A Loopy Belief Propagation Approach for Robust Background Estimation

Xun Xu and Thomas S. Huang
Beckman Institute, University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
{xunxu,huang}@ifp.uiuc.edu

Abstract

Background estimation, i.e. *automatic recovery of the background image from a sequence of images containing moving foreground objects, is an important module in many applications,* e.g. *surveillance and video segmentation. In this paper, we present a simple, yet effective and robust approach for background estimation based on Loopy Belief Propagation. Robustness of the proposed approach means: (i) minimal assumption on the input frames, and (ii) no need to tune parameters. Basically, the background can be recovered even when the occluding foreground objects stay still for a long time. Furthermore, no motion information needs to be known or estimated for the foreground objects, which implies that background can be recovered from a set of frames which are not consecutive temporally. Analysis and experiments are provided to compare the proposed approach to related methods. Experimental results on typical surveillance videos demonstrate the effectiveness of our approach.*

1. Introduction

Background estimation, *i.e.* automatic recovery of the background image from a sequence of images containing moving foreground objects, is an important module for many video processing applications, *e.g.* intelligent surveillance and video segmentation. In intelligent surveillance, background estimation is essential for high quality background subtraction [7, 5]. In video segmentation, background provides rich information to extract foreground objects. Recently, effective algorithms [10, 4] have been proposed for real-time foreground/background segmentation. However, background must be provided as an input for these algorithms. Another application of background estimation is in computational photography, where the user wants to obtain a clean background plate from a set of input images containing cluttering foreground objects [1].

If it is safe to assume that the background is stationary, and every background pixel is disclosed for more than 50% time of the sequence, then a simple per-pixel median filter is sufficient to obtain a fairly good background estimation. However, in practice the second assumption is often, if not always, violated. For instance, in surveillance scenario, cars in the camera's field of view may pull over for quite a while, or be stuck in a heavy traffic jam. Unfortunately, many well known background subtraction techniques [7, 5] more or less rely on the assumption that the background would be occluded only for a small portion of time. In [7] background is modeled probabilistically at each pixel location by fitting a GMM to pixel values observed in a recent temporal window. Therefore if a foreground object persists on the same location sufficiently long, the background model will be overfitted to this object, absorbing it into the background. [5] models background in a similar way, but using a more flexible, nonparametric probabilistic model. Although it is capable of modeling the distribution of background pixels more precisely, it still has the same problem as [7].

In this paper we show that background estimation can be modeled as a low level vision problem, formulated under the energy minimization framework, and solved with Loopy Belief Propagation. This leads to a simple, yet effective and robust approach for background estimation, in the sense that it makes minimal assumption on the input frames, and waives the need to tune parameters. With the proposed approach, the background can be estimated if each background pixel is revealed at least once, and no motion information needs to be known or estimated from the input frames. This means that our approach will not absorb the foreground object into the background even if it stays still for a long time. Meanwhile, the proposed approach applies to input frames sampled at large time interval, or even frames with no temporal correlation.

In brief, our approach works by composing a visually smooth background image using pixels picked from input frames. In literature there are relevant works [1, 3] which also recover the background by enforcing the visual smoothness constraint. In this paper, a detailed comparison to these methods will be presented, analyzing the uniqueness and advantages of our approach.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

The paper is organized as follows. In Section 2 our approach for robust background estimation is presented. In Section 3, we compare the proposed approach to related existing methods, analyzing their differences in detail. Experimental results are then presented in Section 4, comparing the result of our approach to other representative methods. Finally we conclude the paper and discuss future work in Section 5.

