ELSEVIER

Signal Processing 66 (1998) 203–217

SIGNAL PROCESSING

A noise robust method for 2D shape estimation of moving objects in video sequences considering a moving camera

Roland Mech*, Michael Wollborn

Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, Universität Hannover, Appelstraße 9A, 30167 Hannover, Germany

Received 29 April 1997; revised 28 September 1997

Abstract

For the coding of image sequences at very low bit rates, it has been shown that the coding efficiency can be increased by taking into account the shape of moving objects in the scene. Moreover, the upcoming ISO/MPEG-4 standard considers the 2D shape of moving objects not only for reasons of coding efficiency, but also to provide the user with so-called content-based functionalities. However, the perfect automatic segmentation of moving objects in image sequences is still an unsolved problem. In this paper, an algorithm for automatic, noise robust 2D shape estimation of moving objects in video sequences is presented, which considers a moving camera. The algorithm consists of four main steps. In the first step, a possibly apparent camera motion is estimated and compensated. By the second step, a possibly apparent scene cut is detected, and if necessary the segmentation algorithm is reset. In the third step, a change detection mask is estimated by a relaxation technique, using local thresholds which consider the state of neighbouring pels. By the fourth step, regions where the background has been uncovered by a moving object are estimated, using motion information from a displacement vector field. Finally, the resulting object mask is adapted to luminance edges in the corresponding image, in order to improve the shape accuracy. Furthermore, the temporal coherency of the object mask is improved by applying a memory. The proposed algorithm is compared to an approach for 2D shape estimation of moving objects from the literature, which does not use any edge adaptation or object memory and which uses only one global threshold. The experimental results show that the resulting object shapes from the proposed algorithm look subjectively much better than those from the reference algorithm.

Zusammenfassung

Es ist bekannt, daß die Effizienz der Codierung von Bildsequenzen bei geringen Datenraten durch Berücksichtigung der Form der bewegten Objekte in einer Szene erhöht werden kann. Darüberhinaus berücksichtigt der ISO/MPEG-4 Standard die 2D-Form bewegter Objekte nicht nur ausschließlich für die Codiereffizienz, sondern auch um dem Benutzer sogenannte content-based functionalities zur Verfügung zu stellen. Die perfekte automatische Segmentierung bewegter Objekte ist jedoch nach wie vor ein ungelöstes Problem. In diesem Beitrag wird ein gegenüber Rauschen robuster Algorithmus zur automatischen Schätzung der 2D-Form bewegter Objekte in Video-Sequenzen vorgestellt, welcher eine bewegte Kamera berücksichtigt. Der Algorithmus besteht im wesentlichen aus vier Schritten. Im ersten Schritt wird eine mögliche Kamerabewegung geschätzt und kompensiert. Der zweite Schritt detektiert einen möglichen Szenenschnitt und setzt gegebenenfalls den Segmentieralgorithmus in seinen initialen Zustand zurück. Im dritten Schritt wird mittels einer

* Corresponding author. Tel.: + 49-511-762-5308; fax: + 49-511-762 5333; e-mail: mech@tnt.uni-hannover.de.

0165-1684/98/$19.00
PII S0165-1684(98)00006-1

Relaxationstechnik, die unter Berücksichtigung der benachbarten Bildpunkte lokale Schwellwerte verwendet, eine Änderungsdetektionsmaske geschätzt. Im vierten Schritt werden solche Regionen detektiert, in denen durch Bewegung von Objekten Hintergrund aufgedeckt worden ist. Hierzu wird Bewegungsinformation aus einem Displacementvektorfeld verwendet. Schließlich wird die resultierende Objektmaske an Luminanzkanten im zugehörigen Bild adaptiert, um die Genauigkeit der Formschätzung zu erhöhen. Desweiteren wird die zeitliche Kohärenz der Objektmasken durch Verwendung eines Gedächtnisses verbessert. Der vorgeschlagene Algorithmus wird mit einem Ansatz aus der Literatur zur Schätzung der 2D-Form bewegter Objekte verglichen, der weder eine Kantenadaption noch ein Gedächtnis beinhaltet und nur eine globale Schwelle verwendet. Die experimentiellen Ergebnisse zeigen, daß die geschätzten Objektformen des hier vorgeschlagenen Algorithmus subjektiv deutlich besser aussehen, als die des Referenzalgorithmus.

Résumé

Dans le cadre du codage de séquences d'images à très bas débit, on a montré que l'efficacité du codage peut être améliorée en tenant compte de la forme des objets animés dans la scène. De plus, le futur standard ISO/MPEG-4 ne prend pas seulement en compte la forme 2D des objets en mouvement pour des raisons d'efficacité de codage, mais également pour offrir à l'utilisateur des fonctionnalités dites "basées sur le contenu". Cependant, la technique parfaite de segmentation automatique d'objets animés dans des séquences d'images reste encore un problème non résolu. Dans cet article, un algorithme est présenté pour la segmentation automatique et résistante au bruit d'objets animés dans les séquences vidéo, considérant de plus une caméra en mouvement. L'algorithme se décompose en quatre parties principales. Dans la première, le mouvement éventuel apparent de la caméra est estimé et compensé. Dans la seconde partie, une éventuelle coupure de scène apparente est détectée, et au besoin l'algorithme de segmentation est réinitialisé. Un masque de détection de changement est estimé dans la troisième partie par une technique de relaxation utilisant des seuils locaux qui prennent en compte l'état des pixels avoisinants. Dans la quatrième partie, les régions où l'arrière-plan ont été découverts par un objet en mouvement sont estimées en utilisant l'information de mouvement obtenue par un champ de vecteurs de déplacement. Le masque d'objet résultant est finalement adapté aux transitions de luminance dans l'image correspondante afin d'améliorer la précision de forme. En outre, la cohérence temporelle du masque d'objet est améliorée via l'utilisation d'une mémoire. L'algorithme proposé est comparé à une approche d'estimation de formes 2D d'objets animés trouvée dans la littérature. Cette dernière n'utilise pas d'adaptation des contours ni de mémorie d'objet mais uniquement un seuillage global. L'expérimentation montre que les formes d'objet résultant de l'algorithme proposé semblent subjectivement bien meilleures que celles de l'algorithme de référence.

Keywords: Very low bit-rate coding; Object-based image coding; Content-based; Videophone; MPEG-4; Segmentation; Shape estimation

1. Introduction

For the coding of image sequences at very low bit rates, standardized block-based hybrid coding techniques [5,12] transmit only texture and motion information in order to minimize the required transmission bit rate at a specific image quality. However, within the area of object-based analysis–synthesis coding [9,10,23,24] it has been shown that shape information with respect to arbitrarily shaped moving objects in the captured scene can be used to improve the coding efficiency, too. Moreover, the upcoming ISO/MPEG-4 standard [21,26] considers such 2D shape information not only for the coding efficiency but also to allow for the so-called content-based functionalities. Therefore, no longer only rectangular frames are transmitted, but the image sequence can consist of one or more 2D Video Object Planes (VOPs) [6,26]. These VOPs may be of arbitrary shape, so shape information has to be considered in the coding and decoding algorithms. Due to this VOP-based coding, specific parts of the scene can be addressed by the user and can be treated differently from other parts. This allows the realization of a couple of functionalities like object-based spatial and temporal scalabilty [30], users interacting with the image content, etc. However, it is not specified in the standard how the

VOPs are generated, i.e. where the segmentation information comes from. Since this task is independent of the standard, it is and will be open for the user. This makes it possible to use, e.g., colour-keying techniques, interactive segmentation or fully automatic algorithms, depending on the requirements given by the respective application.

