Learning a Scene Background Model via Classification

Horng-Horng Lin, *Student Member, IEEE*, Tyng-Luh Liu, *Member, IEEE*, and Jen-Hui Chuang, *Senior Member, IEEE*

***Abstract*—Learning to efficiently construct a scene background model is crucial for tracking techniques relying on background subtraction. Our proposed method is motivated by criteria leading to what a general and reasonable background model should be, and realized by a practical classification technique. Specifically, we consider a two-level approximation scheme that elegantly combines the bottom-up and top-down information for deriving a background model in real time. The key idea of our approach is simple but effective: If a classifier can be used to determine which image blocks are part of the background, its outcomes can help to carry out appropriate blockwise updates in learning such a model. The quality of the solution is further improved by global validations of the local updates to maintain the interblock consistency. And a complete background model can then be obtained based on a measurement of model completion. To demonstrate the effectiveness of our method, various experimental results and comparisons are included.**

***Index Terms*—Background modeling, boosting, classification, tracking, SVM.**

I. Introduction

VISUAL tracking systems using background subtraction often work by comparing the upcoming image frame with an estimated *background model* to differentiate moving foreground objects from the scene background. Hence, the performance of such systems depends heavily on how the background information is modeled initially, and maintained thereafter. In this work, we aim to establish a learning approach to reliably estimate a background model even when substantial object movements are present during the initialization stage. As illustrated in Fig. 1, the overall idea is to efficiently *identify* background blocks from each image frame through online classifications, and to iteratively *integrate* these background blocks into a complete model so that a tracking process can be automatically initiated in real time. In developing such a progressive processing scheme for initializing a background model, some criteria are considered.

- *Stationary scene adaptation*: It is commonly agreed that stationary scenes are considered as background. Thus, in our design, when a moving object becomes stationary over a certain period of time, it will be incorporated into a background model. This would yield an initial background model accommodating the most recent statistics about the background scene, e.g., a parking car or an occluded area.
- *Gradual variation adaptation*: The computation of a background model should take account of small variations caused by, e.g., gradual illumination changes, waving trees, and faint shadows. It allows a system to reduce the false detection rate of foreground objects.
- *Model completion*: Depending on object movements, the number of image frames needed in estimating an initial background could vary significantly. Hence, a measurement for the availability of a background model has to be defined so that the system can immediately begin to track objects upon the completion of model initialization.
- *Efficiency*: A background model must give rise to efficient online derivations to guarantee real-time tracking performance.

The first two criteria listed above manifest what kind of scene contents are considered as background. The last two ones illustrate the design requirements of a background model initialization system: it should be capable of deriving a complete background model in a progressive manner and in real time.

In the proposed approach, two features will be observed. First, we utilize learning methods to identify background blocks. Rather than developing discrimination rules or models, we adopt learning approaches to construct a background block classifier. This strategy not only provides a convenient way of defining some preferred background types from image examples, but also avoids complicated issues of manually setting discriminating parameters, because they can be resolved by learning from the chosen data. Second, the derived background model fulfills the four criteria. To achieve efficiency, a progressive estimation scheme is developed and a fast classifier adopted. For the model completion criterion, an effective definition is given to indicate that a complete background model is obtained, and the subsequent tracking procedures can be started. Regarding the adaptation criteria, we implement a bottom-up block updating, in either a gradual or an abrupt fashion, for capturing the background variations and scene changes, respectively.

A. Related Work

Background modeling for tracking typically involves three issues: *representation, initialization*, and *maintenance*. For example, one could *represent* a scene background by assuming a single Gaussian distribution for each pixel, *initialize* the model by estimating from an image sequence, and *maintain* it during

Manuscript received February 24, 2008; accepted December 01, 2008. First published February 06, 2009; current version published April 15, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Marcelo G. S. Bruno. This work was supported in part by NSC 95-2221-E-001-031-MY3 and NSC 95-2221-E-009-269.

H.-H. Lin and J.-H. Chuang are with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: hhlin@cs.nctu.edu.tw; jchuang@cs.nctu.edu.tw).

T.-L. Liu is with the Institute of Information Science, Academia Sinica, Nankang, Taipei, 115, Taiwan (e-mail: liutyng@iis.sinica.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2009.2014810

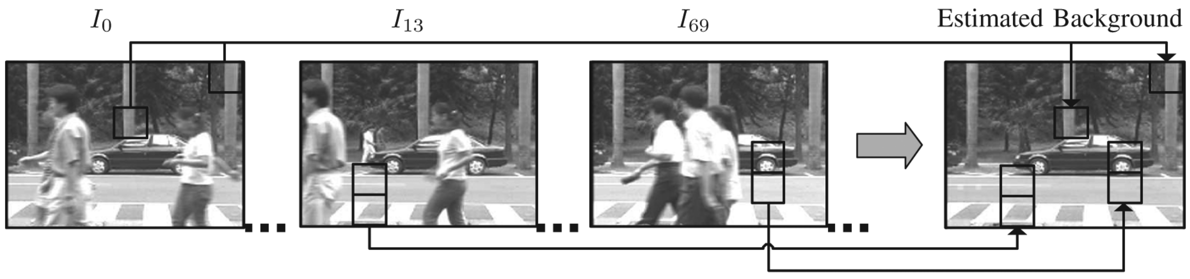


Fig. 1. Through performing online classifications and by iteratively integrating the framewise detected background blocks of images captured with a static monocular camera, the scene background can be reliably estimated in real time.

tracking by updating Gaussian parameters of the background pixels. While the emphases of most previous works, including those to be described later, are mainly on representation and maintenance, the task to compute an initial background model has been somewhat neglected or otherwise simplified by not allowing large object movements throughout the initialization process, e.g., [10], [25], and [38].

1) Background Representation and Maintenance: Gaussian models are perhaps the most popular representation for modeling a scene background, e.g., [4], [23], [26], [33], and [38]. Their maintenance is usually carried out in the form of temporal blending to update intensity means and variances. Thus, related researches often differ in the number of Gaussian distributions used for each pixel, and the update formulas for the Gaussian parameters. In [15], Gao *et al.*, further investigate possible errors caused by Gaussian mixture models, and then apply statistical analysis to estimate related parameters.

Apart from Gaussian assumptions, Elgammal *et al.* [9] consider *kernel smoothing* for a non-parametric estimate of pixel intensity over time. In [35], Toyama *et al.* propose a *wallflower* algorithm to address the problem of background representation and maintenance in three levels: pixel, region, and frame levels. Ridder *et al.* [30] use a *Kalman-filter* estimator to identify the respective pixel intensities of foreground and background from an image sequence, and to suppress false foreground pixels caused by shadow borders. In [19], a mixture of local histograms is proposed to construct a texture-based background model that is more robust to background variations, e.g., illumination changes.

Prior assumptions about the foreground, background, and shadows can be used to simplify the modeling complexity. For vehicle tracking, Friedman and Russell [14] propose three kinds of color models to classify pixels into road, shadow, and vehicle. They employ an incremental EM to learn a *mixture-of-Gaussian* for distinguishing the foreground and background. In [31] and [36], prior knowledge at pixel level is considered in learning the model parameters of the foreground and background. Then, a high-level process based on *Markov random field* is performed to integrate the information from all pixels.

2) Background Model Initialization: The most straightforward way to estimate a background model is to calculate the intensity mean of each pixel through an image sequence. Apparently, this is rarely appropriate for practical uses. Haritaoglu *et al.* [17], instead, compute intensity medians over time. Yet a more general framework by Stauffer and Grimson [33] is to use pixelwise *Gaussian mixtures* to model a scene background. Mittal and Huttenlocher [26] later extend the Gaussian mixture idea to construct a mosaic background model from images captured using a nonstationary camera. In [35], *bootstrapping* for background initialization is proposed, and implemented with a pixel-level *Wiener filtering*.

Among the above-mentioned approaches, initializing a background model is viewed more or less as part of the process for background maintenance. They do not have a systematic way to measure the quality, and determine the degree of completion for such a model. Consequently, these methods often require *simple* initializations, or otherwise start tracking activities with unreliable background models.

For computing an explicit background model, Gutchess *et al.* [16] use optical flow information to choose the most likely time interval of stable intensity at each pixel. However, the quality of their derived background model depends critically on the accuracy of the pixelwise optical flow estimations. Cucchiara *et al.* [6] represent a background model by pixel medians of image samples, and specifically identify moving objects, shadows, and ghosts[1] for different model updates using color and motion cues. Based on the Gaussian mixture model, Hayman and Eklundh [18] formulate a statistical scheme to derive a mosaic background model with an active camera. They consider a *mixel distribution* to correct the errors in background registration. In [7], De la Torre and Black apply *principal component analysis* (PCA) to construct the scene background from an image sequence. More recently, Monnet *et al.* [27] propose an *incremental* PCA to progressively estimate a background model and detect foreground changes. Still, these systems all lack an explicit criterion for determining whether a background initialization is completed or not—a crucial and practical element for a real-time tracking system.

Other techniques that explore layer decompositions of a video sequence can also be used to estimate a background model. Irani and Peleg [20] explore the decompositions of dominant motions and apply them to the construction of an unoccluded background image. In [12] and [21], sprite layers are derived from probabilistic mixture models, in which cues of layer appearances and motions are encoded. In [1] and [2], Aguiar and Moura consider rigid motions, intensity differences, and the region rigidity for figure–ground separation and formulate them as

[1]Ghosts are false foreground objects detected by subtracting an inaccurate background model from image frames.

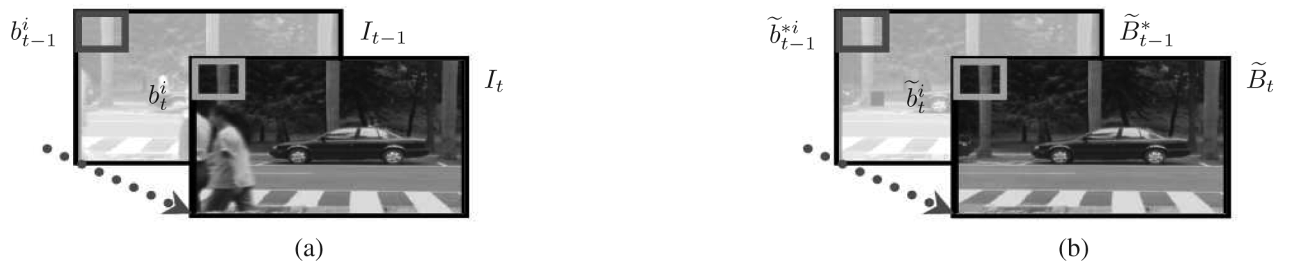


Fig. 2. (a) Online image stream: I_t and I_{t-1} are the image frames at time t and $t-1$, and their ith blocks are denoted as b^i_t and b^i_{t-1}, respectively. (b) Estimated background model: For the background models, $\tilde{B}_t$ is a possible estimation at time t, while $\tilde{B}^*_{t-1}$ is the best estimation up to time $t-1$. Accordingly, their ith blocks are represented by $\tilde{b}^i_t$ and $\tilde{b}^{*i}_{t-1}$ (a) Online image stream and (b) Estimated background model.

a penalized likelihood model that can be optimized in efficient ways. In [5], [22], and [37], graph-cut-based techniques, e.g., [3], are applied to decompose video layers via pixel labeling, with various objective functions being optimized. Though all the layer-based approaches are capable of deriving a background model even for dynamic scenes, they often need to process a video sequence in batch, which is different from the proposed progressive scheme.