2. Approach

2.1. Background modeling

Our goal is to estimate background with minimal assumption made on the input frames. We simply assume that the input frames have a common, stationary background, and the background is disclosed at least *once* at each pixel. Since it is *not* assumed that each background pixel is occluded by foreground objects for only a small portion of time, and nothing is known about the motion of foreground objects, the only property we may expect from the background is that it should contain less occluding boundaries than an image involving foreground objects. In other words, the background should be visually smoother than the latter. Therefore, we define the background as an image with best visual smoothness obtained by composing pixels from input frames. To this end, background estimation is modeled as a pixel-labeling problem formulated under the energy minimization framework. That is, we have an ensemble of input frames $\{I_l, l \in \mathcal{L}\}$ from which a background image B is to be estimated, for each pixel $p \in \mathcal{P}$ we assign to it a label $L_p \in \mathcal{L}$ indicating which input frame this pixel should be selected from. The optimal labeling scheme is the one that minimizes the function encoding visual smoothness:

$$E_s = \sum_{(p,q) \in \mathcal{N}} \left\| I_{L_p}(p) - I_{L_q}(q) \right\|, \tag{1}$$

where $\mathcal{N} \subset \mathcal{P} \times \mathcal{P}$ is the set of all pairs of neighbor pixels and $\|\cdot\|$ refers to the l_1 norm. It essentially means that we want a *piecewise smooth* image composed of pixels from input frames.

This energy function might seem prone to resulting in an over smooth estimation of background. However, it is not. This is because although visual smoothness is sought, the pixels filled into the estimated background must all come from original frames. Therefore, if there is an edge in the background portion of some input frames, it would still be there in the estimated background. It can be observed from the experimental results in Section 4 that edges in the background are generally preserved well. An exception occurs when our assumption on the input frames is partly violated, *e.g*. the luminance level varies across frames (due to the camera's gain control, for instance), so that the backgrounds in different frames are not exactly the same (but differ in intensity). In such cases, some fine edges may be blurred out in the estimated background. However, this effect can be compensated by the gradient domain composition technique, as we shall see in Subsection 2.4.

One may also argue that a foreground object with smooth appearance may be erroneously selected as background. However, although the object itself is smooth, so long as it is distinct from the surrounding background, it will generate large energy terms along the occluding boundary. Since our energy function is *global*, it could hardly be minimized if such erroneous selection occurred.

2.2. Energy minimization with Loopy Belief Propagation

The energy function E_s can be regarded as Gibbs energy of a Conditional Random Field [9] with zero data term. Two well-known algorithms for minimizing such energy functions are Graph-Cuts (GC) [2, 8] and Loopy Belief Propagation (LBP) [12]. GC is usually regarded fast and has some nice theoretical guarantee on the optimality of the solution it can find. However, as our interaction energy term is neither a metric nor a semimetric, Graph-Cuts algorithm [2] is not suitable here. Hence, Loopy Belief Propagation becomes a natural choice.

LBP has two variants, namely max-product and sum-product. Here we work with max-product which computes the MAP label for each pixel, and as we are minimizing a Gibbs energy which is essentially negative log probability, max-product becomes min-sum. The algorithm works by iteratively passing messages across pixels (nodes), briefed as follows:

1. For all neighbor pixel pairs $(p, q) \in \mathcal{N}$, initializing messages m^0_{pq} to zero.
2. For $t = 1, 2, \ldots, T$, updating the messages as:

$$\begin{aligned} m^t_{pq}(L_q) = \min_{L_p} \Bigg[& \left\| I_{L_p}(p) - I_{L_q}(q) \right\| \\ & + \sum_{s:(s,p) \in \mathcal{N}, s \neq q} m^{t-1}_{sp}(L_p) \Bigg] \end{aligned} \tag{2}$$

3. Determining labels as:

$$L^*_q = \arg\min_{L_q} \sum_{p:(p,q) \in \mathcal{N}} m^T_{pq}(L_q) \tag{3}$$

As for implementation, we use a multi-scale version of above LBP algorithm, as suggested in [6], which significantly reduces the total number of iterations needed to reach convergence. Briefly speaking, the message updating process is coarse-to-fine, *i.e*. we start LBP from the coarsest grid, after a few iterations transfer the messages to a finer scale and continue LBP there, so on and so forth.

