Belief Propagation in a 3D Spatio-temporal MRF for Moving Object Detection

Zhaozheng Yin and Robert Collins
Department of Computer Science and Engineering
The Pennsylvania State University, University Park, PA 16802
{zyin,rcollins}@cse.psu.edu

Abstract

Previous pixel-level change detection methods either contain a background updating step that is costly for moving cameras (background subtraction) or can not locate object position and shape accurately (frame differencing). In this paper we present a Belief Propagation approach for moving object detection using a 3D Markov Random Field (MRF) model. Each hidden state in the 3D MRF model represents a pixel's motion likelihood and is estimated using message passing in a 6-connected spatio-temporal neighborhood. This approach deals effectively with difficult moving object detection problems like objects camouflaged by similar appearance to the background, or objects with uniform color that frame difference methods can only partially detect. Three examples are presented where moving objects are detected and tracked successfully while handling appearance change, shape change, varied moving speed/direction, scale change and occlusion/clutter.

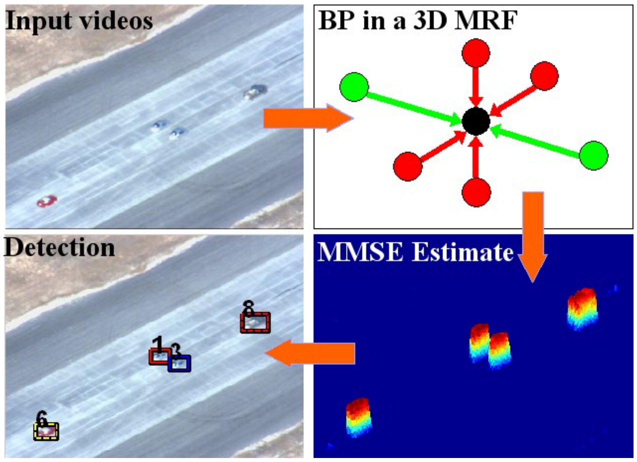


Figure 1. We have developed an approach based on Belief Propagation in a 3D spatio-temporal MRF for solving the problem of moving object detection and tracking.

1. Introduction

Belief Propagation (BP) has emerged as a powerful tool to solve early vision problems such as stereo, optical flow and image restoration using MRF models [4] [6]. Some popular inference algorithms to approximate the best estimate of each node in the MRF model are developed and compared in [13] [14]. In this paper we present a 3D MRF model for detecting moving objects in video streams, and a BP algorithm for approximate inference under this model (Fig.1). This approach overcomes the deficiencies of some common pixel-level change detection methods.

Previous pixel-level change detection methods such as background subtraction, inter-frame differencing and three-frame differencing are widely used [2] [8] [10]. Background subtraction relies on a background model for comparison, but adaptive background updating is costly for a moving camera. With stationary cameras, inter-frame differencing easily detects motion but does a poor job of localizing the object (usually only parts of the object are detected). Specifically, inter-frame differencing only detects leading/trailing edges of translating objects with uniform color. Three-frame differencing uses future, current and previous frames to detect motion but can coarsely localize the object only if a suitable frame lag is adopted. In contrast to the above methods, the motion history image (MHI) representation provides more motion properties, such as direction of motion [1]. Bobick and Davis [1] use MHI as part of a temporal template to represent and recognize human movement. MHI is computed as a scalar-valued image where intensity is a function of recency of motion. Wixson [15] presented another integration approach operating on frame-by-frame optical flow over time, using consistency of direction as a filter. In the W4 system, Haritaoglu et al. [7] used a change history map to update the background model.

Instead of using only forward temporal information, forward/backward spatio-temporal analysis on video sequences is helpful to detect and track objects. Isard and Blake [9] use past and future measurements to estimate the state distribution while smoothing out peaks due to clutter.

Luthon et al. [12] propose an energy minimization strategy in the spatio-temporal domain for object detection and build a data pyramid by 3D low-pass filtering and 3D subsampling. To find the minimum of the energy function, a deterministic relaxation algorithm ICM (Iterated Conditional Modes) is used.

With the realization that each pixel influences neighboring pixels spatially and temporally in the video sequence, we develop a 3D MRF model to represent the system. Each pixel has a hidden state node and an observed data node (e.g. inter-frame differencing result). Each state node has 6-connected neighboring nodes: four 4-connected spatial ones and 2 temporal ones (past and future). We apply the Belief Propagation algorithm to estimate the current state by spatio-temporal message passing. Spatially we perform loopy Belief Propagation [16] on each 2D frame grid. Temporally we perform 1D forward/backward BP for each pixel within a sliding temporal window. The intermediate results of our approach bear a strong resemblance to MHI representations, for example leaving a "trail" behind the moving object. However the best estimate by forward/backward spatio-temporal BP is much smoother than MHI, because our approach uses more (6-connected) neighbors for message passing than forward MHI (only the past temporal neighbor is used).

