowly back to where he began. Figure 6 presents the tracking results of using the standard mean-shift algorithm and the mean-shift algorithms with the proposed attention models (no background subtraction has been used.)

The most different results are the scales of the person in the 5th and the 6th frames. The standard mean-shift algorithm can correctly track the person at the very beginning of the encounter, but soon makes a typical error of merging the two persons as one object. After using the single focus model in the mean-shift algorithm, we obtain a more accurate location for the person during the encounter. The multi-focus model makes the same tracking mistake as the standard mean-shift algorithm. The reason is that the tracked person is small and it is difficult to tune the variance σ_m in Eq. (16). If the variance is too small, the feature does not have enough discriminative power. A large variance leads to a very flat attention distribution, and achieves a similar tracking result to the standard mean-shift. The multi-focus template does not have this difficulty with size, and obtains accurate tracking results. We display the regions with high confidences in the multi-focus results. The locations of these regions further illustrate how the high confidences shift in the occlusion.



Mean-shift with radial model

Multi-focus template (MFT)

Figure 7 Tracking in complex backgrounds using region confidences. The mean-shift algorithm lost objects because the background has similar colors.

The second example, Figure 7, shows a tracking challenge with complex backgrounds. The standard mean-shift algorithm fails to track the person because the background has similar color to the object. By using region confidences, a mean-shift algorithm improved by the MFT can track the person in this case.

The run cost of the region confidence learning is not very high compared to mean-shift tracking. It depends closely on how many regions are used in learning. As we used only 53 (35 *P* and 18 *N*) regions, we can track one object at 10 frames/second.

7. Conclusions

In this paper, we have presented an approach to exploit local region confidences for tracking objects against partial occlusions and complex backgrounds. In the proposed approach, object appearances are updated using KDE layers. We have discussed the estimation of the confidences of local regions of an object in the ideal case and proposed an online learning method to obtain the region confidences based on two factors: the discriminative power of the region in a feature space and the probability of being occluded. We have proposed a solution to model the discriminative power and region selection using matching distances between objection regions and neighborhood background. The probability of being occluded is characterized as a Gaussian on the distance between the object and other frontal layers. To make the region confidence mechanism open to state-of-the-art tracking algorithms, we have proposed three special bias models to integrate region confidence information. We have evaluated the proposed region confidence learning and special bias models using the distribution of the matching distance in the feature space. Experiments have demonstrated the robustness of the integration of the proposed approach with the mean-shift tracking algorithm in tracking people against partial occlusions over a long period of time and backgrounds with similar colors.

We have also compared the tracking performance of using three different bias models. Experiments have shown that, although some of them should theoretically perform better, in practice they may lead to worse results (for example, the MFM.) Simpler models, such as the MFT and SFM, can gain more benefit from the spatial bias tracking approach. For the results of tracking multiple objects, please see the enclosed video.

The proposed approach is easily integrated into other existing tracking methods as an improvement for occlusions and complex backgrounds. As an example, we have integrated it into a mean-shift algorithm by simply modifying the computation of the color histogram. We did not investigate temporal constraints on the translation of the spatial bias in consecutive frames, which might provide interesting knowledge to improve tracking. This will be our future effort.

8. References

[1] R. T. Collins and Y. Liu. On-line selection of discriminative tracking features. ICCV 2003.

[2] A. Galata, N. Johnson, and D. Hogg. Learning variable length Markov models of behaviour. Computer Vision and Image Understanding, 81(3):398--413, 2001

[3] W. E. L. Grimson and C. Stauffer. Adaptive background mixture models for real-time tracking, CVPR, pp. 22–29 1999.

[4] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. Proc. of the ICCV, pp. 255–261, 1999.

[5] I. Haritaoglu, D. Harwood, and L. S. Davis. W^4 - a real time system for detection and tracking people and their parts. Proc. of the FG, Nara, Japan, 1998.

[6] N. Paragios and R. Deriche. A PDE-based Level Set Approach for detection and tracking of moving objects. Technical Report 3173, INRIA Sophia Antipolis, 1997.

[7] B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J. Bouhman, Topology free hidden markov models: Application to background modeling, Proc. of the ICCV pp. 294–301, 2002.

[8] A. Elgammal, R. Duraiswami, D. Harwood, L.S. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proc. of IEEE, 7(90) pp. 1151–1163, 2002.

[9] B.D. Lucas and T. Kanade, An Iterative Image Registration Technique with an Application to Stereo Vision, Proc. of DARPA Image Understanding Workshop, pp. 121-130, 1981.

[10] C. Davatzikos, J.L. Prince, and R.N. Bryan. Image registration based on boundary mapping. IEEE Trans. on Medical Imaging, 15(1), pp.112- 115, 1996.

[11] D. Comanicu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In Proc. CVPR Vol (2), pp. 142-149, 2000.

[12] G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. Intel Tech. Journal, Q2, 1998

[13] A. Gelb. Applied Optimal Estimation, MIT Press, Cambridge, MA 1974

[14] M. Isard and A. Blake. CONDENSATION – conditional density propagation for visual tracking. IJCV Vol 29(1), pp. 5-28, 1998.

[15] W. M., Shim and P. Cavanagh. Attention shift induced by apparent motion can cause position compression. Journal of Vision, 4(8), 575a, 2004.

[16] J. Y. A. Wang and E. H. Adelson. Layered representation for motion analysis. In proc. of CVPR pp: 361-366, 1993.

[17] T. Darrel and A. Pentland. Cooperative robust estimation using layers of support. In IEEE T-PAMI 17: 474-487, 1995.

[18] H. Tao, H. S. Sawhney, R. Kumar. Object tracking with Bayesian estimation of dynamic layer representations. IEEE T-PAMI Vol. 24(1), pp: 75-89, 2002

[19] R. Stiefelhagen, J. Yang and A. Waibel. Estimating focus of attention based on gaze and sound. ACM Proc. workshop on Perceptive user interfaces, pp: 1-9, 2001.

[20] K. Toyama and G. Hage. Incremental focus of attention for robust vision-based tracking, IJCV Vol. 35:(1), pp: 45-63, 1999.