2.3. Additional energy function for acceleration

The basic approach introduced so far has the capability of recovering background with minimal requirement for the input frames. However, because energy function (1) does not contain any data term, at each iteration the only information utilized to update messages is the interaction between neighboring pixels. As a result, the efficiency of message updating is relatively low. This suggests adding an appropriate data term, which supplies some heuristic information about the background, to speed up the energy minimization process. To this end, we define an additional energy function

$$E_d = \sum_p \min_{l \neq L_p} \left\| I_{L_p}(p) - I_l(p) \right\| \tag{4}$$

This function relates to the assumption that the background is stationary, therefore we favor a pixel to be labeled as background if at the same location of some other frame(s) there are pixel(s) of identical color. Note that the introduction of this additional energy function puts a little bit stronger assumption on the input frames, *i.e.* the background should be disclosed at least *twice* at each pixel. In video surveillance scenario, this assumption is still not demanding.

It should be pointed out that, alternatively one might consider defining the data term as something related to foreground motion, such as the magnitude of optical flow between consecutive frames. On the contrary, our data term defined in (4) still does not explicitly rely on any motion information. This implies that our approach applies to input frames sampled at rather large temporal interval where motion between consecutive frames could not be accurately estimated, or even frames with no temporal correlation at all. This again reflects our attempt to minimize the assumption on the input frames.

Now the total energy function reads:

$$E = E_s + \lambda E_d, \tag{5}$$

where λ is a weighting parameter. Accordingly, the message update equation (2) changes to

$$m_{pq}^{t}(L_q) = \min_{L_p} \Bigg[\left\| I_{L_p}(p) - I_{L_q}(q) \right\| + \lambda \min_{l \neq L_p} \left\| I_{L_p}(p) - I_l(p) \right\| + \sum_{s:(s,p) \in \mathcal{N}, s \neq q} m_{sp}^{t-1}(L_p) \Bigg], \tag{6}$$

and (3) becomes

$$L_q^* = \arg\min_{L_q} \left[\lambda \min_{l \neq L_q} \left\| I_{L_q}(q) - I_l(q) \right\| + \sum_{p:(p,q) \in \mathcal{N}} m_{pq}^{T}(L_q) \right] \tag{7}$$

The introduction of E_d supplies more information about how likely a particular pixel should be labeled as background so that messages can be updated more effectively, hence speeds up the energy minimization process. To validate this, we plot in Figure 1 how the visual smoothness energy E_s converges with ($\lambda = 1$) or without ($\lambda = 0$) E_d, while estimating the background of sequence MO (see Figure 5 for input frames) with single-level LBP.

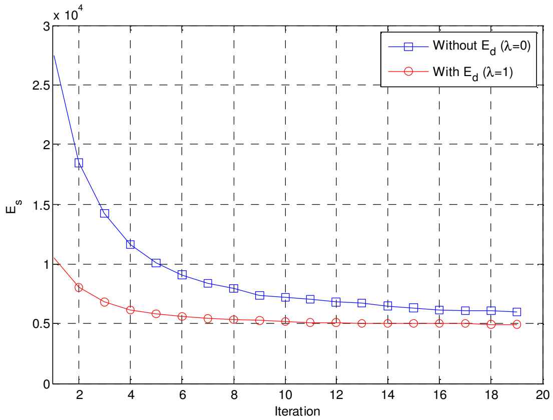
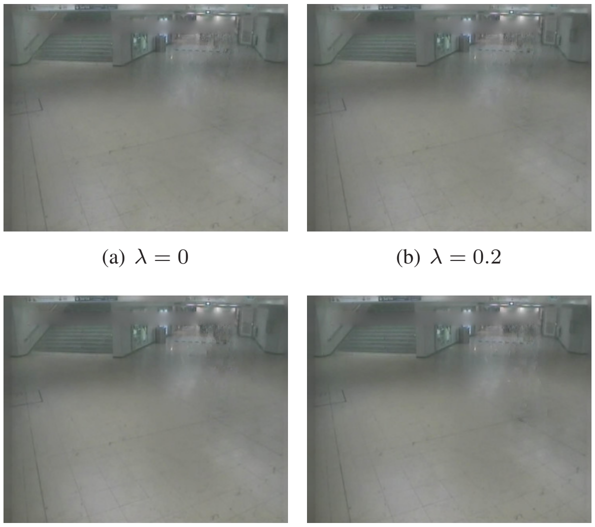


Figure 1. The evolution curves of E_s with or without additional energy function E_d: including E_d accelerates energy minimization.

