A Novel Probability Model for Background Maintenance and Subtraction

Dongsheng Wang† Tao Feng‡ Heung-Yeung Shum‡ Songde Ma†
† National Laboratory of Pattern Recognition, Chinese Academy of Science
‡ Microsoft Research Asia, Beijing, China
dswang@research.msrchina.microsoft.com

Abstract

"Background maintenance and subtraction" is an important problem for many computer vision applications. This paper proposes a novel model for background. This model includes two components and it processes the video sequence at pixel level and frame level alternatively. The advantage of this model is that it can capture both the temporal and spatial context of the video sequence. At pixel level, we believe that any probability model for "pixel process" can be used, at frame level, we use Markov Random Field.

And for a particular application - video surveillance on freeway, we propose a new pixel level model-adaptive HMM. In our experiments about the video surveillance on freeway, the model can solve the problems encountered: bootstrapping, gradual change of illumination, and it can detect both moving vehicles and shadows.

1 Introduction

In many computer vision applications, one basic module is "Background subtraction" which subtracts the estimated background from current image to find those pixels to be processed further. Typically, these systems have one or several fixed cameras directed at the regions interested: freeways, parking lots or the scenes wanted to be rendered. For example, in video surveillance systems (e.a. VSAM [1]), this module can find moving vehicles and people that should be identified or tracked. In real-time rendering systems (e.a. Tele-immersion [2]), the module can find the objects whose depth should be re-estimated for rendering, and so on.

Background changes sometimes. In Figure 1, illumination changes may make the difference between background and shadow becomes less noticeable, as shown in the right image. In practical systems, the difficulty is not the "subtraction" itself, but estimation of the current background. We must adapt the background model to the change of background. This procedure is called Background Maintenance: "maintenance of background model- some representation of the background and its associated statistics" - Kentaro Toyama [3]. Considering the variant cases [3] and subtle tradeoffs [4], background maintenance and subtraction is a hard problem. Several models have been proposed. We give a brief review of these models in next section.

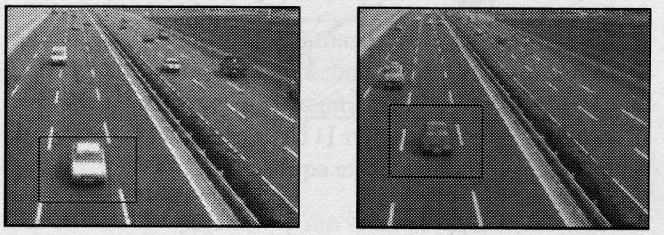

Figure 1: *Freeway background varies with illumination.*

In this paper, we propose a new model for background maintenance and subtraction. The model includes two components. The two components process the images at pixel level and frame level alternatively. The model is relatively general. Any probability model for pixel process [1] can be incorporated into our model as the pixel level component. At frame level, we use Markov Random Field (MRF). With this model, both temporal and spatial context is modelled explicitly. Specifically, we propose a new pixel level model - adaptive Hidden Markov Model (HMM). For HMM, both offline learning algorithm, which can be used for initialization, and online learning algorithm, which can be used for adaptation, are discussed. For MRF, we use Belief Propagation algorithm [5].

This paper is organized with following fashion. In section 2, we give a brief review of previous background models. In section 3, detail of our model and learning algorithm are described. Section 4 and 5 are our experiment results, discussions and future work.

2 Previous background models

Several models have been put forward for background maintenance and subtraction in the literature[1] [3] [6] [7] [8] [4]

[9] [10] . All the models for background can be roughly divided into pixel-level and nonpixel-level.

Pixel-level background models in fact are models of pixel process. The value of a particular pixel over time is called pixel process, i.e. pixel process is a time series of pixel values[1]. For a particular pixel $\{x, y\}$, the pixel process can be written as:

$$\{X_1, X_2...X_t, \} = \{I(x, y; t) : 1 \leq i \leq t\} \quad (1)$$

In $19th$ century, it was known that the background image could be obtained by exposing a film for long enough period of time[8]. It is said that the background image is average of a image sequence that is long enough. For pixel $\{x, y\}$, if $B(x, y; t)$ simplified as B_t stands for estimate of background value at time t. We have formulas

$$B_t = \frac{1}{t}\sum_{i=1}^{t} X_i \quad or \quad B_t = \frac{t-1}{t}B_{t-1} + \frac{1}{t}X_t \quad (2)$$

