

# MSER Algorithm to Characterize an Active Camera Movement

Younis E. Abdalla  
Electrical and Computer Eng.  
Memorial University  
St. John's, NL, Canada

T. Iqbal  
Electrical and Computer Eng.  
Memorial University  
St. John's, NL, Canada

M. Shehata  
Electrical and Computer Eng.  
Memorial University  
St. John's, NL, Canada

**Abstract** – A new method presented in this paper to describe the movement of a camera mounted on a mobile platform using local Maximally Stable Extremal Region detector (MSER) algorithm. In fact, it was established that MSER tracker method is quicker than other methods and it needs less resources. Furthermore, MSER outperforms other interest points detection and matching algorithms. Using Motion Estimation (ME) technique to detect the direction and speed of the camera during a video recording demonstrates the reliability of the algorithm. Results indicate over 93 percent tracking accuracy in variant environment. The paper presents the details of algorithm, implementation setup and results for a set of recorded videos.

**Keywords** – MSER detection, Matching, Camera tracking, Image processing.

## I. INTRODUCTION

The camera movement usually follows the carrier movements. In order to determine the direction and movement characteristics we proposed to use Maximally Stable Extremal Regions (MSER). There are three main reasons to use MSER. First, MSER has high stable connectivity between the components in the same gray level image. Second, it is a blob detector method of image processing. Third, the MSER algorithm does extract the local feature from the image. The MSER technique is to take regions which stay virtually same through a wide range of threshold, then used them as interest regions since their repeatability and efficiency are way better than other detectors [1]. Furthermore, MSERs perform well on small regions, even when the image contains homogeneous regions and smart implementation makes it a fast region detector [2].

## II. RELATED WORK AND BACKGROUND

Many research and studies use features to implement different tasks. For example, local feature detection, feature tracking and feature description. There is a variance algorithms deal with feature to achieve the above tasks. MSER algorithm is one of these algorithms which have been used for different applications of computer vision [3].

MSER is a method of extracting a sufficient amount of corresponding image elements that contributes to the wide-baseline matching [4], and it detects the levels of lines which are local extrema of contrast. Mathematical morphology implemented the shape analysis by correlating

with any binary images obtained by applying a specific threshold to the image at all levels and taking the upper level connected components of these binary images. These upper level sets, give a complete representation for the Image. The gray level image of the upper level set  $u : R^2 \rightarrow R$  at level  $\lambda$  is defined by [5]:

$$\chi_\lambda(u) = \{x \in R^2, u(x) \geq \lambda\}. \quad (1)$$

$$u(x) = \sup \{\lambda \in R, x \in \chi_\lambda(u)\}. \quad (2)$$

The MSERs can be used to describe image regions within its frame according to the intensity of the image [6]. According to possible thresholding of a gray-level image, all pixels above or equal threshold will be referred to white, and all below the threshold will be referred as black. In fact, MSER has been improved based on the color gradient by using a color threshold to detect and track the colored objects [7], [4]. Mobile applications of MSERs have increased and show high stability based on Maximally Stable Regions (MSR) [8], and they set a threshold of different intensity images,  $I$ , to binary images as [7]:

$$E_t(x) = \begin{cases} 1 & \text{if } I(x) \geq t \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

In sample word, MSER technique does the following: Intensity function will be used to compute the stability  $\psi$  of the region [9].

$$\psi = (Q(i + d) - Q(i - d))/Q(i) \quad (4)$$

Or we can find the stability by rewriting the function as follows [10]:

$$\psi(R_i^g) = (|R_i^{g-\Delta}| - |R_i^{g+\Delta}|)/(|R_i^g|) \quad (5)$$

Where  $R_i$  is the connected region,  $R_i^g$  the region detected by applying a threshold on a gray value  $g$ ,  $\Delta$  is stability factor.  $R_i^{g-\Delta}$  and  $R_i^{g+\Delta}$  are the extremal regions located above and under the region  $R_i^g$ . MSERs are recognized at the intensity level at which function “Q” hits a local minimum. The parameters which controlled the MSERs are delta  $\Delta$  and the extremal region area  $R$ . Whenever  $\Delta$  increases as lesser regions will be detected. The intensity variation will be given as:

$$\frac{|R(+\Delta) - R|}{|R|} \quad (6)$$

MSER detects the maximally stable region, which has lowest variation than regions above and below [11]. However, MSERs fail to achieve good performance with some types of motion, also, don't count the acceleration for the tracker feature which describe the platform movements. In this paper, we integrate motion estimation (ME) with MSER into MSERME to achieve great performance.

