

MISREGISTRATION ERRORS IN CHANGE DETECTION ALGORITHMS AND HOW TO AVOID THEM

Dirk Farin¹ and Peter H. N. de With^{1,2}

¹ Univ. of Technol. Eindhoven, PO Box 513
5600 MB Eindhoven, Netherlands
d.s.farin@tue.nl

² LogicaCMG, PO Box 7089
5605 JB Eindhoven, Netherlands
P.H.N.de.With@tue.nl

ABSTRACT

Background subtraction is a popular algorithm for video object segmentation. It identifies foreground objects by comparing the input images with a pure background image. In camera-motion compensated sequences, small errors in the motion estimation can lead to large image differences along sharp edges. Consequently, the errors in the image registration finally lead to segmentation errors. This paper proposes a computationally efficient approach to detect image areas having a high risk of showing misregistration errors. Furthermore, we describe how existing change detection algorithms can be modified to avoid segmentation errors in these areas. Experiments show that our algorithm can improve the segmentation quality. The algorithm is memory efficient and suitable for real-time processing.

1. INTRODUCTION

Video object segmentation is a prerequisite for a number of applications, such as surveillance, intelligent video databases, or object-oriented video coding. A popular approach within video object segmentation is the background subtraction algorithm. This algorithm assumes that a picture of the scene background is available, which does not show any foreground objects. The background subtraction algorithm compares the current input image with the background image and it detects foreground objects at places where the differences between both images are large.

From an algorithmic point of view, we can partition the applications into two classes, based on the typical type of video that is observed. The first class contains videos that are recorded with a static camera. In this case, the background image can be recorded explicitly, or it can be estimated by some temporal averaging process. The second class covers videos with a (rotationally) moving camera. Hence, the camera-motion has to be compensated prior to the background subtraction process.

For videos with a moving camera, the background image is usually generated by estimating the camera-motion and aligning all input frames into a large background sprite

image [4]. The background image can then be synthesized with a temporal averaging process. To carry out the background subtraction, the current camera view has to be extracted from this background sprite image and aligned to the current input image. After the camera-motion compensation, corresponding pixels in the foreground and the background should be co-located at the same position. However, in practice, small inaccuracies in the motion estimation can occur that lead to image *misregistration*. Even though this is usually less than a pixel distance, it can cause interpolation artifacts along sharp edges. If there is a large difference in brightness across the edge, a tiny inaccuracy in the motion model or aliasing in the input video can cause a large difference between the images. The same effect occurs for setups with static cameras, where the camera moves a little bit because of vibrations or wind in outdoor environments.

A similar misregistration problem is already known from remote sensing applications [3, 6] and a few algorithms have been proposed to reduce them. For example, in [2] an algorithm for multispectral images is proposed. The idea is to estimate the distribution of the registration noise for each pixel by synthetically generating a number of misregistered background images and comparing these images to the original one. However, the approach is designed for multispectral images. Moreover, the high computational complexity for estimating the distributions makes it inappropriate for real-time processing of sequences with moving camera.

To our knowledge, current algorithms for video object segmentation do not consider this misregistration error in their input and consequently, erroneous foreground objects are detected in areas with strong edges (for example, see Fig. 3(b)). This paper presents a new technique to detect areas that are likely to show misregistration errors, and it describes how standard change-detection algorithms can be modified to accommodate for these special areas.

We will further explore the misregistration effect and introduce the concept of a *misregistration risk map* in Section 2. The integration of these risk maps into background subtraction algorithms will be described in Section 3. Results are presented in Section 4 and the paper concludes with Section 5.

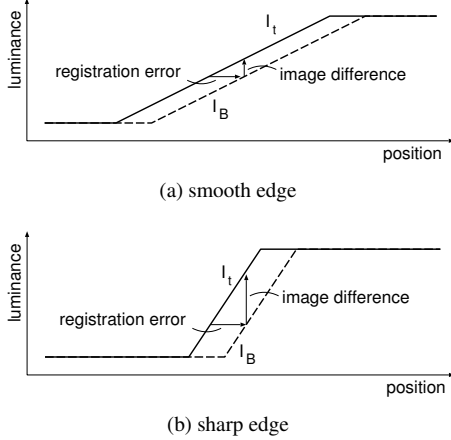


Fig. 1. Two edge-profiles, with (a) smooth, (b) sharp edges. At the sharp edge, a misregistration of two edges induces a larger luminance error than at the smooth edge.

2. MISREGISTRATION ERRORS

Change detection algorithms are usually based on the assumption that pixels in the input image I_t correspond exactly with the same pixels in the background image I_B . However, errors in the camera-motion compensation lead to small displacements of the pixels. These displacements are usually smaller than one pixel distance, but along strong edges, they may induce large values in the difference image (see Fig. 1).

