A CASCADED HIERARCHICAL FRAMEWORK FOR MOVING OBJECT DETECTION AND TRACKING

Ching-Chun Huang and Sheng-Jyh Wang
Department of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan.
E-mail: chingchun.huang3@gmail.com

ABSTRACT

In this paper we propose a cascaded hierarchical framework for object detection and tracking. We claim that, by integrating both detection and tracking into a unified framework, the detection and tracking of multiple moving objects in a complicated environment become more robust. Under the proposed architecture, detection and tracking cooperate with each other. Based on the result of moving object detection, a dynamic model is adaptively maintained for object tracking. On the other hand, the updated dynamic model is used for both temporal prior propagation of object labels and the update of foreground/background models, which step further to help the detection of moving objects. The experiments show accurate results can be obtained under situations with foreground/background appearance ambiguity, camera shaking, and object occlusion.

Index Terms- Background subtraction, Object labeling, Dynamic tracking system, and Hierarchical framework.

1. INTRODUCTION

Recently, intelligent surveillance systems are getting more and more popular. For a typical surveillance system, most cameras are kept static and several background subtraction algorithms, like [1-2], can be used to detect foreground objects. These background subtraction methods focus mainly on the modeling of background information, like the usage of the GMM model in [2] and many others. Even though this type of approach works pretty well for scenes with stationary background, it has difficulty in handling the appearance ambiguity [3] between the foreground objects and the surrounding background. Moreover, in an outdoor scene, occasional camera shaking caused by strong wind may also seriously degrade the performance of detection.

On the other hand, many object tracking algorithms focus on foreground modeling. The color-based mean-shift tracking method [4] tries to find the image patch that best matches the target model. Since the background model has not been considered, this approach suffers from the foreground/background ambiguity problem, and the tracked result may get distracted. To improve the performance, few methods try to take into account the background model. For example, in [5], the authors adopted an online training process to select discriminative foreground features with respect to the surrounding background. With this mechanism, the foreground/background ambiguity problem can be relieved. But, those methods mostly focus on the tracking of single object. In [6], Zhao et al. proposed a multi-target tracking system and handled the occlusion among objects. However, the occurrence of new comers and the disappearance of tracked objects are still big challenges to a practical tracking system.

In this paper, instead of individually performing detection and tracking, we propose a scheme to integrate both detection and tracking into a unified framework. The proposed framework adopts a temporal prediction to provide object-level prior knowledge and to continuously update the pixel-level foreground model and background model. Based on this scheme, the object labeling, foreground modeling, and background modeling are effectively fused together to better handle the foreground /background ambiguity problem. Moreover, with the estimated depth order, the inter-occlusion problem can be better solved. Also, the emergence of new comers and the disappearance of tracked objects are handled.

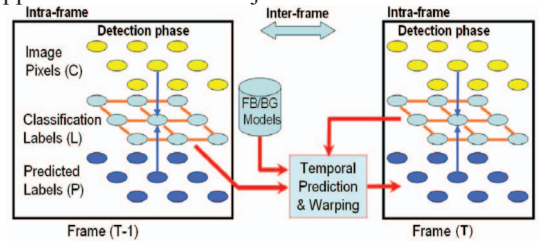

Fig. 1: Proposed scheme for object labeling and tracking.

2. PROPOSED SCHEME AND FOREGROUND/BACKGROUND LABELING

The proposed scheme is illustrated in Fig. 1. This scheme contains two major parts: the inter-frame part and the intra-frame part. The inter-frame part handles how a temporal message is propagated between successive frames; while the intra-frame part deals with object labeling. In this section, we focus on the description of the intra-frame part, which involves foreground model, background model, spatial MRF (Markov random field) constraints, and

This work was supported by NSC (97-2221-E-009-132), and Ministry of Economic Affairs (98-EC-17-A02-S1-032), Taiwan.

temporal prior message. In the next two sections, we will explain the details of the inter-frame part.

