Segmenting Highly Textured Nonstationary Background

David Russell and Shaogang Gong
Department of Computer Science
Queen Mary, University of London
London E1 4NS, UK
{dave,sgg}@dcs.qmul.ac.uk
http://www.dcs.qmul.ac.uk/researchgp/vision

Abstract

Detection of unusual objects amongst a highly textured background is a difficult problem, especially when the texture is manifest in the temporal dimension as well. Outdoor scenes involving waving trees or moving water are examples of such a scenario, but are nevertheless frequently encountered in real world vision applications. By defining a simple but rotationally sensitive Local Binary Pattern (LBP) operator and applying it in a probabilistic sense we present a compact but useful feature for tackling moving textures. But as we demonstrate, this alone is not sufficient for good segmentation in difficult circumstances. Cooccurrence of different features in a pixel's local neighbourhood provides a powerful mechanism for boosting the reliability of the foreground/background decision task. By using the conditional probabilities yielded by pairwise cooccurrence of 4-connected pixels, and casting the problem as one of Combinatorial Optimization, our results show that useful segmentation *is* possible from challenging dynamic backgrounds.

1 Introduction

Effective background modelling is a crucial first stage in most computer vision applications, especially in outdoor environments. The reliability with which potential foreground objects can be identified directly impacts on the efficiency and performance level achievable by subsequent processing stages such as tracking, recognition and threat evaluation. The nature of such a background is intrinsically statistical. Whilst the concept of statistical scene modelling suggests that there is no exact distinction between what constitutes foreground and background, a useful practical definition for surveillance in a busy urban scene is that people and the objects they cause to move are foreground. Buildings, fixtures, trees and permanent objects, together with any environmental change in lighting such as shadow caused by moving clouds, form the background. Critically, we consider that background is in general necessarily amongst foreground, i.e. it can be literally *behind* and *in front* of foreground objects, especially in urban outdoor scenes, as later examples show. The task of the background model in such a setting is to discriminate between the two classes under a potentially wide variety of lighting conditions. Evidently,

BMVC 2007 doi:10.5244/C.21.56

confusion might still arise, since trees sway in the wind, tending to become foreground, whilst people park their cars, which are eventually subsumed by the background. The most commonly encountered models are based on per pixel techniques such as adaptive Gaussian Mixture Models [13, 14], or subspace analysis based methods [10, 8], and both approaches have been used with success in many applications. However, all these techniques require and assume that the background in a scene settles quickly into a stationary state, so that the distribution of background pixels becomes stable and tight, although not necessarily continuous. This is often not the case, especially in outdoor scenes.

The focus of this paper is to tackle the more challenging problem of modelling highly textured nonstationary backgrounds, and in particular, segmenting people moving amongst dense nonstationary trees and foliage excited by the wind. Traditionally this has been a difficult problem to solve effectively due to the highly chaotic nature of such image areas containing branches and leaves. The high information content, or *entropy*, of patterns encountered and their temporal behaviour make them inherently incompressible and thus hard to model compactly. With subspace techniques as employed in [10, 8], the eigenvectors of image covariance represent linkage of pixel variations across the entire scene, and are thus inefficient at capturing such independent local stochastic processes. Connectivity in the temporal dimension as exploited by Linear Prediction [14] is also likely to be ineffective due to the lack of cyclic components of intensity at a pixel.

On the other hand, Gaussian Mixture Models (GMM) [13] have been shown highly effective when it comes to acquiring and adapting to the statistical characteristics of behaviour at a pixel. However, high variance (or covariance for a colour image) inevitably implies low selectivity for a Gaussian component, so unless the spread of common pixel values is confined to several narrow modes, there is the danger that a foreground object will fail to be detected reliably. In addition, the GMM is at the opposite extreme from the subspace model when it comes to connectivity: in general it offers no mechanism for support regarding foreground/background decisions between pixels, either local or global.

A further additional requirement for a background model intended for outdoor use is that it must not be severely affected by changes in scene illumination with regard to both intensity and chromaticity, although in some implementations [6] constancy of the latter is used to mitigate the effect of false positives caused by shadows.

