Object-based Postprocessing of Block Motion Fields For Video Applications

Aishy Amer and Eric Dubois

INRS-Télécommunications

16 Place du Commerce; Verdun, Québec
H3E 1H6 Canada
Email: amer@inrs-telecom.uquebec.ca

ABSTRACT

It is likely that in many applications block-matching techniques for motion estimation will be further used. In this paper, a novel object-based approach for enhancement of motion fields generated by block matching is proposed. Herein, a block matching is first applied in parallel with a fast spatial image segmentation. Then, a rule-based object postprocessing strategy is used where each object is partitioned into sub-objects and each sub-object motion histogram first separately analyzed. The sub-object treatment is, in particular, useful when image segmentation errors occur. Then, using plausibility histogram tests, object motions are segregated into translational or non-translational motion. For non-translational motion, a single motion-vector per sub-object is first assigned. Then motion vectors of the sub-objects are examined according to plausibility criteria and adjusted in order to create smooth motion inside the whole object.

As a result, blocking artifacts are reduced and a more accurate estimation is achieved. Another interesting result is that motion vectors are implicitly assigned to pixels of covered/exposed areas. In the paper, performance comparison of the new approach and block matching methods is given.

Furthermore, a fast unsupervised image segmentation method of reduced complexity aimed at separating objects is proposed. This method is based on a binarization method and morphological edge detection. The binarization combines local and global texture-homogeneity tests based on special homogeneity masks which implicitly take possible edges into account for object separation. The paper contributes also a novel formulation of binary morphological erosion, dilation and binary edge detection. The presented segmentation uses few parameters which are automatically adjusted to the amount of noise in the image and to the local standard deviation.

Keywords: Motion Compensation, Motion Estimation, Block Matching, Object Motion Analysis, Motion Homogenization, Image Segmentation, Binarization, Morphological operations, Edge Detection.

1. INTRODUCTION

In real-time video applications (e.g., consumer-oriented), low implementation cost of video processing methods is of crucial importance.[1] This is why block matching is the most frequently used and hardware implemented motion estimation method.[1] However, because of the difficulties of block matching (e.g., erroneous motion at object boundaries, blocking artifacts, and non homogeneous motion inside objects) various object-based motion estimation methods have been introduced.[2–5] Still block matching is stable, i.e., it never breaks down totally. This cannot be said of object- and model-based methods, that generally give better motion vectors but occasionally fail to interpret object motion correctly. Thus, it is likely that in many applications block-based techniques will be further used.

In this paper, a fast segmentation-based block motion field postprocessing and an image segmentation of reduced complexity are proposed. The goal of the new approaches is to enhance block-based motion fields. Such enhancement increases in turn the quality of motion-compensated video approaches.[6] The remainder of this paper is organized as follows. In Sec. 2, the image segmentation method is presented. Sec. 2.1 provides introductory material and some definitions for the proposed image segmentation. Sec. 2.2 presents the proposed binarization method and simulation results. Sec. 2.3 provides the formulation and performance evaluation of the proposed morphological operations. Sec. 2.4 concludes the proposed image segmentation. In Sec. 3, the integration of the segmentation data to enhance the block motion fields is first presented (Sec. 3.2) and then experimental results of the motion homogenization are shown and discussed (Sec. 3.3). Finally, Sec. 4 concludes the paper and provides an outlook.

2. IMAGE SEGMENTATION BASED ON LOCAL AND GLOBAL CRITERIA

2.1. Introduction

Because of its multiple applications (e.g., coding, retrieval) image segmentation is gaining more importance. However, the demands (e.g, robustness, complexity) are application dependent. While in some applications exact segments (e.g., in upconversion) are needed, approximate segments are satisfactory for the needs of other applications (e.g, retrieval). Another issue is that within a digital coding and storage imaging system, the performance of the system's segmentation technique in the presence of coding artifacts[2] is an important question. Furthermore, the complexity of segmentation methods is a key issue since the wide use of a segmentation tool strongly depends on its computational efficiency.[7]

