Efficient Maximally Stable Extremal Region (MSER) Tracking

Michael Donoser and Horst Bischof
Institute for Computer Graphics and Vision
Graz University of Technology
{donoser,bischof}@icg.tu-graz.ac.at

Abstract

This paper introduces a tracking method for the well known local MSER (Maximally Stable Extremal Region) detector. The component tree is used as an efficient data structure, which allows the calculation of MSERs in quasi-linear time. It is demonstrated that the tree is able to manage the required data for tracking. We show that by means of MSER tracking the computational time for the detection of single MSERs can be improved by a factor of 4 to 10. Using a weighted feature vector for data association improves the tracking stability. Furthermore, the component tree enables backward tracking which further improves the robustness. The novel MSER tracking algorithm is evaluated on a variety of scenes. In addition, we demonstrate three different applications, tracking of license plates, faces and fibers in paper, showing in all three scenarios improved speed and stability.

1. Introduction

The detection of interest points and local features constitutes the basis for many important computer vision tasks. For example, object recognition, stereo matching, mosaicking, object tracking, indexing and database retrieval, robot navigation etc. rely on the detection of interest points which possess some distinguishing, highly invariant and stable properties. Such structures are often called distinguished regions (DR) [1] and provide a compact and abstract representation of patterns in an image.

Many different interest point detection algorithms were proposed and detailed evaluations and comparisons are available. Evaluations of Mikolajczyk and Schmid [11], as well as Fraundorfer and Bischof [5] revealed that the Maximally Stable Extremal Region (MSER) detector from Matas *et al*. [8] performs best on a wide range of test sequences. MSERs denote a set of distinguished regions, which are defined by an extremal property of its intensity function in the region and on its outer boundary. In addition, MSERs have all the properties required of a stable local detector.

Recent approaches use temporal information for increased stability in interest point detection. For example, Video Google [15] describes an approach to object and scene retrieval based on tracked distinguished regions, where tracking and interest point detection are realized by different algorithms. Obviously, results would be improved if both detection and tracking would be based on the same principles.

Thus, the detection and stable tracking of distinguished regions through an image sequence is of high interest. In this paper we introduce a novel algorithm for the detection and tracking of Maximally Stable Extremal Regions (MSERs). Our contribution is twofold, first, our method improves the computational time of MSER detection and second, via tracking we obtain faster and more stable results compared to single frame based MSER detection.

In section 2 an efficient method for the calculation of MSERs by analysis of the component tree is introduced. Based on this data structure MSERs can be calculated in quasi-linear time.

The main contribution of this paper is a method for tracking of MSERs making use of the component tree, which is presented in section 3. The input to the algorithm is a sequence of images, e. g. video sequence or 3D volumetric data. The novel method uses additional, temporal information to track single connected regions derived from the stability criterion through the entire image sequence.

The result of MSER tracking is a set of distinguished regions for every single image. Additionally, correspondence information between the regions of subsequent images is provided. Especially for longer image sequences MSER tracking outperforms single frame MSER detection. Its main advantages – the speedup of the computation time and the improvement of the detection and tracking stability – are evaluated in detail in section 4.

MSER tracking allows a wide range of different applications. Section 5 presents examples on such diverse tasks as robust tracking of numbers and letters on license plates, face tracking and segmentation of fibers within a 3D data set of a digitized paper sheet.

Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
0-7695-2597-0/06 $20.00 © 2006 IEEE

2. Efficient MSERs by component tree

The concept of Maximally Stable Extremal Regions (MSERs) was proposed by Matas *et al*. [8]. MSERs denote a set of distinguished regions that are detected in a gray scale image. All of these regions are defined by an extremal property of the intensity function in the region and on its outer boundary. MSERs have properties that form their superior performance as stable local detector. The set of MSERs is closed under continuous geometric transformations and is invariant to affine intensity changes. Furthermore MSERs are detected at different scales.

This paper introduces MSER tracking, which requires a data structure that can be efficiently built and managed. The component tree is a structure which allows the detection of MSERs within an image and, in addition, constitutes the basis for MSER tracking. The component tree has been recently used by Couprie *et al*. [3] for efficient implementation of watershed segmentation.

