Using Trajectories derived by Dense Optical Flows as a Spatial Component in Background Subtraction

Martin Radolko
University of Rostock
and Fraunhofer IGD
Joachim-Jungius 11
Rostock 18059
martin.radolko@igd-r.fraunhofer.de

Fahimeh Farhadifard
University of Rostock
and Fraunhofer IGD
Joachim-Jungius 11
Rostock 18059
fahimeh.farhadifard@igd-r.fraunhofer.de

ABSTRACT

Foreground-Background Segregation has been intensively researched in the last decades as it is an important first step in many Computer Vision tasks. Nonetheless, there are still many open questions in this area and in this paper we focus on a special surveillance scenario where a static camera monitors a predefined region. This restrain makes some aspects easier and good results could be achieved with Background Subtraction methods. However, these only work pixelwise and lack the spatial component completely. We suggest an approach to add the crucial spatial information to the segmentations with Dense Optical Flows. For this, a number of successive images are taken from the video to compute the Trajectories of the pixels through these frames. This enables us to fuse the information from the several images and use this for segmentation. The algorithm was evaluated on a video from a surveillance camera and showed promising results.

Keywords

Foreground-Background Segregation, Background Subtraction, Dense Optical Flows, Video Segmentation

1 INTRODUCTION

Detecting objects of interest in an image or video is an arduous task which has distressed the computer vision community for decades. The abundance of different circumstances and aims (definitions of objects of interest) makes it impossible to create one general solution to this problem. Hence, different special areas have developed over time like the recognition of a few previously specified objects in an image or video [Duraisamy and Jane, 2014], general object detection for images [de Carvalho et al., 2010] or detection of moving objects in a video [El Harrouss et al., 2015].

In this paper we consider the last case and want to detect all moving objects in a video scene. Therefore, we assume a video from a static camera, e.g. a surveillance camera mounted in a shopping street. This makes it easier to detect the movement of objects because the background (buildings, streets etc.) are completely stable in their position over time. There are some attempts to extend these techniques to non-static cameras by estimating the three dimensional trajectories of the background[Sheikh et al., 2009]. However, the creation of a meaningful background model is error-prone because small imperfections in the trajectories will accumulate over time and thwart any precise modeling. That's why, the trajectories themselves were only used as cues and everything with a different trajectory than the background was labeled as foreground.

If the video is taken with a static camera, the background can be modeled over time with statistical methods. Moving objects, which are the interesting objects to most users, can then be detected via Background Subtraction. This is a very popular approach since it gives State of the Art results for one of the main use cases of segmentation algorithms, static surveillance cameras. Also, it is very effective and can be done in real time[Tabkhi et al., 2015]. However, there are two issues with this methods which have to be addressed.

The first one is the background modeling itself, which is affected by the presence of the ubiquitous noise in images. This noise can be reduced by applying a Gaussian Filter on the image but slight variations are always present (otherwise too much information would be lost due to the strong Gaussian smoothing) and have to be compensated with an adequate threshold, ideally one which adapts to the specific scene and camera [Soeleman et al., 2012]. Furthermore, the background can change over time, e.g. a car that parks or

light which is turned on or off. To model these occurrences an accurate adjustment of the learning rate of the model is required and sometimes an event detection, which changes the learning rate or resets the model for special cases, can be beneficial [Radolko et al., 2015, Toyama et al., 1999].

The second issue to be addressed is the pixelwise thresholding and labeling in the Background Subtraction phase which incorporates no spatial component. Images are usually quite smooth with only few distinctive edges. Without a spatial model, this smoothness cannot be represented by the method and no coherent objects will be segmented but many small fragments. There are a vast number of models which have been applied on Background Subtraction results to add spatial information. One example is Graph Cut [Tang and Miao, 2008], which finds the best cut in a probability map created by the Background Subtraction. Another method is the Markov Random Field which models direct neighbourhood relations between single pixels and therefore enforces a spatial coherency [Wan and Wang, 2010].

In this paper, we suggest a different approach of addressing this lack of spatial information of the Background Subtraction method. We use Dense Optical Flows to acquire spatial-temporal information about each pixel, which allows us to track pixels through several frames of the video. Therefore, the data from several others frames of the video can be used to enhance the segmentation, which results in a less noise-sensitive algorithm. Also, the Dense Optical Flow gives us a second cue for the foreground detection. High flows correspond to moving objects in the video, which should be detected by the Background Subtraction. Thereby, the optical flow can verify and improve the Background Subtraction result in two different ways.