From these observations it becomes apparent that a candidate solution should satisfy the following: (1) have a probabilistic basis, (2) encode local pixel patterns, (3) exhibit resilience to lighting variations, (4) provide local support among pixels, and (5) be efficient in implementation. To this end, we propose a solution which embraces three important aspects. Firstly, a rotationally variant simplification of the LBP_8 operator as the image feature reduces susceptibility to illumination changes and provides an initial level of pattern sensitivity. Secondly, a cooccurrence map representing mutual conditional probabilities between adjacent pairs of pixels lends local support to the foreground/background segmentation decisions, encoding a further degree of pattern dependence. Finally, the array of image pixels is treated similarly to a Markov Random Field (MRF), and an optimal realization of the segmentation in terms of pixel labelling in a combinatorial sense is arrived at by a minimum cut on a related graph. Experiments on a challenging dataset involving objects heavily obscured by tree branches demonstrate the advantage of this approach.

2 High Entropy Scenes

Many typical scenes contain areas of high inherent complexity such as specular reflection from disturbed water and chaotic occlusion and appearance variation of vegetation moving under the influence of air flow. From the standpoint of information theory these represent *high entropy* sources [7], whilst signal processing tends to consider the effect spectrally, and refers to sources emitting *wideband noise*. In single frames, the chaos is a spatial property, whilst in video such stochastic variation occurs temporally as well. Exact modelling of the precise characteristics of intensity over time in a high entropy image area is by definition almost impossible: the information is highly incompressible.

From a foreground/background detection point of view we wish to highlight unusual state or behaviour of the objects in view, which might for example entail a person walking in front of a tree in leaf, or perhaps passing *behind* it, causing partial occlusion due to the *background* now being in front of the person of interest. In both cases, the requirement is to identify some less common pixel intensity configurations amongst a potentially broad range. Occlusion of the foreground, as in the second case, merely compounds the detection problem by fragmenting the available useful evidence.

2.1 Rotationally Specific LBP_4

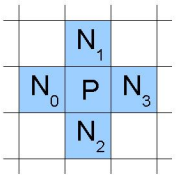
High entropy image content is commonly modelled as texture [15]. This general approach does not encode *exact* pixel configurations, rather *typical* patterns exemplary of the region. The LBP_8 operator described in [9] cleverly encodes a summary of patterns in a 3×3 pixel block into one of ten different codewords in a way which renders it insensitive to both absolute illumination and pattern rotation. These are both crucial attributes in texture analysis. Using such a scheme, segmentation on the basis of texture may be achieved by identifying regions with a similar probability distribution over the ten possible codewords.

But the requirement for foreground/background segmentation is different. We are not interested in regional statistics, but pattern statistics at a pixel, and furthermore, rotational invariance is not only unnecessary, but a hindrance with regard to our modelling requirement. Thus we introduce the concept of a Rotationally Specific Local Binary Pattern (RSLBP) operator for grayscale images, obtained by simplifying LBP_8. As shown in Figure 1 the value of the RSLBP_4 operator at a pixel is given by subtracting the intensity value of the centre pixel from each of its 4-connected neighbours. The sign of the result of each subtraction contributes a single bit to form a 4 bit codeword. For us, the spatial mapping from neighbour to bit position is immaterial as long as it is applied consistently. This rotationally specific texture feature is quick and simple to compute, and yields a compact characterization of two-dimensional image gradient at a pixel fit for our purpose.

Application of the RSLBP_4 operator to an image produces a symbol $S_r = \{0 \ldots 15\}$ at pixel location r. By considering the 16 bin histogram of these symbols at each pixel over a training set of K frames, we obtain an estimate of Probability Density Function (PDF) representing pixel configuration over this feature:

$$p(r = S_r | x, y) = \frac{1}{K} \sum_{k=1}^{K} u \quad \text{where} \quad u = \begin{cases} 1 & \text{if } R(I^T_{x,y,k}) = S_r \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

A query image I^Q may be tested against this simply by evaluating the RSLBP_4 operator at every pixel and obtaining the appropriate probability from the histogram, which in turn is tested against a threshold to yield a rudimentary foreground/background segmentation.



$$RSLBP_4 = \sum_{i=0}^{3} (N_i > P).2^i$$

Figure 1: Kernel for the new RSLBP$_4$ operator: a 4 bit word is composed from the boolean results of thresholding the 4-connected neighbours against the centre pixel.