In this paper, we assume cameras are static but may suffer from slight shaking caused by winds or other factors. Hence, the background in the captured images may be trembling all the time. To handle this non-stationary background, we adopt Sheikh and Shah's approach [7] with some modifications to construct a joint spatio-chromatic probability distribution of multiple foreground/background objects based on kernel density estimation. By combining the spatial location (x,y) and the pixel color values (r,g,b) into a five-dimensional random vector $\vec{c}=(x,y,r,g,b)$, the joint spatio-chromatic probabilities are defined as

$$\begin{cases} p(\vec{c}\mid\Omega_B)=\dfrac{1}{n}\sum_{i=1}^{n}\phi(\vec{c}-\vec{c}_{Bi}) \\ p(\vec{c}\mid\Omega_F^g)=U^{-1},\ if\ g=0 \\ p(\vec{c}\mid\Omega_F^g)=\dfrac{1}{m}\sum_{i=1}^{m}\phi(\vec{c}-\vec{c}_{Fi}^{\,g}),\ if\ g=1\sim G \end{cases}, \qquad (1)$$

where Ω_B and Ω_F^g denote the background and the g^{th} foreground, respectively. Ω_F^0 is especially designed for new comers. In Eq. (1), $\phi(.)$ is a symmetric and normalized kernel function, $\vec{c}_{Bi}$ denotes one of the n background samples, $\vec{c}_{Fi}^{\,g}$ denotes one of the m foreground samples of the g^{th} target, G is the number of foreground objects in the current image, and U^{-1} describes a uniform distribution over the five-dimensional domain. Based on the above definition, the spatial uncertainty caused by camera shaking and the chromatic uncertainty caused by lighting change can be properly modeled.

In our approach, object detection is treated as a classification problem. Besides background model $p(\vec{c}\mid\Omega_B)$ and foreground models $p(\vec{c}\mid\Omega_F^g)$, we also take into account current observation, spatial smooth constraint, and temporal prior knowledge. As shown in Fig. 1, we adopt a 3-layer structure at each time instant. The top layer $\boldsymbol{C}$ represents the observation layer at that time instant. In our approach, we assume $\boldsymbol{C}$ contains the spatio-chromatic information of the observed image data. The middle layer $\boldsymbol{L}$ contains the classification label for each image pixel. In principle, we aim to assign to each labeling node L_i a suitable ID from the set $\{\Omega_B,\Omega_F^0,\ \ldots,\ \Omega_F^G\}$. The bottom layer $\boldsymbol{P}$ represents the predicted label messages propagated from the previous time instant. To find out a suitable classification label $\boldsymbol{L}$ under the given image observation $\boldsymbol{C}$ and the predicted labels $\boldsymbol{P}$, we solve the following MAP optimization problem:

$$\begin{aligned} L^* &= \arg\max_L p(L\mid C,P)=\arg\max_L p(C\mid L)p(L,P) \\ &= \arg\max_L[\ln(p(C\mid L))+\ln(p(L\mid P))+\ln(p(P))], \end{aligned} \qquad (2)$$

where $p(C|L)$ is the likelihood terms and $p(L|P)$ denotes the label messages form temporal prior. Since $p(P)$ is a constant, it can be ignored. In our approach, once if $\boldsymbol{L}$ is given, we assume the conditional probability density function of the observation data at two different pixels are independent of each other. We also assume the data $\vec{c}_i$ at Pixel i does not depend on the labels at other pixels. With these two assumptions, we define

$$p(C\mid L)=\prod_{i=1}^{K} e^{-E_D[\vec{c}_i,L_i]}e^{-E_A[\vec{c}_i,L_i;N_p]}, \qquad (3)$$

where K is the total number of image pixels. $E_D[\vec{c}_i,L_i]$ is the "classification energy" for the labeling node L_i and the feature data $\vec{c}_i$ at the i^{th} pixel. Here, we define $E_D[\vec{c}_i,L_i]$ as

$$E_D[\vec{c}_i,L_i]=\begin{cases} -\ln(p(\vec{c}_i\mid\Omega_B)) & \text{if } L_i=\Omega_B \\ -\ln(p(\vec{c}_i\mid\Omega_F^g)) & \text{if } L_i=\Omega_F^g \end{cases}. \qquad (4)$$