The rest of the paper is organized as follows. In Section II, the proposed background estimation approach is presented. In particular, the relationship between the classification and the estimation scheme is elaborated. Then the adopted classifiers are introduced in Section III. In Section IV, some experiments are demonstrated, including training results, background estimation performance, and comparisons to other approaches. Finally, a brief discussion is given in the same section to conclude this work.

II. Background Model Estimation via Classification

Due to the restriction of limited memory space and the requirement of real-time performance, only a small number of recent image frames are stored and referred during the construction of a background model. Thus, an iterative estimation scheme is proposed in the following to progressively identify background blocks in image frames and to incorporate their information into a background model.

A. *Iterative Estimation Scheme*

To illustrate the idea of the proposed iterative estimation scheme, we begin by summarizing the notations and definitions adopted in our discussion.

- We denote the test image sequence up to time instant t as $\mathbf{I}_t = \{I_1, I_2, \ldots, I_t\}$, and the most recent ℓ image frames as $\mathbf{I}_{t,\ell} = \{I_{t-\ell+1}, \ldots, I_{t-1}, I_t\}$. We also use b^i_t to stand for the ith block of I_t, and $\mathbf{b}^i_{t,\ell} = \{b^i_{t-\ell+1}, \ldots, b^i_{t-1}, b^i_t\}$ for the set of ith blocks from $\mathbf{I}_{t,\ell}$ (see Fig. 2).
- Let $\tilde{B}_t$ be any possible background model estimation at time t, and $\tilde{B}^*_{t-1}$ be the estimated background model at time $t-1$. Then, the ith blocks of $\tilde{B}_t$ and $\tilde{B}^*_{t-1}$ are denoted as $\tilde{b}^i_t$ and $\tilde{b}^{*i}_{t-1}$, respectively.
- A training set of m samples, $\mathbf{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$, is used to build a binary classifier, where each $\mathbf{x}_i$ is a fixed-size image block (or simply the extracted feature vector), and $y_i \in \{-1(\text{foreground}), +1(\text{background})\}$ is its label.
- With training data $\mathbf{D}$, an *optimal* classifier f^* can be defined as

$$f^* = \operatorname*{argmax}_f p(f \,|\, \mathbf{D}). \tag{1}$$

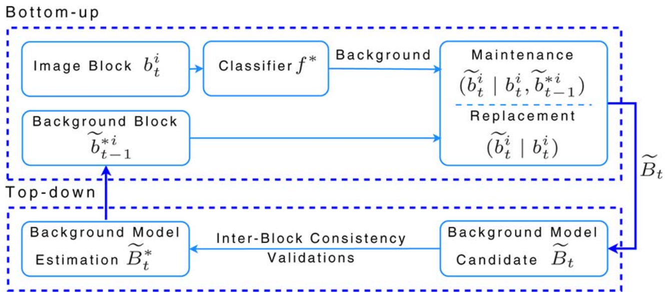

Bottom-up
Image Block b^i_t
Classifier f^*
Background
Maintenance
$(\widetilde{b}^i_t \mid b^i_t, \widetilde{b}^{*i}_{t-1})$
Replacement
$(\widetilde{b}^i_t \mid b^i_t)$
Background Block
$\widetilde{b}^{*i}_{t-1}$
$\widetilde{B}_t$
Top-down
Background Model Estimation $\widetilde{B}^*_t$
Inter-Block Consistency Validations
Background Model Candidate $\widetilde{B}_t$

Fig. 3. Flowchart depicts the interactions between the bottom-up block updating and the top-down model validation processes. While the bottom-up process handles blockwise updates of the background, the top-down one deals with interblock consistency validations. The coupling of the two processes forms an efficient scheme for deriving a background model.

Equation (1) manifests that a classifier f^* can be derived from a probabilistic *maximum a posteriori* (MAP) treatment [32]. It is thus more desirable to have not only classification labels/ scores but also probabilistic outputs of f^*. In Section III, we will explain that either an SVM or a boosting-with-soft-margins classifier is appropriate for delivering such probabilities. With probabilistic outputs, a threshold can then be set to adjust the classification boundary, which is useful for our background estimation. We will demonstrate this usage in Section IV-A-3).

The proposed iterative estimation scheme for deriving a background model consists of a bottom-up block updating and a top-down model validation process. As shown in Fig. 3, a flowchart is given to illustrate the interactions between the two processes. The aim of the bottom-up process is to blockwise integrate identified background blocks into a model and to form a model candidate $\tilde{B}_t$. Then, in the top-down process, the interblock consistency for all the updated background blocks are validated. By assuming that significant background updates often occur in groups, isolated updates that mainly result from noises will be eliminated by restoring their block statistics back to the previous estimates $\tilde{b}^{*i}_{t-1}$.

More specifically, in the bottom-up process, the image block b_t^i classified as background and the previously estimated background block $\tilde{b}_{t-1}^{*i}$ act as two inputs to the background adaptation. Based on a dissimilarity measure between the current image block b_t^i and the previous background block $\tilde{b}_{t-1}^{*i}$, either a maintenance step or a replacement step is invoked for a block update. In the maintenance step, the case of the small block difference is handled, assuming it is mostly caused by gradual lighting variations or small vibrations. A new background block estimate $\tilde{b}_t^i$ can thus be computed by a weighted average of the two blocks b_t^i and $\tilde{b}_{t-1}^{*i}$. On the other hand, when b_t^i and $\tilde{b}_t^i$ are dissimilar, implying an occurrence of an abrupt scene change, a replacement step is employed to calculate a renewed background block estimate $\tilde{b}_t^i$, which is consistent with the image block b_t^i.

After the above bottom-up updates, a background model candidate $\tilde{B}_t$ is obtained. To turn this model candidate into a final estimate, a top-down process is introduced to assure the model consistency between the current candidate $\tilde{B}_t$ and the previous estimate $\tilde{B}_{t-1}^*$, by assuming a smooth changing in background models. Although the checking of model consistency can be realized in various ways, we choose to implement it in a simple manner by finding the updates of isolated blocks and undoing them. Thus, large and grouped background block updates are preserved in this design, since they most likely belong to significant and stable background changes, such as newly uncovered scenes or stationary objects. Through the validation process, a final background model estimation $\tilde{B}_t^*$ is derived.

It is worth mentioning that the entire approach is somehow linked to a MAP formulation, i.e.,

$$\tilde{B}_t^* = \underset{\tilde{B}_t}{\operatorname{argmax}} \left\{ \underbrace{\left(\prod_{i^+} P\left(\tilde{b}_t^i \,|\, b_t^i, \tilde{b}_{t-1}^{*i}\right) \prod_{i^-} P\left(\tilde{b}_t^i \,|\, \tilde{b}_{t-1}^{*i}\right) \right)}_{\text{Likelihood}} \times \underbrace{P(\tilde{B}_t \,|\, \tilde{B}_{t-1}^*)}_{\text{Prior}} \right\} \quad (2)$$

where $i^+ = \{i \,|\, b_t^i$ is classified as a background block by $f^*\}$ and $i^- = \{1, \ldots, n\} - i^+$. (Assume there are n blocks in an image frame.) Interested readers can find the derivation of (2) in the Appendix. The connections between (2) and our approach are elaborated as follows. Regarding the *likelihood* part, the two products can be viewed as blockwise updates after the background classification. For the image block classified as background, maximizing the probability $P(\tilde{b}_t^i \,|\, b_t^i, \tilde{b}_{t-1}^{*i})$ implies that similarities among the background estimate $\tilde{b}_t^i$, the image block b_t^i and the previous estimate $\tilde{b}_{t-1}^{*i}$ should be retained. Likewise, for a foreground block, the corresponding probability $P(\tilde{b}_t^i \,|\, \tilde{b}_{t-1}^{*i})$ is maximized by setting the current block estimate $\tilde{b}_t^i$ equal to the previous one $\tilde{b}_{t-1}^{*i}$. This is what we do in the bottom-up process. Referring to the *prior* term, it indicates that model level consistency between $\tilde{B}_t$ and $\tilde{B}_{t-1}^*$ needs to be maintained for maximizing the probability $P(\tilde{B}_t \,|\, \tilde{B}_{t-1}^*)$. This, as well, corresponds to the top-down model validation. However, we note that the background model $\tilde{B}_t^*$ derived by our approach is only a rough *approximation* to the MAP solution, since (2) is not exactly solved. In fact, to optimize (2), the underlying distributions of the probability terms should be further specified, and complicated optimization techniques, e.g., EM-based estimations, may need to be employed. Hence, instead of pursuing the MAP solution, our focus is on the design of a practical and efficient algorithm for background model estimation.

B. The Algorithm

1) Bottom-Up Process: We start by applying f^* to each b_t^i to determine its probability of being a background block. A simplified notation $P(b_t^i \,|\, f^*)$ will be hereafter adopted to denote such a probability, with the understanding that the most recent ℓ ith-blocks b^is (i.e., $\mathbf{b}_{t,\ell}^i$) are available for calculating useful features, e.g., optical flow values, for classification. Observe that only for those image blocks classified as background at each time t, their corresponding blockwise updatings would modify the background model. It is therefore preferable to have as few false positives by f^* as possible. Hence, we use a strict thresholding τ^*, i.e., the decision boundary of f^*, on $P(b_t^i \,|\, f^*)$ such that image blocks with $P(b_t^i \,|\, f^*) > \tau^* \geq 0.5$ are considered background. Given this setting, there are two possible cases for a block updating.