2 APPROACH

The aim of our approach is to detect all objects of interest in a video. The camera is assumed to be static to simplify the task and consequently the areas of interest are defined as all objects in motion. If immobile objects are of interest it would be meaningless to make a video with a static camera, a single picture would be enough. First we will shortly introduce the Background Subtraction method and the algorithm to estimate the Dense Optical Flow of the video. In the next step the data from the Dense Optical Flow is used to estimate the position of an object in the previous and future frames of the video. This is used to unify succesive segmentations, which increases the reliability of the object detection.

2.1 Background Subtraction

To get a first estimation about the moving objects in the scene, we use the Gaussian Switch Model Background Subtraction described in [Radolko and Gutzeit, 2015]. There are exactly two pixelwise Background models trained over time and both use a running Gaussian estimation. However, only one is completely updated with every new frame and the other only partially for those pixels which are classified as background.

The model which gets only partially updated usually gives a better background estimation because foreground objects are omitted in the updating process and do not corrupt the model. However, since the foreground-background classification is based on the same model it works sometimes like a self-fulfilling prophecy and cannot adapt to some special cases. Therefore, the second model, which is completely updated with every frame, is necessary. A comparison between both models allows us to detect these special cases and then switch to the appropriate model. For more details see [Radolko and Gutzeit, 2015].

To further process these information simple foreground-background labels are not sufficient. Therefore, we calculate the probability of being in the foreground for each pixel. For every color channel of the image the Euclidean distance between the values of the current frame and the model is taken and weighted with the variance of the Gaussian model for that channel and pixel. To unify these to a single probability measure, the three values are scaled, bounded (to avoid probabilities larger than one) and added together. This gives us a continuous probability value, which carries much more information than a simple binary label.

2.2 Dense Optical Flow

The Dense Optical Flow is computed with the algorithm described in [Farnebäck, 2003]. The first step in creating the Optical Flow between two images consists of modeling both images as Polynomial Expansions. By solving a system of linear Equations, a displacement vector between these two models can be computed for each pixel location. However, single pixels are too vulnerable to noise, so that no smooth or accurate solution is attainable in this way. This issue can be resolved by using small areas instead of single pixels. We always use patches with the size of 15×15 pixels and radially decreasing weights for the computation of the displacement vectors in this paper.

A priori knowledge can be incorporated into the computation by setting the starting values accordingly. In this case, the algorithm only has to compute the displacement vector to the a priori knowledge, which is usually much smaller than the raw displacement between the two images. This is very beneficial because the calculation of large displacement vectors is prone to errors. This approach can be used in two ways to improve the Optical Flows. Since we use a video and want to compute the Dense Optical Flows for all frames, the

Figure 1: In the top row the intensities of the optical flow are depicted for an image. On the left side for the right-left movement and on the right for the up-down movement. In the bottom row are the orignal frame from the video and the segmentation result.

flow from the last frame can be used as a good starting approximation for the current frame. In the second strategy a hierarchical approach is used and the Optical Flow is calculated first on a low resolution version of the image. The result of this calculation is then used as a start value for versions with higher resolution.

Thereby, the Optical Flows can be computed faster and in a more stable way over the course of a video. An example of the results of the optical flow can be seen in Figure 1. In the next step these extracted information are used to enhance the segmentations derived from the Background Subtraction.

2.3 Using Optical Flows as a Spatial Component

Background Subtraction algorithms usually give good segmentation results but suffer from the lack of spatial information incorporated. Therefore, false detections of single pixels or small areas are common and prevent the precise recognition of smooth and coherent foreground objects. We try to mitigate this behavior by calculating the pathway of each pixel over several frames (past and future) and smooth the foreground probability with a Gaussian filter over this pathway.

The pathway of each pixel can be easily derived from the Dense Optical Flows. Here they are denoted as

$$DOF^x_{n-1,n}(x,y) \quad \text{and} \quad DOF^y_{n-1,n}(x,y). \tag{1}$$

The Optical Flow is here computed between the $n-1$-th and n-th frame of the video for the pixel at location (x,y) in frame $n-1$. DOF^x denotes the horizontal flow and accordingly DOF^y the vertical flow. Thereby, the pixel $p=(x,y)$ in frame $n-1$ has moved to location

$$\tilde{x} = x - DOF^x_{n-1,n}(x,y) \tag{2}$$

$$\tilde{y} = y - DOF^y_{n-1,n}(x,y). \tag{3}$$

in frame n according to the Optical Flow. The new location for p in the frames $n-2$ or $n+1$ can be calculated in a similar way. With this method pathways for single pixels can be computed over several frames. However, since the Optical Flows are not perfectly accurate and errors accumulate radically over longer pathways, reliable information can only be derived for a small number of past or future frames. In our approach, the pathways for 7 frames ($n-3$ to $n+3$) are computed and used to enhance the foreground probabilities of the n-th frame. This is done by using a Gaussian filter along the pathway centered at the n-th frame with standard deviation of 0.75. The low standard deviation ensures that the importance of the values along the pathway decreases rapidly in both directions, this corresponds to the rapidly decreasing certainty of the correctness of the pathway.