The component tree is a rooted, connected tree and can be built for any image with pixel values coming from a totally ordered set. Each node of the tree represents a connected region within the input image I_{in}. For MSERs we only consider extremal regions R_i which are defined by

$$\forall p \in R_i \ , \forall q \in boundary(R_i) \rightarrow I_{in}(p) \geq I_{in}(q). \quad (1)$$

These extremal regions – the nodes of the component tree – are identified as connected regions within binary threshold images I^g_{bin}, which are the result of the calculation

$$I^g_{bin} = \begin{cases} 1 & I_{in} \geq g \\ 0 & otherwise \end{cases} \quad (2)$$

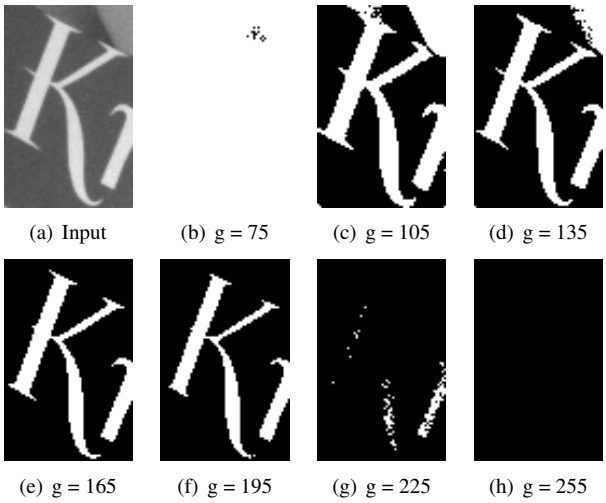
where $g \in [min(I_{in})\, max(I_{in})]$. Figure 1 depicts an exemplary input image and some of the threshold images I^g_{bin} that are analyzed during the creation of the component tree. Each node of the component tree is assigned the corresponding gray value g at which it was determined.

The edges within the tree define an inclusion relationship between the connected regions. Thus, for a region R_i that is the son of a region R_j within the tree,

$$\forall p \in R_i \rightarrow p \in R_j \quad (3)$$

is fulfilled. By moving in the component tree upwards, the corresponding gray value g of the extremal regions becomes lower, which leads to increased region sizes. The root of the tree represents a region which includes all pixels of the input image I_{in}. Figure 2 shows typical parts of the component tree created for the image shown in Figure 1(a).

MSERs are identified by an analysis of the component tree. For each connected region R_i within the tree a stability value Ψ is calculated. This value Ψ is defined as



(a) Input (b) g = 75 (c) g = 105 (d) g = 135

(e) g = 165 (f) g = 195 (g) g = 225 (h) g = 255

Figure 1: Threshold images analyzed during creation of component tree. Figure (a) shows the considered area and figures (b) to (g) the results of thresholding this image at gray level g. The letter k is identified as MSER because the size of the connected region does not change significantly in the gray level range from 135 to 195.

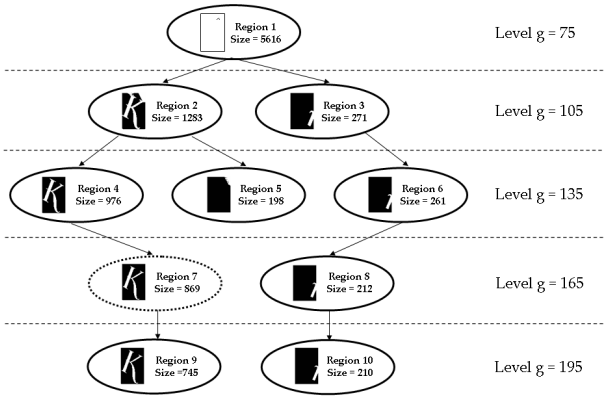


Figure 2: Parts of the created component tree for the image shown in Figure 1(a). Region 7 is identified as MSER.