### III. THE PROPOSED ALGORITHM FRAMEWORK

The system has been implemented in a mobile device. The novelty of our proposal is to be effectively able to describe the camera behaviors based on track local features in order to detect the camera carrier behavior as well. Figure (1) shows the main blocks in the proposed algorithm.

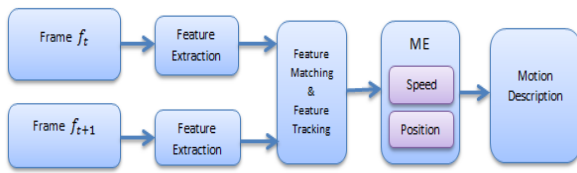


Fig. 1. The main blocks of the algorithm.

### IV. LOCAL FEATURE DETECTION AND EXTRACTION

Computer vision has well established detectors techniques; such as: SIFT, SURF, HARRIS, BRISKP and MSER. The main tasks for these detectors are detecting the interest points and local features. In fact, these are the main blocks of object tracking, object recognition, stereo vision matching and robotics navigation. Overall, an input image features will be detected by MSER in allocating regions. After extracting the features MSER will achieve matching between the same features in two frames as shown above in this case  $f_t$  &  $f_{t+1}$ . Based on success matching, tracking process will be activated. The Motion Estimation ME algorithm will calculate the speed of motion and the direction. In this work we used Optical Flow for this purpose.

### V. TRACKING SELECTED INTEREST POINTS

The tracking processes are used to construct the trajectories to include all information of the object's motion and appearance. The evaluation of these trajectories will be based on the quality of their tracking and that shows the summary of object's robustness and representations [1]. Therefore, the target of tracking is tracing the object motion to view the best representation for each feature in the region independently. Indeed, the local features which are detected can be tracked by MSERs as identified regions. These regions should be converted to gray values, and then MSER will analyze them. For longer video stream or image sequences, there will be more MSERs will appear and later will disappear in the same sequences [12]. That means MSERs will be extracted from each video frame. Moment-based descriptors are used to track MSERs from frame to the next. Camera motion should be compensated by using such technique for example feature-based registration [9].

### VI. MATCHING AND TRACKING

Matching process needs at least two frames. After we done the local feature detection and have enough interest points. Most of these interest points will appear again in the next frame. Matching these points between two frames should be done to insure a smooth transaction. In order to achieve good tracking we should have good matching with enough interest points. In our process we used MSER to achieve this task. Figure (2) shows an example of matching between two frames. The interesting and powerful factor of using MSERs is the matching with fully affine distortion which comes under performance of use of Epipolar geometry and dense matching. The figure (3) shows two MSERs matched regions with different angles for their feature's appearance.

The pairs of matched local features can be illustrated as  $m_i = \{(x_{i1}, y_{i1}), (x_{i2}, y_{i2}), w_i\}$  and  $m_j = \{(x_{j1}, y_{j1}), (x_{j2}, y_{j2}), w_j\}$  same as figure (3)(b). If the local feature angles  $\alpha_1$  &  $\alpha_2$  are similar, the two regions are matched successfully. The matching between  $m_i, m_j$  Based on measuring the two local angles can be given by this assumption:

$$\delta(m_i, m_j) = \begin{cases} 1 & \text{if } |1 - \alpha_2/\alpha_1| < \delta_D \\ 0 & \text{Otherwise} \end{cases} \quad (7)$$

$\delta_D$  is a predefined threshold that gives the acceptable error range of angle change [5]. The similarity between MSERs can be identified by following equation:

$$SS(r_i, r_j) = \frac{\sum_{i \neq j, m_i, m_j \in M} \delta(m_i, m_j)}{C_{|M|}^2} \quad (8)$$

Flow chart in figure (4) shows the steps of applying these equations in our code.