1) If b_t^i is not a background block, then $\tilde{b}_t^{*i} = \tilde{b}_{t-1}^{*i}$, i.e., the pixel means and variances of $\tilde{b}_{t-1}^{*i}$ are assigned to $\tilde{b}_t^{*i}$.
2) If b_t^i is classified as a background block, we measure the dissimilarity between b_t^i and $\tilde{b}_{t-1}^{*i}$ by

$$\operatorname{diss}\left(b_t^i, \tilde{b}_{t-1}^{*i}\right) = \frac{\left\| b_t^i - \tilde{b}_{t-1}^{*i} \right\|^2}{|b^i|}$$

where $\|b_t^i - \tilde{b}_{t-1}^{*i}\|^2$ is the sum of squared pixel intensity differences, and $|b^i|$ is the block size. Depending on the value of $\operatorname{diss}(b_t^i, \tilde{b}_{t-1}^{*i})$, either a *maintenance step* or a *replacement step* is invoked (see Algorithm 1). We apply the iterative maintenance formulas proposed in [4] to update the latest small variations into $\tilde{B}_t^*$. Notice that a block replacement in evaluating $\tilde{B}_t^*$ takes place only when the particular block has been classified as background for N consecutive frames. Indeed, the maintenance phase is designed to adapt the gradual variations, and the replacement phase is to accommodate new stationary objects.

2) Top-Down Process: A top-down process based on comparing $\tilde{B}_t$ with $\tilde{B}_{t-1}^*$ is employed to detect isolated block updates in the bottom-up evaluation of $\tilde{B}_t$, and undo these updates with the statistical data from $\tilde{B}_{t-1}^*$.[2] Conveniently, in implementing the algorithm, the top-down process can be carried out right after the background block classifications. This would yield a set of *valid* background blocks; all of them are not isolated. Hence, the bottom-up updatings over these valid blocks would directly lead to the final estimate $\tilde{B}_t^*$.

[2]An isolated block updating (either for maintenance or for replacement) has less than three of its 4-connected neighboring blocks being updated in the bottom-up process.

Algorithm 1: Constructing the $\widetilde{B}_t^*$.

```
Data: Process $I_t$ using $f^*$, $\widetilde{B}_{t-1}^*$ and an auxiliary image
      $\bar{B} = \{\bar{b}^i\}$. When $t = 0$, we have $\widetilde{B}_0^* = \emptyset$, $\bar{B} = \emptyset$, and
      $\forall i$, $age(i) = 0$, $counter(i) = 0$, and
      $replace(i) = false$.
Result: Obtain a MAP estimate $\widetilde{B}_t^*$.
begin
    $\widetilde{B}_t^* \longleftarrow \widetilde{B}_{t-1}^*$
    for image block $b_t^i \in I_t$ do
        if $b_t^i$ is a valid background block then
            if $diss(b_t^i, \widetilde{b}_{t-1}^{*i}) \leq \delta(= 15^2 = 225)$ then
                /* Maintenance */
                $\widetilde{b}_t^{*i} \longleftarrow$ IterativeAverage($b_t^i, \widetilde{b}_{t-1}^{*i}$)
                $age(i) \longleftarrow age(i) + 1$
                $counter(i) \longleftarrow 0$
                $\bar{b}^i \longleftarrow 0$
            else
                /* Replacement */
                $\bar{b}^i = \bar{b}^i + \frac{1}{N} b_t^i$
                $counter(i) \longleftarrow counter(i) + 1$
                if $counter(i) = N$ then
                    $\widetilde{b}_t^{*i} = \bar{b}^i$
                    $age(i) \longleftarrow 0$
                    $counter(i) \longleftarrow 0$
                    $replace(i) \longleftarrow true$
        else
            $age(i) \longleftarrow age(i) + 1$
            $counter(i) \longleftarrow 0$
            $\bar{b}^i \longleftarrow 0$
    output $\widetilde{B}_t^*$
end
```

3) The Background Model: Having described our two-phase scheme to iteratively improve $\tilde{B}_t^*$, we are now in a position to define a meaningful and steady initial background model $\tilde{B}^*$.

Definition 1: The initial background model $\tilde{B}^*$ is said to be $\tilde{B}_{t^*}^*$, if t^* is the earliest time instant satisfying the following three conditions: i) there is no block replacement occurred for the last N image frames, i.e., in calculating $\tilde{B}_{t^*-N+1}^*, \tilde{B}_{t^*-N+2}^*, \cdots, \tilde{B}_{t^*}^*$; ii) all image blocks in $\tilde{B}_{t^*}^*$ have been replaced at least one time since $t = 0$; and iii) they are of ages at least L. (See Algorithm 1 for details. In all our experiments, we have $N = 45$ and $L = N + 15 = 60$.)

III. Fast Classification With Soft Margins

In this section, issues related to the feature selection and the classifier formulation are addressed for the construction of an efficient background block classifier. In the feature selection, we have chosen to use features as general as possible so that the resulting classifier can handle a broad range of image sequences. Regarding the classifier formulation, two learning methods, *support vector machines* (SVMs) and *column generation boost* (CGBoost), are explored by investigating the following two issues. First, rather than binary-value classifiers, a classifier with probability outputs is required for our application. Second, the efficiency of the resulting classifier should fulfill the demand of real-time performance.

A. Feature Selection

For our purpose, the task of training is to learn a binary classifier for identifying background blocks from a video sequence captured by a static camera. We use a two-dimensional feature vector to characterize an image block b^i. The first component is the average *optical flow value*, where we apply the Lucas–Kanade's algorithm [24] to compute the flow magnitude of each pixel in b^i. In our implementation, it takes three image frames, I_{t-2}, I_{t-1}, and I_t, to calculate the flow values properly. However, we note that if one-frame delay is allowed, a slightly better results in evaluating the values of optical flow can be achieved by referencing I_{t-1}, I_t, and I_{t+1}. The second component of a feature vector is derived from the (mean) *interframe image difference* by $(1)/|b^i|^{-1} \sum_{(x,y)\in b^i} |I_{t-1}^{x,y} - I_t^{x,y}|$. To ensure good classification results, the feature values of both dimensions are normalized into [0,1] for training and for testing.

The two feature components are discriminant enough for our application owing to their generality and consistency in classifying background blocks of varied image sequences. We should also point out that since the optical flow values are computed using just three consecutive image frames, it may occur that a few pixels would have erratic/large flow values. Hence, an estimated upper-bound threshold is enforced to eliminate such errors. On the other hand, the additional cue using temporal differencing is more stable and easier to calculate, but it may fail to detect all the relevant cases. For example, the interframe difference may not be small in evaluating a background block that consists of slightly waving trees. Instead, an optical flow value is more informative to capture such a background block with small motions.

B. SVMs With Probability Outputs

For binary classifications, SVMs determine a separating hyperplane $f_S(\mathbf{x}) = \mathbf{w} \cdot \phi(\mathbf{x}), \mathbf{x} \in \mathbf{D}$ by transforming $\mathbf{D}$ from the input space to a high dimensional feature space, through a mapping function ϕ. The optimal hyperplane f_S^* can be obtained by solving the following soft-margin optimization problem:

$$\min_{\mathbf{w},\xi_i} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C_S^+ \sum_{i^+} \xi_i + C_S^- \sum_{i^-} \xi_i$$
$$\text{subject to} \quad y_i f_S(\mathbf{x}_i) \geq 1 - \xi_i, \quad i = 1, \ldots, m \tag{3}$$

where $\xi_i \geq 0$ are slack variables for tolerating sample noises and outliers. The two parameters C_S^+ and C_S^- are useful when dealing with unbalanced training data. (Recall that "+" is for background image blocks and "−" for foreground image blocks.) For the sake of reducing false positives, which may lead to more serious flaws in the estimated background model than false negatives would cause, C_S^- is given a value four times larger than the one for C_S^+ to penalize more the misclassifications of foreground blocks. In solving (3), we use a degree 2 polynomial kernel to yield satisfactory classification outcomes efficiently.

1) Probability Output: We use a *sigmoid model* to map an SVM score into the probability of being a background block by

$$P(\mathbf{x} \,|\, f_S^*) = \frac{1}{1 + \exp(A f_S^*(\mathbf{x}) + B)} \tag{4}$$

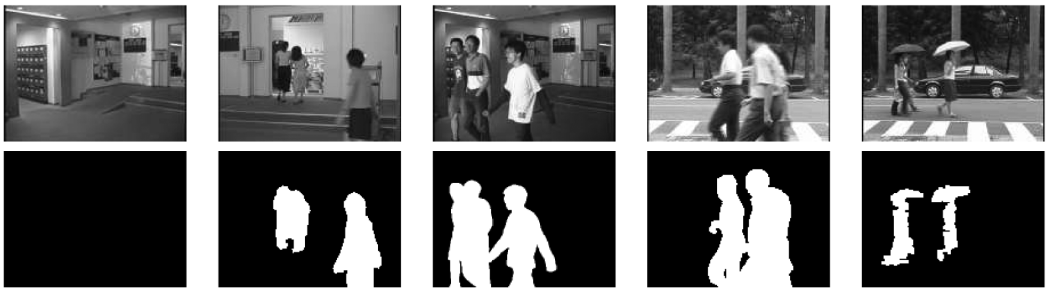


Fig. 4. Training data. Examples of collected images and their binary maps of the foreground (white) and the background regions (black) are plotted, top and bottom, respectively.

where the two parameters A and B can be fitted using *maximum likelihood estimation* from $\mathbf{D}$. Following [28], a *model-trust* algorithm is applied to solve the two-parameter optimization problem. In our experiments, 65% of the training blocks are used for deriving an SVM, and the other 35% are for calibrating probability outputs. The two fitted parameters are $A = -0.673724$ and $B = -2.359339$.

C. CGBoost With Probability Outputs

Among the many variants of boosting methods, the AdaBoost, introduced by Freund and Schapire [11], is the most popular one to derive an effective ensemble classifier iteratively. While AdaBoost has been proved to asymptotically achieve a maximum margin solution, recent studies also suggest the adoption of soft margin boosting to prevent the problem of overfitting [8], [29]. We thus employ the linear program boosting proposed by Demiriz *et al.* [8] for achieving soft-margin distribution over the training data $\mathbf{D}$ and acquiring an ensemble classifier $f_B = \sum_{j=1}^{T} \alpha_j f_j$, which is comprised of T weak learners f_js and weights α_js. Actually, Demiriz *et al.* apply a column generation method to solve the linear program by part, and establish an iterative boosting process that is similar to AdaBoost. Note that in implementing the CGBoost, the weak learners are constructed from *radial basis function* (RBF) networks, denoted as hs [29]. And each h has three Gaussian hidden units where two of them are initialized for the background, and the remaining one is for the foreground training data. Let $f_j(\mathbf{x}) = \mathrm{sign}(h_j(\mathbf{x}))$ be the weak learner selected at the jth iteration of CGBoost. Then, the RBF network h_j is derived by minimizing the following weighted error function

$$E_j = \frac{1}{2}\sum_{i=1}^{m} w_i (h_j(\mathbf{x}_i) - y_i)^2 \tag{5}$$

where $\{w_i\}$ is the weight distribution over training data $\mathbf{D}$ at the jth iteration.