$$\Psi(R^g_i) \ = (|R^{g-\Delta}_j| \ - |R^{g+\Delta}_k|)/|R^g_i|, \quad (4)$$

where $|.|$ denotes the cardinality, R^g_i is a region which is obtained by thresholding at a gray value g and Δ is a stability range parameter. $R^{g-\Delta}_j$ and $R^{g+\Delta}_k$ are the extremal regions that are obtained by moving upwards respectively downwards in the component tree from region R^g_i until a region with gray value $g - \Delta$ respectively $g + \Delta$ is found. MSERs correspond to those nodes of the tree that have a stability value Ψ, which is a local minimum along the path

(a) Gray scale input image (b) Detected MSERs

Figure 3: Typical result of the MSER detector.

to the root of the tree. Hence, maximally stable regions are those regions which have approximately the same region size across 2Δ neighboring threshold images. For example, in the component tree shown in figure 2, one of the detected MSERs is the region 7, showing the letter k, because its size does not change significantly across 2Δ threshold images, which leads to a local minimum of the stability value Ψ.

The standard MSER algorithm detects bright homogeneous areas with darker boundaries (MSER+). The same algorithm can be applied to the negative of the input image, which results in a detection of dark areas with brighter boundaries (MSER-). In general, the union of both sets is used as MSER detection result. Figure 3 shows a typical result of MSER detection.

Various algorithms have been proposed to compute the component tree, the most efficient one by Najman and Couprie [12] was used for our implementation. The algorithm is based on two subsequent steps. First, the pixels of the input image have to be sorted in decreasing order. This is done by CountingSort [2], which runs in linear time if the range of the gray values is small. Second, a sequence of union-find steps [16] is performed in order to build the connected regions and the component tree iteratively in a bottom-up type manner.

Two key techniques called weighted union rule and path compression [2] ensure that the complexity of the creation process for the component tree is as small as possible, namely $\mathrm{O}(N\alpha(N))$, where $N = n + m$, n is the number of pixels and m is the number of arcs in the image (i. e. approximately $2n$ for the 4-neighborhood). The function α is the inverse Ackermann function, which is a very slowly growing function, that is for all practical purposes below 4. The stability values Ψ can be calculated incrementally during the creation of the component tree, and the detection of the minimum of the stability values can therefore also be done in linear time. Thus, analysis of the component tree enables the detection of MSERs in quasi-linear time, which improves the original implementation that runs in $\mathrm{O}(N \log\log N)$ time – of course for practical image sizes there is not much difference.

In addition, the component tree is a perfect data structure for the extension of the concept to MSER tracking, as is presented in the next section.

3. Concept of MSER tracking

The main contribution of this paper is efficient tracking of MSERs through an image sequence. Including information from the MSER detection result of the preceding image, the computational time and, in addition, the stability of the track can be improved significantly. The improvements are evaluated in detail in section 4. This section describes the main ideas behind the tracking concept.

The algorithm starts with the analysis of the entire image I_t at time t, which results in a detection of MSERs for this image. Then every detected MSER of image I_t is tracked by performing the two following steps on the image I_{t+1}.

First, a region of interest (ROI) of predefined size, centered around the center of mass of the MSER to be tracked, is propagated to the next frame. If a motion model is available it can be incorporated here. Then the component tree for this ROI is built in quasi-linear time by the algorithm presented in section 2. Second, the entire tree is analyzed and the node which best fits to the input MSER is chosen as the tracked extremal region representation. It has to be pointed out, that this step considers all extremal regions, not only maximum ones, which is the reason for the increased stability of the tracking algorithm.

In order to identify the best fit to the input MSER we compare feature vectors that are built for each of the extremal regions of the component tree. The region, which has the smallest weighted Euclidean distance between its feature vector and the one from the input MSER, is chosen as the tracked representation. The features calculated are mean gray value, region size, center of mass, width and height of the bounding box and stability Ψ. The weights for the features can be used to adapt to different kinds of input data. These simple features are sufficient for determination of the correct tracked representation, because the considered extremal regions differ significantly from each other. We have never encountered any problems to identify the correct extremal region based on these features. In addition, also different weight settings do not have a significant effect on the tracking results.