The introduction of E_d also brings a weighting parameter λ which needs to be set. Fortunately, because the visual smoothness is completely encoded in E_s, whereas E_d exists mainly to accelerate the energy minimization process, the estimated background is quite insensitive to the choice of λ. In Figure 2 background images of sequence MO estimated with different λ's are displayed. It can be seen that the estimated background images are almost identical while λ varies. This indicates that there's almost no need to tune any parameter for the proposed approach. One exception, however, is the case where the assumption under which E_d is introduced is violated, *i.e.* some part of the background is indeed disclosed *only once* across all the input frames. In surveillance or video processing scenario this rarely happens. For all the experimental results reported in Section 4, we set $\lambda = 1$. However, when this does happen in some scenario (*e.g.* computational photography as in [1]), $\lambda = 0$ is a better choice, at the cost of slower convergence.

2.4. Background composition: image domain vs. gradient domain

After the energy minimization process ends, we obtain a MAP label for each pixel indicating from which frame the pixel should be selected from. A straightforward way to reconstruct the background could be simply putting the selected pixels together, *i.e.* $B(p) = I_{L_p}(p)$, which we call image domain composition. Alternatively, gradient domain



(a) $\lambda = 0$ (b) $\lambda = 0.2$

(c) $\lambda = 1$ (d) $\lambda = 2$

Figure 2. Estimated background of sequence MO with different λ's.

composition suggested in [1] can be used. In short, the gradients of every pixel, instead of the pixel values, are collected from original frames and put together as a composite gradient field, then the background image, with gradient field closest to the composite one, is reconstructed by solving a discretized Poisson equation.

Since gradient domain composition recovers the background image up to an additive constant (for each color channel), this constant must be determined. In [1], the user is required to specify the color of a pixel in the final composite image. In our system, the additive constant is estimated in a least squares fashion as:

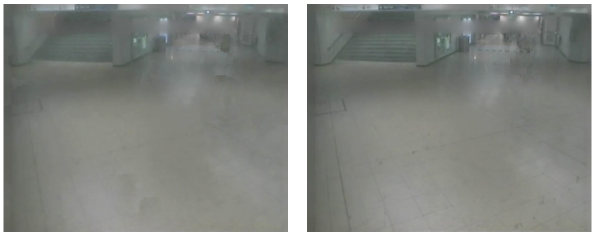
$$s^* = \arg\min_s \sum_p \left\| I_{L_p}(p) - B_G(p) - s \right\|^2, \qquad (8)$$

where B_G is the background obtained from gradient domain composition with arbitrary additive constant (*e.g.* setting the first pixel's value to be 0). s^* has a simple form and the final background is given by:

$$B(p) = B_G(p) + \tfrac{1}{N} \sum_{p'} \left[I_{L_{p'}}(p') - B_G(p') \right], \qquad (9)$$

where N is the number of pixels in the image. Note that the second term is simply the average difference between image domain composition and gradient domain composition. Instead, one may consider employing robust statistics to estimate the additive constant, *i.e.* replacing the square terms in (8) with more robust error measure, so as to reduce the effect of outlier pixels, at the cost of more computation. In all our experiments, however, this simple technique worked well.

In our scenario, gradient domain composition has two advantages. First, it takes care of small intensity discrepancy among input frames, achieving a visually more consistent output. Meanwhile, it helps better preserve the edges in the background, as the edge information (*i.e.* gradients) from original frames are directly taken into account in the composition. In Figure 3, we display two background images of sequence MO reconstructed respectively with image and gradient domain composition. It can be observed that gradient domain composition results in better visual quality. Specifically, the fine edges of the stairs are sharper in the gradient domain composition; whereas they are somewhat blurred (due to global luminance variation across frames) in the image domain composition.



(a) Image domain composition (b) Gradient domain composition

Figure 3. Image domain vs. gradient domain composition

It should be noticed that, employing which composition method depends on application. For surveillance applications such as background subtraction, the background recovered with image domain composition, which is computationally much cheaper, is usually satisfactory enough. For applications where high visual quality is a concern, such as computational photography, gradient domain composition is a better choice.

