

Motion-based Object Segmentation using Sprites and Anisotropic Diffusion

Andreas Krutz, Matthias Kunter,
Thomas Sikora, TU Berlin
Commun. Systems Group
Berlin, Germany
krutz@nue.tu-berlin.de

Mrinal Mandal
University of Alberta
Department of EE
Edmonton, Canada
mandal@ece.ualberta.ca

Michael Frater
Univ. of New South Wales
School of IT and EE
Canberra, Australia
m.frater@adfa.edu.au

Abstract

Many algorithms have been developed to recognize regions, edges, color, and objects in images and videos. For applications like surveillance or object-based video coding, it is important to segment the foreground objects from the background. The task is very challenging in the case of a moving camera. We present a foreground segmentation approach that is designed for sprite coding as well as other applications, e.g. video surveillance. Accurate frame-to-frame image registration and sprite generation build the pre-processing step. The segmentation algorithm operates on error images, which are produced by the image registration and subtraction from reconstructed background frames. It is processed in several steps including low-pass filtering using anisotropic diffusion. Experiments show excellent results with single- and multi-view test sequences.

1. Introduction

Object segmentation is a key technique in applications such as object-based video coding and video surveillance. There are many approaches to segment or track foreground objects in scenes recorded by a still camera, such as in surveillance applications. In those scenarios, the background can be modeled and thus, a background subtraction method can be applied. However, the task of finding a pre-processing method for separating the foreground and the background in scenes with moving camera, e.g. pan, tilt, roll is more difficult. The use of image registration and sprite generation techniques are a good base to tackle this problem. If a specific camera motion model, e.g. the perspective model, can be applied, it is possible to generate a sprite from this sequence. Here a sprite is a storage where all aligned frames are summarized in one image [6]. All considered frames are blended into one image controlled by higher-order motion parameters. For the blending process, a filter removing all foreground objects is used. As

a result, the sprite contains the background of a number of frames from this video sequence. The next step is to reconstruct the frames from the sprite. This generates a video sequence only containing the background. If these frames are used for subtraction with the original frames for the ideal case only the foreground object pixels appear in the resulted error frame [1]. Several other applications using sprites, e.g. super-resolution, are possible as shown in [3]. During the sprite-generation step it can occur that the blending filter produces several deviations in the reconstructed frames compared to the original frames, possibly without any impact on the subjective quality. However, if the reconstructed background frames are to be used for segmentation, these different pixel values appear in the error frame and influence the segmentation algorithm. Furthermore, the reconstructed frames can be distorted by the sprite generation process or by a non-static background. We address this problem by applying a second error frame pre-computed by a frame-to-frame short-term image registration algorithm. To achieve this, it is very important to use an accurate method for the precise estimation of the background. The utilized algorithm has been introduced in [2]. Thus, the segmentation algorithm is applied on both error frames.

The segmentation is conducted in several steps. Before the error frame is converted to a binary frame, it is low-pass filtered using anisotropic diffusion. This technique was first proposed in [5]. The advantage of this method is that small edges between adjacent pixel values are blurred whereas large differences of adjacent pixel values remain. The binary image is produced thresholding the filtered error image. In our approach, the threshold is calculated using the average and the maximum of the pixel values. In order to remove small objects and to fill holes in the main objects, several well-know morphological operations are applied on the binary image. The resulted binary masks are combined together.

The remainder of this paper contains a more detailed description of the proposed algorithm. Section 2 describes the pre-processing step where the frame-to-frame image regis-

tration and the sprite generation methods are introduced. Section 3 shows the functionality of the segmentation algorithm step by step. Experimental results are presented in Section 4. The conclusion presents the key features and results of the work and show perspectives for further work.

2 Image Registration and Sprite Generation for Object Segmentation

2.1 Image Registration

The core of the whole algorithm is a precise estimation of the background considering adjacent frames of the test sequence. Therefore, an image registration algorithm has to be used that accurately estimates the motion of the background object. An enhanced algorithm, which we apply here, has been proposed in [2]. The algorithm is based on a gradient-based minimization method. Two further important issues are included. First, phase correlation is applied for the initialization of the translational parameters. In order to reduce the computational complexity and deform the error surface, the input images are down-sampled using image pyramids. Second a windowing approach is applied, which is fundamental for sequences with large background occlusions. This is important since the segmentation has to rely on an exact image registration. The registration is processed frame-to-frame during the test sequences. It yields short-term higher-order motion parameters, e.g. the well-known 8-parameter perspective model, which set up the calculation of the long-term parameters for the sprite generation. Additionally the error frames for all adjacent frames are computed.