Moving objects can be identified simply by thresholding the distance between B_t and X_t . To handle the change of lighting condition, a moving-window average method is proposed. In the latter paper [11], exponential forgetting is used. The background update equation is:

$$B_{t+1} = (1 - \alpha)B_t + \alpha X_{t+1} \quad (3)$$

An obvious problem with this equation is that all information coming from both background and foreground is used to update the background model. If some objects move slowly, these algorithms will fail. The solution to this problem is that only those pixels not identified as moving objects are used to update background model. This is implemented by updating equation:

$$B_{t+1} = B_t + (\alpha_1(1 - M_t) + \alpha_2 M_t)D_t \quad (4)$$

where D_t is the difference between present frame and background model, and M_t is the binary moving objects hypothesis mask.

In [1], a generalization of the previous approaches was presented. Each pixel process was modelled with mixture of K Gaussian distributions. Problems like "tree moving", "moved objects"[3] were solved in a reasonable speed. The probability that a certain pixel has intensity x_t at time t is estimated as:

$$P(X_t) = \sum_{j=1}^{k} \frac{w_j}{(2\pi)^{\frac{d}{2}}|\sum_j|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t-\mu_j)^T\sum_j{}^{-1}(X_t-\mu_j)} \quad (5)$$

$$B = arg\min_b[(\sum_{j=1}^{b} w_j/\sigma_j)/(\sum_{j=1}^{K} w_j/\sigma_j) > T] \quad (6)$$

where w_j is the weight, μ_j is the mean and $\sum_j$ is the covariance for the jth Gaussian distribution. The k distributions are ordered based on w_j/σ_j and the first B distributions are the model of background.

In all of the probability models above, pixel processes are treated as data set without order. Temporal context of data is not used explicitly. In [12], general Topology Free HMM are described, and several states splitting criterions are compared. In [6], a three-states HMM without adaptation is introduced to model background.

Nonpixel-level models include Eigen-background[13], Wallflower[3]. Principle component analysis is used to determine means and variances over entire sequence (whole image as vectors) in Eigen-background. But, being not adaptive is a shortcoming of this algorithm. The idea of Wallflower is most similar to ours, it processes images at various spatial scales: pixel level (linear prediction), region level (filling) and frame level (model switch). Author [3] adopts linear prediction algorithm to model pixel process, and by updating the stored previous image data, background maintenance is accomplished, but more storage space is need.

Some nonpixel-level model can solve the problems that are thought to be inherent in pixel model [3]. This motivates us to propose a model not only in pixel-level. We will describe our model in detail in next section.

3 A novel background model

Mathematically, background maintenance and subtraction can be formulated as a labeling problem in a series of images. At any given time, any given pixel is not only one element of a particular pixel process, but also one element of image. Contextual constraint of both temporal and spatial is necessary in the robust labeling. To model the temporal and spatial contextual information, our model for background has two components, as shown in Figure 2. One component processes images at pixel level and the other one processes images at frame level.

At pixel level, we believe that any probability model for pixel process can be used. In next section, we focus our attention on video surveillance on freeway. For this particular application, we propose a new pixel process model - three states adaptive HMM. And the learning algorithms for the HMM are presented as well.

At frame level, we adopt the Bayesian inference approach to this labeling problem. We use background subtraction to determine the whole moving object. So the labeling result should be constant in a region without Label Discontinuity. The Label Discontinuity is inspired by the similar work about Depth Discontinuity in stereo matching. To handle Label Discontinuity, we introduce a spatial line process D, located on the dual lattice, and representing explicitly the presence or absence of Label Discontinuity. We will see that the D will cancel from the posterior probability

formula eventually. At any given time t, denote the image as I_t, binary spatial line process D_t and the corresponding configuration is $L_t = \{l_1, l_2 ... l_s ...\}, l_s \in \{0, 1, 2 ...\}$, and s is the pixel indices. Using Bayes' rule, we get joint posterior probability over L_t given I_t is:

$$P(L_t, D_t|I_t) = P(I_t|L_t, D_t)P(L_t, D_t)/P(I_t) \quad (7)$$

Because the observation is pixel-based, we assume that the probability above is independent with respect to D_t. Then, the posterior probability becomes

