IMPROVED VIDEO SEGMENTATION THROUGH ROBUST STATISTICS AND MPEG-7 FEATURES

Patrick Ndjiki-Nya*, Sebastian Gerke*, Thomas Wiegand*†

*Image Communication Group, Image Processing Department
Fraunhofer Institute for Telecommunications - Heinrich-Hertz-Institut, Berlin, Germany
† Image Communication, Faculty of EE and CS, Technical University of Berlin, Germany
E-Mail: {patrick.ndjiki-nya|sebastian.gerke|thomas.wiegand}@hhi.fraunhofer.de

ABSTRACT

Video segmentation is an important task for a wide range of applications like content-based video coding or video retrieval. In this paper, a new spatio-temporal video segmentation framework is presented. It is based upon robust statistics, namely an M-estimator, and incorporates an MPEG-7 descriptor for consistent temporal labeling of identified textures. The algorithm is based on assumptions about the geometric modifications a given moving region undergoes with time as well as on its surface properties. Homogeneously moving segments are described using a parametric motion scheme. The latter is used to piecewise fit the optical flow field in order to extract rigid motion areas. Robust statistics are used to carefully constrain split, merge and contour refinement decisions. Experimental results show that regions detected by the proposed method are more reliable than the state-of-the-art. True region boundaries are moreover better detected.

Index Terms— *Image segmentation, Motion analysis, Image sequence analysis, Texture analysis, M-estimation*

1. INTRODUCTION

Video analysis typically requires segmentation of the signal into uniform regions. Video segmentation is both critical and essential, as its accuracy has a significant impact on the quality of the final analysis result.

Some spatio-temporal approaches have been proposed in the literature. The better ones typically combine spatial and temporal inferences. Spatial information is often used to constrain the temporal segmentation results [1],[2]. The latter are typically based on short-term motion estimation, where motion similarity is evaluated by an adequate norm either in the motion parameter space or in the spatial domain. Some approaches conduct spatio-temporal video segmentation in the transform domain. Zhu et al. [3], for instance, use DCT-based features to exploit spatio-temporal correlations in video sequences.

In this paper, a model-based video segmentation method is proposed. The underlying model relies on assumptions about the geometric modifications a given moving region undergoes with time as well as on its surface properties.

Homogeneously moving segments are described using a parametric motion scheme. The latter is used to piecewise fit the optical flow field for rigid motion extraction. Robust statistics are used to carefully constrain split, merge and contour refinement decisions.

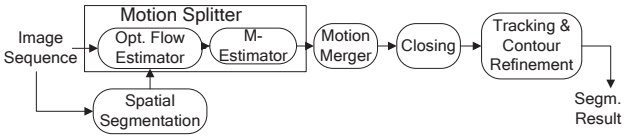

Fig. 1. Block diagram of the proposed spatio-temporal segmentation algorithm

2. OVERALL FRAMEWORK

A new spatio-temporal, parametric segmentation algorithm is presented in this work. It is based on robust statistics, namely an M-Estimator and uses an MPEG-7 descriptor for temporally consistent labeling. It is inspired by the work of Adiv [4] that has influenced the formulation of various video segmentation algorithms (e.g. [2]). The principle of the algorithm presented in this paper is depicted in Fig. 1. As can be seen, the proposed approach corresponds to a split and merge segmentation strategy with tracking abilities. That is, at a given picture transition, the optical flow field is split into homogeneously moving regions using robust statistics, namely a maximum-likelihood estimator called M-estimator. The optical flow and subsequent M-estimation can be initialized with segmentation masks delivered by a spatial segmentation module to improve their performance. Each spatial region is then handled individually.

The typically over-segmented masks obtained after the splitting step are further processed by the motion merger module, which aims to convey all regions featuring similar motion properties to the same class.

After the merging step, a morphological closing operation is applied to remove small clusters located within a much larger homogeneous texture of a different label. Finally, temporal tracking as well as contour refinement of the detected regions are performed. The output of the proposed segmentation algorithm is a mask sequence showing the location of homogeneously displaced spatio-temporal segments.

Note that, for spatial segmentation, the algorithm by Spann and Wilson [5] is used in this work as it has been shown to be very effective. For optical flow estimation, the algorithm

by Black and Anandan [6] is used due to its robustness to multiple motions, transparency, occlusions and specular reflections. The remainder modules will be explained into detail in the following.