3. Discussion

In literature of most recent years, there are some works on background estimation taking visual smoothness into account. It is helpful to compare those methods to ours in order to gain a deeper understanding on both.

In [1] Agarwala *et al.* proposed a unified, energy minimization based framework for interactive image composition, under which various image editing tasks can be done by choosing appropriate energy functions. Following the conventional formulation of pixel-labeling problems, their cost function consists of data term and interaction term: the interaction term penalizes perceivable seams in the composite image, whereas the data term reflects various objectives of different image editing tasks. Under their framework, two data terms are dedicated to background reconstruction, namely the "maximum likelihood" and "minimum contrast" image objectives. The "maximum likelihood" objective, although able to achieve visually more consistent composition than a naïve per-pixel maximum likelihood approach, still relies on the assumption that background is occluded only for a small portion of time. The latter objective, *i.e.* "minimum contrast", is relevant to our approach, as it also reconstructs background by seeking visual smoothness.

Our approach might seem similar to that of [1] as both formulate background estimation as a pixel-labeling problem under energy minimization framework. However, the difference is indeed fundamental. Recall that our approach encodes visual smoothness in the interaction energy function E_s, and incorporates a data energy function E_d for acceleration. However, in [1] visual smoothness is modeled in the "minimum contrast" data term, while the interaction term aims at reducing the seam artifacts. When these two terms are combined into the total cost function after weighting the data term by a parameter μ (apparently $\mu = 1$ in [1] from their equation 1), it can be found that the choice of μ is rather hard. When μ is large and the data term dominates, pixels with small gradients (*i.e.* located within a smooth region) tend to be selected, even if they indeed come from the foreground. This causes smooth segments of the foreground to be included into the estimated background. On the other hand, if μ is too small and the interaction term dominates, the visual smoothness constraint could not be effectively enforced. As a result, foreground objects may appear in the estimated background as well. As the extreme case, when $\mu = 0$ any input frame leads to zero cost hence can be outputted as the estimated background. In Figure 4 the background images estimated with different μ's are shown. Unlike our approach's robustness to λ (see Figure 2), the choice of μ significantly affects the result, and it is really hard to find a μ resulting in good background estimation.

Another interesting approach to background estimation considering visual smoothness is due to Colombari *et al.* [3], where input frames are divided into overlapping patches and the background is reconstructed by incrementally inserting image patches while enforcing the visual smoothness constraint. Like ours, their approach also aims at recovering background even when it is occluded for considerable portion of time. However, although it is claimed in [3] that the background can be reconstructed so long as it is revealed once at each pixel, a more precise statement of this assumption should be "among patches along the same timeline the background is completely revealed at least once", as the estimated background is composed of patches, not pixels. It could be noted that this is a much stronger assumption than ours. In many surveillance scenarios, especially crowded scenes, it is usually hard to find a completely revealed background *patch*, whereas finding disclosed background *pixels* is far easier. Another essential difference between [3] and our approach is that the former enforces visual smoothness *locally*, growing the background incrementally by inserting a patch at each iteration. A potential problem is: once a patch containing foreground is erroneously inserted, the error may propagate further, including more and more of that foreground object. On the contrary, our approach enforces visual smoothness *globally* by minimizing an energy function of all the pixel labels as a whole, hence is more robust.