2.2 Cooccurrence Matrix

In order to provide local support between pixels, we also use the training data to build a cooccurrence matrix between every adjacent pair of 4-connected pixels both horizontally and vertically in the image. This two-dimensional histogram represents the joint probability of two separate RSLBP$_4$ symbols occurring simultaneously at the two adjacent locations. Although conceptually, cooccurrence between pixels horizontally and vertically is the same, from an implementation point of view it is preferable to consider it in two separate arrays, C_h of size $(M-1)\times N\times 16\times 16$ elements, and C_v of size $M\times(N-1)\times 16\times 16$ elements.

$$C_h(x,y,i,j) = \frac{1}{K}\sum_{k=1}^{K} u \quad \text{where} \quad u = \begin{cases} 1 & \text{if } R(I^T_{x,y,k}) = i \;\&\; R(I^T_{x+1,y,k}) = j \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$C_v(x,y,i,j) = \frac{1}{K}\sum_{k=1}^{K} u \quad \text{where} \quad u = \begin{cases} 1 & \text{if } R(I^T_{x,y,k}) = i \;\&\; R(I^T_{x,y+1,k}) = j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

at location (x,y) where $i,j = \{1,2\ldots 16\}$, $R(\cdot)$ is the RSLBP$_4$ operator, and I^T_k $k = \{1,2\ldots K\}$ is the training set. The cooccurrence matrices at each pixel are normalized to the number of training samples K such that they correctly reflect the joint PDF.

Now consider two horizontally adjacent pixels r and s in a query image I^Q having RSLBP$_4$ symbols S_r and S_s respectively. If on the basis of our information solely about pixel r from the training data we decide that it is background, then we can obtain from the cooccurrence relationship a conditional probability of symbols over pixel s from $C_h(x_r,y_r,S_r,S_s)$. But in order for this to be a valid probability, we have to normalize C_h over its last dimension such that the conditional probability of s given r is:

$$p(s = S_s | r = S_r, x_r, y_r) = \frac{C_h(x_r,y_r,S_r,S_s)}{\sum_j C_h(x_r,y_r,S_r,j)} \tag{4}$$

However, the relationship between r and s is symmetrical, so if s were known to be background then the conditional probability over r comes from a similar expression. We note however, that the normalization constant in the denominator must be obtained by summing along the *third* dimension of C_h this time:

$$p(r = S_r | s = S_s, x_r, y_r) = \frac{C_h(x_r,y_r,S_r,S_s)}{\sum_i C_h(x_r,y_r,i,S_s)} \tag{5}$$

It becomes apparent that these mutually dependent results cannot be acted on sequentially, especially when it is remembered that a pixel is potentially supported by four neighbours. We consider that for any given query image there will be a combination of foreground/background decisions amongst the pixels, i.e. a segmentation by pixel labelling, such that the labelling process is made optimal according to our localized support measure introduced above. To find the optimal labelling of all pixels in a scene, we assert that the problem is an exercise in Combinatorial Optimization and look to a graph cut technique in order to solve it.

3 Combinatorial Optimization

The problem of choosing a label for each pixel in an image from a finite set of labels according to a set of penalty expressions is the essence of discrete optimization. The objective is to separate the pixels according to their labels in the configuration which incurs the least penalty. If the penalty criteria are correctly designed, the optimal separation is useful in some way.

The labelling of pixels from a discrete set is directly equivalent to making a cut on a graph consisting of vertices and edges as shown in Figure 2. In such a graph there is a vertex for each pixel, and a special *terminal* vertex representing each element of the label set. Every pixel node is coupled by an edge to every terminal, but edges also exist between the pixels to represent their interdependencies. According to a scheme of penalties, every edge is assigned a weight determined by the cost of cutting that edge. The optimal solution is obtained by cutting enough edges to leave every pixel connected to exactly one terminal, thereby taking on that terminal's label, and yielding the combination of pixel to terminal assignments which gives the minimum cost cut of the graph and hence the overall problem solution.