1) Probability Output: Different from (4), it is more convenient to link boosting scores to probabilities. Friedman *et al.* [13] have proved that the AdaBoost algorithm can be viewed as a stagewise estimation procedure for fitting an additive logistic regression model. Consequently, a logistic transfer function can be directly applied to map CGBoost scores to posterior probabilities by

$$P(\mathbf{x} \mid f_B^*) = \frac{1}{1+\exp(-2f_B^*(\mathbf{x}))} \tag{6}$$

where the mapping in (6) is valid when the training data $\mathbf{D}$ do not contain a large portion of noisy samples or outliers. For the general case, it should still yield reasonable probability values with respect to the classification results by f_B^*s.

To summarize, both the two classifiers, f_S^* and f_B^*, seek a soft-margin solution when deciding a decision boundary for the training data $\mathbf{D}$. They indeed achieve similar classification performance in our experiments. However, SVMs are generally less efficient than boosting, as the number of support vectors increases rapidly with the size of $\mathbf{D}$. We thus prefer a CGBoost classifier for estimating an initial background model.

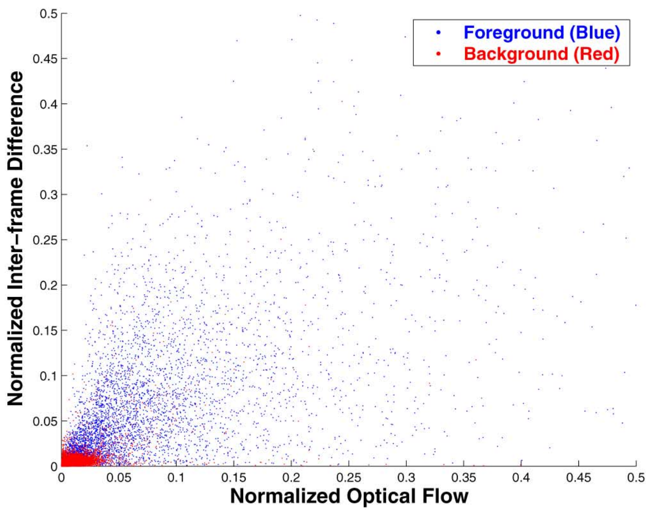
IV. Experimental Results and Discussions

To demonstrate the effectiveness of our approach, we first describe how the classifiers are learned for the specific problem. We then test the algorithm with a number of image sequences on a P4 1.8 GHz PC. Through illustrating with the experimental results, we highlight the advantages of learning a background model by classification, and make comparisons with those related works. Finally, possible future extensions to the current system are also explored.

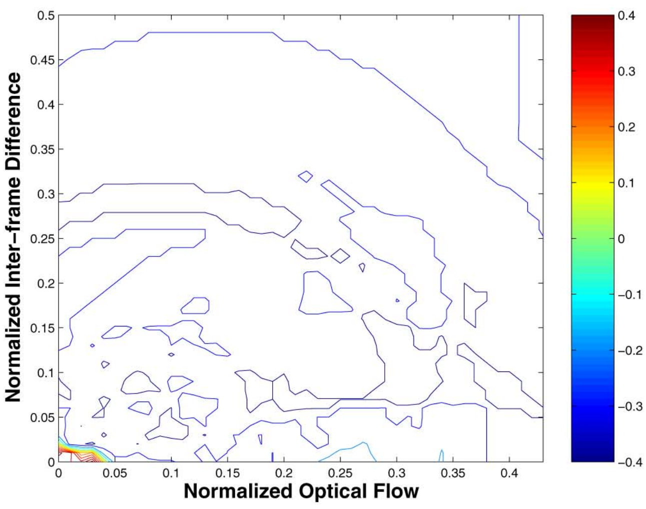
A. Classifier Learning

1) Training Data: We begin by collecting images that contain moving objects of different sizes and speeds from various indoor and outdoor image sequences captured by a static camera. These images are analyzed using a tracking algorithm (with known background models), decomposed into 8×8 image blocks, and then manually labeled as $+1$ for background blocks, or -1 for foreground ones. Examples of the collected images and the detected foreground and background regions are shown in Fig. 4. Since we prefer a resulting classifier to accommodate small variations, image blocks from regions of faint shadows or lighting changes are labeled as background. The feature vector of an image block can be computed straightforwardly by referencing the related $\ell = 3$ blocks from the respective image sequence. Totally, there are 27 600 image blocks collected to form the training data $\mathbf{D}$. As shown in Fig. 5(a), the features extracted from the background blocks are mostly of small values, while those extracted from the foreground blocks mostly have feature values corresponding to the regions of large motions.

2) Classifier Evaluations: The training and the classification outcomes by implementing the classifier respectively with f_S^* and f_B^* are summarized in Table I. Owing to the soft-margin



(a)



(b)

Fig. 5. (a) Distribution of the training data. The training features are normalized to the values between 0 and 1. In order to detail the distribution of background samples, only the part of 0 to 0.5 is plotted. (b) Level curve of f_B^*'s decision scores. The zero-score decision boundary is located on the lower-left corner of the plot.

TABLE I
COMPARISONS BETWEEN SVMS AND CGBOOST
(USING ADABOOST AS A BENCHMARK)

Classifier	SVM f_S^*	CGBoost f_B^*	AdaBoost f_A
Settings	Image Size: 320×240, Platform: P4-1.8GHz PC		
Parameters	$C_S^+ = 20$, $C_S^- = 80$	$C_B = \frac{10}{27600}$	None
Components	4185 SVs	33 f_js	33 f_js
Error Rate*	0.0466	0.0466	0.0507
Test Speed	0.4fps	9.5fps	9.5fps

* Error Rate = # of Misclassified Blocks / # of Training Blocks

TABLE II
AVERAGE ERROR RATES OF TENFOLD CROSS VALIDATION IN DIFFERENT THRESHOLD SETTINGS

τ^*	0.5	0.6	0.7
False Positive	0.02912	0.02768	0.01196
False Negative	0.01877	0.02062	0.49652

property of the two classifiers, almost the same training errors have been obtained. However, the classification efficiency of f_B^* is more than 20 times faster than that of f_S^*. To visualize the distribution of a derived classifier, for example, f_B^*, its level curves of the decision scores are plotted in Fig. 5(b). It can be observed that the area of positive scores is located near the lower-left corner, which is consistent with the distribution of feature values computed from the training data.

3) Probability Thresholding: For the sake of reducing false positives, we adopt a stricter probability threshold $\tau^* = 0.6$ in setting the decision boundary of a CGBoost classifier. This value is determined through tenfold cross validation. In Table II, the average values of the false positive and false negative rates in cross validation with respect to different threshold settings are listed. While false negatives mainly affect the needed time in estimating an initial background model, the false positives, i.e., misclassifying foreground blocks as background, will have direct impacts on the quality of the background model. Thus, it is preferable to choose 0.6 as the probability threshold in that it causes fewer false positives without introducing too many false negatives.

B. Some Experiments

Since the classification efficiency of CGBoost is more than 20 times faster than that of an SVM implementation (see Table I), we describe below only the experimental results yielded by using the CGBoost classifier f_B^*. For testing the generality of the proposed scheme, all the to-be-estimated scenes of the testing sequences are completely different from those of the training data. The testing sequences also contain complex motions, e.g., substantial object interactions, and varied lighting conditions, like cloudiness.

1) Background Model Estimation: We first demonstrate the efficiency of our method for an outdoor environment. The sequence $\mathcal{A}$ contains different types of objects, including slightly waving trees, walking people, slow and fast moving vehicles, and even a stationary bike rider. We shall use this example as a benchmark to analyze the quality of our results, detection rates, and comparisons to other existing algorithms. As illustrated in Fig. 6(a) and (b), the background model is initialized into an empty set at $t = 0$, and it is until the forty-fourth frame that stationary regions of the scene are started to be incorporated into the model (due to $N = 45$ in our setting). Fig. 6(c) shows a very slow moving car is falsely adapted into the background in transient (and is eventually removed after its leaving the scene). More interesting is the scenario depicted in Fig. 6(d) and (e) that a bike rider waiting for a green traffic light has remained still long enough to become a part of the derived background model at $t^* = 650$. Then, the system can start to track objects via frame differencing and proper model updating. On the other hand, if we subtract the model from the first t^* frames, it gives the *complexity* of how the background model is initialized. Factors such as dark shadows and waving trees can now be easily identified from those shown in Fig. 6(f)–(j).

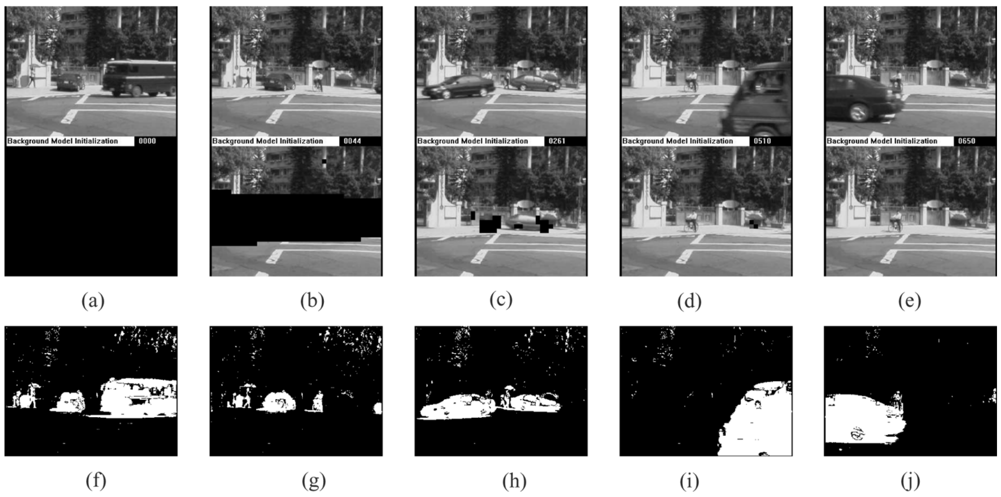

Background Model Initialization 0000
Background Model Initialization 0044
Background Model Initialization 0261
Background Model Initialization 0510
Background Model Initialization 0650
(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)

Fig. 6. (a)–(e) Upper row shows image frames from sequence $\mathcal{A}$, and the lower row depicts the progressive estimation results. The initial background model is completed at $t^* = 650$. (f)–(j) The frame subtraction results by referencing the derived background model $\tilde{B}^*_{650}$ (a) $\mathcal{A}000, \tilde{B}^*_0$, (b) $\mathcal{A}044, \tilde{B}^*_{44}$, (c) $\mathcal{A}261, \tilde{B}^*_{261}$, (d) $\mathcal{A}510, \tilde{B}^*_{510}$, (e) $\mathcal{A}650, \tilde{B}^*_{650}$ (f) $\mathcal{A}000 - \tilde{B}^*_{650}$, (g) $\mathcal{A}044 - \tilde{B}^*_{650}$, (h) $\mathcal{A}261 - \tilde{B}^*_{650}$, (i) $\mathcal{A}510 - \tilde{B}^*_{650}$, (j) $\mathcal{A}650 - \tilde{B}^*_{650}$.