3. M-ESTIMATION

3.1 Principle

For each spatial region, the M-estimator creates a piecewise parametric model of a motion vector field obtained from an optical flow estimator. M-estimation eliminates potential outliers a posteriori, i.e. without a prior knowledge of the data. In the proposed spatio-temporal texture analyzer, the data to model correspond to a motion field determined by the optical flow estimator at a given image transition. Outlier motion vectors are identified and their bias on the outcome reduced through the M-estimator in the course of a global motion estimation process.

3.2 Formalization

Within the M-estimation process described in this paper, the perspective motion model [7] defined as

$$v_x = \frac{a_1 + a_3 x + a_4 y}{1 + a_7 x + a_8 y} - x$$
$$v_y = \frac{a_2 + a_5 x + a_6 y}{1 + a_7 x + a_8 y} - y \quad (1)$$

is used, where $a_{1,\ldots,8}$ correspond to the model parameters, with $a_{1,2}$ being translation, $a_{3,6}$ scaling, $a_{4,5}$ shearing, and $a_{7,8}$ perspective motion parameters. $\mathrm{v} = (\mathrm{v_x}, \mathrm{v_y})$ corresponds to a spatial motion vector.

The motion field obtained by optical flow estimation is first modeled using the estimated motion parameters as

$$\boldsymbol{\omega}^{(n)}(\tau) = \begin{pmatrix} \omega_x^{(n)}(\tau) \\ \omega_y^{(n)}(\tau) \end{pmatrix} = \begin{pmatrix} \frac{(a_1(\tau) + a_3(\tau)x^{(n)} + a_4(\tau)y^{(n)}}{1 + a_7(\tau)x^{(n)} + a_8(\tau)y^{(n)}} - x^{(n)} \\ \frac{a_2(\tau) + a_5(\tau)x^{(n)} + a_6(\tau)y^{(n)}}{1 + a_7(\tau)x^{(n)} + a_8(\tau)y^{(n)}} - y^{(n)} \end{pmatrix} \quad (2)$$

where $\boldsymbol{\omega}^{(n)}(\tau)$ correspond to the predicted motion vectors at iteration step τ, while $\mathbf{p}(\tau)$ is the motion parameter set $(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8)^{\mathrm{T}}$ at the same iteration step and $(x^{(n)}, y^{(n)})$ correspond to the n^{th} sample location under consideration. In a subsequent step, the deviations between estimated motion vectors $\boldsymbol{\omega}^{(n)}(\tau)$ and reference motion vectors $\boldsymbol{v}^{(n)}$, determined through optical flow estimation, are evaluated at each pixel location as

$$e^{(n)}(\tau) = \left| v_x^{(n)} - \omega_x^{(n)}(\tau) \right| + \left| v_y^{(n)} - \omega_y^{(n)}(\tau) \right| \quad (3)$$

Each motion vector's weights can now be determined as

$$w^{(n)}(\tau) = \begin{cases} 1 - \left(\left(\frac{e^{(n)}(\tau)}{q\mu_e(\tau)} \right)^2 \right)^2, & \left| e^{(n)}(\tau) \right| < q\mu_e(\tau) \\ 0, & \left| e^{(n)}(\tau) \right| \geq q\mu_e(\tau) \end{cases} \quad (4)$$

where

$$\mu_e(\tau) = \frac{1}{N} \sum_{n=1}^{N} e^{(n)}(\tau) \quad (5)$$

is the mean error, N is the number of samples and q is a degree of freedom of the M-estimator that steers the outlier threshold. The larger q, the more sensitive it becomes to outliers. On the other hand, the smaller ,q the more conservative does the system become with regard to outliers. A robust formulation of the unknown $\mathbf{p}(\tau)$ can now be given as

$$\mathbf{p}(\tau) = (\mathbf{D}^{\mathrm{T}}\mathbf{W}(\tau)\mathbf{D})^{-1}\mathbf{D}^{\mathrm{T}}\mathbf{W}(\tau)\mathbf{k} \quad (6)$$

Outliers are assigned low weights $\mathbf{W}(\tau)$ by this approach. Hence, their influence on the global motion estimation process is reduced in the next iteration step. The weight matrix can be written as

$$\mathbf{W}(\tau) = \begin{pmatrix} w^{(1)}(\tau) & 0 & \cdots & \cdots & 0 \\ 0 & w^{(1)}(\tau) & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & w^{(n)}(\tau) & 0 \\ 0 & \cdots & \cdots & 0 & w^{(n)}(\tau) \end{pmatrix} \quad (7)$$

Each weight appears twice in the weight matrix as, in matrix $\mathbf{D}$, each data influences two rows of the matrix. The weights correspond to real numbers normalized to the interval $[0\ 1]$ and are exactly zero only in case of very crude outliers.