It was shown in [4] that for the special case of two labels, an optimal solution can be obtained in polynomial time using the Minimum Cut/ Maximum Flow (MinCut/MaxFlow) algorithm. Fortunately our foreground/background segmentation is just such a binary problem. Segmentation into more regions than this is potentially interesting, but the multi-way cut has been shown to be NP-complete [3], although [2] describes a way of achieving a *local* energy minimum within a constant factor of the *global* minimum by their alpha expansion algorithm.

The graph cut problem has much in common with the solution of Bayesian networks and Markov Random Fields (MRF) [5], whereby a realization of the field encompasses the interdependencies of the nodes. A method described in [12] demonstrates how local support can be achieved by considering the grid of pixels as an MRF utilising the Potts interaction model [11], in which the penalty for separating pixels is a constant. This leads to their goal of overall smoothness in the segmentation, which whilst might look appealing may eventually not be accurate.

In solution of the binary label case by the MinCut/MaxFlow algorithm, one can imagine trying to transport as much water from the *source* node to the *sink* node by a system of pipes having capacity limits equal to the edge weights. When no more capacity can be added to the network, the path traced by the saturated pipes (edges) defines the minimum cut. In our case, the capacity of a pipe depends on which way the water is flowing, i.e. which of its end nodes is joined to the source and which to the sink. This is crucial in

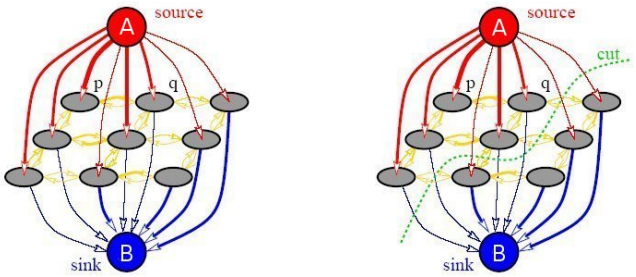
determining which conditional probability, and hence penalty, is applied at the final segmentation. In our algorithm, illustrated for clarity here by only three pixels in Figure 3(a), the cost of a given labelling $\mathcal{L}$ is the energy function

$$E(\mathcal{L}) = \sum_{r \in I^{\mathcal{Q}}} D_r(l) + \sum_{\{r,s\} \in \mathcal{N}} V_{rs}(p(r|s), p(s|r)) \tag{6}$$

consisting of penalty terms D derived from a pixel's probability in isolation of being background and an interaction term V based on conditional probability from cooccurrence. Here $\mathcal{N}$ represents the 4-connected neighbourhood of connections as shown in Figure 2 (not to be confused with the 4-connectivity earlier in RSLBP_4, even though it involves the same pixels). The edge weights illustrated in Figure 3(a) are assigned as follows:

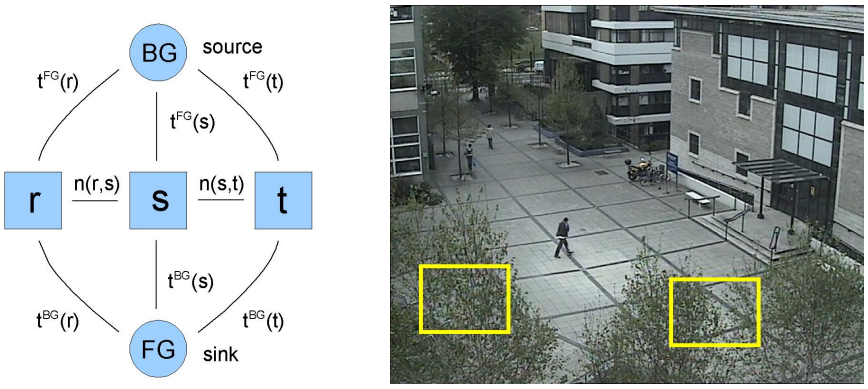
Edge	Forward Capacity	Reverse Capacity
$t^{BG}(r)$	1	1
$t^{FG}(r)$	$\frac{\beta}{(p(r=S_r)+0.01)}$	$\frac{\beta}{(p(r=S_r)+0.01)}$
$n(r,s)$	$\lambda\, p(s=S_s \mid r=S_r, x, y)$	$\lambda\, p(r=S_r \mid s=S_s, x, y)$