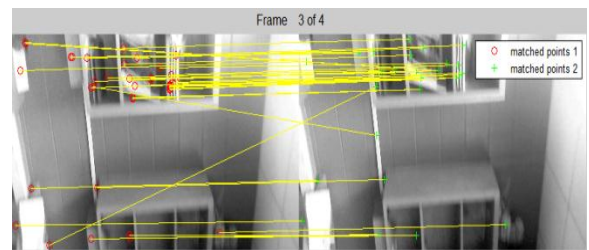


Fig. 2. Matching between two frames.

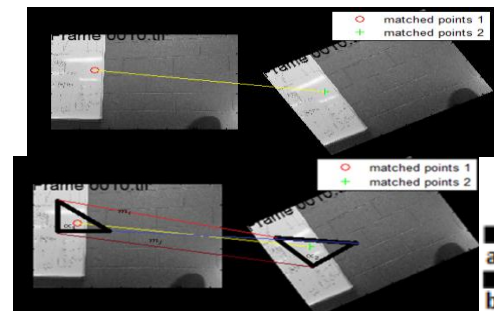


Fig. 3. (a, b) Show MSERs matching performance when there is difference between the matching frames ( $\alpha_1$  frame 1,  $\alpha_2$  frame 2 after matching.)

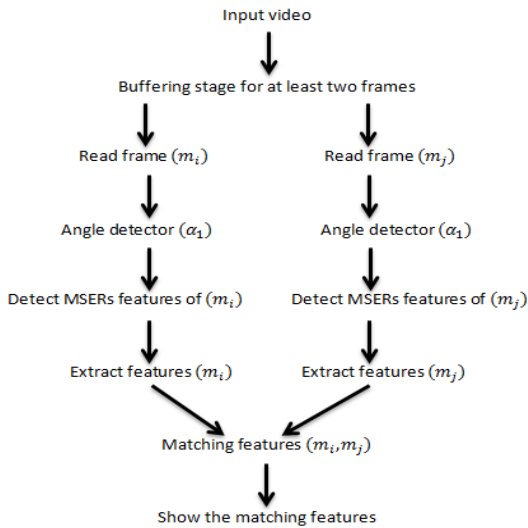


Fig. 4. Flow chart of matching performance

This study uses MSER for tracking tasks and the advantage of using MSER over other trackers; for example, corner tracking has high challenges when the objects getting smaller and smaller. However, MSER requires a structure of controlled data. This structure is a component tree which allows MSERs for detection and tracking.

### VII. MOTION ESTIMATION

Moving features are appearing in the video stream in different positions and this is called motion. To describe this motion we should first be able to detect this motion and then track it. These two steps call motion estimation (ME). This can be used to reduce sequential redundancy between adjacent frames [13]. Since the techniques we used are feature based, motion estimation can be applied for long range, which is usually accrued in high velocity of the image plane [14].

Motion can be caused by three main factors: Object movement, camera movement and light changes. In most cases there will be more than one factor causing the motion [15]. In general, motion is companions of multi transformations such as: Affine, Homography, Similarity and Translation. Therefore, it comes as strong technique to solve many problems in computer vision. For example, image alignment and matching. Based on what we

discussed above, video processing applications have much to do with motion estimation; for instance, tracking and recognition. Motion has many figures; for example, robust estimation, motion representation, motion perception, parametric motion and dense optical flow which we used in this study.

### VIII. SPEED MOTION ESTIMATION

The motion speed shows the movement speed of objects throughout the distance and the direction of the object would cover respect to the time. Therefore, motion estimation sensing the motion and speed motion is the rate at which the object changes its location. Basically, speed of motion which is in this work is a speed with a direction. The goal of this process is to find the velocity vector for each pixel which tells how quickly this pixel is moving in this moment across the image. Secondly, shows the direction of this movement  $\vec{u} = (u, v)$ .