As can be seen above, the motion estimation approach via M-estimation is an iterative method. The estimation is stopped either if the mean prediction error $\mu_e(\tau)$ is smaller than a given threshold or if a maximum number of iterations has been reached.

4. MOTION SPLITTER

The motion splitting module initially operates dense optical flow estimation at image transitions using the algorithm by Black and Anandan [6] (cf. Fig. 1). The M-estimator is then applied to the dense motion field in order to identify homogeneously moving regions. Homogeneity is thereby defined with regard to the perspective motion model (1) as described above. Although a robust M-estimator is used, hints provided by the spatial texture analyzer typically yield an improved segmentation of the motion field. M-estimation is executed recursively by the motion splitter module. The given motion field is first split into inliers and outliers. M-estimation is further applied to the outlier cluster if and only if it is large enough. Outlier clusters that are too small are not further processed.

5. MOTION MERGER

Initialization of the motion splitter module via spatial texture analysis typically yields over-segmentation. Hence, motion merger is required. The latter fuses regions with similar motion properties. Initially, the homogeneous regions are sorted in descending order with regard to their size. The merger of a region pair with respective mean modeling errors μ_e^1 and μ_e^2 is then simulated to give the mean modeling error, $\mu_e^{1,2}$, of the merged regions. Large regions are first compared to and eventually merged with smaller once. Two regions are hereby assumed to feature similar motion properties if $\mu_e^{1,2}$ is not larger than the mean errors μ_e^1 and μ_e^2 of the individual regions. In case the modeling costs increase due to the merger, the considered regions are not fused. Merged regions are assigned the modeling error $\mu_e^{1,2}$ otherwise.

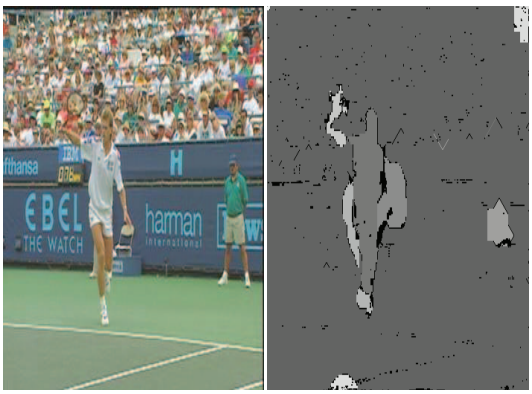

Fig. 2. Keyframe of the "Stefan" test sequence (left), segmentation mask after merging and closing (right)

6. MORPHOLOGICAL CLOSING, TRACKING, AND CONTOUR REFINEMENT

After the motion splitting and merging steps, the segmentation masks typically exhibit a number of "black holes". I.e. scattered small clusters within much larger ones can be observed. The former typically relate to optical flow estimation errors or modeling inaccuracies in the M-estimation process (cf. Secs. 3 and 4). The achievement of larger homogeneous areas is enforced by applying a closing operation on the output of the merger module. Larger "holes", assumingly having a high likelihood to indicate real local motion activity, are thereby kept unchanged, while smaller ones are closed, i.e. assigned the same label as the surrounding texture.

Up to this stage, the segmentation masks have been generated at image transitions. That is, the label assignments are only consistent for contiguous image pairs. In order to extend label consistency to the entire input sequence, a tracking module is required. For that, the textures identified in the course of the video sequence are indexed. Each new texture found in the sequence is matched with the indexed textures. In case it is already known, the corresponding label is assigned to it, the texture is indexed as unknown otherwise and assigned a new label. The Scalable Color descriptor (SCC) defined by MPEG-7 [8] is used for similarity estimation. It is basically a color histogram in the HSV color space. Two textures are considered to be similar if the distance between their feature vectors lies below a given threshold. The Earth Mover's Distance (EMD) [9] is used as similarity measure in order to enable some invariance against luminance and saturation variations that are entailed by effects as shadowing or reflection.