On the other hand, we define the "adjacency energy" $E_A[\vec{c}_i,L_i;N_i]$ based on a 4-neighbor MRF model [8], where N_i denotes the connectivity neighborhood of Pixel i. That is,

$$E_A[\vec{c}_i,L_i;N_i]\equiv\sum_{j\in N_i}(\beta\times(1-\delta[L_i,L_j])/(\|\vec{c}_i-\vec{c}_j\|+\alpha)), \qquad (5)$$

where β is a normalized constant, α is a small constant to avoid division by zero, and $\delta(\cdot)$ is defined as

$$\delta[p,q]=\begin{cases} 1 & \text{if } p=q \\ 0 & \text{otherwise} \end{cases}. \qquad (6)$$

In principle, $E_A[\vec{c}_i,L_i;N_i]$ denotes the spatial correlation between pairs of classification labels (L_i,L_j). This energy softly forces neighboring pixels to share the same label, especially when they have similar spatio-chromatic features.

In addition, $p(L|P)$ represents the expected labeling map based on the previous prediction. Here, we assume the predicted image location of each foreground object at the current instant t could be modeled as probability $p_g(x,y;t)$; the g^{th} object at time instant t-1 is bounded by a compact rectangular box $RB_{g;t-1}$ around the g^{th} object. The extraction of $p_g(x,y;t)$ and $RB_{g;t-1}$ are to be explained later. To model $p(L|P)$, we adopt the Monte Carlo based method to draw many expected labeling samples and approximate $p(L|P)$ in a sample-based manner. To generate a labeling sample, we draw a location sample $(x_s,y_s)_g$ from $p_g(x,y;t)$ for each object, and warp the center of $RB_{g;t-1}$ to $(x_s,y_s)_g$. While the rectangular boxes get overlapped, inter-occlusion is expected to occur and the depth order is needed to determine the occlusion pattern. Here, we adopt the Bhattacharyya coefficient (BC) based metric [4] to determine the depth order. If a predicted target region is more similar to its target model in appearance, that target has a higher possibility to be the object that occludes the others. In detail, for a target g, we measure the Bhattacharyya coefficient at location $(x_s,y_s)_g$ as

$$\rho_{x_s,y_s}(g)=\int\sqrt{h_{x_s,y_s}(z;g)p(z;g)}dz, \qquad (7)$$

where $h_{x_s,y_s}(z;g)$ is the normalized color histogram of the image region inside the warped $RB_{g;t-1}$ centered at $(x_s,y_s)_g$, $p(z;g)$ is the normalized color model of target g, and z

denotes a possible (r,g,b) color feature. Here, $p(z;g)$ is derived from the foreground model of the target g based on

$$p(z;g) = \iint p(\vec{c}\,|\,\Omega_F^g)dxdy\,. \qquad (8)$$

By comparing the BC values among inter-occluded objects, the depth order is determined and an expected labeling sample is generated. By accumulating the occurrence number of different labeling IDs at each pixel from many expected labeling samples, we can model $p(L|P)$ to well handle occlusion. In Fig. 2 we show an example of $p(L|P)$.