The V terms can be seen as a penalty for separating pixels which, according to cooccurrence, should belong together and to the background. To cause them to end up separated, one would have to have a very low individual probability of occurring. The constants β and λ control the magnitude of the effect of the D and V penalties relative to each other, and also to the unity penalty assigned to the cost of being background.



(a) A graph $\mathcal{G}$ (b) A cut on $\mathcal{G}$

Figure 2: Graph for an array of only 9 pixels: *source* and *sink* nodes represent the two classes A and B. A cut must separate A and B: the MinCut/MaxFlow algorithm finds the cheapest. A practical graph contains a node for *every* image pixel. Figure taken from [1].



(a) (b)

Figure 3: (a) More detailed graph for an array of only 3 pixels, showing Background as the *source* label and Foreground as the *sink*. Terminal and neighbourhood link edge weights are shown as t and n respectively. Cutting a lower t-link joins a pixel to the BG label incurring cost t^{BG} (b) Scenes chosen for the experiment lie within highlighted windows.

4 Experiment

To demonstrate the effectiveness of our algorithm using RSLBP_4 and MinCut-MaxFlow, the challenging scene shown in Figure 3(b), containing a leafy tree in a courtyard, was chosen. The leaves move significantly in the wind whilst people pass *behind* the tree, but remain visible through the foliage. From a dataset of 2500 monochrome frames of size 128×96 pixels, 2000 are used as the training data to build the probability distributions and the cooccurrence matrices C_h and C_v. From the remaining frames we select an interesting subset in which a person enters the scene from the top right and walks towards the camera. We compare our RSLBP_4 operator with the standard LBP_8 operator, and also with a more primitive feature: a 16 level grayscale derived by merely truncating the pixel intensity to 4 bits. The MinCut algorithm and the previously tabulated weighting scheme was applied in all cases, and results are shown in Figure 4. We further demonstrate contribution of the MinCut stage with a comparative result in which it is *not* used: Figure 5 shows what happens when individual pixel probabilities alone are used for segmentation using the RSLBP_4 operator. Even when the foreground detection threshold is optimized manually to 0.045, there is only a hint of the presence of a person, and most of the foreground pixels are noise. Figure 6 provides further evidence in support of the RSLBP_4 and MinCut combination, with images from the right hand window in the scene of Figure 3(b).

Although the Combinatorial Optimization algorithm chooses discrete labels as its solution, the notion of a detection threshold still exists in the form of the relative scaling of the various edge weights. In our implementation, β controls the effect of the pixels' individual probabilities, whilst λ regulates the influence of the inter-pixel support. In each case, since the probabilities vary between 0 and 1, the two constants act as maximum values for their own particular type of edge. Choosing $\beta = 7$ and $\lambda = 10$ scales the optimization favourably when the t^{BG} edges are set to unity.