As already said above, the motion analysis conducted up to this stage has been done for image pairs. Hence, no use is made of the prior knowledge related to the segmentation of previous image pairs. The contour refinement module tackles this issue by applying a temporal median filtering algorithm on the output of the tracking module. The usage of this approach for contour refinement is motivated by the fact that it enhances noisy images and preserves edge information. Temporal median filtering is operated on the segmentation masks. A freely settable amount of motion compensated masks preceding the current mask is thereby considered. This operation yields a label update in the current picture with the label of the majority of the past pictures at the specified location. This contributes to stabilize region shapes in the course of the video sequence. On the other hand, at scene changes or in case of fast motion, wrong masks may be generated. For that, in this work, the costs of the masks generated through temporal filtering are compared to those of the masks without temporal filtering. The masks with the lowest costs in terms of modeling inaccuracies are kept.

$$\begin{cases} \mu_e^{tm} < \mu_e^{no\_tm} \Rightarrow \mathcal{R}_{tm} \\ \mu_e^{no\_tm} \leq \mu_e^{tm} \Rightarrow \mathcal{R}_{no\_tm} \end{cases} \quad (8)$$

The decision criterion between temporal mapping (tm) and no temporal mapping (no_tm) is formalized in (8). μ_e^{tm} corresponds to the mean modeling error for a single picture transition and temporal mapping. $\mu_e^{no\_tm}$ is the mean error for a single picture transition and no temporal mapping. The regions obtained before and after temporal mapping are referred to as $\mathcal{R}_{no\_tm}$ and $\mathcal{R}_{tm}$ respectively. An exemplary segmentation result is shown in Fig. 2, where the largest motion is executed by the foreground object, Stefan, and in some areas of the background as indicated by the labels in the mask (cf. Fig. 2, right). The largest homogeneous picture area refers to the background featuring rigid, i.e. no local motion

7. EXPERIMENTAL RESULTS

The benchmark of the spatio-temporal segmentation algorithm presented in this work is conducted by operating a comparative evaluation w.r.t. the COST 211quat Analysis Model (AM) developed by an European research forum with the aim to increase the acceptance of content-based functionalities provided by MPEG-4 and MPEG-7 [1]. For objective evaluation, the quality measures introduced by Huang and Dom [10] are used. They split the automatic

segmentation evaluation into region-based and boundary-based quality assessment. The discrepancy between true and segmented regions or contours is thereby measured. Two region-based measures are defined, i.e. the missing region error rate e_r^m, and the false region error rate e_r^f. Similar measures, e_b^f and e_b^m, are defined for boundary-based quality measurements. Additionally, w_b^f and w_b^m, resp. the false and the missing boundary weights, determine the distance between the misclassified samples and the ground truth boundary.

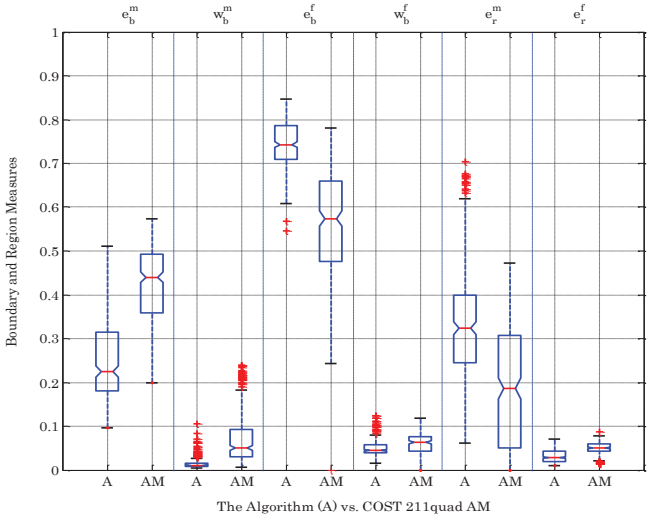

Fig. 3. Boxplots of Huang and Dom's measures for the "Stefan" sequence

Seven test sequences, at CIF resolution (352x288), are considered for the comparative evaluation. Namely "Bus", "Canoe", "Coast Guard", "Container Ship", "Football", "Foreman", and "Stefan". For "Coast Guard", "Foreman", and "Stefan", the segmentation masks provided by MPEG-4 are used as reference. Manual segmentation is done for the remaining test sequences. Each of the texture analysis modules is tuned to achieve the best possible result for all test sequences given a single configuration.