Furthermore, the Optical Flows are an indicator of movement in the scene themselves and can be used as a foreground cue. For this purpose, the location of the pixel p in the frames $n-1$ and n are taken and the Euclidean distance $d_p^{n-1,n}$ is computed. The same is done for the locations in the frames n and $n+1$. The foreground probability w^p of pixel p derived from the Background Subtraction is taken and enhanced in the following way

$$\tilde{w}^p = \frac{2}{3}w^p + \frac{1}{6}\min(d^p_{n-1,n},1) + \frac{1}{6}\min(d^p_{n,n+1},1). \tag{4}$$

This adds the Dense Optical Flow as prior to the foreground probability and ensures that the value stays in the range of $[0,1]$.

The third step is a slight spatial smoothing of each probability map separately in which the differences of the value of the central pixel with that of all other pixels in a 3×3 neighborhood are summed up, weighted and the result is added to the probability value of the central pixel. These three steps; the smoothing over the trajectories, the adding of the Optical Flow prior and the spatial smoothing; are applied successively on a batch of segmentations derived from the Background Subtraction. For the tests in this paper a batch size of 100 frames was used.

We iterated several times over the whole batch and applied all three methods each time. This elaborate process is useful because the changes in the segmentation in the first step influence the smoothing over the trajectories in the next steps. The quality and smoothness of the probability maps increases gradually over the iterations and with this also the trajectories provide better information in each step and thereby improve the segmentations further. As the trajectories cannot be fully computed for the first and last three frames of the batch, the segmentations are only computed for the frames 4 to 97 of the batch. Some trajectories cannot be computed because for each trajectory we have to reach three frames into the future and past, but for the first and last three frames of the batch this is obviously not possible.

This process, of course, needs a termination criteria which stops the iteration. Instead of a fixed number of iterations, we measured the changes in the probability fields from one iteration to the next, and if these changes where smaller than a specified threshold the loop would break. To reduce the computational cost we constrained this measurement to the probability map for one frame of the batch. A summary of the whole segmentation process can be seen in Algorithm 1.

Algorithm 1 Our Method

```
** initial segmentations **
for i=1:100 do
    frames[i] ← getimage(video)
    UpdateBackgroundModel(frames[i])
    segs[i] ← BackgroundSubtraction(frames[i])
**get  Dense Optical Flows**
for i=1:99 do
    DOF[i] = getDOF(frames[i], frames[i+1])
** get Trajectories **
Paths ← getTrajectories(DOF)
** use DOF and Paths to improve results **
while TerminationCriteria do
    for i=4:97 do
        segs[i] ← spatialsmoothing(segs[i])
        segs[i] ← AddDOFPrior(segs[i], DOF[i])
    for i=4:97 do
        segs[i] ← PathSmoothing(segs, Paths[i])
```

3 RESULTS

To test our method we used the Town Center Video [Benfold and Reid, 2011] which is made with a surveillance camera in a shopping street. The provided ground truth data is made for the comparison of tracking algorithms and not suitable for our purpose. Hence, we created several groundtruth picture ourselves manually to evaluate the algorithm. The ground truth data consists of a trimap which classifies each pixel as foreground (white), background (black) or uncertain (gray). Two examples of this can be seen in Figure 2, where also the results of our algorithm are depicted.

As comparison the algorithm from [Zivkovic, 2004] and [Zivkovic and Heijden, 2006] were used on the same data set. To have a neutral implementation of these algorithms we used that provided by OpenCV.

The results of these two and our algorithm on the test images from Figure 1 can be seen in Table 1. The F1-Score and Matthews correlation coefficient (MCC) are taken as a measure for accuracy. In both measures our approach achieved a better result than the other algorithms and could significantly improve the results achieved by just using the GSM Background Subtraction. All in all our approach shows good first results and is a promising novel method to add spatial information to pixel based segmentation methods. The computation of the optical flows is done on the GPU and therefore very fast, nonetheless is the fusion of the data from the different methods too costly for a real time approach, it took us about one to two seconds per image for the computation.

