Background Modeling: Dealing with Pan, Tilt or Zoom in Videos

Martin Radolko
University Rostock
Joachim-Jungius Str. 11
martin.radolko@igd-r.fraunhofer.de

Fahimeh Farhadifard
University Rostock
Joachim-Jungius Str. 11
Fahimeh.Farhadifard@igd-r.fraunhofer.de

Uwe Freiherr von Lukas
University Rostock
Joachim-Jungius Str. 11
uwe.freiherr.von.lukas@igd-r.fraunhofer.de

ABSTRACT

Even simple camera movements like pan, tilt or zoom constitute enormous problems for background subtraction algorithms since the modeling of the background works only under the assumption of a static camera. The problem has been mostly ignored and other algorithms have been used for videos with non-static cameras. Nonetheless, in this paper we introduce a method that adapts the background model to these camera movements by using affine transformations in combination with a similarity metric, and thereby the algorithm makes background subtraction usable for these situations. Also, to keep the generality of this approach, we first apply a detection step to avoid unnecessary adaptions in videos with a static camera because even small adaptions might otherwise deteriorate the background model over time. The method is evaluated on the extensive *changedetection.net* data set and could reliably detect camera motion in all videos as well as precisely adapt the model of the background to that motion. This does improve the quality of the background models significantly which consequently leads to a higher accuracy of the segmentations.

Keywords

Background Subtraction, Background Modeling, Video Segmentation, Change Detection, Pan, Tilt, Zoom

1 INTRODUCTION

Segmentation in videos is a particularly difficult problem of the computer vision field. Often it is the first step in a whole pipeline and all further methods are dependent on the exactness of the segmentations. The aim is to identify areas of interest in the video which can be processed further for classification, event detection and so forth. Therefore, the task for videos is usually to create a simple binary segmentation with areas of interest (foreground) and uninteresting parts (background).

The easiest scenario to create such a segmentation is a static camera because it allows the modeling of the background scene. By subtracting this model from the current frame accurate foreground-background classifications can be created in real time. For moving cameras the task becomes far more complicated and only a few algorithms have been proposed so far. Often the background modeling is skipped altogether in this scenario and instead the segmentation relies on other cues, for example the optical flow.

In this paper, we propose an addition to a background subtraction method that adapts the model to pan, tilt and zoom motions of the camera. An evaluation is done on the *changedetection.net* data set, which contains several videos with a panning or zooming camera as well as videos captured by a shaky camera. Until now, these videos have been handled with the normal algorithms for static scenes and the results were consequently unsatisfactory. Our algorithm can detect these events precisely and then lets the model mimic the motion of the camera which improved the segmentation results during camera motion without influencing the normal background subtraction for static scenes.

This research has been supported by the German Federal State of Mecklenburg-Western Pomerania and the European Social Fund under grant ESF/IV-BMB35-0006/12]

2 STATE OF THE ART

One of the most frequently used state of the art algorithms for change detection is the Mixture of Gaussian (MoG) method proposed by Stauffer et al. in [SG99]. There, the model is build of several Gaussians so that even complex scenes (swaying trees, changing lightning conditions) can be modeled accurately. Often they are used in combination with other approaches, e.g. in [WBSP14] together with a Flux Tensor. The Flux Tensor gave them a second, completely independent, cue to

their segmentation process and at the same time made the results spatially more consistent. This is important as the standard MoG is completely pixel-wise and no spatial coherency is present. A different approach for this problem was proposed in [VMZ13], there the MoG itself is enhanced with spatial method so that neighboring pixels also influence each other during the modeling process.

All of the aforementioned methods perform quite good on the standard change task for static videos but do not take into consideration the special cases of the pan, tilt and zoom videos or shaking cameras of the *changedetection.net* data set. One approach that took this into consideration is the background subtraction from [SC15], where the training images are first clustered into N groups using $K-means$ and then on each group a Single Gaussian background modeling is applied. From these different models the best is selected with a correlation coefficient, which increases the robustness to pan, tilt or zoom movement. In the overall performance the method was mediocre on the *changedetection.net* data set but on the pan, tilt and zoom videos it excelled all previous approaches.

A different approach was suggested in [FM14], where a background model based on a Neural Network is created which tries to mimic the self-organized learning behavior of the human brain. Afterwards, the model is adapted to camera movements by computing the transformation between the frame I_t and the previous frame I_{t-1} and applying it to the model.