In this paper, a fast unsupervised image segmentation method of reduced complexity aimed at separating objects is proposed. Within this method the whole segmentation process is divided into simple tasks so that complex operations are avoided. In this paper, *image segmentation* denotes the technique for extraction of image entities or structures (regions or objects) so that the outlines of these structures will coincide as accurately as possible with the physical 2-D object outlines in the recorded 3-D real scene. A *2-D object* is a projection of a 3D real-world object (e.g., a tree) onto the image plane. A *region* (e.g., the leaves of a tree) is a set of image pixels which are similar with respect to a homogeneity criterion such as color or texture. The proposed approach combines local and global spatial homogeneity criteria based on intensity variations. The algorithm is demonstrated on gray-level and texture images. In this paper, two regions R_i and R_j with means μ_i, μ_j and standard deviations σ_i and σ_j are *similar with respect to their luminance variations (texture)* when $|\mu_i - \mu_j| < T_\mu = \psi(\sigma_{noise})$ and $|\sigma_i - \sigma_j| < T_\sigma = \phi(\sigma_{\mathcal{C}})$, where $\mathcal{C}$ denotes a neighborhood and both $\psi(x)$ and $\phi(x)$ are strictly monotonic functions

The proposed image segmentation consists of three basic steps: texture-based separation of object regions (binarization, Sec. 2.2), morphological edge detection (Sec. 2.3), and contour analysis.[8] Contributions in this paper are related to binarization and edge detection.

The binarization provides the morphological edge detector with a binary image in which object regions are separated by one or more black pixels. In this paper, a novel non-thresholding approach for effective, noise adaptive object separation is proposed. This method combines local and global texture-homogeneity tests based on special homogeneity masks which implicitly takes possible edges into account. The method presented uses several parameters which are automatically adjusted to the amount of noise in the image and to the local standard deviation.

Furthermore, novel definition of binary morphological erosion, dilation and binary edge detection is proposed. Their goal is to reduce significantly the computational costs of the binary edge detection. In the paper, a comparative study of the proposed and standard morphological edge detection is given showing that the new operations have significantly lower implementation costs.

2.2. Proposed Texture-based object separation (binarization)

The goal of the binarization is to differently label pixels of the input gray-level image $I(x, y, t)$ which are located in texture-homogeneous regions and pixels at its boundaries. This pixel labeling consists of two steps: (i) detecting the interior of texture-homogeneous regions using local and global detection masks and criteria; (ii) completing pixels at the boundaries which are not labeled in the first step due to the large masks used or due to noise.

2.2.1. Detection of texture-homogeneous regions

Let $I(x, y, t)$ represent the input (gray-level) image at time instant t with X columns and Y rows and $B(x, y, t)$ denote the binary image of the same size. Let $B_c(x, y)$ (or B_c) be the block of size $L \times L$ centered at position (x, y). Let the eight neighboring blocks of B_c corresponding to the eight directions (2× horizontal, 2× vertical, 4× diagonal, Fig. 1) of the same size be $B_0, \cdots, B_7$. Let the distance (in pixels) between the central pixel of block B_c and the centers of the 8 blocks be δ. Note that in this processing step $\delta = L$ (Fig. 1).

A pixel $\mathbf{p} = (x, y)$ of $I(x, y, t)$ is labeled white in $B(x, y, t)$ if B_c is located in a *texture-homogeneous sector* $S_* \in S$, with $S = \{S_{to}, S_{bo}, S_{ri}, S_{le}, S_{ab}, S_{be}, S_{re}\}$ as defined in Table 1 and shown in Fig. 1.

S_* is texture-homogeneous if

$$|\mu_{B_c} - \mu_{B_i}| < T_\mu \quad \wedge \quad |\sigma_{B_c} - \sigma_{B_i}| < T_\sigma, \quad \forall B_i \in S_* \tag{1}$$

	Sector	consists of
S_{to}	"horiz. top"	B_c,B_0,B_1,B_2,B_3,B_4
S_{bo}	"horiz. bottom"	B_c,B_0,B_4,B_5,B_6,B_7
S_{ri}	"vert. right"	B_c,B_0,B_1,B_2,B_6,B_7
S_{le}	"vert. left"	B_c,B_2,B_3,B_4,B_5,B_6
S_{ab}	"diag. above"	B_c,B_1,B_2,B_3,B_4,B_5
S_{be}	"diag. below"	B_c,B_0,B_1,B_5,B_6,B_7
S_{re}	"rect."	B_c,$B_0 - B_7$