Due to efficiency reasons all features of the extremal regions are computed incrementally during creation of the component tree. Thus, no additional computation time is required. The update takes place each time connected components are united by the union step. The update equations for the features size, gray value, bounding box, center of mass

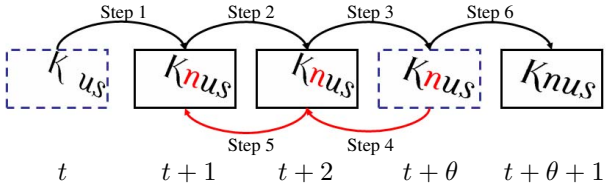


Figure 4: Backward Tracking of MSERs. Entire MSER detection is performed on image I_t and image $I_{t+\theta}$. The letter n is detected in frame $I_{t+\theta}$ and then backward tracked to the frame I_{t+1}.

and stability are straightforward and can be found e. g. in the work of Matas *et al*. [9].

In longer image sequences MSERs may leave the field of view and new MSERs will appear. The previously presented MSER tracking approach just tracks already detected regions. Therefore, new regions cannot be identified. In order to solve this problem a complete detection of MSERs on the entire image is done after a specific number of θ frames. The parameter θ has almost no influence on the tracking stability, because the MSERs that cannot be tracked within the θ images are the least stable ones – usually these also cannot be detected by MSER analysis on the entire image.

In order to further improve the tracking stability, backward tracking is integrated in the concept. Similar to forward tracking, an MSER from image I_t can be tracked back to the image I_{t-1}, which is called backward tracking. After detecting MSERs in the entire image, newly appearing MSERs are identified by a comparison to the tracked MSERs. These identified MSERs in image I_t are then tracked backwards until the frame $I_{t-\theta}$. Figure 4 illustrates this concept. Thereby, no available MSER information is lost. Because θ images are always completely stored in the memory, this step does not increase the computational time.

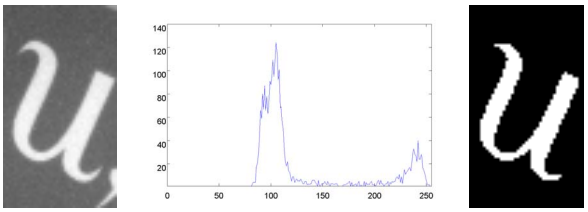
The final result of MSER tracking is a set of connected regions for each image including correspondence information between the regions of subsequent images.

4. Analysis of improvements due to tracking

Using the additional information from the MSER detection result of the preceding image a twofold improvement in the detection of the tracked regions can be achieved. First, the computational time for the detection of a single MSER can be decreased and second, the robustness of the detection and of the tracking results can be improved significantly.

4.1. Speedup of computational time

MSER tracking speeds up single MSER detection due to three main points. First, for each region only either MSER+ or MSER- detection is performed, second, only parts of the component tree are built for the correct detection of single MSERs and third, the search region for tracked MSER representations is constrained.



(a) Input Image (b) Image histogram (c) MSER result

Figure 5: Illustration of speedup by MSER tracking. Constraining the gray scale range can significantly reduce the size of the component tree and therefore the calculation time without having any impact on the result.

The main idea for improving the computational time is that only a part of the component tree has to be built in order to identify the tracked region correctly. The gray value, at which the MSER to be tracked was detected in the preceding image, is used to constrain the range of the gray values that have to be analyzed in the next image. Therefore, connected regions are only calculated within the defined gray value range. The corresponding component tree becomes much smaller and hence, the MSERs can be calculated faster. The gray value range, which has to be considered, is set proportional to the distance between the histograms of the preceding and the current ROI. The comparison is done by calculating the Chi-Square distance [13]

$$\chi^2(g,h) = \frac{1}{2}\sum_{i=1}^{N_b}\frac{(g_i - h_i)^2}{g_i + h_i} \tag{5}$$

between the two histograms g and h, where N_b is the number of histogram bins. This ensures, that abrupt intensity changes between subsequent frames are handled correctly.