### IX. MOTION DIRECTION ESTIMATION

The direction of motion will be as same as the direction of object movements regardless of object speedy either is slow or fast. As we mentioned above, speed of motion is a combination of speed with direction, so whenever there is a motion, there will be speed and direction estimation. Figure (6) shows some different movements and the small colored arrows show the motion direction. In this task we used different colored to show directions.

In this paper, we consider the Optical Flow invented by Lucas Kanade (LK) [16]. The motion field presents 3D motion while Optical Flow presents the motion field projection onto the 2D image. Modeling the camera can be done by using six parameters to estimating the camera motion. 2D affine transformation camera model is identified by the following matrix:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Any point on the reference frame has position  $(x, y)$ . After applying the transformation model, will have the point  $(x', y')$  on the new frame.  $m_1 \dots m_6$  are translation parameters.

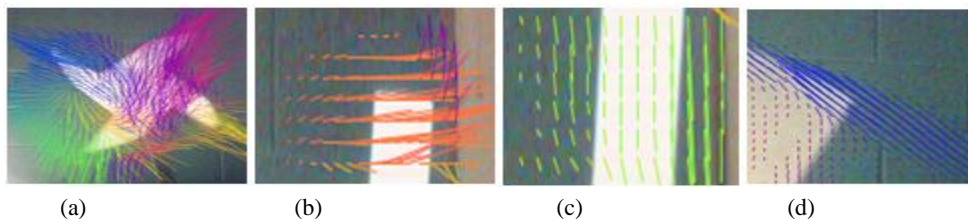


Fig. 6. Show the motion directions in different situation (a) Shows rotation movements. (b) Movement from left to right horizontally. (c) Vertical movement from up to down. (d) Diagonal movement form the down right corner up to the left corner.

X. ANALYSES AND DISCUSSION

In order to describe any motion by using different existing algorithms, basically we have to apply many steps of processing before achieving that point. As we showed in figure (1), upload video frames is a first step. Using machine Intel(R) Core(TM) i3, the proposed algorithm achieved quick respond to very small movements from the eyebrows (less than 0.39 Inch) and small camera movements in MSER, ME and MSERME perspectives as a figure (7) shows.

MSERs give the information related to what exists in the seen and that indeed is important to insure the tracking and description tasks which create the relationship between the sequences of frames. MSERs don't distinguish between the standing and moving objects, however, it shows the different orientation for features over time, which give an estimation about where the camera is going based on the object position and vice versa. Figure (8) shows the two situations. As a result of these experiments by using recorded data set and real time videos with different backgrounds and illuminations, we achieved the objectives of this study. High tracking accuracy with almost zero-time delay to estimate any motion into the frame boundaries and robustness were provided.



Fig. 7. Shows the response of motion algorithm to very small movements.

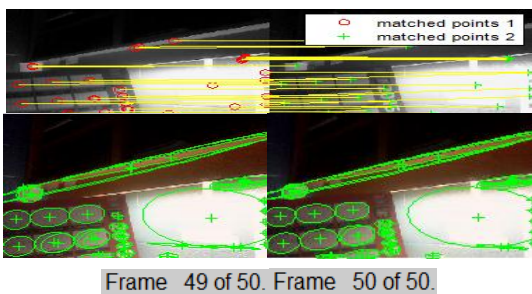


Fig. 8. (a) Shows selected features before matching. (b) Shows the matched selected features in the next frame. (c) Show the orientations of selected features before matching. (d) Show the orientations of selected features after matching.