As a first option to reduce this effect, we investigated to change the image difference measurement. Instead of using the direct luminance difference $d(x, y) = |I_B(x, y) - I_t(x, y)|$, we compensate for the expected misregistration along edges by dividing by the luminance gradient in the background image, leading to

$$d'(x, y) = \frac{|I_B(x, y) - I_t(x, y)|}{\|\nabla I_B(x, y)\|}. \quad (1)$$

Note that this defines the pixel difference as the horizontal distance that a pixel would have to be shifted to meet a pixel of equal luminance (under the assumption that the change of luminance in the neighborhood is linear). Even though this approach succeeds to reduce the misregistration errors, we observe two problems. In flat image areas (where $\nabla I_B(x, y) \approx 0$), the image difference signal is amplified excessively. On the other hand, the difference signal is small if there is high contrast in the background, but no texture in the foreground object. The latter case reduces the ability to detect foreground objects in front of textured background even though the foreground might have a clearly different color. However, since we can distinguish high-contrast texture areas in the background image from low-contrast areas in the foreground, this first algorithm is not optimal for the

described case.

For this reason, we propose an algorithm that employs a boolean image indicating for each pixel whether we expect large differences that result from misregistration. We denote this boolean image as the *risk map*. To determine the risk map, we consider the gradient strength in the background image as well as in the foreground image. This leads to the following four cases. (A) If the gradient strength is low in both the background image and the input image, misregistration has little effect to the difference image. (B) If the gradient is strong in both the background image as well as the input image, this is possibly because the input image shows the same content as the background image. Since both images show a strong edge at the same position, misregistration can lead to large differences and segmentation errors. (C,D) If only the background image or only the foreground image shows a strong edge, the image content varies from each other, so that large differences are not the result of misregistration.

As a consequence, misregistration errors will only have serious influence at pixels of case (B). Expressed as a formal rule, we detect those pixels if the gradient magnitude in the background image $\|\nabla I_B(x, y)\|$ exceeds a threshold τ_m and the input image gradient $\|\nabla I_t(x, y)\|$ also exceeds this threshold at the same position (x, y) . The boolean risk map R_M is constructed accordingly, using

$$R_M(x, y) = (\|\nabla I_B(x, y)\| > \tau_m) \wedge (\|\nabla I_t(x, y)\| > \tau_m). \quad (2)$$

We apply a simple $(1/2 \ 0 \ -1/2)$ gradient filter and a threshold of $\tau_m \approx 0.2$ for a maximum pixel value of 1.0.

3. SEGMENTATION ALGORITHM

In this section, we show how the misregistration risk map can be integrated into the change-detection algorithm proposed in [1]. We used this algorithm as part of our automatic video object segmentation system, which is described in [5]. Here, we will only describe the background subtraction process, where we assume that a suitable camera-motion compensated background sequence has already been computed.

The input data for the background subtraction algorithm is the current input image and a corresponding view of the pure background. First, the difference image $d(x, y)$ between both inputs is calculated. Instead of classifying the pixels independently as foreground and background pixels, we apply two algorithms that also consider the neighborhood of the pixel, to improve the robustness (see Fig. 2 for an overview). Both algorithms follow the approach described in [1], but are modified to integrate the previously computed risk map. The first is a χ^2 significance test to increase the robustness to camera noise. The result is used as initialization of a Markov random field based statistical

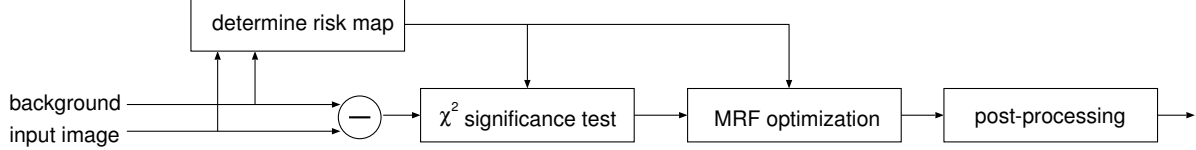


Fig. 2. Data-flow in the background subtraction algorithm.

MAP segmentation. After that, a morphological postprocessing step (which is not described here) removes small and instable clutter regions from the segmentation mask.