We first derive the 3D MRF model in Section 2. Belief Propagation is discussed in Section 3. To illustrate the effectiveness of this approach, we test it on challenging moving object detection cases containing camouflaged objects and objects with uniform color that are only partially detected by frame differencing. In Section 4 several tracking examples are presented where moving object detection is used to track vehicles. The tracking examples illustrate the algorithm's performance in the presence of appearance change, shape change, varied moving speed/direction, scale change and occlusion/clutter.

2. Markov Random Field Model

Markov chains defined in the time domain, e.g. Hidden Markov Models (HMM), are often used to perform sequential analysis. We note that a moving object covers a volume in the (x,y,t) space, and therefore use 3D Markov Random Fields for analyzing image sequences in the spatio-temporal domain to detect moving objects (Fig.2).

The problem of detecting moving objects in the current image is equivalent to determining whether each pixel is or is not a motion pixel based on the given video observations. In other words, we need to estimate each pixel's motion likelihood state, given observed image data. Belief Propagation is a powerful algorithm for making approximate inferences over joint distributions defined by MRF models.

Using the 3D spatio-temporal MRF model, pixel (m, n) at time instant k has a hidden state node $s(m, n, k)$ (center black node in Fig.2) and a corresponding data node $d(m, n, k)$. The network consists of nodes connected by edges that indicate conditional dependency. The state node $s(m, n, k)$ has a 3D neighborhood involving six spatio-temporal interactions. In the spatial domain, node $s(m, n, k)$ has four-connected spatial neighbors: $s(m \pm 1, n, k)$ and $s(m, n \pm 1, k)$. In the time domain, node $s(m, n, k)$ has two nearest temporal neighbors: $s(m, n, k \pm 1)$, which correspond to the past time instant $k-1$ and future time instant $k+1$ separately. The design of these cliques and nodes is described in Section 3.1. This spatio-temporal MRF model satisfies the 1st-order Markov characteristics—state node $s(m, n, k)$ only interacts with its neighboring nodes, i.e. the state of node $s(m, n, k)$ is determined if its corresponding data node, four spatial neighboring state nodes and two temporal neighboring state nodes are known. Since the neighborhood of a pixel in the current frame includes future frames, it is a bilateral MRF model and introduces a fixed lag in the detection system.

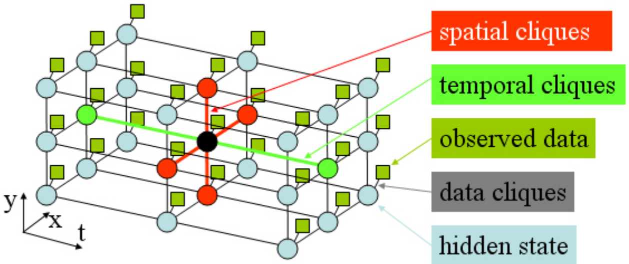


Figure 2. 3D lattice defining the neighbors of a pixel.

3. Belief Propagation

The above model contains only pairwise cliques, and the joint probability over the 3D volume is

$$P(s_1, ..., s_N, d_1, ...d_N) = \prod_{i \neq j} \psi_{ij}(s_i, s_j) \prod_k \phi_k(s_k, d_k) \tag{1}$$

where s_i and d_i represent state node and data node separately. ψ is the state transition function between a pair of different hidden state nodes and ϕ is the measurement function between the hidden state node and observed data node. N represents the total number of state or data nodes in the 3D volume. Under the squared loss function, the best estimate for node s_j is the mean of the posterior marginal probability (minimum mean squared error estimate, MMSE estimate):

$$\hat{s}_{jMMSE} = \sum_{s_j} s_j \sum_{s_i, i \neq j} P(s_1, ..., s_N, d_1, ...d_N) \tag{2}$$

where the inner sum gives the marginal distribution of s_j.

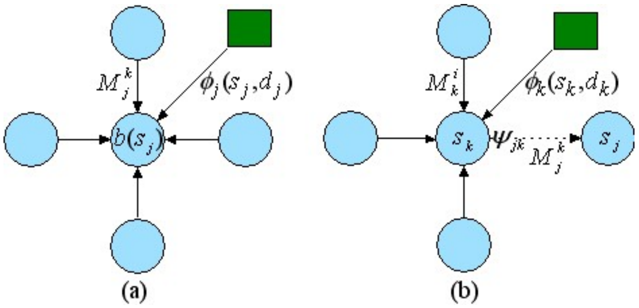


Figure 3. (a) Computing belief; (b) Computing message.