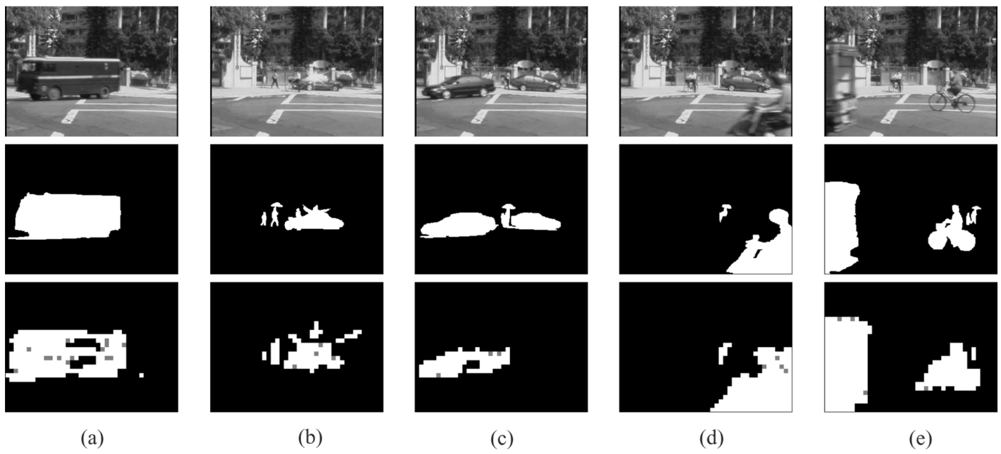


Fig. 7. Row one: Image frames from sequence $\mathcal{A}$. Row two: The manually labeled foreground (white) and background (black) maps. Note that the very slow-moving car in (c) that later becomes fully stationary in (d) is labeled as foreground and background, respectively. Row three: Our background block detection results. The foreground blocks in gray are identified by the top-down validation process. (a) $\mathcal{A}020$, (b) $\mathcal{A}166$, (c) $\mathcal{A}261$, (d) $\mathcal{A}372$, and (e) $\mathcal{A}540$

2) Background Block Detection: To quantitatively evaluate the accuracy of the bottom-up block classifications and the improvement with the top-down validations, we select 20 image frames from sequence $\mathcal{A}$ that contain moving objects of different sizes and speeds, specular light, and shadows. We then manually label each image block of the twenty frames to result in a set of 20 061 background and 3939 foreground blocks, where we shall use them to examine the accuracy of our scheme for background block detection. In Fig. 7, we show results for five selected frames. Note that those gray blocks are detected as foreground through the top-down validation process. To further justify the need of a local and global approach, a comparison of the detection error rates with or without the top-down validation step is given in Table III. Although the values of detection rates could vary from testing our system in different environments, it is clear that the improvement of reducing the errors by applying the top-down validation is significant. As in this example, the reduction rate of false positives is about 18.744% while the increase rate of false negatives is only 3.059%. Two observations could arise from the foregoing verification for the accuracy of our scheme in detecting background blocks.

TABLE III
DETECTION ERROR RATES WITH OR WITHOUT THE TOP-DOWN VALIDATION

BG/FG Block Detection	Without top-down	With top-down	% Improvement
Detection Error Rate*	0.04142	0.03779	8.764 %
False Positive Rate	0.02246	0.01825	18.744 %
False Negative Rate	0.01896	0.01954	-3.059 %

* Detection Error Rate = # of Misclassified Blocks / # of Testing Blocks

1) For the classifier to accommodate small variations like waving trees, it may mistakenly classify very slow-moving objects into background [see Figs. 6(c) and 7(c)]. This is indeed a trade-off, and we resolve the issue by learning a proper decision boundary from the training data.
2) Our classification scheme may suffer from the aperture problem in detecting large objects in that we use motion features to construct a general classifier [see Fig. 7(a) and

(c)]. With the top-down validation, this problem can be alleviated to some degree. Still a number of false positives caused by the aperture problem exist framewise. However, since only the same false positive occurring for N consecutive frames would be adapted into a background model, such an event rarely happens in practice (with a very low probability, e.g., around 0.01825^N for the example in Table III).

3) Parameter Settings: We next investigate the sensitivity of our method with respect to different values of the two parameters N and L. (Since $L = N + 15$, it is indeed a one-parameter scheme.) Specifically, we have experimented with $N = 30$, 45, 60, 75, and 90. Our results show that they mainly affect the needed time to compute a stable initial background model. The larger the value of N is, the longer period of time it takes to complete the estimation. Except for $N = 30$, which is too short a time period for yielding a stationary adaptation, all other settings of N lead to stable background models.

4) Related Comparisons: A clear advantage of our formulation is the ability to know when a well-defined initial background model is ready to be used for tracking. We demonstrate this point by making comparisons with the popular mixture of Gaussians model [33] and the local image flow approach [16]. While the two methods are also effective for background initialization, they both lack a *clear* definition of what an underlying background scene is at any time instant of the estimation processes. For systems based on the mixture of Gaussians, they work by memorizing a certain number of modes for each pixel, and then by pixelwise integrating the most probable modes to form a background model. This is in essence a local scheme that the overall quality of a background model is difficult to evaluate. On the other hand, the method described in [16] is designed to process a whole image sequence to output a background model. We thus need to modify the algorithm into a sequential one so that the comparisons can be done by framewise examining the respectively derived background models.

The first experiment is carried out with image sequence $\mathcal{A}$ where the three algorithms are alternately run till the image frame $t^* = 650$ that our method completes its estimation for an initial background model. For the mixture model, we use three Gaussian distributions and a blending rate of 0.01, and initialize the background model at $t = 0$ to the first image frame. For the local image flow implementation, the values of w and $\delta_{\max}$ are set to 30 and 15, and the background model is an empty set at $t = 0$. In Fig. 8, we show some intermediate results of ours and the corresponding background models produced by the other two methods. Due to the batch nature of the local image flow scheme, its three background models shown in Fig. 8 are obtained by running the algorithm three times, using the respective periods of image frames as the inputs. Overall, the results produced by ours and the mixture of Gaussians are more reliable than those of the local image flow, largely because the local flow scheme relies heavily on the estimations of optical flow directions and their accuracy. While the outcomes by the mixture of Gaussians seem to be satisfactory and similar to ours, the absence of a good measurement to guarantee the quality of the resulting background models remains a disadvantage of the approach. Furthermore, as one would expect that a mixture of Gaussians method should be sensitive to lighting variations in that it is done by locally combining pixel intensities. We shall further elaborate on this issue with the next experiment.

Our second comparison focuses on the effects of lighting changes. For the outdoor sequence $\mathcal{B}$ (see Fig. 9), the lighting condition varies rapidly due to overcast clouds. And the experimental results show that our method is less sensitive to variations of this kind. Specifically, in Fig. 9(e) and (f), we enlarge the sizes and enhance the contrasts of the two derived background models for a clearer view. Note that especially in the road area our background model estimation is clearly of better quality than the one yielded by the mixture of Gaussians. This is mostly because of our uses of motion cues for identifying background blocks and the properties of the MAP background model for integrating local and global consistency. On the other hand, the mixture of Gaussians approach uses only the pixelwise intensity information so that its performance depends critically on the variations of intensity distribution about the background scene.

5) Initialization and Tracking: To further illustrate the efficiency of using our proposed algorithm to estimate a background model for tracking, we show the estimations of initial background models of test sequences $\mathcal{C}$ and $\mathcal{D}$, and some subsequent tracking results in Fig. 10. Below each depicted image frame I_t, the corresponding background model $\tilde{B}_t^*$ is plotted. In the two experiments, the estimations of the initial background model $\tilde{B}_{t^*}^*$ are completed at frame number $t^* = 470$ and 243, respectively. Once the $\tilde{B}_{t^*}^*$ is available, the system can start to track objects immediately, using the scheme described in [4] (see Fig. 10). We also note that, as demonstrated in Fig. 10(j)–(l), the background model can be updated appropriately during tracking, even when significant changes in the scene background have occurred.

C. Discussions

An efficient online algorithm to establish a background model for tracking is proposed. The key idea of our approach is simple but effective: If one can tell whether an image block is part of the background, the additional knowledge can help to perform appropriate block updates. In addition, we introduce a global consistency check to eliminate noises in the updates. The two mechanisms, together, lead to a reliable system.

In background classifier learning, a classifier formulation with probability outputs is adopted so that the classification boundary can be easily tuned. While both an SVM and a CG-Boost classifier are appropriate for this purpose, the latter has an advantage of efficiency and is thus applied to the experiments. We also note that the relevance vector machines (RVMs) [34] are another possible choice. Particularly, RVMs are derived from MAP equations, and are truly probabilistic.