3.1. χ^2 significance test

The first algorithm assumes that all pixels in a small neighborhood window \mathcal{W} (of typically 5×5 pixels), are either changed or unchanged. Furthermore, it assumes that the camera noise is Gaussian. With these two assumptions, a χ^2 significance test can be carried out on the pixels in a small neighborhood. In this test, the sum of squared differences Δ in the neighborhood is compared to a threshold t_α . This threshold is obtained from the cumulative function of a χ^2 distribution $P_{\chi^2;|\mathcal{W}|}$ with $|\mathcal{W}|$ degrees of freedom and a chosen significance level α . More specifically, the threshold t_α is selected such that

$$P_{\chi^2;|\mathcal{W}|}(\Delta > t_\alpha) = \alpha, \quad (3)$$

where Δ is the sum of squared differences, normalized with the camera noise variance σ ,

$$\Delta = \sum_{(x,y) \in \mathcal{W}} (d(x,y)/\sigma)^2. \quad (4)$$

Considering several pixels in a neighborhood helps to increase the robustness to noisy pixels. However, this is not sufficient to compensate for the misregistration errors (see Fig. 3(b)). Hence, we modified the algorithm to include only those pixels in the computation that are not marked as risky in R_M . More specifically, we replace the computation of Δ to

$$\Delta = \sum_{(x,y) \in \mathcal{W}} \begin{cases} (d(x,y)/\sigma)^2 & \text{if } R_M(x,y) = \text{false}, \\ 0 & \text{if } R_M(x,y) = \text{true}. \end{cases} \quad (5)$$

The result of this processing step is taken as initialization for the successive Markov random field optimization.

3.2. Markov field based segmentation

The Markov field based segmentation employs a model about the probability of segmentation mask shapes. The probability that a pixel is *changed* increases with the number of changed pixels in its neighborhood, and vice versa. Assuming again that the luminance differences are Gaussian distributed with variance σ for unchanged and σ_c for changed

pixels, the decision rule

$$d(x,y)^2 \underset{u}{\underset{c}{\geq}} 2 \frac{\sigma_c^2 \sigma^2}{\sigma_c^2 - \sigma^2} \left(\ln \frac{\sigma_c}{\sigma} + (v_B(c) - v_B(u))B + (v_C(c) - v_C(u))C \right) \quad (6)$$

can be derived, where the right hand side determines the threshold for a specific pixel. The $\ln \sigma_c/\sigma$ term together with the factor in front of the parenthesis can be considered the base threshold for the segmentation. It is depending only on the noise variances σ, σ_c . Additionally, the segmentation threshold is shifted by considering the labels of the pixels in the neighborhood. The configuration of the pixels in the neighborhood is described by the values v_B, v_C while the strength of the regularization is controlled with the parameters B, C . See [1] for more details.

For pixels that were classified as risky, we cannot rely on the difference values $d(x,y)$, such that we classify these pixels only based on the shape prior. Specifically, we use the spatial context bias to decide if a pixel is more likely to be changed or unchanged. This leads to the decision function

$$0 \underset{u}{\underset{c}{\geq}} (v_B(c) - v_B(u))B + (v_C(c) - v_C(u))C \quad (7)$$

if the considered pixel is risky. Otherwise, Eq. (6) is used without modification.

4. RESULTS

To quantify the quality of the segmentation result, we compared the results with manually generated reference segmentation masks. Since soft shadows or motion blurs complicate the definition of reference masks, we classified the pixels into three classes: *background* pixels, *foreground* pixels, and *don't-care* pixels in unclear cases. These *don't-care* pixels were not considered in the evaluation.

Table 1 summarizes the improvements that we obtained for some example sequences. Generally, it can be observed that our algorithm can clearly reduce the segmentation errors that are due to image misregistration. At the same time, our algorithm slightly increases the number of errors in the foreground object if foreground and background are both textured. Pictures from two of the example sequences are shown in Figures 3 and 4. The *stefan* sequence is a well-known MPEG-4 test-sequence. Note that the *stefan* sequence shows almost no misregistration errors. Hence, the segmentation result of our algorithm for this sequence is similar to a segmentation without misregistration reduction.