Up to now, the automatic segmentation of moving objects in image sequences is an unsolved problem. Current approaches are either based on motion information, on texture information or on both, i.e. on motion as well as on texture information. In [15,25], an estimated displacement vector field is clustered using an affine motion model, resulting in a number of regions with similar motion. These regions are furtheron smoothed by a relaxation technique. However, since the estimation of the displacement vector field leads to errors especially at object boundaries, the segmentation is not accurate enough. Furthermore, this technique leads in general to oversegmentation, i.e. the number of regions is larger than the number of moving objects in the scene. More accurate results with respect to the estimation of object boundaries are achieved by segmentation techniques which consider motion and texture information, as e.g. proposed in [3,8,29]. E.g. in [8], an image is first segmented using its textural information, resulting in regions with similar texture. Then, these regions are merged depending on the similarity of their motion. As motion information, either a dense displacement vector field or an affine motion model for each region is used. Due to the successive application of texture and motion analysis, these techniques still have some problems especially at object boundaries. Therefore, the approach in [27,28] proposes a simultaneous estimation of the texture segmentation and a dense displacement vector field by minimizing the displaced frame difference of two successive images. One disadvantage of this technique is its high complexity due to the estimation of a dense displacement vector field. Furthermore, in general, this approach also results in an oversegmentation with respect to the number of moving objects in the scene. An approach which uses mainly the luminance difference of two successive images is presented in [11]. Here, after thresholding the difference image using a global threshold, motion information from an estimated displacement vector field is used to eliminate regions of background being uncovered. This technique avoids the problem of oversegmentation, since it distinguishes only moving foreground and static background. Drawbacks are first the use of a global threshold, which makes the algorithm very sensitive against noise. Second, since no texture or colour information is used directly for segmentation, the resulting object boundaries are not accurate enough.

In this paper, an automatic, noise robust segmentation technique for 2D shape estimation of moving objects in video sequences considering a moving camera is presented, which overcomes the problems of inaccurate object boundaries and noise sensitivity. In order to avoid the problem of oversegmentation, the presented approach is based on the method described in [11], which is therefore used as a reference. Objects are assumed to be non-overlapping. Furthermore, parts of the work are based on a former segmentation algorithm which has been developed in the framework of object-based analysis–synthesis coding [10] and has been lateron adapted to the *Simulation Model for Object-based Coding of the COST* 211^{ter} *Simulation Subgroup* (SIMOC) [7]. The proposed algorithm estimates and compensates possibly apparent camera motion, first. For that approach the eight-parameter motion model described in [11] is used. Then, a possibly apparent scene cut is detected, and if necessary the segmentation algorithm is reset. The presented algorithm is mainly based on a change detection step, using a local thresholding technique. Therefore, the relaxation technique proposed in [1,2] is used which takes into account local neighbourhoods. Furthermore, in order to get more time coherent segmentation results, a memory for the change detection masks is applied. After the change detection, an estimated displacement vector field is used in order to subdivide the changed area into moving object area and area of uncovered background. Finally, the boundaries of the resulting object regions are adapted to luminance edges of the current image in order to improve the accuracy of the estimated object shape. In order to cope with different kinds of sequences, the parameters used in the different steps adapt automatically to the current sequence, starting with appropiate initial values.

The proposed algorithm is still under investigation within the framework of an MPEG-4 core experiment on automatic segmentation. Results have been presented at the MPEG meetings in Firence [16], Tampere [17], Chicago [18] and Maceio [19] in 1996 and in Sevilla [20] and Bristol [22] in 1997. A previous version of this algorithm has been presented at ICASSP'97 [13]. This version has been extended by consideration of a moving camera and has been presented at WIAMIS'97 [14].

The paper is organized as follows. In Section 2, the principle of the algorithm is described. In Sections 3–6, the estimation and compensation of camera motion, the scene cut detection and the estimation of the change detection mask and the object mask, which represents the 2D shapes of the moving objects in the scene, are described, respectively. In Section 7, experimental results are given. Section 8 gives a conclusion of the main results of the paper.

2. Principle of the algorithm

A segmentation of each frame of an image sequence into non-overlapping moving objects (denoted as foreground) and static and uncovered background (denoted as background) is performed, considering a moving camera. Fig. 1 gives an overview of the proposed segmentation method.

The proposed segmentation method can be subdivided into the following four steps: by the first step, an apparent camera motion is estimated and compensated using an eight-parameter motion model [11].

In the second step, an apparent scene cut or strong camera pan is detected by evaluating the MSE between the two successive frames, considering only background regions of the previous frame. In case of a scene cut the algorithm is reset.

By the third step, an initial change detection mask (CDMi) between two successive frames is generated. In this mask, pels with image luminance changed due to a moving object are marked. After that, boundaries of changed image areas are smoothed by a relaxation technique [1,2], resulting in a mask, called CDMs. Thereby, the algorithm adapts frame-wise automatically to camera noise. While in [1] the variance of camera noise and the variance of the difference frame within object regions

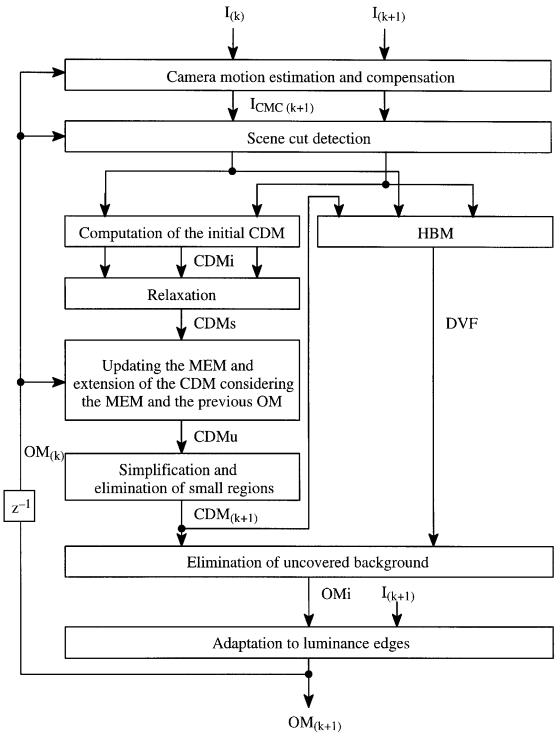

Fig. 1. Block diagram of the segmentation algorithm.

are both measured based on the CDMi, here the previous mask of object regions (OM) is additionally used, where pels are marked which belong to a moving object. In order to get temporally stable object regions, a memory for change detection masks (CDM) is applied, denoted as MEM. The temporal depth of MEM adapts automatically to the sequence. Now, the CDMs is simplified by usage of a morphological closing-operator and elimination of small regions, resulting in the final CDM.