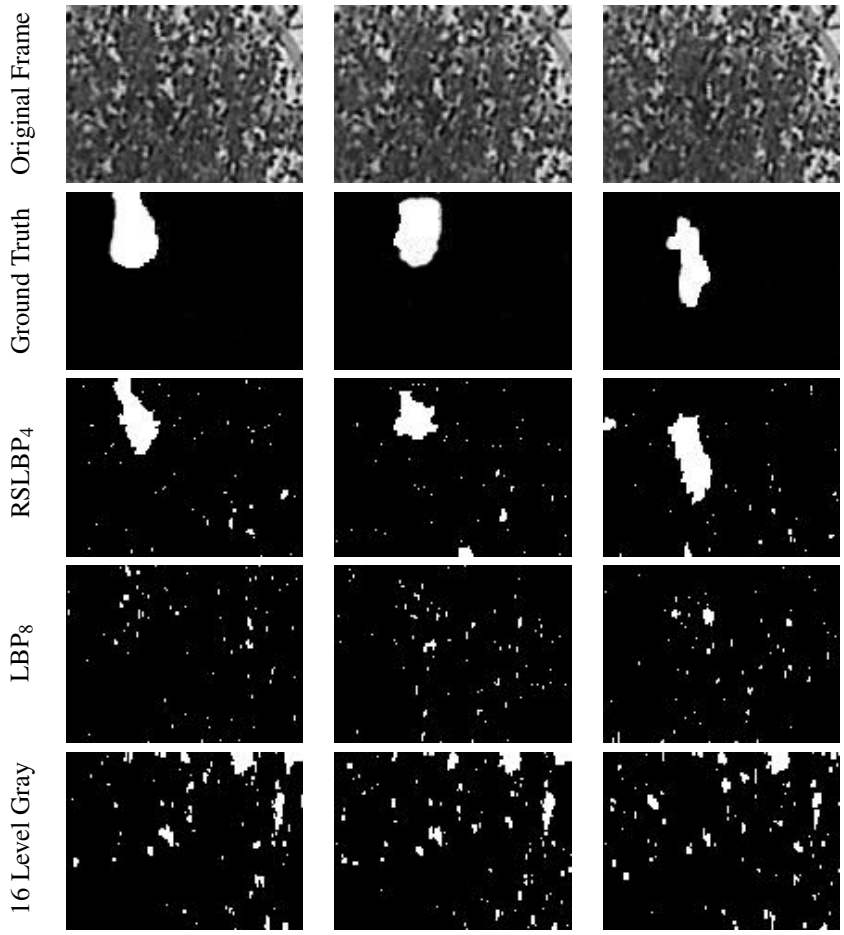


Figure 4: Three frames from the left hand window of Figure 3(b) in which a person walks behind a tree. From top to bottom: Original, Ground Truth, using $RSLBP_4$ operator, using LBP_8 operator, and using 16 level grayscale, all *with* MinCut. Note that our $RSLBP_4$ operator is the only one to produce a useful segmentation here.

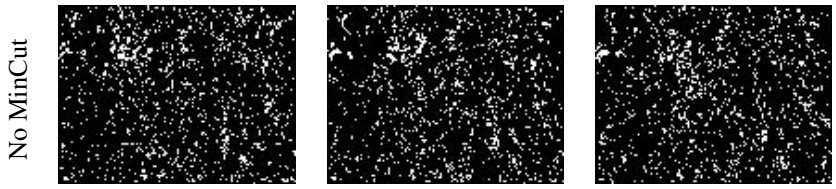


Figure 5: The same three frames using $RSLBP_4$ but *without* MinCut, and hence no local support. The person is barely discernible amongst the noise. LBP_8 and Grayscale are similarly ineffectual without the vital MinCut stage.

Our $RSLBP_4$ operator can generate 16 different values as currently defined, leading to a cooccurrence matrix with only 16×16 entries. This compactness is convenient for two practical reasons. Firstly the memory required to store the inter pixel data is manageable, and secondly the quantity of training data to adequately estimate it remains modest. LBP_8 generates only 10 possible values, but as the experiments show, its rotational invariance renders it useless for our purpose. A rotationally *variant* version of LBP_8 generates 59 combinations and is thus, according to the previous arguments, not so attractive.

Overall the favourable segmentation afforded by $RSLBP_4$ in our results in Figures 4 and 6 strongly supports the idea that it is a better choice than the other two commonly encountered features for the current application. Furthermore, the comparison between Figures 4 and 5 show clearly that the graph cut technique contributes enormously to the quality of the segmentation. We believe that the 'double level' of local spatial support afforded by the partnership of the two techniques is the reason for the distinctive result.