Since the joint probability involves all the hidden state nodes and data nodes in the 3D volume, it is hard to compute the MMSE estimate based on the implicit multi-variable probability distribution. However belief propagation messages are effective to compute the MMSE estimate recursively. Each hidden state node has a **belief**, which is a probability distribution defining the node's motion likelihood. Thus the MMSE estimate of one node is computed as

$$\hat{s}_{jMMSE} = \sum_{s_j} s_j b(s_j) \tag{3}$$

where

$$b(s_j) = \phi_j(s_j, d_j) \prod_{k \in \text{Neighbor}(j)} M_j^k \tag{4}$$

is the belief at node s_j and k runs over all neighboring hidden state nodes of node s_j. The belief at node s_j is the product of all the incoming messages (M_j^k) and the local observed data message ($\phi_j(s_j, d_j)$). The computation is shown in Fig.3(a). The passed **messages** specify what distribution each node thinks its neighbors should have. Fig.3(b) shows how to compute the message from node s_k to s_j:

$$M_j^k = \sum_{s_k} \psi_{jk}(s_j, s_k) b(s_k) \tag{5}$$

After substituting $b(s_k)$ by Eq.4, we have

$$M_j^k = \sum_{s_k} \psi_{jk}(s_j, s_k) \phi_k(s_k, d_k) \prod_{i \in \text{Neighbor}(k) \backslash j} M_k^i \tag{6}$$

where $i \in \text{Neighbor}(k) \backslash j$ denotes all the neighboring nodes of k other than j. After multiplying all the incoming messages (M_k^i) from neighboring nodes (except from the node s_j) and the observed data message ($\phi_k(s_k, d_k)$), the product is evolved from the message-sender to the message-receiver by transition function $\psi_{jk}(s_j, s_k)$.

3.1. Compatibility Function Selection

The hidden state s_j represents the likelihood that a pixel contains object motion, and its corresponding observation data d_j represents the binary motion detection result computed by inter-frame differencing at that time instant. We coarsely quantize each node's belief and incoming/outgoing messages into C buckets ($C \geq 2$) and describe them as a discrete probability mass function over $[1, C]$.

$\phi_j(s_j, d_j)$ describes the measurement relation (data clique in Fig.2) between observation d_j and hidden state node s_j. If d_j equals zero, no motion is detected at this pixel by inter-frame differencing, and a **uniform distribution** is used to represent "**no-motion**". However, when d_j equals one, i.e. **motion** is definitely detected by inter-frame differencing at this pixel, s_j has an **impulse distribution** at $s_j = C$ to show the confidence of existing motion regardless of what messages are passed to the node. Let s_j^p denote the pth state candidate at node s_j ($p \in [1, C]$), the compatibility function between observation and hidden state is

$$\phi_j(s_j^p, d_j) = \begin{cases} \frac{1}{C} & \text{if } d_j = 0 \\ \delta(s_j^p = C) & \text{if } d_j = 1 \end{cases} \tag{7}$$

Currently we make "hard" decisions on the data term. A "soft" frame-differencing threshold for determining motion could also be explored, i.e. the larger the intensity difference between pixels in two frames, the more likely that pixel is moving.

$\psi_{ij}(s_i, s_j)$ defines a state transition function (spatial/temporal cliques in Fig.2) between two neighboring nodes via the Potts model:

$$\psi_{ij}(s_i^p, s_j^q) = \begin{cases} \theta & \text{if } p = q \\ \epsilon & \text{otherwise} \end{cases} \tag{8}$$

where $0 < \theta < 1$, $\epsilon = (1 - \theta)/(C - 1)$ and $\theta \gg \epsilon$. This compatibility function encourages neighboring nodes to have the same states. It also acts as a decay function to reduce the motion likelihood in the absence of current intensity differences. A pixel is NOT exactly as likely to move as its neighbors, since the Potts model introduces the decay term θ. Spatial cliques and temporal cliques could have different θ values, but in our testing, we use the same compatibility function for both.

3.2. Message Computation

Fig.4 shows a 1D example to compute messages in terms of matrix operations. The message from observed data node d_2 is a vector of C elements based on Eq.7:

$$\phi_2(s_2, d_2) = \begin{cases} [\frac{1}{C} \cdots \frac{1}{C}]^T & \text{if } d_2 = 0 \\ [0 \ldots 0\; 1]^T & \text{if } d_2 = 1 \end{cases} \tag{9}$$

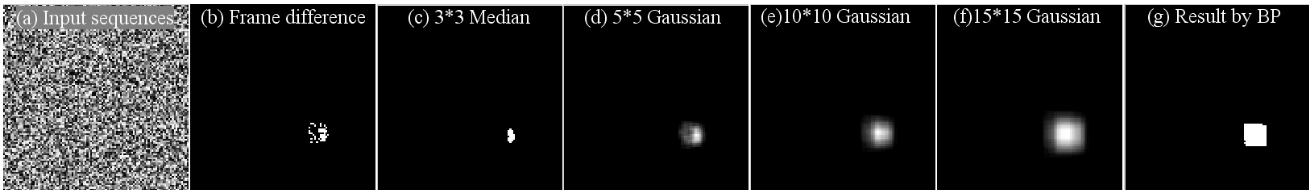


Figure 5. Test on the synthesized video sequence.