In the fourth step, an initial object mask (OMi) is calculated by eliminating the uncovered background areas from the CDM as in [11]. Therefore, displacement information for pels within the changed regions is used. The displacement is estimated by a hierarchical blockmatcher (HBM) [4]. For a higher accuracy of the calculated displacement vector field (DVF), the CDM from the first step is considered by the HBM. Finally, the boundaries of the OMi are adapted to luminance edges in the corresponding image in order to improve the shape accuracy. The result is the final object mask (OM).

3. Camera motion estimation and compensation

Given two successive frames $I_{(k)}$ and $I_{(k+1)}$ of an image sequence, an apparent camera motion is estimated and compensated. For this purpose, a method is used which estimates the motion parameters of a rigid plane object within a 3D space [11]. The method is based on an affine motion model. Its eight parameters a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8 can reflect any kind of motion, especially zoom and pan. For every pel (X_0, Y_0) in frame $I_{(k)}$ the corresponding pel (X_1, Y_1) in frame $I_{(k+1)}$ is given by

$$X_1 = \frac{a_1 X_0 + a_2 Y_0 + a_3}{a_7 X_0 + a_8 Y_0 + 1}, \tag{1}$$

$$Y_1 = \frac{a_4 X_0 + a_5 Y_0 + a_6}{a_7 X_0 + a_8 Y_0 + 1}. \tag{2}$$

The camera motion is estimated by regression considering only pels within background regions of the previous frame. In case of the first frame or after a scene cut has been detected the background regions are not known. Therefore, pels which distance from the left or right border is less than 10 pels are used as observation points, assuming that there is no moving object near the left and right image border.

After the estimation of the motion parameters, the displacement vector for every pel is known. Then, a postprocessing step counts for model failures in background regions due to the assumption of a rigid plane. By this postprocessing step, the estimated displacement vector of every background pel is improved by perfoming a full search within an area of limited size. Of course, only small model failures can be handled by this postprocessing step. If there are larger deviations from the model of a rigid plane background, the estimation fails. It is planned for future work to model the background by more than one rigid plane, and to use a more robust optimization method for the parameter estimation.

For camera motion compensation the bilinear interpolating function is used. The camera motion is only compensated if a moving camera has been detected. This is the case, if the absolute value of one of the estimated motion parameters $(a_1 - 1)$, a_2, a_3, a_4, $(a_5 - 1)$, a_6, a_7, a_8 is greater than 2.5.

4. Scene cut detection

In order to allow the segmentation of sequences with a strong camera pan or a scene cut, a scene cut detector evaluates whether or not the difference between the current original frame $I_{(k+1)}$ and the camera motion compensated previous frame $I_{\mathrm{CMC}(k+1)}$ exceeds a given threshold t_{SC}. In detail, the mean-square error of the frame difference is calculated in the background regions of the previous frame. A scene cut or strong camera pan is detected by applying the following rule:

$$\left\{\frac{1}{N_{\mathrm{BG}}} \sum_{\{p|\mathrm{OM}_{(k)}(p)=0\}} (I_{(k+1)}(p) - I_{\mathrm{CMC}(k+1)}(p))^2\right\} \begin{cases} > t_{\mathrm{SC}} \Rightarrow \text{scene cut}, \\ \leqslant t_{\mathrm{SC}} \Rightarrow \text{no scene cut}. \end{cases} \tag{3}$$

In Eq. (3), N_{BG} denotes the number of pels set to '0' in the previous OM (background pels). If a scene cut or strong camera pan is detected, the segmentation algorithm is reset, i.e. all parameters are set to their initial value.

5. Estimation of the change detection mask

The algorithm for the estimation of the CDM can be subdivided into several steps as shown in Fig. 2. In the following these steps will be described.

5.1. Computation of the initial CDM

Given two successive frames $I_{(k)}$ and $I_{(k+1)}$ of an image sequence, first the initial CDM for the camera motion compensated first frame $I_{\mathrm{CMC}(k+1)}$ and the original second frame $I_{(k+1)}$ is computed by a threshold operation on the squared luminance difference image, as described in [1]. The threshold is calculated by performing a significance test. Therefore, the luminance difference image D is modelled as Gaussian camera noise with a variance σ^2 equal to twice the variance of the camera noise. This means that a luminance difference d_i for pel i is assumed to correspond to camera noise (null hypothesis H_0)

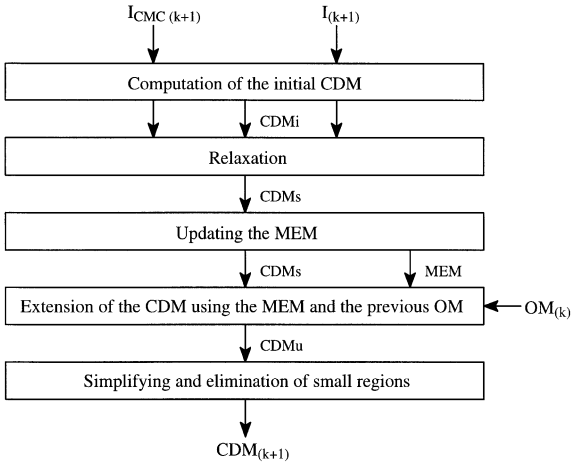

Fig. 2. Block diagram of the algorithm for calculating the CDM.

and not to moving objects (hypothesis H_1) with the following probability:

$$p(d_i \mid H_0) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{d_i^2}{2\sigma^2} \right\}. \tag{4}$$

Assuming that neighbouring pels are statistically independent, the sum of pels in the squared difference image within a window of size $(n \times n)$, normalized by twice the variance of camera noise,

$$T = \frac{1}{\sigma^2} \sum_{i=1}^{n \cdot n} d_i^2 \tag{5}$$

is X^2 distributed with n^2 degrees of freedom.

All pels for which the sum T exceeds a certain threshold t_α are marked as changed, the others are marked as unchanged. The threshold t_α is chosen in the following way: assuming the pel has not changed caused by moving objects, the probability that a calculated sum T is greater than this threshold, equals to a given value α (Eq. (6)) (see Fig. 3).

$$\alpha = p(T > t_\alpha \mid H_0). \tag{6}$$

5.2. Relaxation of the initial CDM for spatial homogeneity

Because real objects have smooth contours, spatial homogeneity of the CDM is required. Spatial homogeneity is achieved by relaxation of the CDMi using a method proposed in [1]. There, the CDMi is processed iteratively. In each iteration step, a decision is made for every border pel in the CDMi, if it belongs to the changed or to the unchanged area. Thus, for every border pel of changed image areas a local threshold is calculated taking into account the local neighbourhood of the pel. For the decision, if a pel has changed by moving objects or not a maximum a posteriori detector (MAP detector) is used:

$$\frac{P(Q_c \mid D)}{P(Q_u \mid D)} \underset{u}{\overset{c}{\gtrless}} 1 \quad \Leftrightarrow \quad \frac{p(D \mid Q_c)}{p(D \mid Q_u)} \underset{u}{\overset{c}{\gtrless}} \frac{P(Q_u)}{P(Q_c)}. \tag{7}$$

Here, Q_c means that a pel is marked changed in the CDM, while Q_u means that this pel is marked unchanged in the CDM. D denotes the given difference image between frame $I_{CMC(k+1)}$ and $I_{(k+1)}$. The probability densities in Eq. (7) are modelled as follows.