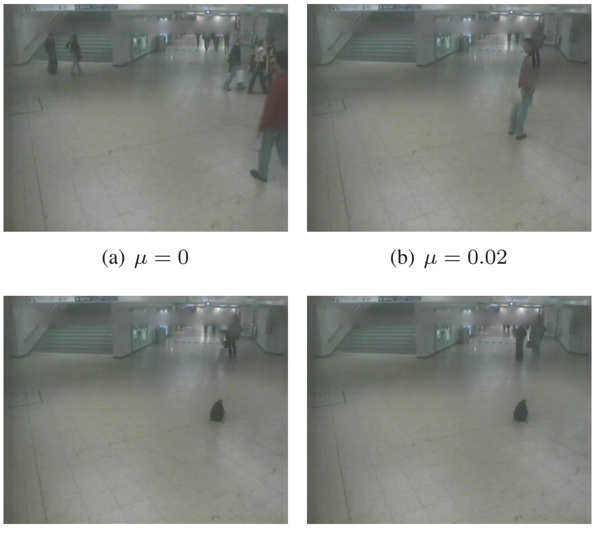

(a) $\mu = 0$
(b) $\mu = 0.02$
(c) $\mu = 0.05$
(d) $\mu = 1$

Figure 4. Background estimated with the cost function of [1], using "minimum contrast" image objective weighted by different μ's.

4. Experiments

In this section we present experimental results of the proposed approach, and compare it to related methods. The experiments were conducted on 5 video sequences (provided by the ETISEO project[1]), namely MO, RD, ST, BE and BC. 6∼15 frames are sampled from these video sequences, at a temporal interval of 3 seconds, and scaled to a size around 320×240. 3 sample input frames for each video sequence are shown in Figure 5. Note that these sequences cover various typical surveillance scenarios, *e.g.* road, parking lot and indoor environment *etc.*

A MATLAB implementation of the multi-scale LBP algorithm [6] was used to minimize energy function (5) ($\lambda = 1$). For all the sequences, LBP was run for 5 iterations at each of 6 scales, in a coarse-to-fine fashion. To compare with the Photomontage approach [1], the same LBP algorithm with same parameters was employed to minimize their cost function (with "minimum contrast" image objective and "colors and gradients" seam objective). Note that Graph-Cuts was used instead in [1], but here the goal of

[1] http://www-sop.inria.fr/orion/ETISEO/

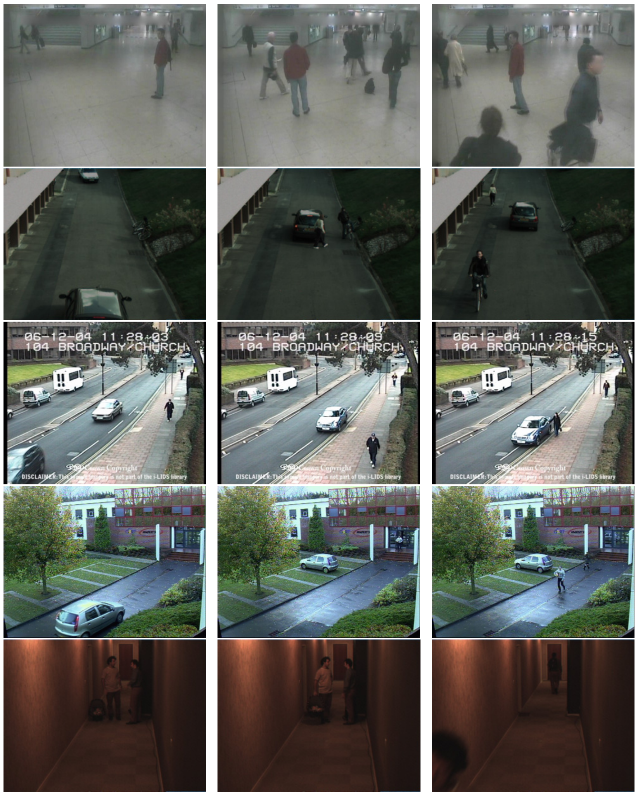


Figure 5. Sample frames of video sequences used in our experiments. Rows from top to bottom: MO, RD, ST, BE, BC.

comparing the two approaches is to study how their energy functions differ, therefore minimizing them with the same algorithm is more appropriate. It should also be mentioned that for results in Figure 6, image domain composition was used for both approaches in order to prevent gradient domain composition from concealing the artifacts, so that all the methods can be compared fairly.

The Maximum Likelihood method is essentially [5], as we employ their density estimation approach at each pixel location. The background image is obtained by putting together pixels which are most likely to be sampled from the background.