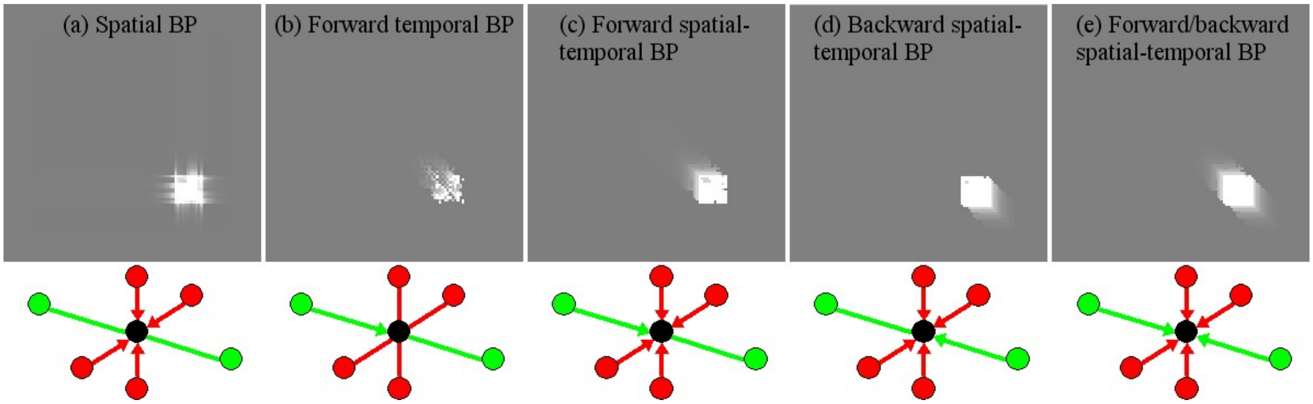

(a) Spatial BP
(b) Forward temporal BP
(c) Forward spatial-temporal BP
(d) Backward spatial-temporal BP
(e) Forward/backward spatial-temporal BP

Figure 6. MMSE estimates of random patch by different kinds of BP with $\theta = 0.7$ and $C = 10$.

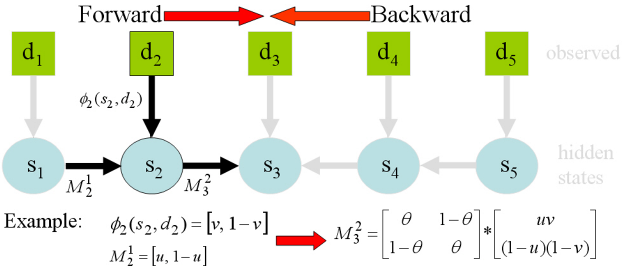

Forward
Backward
d_1
d_2
d_3
d_4
d_5
observed
$\phi_2(s_2, d_2)$
s_1
s_2
s_3
s_4
s_5
hidden states
M_2^1
M_3^2
Example:
$\phi_2(s_2, d_2) = [v, 1-v]$
$M_2^1 = [u, 1-u]$
$M_3^2 = \begin{bmatrix} \theta & 1-\theta \\ 1-\theta & \theta \end{bmatrix} * \begin{bmatrix} uv \\ (1-u)(1-v) \end{bmatrix}$

Figure 4. 1D message computation example with $C = 2$.

The state transition is a C-by-C compatibility matrix based on the Potts model of Eq.8:

$$\Psi = \begin{pmatrix} \theta & \epsilon & \dots & \epsilon \\ \epsilon & \theta & \dots & \epsilon \\ \vdots & \ddots & \ddots & \vdots \\ \epsilon & \dots & \epsilon & \theta \end{pmatrix} \tag{10}$$

The belief at node s_2 (C element vector) is computed by Eq.4:

$$b(s_2) = \phi_2(s_2, d_2) \cdot M_2^1 \tag{11}$$

where '$\cdot$' represents the element-wise product of two vectors. The message from node s_2 to s_3 is computed by Eq.6 (matrix multiplication does the marginal sum):

$$M_3^2 = \Psi * b(s_2) = \Psi * (\phi_2(s_2, d_2) \cdot M_2^1) \tag{12}$$

3.3. Message Update Schedule

To illustrate the effects of different message passing schedules, we use a synthesized image sequence as shown in Fig.5(a): the background is a 100*100 random map (each pixel has a uniform random intensity value within $[0, 255]$) and the moving object is a 10*10 random patch. It is hard for an appearance-based detector (e.g. histogram based detector) to detect the camouflaged object, but our forward/backward spatio-temporal BP algorithm easily detects object motion as shown in Fig.5(g). Based on the observed inter frame difference data (Fig.5(b)), a median filter (Fig.5(c)) or Gaussian filter (Fig.5(d-f)) can not recover the whole moving object well. The median filter removes many isolated detected motion pixels resulting in an object mask that is too small. Gaussian filters are rotationally symmetric low pass filters. They blur the differenced image, and different Gaussian window sizes and variance values cause different blur effects (thus prior knowledge about the object size is needed). Neither median filter nor Gaussian filters localize the object position accurately in Fig.5(c-f).