$$P(L_t, D_t|I_t) = P(I_t|L_t)P(L_t, D_t)/P(I_t) \quad (8)$$

We can define

$$P(L_t, D_t) \propto \prod_s \prod_{r \in N(s)} \exp\left(-V_c(l_s, l_r, d_{s,r})\right) \quad (9)$$

where $N(s)$ is the neighborhood of pixel s and only the cliques whose size equal to two are considered, $V_c(l_s, l_r, d_{s,r})$ is joint clique potential function of $l_s, l_r, d_{s,r}$, which is defined as :

$$V_c(l_s, l_r, d_{s,r}) = V(l_s, l_r)(1 - d_{s,r}) + \nu(d_{s,r}) \quad (10)$$

where $V(l_s, l_r)$ penalizes the different assignments of neighbor sites when no discontinuity between them, $\nu(d_{s,r})$ penalizes the occurrence of discontinuity between site s and r. The likelihood $P(I_t|L_t)$ is defined as:

$$P(I_t|L_t) = \prod_s \exp\left(-F(s, l_s, I_t)\right) \quad (11)$$

where $F(s, l_s, I_t)$ is cost function of pixel s with label l_s given observation I_t. Its value is decided by the label l_s, pixel value, and pixel process model at s.

This leads to an expensive joint estimation problem where one not only has to estimate L_t, but also the spatial line process. But fortunately, the unification of line process and robust statistics is provided by [14]. It gives us a way to eliminate the explicit binary random variable from our MAP model by defining a robust estimator whose objective function is right the posterior probability. We use such robust function:

$$\rho(l_s, l_r) = -\ln\left((1 - e)\exp\left(-\frac{|l_s - l_r|}{\sigma}\right) + e\right) \quad (12)$$

The posterior probability now is

$$P(L_t|I_t) \propto \prod_s \exp\left(-F(s, l_s, I_t)\right) \times \prod_s \prod_{r \in N(s)} \exp\left(-\rho(l_s, l_r)\right) \quad (13)$$

We convert the model with explicit label discontinuity into defining robust function that model discontinuity implicitly. Formula (13) includes a data term (the first term), which is decided by current image and pixel process model and enforces the smooth in temporal, and a regularization term (the second term), which embodies assumptions about the spatial variation of the data. The data term is often a distribution of labels computed according the pixel process model at every pixel. In section 3.4, we address this MAP problem by Belief Propagation.

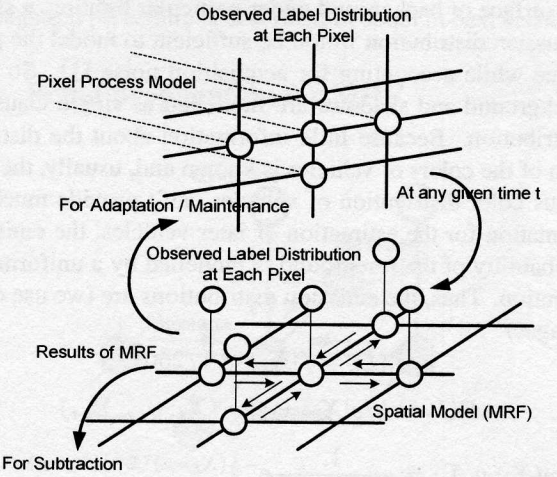

Figure 2: *The background model.*

Thus, the temporal and spatial contexts are modelled explicitly by the pixel level model and frame level model respectively. All the pixel-level processing happens at each pixel independently, i.e. the inter-pixel information is ignored at all in pixel level, and it is considered in frame level model. But we don't use the pixel level model and frame level model separately. It is easy to see in Figure 2 that they form a loop. The output of pixel level models - an array of distribution for label, not the ultimate label of decision- is treated as input of the frame level model. In frame level model, a MAP-MRF framework is used to "smooth" this array of distribution. The result of the frame level model is still an array of distribution of label. The new distribution array is used for decision of labels (for background subtraction) and for the adaptation of pixel level models (for background maintenance). We will discuss our algorithm in detail in following section.

