Integrating a discrete motion model into GMM based background subtraction

Christian Wolf Jean-Michel Jolion
Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205, F-69621, France
{*christian.wolf,jean-michel.jolion*}*@liris.cnrs.fr*

Abstract

GMM based algorithms have become the de facto standard for background subtraction in video sequences, mainly because of their ability to track multiple background distributions, which allows them to handle complex scenes including moving trees, flags moving in the wind etc. However, it is not always easy to determine which distributions of the mixture belong to the background and which distributions belong to the foreground, which disturbs the results of the labeling process for each pixel. In this work we tackle this problem by taking the labeling decision together for all pixels of several consecutive frames minimizing a global energy function taking into account spatial and temporal relationships. A discrete approximative optical-flow like motion model is integrated into the energy function and solved with Ishikawa's convex graph cuts algorithm.

1. Introduction

Background subtraction, the task of separating foreground (object) pixels from background pixels in a video, is an important step in many applications, either because one is interested in an object's silhouette itself, or as a preprocessing step, for instance for tracking algorithms. Most existing methods build an explicit background model either using a unimodal distribution through median [2] or Kalman filtering [12] or similar techniques, or a multi-modal distribution like GMM's [9, 11, 13]. A survey can be found in [8].

In some cases one is interested in a very precise segmentation result, e.g. when an object's shape shall be used to recognize object classes or actions. In this regard, the strengths of the existing methods are also their weaknesses: the FG/BG segmentation decisions are taken on a per pixel level, which is highly suboptimal.

In this paper we present a method [1] which improves existing GMM based algorithms by taking the segmentation decision on "global" level, i.e. simimultanuously for all pixels of a whole block of the spatio-temporal cube. Spatial and temporal interactions are taken into account by a global energy function which is minimized with graph cuts, searching for the exact globally best solution. Temporal interactions handled with a motion model which is calculated through an approximate optical flow algorithm, also solved with graph cuts.

The contribution of this paper is twofold: the experiments show a significant improvement over existing methods. Furthermore, the algorithm for approximate optical flow might be useful in other applications.

Our paper is organized as follows: section 2 is a short reminder on GMM based BG subtraction. Section 3 describes our method and section 4 presents experimental results. Section 5 finally concludes.

2. GMM based Background subtraction

Without loss of generality we present the case for grayscale images in this paper, the adaptation to color images and other multivariate cases is straightforward.

The perhaps most widely known and used background subtraction algorithm is the Stauffer-Grimson algorithm [11] which is also be the base of our method. It keeps K Gaussians for each pixel presenting a multi modal distribution of pixel grayvalues. At each new frame the new grayvalue y is checked against all Gaussians and the best matching Gaussian is selected, if y is within a threshold of standard deviations of the mean, a new Gaussian is created else. The parameters of the

[1] This project was financed through the French National grant "ANR-CaNaDA" — Comportements Anormaux: Analyse, Détection, Alerte, No. 128, which is part of the call for projects CSOSG 2006 Concepts Systèmes et Outils pour la Sécurité Globale.

matched Gaussian (weight, mean, standard deviation) are updated using a learning rate parameter.

The difficulty lies in the decision whether a matched Gaussian corresponds to the background (BG) or the foreground (FG) distribution. In [11], the Gaussians are ordered by $\frac{w}{\sigma}$, where w is the weight and σ is the standard deviation of a Gaussian, and it is assumed that 70% of the time a pixel matches against background.

In our method, we modified the final decision whether a pixel is FG or BG. The set of background Gaussians is determined by directly thresholding $\frac{w}{\sigma}$. The decision on the pixels FG/BG label is set accordingly, additionally integrating a global model described in the next section.

3. Spatio-temporal regularization

In our method the decisions on the pixels' FG/BG labels are taken jointly using a global energy function. They are driven by a measure Δ_i for each pixel i corresponding to the deviation from the best matching background distribution:

$$\Delta_i = \frac{|y_i - \mu_i|}{\sigma_i} \tag{1}$$

where y_i is the pixel's grayvalue and μ_i and σ_i are, respectively, the mean and the standard deviation of the best matching BG distribution.