The probability density for a luminance difference d_i caused by camera noise ($p(D|Q_u)$) is described by the same Gaussian distribution as given in Eq. (4). The corresponding density ($p(D|Q_c)$) of a luminance difference d_i caused by moving objects is modelled as a Gaussian distribution, too, but the variance of this distribution is the variance σ_c^2 of the difference image D measured in foreground regions:

$$P(D|Q_u) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{d_i^2}{2\sigma^2} \right\}, \tag{8}$$

$$P(D|Q_c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left\{ -\frac{d_i^2}{2\sigma_c^2} \right\}. \tag{9}$$

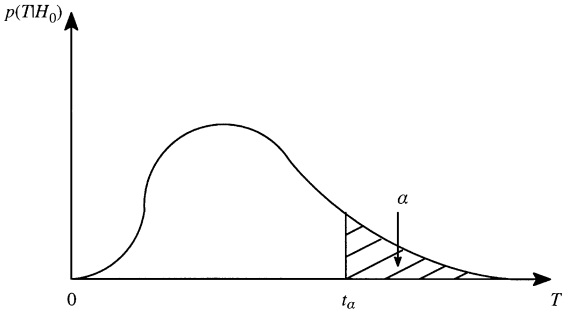

Fig. 3. Significance test for calculating the threshold t_α.

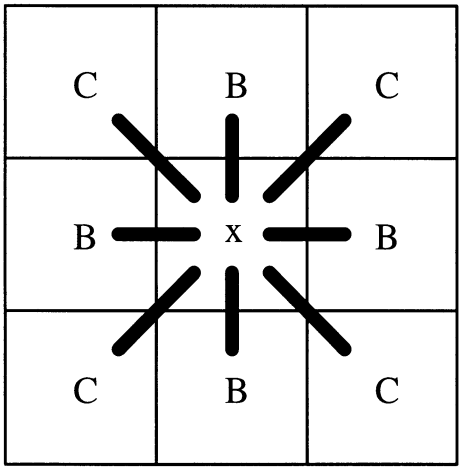

Fig. 4. Neighbourhood of pel x and potentials B and C of the neighbours.

The two a priori probabilities in Eq. (7) are modelled by a Markov random field considering a (3×3)-neighbourhood of a pel as shown in Fig. 4.

$$P(Q_u) = \frac{1}{Z} \exp\{ - E(Q_u)\}, \tag{10}$$

$$P(Q_c) = \frac{1}{Z} \exp\{ - E(Q_c)\}. \tag{11}$$

The constant Z is for normalization, while E denotes an energy term defined as

$$E(Q_{u/c}) = v_B(u/c)B + v_C(u/c)C. \tag{12}$$

The terms $v_B(q_k)$ and $v_C(q_k)$, $q_k \in \{u, c\}$, are a measure for the inhomogeneity of the neighbourhood of pel k, which is separated into horizontal/vertical neighbours (potential B) and diagonal neighbours (potential C), as shown in Fig. 4. So, the term $v_B(q_k)$ denotes the number of horizontal and vertical neighbours of pel k with the opposite label to q_k. In the same way the term $v_C(q_k)$ denotes the number of diagonal neighbours of pel k with the opposite label to q_k.

From Eqs. (7)–(12) the following decision rule can be derived [1], where d_i^2 denotes the squared luminance difference of pel i:

$$d_i^2 \underset{u}{\overset{c}{\gtrless}} 2 \frac{\sigma_c^2 \sigma^2}{\sigma_c^2 - \sigma^2} \times \left(\ln \frac{\sigma_c}{\sigma} + (v_B(c) - v_B(u))B + (v_C(c) - v_C(u))C \right). \tag{13}$$

The label of every border pel in the CDMi is decided by using this decision rule. The rule should read as follows: if d_i^2 exceeds the threshold term on the right-hand side of Eq. (13), the pel is set to changed ($\mathrm{CDMs}_{(k+1)}(i) := 1$), otherwise it is set to unchanged ($\mathrm{CDMs}_{(k+1)}(i) := 0$). The relaxation is processed iteratively, until only a small number of pels are changed by relaxation or the maximal number of iteration steps N is reached. The so obtained CDM is denoted as CDMs.

For calculating the CDM, the variances of the static background σ^2 and within object regions σ_c^2 are needed. While in [1] always the CDMi is used for calculating the variances σ_c^2 and σ^2, it is more accurate to calculate the variance σ^2 within the unchanged regions of the CDM and the variance σ_c^2 within the object regions of the OM. Thus, the variance σ_c^2 does not include uncovered background. Since the positions of static background, uncovered background and moving objects are still unknown for the current frame, in this approach the OM and CDM from the previous frame are used. In order to improve the accuracy of the measured variances, the values of the last three images are averaged.

5.3. Temporal coherency of the object shapes

In order to finally get temporal stable object regions, the CDMs is connected with the previous OM, i.e. in the CDMs additionally all pels are set to changed which belong to the OM of the previous frame. This is based on the assumption that all pels which belonged to the previous OM should belong to the current change detection mask. However, in order to avoid infinite error propagation, a pel belonging to the previous OM is only labelled as changed in the CDMs, if it was also labelled as changed in the CDMs of one of the last L frames,

too. For this purpose, a storage for every pel is applied, building a memory (MEM). The value of this storage indicates, if the respective pel was set to changed in one of the L previous CDMs; the value L denotes the depth of the memory. Considering that CDMs, OM and MEM are two-dimensional fields and that MEM is the zero-matrix for the first frame, the update of MEM can be formulated as

$$\mathrm{MEM}_{(k+1)}(x, y) = \begin{cases} L & \text{if } \mathrm{CDMs}_{(k+1)}(x, y) = 1, \\ \max(0, \mathrm{MEM}_{(k)}(x, y) - 1) & \text{if } \mathrm{CDMs}_{(k+1)}(x, y) = 0. \end{cases} \tag{14}$$

The current CDMs is then updated by logical OR operation between CDMs and the previous OM, taking into account the memory MEM:

$$\mathrm{CDMu}_{(k+1)}(x, y) = \mathrm{CDMs}_{(k+1)}(x, y) \vee \begin{cases} \mathrm{OM}_{(k)}(x, y) & \text{if } \mathrm{MEM}_{(k+1)}(x, y) > 0, \\ 0 & \text{if } \mathrm{MEM}_{(k+1)}(x, y) = 0. \end{cases} \tag{15}$$

It is reasonable to set the depth of the memory L to a large value, if the motion of the moving objects in the scene is small, and vice versa. Thus, L shall be decreased with the average amplitude D of the displacement vectors measured within object regions. Further, it is reasonable to increase the depth of the memory L with the size S of the moving objects. Thus, the value of L is automatically adapted throughout the sequence, using the following heuristic adaptation rule:

$$L = \max\left(1, L_{\max} - \frac{D^4}{c_{L_1}} - c_{L_2}\exp\left(-10^{-3}\frac{S}{c_{L_3}}\right)\right). \tag{16}$$

As can be seen in Eq. (16), the values of L are within the interval $[1, L_{\max}]$. L is decreased by the two terms D^4/c_{L_1} and $c_{L_2}\exp(-10^{-3}S/c_{L_3})$. The first term increases with the average amplitude of the displacement vectors within object regions powered by 4. The second term decreases exponentially with the size of the object regions. While the constants c_{L_1} and c_{L_3} are for normalization to different image formats, c_{L_2} is a weighting factor.