Table 1. Sectors in which the texture analysis is performed

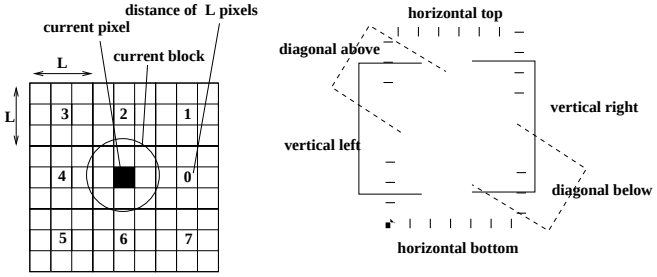


Figure 1. Directions of the texture analysis

where μ_i and σ_i denote the mean and standard deviation of B_i, $T_\mu = \psi(\sigma_{noise})$, $T_\sigma = \phi(\sigma_{S_{re}})$. Both $\psi(x)$ and $\phi(x)$ are strictly monotonic functions.

This detection is done hierarchically, i.e., the labeling starts with a block size $L \times L$ ($L = 2n-1$, $n \in \mathcal{N}, n > 2$) and then the block size is set to $(L-2) \times (L-2)$. This is repeated until the block size is 3. In each step, non-labeled pixels are examined and eventually set to white. As can been seen in Fig. 2(c), in this step, the interior of each region is detected and set to white. Observe that the smallest block is 3, i.e., the smallest surrounding area S_{re} is 9 and that a large $\delta = L$ is taken in this step. This conservative setting, which is necessary to prevent incorrect region merging, does not allow pixels which are located at a region's boundaries to be labeled.

2.2.2. Completing object boundaries

In case of complex intensity variation and in the presence of image artifacts, local homogeneities could be lost. Furthermore, because of the large masks, pixels which are located at a region's boundaries are not labeled. In this step, such pixels are completed using the same strategy as described above but with variable $\delta = 3, 2, 1$. However, before setting pixels, a test is made if this setting would merge two regions. Only pixels which lay at object boundaries or lay inside a binary region are set. Thus a region's separation is preserved but boundaries and small holes are completed.

2.2.3. Adaptation to image changes

Because of noise and other artifacts, homogeneities could be lost, and therefore the thresholds T_μ and T_σ are adapted to the amount of noise present in the image and to the local standard deviation.

First white noise is estimated. Then, T_μ and T_σ are adapted as follows: let a "clean" region R_c be a set of pixels x_i with the mean μ_{R_c} and variance $\sigma^2_{R_c}$. If noise (with estimated variance σ^2_{noise}) is added to that "clean" region and if we assume that there is no correlation between the noise and the image signal, then the new mean and variance of the region are: $\mu_{R_n} = \mu_{R_c} + \mu_{noise}$ and $\sigma^2_{R_n} = \sigma^2_{R_c} + \sigma^2_{noise}$. In this paper, the following adaptation functions are used:

$$T_\mu = \alpha + \beta\sigma^2_{noise}, \qquad T_\sigma = \delta + \gamma\sigma^2_{S_{re}} \tag{2}$$

with $\alpha, \beta, \gamma, \delta \in \mathcal{R}$, $\beta = \psi(\sigma_{noise_{max}})$, $\gamma = \phi(\sigma_{S_{re_{max}}})$, $\sigma_{noise_{max}}$ ($\sigma_{S_{re_{max}}}$) is the maximum assumed noise (local standard deviation), and α and δ are the minimum thresholds.

(a) Original (b) Block of $L = 11$ (c) Block of $L = 11 - 3$ (d) Binarization after completing object boundaries

Figure 2. Hierarchical Binarization

2.2.4. Experimental results

The proposed binarization method was verified by simulation with images undergoing non-uniform illumination changes, images with artifacts. It was shown to be noise-robust. Fig. 2 shows a segmentation of a natural image. Although this image is overlaid with noise (about $30dB$PSNR), the hand and both rings are separated from the background. Also in textured images the segmentation yields comparable results with high-performance image segmentation algorithms.[9] On the other hand, the proposed method needs less computational cost. Note that all the simulation results are obtained with the same parameter setting. No image-wise manual adjustment was undertaken. This robustness is due to the adaptation of the binarization process to the image content using local and global criteria.