The global energy function includes a data attached term involving Δ_i, a Potts model [7] regularizing spatial interactions as well as a Potts model for temporal interactions. Equation (2) first gives the model assuming zero motion in the spatio-temporal cube. For simplicity, we index the pixels in the spatio-temporal cube with a single index instead of three indices:

$$\begin{aligned} E(x,y) &= \alpha_d \sum_i E_d(\Delta_i, x_i) \\ &+ \alpha_s \sum_{i \sim j} \delta(x_i, x_j) \\ &+ \alpha_t \sum_{i \wr j} \delta(x_i, x_j) \end{aligned} \tag{2}$$

where x_i are the binary labels denoting the foreground/background decisions for pixels i, $i \sim j$ indicates that i and j are spatial neighbors and $i \wr j$ indicates that i and j are temporal neighbors. The different α are weights and δ denotes the Kronecker delta given as $\delta(a,b) = 1$ if a=b and 0 otherwise. The data attached term E_d is given as:

$$E_d(\Delta_i, x_i) = \begin{cases} \Delta_i & \text{if } x_i = 0 \\ 2D - \Delta_i & \text{if } x_i = 1 \end{cases} \tag{3}$$

This choice of E_d results in thresholding Δ_i using a fixed threshold D if the regularizing Potts model is removed, i.e. the Potts model serves as an improving regularizer of existing methods based on thresholding.

With motion present in the cube, the temporal regularizer is sub optimal. We therefore include a dense motion vector field $\boldsymbol{u}_i$ with horizontal and vertical components $(u_{i,x}, u_{i,y})$ into the global model:

$$\begin{aligned} E(x,y,\boldsymbol{u}) &= \alpha_d \sum_i E_d(\Delta_i, x_i) \\ &+ \alpha_s \sum_{i \sim j} \delta(x_i, x_j) \\ &+ \alpha_t \sum_i \delta(x_i, x_{i \rightarrow \boldsymbol{u}_i}) \\ &+ \alpha_m \sum_{i \sim j} E_m(\boldsymbol{u}_i, \boldsymbol{u}_j) \\ &+ \alpha_m \sum_{i \wr j} E_m(\boldsymbol{u}_i, \boldsymbol{u}_j) \end{aligned} \tag{4}$$

where the notation $i \rightarrow \boldsymbol{u}_i$ indicates the index of the site we get when the motion vector $\boldsymbol{u}_i$ is applied to site i, in the frame *following* the frame of i. The expression $E_m(\boldsymbol{u}_i, \boldsymbol{u}_j)$ is an energy functional punishing misaligned motion vectors. The two components in each motion vector $\boldsymbol{u}_i$ take values in $[-T, T]$, T being small (3-5 pixels), which is feasible since motion is usually not significant between subsequent fames.

The energy function given in (4) is in general difficult to minimize due to the complex interactions between x and u. Instead of solving it approximatively, for instance using energy truncation and the α-expansion algorithm [6], or QPBO [5], which often leads to poor results, we prefer to perform some approximations of the function in order to able to solve it exactly. In particular, we remove the motion vectors $\boldsymbol{u}_i$ from the optimization and calculate them from the input images using approximate optical flow. Consequently, in the optimization over x the terms in (4) involving E_m are constant and can be omitted, and the other terms are either unary in x or submodular:

$$\begin{aligned} \hat{x} = \arg\min_x \quad & \alpha_d \sum_i E_d(\Delta_i, x_i) \\ &+ \alpha_s \sum_{i \sim j} \delta(x_i, x_j) + \\ &+ \alpha_t \sum_i \delta(x_i, x_{i \rightarrow \boldsymbol{u}_i}) \end{aligned} \tag{5}$$

The minimization can thus be carried out efficiently with graph cuts using Kolmogorov et al.'s graph construction method for binary labels [6]. Since the motion vectors $\boldsymbol{u}_i$ in the temporal term of the Potts model are constant, they only determine the placement of the n-edges in the st-graph.

Calculating the motion vectors $\boldsymbol{u}_i$ from the cube of input images y_i corresponds to a dense optical flow problem, so existing methods can be applied (e.g. Sand and Teller [10]), but they are painfully slow. We decided to calculate approximate optical flow using a similar global energy function as for the calculation of the x_i, roughly based on the last 3 expressions of eq. (4).