5.4. Simplification of the CDM

In order to simplify the regions in the CDM, the morphological closing-operator is performed. Now, for elimination of small regions a ternary mask is generated, in which pels are labelled as changed, if they are set to '1' in the CDM before and after the closing operation. Pels, which are set to '0' in the CDM after the closing operation, are set to unchanged in the ternary mask, while those pels, which were set to '0' before and which are set to '1' after the closing operation, are set to a third label. Within this ternary mask, small changed and unchanged

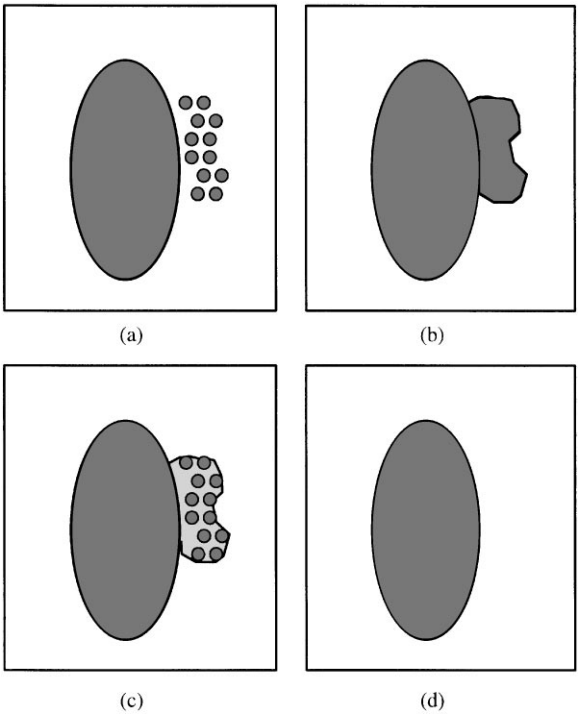

Fig. 5. An example for elimination of small regions by using a ternary mask: (a) CMD with a large region neighboured by smaller regions caused by noise; (b) CMD after elimination of small regions without using a ternary mask; (c) ternary mask; (d) CMD after elimination of small regions using a ternary mask.

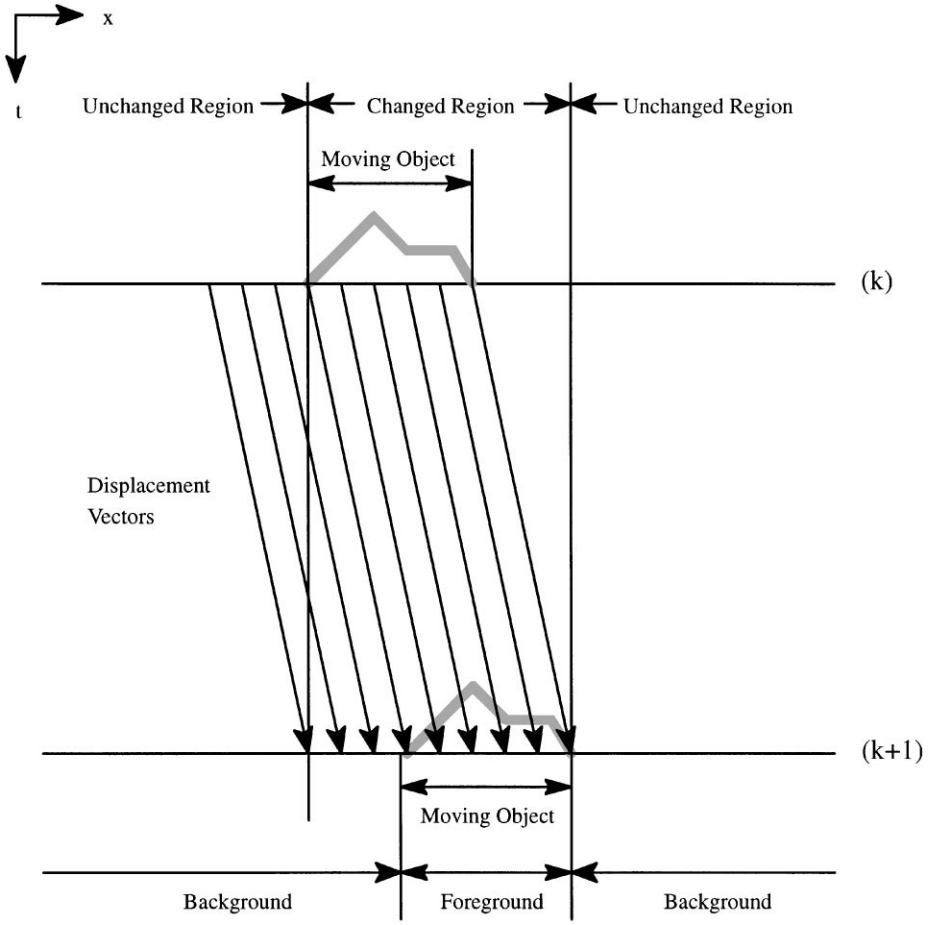

Fig. 6. Example for the separation of changed areas into foreground and background.

regions with a size below a certain threshold are eliminated. In order to prevent that very small regions, which are mostly due to noise spots, grow together with larger regions when the closing is performed, the third label is treated in a special way considering the label of the neighboured regions. An example for this method is given in Fig. 5. There, in Fig. 5(a), a CDM is shown containing a large region, which is neighboured by many small regions caused by noise. Fig. 5(b) presents the CDM after closing is performed. It can be seen, that the noisy areas are grown together with the large region. In order to avoid this, the ternary mask is generated as described above and as it is shown for the example in Fig. 5(c). The CDM after elimination of small regions by using a ternary mask is presented in Fig. 5(d). At this time the final CDM is processed.

6. Estimation of the object mask

For estimating the OM two steps are processed. First, the uncovered background is detected and eliminated from the CDM, resulting in the OMi. Then, the OMi is adapted to luminance edges of the corresponding frame, resulting in the final OM. Both steps are described in the following.

6.1. Detection and elimination of uncovered background

Based on the estimated CDM, the OMi is calculated by detecting and eliminating uncovered background, taking into account the displacement for pels within the calculated CDM. Therefore, a DFV for the changed image areas is generated by hierarchical blockmatching [4]. In order to improve the DVF at object boundaries, the matching criterion is only evaluated for changed regions, leading to a so-called polygon-matching for boundary blocks [21].