Each state node in our 3D MRF model has 6-connected neighbors (two temporal and four spatial ones). There are numerous ways to update the node's belief and schedule its message passing (Fig.6). For each frame (2D spatial grid graph), we perform asynchronous accelerated message updating [14]. Four 1-dimensional BP sweeps (left to right, up to down, right to left, and down to up) are performed individually and in parallel as shown in Fig.7. Here 1D BP

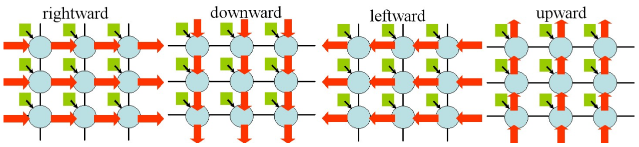


Figure 7. 2D BP on spatial grid graph.

means that BP on the spatial grid is executed simultaneously row by row, or column by column. The four 1D BP sweep beliefs (distributions for each pixel) are then fused together by '·' operation. The MMSE estimates of the "up-down-left-right" message passing (spatial domain only) are shown in Fig.6(a).

In contrast, if we only perform BP in the temporal domain, in the forward direction (forward HMM), the result is shown in Fig.6(b), which accumulates only past motion information and is quite similar in appearance to a forward motion history image. However if we simultaneously combine BP in the forward temporal and spatial domain, we get a better result than Fig.6(b), as shown in Fig.6(c). Similarly, Fig.6(d) provides the BP results combining backward temporal and spatial domain. The only difference between forward and backward temporal BP is that we maintain a sliding temporal window of length L for the latter. L future frames (fixed lag) are needed for the backward temporal BP. For example, $L = 2$ in Fig.4.

In our 3D MRF model, we use all the information from 6-connected neighbors by forward/backward spatio-temporal BP. For each frame, 2D spatial BP on a grid graph is executed by asynchronous accelerated message updating such that each pixel has a belief (motion likelihood distribution) at that time instant. The belief is then evolved to its temporal neighbor (forward or backward) by temporal BP (a 1D BP sweep). Fig.6(e) gives the MMSE estimates by forward/backward spatio-temporal BP, which can be thresholded into the final detection mask as shown in Fig.5(g).

Fig.8 illustrates another challenging motion detection problem: inter-frame differencing can only detect slivers of a moving object that has uniform color as shown in Fig.8(b). However the forward/backward spatio-temporal BP does a good job at finding the whole moving object, because each pixel propagates local frame differencing information spatially and temporally to its neighbors in the 3D MRF model. We get the above results after one iteration of BP. The algorithm currently runs about 6fps for the synthesized videos (P4 3.2GHZ, 3G RAM, Matlab). At the cost of longer runtime, using multiple iterations or larger temporal window (L) would work better, e.g. propagating the information from the edges of the object (where intensity change and motion are detected) to the center, thus filling in the hole in Fig.8(h).

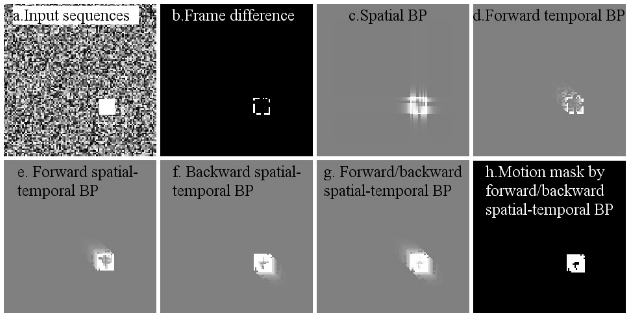

a.Input sequences
b.Frame difference
c.Spatial BP
d.Forward temporal BP
e. Forward spatial-temporal BP
f. Backward spatial-temporal BP
g. Forward/backward spatial-temporal BP
h.Motion mask by forward/backward spatial-temporal BP

Figure 8. MMSE estimates of uniform rectangle by different kinds of BP with $\theta = 0.7$ and $C = 10$.

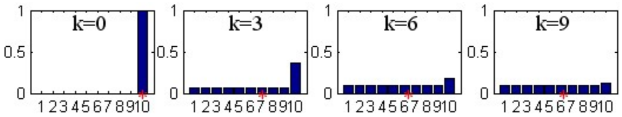


Figure 9. Belief at time instant k with $C = 10$ and $\theta = .7$; '*' denotes MMSE estimate.

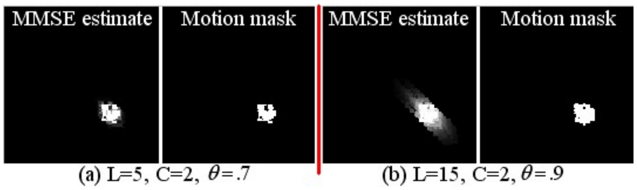
3.4. Discussion

BP in a 3D MRF vs. Motion History Image

MHI accumulates change detection results with a decay term over a short period of time. For example, if a pixel is detected as a motion pixel at time instant k, we assign that pixel value 255 in the MHI representation. With a constant decay term of h, the motion effect of this pixel will die out after $255/h$ frames if no change is detected at this pixel within the next $255/h$ frames. In our BP implementation, if change is detected at a pixel, we assign its state node an impulse distribution at the maximum belief state value. If no motion is detected around this pixel in the following frames, i.e. all neighboring nodes receive a uniform distribution belief, the center node's impulse distribution will gradually decay into a uniform distribution by transition matrix Ψ as shown in Fig.9. The resulting MMSE image has high belief values on the object with a gradually decaying trail left behind, which is quite similar in appearance to an MHI.