2.3. Proposed morphological operations

2.3.1. Preliminaries

Morphological operations are very effective for edge detection of binary images, where white pixels denote uniform regions and black pixels denote region boundaries.[10,11] Usually, the following detectors are used:

$$\begin{aligned} E_{\mathcal{E}}(x,y,t) &= B(x,y,t) - \mathcal{E}[B(x,y,t), K_{(n\times n)}] \\ E_{\mathcal{D}}(x,y,t) &= B(x,y,t) - \mathcal{D}[B(x,y,t), K_{(n\times n)}) \end{aligned} \tag{3}$$

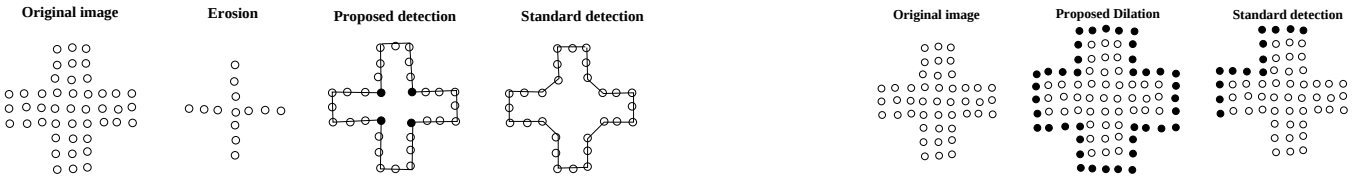
where $B(x,y,t)$ denotes the binary image, $E(x,y,t)$ is the edge image, $\mathcal{E}$ ($\mathcal{D}$) denotes erosion (dilation) operator, $K_{n\times n}$ is the erosion (dilation) $n \times n$ kernel used, and $-$ is the set-theoretical subtraction.

Standard morphological erosion and dilation are defined around an origin, where its position is crucial for the detection. For each step of an erosion (dilation), only one pixel is set (at a time) in $B(x,y,t)$. To achieve precise edges with single-pixel width, 3×3 kernels (defined around the center) are used: when a 3×3 *cross* kernel is used, an incomplete corner detection is obtained (Fig. 3(a)); a 3×3 *square* kernel gives complete edge but requires more computation (which grows rapidly with increased input data, Fig. 5(a)); and the use of a 2×2 kernel will produce incomplete edges (Fig. 3(b)).

To avoid these drawbacks, new operational rules for edge detection by erosion (dilation) are proposed which use a fixed sized (square 2×2) kernel and set all four pixels of the kernel at a time in $B(x,y,t)$. For edge detection based on the novel rules, accurate complete edges are achieved and the computational cost is significantly reduced. Since the wide use of image segmentation strongly depends on its computational efficiency, this is a key point.[7]

2.3.2. Proposed Erosion

In the new erosion rule, *if all four binary-image pixels inside the* 2×2 *kernel are white, then all four pixels in the output image are set (at a time) to white, if they were not eroded in a previous step. If at least one of the four binary-image pixels inside the kernel is black, then all the four pixels in the output image are set (at a time) to black.*



(a) Erosion by a 3×3 cross kernel

(b) Dilation using a 2×2 square kernel, origin at left upper pixel

Figure 3. Proposed versus standard operations

2.3.3. Proposed Dilation

In analogy with the proposed erosion operation, the new dilation definition is as follows: *If at least one of the four binary-image pixels inside the kernel is white, then all the four pixels in the output image are set (at a time) to white.*

2.3.4. Set-theoretical formulation

An advantage of the proposed operations is that they can be formally defined based on set-theoretical intersection, union, and translation in analogy to the formal definitions of the standard operation.[10] The standard erosion and dilation satisfy the following two properties[10]: 1) the erosion of an image by the union of kernels is equivalent to erosion by each kernel independently and then intersecting the result (Eq. 4); 2) the dilation of an image by the union of kernels correspond to dilation by each kernel and then performing union of the resulting images (Eq. 5). Given image A and kernels B and C in $\mathcal{R}^2$,

$$\mathcal{E}_s[A, B \cup C] = \mathcal{E}_s[A, B] \cap \mathcal{E}_s[A, C] \tag{4}$$

$$\mathcal{D}_s[A, B \cup C] = \mathcal{D}_s[A, B] \cup \mathcal{D}_s[A, C] \tag{5}$$