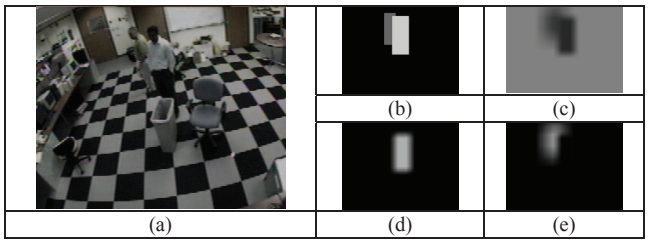

Figure 2. (a) Test image. (b)An expected labeling sample of (a). (c) Estimated p(L|P) for $L=\Omega_B$ or Ω_F^0. (d) Estimated p(L|P) for $L=\Omega_F^1$. (e) Estimated p(L|P) for $L=\Omega_F^2$.

Based on Eq. (1)~(8), we form the formulae for MAP optimization. We adopt the Graph Cuts method [8] to find the optimal label $\boldsymbol{L}^*$ that maximizes Eq. (2). Based on the classified labels, we detect foreground objects. Moreover, for each non-occluded foreground object, the rectangular box $RB_{g;t}$ at the current time t is estimated from the vertical and horizontal projection histograms of its foreground region. For both vertical and horizontal directions of $RB_{g;t}$, we search for the minimum continuous-valued ranges that can cover 95% energy of the projection histograms. For occluded objects, the size of rectangular box remains the same value at the previous time instant but the center of the box is shifted to the new object center. Besides, we also identify the new comers by evaluating the vertical projection histogram of the foreground region with the ID Ω_F^0.

3. OBJECT TRACKING

As mentioned above, the predicted temporal prior $\boldsymbol{P}$ at the current frame is warped from the classification results $\boldsymbol{L}^*$ at the previous frame. To provide the temporal message and to model the inter-frame relation, a dynamic tracking model is maintained for each foreground object. Moreover, since there could be some errors in the prediction of foreground movement, the result of classification labeling is fed back to update the dynamic models of foreground objects. Under the proposed architecture, object tracking is actually treated as the temporal prediction and update of object labels.

To design a tracker for each foreground object, the Bayesian-based filters are widely used. In this work, we adopt the Kalman filter for the sake of computational complexity. Here, we define $\boldsymbol{S_t}=(\boldsymbol{x_t},\boldsymbol{v_t})$ as our motion state, including object center $\boldsymbol{x_t}=(x,y)$ and object velocity $\boldsymbol{v_t}=(v_x,v_y)$. Based on the Kalman filter updating rule, $F(.)$, for each object, the optimal estimation of object motion state $\boldsymbol{S_t}$ is determined by

$$S_{t|t} = F(S_{t|t-1}, K_t, z_t), \qquad (9)$$

where $\boldsymbol{S_{t|t-1}}$ is the optimal prediction of $\boldsymbol{S_t}$ based on its previous motion state $\boldsymbol{S_{t-1|t-1}}$; $\boldsymbol{z_t}$ is the observed object center determined by the object detection in Eq. (2); K_t is the Kalman gain. With the Kalman filter, the probability of the predicted location of g^{th} object $p_g(x,y;t)$ is modeled by a Gaussian distribution $N(\boldsymbol{x_{t|t-1}},\boldsymbol{Q_k})$, with the covariance of the noise process $\boldsymbol{Q_k}$. Due to the limit of space, the detail of the Kalman filter is not stated here.

To explain the interaction of object detection and tracking, we assume the classification label L^* at time instant t-1 has been determined. Based on the classified label L^*, a few foreground objects are detected. For each foreground object, we calculate its $RB_{g;t-1}$ and measure its object-mass-center (OMC_{t-1}) as the observation data $\boldsymbol{z_{t-1}}$. At the current time instant t, we track the location of each foreground object based on its OMC_{t-1} and $RB_{g;t-1}$.This object tracking process consists of the following 4 major steps. An illustration of this object tracking process is shown in Fig. 3.

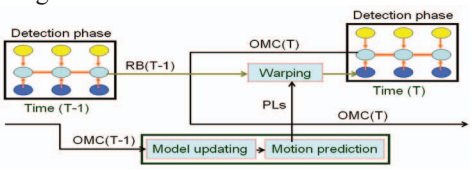

Fig.3: Illustration of the tracking process.