Regarding feature selection, two general motion cues, the interframe difference and the optical flow value, are adopted to discriminate background scenes. While the interframe difference is effective in detecting static background blocks, the optical flow value, on the other hand, provides discriminability in classifying image blocks in small motions into gradually-varying background or moving foreground. To further justify

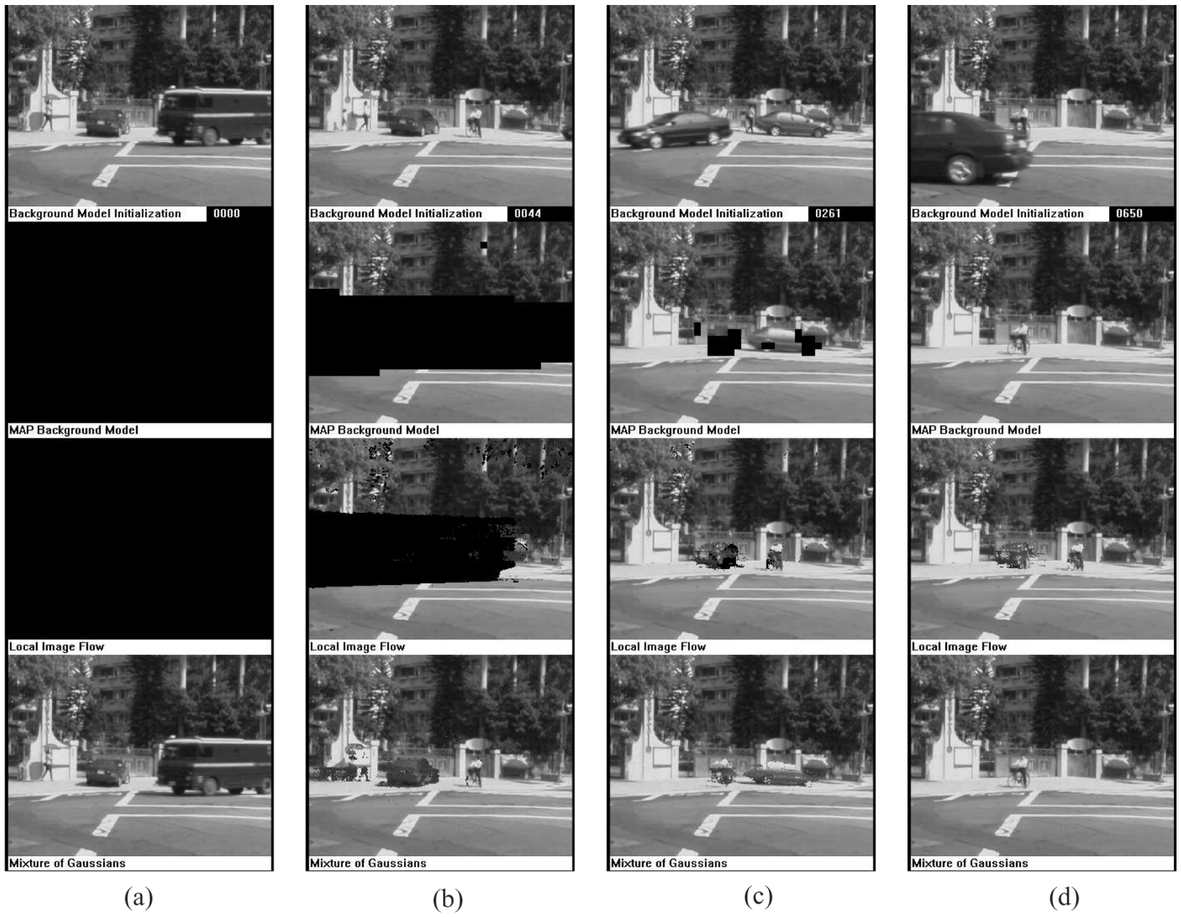


Fig. 8. Row one: Images frames from image sequence $\mathcal{A}$. Row two: Intermediate results of background estimation by our method that completes at $t^* = 650$. Row three and four: The results respectively derived by the local image flow approach [16] and the mixture of Gaussians method [33] at each corresponding time instant.

the use of the optical flow cue, additional evaluations using the interframe difference alone are provided. With the best setting of the difference threshold at 0.013, the training error is raised from 0.0466 to 0.0528 (or a 13.3% increase), and the testing error for the 20 evaluation image frames increases from 0.0378 to 0.0436 (or a 15.3% increase). Hence, the benefit of incorporating the optical flow value is obvious.

About the three parameters, τ^*, δ and N, used in our method, only N needs to be determined manually. Indeed, the experimental results demonstrate that the algorithm is robust to different parameter settings, and can handle lighting variations. Overall, our system is shown to be useful and practical for real-time tracking applications. For the future work, we are now extending the system to accommodate a pan/tilt/zoom camera. Such a modification would lead to a more challenging problem of simultaneous estimations of several initial background models.

Appendix

Described below are the derivations of the MAP formulation (2) in Section II-A. For easy explanation, the derivations are decomposed into six parts, which are classifier training, iterative formulation, posterior probability decomposition, likelihood probability decomposition, background block classification, and the final MAP formulation.

1) Classifier Training: To begin with, a MAP classifier derived from the training data $\mathbf{D}$ is defined by $f^* = \mathrm{argmax}_f P(f \,|\, \mathbf{D}) = \mathrm{argmax}_f P(f \,|\, \mathbf{X}, \mathbf{Y})$. It can be interpreted as a supervised learning process to train an optimal classifier f^* from the training data $\mathbf{D} = \{\mathbf{X}, \mathbf{Y}\}$. With the definition of f^*, we can start to derive the following equations to estimate a background model:

$$\begin{aligned} &P(\tilde{B}_t \,|\, \mathbf{I}_t, \mathbf{D}) \\ &\quad = \int P(\tilde{B}_t \,|\, \mathbf{I}_t, \mathbf{D}, f)\, p(f \,|\, \mathbf{I}_t, \mathbf{D})\, df \\ &\quad = \int P(\tilde{B}_t \,|\, \mathbf{I}_t, f)\, p(f \,|\, \mathbf{D})\, df \\ &\quad \approx P(\tilde{B}_t \,|\, \mathbf{I}_t, f^*) \end{aligned}$$

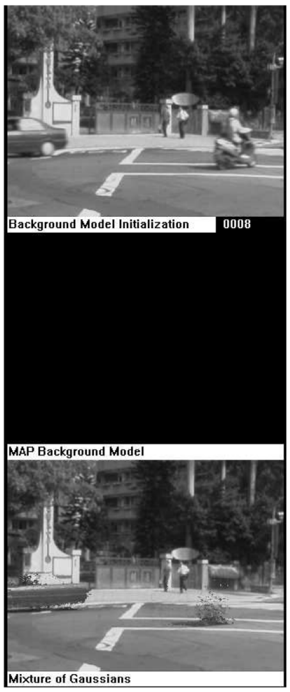
where $P(f \,|\, \mathbf{D})$ is assumed to peak at the optimal classifier f^* (e.g., see [32, pp. 474–476]).

2) Iterative Formulation: To develop an iterative form for estimating a background model, we first define

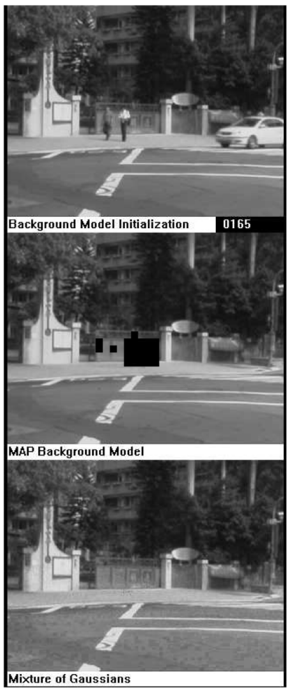
$$\tilde{B}_t^* = \underset{\tilde{B}_t}{\mathrm{argmax}}\, P(\tilde{B}_t \,|\, \mathbf{I}_t, f^*)$$

and

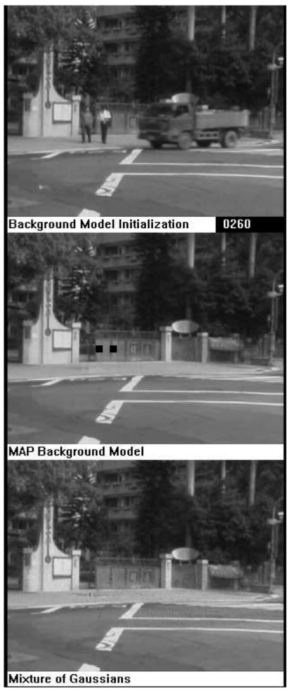
$$\tilde{B}_{t-1}^* = \underset{\tilde{B}_{t-1}}{\mathrm{argmax}}\, P(\tilde{B}_{t-1} \,|\, \mathbf{I}_{t-1}, f^*).$$



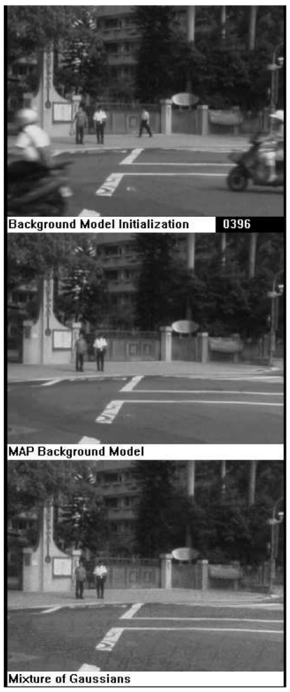
(a)



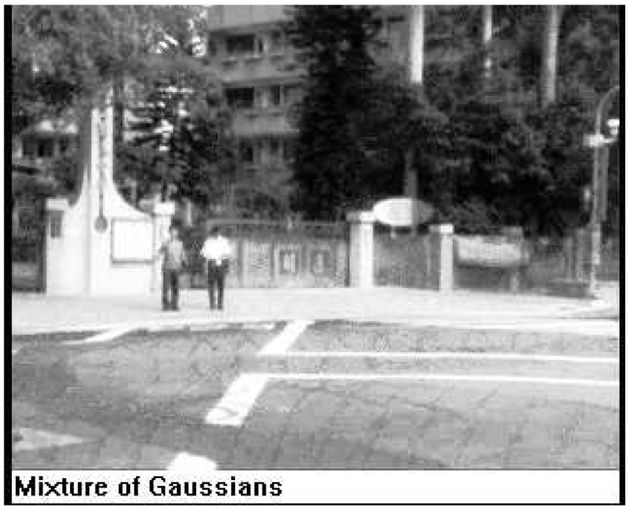
(b)



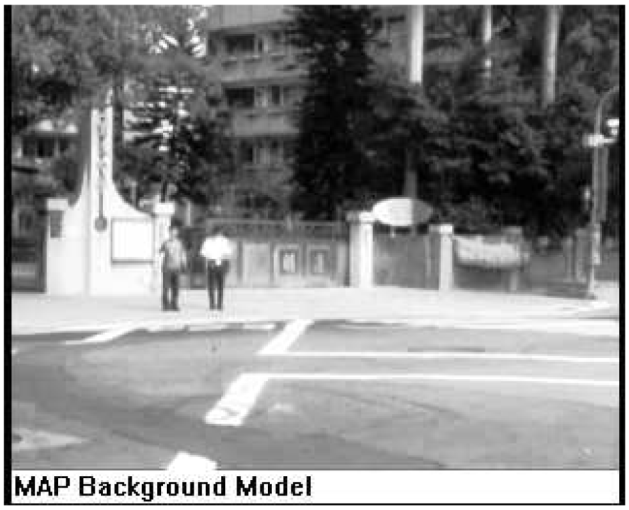
(c)



(d)



(e)



(f)

Fig. 9. Test sequence $\mathcal{B}$ for lighting variations. (a)–(d) Due to overcast clouds, the outdoor lightings over the road change significantly throughout the sequence. As a result, the quality of background models yielded by the mixture of Gaussians is considerably affected. However, our formulation is more robust to such lighting perturbations. (e)–(f) Two derived background models at $t^* = 475$ are enlarged and enhanced in contrast. False textures and extra noises can be observed in the road areas of (e). (a) $\mathcal{B}008$, (b) $\mathcal{B}165$, (c) $\mathcal{B}260$, (d) $\mathcal{B}396$, (e) Mixture of Gaussians, and (f) MAP.