As an example, Figure 5 illustrates the improvement that can be achieved by constraining the gray scale range. It shows a region of interest, which is analyzed for tracking of an MSER and the corresponding histogram. In addition, the information that the MSER of the preceding image was detected at gray value 225 is available. If the entire range of gray values is analyzed, hence, the entire component tree has to be built, 22 845 operations are required to detect the MSER. However, exactly the same result can be achieved if the analysis is constrained to the gray value range $[200..255]$. The total number of calculations decreases to 4 506, and hence, in this example, the calculation is done approximately five times faster, just by constraining the gray scale range.

(a) Football Sequence (b) Boxes from different viewpoints (c) Paper structure

Figure 6: Three different data sets that are analyzed for evaluation of the concept.

Since the component tree is built in a bottom-up manner, its creation can be stopped at those branches, where the size of the area is outside a predefined range around the size of the MSER region to be tracked. This results in another speedup of the detection.

Consequently, by constraining the analysis on parts of the component tree, exactly the same MSER detection results can be achieved with a smaller amount of memory and calculation time.

In order to evaluate the speedup statistically, the presented MSER tracking concept was applied to three different data sets: a video sequence from a football match, an image sequence acquired from 20 different viewpoints on two boxes and microscopic cross-section images of a paper sheet. Figure 6 shows the three test data sets and Figure 7 evaluates the experimental speedup that is gained by using the previously described technique of constraining the analysis on parts of the component tree. It compares the number of required calculations for the complete analysis of the component tree to the number of calculations of the constrained analysis. It can be seen that an average speedup of 2 to 4 is achieved. The speedup for the fiber data set is much lower, because the background of those images has approximately the same gray value as the tracked MSER regions. Thus, for such images almost the entire component tree has to be built.

In addition, for the purpose of tracking MSERs it is not necessary to analyze the entire image. The area to search for tracked MSERs can be constrained to a region of interest (ROI) around the center of mass of the preceding MSER representation. The size of the ROI is one parameter of the algorithm, and is set to the expected maximum change in location of MSERs between subsequent images of the input sequence. Thus, by means of constraining the ROI another speedup is possible, which strongly depends on the number of MSERs that are tracked. Figure 8 shows the overall speedup, including all the constraints, that is achieved by MSER tracking. Again the number of required calculations is compared and speedups of up to 10 are achieved. For all data sets approximately 100 regions are tracked.

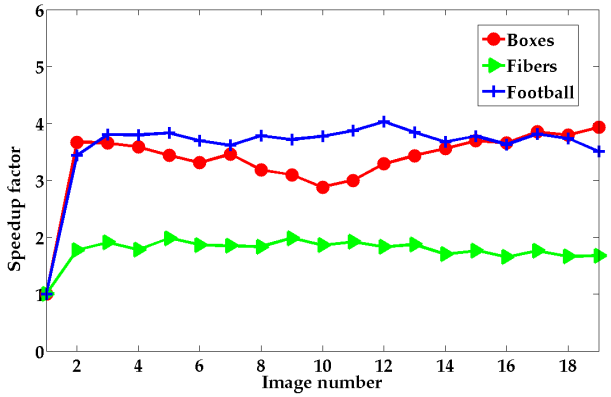


Figure 7: Speedup due to constraining the MSER detection on parts of interest of the component tree. Three different data sets are analyzed over a sequence of 20 images.

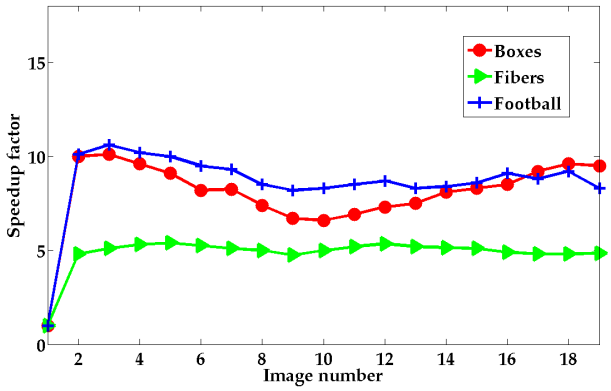


Figure 8: Overall speedup achieved by MSER tracking. Three different data sets are analyzed over a sequence of 20 images. For each approximately 100 MSER regions are tracked.