Keypoints were used for the registration of the background model to the current frame of the video in [ACF+16]. The keypoints are matched by a K-nearest neighbor approach with the Hamming distance. However, as this method is in general prone to errors some safeguards had to be included, e.g. only pan, tilt or zoom transformations are allowed or the requirement that there are more keypoints in the background of the scene than on foreground objects.

In [ScSCJ16] motion vectors from an optical flow are used to adapt the background model to camera motion, but still only simple movements like pan or tilt can be handled easily with this method. A complete segmentation approach based on the optical flow was suggested in [OMB14]. The objects are segmented solely based on their movement vectors and therefore the method inherently can handle moving or shaking cameras quite well. The disadvantages of this approach are the slow computation time and the need of large batches of frames for an accurate calculation of the movement vectors.

3 PROPOSED APPROACH

Our approach to adapt a background subtraction algorithm for videos with pan, tilt or zoom camera movements consists of two phases. The first step is the detection of times in the video when one of the aforementioned camera movements is present. If such an event was detected the second step will compute the exact affine transformation of the background model to the current scene so that the model can be adapted correctly.

3.1 Detection of Pan, Tilt or Zoom

At first glance the detection step seems redundant, because when there is no pan, tilt or zoom of the camera the adaption step would just compute an affine transformation of zero (or very close to it) and therefore could be ignored. This is true in theory but in practice these adaptions pose a problem to the background modeling for two reasons. First, as the adaption step adapts the model of the background subtraction algorithm the comparison will always be between the current frame of the video and this model, Therefore, false detections of a panning, tilting or zooming camera can occur due to a bad background model or the presence of large foreground objects. The second problem is small misdetection e.g. due to moving objects or rounding errors at edges, which can cause small adaptions. Usually the effect of these misdetections is repaired by the algorithm itself in the next frame. However, the bigger problem of these misdetections is that every adaption of the background model is an affine transformation and causes small errors and deformations on the model which accumulate over time and deteriorate the model instead of making it better.

Hence, a good detection algorithm is necessary so that the adaption of the model only happens when it is necessary. For the detection we compare the current frame t of the video with the frame $t-2$. We chose the frame $t-2$ instead of just the previous frame $t-1$ because the constant changes like pan or tilt are more prominent then and easier to detect. This is of course very dependent on the speed of the pan or tilt as well as on the frame rate of the video, and therefore for other situations a bigger or smaller distances between the two frames which are compared might be appropriate.

The comparison between the two frames is done by applying different affine transformations on one of the frames and comparing the result with the other. The affine transformation that creates the best match is then taken as the true transformation between these two frames. To limit the possibility space of the affine transformations we confine ourselves to two basic transformations that correspond to a panning, tilting or zooming camera. First, different translations are evaluated and afterwards a transformation that corresponds to zoom is applied on top of the optimal translation.

To compare the two frames we use a similarity metric that counts the number of outliers. The reasoning behind this is that there is a natural variation between two

frames of a video, e.g. due to camera noise, but this change is usually very small. Therefore, every pixel that fulfills the following inequality

$$\|I_1(x,y) - I_2(x,y)\|_2^2 > T_{outlier}, \quad (1)$$

is counted as an outlier. In our case $T_{outlier}$ was set to 10 based on experiments. In the equation $I_1(x,y)$ is the pixel at position (x,y) of the first frame. In the end the affine transformation that produces the least outliers is taken.

With this method we obtain for every frame t a transformation that adapts it to the frame $t-2$. The extent of the transformation is then expressed in the number

$$\tau = \tau_T + w_z \cdot \tau_Z, \quad (2)$$

where τ_T is the amount of pixels that the image was translated (in x and y direction), τ_Z is the number of pixels that a corner pixel of the frame was moved by the zoom transformation and w_z is a weight parameter that controls the impact of them. As the number τ_Z is substantially smaller than τ_T for standard zoom and translation motions in videos, we gave the zoom a higher weight so that both effects have a similar influence (heuristically we choose $w_z = 5$).

The number τ is now a good measure for the camera movements we want to detect but still too sensitive to single outliers, e.g. the sudden appearances of large objects or shadows can cause erroneous calculations of τ. Therefore, we also take past calculations of τ into account to weaken the impact of single errors. This is done in the fashion of running Gaussian update by

$$\tau^{new} = (1-\alpha) \cdot \tau^{old} + \alpha \cdot (\tau_Z + w_z \cdot \tau_T). \quad (3)$$

The update rate α was set empirically to 0.1 as this allowed the detection of camera movements usually after 3 or fewer frames and is also small enough to eliminate most outliers. A smaller update rate would eliminate even more false detections but also increases the delay between the occurrence of camera movement and the detection of them by our algorithm. If this delay gets too large, an adaption of the (still untouched) background model to the already changed (over several frames) scene becomes increasingly difficult. For our data $\alpha = 0.1$ is a good compromise.