Table 1. The accuracy of using ME, MSER and MSER-ME

Environment		Dark		Light		Accuracy	
		Busy	Poor	Busy	Poor		
ME	Small + Slow	D	N	D	D	75%	75%
	Small + Fast	D	N	D	N	50%	
	Large + Slow	D	D	D	D	100%	
	Large + Fast	D	D	D	D	75%	
MSER	Small + Slow	D'	N	D	D	50%	68.75%
	Small + Fast	D'	N	D	N	25%	
	Large + Slow	D	D	D	D	100%	
	Large + Fast	D	D	D	D	100%	
MSER-ME		100%	50%	100%	100%	87.5%	

For motion detection we only have two situations either to detect the moving features or don't, and then we transfer that to a percentage value, so N=0% & D= 50% or 25%, for vertical or horizontal calculation respectively as we can see in the table (1). MSER-ME accuracy is calculated by considering each type of movement was detected either by MSER and/or ME and will be counted on both sides, the side which failed to detect the motion will get D' which has the same value of D. Practically, most of detections will be done in light environment so the accuracy will be found as follows:

$$A = \frac{(100 + 87.5)\%}{2} = 93.75\% \approx 94\% \quad (9)$$

XI. CONCLUSION AND FUTURE WORK

Using MSER algorithm contributed fast and the stable response system. However, MSER-ME algorithm provides considerable efficiency by overpowering the weakness of MSER: sensitivity to very small movements, acceleration sensing and direction prediction. The algorithm promoted high accuracy above than 93% of detecting and tracking even for small objects in a diversity of distances and backgrounds, then the description task done based on that to characterize all the camera movements. However, combining the system with probabilistic estimation will escalate the proficiency of the algorithm.

REFERENCES

- [1] M. D. H. B. Hayko Riemenschneider, "Online Object Recognition by MSER Trajectories," in *19th International Conference on Pattern Recognition (ICPR 2008)*, Tampa, Florida, USA, 2008.
- [2] D. L. S.-H. W. P. Forss'en, "Region detectors," 2014.
- [3] P. K. B. Manish Okade, "Improving Video Stabilization Using Multi-Resolution MSER Features," *IETE Journal of Research*, vol. 60, pp. 373 - 380, 2015.
- [4] X.-C. Y. X. Y. H. A. H.-W. H. Khalid Iqbal, "Classifier Comparison for MSER-Based Text Classification in Scene Image," *Neural Networks (IJCNN), International Joint Conference on*, pp. 2218 - 2225, 2014.
- [5] J.-L. L. J.-M. M. P. M. F. S. Frédéric Cao, *A Theory of Shape Identification*, Springer, 2008.
- [6] O. C. M. U. T. P. J. Matas, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions," *Image and Vision Computing*, pp. 761 - 767, 2004.
- [7] P.-E. Forssen, "Maximally Stable Colour Regions for Recognition and Matching," *CVPR '07. IEEE Conference on*, pp. 1 - 8, 2007.
- [8] O. C. M. U. T. P. J. Matas, "Robust wide baseline stereo from maximally stable extremal regions," *In 13th BMVC*, p. 384 - 393, 2002.
- [9] N. G. Sean Varah, "Target Detection Tracking Parallel MSER Implementation," MotionDSP Inc., 2013.
- [10] F. Hamprecht, "Maximally Stable Extremal Regions | Image Analysis Class. youtube," HCI / Heidelberg University , 21 5 2013. [Online]. [Accessed 27 9 2015].
- [11] B. F. ., K. L. D. P. M. P. M. S. ., H. S. Andrea Vedaldi, "www.VLFeat.org," 2013. [Online]. [Accessed 30 09 2015].
- [12] H. B. Michael Donoser, "Efficient Maximally Stable Extremal Region (MSER) Tracking," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 1, pp. 553 - 560, 2006.
- [13] G. C. X. G. Dongming Zhang, "Improved Motion Estimation Based on Motion Region Identification," *International Conference on Systems and Informatics (ICSAI)*, pp. 2034 - 2037, 2012.

- [14] N. N. W. N. M. Wei-Ge Chen, "Image Motion Estimation From Motion Smear A New Computational Model," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 18, no. 4, pp. 412 - 425, 1996.
- [15] C. Liu, "Motion Estimation (I)," Microsoft Research, New England, 2011.
- [16] B. M. Ng Teck Khim, "Lucas-Kanade Tracking," CMU, Penn State, 2004.