The overall speedup factor strongly depends on the number of tracked regions. Figure 9 shows the speedup factor versus the number of tracked MSERs.

4.2. Improvement of stability

Another contribution of MSER tracking is that not only maximally stable regions – i. e. the nodes of the component tree with the lowest stability value Ψ – are considered as tracked MSER representations (see section 3). This improves the quality and the stability of the correspondences between the regions. This is the reason, why MSER tracking outperforms all attempts to establish correspondences between the results of single image based MSER detection.

Figure 10 shows the results of single frame MSER detection with constant parameters (a) - (c), in comparison to the results of the MSER tracking concept (d) - (f). It can be

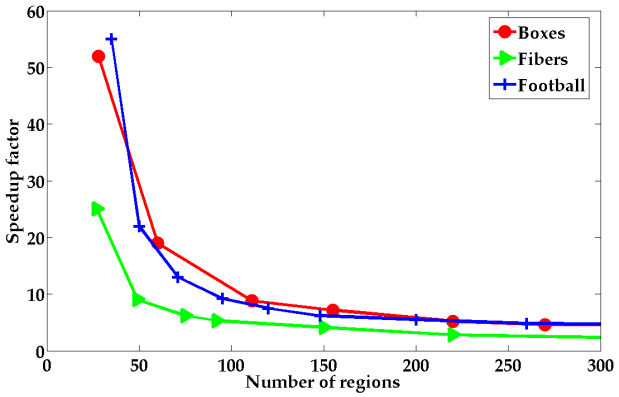


Figure 9: Overall speedup for MSER tracking depends on the number of regions to be tracked.

(a) Single frame MSER – image I_t

(b) Single frame MSER – image I_{t+6}

(c) Single frame MSER – image I_{t+12}

(d) MSER tracking – image I_t

(e) MSER tracking – image I_{t+6}

(f) MSER tracking – image I_{t+12}

Figure 10: Illustration of the stability gain by the MSER tracking approach. With MSER tracking – (d) to (f) – the letters are successfully tracked through the entire sequence, whereas with the single frame approach – (a) to (c) – the letters are identified only partially.

seen that e. g. the *'Knuspermüsli'* text is not robustly identified by the single frame MSER detection, whereas with the MSER tracking approach the text and other MSERs are correctly tracked through the entire sequence.

In order to evaluate the introduced stability improvement an evaluation framework proposed by Fraundorfer and Bischof [5] was used on the data set shown in Figure 6(b), which consists of a sequence of images showing two boxes acquired from varying viewpoints from $0°$ to $90°$.

Fraundorfer and Bischof used a framework similar to

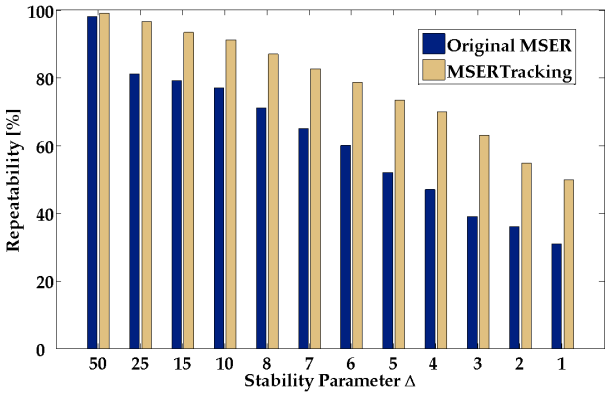


Figure 11: Comparison of the tracking quality based on analysis of repeatability values for different stability range values Δ of MSER detection.

Mikolajczyk and Schmid [11] to evaluate local detectors by a repeatability score. Unlike [11] this framework allows the evaluation of non planar scenes. In order to enable coordinate transfer of the interest points between images acquired from different viewpoints, a ground truth is determined using the trifocal tensor geometry [6]. The ground truth is compared to the experimental results and a repeatability score for the region-to-region correspondence is calculated. The calculated repeatability score can be interpreted as a quality benchmark of the tracking stability.