3.1 Pixel level model

Now, let's focus our attention on a particular situation -video surveillance on freeway. Example images can be seen in Figure 1. Here we represent pixel process with HMM. Detailed information about HMM can be found in [15]. Here, we use three-states HMM, the three states stand for Background, Shadows and Foreground. Suppose that $L(x, y; t)$, simplified as L_t,is the state label of pixel (x, y) at time t, and

$L \in \{b, s, f\}$. The parameters defining the HMM model can be abbreviated as

$$\Theta_t \equiv \{A_t, \gamma_t(L), P_L(X_t) = P(X_t|L_t = b/s/f)\}$$

where $\gamma_{t(L)} = P(L_t = b/s/f|X_t)$ and $A_t = \{a_{ij}\}$, are the normalized forward variable [16] and transmit matrix at time t.

If a pixel process can be assume to result from a particular surface of background under particular lighting, a single Gaussian distribution would be sufficient to model the pixel value while accounting for acquisition noise [1]. So both background and shadows are modelled as single Gaussian distribution. Because little information about the distribution of the colors of vehicles is known and, usually, the previous color distribution of vehicles can't provide much information for the estimation of later vehicles, the emission probability of the foreground is modelled by a uniform distribution. Thus, the emission distributions are (we use color images)

$$P(L_t = f|X_t) = 1/256^3 \tag{14}$$

$$P(L_t = b/s|X_t) = \eta_{b/s}(X_t; \mu_{i,t}, \Sigma_{i,t})$$

$$\eta(X_t; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{3}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t-\mu)^T\Sigma^{-1}(X_t-\mu)} \tag{15}$$

where μ is the mean value of Gaussian distribution, and Σ is covariance matrix. We assume that the red, green, and blue pixel values are independent.

How to learn and adapt the parameters is the content of following sections.

3.2 Initialization for learning algorithm

Training an HMM, we use Baum-Welch algorithm. It is only guaranteed that you can get local maximum. Results are very sensitive to initialization. So, it is necessary that we here explain how the initialization is done.

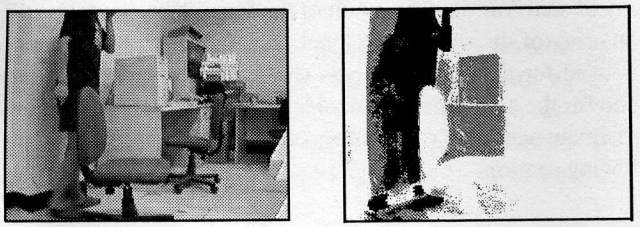

Figure 3: *Results of the initialization algorithm.*

Some work has been done about how to get a good initialization value of HMM model. Many methods use a additional learning process to estimate a Multi-Gaussian model and then use it to initialize the HMM model. Here, we use an easier method with the help of modified version of the algorithm from [10]. The algorithm is based on a heuristic color model, which separates the brightness from the chromaticity component. The pixel is labeled as background, shadow or foreground by a decision procedure whose threshold values are used to determine the similarities of chromaticity and brightness between background image and the current observed image[10]. If chromaticity difference is large, pixel is labeled as foreground, then if brightness difference is large, pixel is labeled as shadow. Figure 3 is our result of this algorithm. The result is then used to initialize our HMM model.

3.3 Learning algorithm for HMM

We use an offline Baum-welch algorithm to learn the parameters of HMM, and we use an online algorithm to adapt the parameters. At first, several assumptions are made about the training video to make the learning task easier:

- No obvious sudden change of illumination.
- The background is approximately stationary, it is to say that there is no "tree waving" problem.
- The speed of the cars doesn't vary greatly.

In a small period of time, "no obvious sudden change of illumination" is not a rigorous assumption. Allowing parts of the background to move would require us to discriminate different types of motion, which is a research topic of future. This assumption allows us to avoid adding a module of recognition of background motion. If the speed of cars doesn't vary greatly, the pixel process can be seen as a stationary stochastic process approximately, and the algorithm will be more robust relatively. For these assumptions are the nature property of video sequence on freeway, obtaining such video sequence is very easy. Detail about offline Baum-welch algorithm can be found in [16][15]. It is not our emphasis.

To maintain the background model, we use the online learning algorithm. Theoretical detail of online learning algorithm can be found in [8][17][18] as incremental version of the EM algorithm.