The parameter τ is very dependent on the resolution of the camera and therefore the threshold T_τ for τ should reflect this. We make the threshold depend on the *height* of the image and assume a detectable pan or tilt to have at least the size of one pixel. Since the smallest videos in the data set have a *height* of 240 pixels, this translates to the threshold $T_\tau = \frac{\text{height}}{240}$. If τ becomes larger than that a camera movement is detected and the background will get adapted accordingly (see next section). However, the threshold for τ depends also on the frame rate and the zoom or rotation speed of the camera and therefore cannot be used universally for other data sets.

3.2 Adapting the Background Model

After the successful detection of pan, tilt or zoom events the next step is the adaption of the background model. It would be easier to make this the other way around, adapt the new frame to the existing background model, because then the model can stay untouched. However, this is only possible when the camera is shaking slightly (e.g. due to wind or vibrations). When a real pan, tilt or zoom occurs this would lead, after a short time, to a situation where the incoming frame shows a completely different scene than the background model and then no adaption would be possible anymore.

To adapt the model, the first step is to extract an image from the statistical model of the background that reflects the current background so that the best affine transformation between this image and the current frame of the video can be computed. The background model in our case is created with the Gaussian Switch Model (GSM) which is a special gaussian model [RG15]. There, each background pixel is represented by two Gaussian and for every pixel and every channel we take the mean value of the currently active Gaussian and use it to create the most likely representation of the background.

Afterwards, the best affine transformation between this representation of the background and the current frame should be found. A higher accuracy (sub-pixel scale) version of the algorithm from the detection phase is used because even small errors accumulate over time and should be avoided as much as possible. The objective function used now is also different, instead of the outlier detection from Equation 1 we use

$$\sum_{x,y} \|I_1(x,y) - I_2(x,y)\|_2^2. \quad (4)$$

This function is a more exact measure of the difference between two frames and therefore better suited to determinate the precise direction and amount of movement. To lessen the impact of large foreground objects, which could disturb the accurate detection, we use the last segmentation derived from the background subtraction and exclude areas which are marked as foreground.

Similar to the first phase, we begin by looking for a translational deformation. For every direction we allow up to 10 pixels translation. To speed the process up we first look coarsely (2 pixels steps) over the whole 20 pixel range of one axis and then refined the result subsequently. This accounts already for pan, tilt or shaky

cameras but not for zoom. The zoom motion is evaluated afterwards on top of the optimal translation and the step size here is $\frac{1}{5}$ pixel. Combined, these transformations give the optimal adaption of the background model to the current frame.

This adaption now has to be applied on the complete background model, so that afterwards the segmentation of the scene and the updating of the model can succeed. For the GSM there are two Gaussian models, and the transformation has to be applied on both of them. One Gaussian model consists of two values for each pixel in the frame, mean and variance. Only jointly these values can give a comprehensive model of the background and therefore they cannot be separated by the transformation (e.g. by rounding errors) but have to represent a new pixel together after the affine transformation.

The borders pose a special problem in the adaption stage since if the background model is moved 5 pixels to the left there will be an empty space with no information on the right side of the model. After the affine transformation we identify these areas and fill them with information from the current frame of the video. The mean of the Gaussians will be set to the color value of the frame and the variance to a fixed and high value, 0.01 in our case. This ensures that the new area will be considered background in the following segmentation which is the best assumption we can make.

Lastly, to deal especially with short but fast camera motions, we evaluate the amount of foreground during times of camera motion. If there are more than 50% of the pixels classified as foreground we suppose that the model does not reflect the current scene anymore due to strong camera motion. In this case we reset the whole model and retrain it.

4 RESULTS

The method is evaluated on the comprehensive *changedetection.net* data set. It consists of 53 videos in eleven categories and two of them are *PTZ* (Pan, Tilt and Zoom) and *Camera Jitter*. In these categories there are a total of 8 videos in which the camera either is shaking or exhibits pan, tilt or zoom motions and therefore our algorithm should detect camera movements in these videos and adapt the background model accordingly. Each of the videos consists of three parts, they begin with a learning phase, then an evaluation phase for which the ground truth data is provided and lastly a part for which the ground truth data is not publically available. In this paper we only use the first two phases to assess the impact of our algorithm.