MMSE vs. MAP

The MMSE estimate is the mean of the posterior probability, which is known to produce smoother image results than the mode (maximum a posterior, MAP estimate) [14]. Furthermore, Fox and Nicholls [5] show the MAP state is not representative of the bulk of feasible reconstructions based on the deviation of the MAP state from the mean, particularly at larger smoothing parameters. In Fig.9, if using MAP estimation, the mode of the distribution is still at $s_j = 10$ at time instant $k = 9$, which implies this pixel is definitely a motion pixel. The MAP estimate converges to the max-



(a) L=5, C=2, θ=.7 (b) L=15, C=2, θ=.9

Figure 10. MMSE estimate with different L and θ.

imum of posterior probability and gives no importance to the tails of the distribution. However when no motion is detected during 9 frames, the motion certainty should decrease, i.e. motion likelihood of this pixel should be decreasing. Using MMSE estimation, the mean decreases from 10 (for an impulse distribution at $k = 0$) to 5.6 (mean of a roughly uniform distribution at $k = 9$). This observation shows that the MMSE estimate (mean) summarizes the desired motion likelihood distribution better than the MAP estimate (mode) in our application.

Parameter Selection

The parameters to be chosen for our algorithm are C and θ in Eqs.1-12, and sliding temporal window length L for backward temporal message passing. As shown in Fig.10, a larger window length L gives longer trails behind and ahead of the object, and it also introduces a bigger lag, i.e. more future frames are needed. While smaller L does not provide enough future temporal neighbors to do the backward temporal BP, it is wise to choose a moderate L, say $L = 10$. In Fig.9, the node's MMSE estimate (marked by '*') actually has an exponential decay process rather than the linear decay of MHI. The top row of Fig.11 shows the pixel's MMSE estimate over time with regards to different choices of C and θ. Large θ yields slow decay and gives a longer trail behind moving objects. Small θ decays the MMSE estimate quickly to the mean of uniform$[1, C]$. For given C and L, we choose a θ such that at time L the MMSE estimate decays to the mean of uniform$[1, C]$. The bottom row of Fig.11 compares motion estimation results for varying parameters. When we use two belief buckets ($C = 2$) to represent the impulse or uniform distribution, the MMSE estimate is not good enough to locate the random patch (Fig.11(a)). Choosing a larger C gives better results (Fig.11(b-d)). In our experiments, we choose $C = 10$ without incurring much computational burden.

4. Experiments

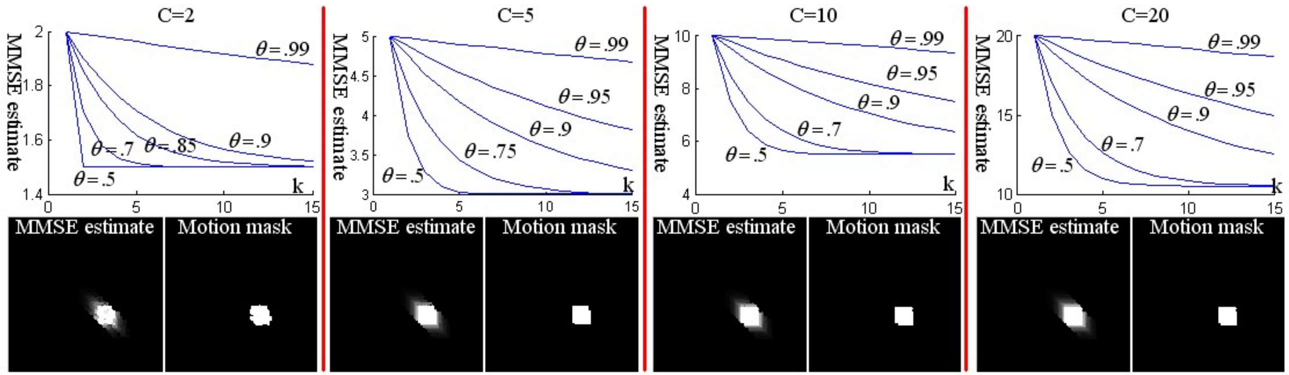
To validate the effectiveness of MRF-based motion detection, we apply it for object detection and tracking on several airborne sequences. Since the videos are captured by a moving camera, the image sequences are stabilized before frame differencing such that consecutive images have a stationary background. The stabilization uses a feature-based parametric alignment method via RANSAC. The first aerial video (total 1180 frames) shows a convoy of vehicles. The tracking results and corresponding state estimates are shown in the upper part of Fig.12. Moving objects are detected automatically in the 20th frame. Each object is given a unique numeric label. Object number one appears in the whole sequence. New objects just coming into the scene are marked by dashed rectangles. For each frame, the current state estimate (motion likelihood) is computed based on the forward/backward spatio-temporal BP. Data association between smoothed estimates and tracked objects is performed by a nearest neighbor method. As shown in the example, the object shape and appearance change a lot during the first several hundred frames, since the objects are performing a U-turn. Thus, simple template matching can not successfully track the objects. The speed and moving direction of the objects also change within the sequence, such as slowing down to turn, and speeding up to pass vehicles with similar appearance. Without any failure, the tracker using 3D MRF-based motion detection copes with the appearance change and clutter successfully.