In online algorithm, the model parameters are updated after we got each new observation[12]. That is, the forward variables[16] in the forward algorithm are updated for each new data. Following is the detail of algorithm. The forward variables

$$\gamma_t(i) = P(O_1O_2...O_t, S = s_i|\Theta_{t-1}) \tag{16}$$

are the probability that Markov process is in state i having generated observations $O_1O_2...O_t$. For each new observation data, the γ_t are updated by adding up the probability of all possible transmit to this state multiplied a emission probability given the observation O_t:

$$\gamma'_t(j) = [\sum_i \gamma_{t-1}(i)a_{ij}]P(O_t|S_t = j) \tag{17}$$

The backward variables are setting to 1, because at time later than t, no information is known.

Now, we can compute $\xi_(t(i,j))$:

$$\begin{aligned}\xi_t(i,j) &\equiv P(S_t = i, S_{t+1} = j | O_1 O_2 ... O_t) \\ &= \gamma'_t(i) a_{ij} P(O_{t+1} | S_{t+1} = j) / \sum_k \gamma'_t(k) \quad (18)\end{aligned}$$

$\xi_{t(i,j)}$ can be seen as the expected number of transitions from sate i to j given $O_1 O_2 ... O_t$ at time t. Normalizing $\alpha_{(i)}$,we get:

$$\gamma_t(i) \equiv P(S_t = i | O_1 O_2 ... O_t) = \gamma'_t(i) / \sum_j \gamma'_t(j) \quad (19)$$

which is the probability of state i at time t given observation sequence $O_1 O_2 ... O_t$. Using the formulas above, we re-estimate model parameters incrementally. In every step, the variable above is computed and then the parameters of HMM are updated as following:

$$\begin{aligned}a_{ij}^T &= \frac{\sum_{t=1}^{T} \xi_t(i,j)}{\sum_{t=1}^{T} \gamma_t(i)} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T} \gamma_t(i)} a_{ij}^{T-1} + \frac{\xi_T(i,j)}{\sum_{t=1}^{T} \gamma_t(i)} \\ \mu_i^T &= \frac{\sum_{t=1}^{T} \gamma_t(i) O_t}{\sum_{t=1}^{T} \gamma_t(i)} = \frac{\sum_{t=1}^{T-1} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)} \mu_i^{T-1} + \frac{\gamma_T(i) O_T}{\sum_{t=1}^{T} \gamma_t(i)} \quad (20) \\ \Sigma_i^T &= \frac{\sum_{t=1}^{T} \gamma_t(i)(O_t - \mu_i^T)(O_t - \mu_i^T)^\tau}{\sum_{t=1}^{T} \gamma_t(i)} \\ &= \frac{\sum_{t=1}^{T-1} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)} \Sigma_i^{T-1} + \frac{\gamma_t(i)(O_t - \mu_i^T)(O_t - \mu_i^T)^\tau}{\sum_{t=1}^{T} \gamma_t(i)}\end{aligned}$$

The problem with this approach is that this algorithm treats all the observations equally. So it is very sensitive to initial parameter setting. It can't capture the essences of a non-stationary process. A finite time moving-window can give a solution, more efficiently, a version of algorithm with exponential forgetting is used. In this algorithm, each pixel value's contribution is weighted so as to decrease exponentially as it recedes into the past. The algorithm is implemented by:

$$R_\gamma^T(i) = (1 - \omega) R_\gamma^{T-1}(i) + \omega \gamma_T(i) \quad (21)$$

where $R_\gamma^T = \sum_{t=1,2...T} \gamma_t(i)$ and $\omega \in (0,1)$ is the time constant of forgetting or learning rate. Unlike the moving-window method, this requires no additional storage. This algorithm can be adapted to changes of background if it is given a good initialization. $\gamma_t(i)$ is distribution of label i for a given pixel at time t. In our algorithm, $\gamma_t(i)$ is not used to update the parameters of HMM directly and it is not used to make the decision about the most probable label too. We use $\gamma_t(i)$ as the evidence of Belief Propagation algorithm for MAP-MRF. The belief computed via Belief Propagation, in fact the "Smoothed $\gamma_t(i)$" is used to update HMM, and to decide the most probable labels. The Belief Propagation procedure is described in detail in next section.