Detection Phase

In the first step we measure the detection accuracy of the proposed algorithm and afterwards the effect of the background model adaption in the *PTZ* and *Camera Jitter* videos is evaluated. The detection rate is measured by manually marking all frames with camera movement and comparing this with the results of the proposed algorithm. The results can be seen in Table 1. Since most of the videos in the data set do not exhibit camera movement there are no *True Positives* (frames in which camera movement occurred) in most categories and hence also no *false negatives*. It is vice versa in the *Camera Jitter* category since here the camera is constantly shaking. Only the *PTZ* videos contain both, times with a static camera and times with a moving camera.

The results show that the detection accuracy is very high, only in one video in the *Turbulence* category there are over 1000 false detection due to a severe heat shimmer which is difficult to differentiate from shaking. Therefore, the videos without camera movements will get hardly disturbed by the algorithm as there are only very few unnecessary background model adaptions. The detection of actual camera movements is also reliable, in the *PTZ* category the *False Negatives* are only because of the detection delay at the beginning of a movement (see equation 3). The results for the *Intermitten Pan* video are shown in detail for the first 1300 frames in Figure 1. There the detection delay at beginning can be seen as well as the false detections between the camera movements. These false detections are not a serious problem as the affine transformation applied there is usually close to zero. They would only become a problem if they would occur over longer periods without camera motion because even small transformations would then create blur effect on the model and thereby diminishes its quality.

Adaption Phase

After showing that the impact on videos with a static camera is basically nonexistent because of the few false detections, the effect of the background model adaption on the segmentations is evaluated. First the GSM background subtraction from [RG15] is used to segment the videos from the *PTZ* and *Camera Jitter* category and afterwards the same GSM algorithm is used in conjunction with the proposed model adaption. The results can be seen in Table 2 and the Figures 2 and 3 show examples from a video with a panning and zooming camera respectively.

Especially for the *PTZ* videos the results show a significant improvement over the normal background subtraction method, whereas the results for the *Camera Jitter* category show only a minor improvement overall. The reason for this is that the background modeling can deal with a shaking camera quite well, it just learns a slightly blurred version of the scene, and therefore gives still good results even without adapting the background

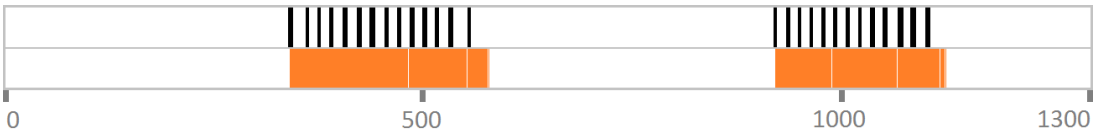

Figure 1: A detailed view on the detection accuracy on the *Intermitten Pan* video of the *PTZ* category. Shown are the results for the first 1300 frames. The top row shows the ground truth data, white areas represent times with no camera movement and black areas signify times with camera movement. The bottom row illustrates the results of our algorithm and orange areas signify detected camera motion.

model to the shaking. The adaption does improve the background model in a way that it is sharper and does include more details but this does not have a great impact on the segmentation quality.

In the pan, tilt and zoom videos the proposed algorithm shows a massive improvement because standard Background Subtraction cannot deal with these widespread camera movements. In this case the model of a normal background subtraction does not become only blurry but instead does not adapt to the movement at all (see Figure 2) or is so heavily blurred that it will not reflect the scene anymore (Figure 3). Hence, the very bad F1-Scores in this category. With our adaption algorithm the background models recreate the movements of the camera and therefore stay sharp and accurate which consequently improves the segmentation quality substantially.

The whole process of comparing different affine transformations is computational quite demanding, for a 720×480 frame the detection phase took about 0.05 seconds but if a camera movement was detected the adaption would then take another 0.8 seconds.

Figure 2: In the top row are 3 frames from the *zoomIn-zoomOut* video of the *changedetection.net* data set. The second row shows corresponding representations of the background model if the normal GSM is applied. There is almost no change in the background model although the camera zooms out because the model was trained already over a long time and adaptions now take a long time to happen. The last row shows the model with the addition of the proposed algorithm where the model adapts to the zooming motion.

5 CONCLUSION

In this paper an approach for the adaption of background models to pan, tilt or zoom camera movements is proposed. The method consists of two steps and the first part is the detection of camera movement. This is important to avoid unnecessary and potentially inaccurate adaptions of the model. We have shown on the large *changedetection.net* data set that the proposed detection works very accurately and therefore the whole algorithm does barely affect the background subtraction on videos with a static camera. If camera movement is present, the second phase adapts the background model to the slightly changed scene due to the moving camera movement and thereby we can improve the background model consistently. These sharper and more accurate models lead to overall significantly increased segmentation qualities when camera movement is present.