2.2 Sprite Generation and Background Reconstruction

We construct a background sprite of the scene shot by registration of every frame into the reference frame coordinate system. The long-term transformation parameters are computed by Levenberg-Marquardt gradient descent minimization of the image differences between the compensated frames and the sprite image. The concatenated short-term parameters are utilized as initial values for this process. We can then robustly blend all pixel candidates to remove the foreground objects. An overview of our sprite generation method is given in [3]. We produce single sprites as well as multiple sprites [4]. A background movie of the scene is produced by reprojecting the sprite using the inverse transformation parameters. Since camera-dependent illumination properties get lost, e.g. in scenes with dark border regions, the sprite background will not exactly match the original frame. We model the illumination characteristics

by a separable parabolic functionality in horizontal and vertical image direction to compensate this difference. The *illumination correction* value is obtained by $\Delta Y(x, y) = a_0 + a_1x + a_2x^2 + b_0 + b_1y + b_2y^2$. The correction parameters **a** and **b** are determined modelling the averaged illumination difference along the lines through the image center. Thus, calculating this additional six parameters for a shot we can significantly increase the background reconstruction quality.

3 Segmentation Approach Using Anisotropic Diffusion

The segmentation approach relies on pre-computed error frames. It depends on the significant difference of the pixel values of the foreground and the background object. The use of two error frames improves the segmentation results remarkably. Computing the error image using the reconstructed background frames from the sprite produces a precise shape of the foreground object. The drawback is that due to sprite generation further regions of larger values appear in the error frame which are not related to the foreground objects. To prevent this, it is very important to have a very good estimation of the background object. This can be achieved by the use of the error frame by the frame-to-frame image registration. Here the background object can be estimated more precisely while the foreground objects appear twice in the error image. Combining both calculated objects masks using a logical AND-operator removes all drawbacks from both approaches. The block chart of the whole segmentation algorithm is given in Fig. 1. The creation of the binary object mask is realized in several steps. Table 1 lists the steps explicitly. First, the error images are computed using the pre-processing methods described above. After applying the absolute value function, because several values can be negative, the resulting error image is low-pass filtered using anisotropic diffusion. A numerical algorithm was derived from the diffusion equation (1) for images in [5]. Two weighted functions for ρ can be applied at the image gradient to influence the diffusion in two different ways [5]. In this work, the function shown in (2) is used.

$$I_t = \text{div}(\rho(|\nabla I|)\nabla I) \quad (1)$$

$$\rho_1(\nabla I) = e^{-\left(\frac{\|\nabla I\|}{K}\right)^2} \quad (2)$$

Here ∇I is the image gradient and K is a fixed constant. Equation (2) shows the function which prefers higher contrast edges in comparison to lower contrast edges. In this work the interest is to define higher contrast edges which separate the foreground object from the background. Therefore, function 2 is taken for diffusing the error image. Afterwards, the pixel values of the resulted diffused image are