Now, pels are set to foreground, if the foot- and the top-point of the corresponding displacement vector are both inside the changed area in the current CDM [11]. If not, these pels are set to background. An example is given in Fig. 6, where an object moves from the left to the right.

6.2. Adaptation of the OM to luminance edges

The OMi is adapted to luminance edges in the current frame, resulting in the final OM. The adaptation must not exceed the following limits: the

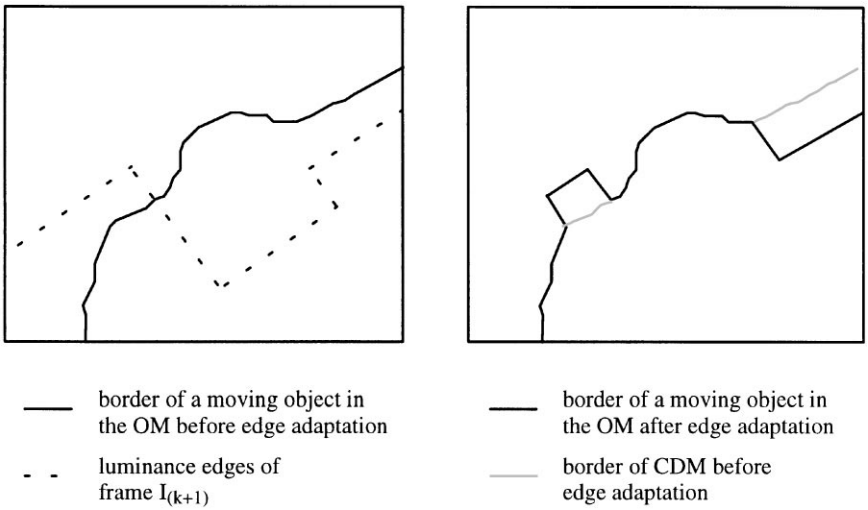

Fig. 7. An exemplary result of the edge adaptation algorithm.

Table 1
Parameter values used for presented simulation results

	Symbol	QCIF	CIF
Threshold for scene cut detection	t_{sc}	250	250
Threshold for estimation of CDMi	t_z	220	165
Potentials used in energy term (Eq. (12))	B	5	5
	C	2.5	2.5
Maximal number of iteration steps	N	20	20
Adaptation radius	R	6	12
Maximum depth of MEM	L_{max}	34	34
Weighting factors used for calculation of memory length	C_{L_1}	1	15
	C_{L_2}	80	80
	C_{L_3}	1	4

outer limit of the area for adaption is given by the contour of the OMi after morphological blowing for two times, the inner limit is set to 6 pel.

Within these limits the luminance edges of the current image $I_{(k+1)}$ are calculated using a Sobel operator. The edge adaptation technique warps every border pel of the OMi to the nearest luminance edge of frame $I_{(k+1)}$, if there exists such an edge within an adaptation radius R around that pel. In Fig. 7, an example for edge adaptation is given: On the left side, a part of the border of a moving object in an OM is shown before edge adaptation is processed. Additionally, the found luminance edges of the current image are shown. On the right side, the corresponding part of the border after edge adaptation is presented.

In some situations it may happen that the OM is adapted to a luminance edge which belongs to the background and not to the foreground. This is the case, if a moving object is separated from the background only by very small luminance gradients, while the luminance gradients in the background are quite large. However, the values of the parameters which determine the area for adaptation are chosen in such a way that the appearance of this problem is neglectible for all investigated test sequences.

7. Experimental results

The proposed algorithm was applied to typical videoconference sequences like mother–daughter

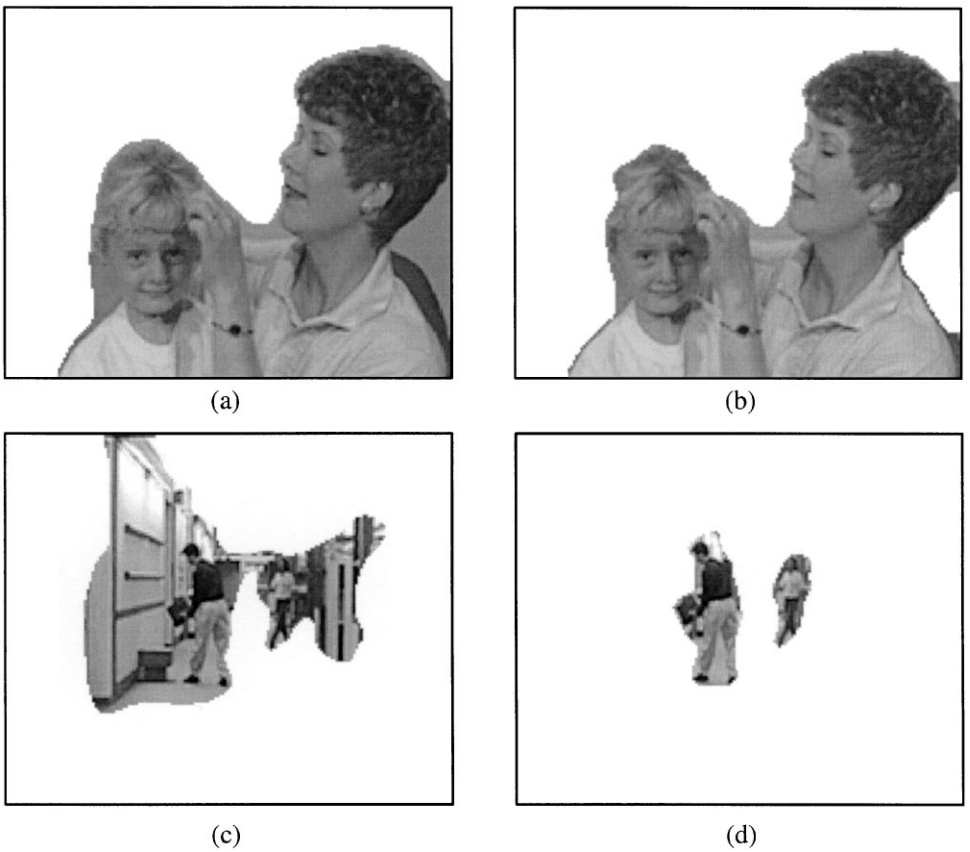

Fig. 8. Results of both segmentation methods for QCIF-sequences at 10 Hz: (a) Mother–daughter (frame 30) segmented by the method described in [11]; (b) Mother–daughter (frame 30) segmented by the proposed method; (c) Hall–monitor (frame 30) segmented by the method decribed in [11]; (d) Hall–monitor (frame 30) segmented by the proposed method.

and akiyo, to sequences containing objects with straightforward motion like hall-monitor and container-ship and to sequences with a moving camera like coastguard and table-tennis. All sequences have a frame rate of 10 Hz and are tested in QCIF and CIF format. For all test sequences of a given image format, the same parameter values have been used, which then are partly adapted automatically to the current sequence. The resulting object masks are subjectively valued, because real masks are unknown.