Another approximation is motivated by the goal of efficient minimization with graph cuts. We completely separate the horizontal and vertical components of each motion vector, which means that there are no edges in the graph between the nodes of $u_{i,x}$ and $u_{i,y}$, so the optimization of both parts will be carried out independently. This is possible by choosing L_1 (the Manhattan distance) for the motion vector distance functional E_m and by replacing the respective unknown motion vector component by a minimum over the possible values in the data attached term:

$$\begin{aligned}\hat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u}} \quad & \alpha_t \sum_i \min_{a\in[1,T]} |y_i - y_{i\to\left[\begin{smallmatrix}u_{i,x}\\ a\end{smallmatrix}\right]}| \\ + \quad & \alpha_t \sum_i \min_{a\in[1,T]} |y_i - y_{i\to\left[\begin{smallmatrix}a\\ u_{i,y}\end{smallmatrix}\right]}| \\ + \quad & \alpha_0 \sum_i |u_{i,x}| + \alpha_0 \sum_i |u_{i,y}| \\ + \quad & \alpha_m \sum_{i\sim j} |u_{i,x}-u_{j,x}| + |u_{i,y}-u_{j,y}|\end{aligned} \tag{6}$$

The first two terms are the data attached terms, they favor constant grayvalue in the motion direction. The terms corresponding to the (small) weight α_0 slightly favor zero motion, they are necessary since the minimum expressions in the data attached terms tend to decrease the differences in energy between different labels. The second order terms favor homogeneous motion in a spatial and temporal neighborhood. They involve multiple labels per variable and are not necessarily submodular, so they cannot be solved using Kolmogorov's graph construction method. However, the labels are linearly ordered and the energy potentials are convex in label differences, the function can therefore be minimized using Ishikawa's graph construction method [3] which results in $2T+1$ binary labels for each pixel, one for each possible motion label.

4. Experimental results

We evaluated the proposed method on our dataset containing difficult scenes with several moving people. The approximate optical flow works quite well given the approximations made to achieve the desired speedup. Figure 1 shows a part of a pair of consecutive frames together with a motion magnitude image as well as a zoom into the head part showing the motion vectors.

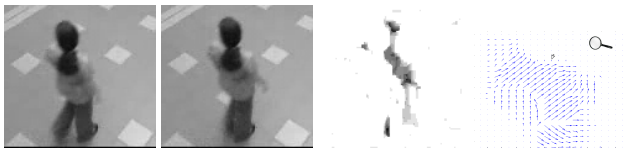

Figure 1. Approx. dense opt. flow: From left to right: 1^{st} frame, 2^{nd} frame, motion magnitude, zoom into the vector field.

Figure 2 shows some results of the FG/BG segmentation on our dataset. As can be seen, the new method produces significantly cleaner and preciser images. More importantly, the result in the second row shows that the method is able to correct bursts of illumination changes, which disturb the original GMM only method. Note that postprocessing, e.g. with mathemathical morphology, will not be able to clean up the noise. A video with more details in animated gif format is available online[2].

Although the convex prior in the optical flow algorithm theoretically should tend to oversmooth motion vector field, this is not confirmed experimentally and does not seem to hinder the performance of the method.

The parameters and weights used in the experiments were the following: D=3, T=4 (9 motion labels), α_d=1, α_0=0.2, α_s=2, α_t=4, α_m=3. The block size is 2 frames, better results can be achieved with longer blocks with the cost of higher computational complexity.

We use the graph cut implementation by Boykov and Kolmogorov [1] available on V. Kolmogorov's website and a Matlab wrapper implemented by Miki Rubinstein. The rest of the method has been implemented in Matlab and has not been optimized. Runtime complexity could be improved tremendously by recoding the method in C/C++ and by using the dynamic graph cuts method by Kohli and Torr, which uses the solution of a previous frame to accelerate the optimization step of the current frame [4]. Currently the GMM method without regularization runs in 0.8s per 320×240 frame, whereas the proposed version runs in 3s per frame. Most of the time is spent in the Matlab code, the run time spent on the graph cuts optimization (C++) is negligible.