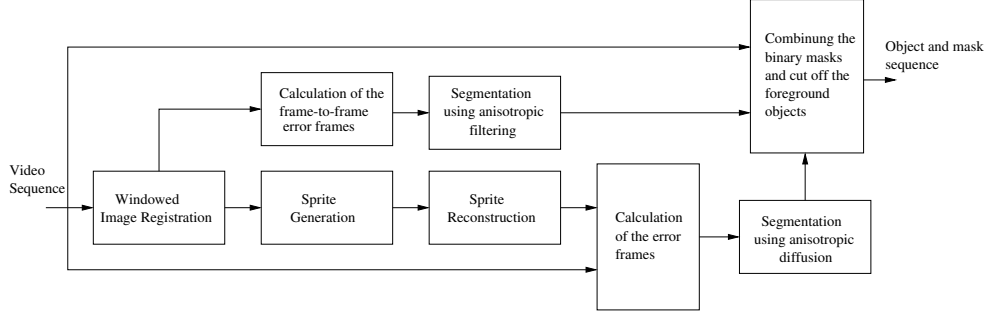


Figure 1. Motion-based Object Segmentation using Anisotropic Filtering

Table 1. Steps for binary foreground mask calculation

- 1) Compute the error images
- 2) Preprocessing : building the absolute value
- 3) Low pass filtering using anisotropic diffusion
- 4) Intensity rescaling
- 5) Creating binary image using thresholding
- 6) Small objects are removed
- 7) Small open holes are closed
- 8) Holes inside the objects are filled
- 9) Small objects are removed again
- 10) Both masks are combined using AND-operator

rescaled to the range of [0 1] as a pre-processing for creating a binary image. A threshold is then defined to separate the foreground pixels from the background. The approach for defining the threshold is very easy. The average of the rescaled image is computed. This value is added by a tuning constant which is fixed during the whole segmentation process over the video sequences. Several well-known morphological operations are then applied, e.g. removing small objects from the background and filling holes inside the foreground objects. The last step is the combination of the two achieved binary masks by the use of the AND-operator. The resulted binary mask cuts off the foreground object from the original image.

4 Experimental Results

Three test sequences are considered for the experimental evaluation. There are two single-view sequences, the sequence “Biathlon” which was recorded from a German TV broadcaster and the well-known “Stefan” sequence. The third test sequence considered is a multi-view MPEG-sequence produced by KDDI, “Race1”. We show the seg-

mentation results on two views to demonstrate the performance of the algorithm on multi-view videos. We compare our approach in Fig. 2 with using only the reconstructed background frames (using multi-sprites) and a technique based on optical-flow and anisotropic diffusion [3]. The background is completely removed by the proposed method for this example. Figure 3 shows an example of the sequence “Biathlon” and two views of the MPEG multi-view sequence “Race1”. It can be seen that the foreground objects are separated from the background almost accurately. As well as in the 2D-sequences, the background has completely vanished. Thus due to the overcome of several drawbacks discussed in Section 3 no errors appear in the segmentation mask. For an accurate measurement of the segmentation performance, ground truth is only available for 200 frames of the “Stefan” sequence. Two values are calculated to show a quantitative evaluation of the segmentation. The “precision” shows how the background is removed from the mask achieved. To show how precise the foreground object is segmented, the “recall”-value is computed. The equations of these two values are shown in (3).

$$P = \frac{TP}{TP + FP} \quad R = \frac{TP}{TP + FN} \quad (3)$$

$$F = 2 \cdot \frac{P \cdot R}{P + R} \quad (4)$$

P is the precision value, R is the recall value, TP are the “true-positive”-pixel values, FP are the “false-positive”-pixel values, and FN are the “false-negative”-pixel values. We calculated both values over 200 frames of the “Stefan”-sequence and the achieved values for the considered algorithms are shown in Table 2. Furthermore, we calculate the F-measure (4) to combine both values for one result. The F-measure values for each algorithm are also given in Table 2. It can be seen that the proposed algorithm significantly improves the whole segmentation result.

Table 2. Comparison of P, R, and F

Alg.	P in %	R in %	F
Multi-Sprite only	55%	93%	0.69
Optical-flow based[3]	60%	89%	0.72
Proposed	81%	85%	0.83

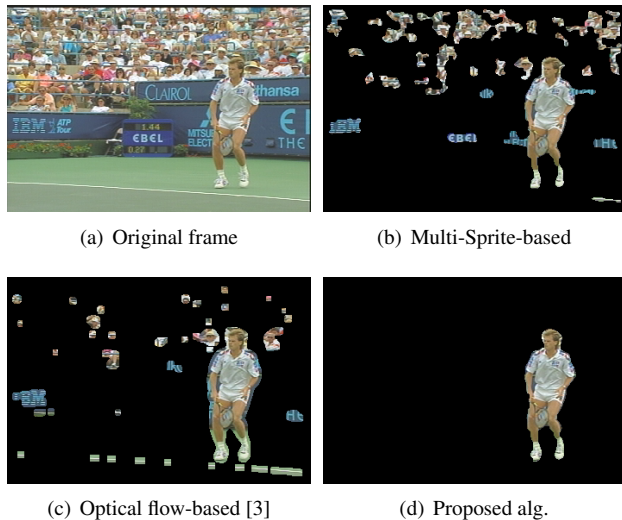


Figure 2. Different segmentation approaches (frame 129, sequence “Stefan”)