Figure 11 compares the calculated repeatability scores for the single frame MSER detection to those from the MSER tracking results. For example the MSER tracking of about 100 regions results in an average repeatability score of 89%, whereas the single frame approach leads to a score of only 68%, which demonstrates the improved performance. It can be seen that the quality of the track significantly depends on the stability range parameter Δ of the detected MSERs. The higher the chosen stability value, the more robust the tracking results become, but the less regions are found. We are aware of the fact that this comparison is unfair because MSER tracking has more information available than single MSER detection.

5. Selected Applications

The presented MSER tracking concept opens a wide range of possible applications. In order to demonstrate its versatile applicability it was applied to such diverse tasks as robust tracking of license plates, face tracking and 3D fiber network segmentation.

5.1. Application: Tracking of license plates

The recognition of license plates is a technology used to identify vehicles by analysis of the signs on their license

(a) Frame 1

(b) Frame 15

Figure 12: Illustration of license plate tracking. The images show a traffic scene and the tracked MSERs on the license plate are highlighted in white.

plates and it is used in various access-control systems, security and traffic monitoring applications.

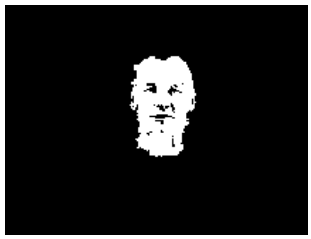
The numbers and letters on a license plate can be detected as MSERs, as e. g. by Matas and Zimmermann [10]. MSER tracking can be used to track them efficiently and robustly through a video sequence. Figure 12 shows two frames of a traffic video sequence, where eight MSERs on the license plate are identified and robustly tracked through the entire sequence.

Assuming that the location of the license plate is a priori given and the MSER search area is constrained to the license plate area, the average speedup due to MSER tracking is 3.6. However, if the location is not known and the entire image has to be processed for single frame MSER detection, the average speedup due to tracking, in this example, is 560.

5.2. Application: Face Tracking

Detection and tracking of human faces has a wealth of possible applications such as in security systems, teleconferencing, human-computer interfaces or gesture recognition. A lot of approaches use color as the key feature for detection and tracking of faces in images. E. g. Raja *et al*. [14] model the skin color distribution using a mixture of Gaussians estimated in the r-g color space. The r-g color space has shown to be well suited for representing skin color over a wide range of lighting conditions [18].

(a) Single frame



(b) Tracked MSER face

Figure 13: Illustration of face tracking application. One single frame of a video sequence and the tracked MSER are shown.

In general, MSERs can be detected in all images whose pixel values are from a totally ordered set. Thus, for face tracking, the pixels of the input color image are ordered by their Mahanalobis distance [4] to the estimated 2D-Gaussian distribution within the r-g color space. This ordering allows the detection of human faces by MSERs (only extremal regions at low gray values have to be considered), and the introduced tracking concept enables the robust track of faces through a video sequence. Figure 13 shows a single frame of a video sequence (a) and the detected and tracked face (b).

In this example, the obtained speedup due to MSER tracking is on average 3.2 if the ROI is constrained based on a preceding face detection. If the entire image has to be analyzed for single frame MSER detection, an average speedup of 13.8 is achieved.

5.3. Application: Segmentation of fiber network

The concept of MSER tracking can also be applied to analysis of 3D data sets. As an example, the segmentation of single fibers within a digitized 3D paper structure was realized. The 3D data consists of a set of microscopic cross-section images, which were acquired by an automated microtomy concept, introduced by Wiltsche *et al*. [17].

Paper can be seen as a complex network of fibers. Fibers within paper are pressed tubes and can be identified as homogeneously colored regions with darker boundary and approximately elliptical shape within the images. Thus, fiber cross sections correspond to MSERs, as can be seen in Figure 6(c).

The borders of the detected fiber cross sections are not smooth. Thus, a method which allows to account for the discrete nature and the noise within the 3D data is required. Therefore, the presented MSER tracking concept was extended for the special case of fiber segmentation. Active contour models were chosen for smoothing. The process is comparable to the one of Lapeer *et al*. [7], where snakes are applied to the result of a watershed segmentation.