3.4 Belief propagation for MRF

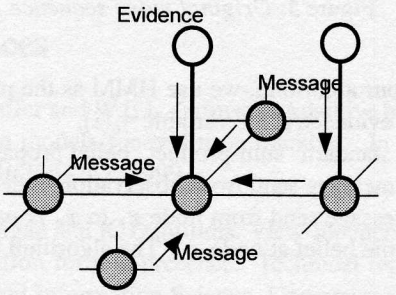

Figure 4: *Local message passing in Markov Network.*

In section 3, we get the posterior probability as following

$$P(L_t | I_t) \propto \prod_s \exp(-F(s, l_s, I_t)) \times \prod_s \prod_{r \in N(s)} \exp(-\rho(l_s, l_r))$$

This posterior probability is exactly a Markov Network in the literature of probability graph model as shown in of Figure 4. In the Markov Network, random variable l_s above is represented by hidden node x_s. A "private" observation node y_s is connected to each x_s. Each y_s is a vector, which represents the distribution of labels. And y_s is calculated from all pixel processes model and the current image. Denoting $X = \{x_s\}$ and $Y = \{y_s\}$, the posterior probability can be represented as:

$$P(X|Y) \propto \prod_s \prod_{r \in N(s)} \Psi_{sr}(x_s, x_r) \prod_s \Psi_s(x_s, y_s) \quad (22)$$

where

$$\Psi_{sr}(x_s, x_r) = \exp(-\rho(x_s, x_r)) \quad (23)$$

$$\Psi_s(x_s, y_s) \propto \exp(-F(s, x_s, I_t)) \quad (24)$$

$\Psi_{sr}(x_s, x_r)$ is called compatibility matrix between node x_s and x_r, and $\Psi_s(x_s, y_s)$ is called local evidence for node x_s. Usually, if the size of label set is n(in HMM, $n = 3$), $\Psi_{sr}(x_s, x_r)$ is a $n \times n$ matrix and $\Psi_s(x_s, y_s)$ is a $1 \times n$

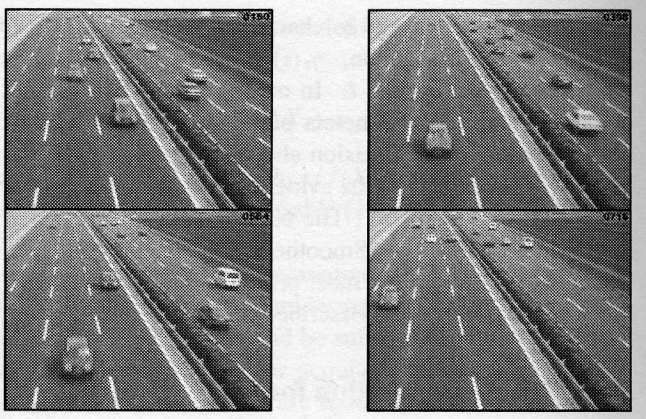

Figure 5: *Original video sequence*

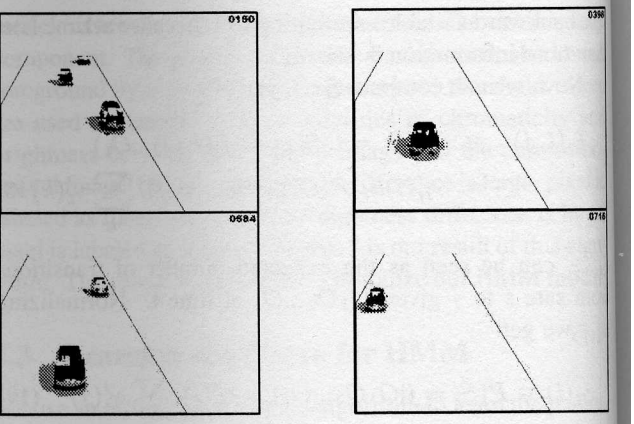

Figure 6: *Experiment results with MRF*

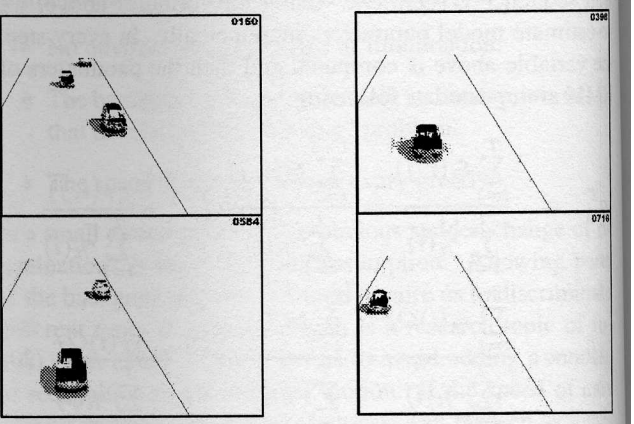

Figure 7: *Experiment results without MRF*

vector. In our algorithm, we use HMM as the pixel process model, the evidence is the variable $\gamma_t(i)$.