Step1: Creation/Update/Deletion of Tracking Model

For new comers, their Kalman trackers are created. Next, for each foreground object, its OMC_{t-1} is used to update the tracking model. Based on the updated model, we draw 255 predicted locations PL's from $p_g(x,y;t)$. For objects having no motion for a long enough period or moving out the scene, their tracking models are deleted.

Step2: Temporal Propagation of Foreground Labels

Based on the PL's, the $RB_{g;t-1}$'s at time t-1 are warped to their new location at time t to construct the expected labeling map $p(L|P)$ at time t.

Step3: Update of object models

Based on classification label L^* at time t-1, we update both foreground and background classification models. Moreover, we predict the location and appearance of each foreground object and update its foreground model before detection. The detail is described in Section 4.

Step4: Foreground/Background Labeling

At time t, we deduce the optimal classification label $\boldsymbol{L}^*$ based on the optimization of Eq. (2). From the optimal $\boldsymbol{L}^*$, we detect a set of foreground objects at the current time t.

4. UPDATE OF OBJECT MODELS

To adapt to a varying environment, the foreground model and background model in Eq. (1) should be updated all the time. Traditionally, model updating is performed after the

detection stage. This makes it very difficult to handle the foreground/background ambiguity problem. On the contrary, we update the foreground model ***before*** we perform object labeling. That is, if the foreground object is currently at $\boldsymbol{x}_{t|t}$ and we predict the optimal location of this object will move to $\boldsymbol{x}_{t+1|t}$, we adjust the foreground model accordingly so that $p(\vec{c} \mid \Omega_F^g)$ will be high around both $\boldsymbol{x}_{t|t}$ and $\boldsymbol{x}_{t+1|t}$. With this mechanism, if the foreground object happens to move into some background region with a similar appearance, both $p(\vec{c} \mid \Omega_B)$ and $p(\vec{c} \mid \Omega_F^g)$ will be high within the ambiguous region. The update of foreground model will reduce the probability that a foreground region being mistakenly classified as a background region. Moreover, since the prediction layer $\boldsymbol{P}$ also provides useful prior knowledge about the predicted location of foreground objects, the foreground/background ambiguity problem can be more effectively solved.

On the other hand, we update the background model $p(\vec{c} \mid \Omega_B)$ based on the result of foreground/background labeling. In our approach, only those pixels labeled as background pixels will be considered in the update of background model. Occasionally, a foreground object may become a part of the background, like the situation that a car parks in the scene for a long time. For this kind of situation, we may simply check whether the foreground object has been motionless for a long enough period. If so, the features of the foreground object can be added into the background model.

5. EXPERIMENTS RESULTS

We test our system over the IBM datasets [9], OVVV datasets, and our own datasets. We also do comparison with the GMM method [2], as shown in Fig 4. In Fig 4(b,c), due to the appearance ambiguity between foreground object and background, the GMM method generates fragmented results. Instead, our method well adopts the object prior from temporal and can still robustly detects the whole foreground object. In Fig 4(a,b), our labeling results clearly identify the inter-object occlusion and the depth order. Moreover, in Fig 4(d), owing to camera shaking, the GMM method generates lots of false detections. With the use of the kernel function in Eq. (1), the proposed method generates reliable detection result. Besides, our system can detect the new comers or the vanishing objects automatically. In fact, an object is leaving in Fig 4(a). To quantitatively evaluate our system, we use the ground truth and the metrics proposed by IBM [9]. The evaluations are listed in Table 1. Currently, the whole system is implemented in Visual C++ on a PC with a 2.4 GHz CPU. It takes about 1 second to perform the detection and tracking for a 320x240 color image frame. For more experimental results, please visit our website at http://140.113.238.220/~chingchun/projects.html. In the future, we plan to move our system to a GPU based platform.