6 REFERENCES

[ACF+16] Danilo Avola, Luigi Cinque, Gian Luca Foresti, Cristiano Massaroni, and Daniele Pannone. A keypoint-based method for background modeling and foreground detection using a {PTZ} camera. *Pattern Recognition Letters*, 2016.

[FM14] A. Ferone and L. Maddalena. Neural background subtraction for pan-tilt-zoom cameras. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(5):571–579, May 2014.

[OMB14] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1187–1200, June 2014.

[RG15] Martin Radolko and Enrico Gutzeit. 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, 2015.

[SC15] H. Sajid and S. C. S. Cheung. Background subtraction for static amp; moving camera. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 4530–4534, Sept 2015.

[ScSCJ16] Hasan Sajid, Sen ching S. Cheung, and Nathan

Category	Bad Weather	Baseline	Camera Jitter	Dynamic Background	Intermittent Object	Shadow	Thermal	Low Framerate	Night Videos	PTZ	Turbulence	Overall
TN	11950	6050	0	18874	18629	16950	21100	6503	10466	3270	8034	121826
TP	0	0	5204	0	0	0	0	0	0	1641	0	6845
FP	0	0	1216	0	0	0	0	0	0	221	0	1437
FN	0	0	0	0	21	0	0	297	0	738	1566	2622

Table 1: Results of the camera movement detection of the proposed algorithm. The *changedetection.net* was used for the evaluation and shown are the numbers of frames that were correctly or falsely classified. For example *True Negatives* are the frames in which no camera movement was present and which were correctly classified as such.

Video	F1-Score		Video	F1-Score	
Camera Jitter	only GSM	proposed approach	PTZ	only GSM	proposed approach
Badminton	0.8476	0.8733	continuousPan	0.0909	0.5501
Boulevard	0.5983	0.6266	intermittenPan	0.0768	0.5744
Sidewalk	0.5423	0.5228	twoPositionPTZCam	0.1988	0.7327
Traffic	0.7288	0.7528	zoomInzoomOut	0.0221	0.3781

Table 2: Compared are the results of the GSM background subtraction with and without the addition of proposed algorithm. A clear improvement can be seen, especially for the *PTZ* videos.

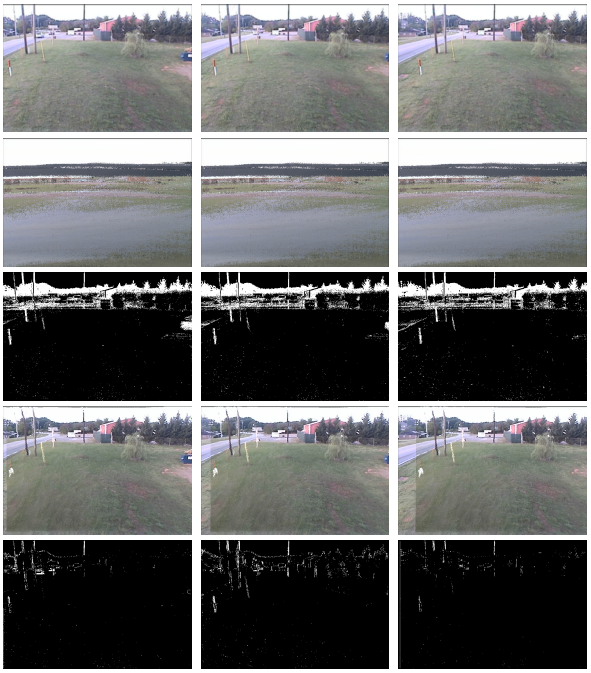

Figure 3: Similar to Figure 2 are here shown the results of the *continuousPan* video. The background model created with the normal background subtraction algorithm in the second row is extremely blurry because of the constant panning of the camera. Hence, the segmentation results show a lot of false detections. The proposed adaption to this pan makes the background model sharper and more accurate so that most false detection vanish (row four and five).

Jacobs. Appearance based background subtraction for {PTZ} cameras. *Signal Processing: Image Communication*, 47:417 – 425, 2016.

[SG99] Chris Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Vol. Two*, pages 246–252, 1999.

[VMZ13] S. Varadarajan, P. Miller, and H. Zhou. Spatial mixture of gaussians for dynamic background modelling. In *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 63–68, Aug 2013.

[WBSP14] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan. Static and moving object detection using flux tensor with split gaussian models. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 420–424, June 2014.