We use standard "sum-product" belief propagation algorithm. Let message send from observation node y_s to x_s be $m_s(x_s)$, message send from node x_s to x_r is $m_{sr}(x_r)$, and $b_s(x_s)$ are the belief at node x_s. The algorithm is described below.

1. Initialize all message with uniform distribution.
2. Update all messages iteratively $i = 1 : T$

$$m_{sr}^{i+1}(x_r) \leftarrow \alpha \sum \Psi_{sr}(x_s, x_r) m_s^i(x_s) \times \prod_{x_k \in N(x_s) \setminus x_r} m_{ks}^i(x_s)$$

3. Compute belief

$$b_s(x_s) \leftarrow \alpha m_s(x_s) \prod_{x_k \in N(x_s)} m_{ks}(x_s)$$

where α denotes a normalization constant.

Detail about belief propagation is in [5][19].

4 Experimental results

Here we present the experiments results of our model: adaptive HMM and MRF. In practice, after offline learning, the adaptation of transition matrix is not necessary, because the speed of vehicles doesn't vary greatly. So, in our experiments, only the adaptation of emission probability is implemented in online learning algorithm. Figure 6, 7, 9 are our experiment results. Because it is difficult to capture the gradual changes of illumination in freeway, we synthesized this kind of change using the video taken in the day.

In Figure 5, there are four original images. Figure 6, 7 are experiment results of background subtraction. The result images are arranged according time. The four images in Figure 7 are results of the algorithm that has no MRF to model the spatial context. We can see that our method results in Figure 6 are better. In Figure 8, the four images are original, and in Figure 9, the four images are corresponding results. Despite the illumination changed in the four original images, our algorithm can work well on all frames.

5 Conclusion and discussion

In this paper, we give a novel model for background maintenance and subtraction that can capture both temporal and spatial context of the image sequence. The model includes two components, which processes the images at two levels. For the particular condition - - video surveillance on freeway, we use adaptive HMM at pixel level. At frame level, we use belief-propagation to solve the MAP-MRF problem.

The readers should notice that the specific traffic surveillance situation is particularly suited to investigate our pixel level model - HMM. Why we focus attentions only on problems on freeway? Because we believe that **"No perfect system exists. Background maintenance and subtraction** in