where $\mathcal{E}_s$ ($\mathcal{D}_s$) denotes the standard erosion (dilation). The proposed erosion and dilation are then defined by:

$$\begin{aligned} \mathcal{E}_p[A, K_{2\times2}] &= \mathcal{E}_s[A, S_{3\times3}] = \\ &\mathcal{E}_s[A, K^{ul}_{2\times2} \cup K^{ur}_{2\times2} \cup K^{ll}_{2\times2} \cup K^{lr}_{2\times2}] = \\ &\mathcal{E}_s[A, K^{ul}_{2\times2}] \cap \mathcal{E}_s[A, K^{ur}_{2\times2}] \cap \mathcal{E}_s[A, K^{ll}_{2\times2}] \cap \mathcal{E}_s[A, K^{lr}_{2\times2}] \end{aligned} \tag{6}$$

$$\begin{aligned} \mathcal{D}_p[A, K_{2\times2}] &= \mathcal{D}_s[A, S_{3\times3}] = \\ &\mathcal{D}_s[A, K^{ul}_{2\times2} \cup K^{ur}_{2\times2} \cup K^{ll}_{2\times2} \cup K^{lr}_{2\times2}] = \\ &\mathcal{D}_s[A, K^{ul}_{2\times2}] \cup \mathcal{D}_s[A, K^{ur}_{2\times2}] \cup \mathcal{D}_s[A, K^{ll}_{2\times2}] \cup \mathcal{D}_s[A, K^{lr}_{2\times2}] \end{aligned} \tag{7}$$

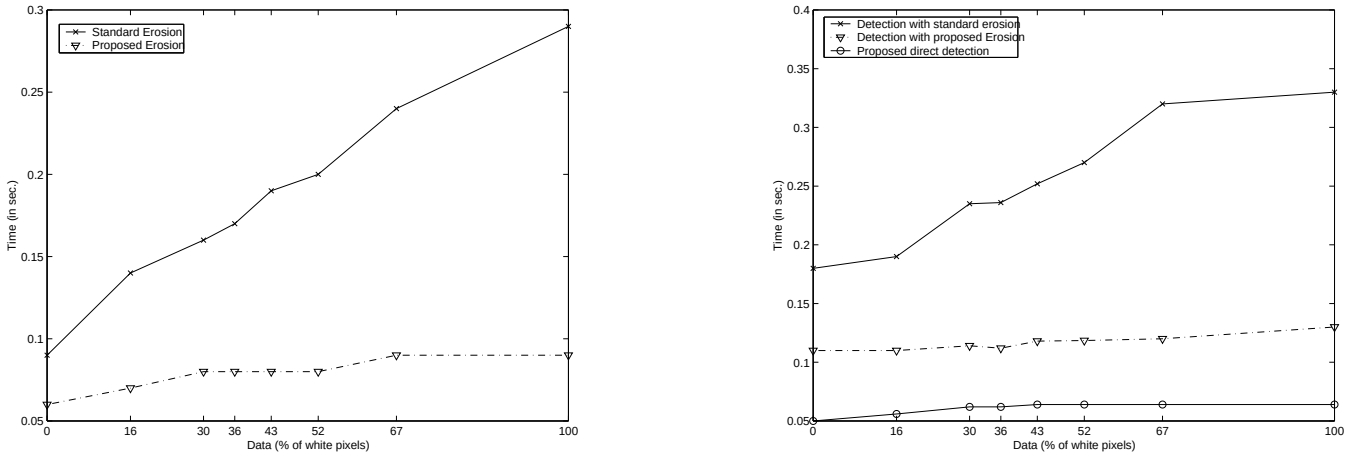
where $\mathcal{E}_p$ ($\mathcal{D}_p$) denotes the proposed erosion (dilation), $S_{3\times3}$ is a 3×3 square kernel, and $K^{ul}_{2\times2}$ is a 2×2 kernel with origin at the upper left (equivalently upper right, lower left, lower right) corner.

2.3.5. Proposed direct binary edge detection

In this section, the need of using two operations (e.g., erosion and set-theoretical subtraction) for a binary morphological edge detection is questioned. When detecting binary edges, erosion and subtraction can be performed implicitly which would further reduce the computational complexity of binary morphological edge detection. Such an implicit detection is proposed as follows: *if at least one of the four binary-image pixels inside the* 2×2 *kernel is black, then set all four pixels of the kernel (at a time) to white if they are white in the binary image.* Thus if the 2×2 kernel fits in a white area, it is implicitly eroded where edge points (kernel does not fit) are kept. Fig. 5(b) gives a complexity comparison of the novel direct edge detection, edge detection using the proposed erosion rule and edge detection using standard erosion with a 3×3 square kernel with origin at the center. As can be seen, the cost of edge detection is significantly reduced.