The second video (total 1115 frames) shows a motorcycle moving with varied speed and direction. The motorcycle's scale, shape and appearance changes often and it is sometimes occluded by nearby objects. Again, the smoothed estimator using 6-connected neighbors provides good motion estimation for each pixel. We consider the smoothed estimates as an explicit weight image for tracking, and use the mean-shift algorithm [3] to find the mode representing the target. The scale of the object is automatically updated by counting the area of the most likely motion pixels of the object. When the object is totally occluded around the 1860th frame, constant velocity prediction (based on the object trail marked by green curves following the object) is applied and a search window is enlarged suitably to avoid losing the target.

The last sequence in Fig.12 (300 frames) shows a low resolution thermal video. Objects are blurred in images, or camouflaged and occluded by nearby trees and poles. It is challenging for an appearance-based tracker to track the small objects, but our forward/backward spatio-temporal BP provides a good motion likelihood image for the tracking task.

5. Conclusion

To detect moving objects within a sequence of video frames, we construct a 3D MRF model such that each pixel's hidden state (motion likelihood) has a 6-connected spatio-temporal neighborhood. Belief of motion or no-motion at each pixel is maintain by a C-state discrete probability distribution. Each hidden state is influenced by ob-



(a) L=10, C=2, θ=.85

(b) L=10, C=5, θ=.75

(c) L=10, C=10, θ=.7

(d) L=10, C=20, θ=.7

Figure 11. Top row: MMSE estimates of one pixel over time with respect to different θ's and C's; Bottom row: Comparison of MMSE estimates of all pixels and thresholded motion masks at one time instant with respect to chosen θ and C.