Figure 14: Part of a segmented fiber (200 μm length), which was identified by MSER tracking.

Our MSER tracking concept allows the detection and the robust tracking of these fiber cross sections, which facilitates the segmentation of the fiber network of a paper sheet. Figure 14 shows a 3D visualization of a part of a fiber, which was segmented by applying the MSER tracking concept to a digitized 3D paper structure stack at a resolution of $0.26 \times 0.26 \times 5\ \mu m^3$. The obtained speedup in this example is on average 5 (see also Figure 8).

6. Conclusion and Outlook

The main contribution of this paper is a novel concept for tracking of Maximally Stable Extremal Regions (MSERs). The presented approach uses temporal information to improve the computational time and, additionally, the detection stability of single MSERs. Experimental evaluations, which compare the single frame approach to the introduced tracking concept on different data sets, proved the increased performance of MSER tracking. The novel concept opens a wide range of possible applications. Tracking of license plates, face tracking and segmentation of fibers within a 3D data set were demonstrated in this paper.

In future, we plan to extend the MSER detection to the third dimension, which will enable the detection of maximally stable volumes within 3D data sets.

Acknowledgements

This work has been supported by the Austrian Joint Research Project Cognitive Vision under projects S9103-N04 and S9104-N04. Part of this work has been carried out within the K-plus Competence center Advanced Computer Vision funded under the K plus program.

Additionally, the authors gratefully acknowledge financial support from Mondi Business Paper Austria, Mondi Packaging Frantschach, M-Real Hallein, Norske Skog Bruck, Sappi Gratkorn, SCA Graphic Laakirchen, UPM Steyrermühl and Voith Paper.

References

[1] D. Chetverikov and J. Matas. Periodic textures as distinguished regions for wide-baseline stereo correspondence. In *Proc. of the 2nd International Workshop on Texture Analysis and Synthesis*, pages 25–30, 2002.

[2] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge, 1990.

[3] M. Couprie, L. Najman, and G. Bertrand. Quasi-linear algorithms for the topological watershed. *Journal of Mathematical Imaging and Vision*, 22(2-3):231–249, 2005.

[4] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.

[5] F. Fraundorfer and H. Bischof. A novel performance evaluation method of local detectors on non-planar scenes. In *Workshop Proceedings of Conference on Computer Vision and Pattern Recognition*, 2005.

[6] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[7] R. J. Lapeer, A. C. Tan, and R. Aldridge. Active watersheds: Combining 3D watershed segmentation and active contours to extract abdominal organs from MR images. In *MICCAI (1)*, pages 596–603, 2002.

[8] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. of British Machine Vision Conference*, pages 384–396, 2002.

[9] J. Matas and K. Zimmermann. A new class of learnable detectors for categorisation. In *Proc. of the 14th Scandinavian Conference, SCIA*, pages 541–550, 2005.

[10] J. Matas and K. Zimmermann. Unconstrained licence plate detection. In *Proc. of the 8th International IEEE Conference on Intelligent Transportation Systems*, pages 572–577, Heidelberg, Germany, 2005.

[11] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1615–1630, October 2005.

[12] L. Najman and M. Couprie. Quasi-linear algorithm for the component tree. In *SPIE Vision Geometry XII*, volume 5300, pages 98–107, 2004.

[13] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.

[14] Y. Raja, S. J. McKenna, and S. Gong. Colour model selection and adaptation in dynamic scenes. *Proc. of 5th European Conference on Computer Vision*, 1:460–474, 1998.

[15] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. of the International Conference on Computer Vision*, pages 1470–1477, 2003.

[16] R. Tarjan. Efficiency of a good but not linear set union algorithm. *Journal of the ACM*, 22(2):215–225, 1975.

[17] M. Wiltsche, M. Donoser, W. Bauer, and H. Bischof. A new slice-based concept for 3D paper structure analysis applied to spatial coating layer formation. In *Proc. of the 13th Fundamental Paper Research Symposium*, pages 853–899, 2005.

[18] J. Yang, W. Lu, and A. Waibel. Skin-color modeling and adaptation. In *Proc. of ACCV*, pages 687–694, 1998.