(a) Original (b) Direct (Sec. 2.3.5) (c) Proposed erosion (d) Standard erosion

Figure 4. Proposed detectors achieve the same performance as standard ones with significantly less complexity (Fig. 5)



(a) Proposed vs. standard erosion (b) Proposed vs. standard detection

Figure 5. Computational efficiency comparison

2.3.6. Experimental results

Fig. 4 shows that the proposed morphological edge detectors achieve the same performance as the standard detectors. On the other hand, Fig. 5(a) gives a software computational cost comparison of the proposed and standard morphological erosion (using *Khoros* software implementation) applied to various natural image sequences. As can be seen, the computational cost using the standard erosion with 3×3 square kernel grows rapidly with the amount of input data, while the cost of the proposed erosion stays almost constant*.

2.4. Conclusion

The proposed edge detectors have been compared to gradient-based (e.g., Canny) and to standard morphological methods showing higher detection accuracy (e.g., gap free edges) and significantly lower computational cost. Another advantage of the new morphological operations is that they can be formally defined based on set-theoretical intersection, union, and translation in analogy to the formal definitions[10] of the standard operation.

The main distinguishing aspects of the proposed binarization process are 1) the separation of texture homogeneous regions, 2) the noise robustness, and 3) regularity and low computational costs. For complex intensity variation,

*Time given in the paper is in seconds measured by the 'clock' command of 'C' on a SUN4 machine with a SPARC 167 Mhz processor for one single standard ITU-R-601 field (720×288)

however, the mean and standard variation alone are not sufficiently discriminatory. In future work, other discrimination factors will be introduced, e.g., the correlation function of the block of pixels. In the proposed method, low-level segmentation criteria are used. Current research is oriented at introducing local and global contextual criteria such as edge continuities.

3. OBJECT-BASED MOTION FIELD POSTPROCESSING

3.1. Preliminaries

Because object-based motion estimation is a circular problem,[3] the motion estimation and image segmentation are treated here as separately as possible. In the proposed object-based motion enhancement, object data (using a fast image segmentation method, Sec. 2) and motion data (using block matching which is suitable for a VLSI realization[1]) are first separately calculated (Fig. 6). Extracted object data are then used to enhance the resulting block motion fields by a regular, non-hierarchical homogenization without raising the total computational costs too high.

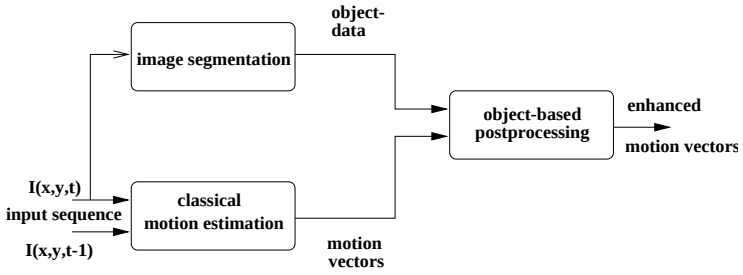


Figure 6. Block diagram of the motion postprocessing

3.2. Integration of object data for motion field enhancement

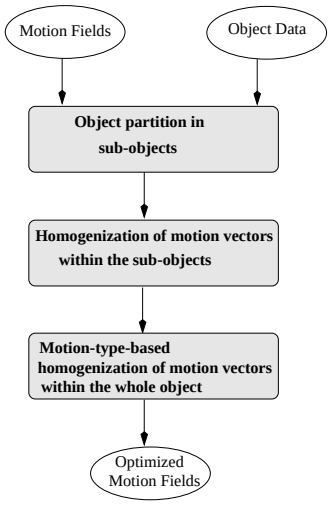


Figure 7. Object-based motion homogenization

After a block matching is first applied in parallel with the above described spatial image segmentation, a rule-based postprocessing strategy (Sec. 3.2.1) is applied where motion fields are homogenized based on the analysis of the motion vectors inside each extracted object (Fig. 7). For this purpose, each object (of area above a given threshold) is partitioned into sub-objects and the sub-object motion histogram is first separately analyzed. Then, using plausibility histogram tests, either a translation or non-translation object motion is detected.