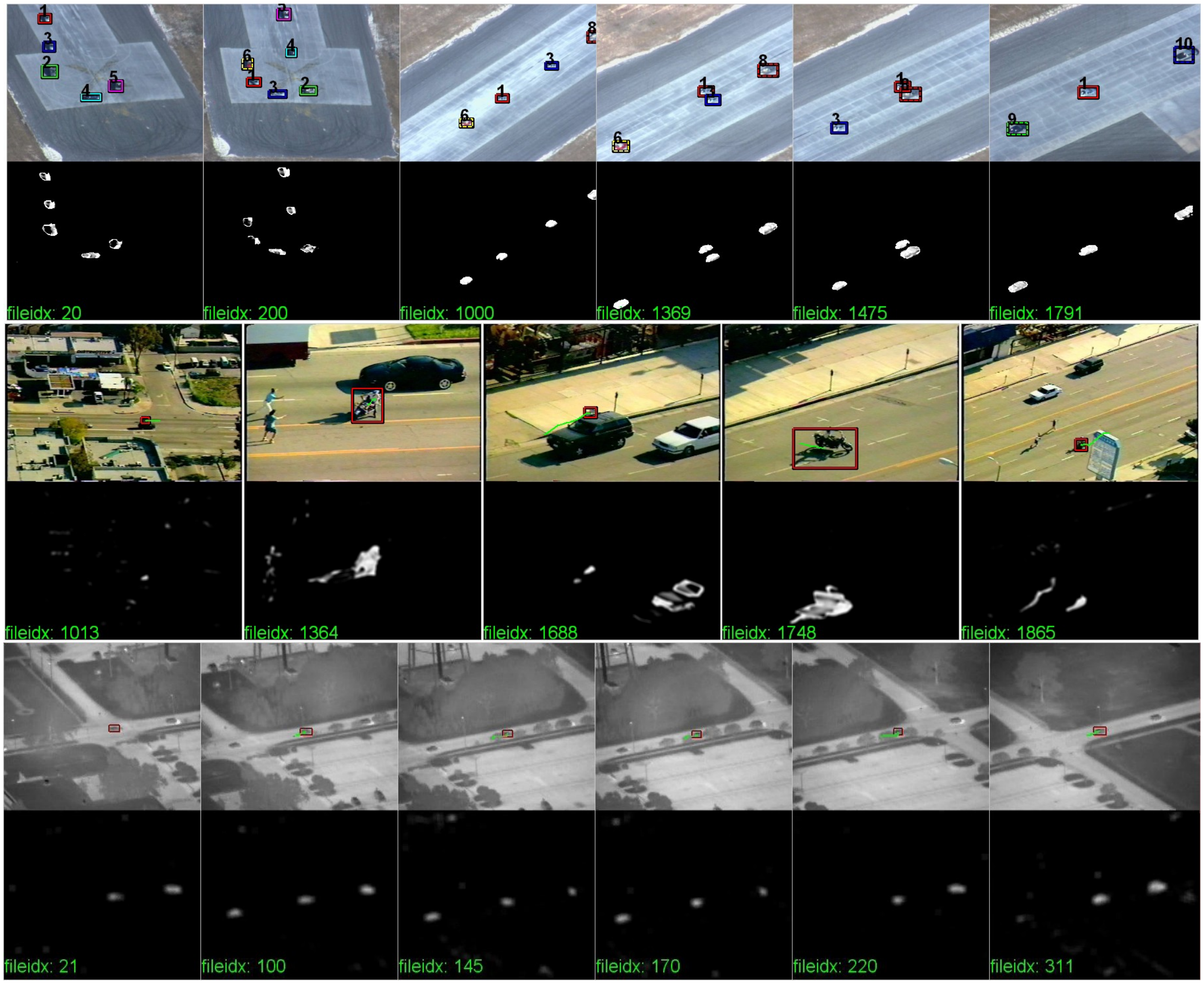


Figure 12. Experimental results on three video sequences.

served inter-frame difference data such that an observed intensity difference at a pixel introduces an impulse spike at state C of the motion likelihood, whereas no observed intensity difference results in a uniform distribution across likelihood states. A compatibility function on spatio-temporal cliques is defined via the Potts model, which in the absence of observed intensity differences at a pixel serves to decay motion belief towards a uniform distribution (representing no-motion). Within the 2D spatial grid graph, asynchronous accelerated message updating is performed on all pixels for each frame, and along the temporal axis, 1D BP (forward/backward) is applied on each pixel through time. Thus each pixel's hidden state node incorporates its six neighbor's messages to decide its state. The MMSE estimate of the state at each node provides an accurate and smooth estimate of the pixel's motion likelihood.

In this paper, the 3D MRF-based detector deals well with difficult motion detection and tracking problems such as objects with uniform color and objects camouflaged by similar appearance to the background. The approach is validated on several synthesized and real-world video sequences. So far, only motion information is smoothed to estimate the current state. In future work, other cues such as shape and appearance can be fused into the 3D MRF network and processed by Belief Propagation.

Acknowledgments

This work was funded under the NSF Computer Vision program via grant IIS-0535324 on Persistent Tracking.

References

[1] A. Bobick and J. Davis, "The Recognition of Human Movement Using Temporal Templates," IEEE Transactions Pattern Analysis and Machine Intelligence, 23(3): p257-267 March 2001.

[2] R. Collins, A. Lipton, H. Fujiyoshi and T. Kanade, "Algorithms for Cooperative Multi-Sensor Surveillance," Proceedings of the IEEE, 89(10): p1456-1477, October 2001.

[3] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-Based Object Tracking," IEEE Transactions Pattern Analysis and Machine Intelligence, 25(5): 564-577 May 2003.

[4] P. Felzenszwalb and D. Huttenlocher, "Efficient belief propagation for early vision," In IEEE Conference on Computer Vision and Pattern Recognition, p261-268, 2004.

[5] C. Fox and G. Nicholls, "Exact map states and expectations from perfect sampling: Greig, Porteous and Seheult revisited," In Proc. Twentieth Int. Workshop on Bayesian Inference and Maximum Entropy Methods in Sci. and Eng., 2000

[6] W. Freeman, E. Pasztor and O. Carmichael, "Learning Low-Level Vision," International Journal of Computer Vision, 40(1): p25-47, 2000.

[7] I. Haritaoglu, D. Harwood and L. Davis, "W4: Real-time Surveillance of People and Their Activities," IEEE Transactions Pattern Analysis and Machine Intelligence, 22(8): p809-830 August 2000.

[8] M. Irani and P. Anandan, "A Unified Approach to Moving Object Detection in 2D and 3D Scenes," IEEE Transactions Pattern Analysis and Machine Intelligence, 20(6): p577-589 June 1998.

[9] M. Isard and A. Blake. "A smoothing filter for Condensation." In Proceedings of the 5th European Conference on Computer Vision, p767-781, 1998.

[10] R. Kumar, H. Sawhney et.al, "Aerial Video Surveillance and Exploitation," In Proceedings of the IEEE, 89(10): p1518-1539 October 2001.

[11] X. Lan, S. Roth et.al, "Efficient Belief Propagation with Learned Higher-Order Markov Random Fields." In Proceedings of the 9th European Conference on Computer Vision, p269-282, 2006.

[12] F. Luthon, A. Caplier and M. Lievin, "Spatiotemporal MRF approach to video segmentation: application to motion detection and lip segmentation." Signal Processing, 76(1): p61-80, July 1999.

[13] R. Szeliski, R. Zabih et.al, "A Comparative Study of Energy Minimization Methods for Markov Random Fields." In Proceedings of the 9th European Conference on Computer Vision, p19-26, 2006.

[14] M. Tappen and W. Freeman, "Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters." In: ICCV, p900-907, 2003

[15] L. Wixson, "Detecting Salient Motion by Accumulating Directionally-Consistent Flow, " IEEE Transactions Pattern Analysis and Machine Intelligence, 22(8): p774-780 August 2000.

[16] J. Yedidia, W. Freeman and Y. Weiss, "Generalized belief propagation." In: NIPS, p689-695, 2000