5 Conclusions

We have proposed a motion-based object segmentation algorithm. As a pre-processing step, a frame-to-frame image registration and a sprite is generated to separate all the background information from the foreground objects. A segmentation algorithm is applied on error frames which relies on a low-pass filter using anisotropic diffusion, re-scaling, and several morphological operations. This method works very well on the test sequences. Further work will be done to extend the approach using tracking and including multi-view content. The method could also be enhanced to exploit the color channels as well.

Acknowledgment

This research was supported by the European Commission under contract FP6-027026-K-SPACE.

M. R. Frater was supported in this work by the Australian Research Council under project DP0667074.

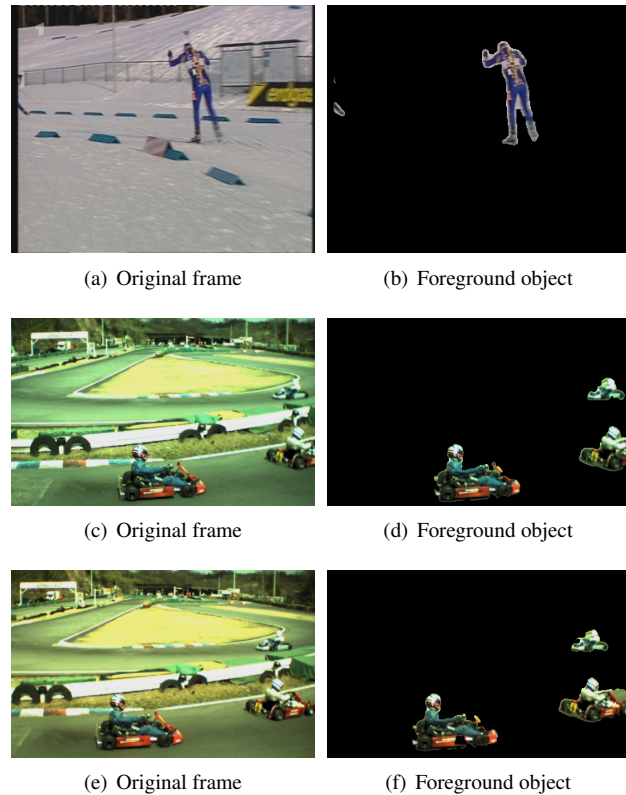


Figure 3. Results for “Biathlon” (top) and multi-view sequence “Race1” (frame 39, view 1 (middle), view 2 (bottom))

References

- [1] D. Farin, P. H. N. de With, and W. Effelsberg. Video object segmentation using multi-sprite background subtraction. In *Int. Conf. on Multimedia and Expo (ICME)*, Taipei, Taiwan, June 2004.
- [2] A. Krutz, M. Frater, M. Kunter, and T. Sikora. Windowed image registration for robust mosaicing of scenes with large background occlusions. In *Int. Conf. on Image Processing (ICIP06)*, Atlanta, USA, Oct. 2006.
- [3] M. Kunter, J. Kim, and T. Sikora. Super-resolution mosaicing using embedded hybrid recursive flow-based segmentation. In *IEEE Int. Conf. on Information, Communication and Signal Processing (ICICS'05)*, Bangkok, Thailand, Dec. 2005.
- [4] M. Kunter, A. Krutz, M. Mandal, and T. Sikora. Optimal multiple sprite generation based on physical camera parameter estimation. In *Visual Communications and Image Processing (VCIP'07)*, San Jose, USA, Jan. 2007.
- [5] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence archive*, 12:629–639, 1990.
- [6] A. Smolic, T. Sikora, and J.-R. Ohm. Long-term global motion estimation and its application for sprite coding. *IEEE TCSVT*, 9(8):1227–1242, December 1998.