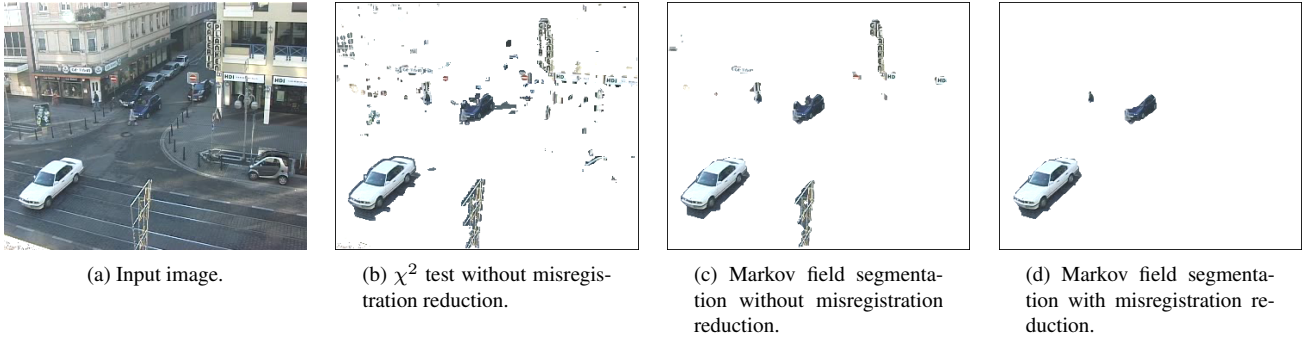


Fig. 3. Example sequence (*surveillance*). Misregistration is caused by a camera that is slightly moving in the wind.

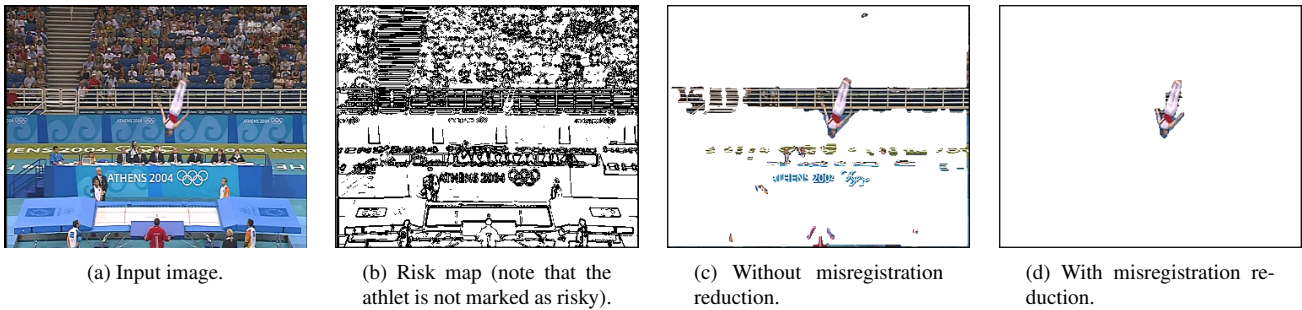


Fig. 4. Example sequence (*sport*) with camera-motion where severe misregistration errors occur along sharp edges.

	correct bkg. (%)	correct fgr. (%)	wrong bkg. (#)	wrong fgr. (#)
surveil. (A)	96.5	86.9	3478	116
surveil. (B)	99.9	79.3	42	241
sport (A)	87.2	100.0	13141	0
sport (B)	99.9	98.1	75	8
stefan (A)	99.6	97.3	365	108
stefan (B)	99.9	93.9	28	251

Table 1. Segmentation quality. (A) without misregistration reduction, (B) our algorithm. Depicted is the percentage of correct pixels and the average number of wrong pixels per frame.

5. CONCLUSIONS

We have described the misregistration effect that results from interpolation artifacts introduced in the camera-motion compensation. This effect usually leads to an increased number of false detections in change detection algorithms. A new algorithm was proposed to explicitly detect areas in which these misregistration effects are likely to occur. With this information available, standard change detection algorithms can be modified to adapt to these areas.

Our detector for misregistration errors considers the contrast and sharpness of edges. Furthermore it combines information about the local texture in the background and the

input image. Nevertheless, the proposed the algorithm is computationally efficient and suitable for integration into real-time segmentation systems.

6. REFERENCES

- [1] T. Aach and A. Kaup. Statistical model-based change detection in moving video. *Signal Processing*, 31:165–180, 1993.
- [2] L. Bruzzone and R. Cossu. An adaptive approach to reducing registration noise effects in unsupervised change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 41:2455–2465, 2003.
- [3] X. Dai and S. Khorram. The effects of image misregistration on the accuracy of remotely sensed change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 36:1566–1577, 1998.
- [4] D. Farin, P. H. N. de With, and W. Effelsberg. Minimizing MPEG-4 sprite coding-cost using multi-sprites. In *SPIE Proc. Visual Communications and Image Processing*, pages 234–245, 2004.
- [5] D. Farin, P. H. N. de With, and W. Effelsberg. Video-object segmentation using multi-sprite background subtraction. In *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, pages 343–346, 2004.
- [6] D. A. Stow. Reducing the effects of misregistration on pixel-level change detection. *International Journal on Remote Sensing*, 20(12):2477–2483, Aug. 1999.