4 CONCLUSIONS

We presented a new method to enhance the results of pixel based segmentation methods like Background Subtraction with spatial information, so that the results better reflect the smoothness of natural images. For this purpose we computed pixel trajectories over several frames of the video with Dense Optical Flows and used this to fuse the data of several successive frames. This decreased the vulnerability of our segmentations to noise as the flaws in a single measurement (frame) could be corrected by the data obtained from the other frames. At the same time, this approach increased the smoothness of the segmentations because the foreground probabilities are smoothed over several frames and the smoothness of the Dense Optical Flows is also gradually transferred to the segmentation.

The usage of the trajectories derived from the Optical Flows to adjust the foreground probabilities for the pixels and the spatial smoothing based the differences of

the probabilities in a 3×3 neighborhood further improved the results. In the future, we would like to develop a Dense Optical Flow algorithm which is especially designed for this approach and the circumstances. It should incorporate the knowledge about the static camera, so that large parts of the flow can be assumed to be zero. Also, to compute better trajectories we want to use several frames at once and compute the best trajectory over these instead of only computing the flow between two frames at a time. We hope to increase the robustness of the trajectories in this way and hence enlarge the benefit of our algorithm for the segmentations.

5 REFERENCES

[Benfold and Reid, 2011] Benfold, B. and Reid, I. (2011). Stable multi-target tracking in real-time surveillance video. In *CVPR*, pages 3457–3464.

[de Carvalho et al., 2010] de Carvalho, M., da Costa, A., Ferreira, A., and Marcondes Cesar Junior, R. (2010). Image segmentation using component tree and normalized cut. In *Graphics, Patterns and Images (SIBGRAPI), 2010 23rd SIBGRAPI Conference on*, pages 317–322.

[Duraisamy and Jane, 2014] Duraisamy, M. and Jane, F. (2014). cellular neural network based medical image segmentation using artificial bee colony algorithm. In *Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on*, pages 1–6.

[El Harrouss et al., 2015] El Harrouss, O., Moujahid, D., and Tairi, H. (2015). Motion detection based on the combining of the background subtraction and spatial color information. In *Intelligent Systems and Computer Vision (ISCV), 2015*, pages 1–4.

[Farnebäck, 2003] Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. In Bigun, J. and Gustavsson, T., editors, *Image Analysis*, volume 2749 of *Lecture Notes in Computer Science*, pages 363–370. Springer Berlin Heidelberg.

[Radolko et al., 2015] Radolko, M., Farhadifard, F., Gutzeit, E., and von Lukas, U. F. (2015). Real time video segmentation optimization with a modified normalized cut. In *Image and Signal Processing and Analysis (ISPA), 2015 9th International Symposium on*, pages 31–36.

[Radolko and Gutzeit, 2015] Radolko, M. and Gutzeit, E. (2015). Video segmentation via a gaussian switch background-model and higher order markov random fields. In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications Volume 1*, pages 537–544.

[Sheikh et al., 2009] Sheikh, Y., Javed, O., and Kanade, T. (2009). Background subtraction for freely moving cameras. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1219–1225.

[Soeleman et al., 2012] Soeleman, M., Hariadi, M., and Purnomo, M. (2012). Adaptive threshold for background subtraction in moving object detection using fuzzy c-means clustering. In *TENCON 2012 - 2012 IEEE Region 10 Conference*, pages 1–5.

[Tabkhi et al., 2015] Tabkhi, H., Sabbagh, M., and Schirner, G. (2015). An efficient architecture solution for low-power real-time background subtraction. In *Application-specific Systems, Architectures and Processors (ASAP), 2015 IEEE 26th International Conference on*, pages 218–225.

[Tang and Miao, 2008] Tang, Z. and Miao, Z. (2008). Fast background subtraction using improved gmm and graph cut. In *Image and Signal Processing, 2008. CISP '08. Congress on*, volume 4, pages 181–185.

[Toyama et al., 1999] Toyama, K., Krumm, J., Brumitt, B., and Meyers, B. (1999). Wallflower: Principles and practice of background maintenance. In *Seventh International Conference on Computer Vision*, pages 255–261. IEEE Computer Society Press.

[Wan and Wang, 2010] Wan, Q. and Wang, Y. (2010). Detecting moving objects by ant colony system in a map-mrf framework. In *E-Product E-Service and E-Entertainment (ICEEE), 2010 International Conference on*, pages 1–4.

[Zivkovic, 2004] Zivkovic, Z. (2004). Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2 - Volume 02*, ICPR '04, pages 28–31, Washington, DC, USA. IEEE Computer Society.