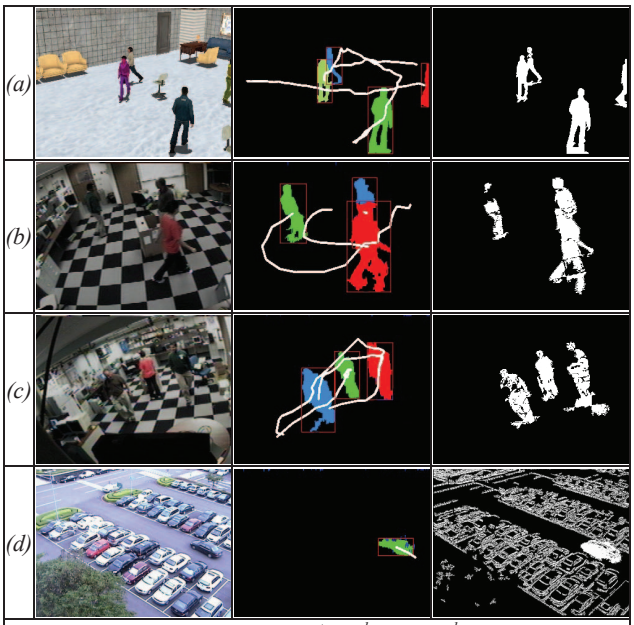

Fig. 4. Experimental results. 1st, 2nd, and 3rd columns are the tested sequences, our results, and results of [2], respectively. (a) OVVV dataset. (b)(c) IBM dataset. (d) Our outdoor dataset. Tracking results are shown as white curves in the 2nd column.

Table. 1. Evaluation of 5 tested sequences. (a)(b)(c)IBM "Line_Circle", "Splite", and "Circle" sequences. (d)(e)Two OVVV sequences. The adopted IBM metrics are frames number (FraN), true positive (TP), false positive (FP), false negative (FN), Track TP (TTP), Track FP (TFP), and Track FN (TFN) [9].

Seqs	FraN	TP	FP	FN	TTP	TFP	TFN
(a)	415	377	4 / 415	8 / 377	2	0	0
(b)	352	372	8 / 352	12 / 372	3	0	0
(c)	371	657	17 / 371	11 / 657	3	0	0
(d)	300	750	1 / 300	0 / 750	3	0	0
(e)	1000	2746	0 / 1000	32 / 2746	17	0	1

6. REFERENCES

[1] Til Aach, Lutz Dümbgen, Rudolf Mester, Daniel Toth, "Bayesian Illumination-invariant Motion Detection," *ICIP*, 2001.

[2] P. Power and J. A. Schoonees, "Understanding Background Mixture Models for Foreground Segmentation," *Image and Vision Computing*, 2002.

[3] B. Bose, X. Wang, E. Grimson, "Multi-class Object Tracking Algorithm that Handles Fragmentation and Grouping," *IEEE Conf. CVPR*, 2007.

[4] D. Comaniciu, V. Ramesh, P. Meer, "Real-time Tracking of Non-rigid Objects Using Mean Shift," *IEEE Conf. CVPR*, 2000.

[5] R.T. Collins, Y. Liu, and M. Leordeanu, "Online Selection of Discriminative Tracking Features," *IEEE Trans. PAMI,* 2005.

[6] T. Zhao, R. Nevatia, B. Wu, "Segmentation and Tracking of Multiple Humans in Crowded Environments," *IEEE PAMI*, 2008.

[7] Y. Sheikh, M. Shah, "Bayesian Modeling of Dynamic Scenes for Object Detection," *IEEE Trans. PAMI*, 2005.

[8] T. Boykov, O. Veksler, R. Zabih, "Markov Random Fields with Efficient Approximations," *IEEE Conf. CVPR,* 1998.

[9] H. Merkl and M. Lu, "Performance evaluation of surveillance systems under varying conditions," *IEEE PETs Workshop*, 2005.