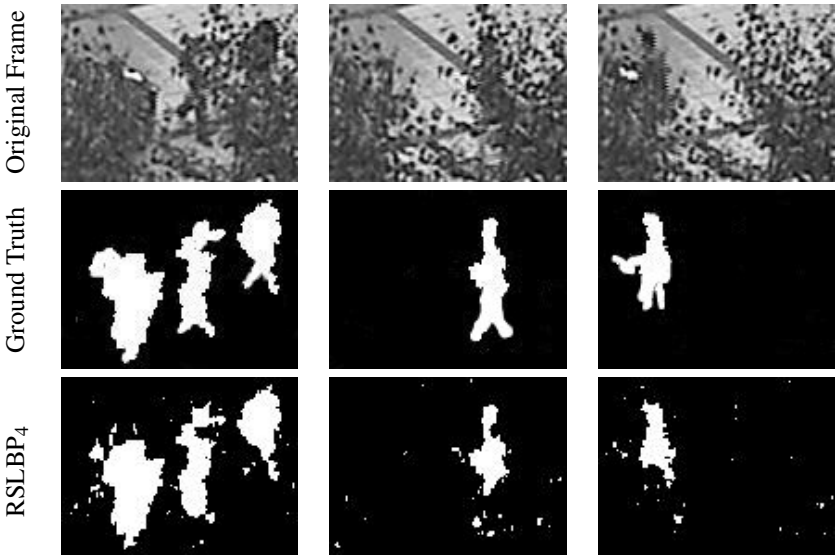


Figure 6: Further results using $RSLBP_4$ and MinCut from the right hand window in Figure 3(b) in which people pass behind trees. The algorithm succeeds in identifying unusual objects in spite of considerable local clutter from the leaves.

5 Conclusion

We have introduced a simple new operator $RSLBP_4$ based on existing LBP methods, and have shown how it can be applied to advantage in the foreground/background segmentation of highly textured dynamic scenes. We claim that its sensitivity to rotation, but resilience to overall illumination variations, both contribute vitally to its success in this application. The restricted range of output symbols of $RSLBP_4$ permits tractable acquisition of adjacent pixel cooccurrence data. We have shown that such data may be used to

construct a graph, of which the minimum cost cut facilitates mutually supporting inferences between pixels, leading to a useful segmentation which would not have been easy to arrive at otherwise. Although a model involving separate collection of training data is described here, it is anticipated that an adaptive online derivative would also be possible, and that this would provide a useful direction for further research.

6 Acknowledgment

The authors would like to thank Vladimir Kolmogorov for use of his C++ implementation of the MinCut/MaxFlow algorithm which is available at:
http://www.adastral.ucl.ac.uk/ vladkolm/software.html.

References

[1] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. *IEEE PAMI*, 26(9):1124–1137, 2004.

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

[3] Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, P. D. Seymour, and Mihalis Yannakakis. The complexity of multiterminal cuts. *SIAM J. Comput.*, 23(4):864–894, 1994.

[4] L. Ford and D. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[5] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE PAMI*, 6(6):721–741, November 1984.

[6] T. Horprasert, D. Harwood, and L.S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *ICCV Frame-Rate WS*, Kerkyra, Greece, 1999.

[7] H. Hung and S. Gong. Quantifying temporal saliency. In *British Machine Vision Conference*, pages 742–749, September 2004.

[8] Y. Li. On incremental and robust subspace learning. *PR*, 37(7):1509–1518, 2004.

[9] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *PR*, 29(1):51–59, January 1996.

[10] N. Oliver, B. Rosario, and A. Pentland. A Bayesian computer vision system for modelling human interactions. *IEEE PAMI*, 22(8):831–843, August 2000.

[11] Renfrey B. Potts. Some generalized order-disorder transformation. In *Transformations, Proceedings of the Cambridge Philosophical Society*, volume 48, pages 106–109, 1952.

[12] Konrad Schindler and Hanzi Wang. Smooth foreground-background segmentation for video processing. In *ACCV (2)*, pages 581–590, 2006.

[13] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE CVPR*, pages 246–252, Colorado, 1999.

[14] K. Toyama, J. Krumm, B. Brummit, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *IEEE ICCV*, volume 1, pages 255–261, Kerkyra, Greece, 1999.

[15] C. S. Zhu, N. Y. Wu, and D. Mumford. Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9(8), November 1997.