Then we have

$$\begin{aligned}
P(\tilde{B}_t \,|\, \mathbf{I}_t, f^*)
&= \sum_{\tilde{B}_{t-1}} (P(\tilde{B}_t \,|\, \mathbf{I}_t, f^*, \tilde{B}_{t-1}) P(\tilde{B}_{t-1} \,|\, \mathbf{I}_t, f^*)) \\
&= \sum_{\tilde{B}_{t-1}} (P(\tilde{B}_t \,|\, \mathbf{I}_{t,\ell}, \mathbf{I}_{t-\ell}, f^*, \tilde{B}_{t-1}) \\
&\qquad \times P(\tilde{B}_{t-1} \,|\, \mathbf{I}_{t-1}, I_t, f^*))
\end{aligned}$$

(The image frames $\mathbf{I}_{t,\ell}$ are used later to compute feature vectors for classification.)

$$\begin{aligned}
&= \sum_{\tilde{B}_{t-1}} (P(\tilde{B}_t \,|\, \mathbf{I}_{t,\ell}, f^*, \tilde{B}_{t-1}) P(\tilde{B}_{t-1} \,|\, \mathbf{I}_{t-1}, f^*)) \\
&\approx P(\tilde{B}_t \,|\, \mathbf{I}_{t,\ell}, f^*, \tilde{B}^*_{t-1})
\end{aligned}$$

where, similarly, $P(\tilde{B}_{t-1} \,|\, \mathbf{I}_{t-1}, f^*)$ is assumed to peak at $\tilde{B}^*_{t-1}$.

3) Posterior Probability Decomposition: Using Bayes' rule, we decompose $P(\tilde{B}_t \,|\, \mathbf{I}_{t,\ell}, f^*, \tilde{B}^*_{t-1})$ into a product of an image likelihood term and a prior.

$$\begin{aligned}
&P(\tilde{B}_t \,|\, \mathbf{I}_{t,\ell}, f^*, \tilde{B}^*_{t-1}) \\
&\quad = P(\tilde{B}_t \,|\, I_t, \ldots, I_{t-\ell+1}, f^*, \tilde{B}^*_{t-1})
\end{aligned}$$

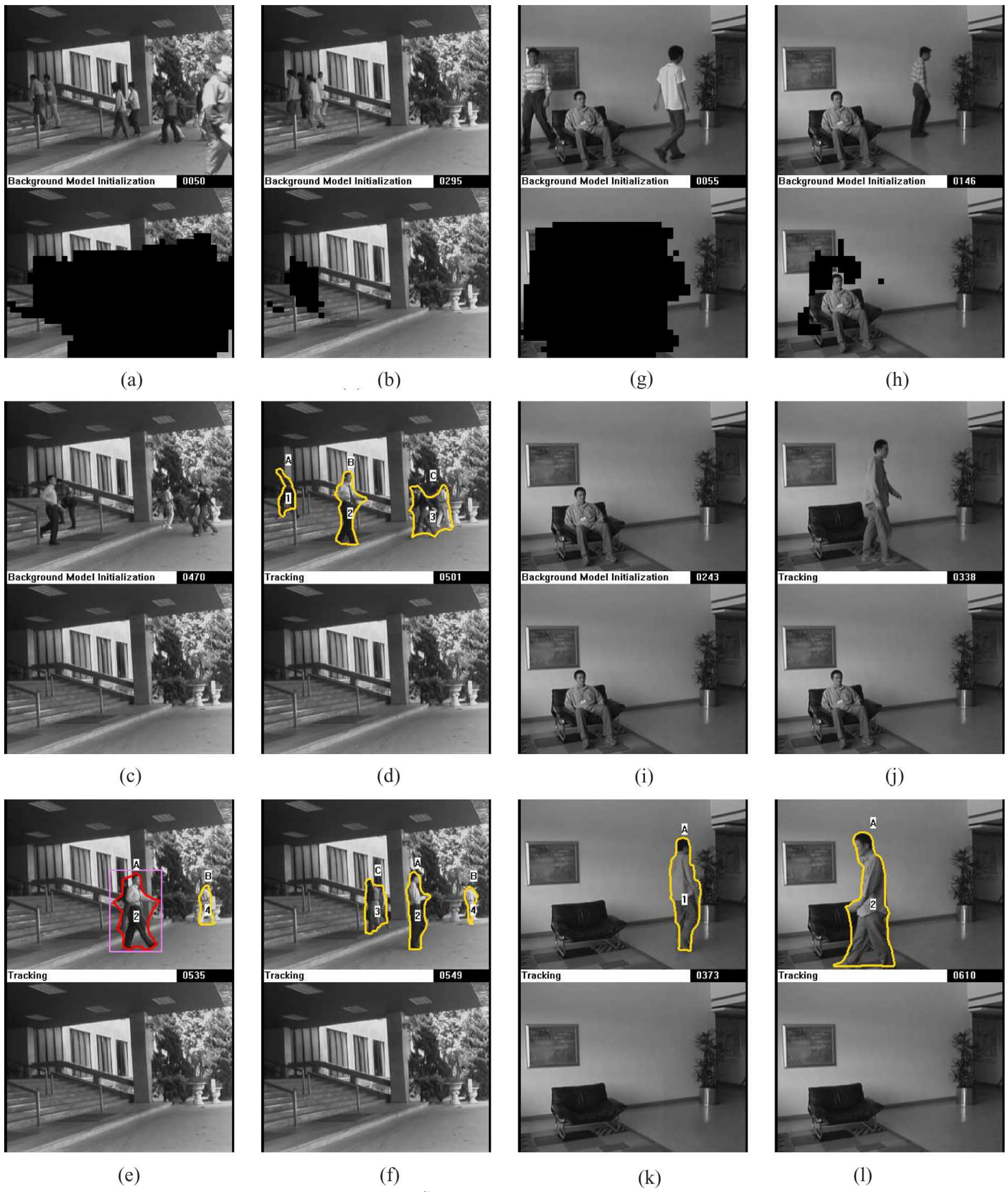

Background Model Initialization 0050
(a)
Background Model Initialization 0295
(b)
Background Model Initialization 0055
(g)
Background Model Initialization 0146
(h)
Background Model Initialization 0470
(c)
Tracking 0501
A
B
C
1
2
3
(d)
Background Model Initialization 0243
(i)
Tracking 0338
(j)
Tracking 0535
A
B
2
4
(e)
Tracking 0549
C
A
B
3
2
4
(f)
Tracking 0373
A
1
(k)
Tracking 0610
A
2
(l)

Fig. 10. Current image frame I_t and the derived background model $\tilde{B}_t^*$ are plotted together, top and bottom, respectively. Some tracking results are also shown in the I_ts: (a) $\mathcal{C}050$, (b) $\mathcal{C}295$, (g) $\mathcal{D}\prime\triangledown\triangledown$, (h) $\mathcal{D}146$ (c) $\mathcal{C}470(t^* = 470)$, (d) $\mathcal{C}501$, (i) $\mathcal{D}243(t^* = 243)$, (j) $\mathcal{D}338$ (e) $\mathcal{C}535$, (f) $\mathcal{C}549$, (k) $\mathcal{D}373$, and (l) $\mathcal{D}610$.

$$\begin{aligned}
&\propto P(I_t \mid \tilde{B}_t, I_{t-1}, \ldots, I_{t-\ell+1}, \tilde{B}^*_{t-1}, f^*)\\
&\quad \times P(\tilde{B}_t \mid I_{t-1}, \ldots, I_{t-\ell+1}, f^*, \tilde{B}^*_{t-1})\\
&= P(I_t \mid \tilde{B}_t, \mathbf{I}_{t-1,\ell-1}, \tilde{B}^*_{t-1}, f^*)\\
&\quad \times P(\tilde{B}_t \mid \mathbf{I}_{t-1,\ell-1}, f^*, \tilde{B}^*_{t-1})\\
&= \underbrace{P(I_t \mid \tilde{B}_t, \mathbf{I}_{t-1,\ell-1}, \tilde{B}^*_{t-1}, f^*)}_{\text{image likelihood}} \underbrace{P(\tilde{B}_t \mid \tilde{B}^*_{t-1})}_{\text{prior}}.
\end{aligned}$$

Because the classifier f^* is used to perform blockwise (local) classifications, and $\mathbf{I}_{t-1,\ell-1}$ are those image frames used in computing feature values, both of them are eliminated from the prior probability which is used to measure the global consistency over image blocks. That is, we simplify the prior term from $P(\tilde{B}_t \mid \mathbf{I}_{t-1,\ell-1}, f^*, \tilde{B}^*_{t-1})$ to $P(\tilde{B}_t \mid \tilde{B}^*_{t-1})$.

4) Likelihood Probability Decomposition: Applying the assumption of blockwise independencies, the likelihood term can be further decomposed as follows:

$$\begin{aligned}
&P(I_t \mid \tilde{B}_t, \mathbf{I}_{t-1,\ell-1}, \tilde{B}^*_{t-1}, f^*)\\
&\quad = \prod_{i=1}^{n} P\left(b_t^i \mid \tilde{b}_t^i, \mathbf{b}^i_{t-1,\ell-1}, \tilde{b}^{*i}_{t-1}, f^*\right)
\end{aligned}$$

$$\begin{aligned}&\propto \prod_{i=1}^{n}\Big(P(b_t^i\,|\,\mathbf{b}_{t-1,\ell-1}^i,\tilde{b}_{t-1}^{*i},f^*)\\&\qquad\times P\left(\tilde{b}_t^i\,|\,b_t^i,\mathbf{b}_{t-1,\ell-1}^i,\tilde{b}_{t-1}^{*i},f^*\right)\Big)\\&=\prod_{i=1}^{n}\Big(P\left(b_t^i\,|\,\mathbf{b}_{t-1,\ell-1}^i,\tilde{b}_{t-1}^{*i},f^*\right)\\&\qquad\times P\left(\tilde{b}_t^i\,|\,\mathbf{b}_{t,\ell}^i,\tilde{b}_{t-1}^{*i},f^*\right)\Big)\\&=\prod_{i=1}^{n}P\left(b_t^i\,|\,\mathbf{b}_{t-1,\ell-1}^i\right)\prod_{i=1}^{n}P\left(\tilde{b}_t^i\,|\,\mathbf{b}_{t,\ell}^i,\tilde{b}_{t-1}^{*i},f^*\right).\\&\propto\prod_{i=1}^{n}P\left(\tilde{b}_t^i\,|\,\mathbf{b}_{t,\ell}^i,\tilde{b}_{t-1}^{*i},f^*\right).\end{aligned}$$

The term $P(b_t^i\,|\,\mathbf{b}_{t-1,\ell-1}^i,\tilde{b}_{t-1}^{*i},f^*)$ is reduced to $P(b_t^i\,|\,\mathbf{b}_{t-1,\ell-1}^i)$, because b_t^i is the ith block of frame I_t from an arbitrary online image stream, and it should be independent from our choice of a classifier f^* and what the ith block of a background model is at time $t-1$, i.e., $\tilde{b}_{t-1}^{*i}$.