From results shown in Figure 6 it is clear that methods assuming the background to be occluded only for a small portion of time, like per-pixel median filter or the maximum likelihood method, often work poorly in practice as it is common for foreground objects to persist on the same location for quite a while, *e.g.* cars pulling over by the curb. Approaches considering visual smoothness, including ours, work better. However, the patch tessellation method [3] does not work very robustly, especially for crowded scenes like that in sequence MO. Photomontage [1] is more robust, usually resulting in visually consistent results. However, it sometimes includes foreground objects with smooth appearance, *e.g.* the bag in sequence MO and the pedestrian in sequence ST. Our approach works almost consistently well, except for sequence BC which shows a case for which our approach fails. Specifically, the foreground person occludes two strong edges in the background, forcing our algorithm to absorb him into the background to obtain a visually more smooth output. For this sequence, indeed none of the other compared methods gave a more impressive output.

5. Conclusion and future work

In this paper we have shown that background estimation can be modeled as a low level vision problem, formulated under the energy minimization framework, and solved with Loopy Belief Propagation. This leads to a simple but effective approach for robust background estimation, in the sense that it makes minimal assumption on the input frames, and waives the need to tune parameters. Specifically, the background can be recovered if each pixel is revealed at least once, meanwhile no motion information needs to be known or estimated from the input frames. When each background pixel is disclosed twice or more, background can be estimated more efficiently.

One practical concern of employing LBP algorithm is its speed. In our experiments, typically the background can be estimated in 2.5 minutes from 10 input frames of size 320×240, with a non-optimized MATLAB implementation on a regular desktop PC (Pentium IV 3GHz). We employed the multi-scale scheme in [6], which significantly reduces the number of iterations needed by LBP to reach convergence. Other techniques, such as bipartite graph [6] and asynchronous message updating [11], may also be employed for further acceleration. On the other hand, since the algorithm is highly parallel in nature, implementation on high performance parallel processors (such as GPU) would be hopeful to achieve a great speed boost.

Our approach currently assumes that the background is stationary, extending it to handle non-stationary background is a relevant topic for future research. This involves two cases: (1)The camera itself is in motion. (2)Some part of the background is undergoing local motion, *e.g.* tree branches waving in the wind. The former case may be handled by employing global image registration techniques to compensate the camera motion. The latter case is more challenging, probably requiring a novel representation of the background.

Acknowledgment

This research was funded in part by the U. S. Government VACE program.