For each of the considered test sequences, it is found that the e_b^m value of the proposed algorithm (referred to as "A" in Fig. 3) is significantly lower than the same error rate for the COST AM. At the same time, our algorithm typically yields significantly lower w_b^m values compared to the AM. This shows that ground truth boundaries can be more accurately found with the proposed algorithm. Furthermore, the missed (true) boundary samples are also very much closer to the algorithm's boundaries compared to the AM. However, our algorithm tends to generate higher e_b^f values than the AM (cp. Fig. 3). This negative outcome is yet attenuated by the fact that the proximity (w_b^f) of the algorithm's false boundary samples to the ground truth boundaries is, in general, significantly higher compared to the AM.

Region estimation evaluation shows that, in general, the false region error rate e_r^f is significantly lower for the algorithm than for the AM (cp. Fig. 3). On the other hand, the missing region error rate e_r^m is typically significantly higher for the algorithm than for the AM. Given the definition of e_r^m and e_r^f, this indicates that the algorithm is more prone to over-segmentation than the AM.

The bottom line of the evaluations is that regions found by our algorithm are more reliable than those found by the AM, i.e. the probability that the automatically segmented regions match the ground truth segmentation masks is higher for the proposed algorithm. The true region boundaries are moreover better detected by our algorithm.

8. CONCLUSIONS

A new spatio-temporal segmentation approach has been presented in this paper. The algorithm is robust due to an incorporated M-estimation process and subsequent constrained segment merger and contour refinement decisions. The proposed method is, however, prone to over-segmentation, which may be explained by a too rigorous merger criterion. Although the evaluations yield an overall favorable outcome for the proposed algorithm, it cannot be ignored that some of the absolute error rates can be seen as relatively high. Hence, long-term motion analysis and a more efficient exploitation of available motion information for tracking via corresponding MPEG-7 descriptors will be considered in our next implementations. Less conservative merger criteria will be evaluated to avoid over-segmentation.

9. REFERENCES

[1] R. Mech, "Description of COST 211 Analysis Model (Version 5.1)", COST 211quat Algorithm Subgroup, 2001.

[2] C. C. Dorea, M. Pardàs, and F. Marqués, "A Motion-based Binary Partition Tree Approach to Video Object Segmentation", *Proc. ICIP*, Vol. 2, pp. 430-433, 2005.

[3] J. Zhu, S. C. Schwartz, and B. Liu, "A Transform Domain Approach to Real-Time Foreground Segmentation in Video Sequences", *Proc. ICASSP*, Vol. 2, pp. 685-688, 2005.

[4] G. Adiv, "Determining Three-dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects", *IEEE TPAMI*, Vol. 7, No. 4, pp. 384-401, 1985.

[5] M. Spann and R. Wilson, "A Quad-Tree Approach to Image Segmentation which Combines Statistical and Spatial Information", *Pattern Recognition*, Vol 18, Nos. 3/4, pp. 257-269, 1985.

[6] M. J. Black and P. Anandan, "The Robust Estimation of Multiple Motions: Parametric and Piecewise-smooth Flow Fields", *Elsevier CVIU*, Vol. 63, No. 1, pp. 75-104, 1996.

[7] J.-R. Ohm, "Multimedia Communication Technology", ISBN 3-540-01249-4, *Springer*, *Berlin Heidelberg New York*, 2004.

[8] B. S. Manjunath, P. Salembier, and T. Sikora, "Introduction to MPEG-7", ISBN 0-471-48678-7, *Wiley*, *Sussex*, *England*, 2003.

[9] Y. Rubner, C. Tomasi, and L. Guibas, "A Metric for Distributions with Applications to Image Databases", *Proc. ICCV*, pp. 207-214, 1998.

[10] Q. Huang and B. Dom, "Quantitative Methods of Evaluating Image Segmentation", *Proc. ICIP*, Vol. 3, pp. 53-56, 1995.