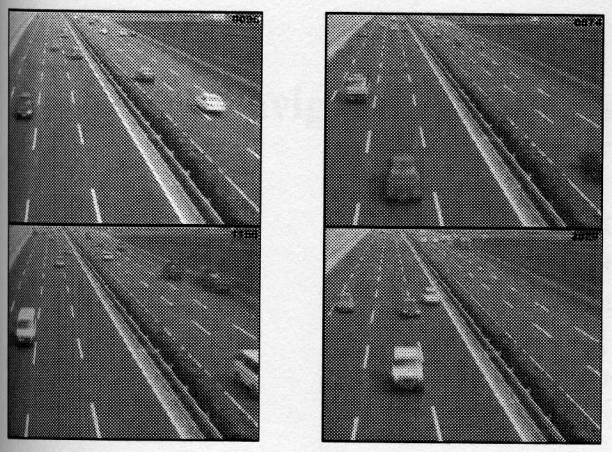

Figure 8: *Original video sequence*

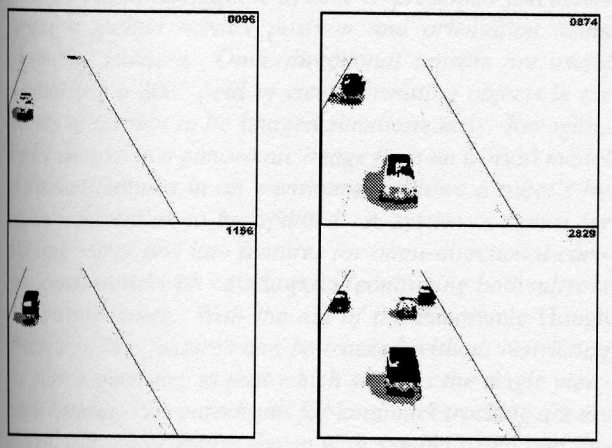

Figure 9: *Experiment results*

itself is applications oriented." Background maintenance and subtraction in itself is only a preprocessing, absolutely not the ultimate task. A perfect background maintenance and subtraction system should solve many problems, such as "bootstrapping", "moved objects", shadows, gradually and suddenly change of illumination, "tree waving", "camouflage" and so on [3]. But some of these can't be solved very well simultaneously because differentiating of them needs semantic understanding of motion of foreground and of background, and it is impossible if you have no information from the ultimate purpose. Further more, in a particular application, not all the problems will be encountered. A good system should use the knowledge derived from its purpose as possible as enough to solve the problems encountered. In our paper, we make the solving of the problems easier using priori knowledge derived from our particular application.

In section 2, many pixel-level models are summarized. But only pixel level model is not enough usually. **Both temporal and spatial context should be modelled explicitly.** In our model, the three states HMM captures the temporal context of the pixel process but omit the information inter pixels. The results by all means have some problems. Some foreground pixels are mistaken as shadow or background. We think that these problems are inherent in the pixel level model. Spatial or multi-spatial context must be considered. By modelling the spatial context using MRF, we can correct some mistakes made in pixel level and get better results.

No perfects system exists, but a good framework will give background maintenance and subtraction much help. In the future, we want to develop a framework for background maintenance and subtraction based on the principles derived in this paper.

References

[1] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR'99*, pages II:246–252, 1999.

[2] J. Mulligan and K. Daniilidis. View-independent scene acquisition for tele-presence. Technical report, Computer and Information Science, University of Pennsylvania, Philadelphia, PA, 19104.

[3] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *ICCV99*, pages 255–261, 1999.

[4] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *ECCV00*, 2000.

[5] Y. Weiss. Bayesian beliefs propagation for image understanding. In *Workshop on Statistical and Computational Theories of Vision 1999 – Modeling, Learning, Computing, and Sampling*, 1999.

[6] J. Rittscher, J. Kato, S. Joga, and A. Blake. A probabilistic background model for tracking. In *ECCV00*, pages 336–350, 2000.

[7] C. Ridder, O. Munkelt, and H. Kirchner. Adaptive background estimation and foreground de-tection using kalman-filtering. In *ICRAM'95*, pages 193–199, 1995.

[8] N. Friedman and S. Russell. mage segmentation in video sequence: A probabilistic approach. In *Annual Conf. on Uncertainty in Artificial Intelligence*, 1997.

[9] D. Gutchess, M. Trajkovic, E. Cohen-Solal, D. Lyons, and A.K. Jain. A background model initialization algorithm for video surveillance. In *ICCV01*, pages I: 733–740, 2001.

[10] T. Horprasert, D. Harwood, and L.S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *Frame-Rate99*, 1999.

[11] D. Koller, J. Weber, T.S. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. In *ICPR94*, pages A:126–131, 1994.

[12] B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J.M. Buhmann. Topology free hidden markov models: Application to background modeling. In *ICCV01*, pages I: 294–301, 2001.

[13] N.M. Oliver, B. Rosario, and A.P. Pentland. A bayesian computer vision system for modeling human interactions. *PAMI*, 22(8):831–843, August 2000.

[14] M.J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *IJCV*, 19(1):57–91, July 1996.

[15] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[16] K. F. Lee. *Automatic Speech Recognition - The Development of the SPHINX System*. Kluwer Academic Publishers, 1989.

[17] R. M. Neal and G. E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Leaning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, 1998.

[18] S. J. Nowlan. *Soft Competitive Adaptation: Neural Network Learning Algorithms based on Fitting Statistical Mixtures*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1991.

[19] J. Yedidia, W.T. Freeman, and Y. Weiss. Bethe free energy, kikuchi approximations, and belief propagation algorithms. In *SCTV01*, 2001.