References

[1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interac-

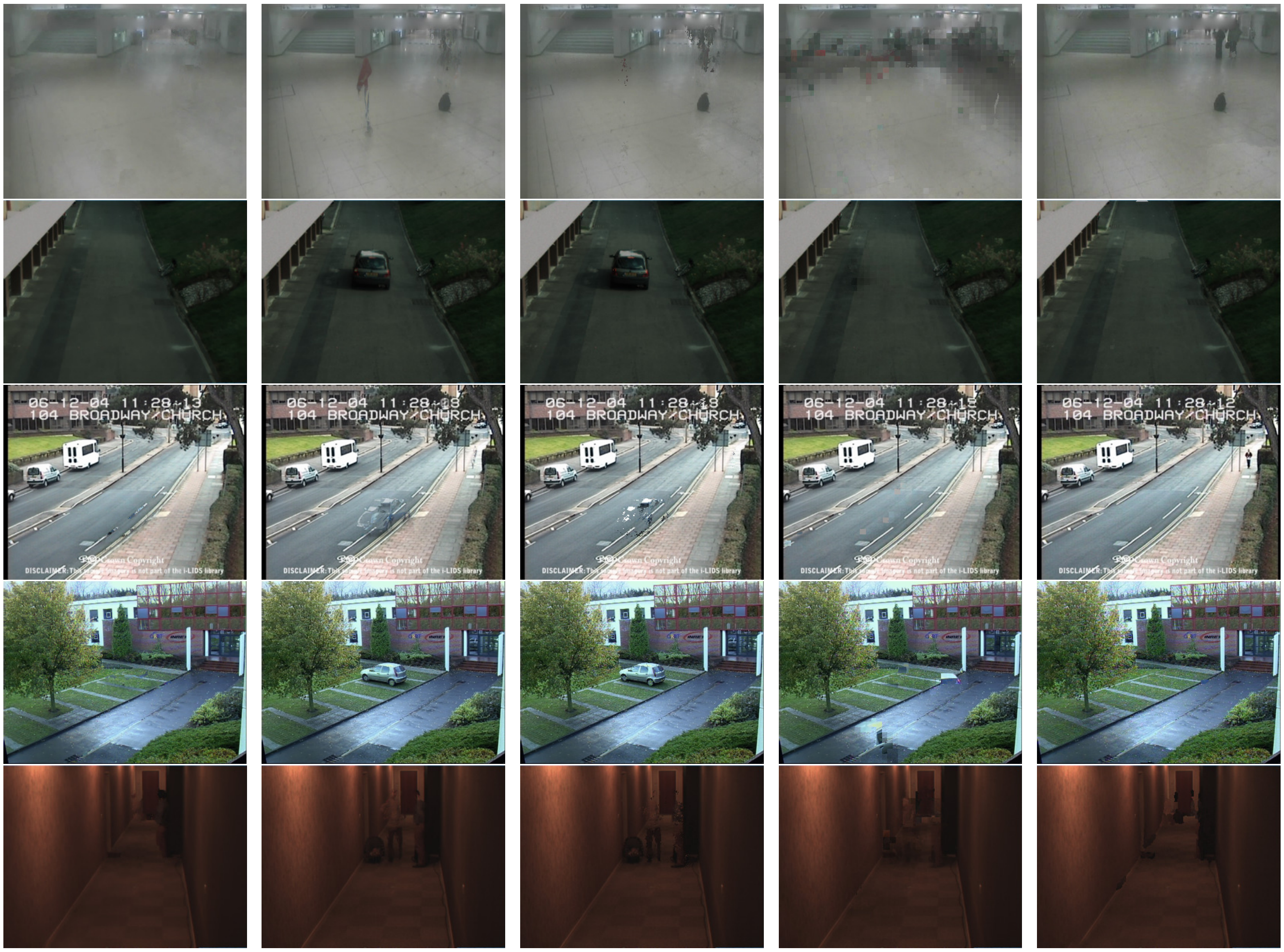


Figure 6. Background estimation with different approaches. Rows from top to bottom: MO, RD, ST, BE, BC. Sample frames of these sequences are shown in Figure 5. Columns from left to right: (a) Proposed approach (b) Median (c) Maximum likelihood [5] (d) Patch tessellation [3] (e) Photomontage [1]

tive digital photomontage. In *SIGGRAPH 2004*, pages 294–302. ACM Press, 2004. 1, 3, 4, 5, 6, 7

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001. 2

[3] A. Colombari, A. Fusiello, and V. Murino. Background initialization in cluttered sequences. In *POCV 2006 (in conjunction with CVPR)*, page 197. IEEE Computer Society, 2006. 1, 5, 6, 7

[4] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer segmentation of live video. In *CVPR 2006*. IEEE Computer Society, 2006. 1

[5] A. M. Elgammal, D. Harwood, and L. S. Davis. Nonparametric model for background subtraction. In *ECCV 2000*, pages 751–767, London, UK, 2000. Springer-Verlag. 1, 6, 7

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. In *CVPR 2004*, volume 1, pages 261–268. IEEE Computer Society, 2004. 2, 5, 6

[7] W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *CVPR 1998*, page 22. IEEE Computer Society, 1998. 1

[8] V. Kolmogorov and R. Zabih. What energy functions can be minimizedvia graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004. 2

[9] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML 2001*, pages 282–289. Morgan Kaufmann Publishers Inc., 2001. 2

[10] J. Sun, W. Zhang, X. Tang, and H.-Y. Shum. Background cut. In *ECCV 2006*, pages 628–641. Springer, 2006. 1

[11] M. F. Tappen and W. T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *ICCV 2003*, page 900. IEEE Computer Society, 2003. 6

[12] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. Technical Report TR-2001-22, Mitsubishi Electric Research Laboratories, 2002. 2