5) Background Block Classification: To utilize background block classification in estimating a background model, we have

$$P\left(\tilde{b}_t^i\,|\,\mathbf{b}_{t,\ell}^i,\tilde{b}_{t-1}^{*i},f^*\right)\doteq\begin{cases}P\left(\tilde{b}_t^i\,|\,b_t^i,\tilde{b}_{t-1}^{*i}\right), & \text{if } b_t^i \text{ is classified as background by } f^*\\ P\left(\tilde{b}_t^i\,|\,\tilde{b}_{t-1}^{*i}\right), & \text{otherwise.}\end{cases}$$

Then we derive the decomposition for the image likelihood

$$P(I_t\,|\,\tilde{B}_t,\mathbf{I}_{t-1,\ell-1},\tilde{B}_{t-1}^*,f^*)\propto\prod_{i^+}P\left(\tilde{b}_t^i\,|\,b_t^i,\tilde{b}_{t-1}^{*i}\right)\prod_{i^-}P\left(\tilde{b}_t^i\,|\,\tilde{b}_{t-1}^{*i}\right)$$

where $i^+=\{i\,|\,b_t^i \text{ is a background block}\}$ and $i^-=\{1,\ldots,n\}-i^+$.

6) The Final MAP Formulation: With all these derivations, we arrive at the following MAP optimization

$$\tilde{B}_t^*=\underset{\tilde{B}_t}{\operatorname{argmax}}\left\{\left(\prod_{i^+}P\left(\tilde{b}_t^i\,|\,b_t^i,\tilde{b}_{t-1}^{*i}\right)\prod_{i^-}P\left(\tilde{b}_t^i\,|\,\tilde{b}_{t-1}^{*i}\right)\right)\times P(\tilde{B}_t\,|\,\tilde{B}_{t-1}^*)\right\}.$$

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments on the manuscript of this paper.

REFERENCES

[1] P. M. Q. Aguiar and J. M. F. Moura, "Figure-ground segmentation from occlusion," *IEEE Trans. Image Process.*, vol. 14, no. 8, pp. 1109–1124, Aug. 2005.

[2] P. M. Q. Aguiar and J. M. F. Moura, "Joint segmentation of moving object and estimation of background in low-light video using relaxation," in *Proc. 2007 IEEE Int. Conf. Image Processing*, 2007, vol. 5, pp. 53–56.

[3] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

[4] H.-T. Chen, H.-H. Lin, and T.-L. Liu, "Multi-object tracking using dynamical graph matching," in *Proc. Conf. Computer Vision Pattern Recognition*, Kauai, HI, 2001, vol. 2, pp. 210–217.

[5] S. Cohen, "Background estimation as a labeling problem," in *Proc. 10th IEEE Int. Conf. Computer Vision*, 2005, vol. 2, pp. 1034–1041.

[6] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1337–1342, Oct. 2003.

[7] F. De la Torre and M. Black, "Robust principal component analysis for computer vision," in *Proc. 8th IEEE Int. Conf. Computer Vision*, Vancouver, Canada, 2001, vol. 1, pp. 362–369.

[8] A. Demiriz, K. Bennett, and J. Shawe-Taylor, "Linear programming boosting via column generation," *Mach. Learn.*, vol. 46, no. 1–3, pp. 225–254, 2002.

[9] A. Elgammal, D. Harwood, and L. Davis, "Nonparametric background model for background subtraction," in *Proc. 6h Eur. Conf. Computer Vision*, Trinity College, Dublin, Ireland, 2000, vol. 2, pp. 751–767.

[10] T. Ellis and M. Xu, "Object detection and tracking in an open and dynamic world," presented at the IEEE Int. Workshop Performance Evaluation of Tracking and Surveillance, Kauai, HI, Dec. 9, 2001.

[11] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. Machine Learning*, San Francisco, CA, 1996, pp. 148–156.

[12] B. J. Frey, N. Jojic, and A. Kannan, "Learning appearance and transparency manifolds of occluded objects in layers," in *Proc. Conf. Computer Vision Pattern Recognition*, 2003, vol. 1, pp. 45–52.

[13] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *Ann. Statist.*, vol. 38, no. 2, pp. 337–374, Apr. 2000.

[14] N. Friedman and S. Russell, "Image segmentation in video sequences: A probabilistic approach," in *Proc. 13th Conf. Uncertainty in Artificial Intelligence*, San Francisco, CA, 1997, pp. 175–181.

[15] X. Gao, T. Boult, F. Coetzee, and V. Ramesh, "Error analysis of background adaption," in *Proc. Conf. Computer Vision Pattern Recognition*, Hilton Head Island, SC, 2000, vol. 1, pp. 503–510.

[16] D. Gutchess, M. Trajkovics, E. Cohen-Solal, D. Lyons, and A. Jain, "A background model initialization algorithm for video surveillance," in *Proc. 8th IEEE Int. Conf. Computer Vision*, Vancouver, Canada, 2001, vol. 1, pp. 733–740.

[17] I. Haritaoglu, D. Harwood, and L. Davis, "A fast background scene modeling and maintenance for outdoor surveillance," in *Proc. 15th Int. Conf. Pattern Recognition*, Barcelona, Spain, 2000, vol. 4, pp. 179–183.

[18] E. Hayman and J.-O. Eklundh, "Statistical background subtraction for a mobile observer," in *Proc. 9th IEEE Int. Conf. Computer Vision*, Nice, France, 2003, pp. 67–74.

[19] M. Heikkilä and M. Pietikäinen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 657–662, 2006.

[20] M. Irani and S. Peleg, "Motion analysis for image enhancement: Resolution, occlusion, and transparency," *J. Vis. Commun. Image Represent.*, vol. 4, pp. 324–335, 1993.

[21] N. Jojic and B. J. Frey, "Learning flexible sprites in video layers," in *Proc. Conf. Computer Vision Pattern Recognition*, Kauai, HI, 2001, vol. 1, pp. 199–206.

[22] D.-W. Kim and K.-S. Hong, "Practical background estimation for mosaic blending with patch-based Markov random fields," *Pattern Recognit.*, vol. 41, no. 7, pp. 2145–2155, 2008.

[23] D.-S. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 827–832, 2005.

[24] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. DARPA IU Workshop*, 1981, pp. 121–130.

[25] S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler, "Tracking groups of people," *Comput. Vision Image Understand.*, vol. 80, no. 1, pp. 42–56, Oct. 2000.

[26] A. Mittal and D. Huttenlocher, "Site modeling for wide area surveillance and image synthesis," in *Proc. Conf. Computer Vision Pattern Recognition*, Hilton Head Island, SC, 2000, vol. 2, pp. 160–167.

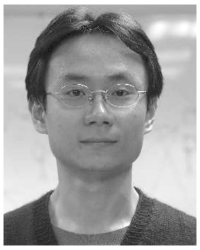
[27] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background modeling and subtraction of dynamic scenes," in *Proc. 9th IEEE Int. Conf. Computer Vision*, Nice, France, 2003, pp. 1305–1312.

[28] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, Eds. Cambridge, MA: MIT Press, 2000.

[29] G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for adaboost," *Mach. Learn.*, vol. 42, pp. 287–320, 2001.

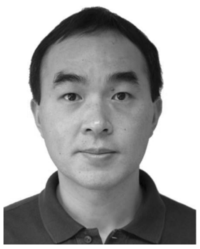
[30] C. Ridder, O. Munkelt, and H. Kirchner, "Adaptive background estimation and foreground detection using kalman-filtering," in *Proc. Int. Conf. Recent Advances Mechatronics*, 1995, pp. 193–199.

[31] J. Rittscher, J. Kato, S. Joga, and A. Blake, "A probabilistic background model for tracking," in *Proc. 6th Eur. Conf. Computer Vision*, Trinity College, Dublin, Ireland, 2000, vol. 2, pp. 336–350.
[32] B. Schölkopf and A. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2002, pp. 469–516.
[33] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. Conf. Computer Vision Pattern Recognition*, Fort Collins, CO, 1999, vol. 2, pp. 246–252.
[34] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Jun. 2001.
[35] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Proc. 7th IEEE Int. Conf. Computer Vision*, Corfu, Greece, 1999, vol. 1, pp. 255–261.
[36] D. Wang, T. Feng, H. Shum, and S. Ma, "A novel probability model for background maintenance and subtraction," in *Proc. 15th Int. Conf. Vision Interface*, Calgary, Canada, 2002, pp. 109–116.
[37] J. Wills, S. Agarwal, and S. Belongie, "What went where," in *Proc. Conf. Computer Vision Pattern Recognition*, 2003, vol. 1, pp. 37–44.
[38] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Realtime tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.



Horng-Horng Lin (S'07) received the B.S. degree in computer science and information engineering and the M.S. degree in computer and information science both from National Chiao Tung University, Taiwan, in 1997 and 1999, respectively. He is currently working towards the Ph.D. degree at the Department of Computer Science at National Chiao Tung University, Taiwan.

Between 1999 and 2004, he served the National Defense Substitute Service on Enterprise, as a Research Assistant at the Institute of Information Science in Academia Sinica, Taiwan. His research interests include computer vision and machine learning.



Tyng-Luh Liu (M'99) received the B.S. degree in applied mathematics from the National Chengchi University, Taiwan, in 1986 and the Ph.D. degree in computer science from New York University in 1997.

He is an Associate Research Fellow of the Institute of Information Science at Academia Sinica, Taiwan. His research has focused on computer vision and pattern recognition.

Dr. Liu received the Research Award for Junior Research Investigators from Academia Sinica in 2006.

Jen-Hui Chuang (S'86–M'91–SM'06) received the B.S. degree in electrical engineering from National Taiwan University in 1980, the M.S. degree in electrical and computer engineering from the University of California at Santa Barbara in 1983, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign in 1991.

Since 1991, he has been on the faculty of the Department of Computer Science at National Chiao Tung University, Hsinchu, Taiwan, where he is currently a Professor. His research interests include robotics, computer vision, 3-D modeling, and image processing.