5. Conclusion

We presented a new BG subtraction method which integrates information from approximated optical flow for a spatial and temporal regularizer. The algorithm calculates the exact solution of a global energy function

[2] http://liris.cnrs.fr/christian.wolf/vids/bgsuboflow.gif

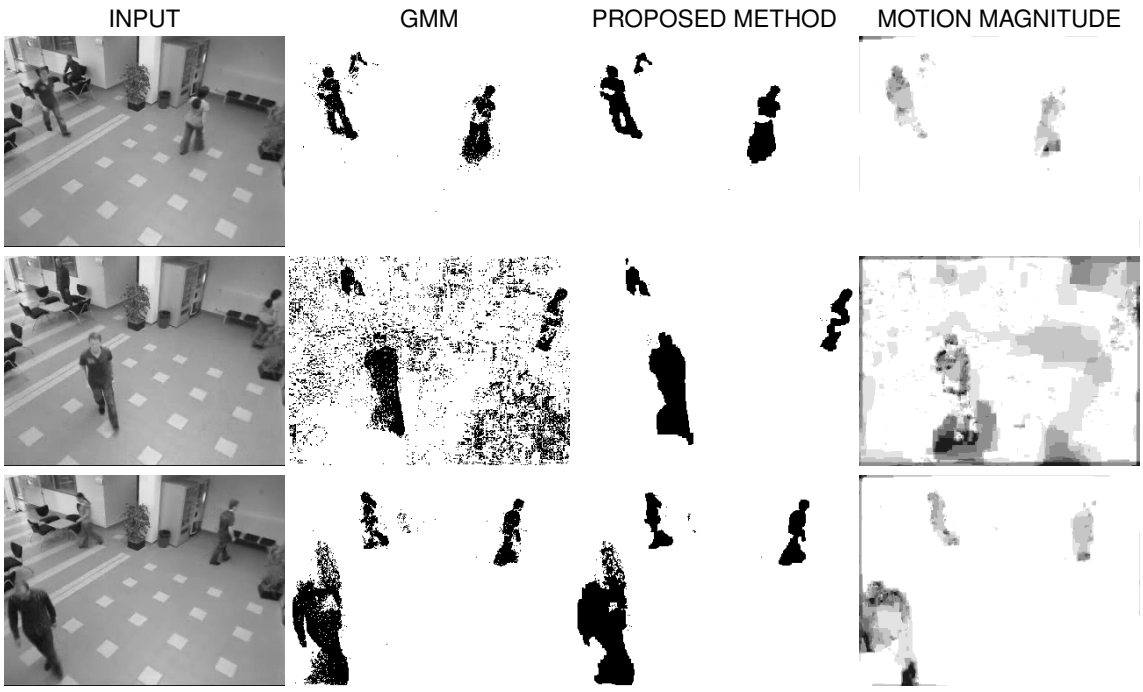

Figure 2. Results on our dataset containing several moving people.

taking segmentation decisions for a short block of the spatio-temporal cube. The method has been tested on a dataset of videos containing several moving people and is able to significantly improve existing methods

References

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

[2] R. Cucchiara, C. Grana, M. Piccardi, , and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(10):1337–1342, 2003.

[3] H. Ishikawa. Exact optimization for markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1333–1336, 2003.

[4] P. Kohli and P. Torr. Dynamic graph cuts for efficient inference in markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2079–2088, 2007.

[5] V. Kolmogorov and C. Rother. Minimizing non-submodular functions with graph cuts - a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7), 2007.

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

[7] S. Li. *Markov Random Field Modeling in Image Analysis*. Springer Verlag, 2001.

[8] D. Parks and S. Fels. Evaluation of background subtraction algorithms with post-processing. In *Proceedings of the International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 192–199, 2008.

[9] P. Power and J. A. Schoonees. Understanding background mixture models for foreground segmentation. *Image and Vision Computing*, 24(5), 2006.

[10] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. *International Journal of Computer Vision*, 80(1):72–91, 2008.

[11] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(8):747–757, 2000.

[12] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):780785, 1997.

[13] Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773–780, 2006.