If translation is detected one motion vector is assigned to the whole object. For non-translation motion, one motion vector is first assigned to each sub-object. Then, the sub-object motion vectors are examined according to plausibility criteria and adjusted in order to create smooth motion inside the whole object. Doing this, discontinuities between sub-objects are avoided and non-translation motion is preserved after the homogenization process. The sub-object treatment is, in particular, useful when segmentation errors occur and erroneous large objects are extracted for example. This allows the motion postprocessing to account for object segmentation errors. Furthermore, this sub-object treatment facilitates the detection and protection of non-translational motion within an object.

In order to further take into account segmentation error, possible unwanted object merging is detected using a simple strategy where a gray-level histogram of each object is first determined. Then the maxima of the histogram are examined. If the difference of the first two gray-level maxima is above a given threshold then an unwanted object merging is detected. If unwanted object merging is detected then the two objects are separated and the motion vector enhancement is done separately for each of the objects.

3.2.1. Rule-based vector optimization

For each sub-object (S), the motion vector histogram is first analyzed and the majority motion vector ($\mathbf{w}_S = (u_S, v_S)$ with u_S (v_S) the horizontal (vertical) component of the sub-object motion measured in pixels per seconds) is fixed. Then the motion of each sub-object is determined according to the following rules:

- **Rule 1**: if the difference of the area (in pixels) of the zero motion vector (zero-motion) and the area of all other motion vectors is below a given threshold, then all zero-motion vectors are replaced by the detected majority motion vector of the sub-object ($\mathbf{w}_S$).
- **Rule 2**: If the difference of the area of a minority non-zero-motion and the area of the majority motion vector is below a given threshold, then the minority non-zero-motion vector is replaced by the majority motion vector.

These rules aim to create unique motion within the sub-objects. All zero-motion and minority motion vectors are replaced by the majority motion vector.

After the motion inside each sub-object is homogenized, the distribution of the motion vectors within the whole object is examined and optimization is done according to the following rules:

- **Rule 3**: if the motion of the sub-objects within the object is continuous, then no further steps are needed (e.g., $\{u_{S_1} = 8, u_{S_2} = 16, u_{S_3} = 24, u_{S_4} = 32\}$).
- **Rule 4**: if Rule 2 does not apply and if the sub-objects have different motion vectors so that a majority motion vector (of the four motion vectors) can be fixed, then this majority motion vector is assigned to the whole object (e.g., if $\{u_{S_1} = 8, u_{S_2} = 8, u_{S_3} = 24, u_{S_4} = 8\}$, then the horizontal motion of the object is set to 8).
- **Rule 5**: if Rule 2 does not apply and if the two central sub-objects have continuous motion and their two neighbors have zero-motion, the motion of the neighboring sub-objects is replaced by the non-zero motion of the central sub-objects (e.g., $\{u_{S_1} = 0, u_{S_2} = 16, u_{S_3} = 24, u_{S_4} = 0\} \Rightarrow \{u_{S_1} = 16, u_{S_2} = 16, u_{S_3} = 24, u_{S_4} = 24\}$).

In the procedure presented, the two motion components (horizontal and vertical) are independently post-processed. An important question is whether through this procedure new motion vectors are created or the vector replacement uses original vectors and does not create new component combinations. Various tests have shown that the separate treatment of the two components of the motion vector fields uses original vectors in 95% of the cases. Further simulations have shown that rule 4 is used often. Another interesting observation is that about $25 - 40\%$ of the original vectors are replaced.

3.3. Experimental Results

In this paper, motion compensation techniques are used in order to evaluate the performance of the estimation algorithm. Motion compensation is a non-linear prediction technique where the current image $I(x, y, t)$ is predicted from $I(x, y, \text{t-1})$ using the motion estimated between these images.[6] Object predictions using block matching and object matching are compared. Fig. 8 shows predicted objects of a natural sequence. Note the prediction enhancement inside the objects and at the boundaries (Fig. 8(b), 8(d),8(f)). As a result, blocking artifacts are reduced (Fig. 8(a)) and a more accurate estimation (e.g., moving textured regions (Fig. 8(e)) are not disrupted as when using non homogeneous block motion) is achieved even from noisy block motion fields. Another interesting result is that because of the object-based motion vector replacement, motion vectors are implicitly assigned to pixels of covered/exposed areas.