The reference for the proposed segmentation method is the algorithm proposed in [11]. There, a change detection mask is calculated by using a constant threshold and a non-adaptive memory for change detection masks, i.e. a memory with depth 1. The object mask is generated by usage of a DVF calculated by the hierarchical blockmatcher. As this algorithm assumes a static camera, no segmentation results for the sequences coastguard and table-tennis can be shown as reference. For the presented simulation results, the parameter values shown in Table 1 have been used.

As can be seen from the results in Fig. 8 for QCIF resolution and in Fig. 9 for CIF resolution, the resulting object masks correspond more obviously to real objects than those from the reference algorithm. This is due to the relaxation step, which makes the algorithm more robust against noise, to the evaluation of the ternary mask in the elimination of small objects and to the adaptation to luminance edges. The temporal coherency, which cannot be

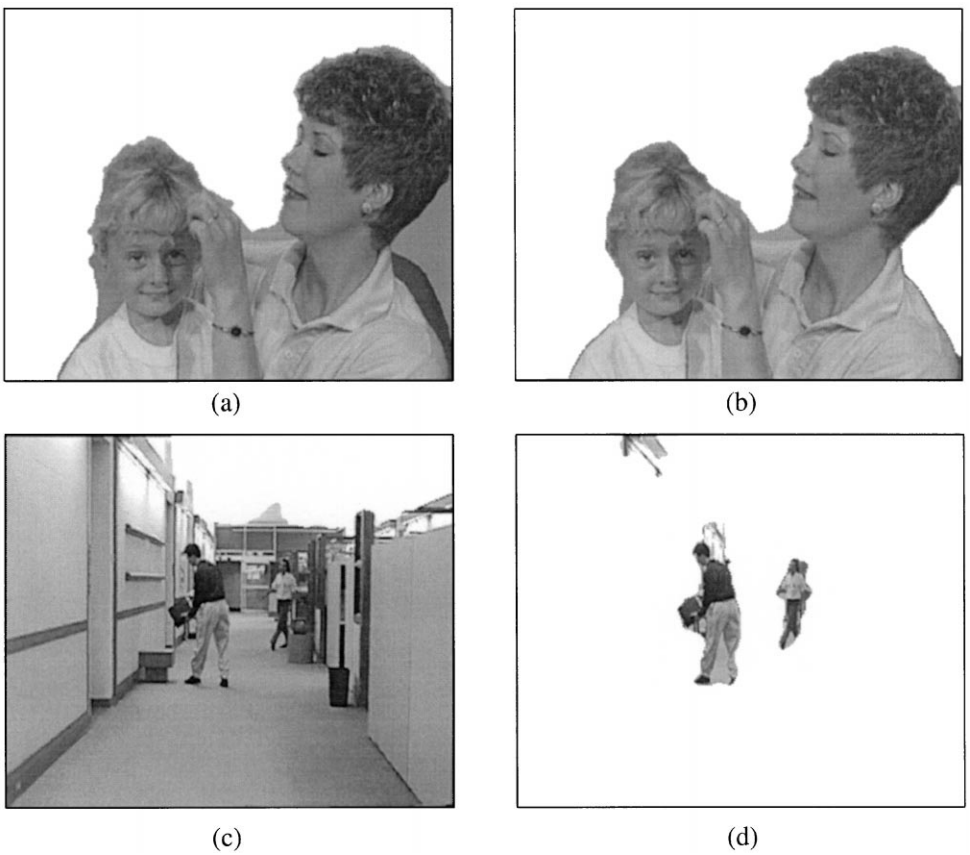

Fig. 9. Results of both segmentation methods for CIF-sequences at 10 Hz: (a) Mother–daughter (frame 30) segmented by the method described in [11]; (b) Mother–daughter (frame 30) segmented by the proposed method; (c) Hall–monitor (frame 30) segmented by the method decribed in [11]; (d) Hall–monitor (frame 30) segmented by the proposed method.

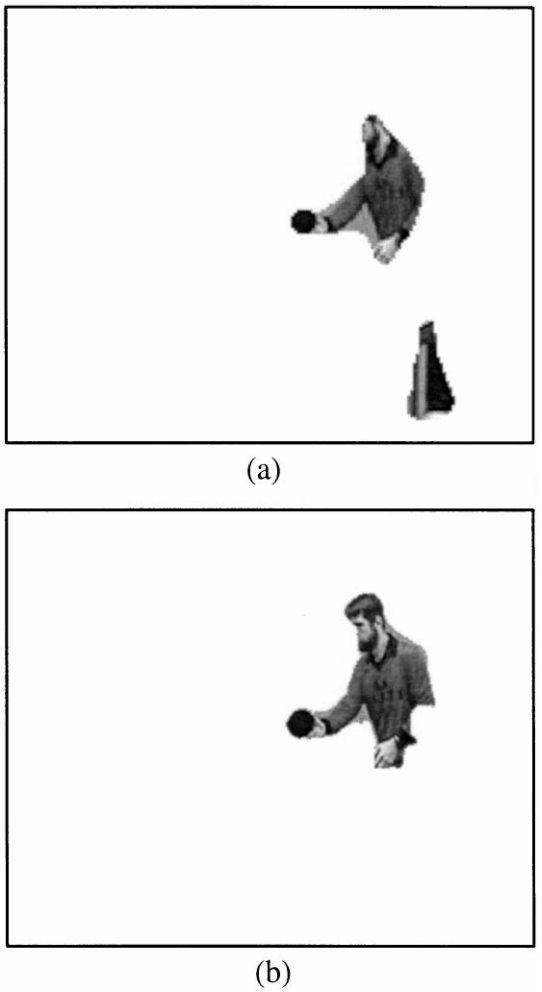

Fig. 10. Results of the proposed segmentation method for frame 30 at 10 Hz: (a) Table-tennis (QCIF format), (b) Table-tennis (CIF format). It is not possible to present results of the reference method due to the moving camera.

demonstrated in a printed paper, is also improved due to the application of an object mask memory. In Fig. 10 the results for a sequence with a moving camera are shown in QCIF and CIF resolution.

Both algorithms, the reference method and the proposed method, are of quite the same computational effort, because the module for hierarchical blockmatching which is used in both methods has by far the largest computational complexity compared to all other used modules. With respect to the execution time, about 5–10 s are required for segmenting one QCIF frame of an image sequence on a Sun Sparc Ultra (200 MHz) workstation. As all used modules are of constant memory effort, the additional payment in terms of memory if using the proposed method instead of the reference method is a constant term.

8. Conclusions

An automatic, noise robust segmentation algorithm for shape estimation of moving objects in video sequences considering a moving camera has been presented. In this algorithm, the 2D shape of moving objects in the captured scene, denoted as object masks, are estimated in four steps. In the first step, a possibly apparent camera motion is estimated and compensated, using an eight-parameter motion model. By the second step, a possibly apparent scene cut is detected. In the third step, a change detection mask is estimated by a local thresholding technique, using a special relaxation algorithm which makes the estimation more robust against noise. In addition, the resulting mask is smoothed by a relaxation. The temporal coherency of the resulting object masks throughout the video sequence is improved by applying a memory for change detection masks. The length of the memory adapts automatically to the amount of motion of the moving objects and to the size of the moving objects in the scene. In the fourth step, an estimated displacement vector field is used to subdivide the change detection mask in parts belonging to the moving objects and parts of uncovered background. Finally, an adaptation of the resulting object mask to luminance edges in the current image is performed in order to improve the accuracy of the estimated object shapes.