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

This research has been supported by the German Federal State of Mecklenburg-Western Pomerania and the European Social Fund under grants ESF/IV-BM-B35-0006/12 and V630-S-179- 2013/238.

left picture of Figure 1	[Zivkovic, 2004]	[Zivkovic and Heijden, 2006]	only GSM	our Approach
MCC	0.8324	0.8591	0.8543	0.8789
F1-Score	0.9899	0.9890	0.9894	0.9916

right picture of Figure 1	[Zivkovic, 2004]	[Zivkovic and Heijden, 2006]	only GSM	our Approach
MCC	0.8292	0.8598	0.7908	0.8937
F1-Score	0.9879	0.9899	0.9816	0.9930

Table 1: Evaluation of the segmentations shown in Figure 1 with two algorithms from Zivkovic for comparison.

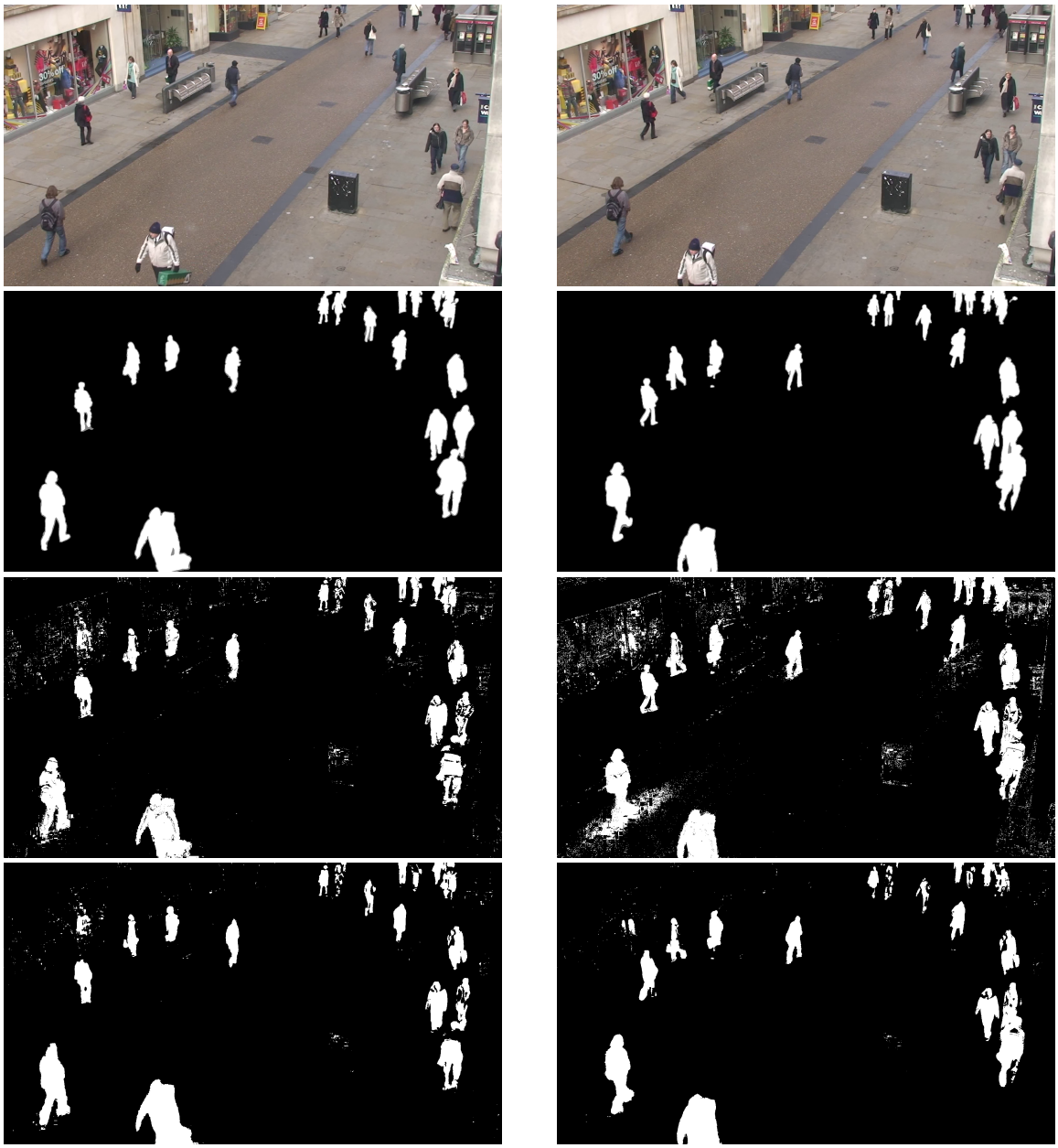

Figure 2: Shown are (top to bottom): original frame, ground truth, GSM Background Subtraction Segmentation, our result