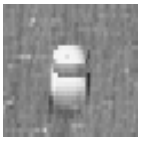
Sequence	PSNR gain (dB)
"Stefan"	1.16
"Ping Pong"	0.5195
"Prlcar"	0.501

Table 2. PSNR gain when using the proposed method

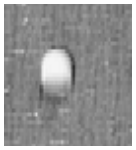
In addition, Table 2 reveals that using the proposed method a PSNR gain (averaged over 60 images for each sequence) of $0.5 - 1dB$ can be achieved. Further simulation results show that the object-based motion postprocessing procedure (including the proposed image segmentation) needs on average about 7 seconds which is about 1/29 of the computational cost of the block-matching method used.

(a) Block-based prediction

(b) Object-based prediction



(c) Block-based prediction



(d) Object-based prediction

(e) Block-based prediction

(f) Object-based prediction (test sequence with pan, translation and rotation). The enhancement is particularly noticeable in the window areas and bars

Figure 8. Comparison of prediction of objects in natural sequences

4. CONCLUSION AND OUTLOOK

The proposed object-based motion estimation approach consists of two main strategies: a noise-robust and regular image segmentation that generates visually relevant disjoint objects; a fast and regular object-based method that homogenizes motion fields and takes account of both translational and non-translational motion and image segmentation errors.

In this paper, performance comparison (PSNR) of the new approach when using optimized parallel-predictive block matching which is suitable for a VLSI realization[1] is given. The computational cost of the object-based post-processing is minor compared to the computational cost of the block matching methods used. Furthermore, the regular structure of image segmentation (which can be effectively implemented) and simulation results showing the performance of the binarization method in noisy images is also introduced.

It is clear that the performance of the proposed postprocessing depends on the performance of the image segmentation used. Therefore, any advances in image segmentation will positively affect the postprocessing technique. Further motion vector enhancement can be achieved by integrating motion data of object of previous images in the estimation of the current object motion.

ACKNOWLEDGMENTS

The authors thank Michael Busian for helping with the software simulations.

REFERENCES

1. G. de Haan *et al.*, "IC for motion compensated 100 Hz TV with smooth movie motion mode," *IEEE Trans. Consum. Electron.* , pp. 165–174, May 1996.
2. A. Amer and E. Dubois, "Segmentation-based motion estimation for video processing using object-based detection of motion types," in *Proc. SPIE Visual Communications and Image Process.*, (San Jose, CA), Jan. 1999.
3. R. Lancini, M. Ripamonti, P. Vicari, M. Caramma, and S. Tubaro, "A moving object identification algorithm for image sequence interpolation," in *Proc. IEEE Int. Conf. Image Processing*, vol. II, pp. 474–477, (Chicago, IL), Oct. 1998.
4. M. Busian, "Object-based vector field postprocessing for enhanced noise reduction," Tech. Rep. S04–97, Dept. Elect. Eng., Univ. Dortmund, 1997. In German.
5. H. Blume and A. Amer, "Parallel predictive motion estimation using object segmentation methods," in *Proc. European Workshop and Exhibition on Image Format Conversion and Transcoding*, (Berlin, Germany), Mar. 1995.
6. E. Dubois and J. Konrad, "Estimation of 2-d motion fields from image sequences with application to motion compensated processing," in *Motion Analysis and Image Sequence Processing*, M. Sezan and R. Lagendijk, eds., ch. 3, pp. 53–87, Kluwer Academic Publishers, 1993.
7. L. Garrido, P. Salembier, and D. Garcia, "Extensive operators in partition lattices for image sequence analysis," *Signal Process.* **66**, pp. 157–180, 1998.
8. T. Pavlidis, *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, Maryland, 1982.
9. S. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.* **18**, pp. 884–900, Sept. 1996.
10. C. Giardina and E. Dougherty, *Morphological Methods in Image and Signal Processing*, Prentice Hall, New Jersey, 1988.
11. R. Haralick and L. Shapiro, *Computer and Robot Vision*, Addison-Wesley, Reading, 1992.