Simulations with the proposed algorithm, using different kinds of test sequences, have shown subjectively large improvements compared to the reference technique which only uses a global thresholding technique for change detection without relaxation or luminance edge adaptation. The resulting masks look much better when the proposed algorithm is applied, i.e. they correspond more obviously to the real objects in the scene. Further,

in comparison to the reference technique, the presented technique can also successfully be applied to sequences with a moving camera. Finally, the temporal coherence of the object masks is improved, although still some jittering effects can be observed. This is firstly due to the fact, that the object masks are estimated for each image independently; the applied object memory improves the coherency, but only on a global level and not in detail at the object boundaries. Secondly, the object mask is only estimated with pel accuracy; however, the real boundaries will in general not be exactly on pel position but also in between two pels. Therefore, future work will deal with subpel 2D shape estimation of moving objects in image sequences.

References

[1] T. Aach, A. Kaup, R. Mester, Statistical model-based change detection in moving video, Signal Processing 31 (2) (March 1993) 165–180.

[2] T. Aach, A. Kaup, R. Mester, Change detection in image sequences using Gibbs random fields: a Bayesian approach, Proc. Internat. Workshop on Intelligent Signal Processing and Communication Systems, Sendai, Japan, October 1993, pp. 56–61

[3] J. Benois, L. Wu, Joint contour-based and motion-based image sequences segmentation for TV image coding at low bit-rate, Visual Communications and Image Processing, Chicago, Illinois, September 1994.

[4] M. Bierling, Displacement estimation by hierachical block-matching, 3rd SPIE Symposium on Visual Communications and Image Processing, Cambridge, USA, November 1988, pp. 942–951.

[5] CCITT, Draft revision of recommendation H.261: video codec for audio visual services at $p \times 64$ kbit/s, Study Group XV,WP/1/Q4, Specialist group on coding for visual telephony, Doc. No. 584, November 1989.

[6] I. Corset, Proposal for a structure allowing wide range of core experiments, contribution to the MPEG-4 ad-hoc group on definition of VMs for content-based video representation, 20 December 1995.

[7] COST 211 ter simulation subgroup, SIMOC I, Doc. SIM(94)61, October 1994.

[8] C. Gu, T. Ebrahimi, M. Kunt, Morphological spatio-temporal segmentation for content-based video coding, Internat. Workshop on Coding Techniques for Very Low Bit-rate Video, Tokyo, Japan, November 1995.

[9] P. Gerken, Object-based analysis-synthesis coding of image sequences at very low bit rates, IEEE Trans. Circuits Systems Video Technol. (Special Issue on Very Low Bit Rate Video Coding) 4 (3) (June 1994) 228–235.

[10] M. Hötter, Object-oriented analysis–synthesis coding based on moving two-dimensional objects, Signal Processing Image Communication 2 (4) (December 1990) 409–428.

[11] M. Hötter, R. Thoma, Image segmentation based on object oriented mapping parameter estimation, Signal Processing 15 (3) (October 1988) 315–334.

[12] ITU-T, Video coding for narrow telecommunication channels at < 64 kbit/s, Draft Recommendation H.263, January 1995.

[13] R. Mech, M. Wollborn, A noise robust method for segmentation of moving objects in video sequences, Internat. Conf. on Acoustic, Speech and Signal Processing, Munich, Germany, April 1997.

[14] R. Mech, M. Wollborn, A noise robust method for 2D shape estimation of moving objects in video sequences considering a moving camera, Workshop on Image Analysis for Multimedia Interactive Services, Louvain-la-Neuve, Belgium, June 1997.

[15] MPEG-4 Video Group, Technical Description of the Video Encoder (Proposel for MPEG-4 Tests), Doc. ISO/IEC JTCI/SC29WG11 MPEG95/504, December 1995.

[16] MPEG-4 Video Group, MPEG-4 Autimatic Segmentation of moving objects (core experiment N2) Doc. ISO/IEC JTCi/SC29WG11 MPEG96/841, March 1996.

[17] MPEG-4 Video Group, MPEG-4 Automatic Segmentation of moving objects (core experiment N2), Doc. ISO/IEC JTC1/SC29WG11 MPEG96/989, June 1996.

[18] MPEG-4 Video Group, MPEG-4 Automatic Segmentation of moving objects (core experiment N2), Doc. ISO/IEC JTC1/SC29/WG11 MPEG96/1188, September 1996.

[19] MPEG-4 Video Group, MPEG-4 Automatic Segmentation of moving objects (core experiment N2), Doc. ISO/IEC JTC1/SC29/WG11 MPEG96/1549, November 1996.

[20] MPEG-4 Video Group, MPEG-4 Automatic Segmentation of moving objects (core experiment N2), Doc. ISO/IEC JTC1/SC29/WG11 MPEG97/1831, February 1996.

[21] MPEG-4 Video Group, MPEG-4 Video Verification Model Version 7.0, Doc, ISO/IEC JTC1/SC29WG11 N1642, April 1997.

[22] MPEG-4 Video Group, MPEG-4 Automatic Segmentation of moving objects (core experiment N2), Doc. ISO/IEC JTC1/SC29/WG11 MPEG97/1949, April 1997.

[23] H.G. Musmann, M. Hötter, J. Ostermann, Object-oriented analysis–synthesis coding of moving images, Signal Processing: Image Communication 1 (2) (October 1989) 117–138.

[24] J. Ostermann. Object-oriented analysis-synthesis coding (OOASC) based on the source model of moving flexible 3D objects, IEEE Trans. Image Process. 3 (5) (September 1994).

[25] F. Pedersini, A. Sarti, S. Tubaro, Combined motion and edge analysis for a layer-based representation of image sequences, IEEE Internat. Conf. on Image Processing, Lausanne, Switzerland, September 1996.

[26] F. Pereira, MPEG-4: A new challenge for the representation of audio-visual information, Keynote speech at Picture Coding Symp. PCS'96, Melbourne, March 1996.

[27] C. Stiller, Object-oriented video coding employing dense motion fields, Internat. Conf. on Acoustic Speech and Signal Processing, Adelaide, South Australia, April 1994.
[28] C. Stiller, Object-based estimation of dense motion fields, IEEE Trans. Image Process. 6 (2) (February 1997) 234–250.
[29] N.T. Watsuji, H. Katata, T. Aono, Morphological segmentation with motion based feature extraction, Internat. Workshop on Coding Techniques for Very Low Bit-rate Video, Tokyo, Japan, November 1995.
[30] M. Wollborn, M. Kampmann, R. Mech, Content-based coding of videophone sequences using automatic face detection, Picture Coding Symposium, Berlin, Germany, September 1997.