

Improving Performances of MSER Features in Matching and Retrieval Tasks

Andrzej Śluzek^(✉)

ECE Department, Khalifa University, Abu Dhabi, United Arab Emirates
`andrzej.sluzek@kustar.ac.ae`

Abstract. MSER features are redefined to improve their performances in matching and retrieval tasks. The proposed SIMSER features (i.e. *scale-insensitive* MSERs) are the *extremal regions* which are maximally stable not only under the *threshold* changes (like MSERs) but, additionally, under *image rescaling* (smoothing). Theoretical advantages of such a modification are discussed. It is also preliminarily verified experimentally that such a modification preserves the fundamental properties of MSERs, i.e. the average numbers of features, repeatability, and computational complexity (which is only multiplicatively increased by the number of scales used), while performances (measured by typical CBVIR metrics) can be significantly improved. In particular, results on benchmark datasets indicate significant increments in *recall* values, both for descriptor-based matching and word-based matching. In general, SIMSERS seem particularly suitable for a usage with large visual vocabularies, e.g. they can be prospectively applied to improve quality of BoW pre-retrieval operations in large-scale databases.

Keywords: MSER features · Scale invariance · Multi-scale image pyramid · Keypoint matching · Image retrieval

1 Introduction and Background

MSER features (originally proposed in [1] and computationally improved in [2]) continue to attract attention of machine vision researchers and practitioners. In comparison to other affine-invariant features, their main advantages are: (1) moderate computational complexity and the algorithmic structure suitable for hardware implementations, e.g. [3,4], and (2) a good identification of significant image parts usually combined with high repeatability under typical image distortions (as reported in [5]).

Nevertheless, some disadvantages of MSER features have been identified. In particular, MSERs have limited performances on blurred and/or textured images. Both cases are actually related to the image scale, since blur (which can distort shapes of extracted MSERs) is equivalent to image down-scaling, e.g. [6]. Similarly, shapes of fine texture details can vary irregularly under image rescaling. Protruding fragments of non-convex MSERs are particularly vulnerable to scale variations (as discussed in [7]).

Thus, a number of papers have been addressing the issue of *actual* insensitivity of MSER features to scale variations. A simple approach (based on the original concept of MSER detection) is proposed in [8]. The authors just detect MSERs in a pyramid of down-scaled images, and keep all of them (after removing near-duplicate MSERs which reappear at different scales). More recently, combinations of the original MSER algorithm with other techniques have been proposed to improve performances of MSER detection. For example, in [7], alternative stability criteria for moment-normalized extremal regions are applied to improve affine-invariance for blurred areas. In [9], MSERs are detected over saliency maps (highlighting boundaries) of images. Again, the primary objective is to improve robustness to blur.

In this paper, we also strive to correct the above-mentioned inadequacies of MSER features and, subsequently, to improve reliability of MSER-based image matching/retrieval. The main objective is to preserve, as much as possible, the original principles of MSER detection and, therefore, our approach is closer to the ideas outlined in [8] rather than to more complicated improvements proposed in other papers.

In general, instead of detecting maximally stable extremal regions in 1D space of intensity thresholds (the original MSER algorithm) the proposed method identifies extremal regions which are maximally stable both under the threshold changes and under the scale variations (i.e. blur). We demonstrate that such a switch to a 2D space only moderately increases computational complexity, while performances (evaluated by most typical metrics of images matching) can be significantly improved. We also illustrate on simple analytical examples that intuitive notions are better satisfied by the proposed model than by the original MSER model.

Section 2 of the paper presents formal details of the proposed algorithm, and illustrates selected effects resulting from this theoretical model. A limited scale experimental verification of the algorithm's performances is presented in Sect. 3. Concluding remarks are included in Sect. 4.

2 Mathematical Models

2.1 Standard MSER Detection

Maximally stable extremal regions have been defined in [1] as black (or white) areas of a thresholded image which only insignificantly vary under threshold changes. Formally, a binarized region $Q(t)$ (where t indicates its threshold level) is considered MSER if the growth rate function $q(t)$ defined by the derivative of the region area over the threshold values:

$$q(t) = \frac{\frac{d}{dt} \|Q(t)\|}{\|Q(t)\|}, \quad (1)$$

reaches a local minimum ($\|\cdot\|$ indicates the region area).

In practice, Eq. 1 is substituted by one of its discrete approximations:

$$q(t_j) = \frac{\|Q(t_j) - Q(t_{j-1})\|}{\|Q(t_j)\|} \quad \text{or} \quad q(t_j) = \frac{\|Q(t_{j+1}) - Q(t_{j-1})\|}{\|Q(t_j)\|}, \quad (2)$$

where the difference $t_j - t_{j-1}$ defines the threshold increment Δt .

The above formulas apply to both dark and bright MSERs (for the latter, images should be inverted).

A number of other parameters is used to control stability of MSER detection and to reduce the nesting effects (e.g. caused by blurs), see [1, 2].

2.2 SIMSER Features

The proposed improvements of MSER features are motivated by the results from several papers (see Sect. 1) which indicate that taking into account multiple resolutions (image blurring) may improve performances. Since (to the best of our knowledge) no formal model of MSER detection in multiresolution images seems to exist, we propose *scale-insensitive maximally stable extremal regions* (SIMSERS) model which extends the mechanism of MSER detection into a 2D space $Threshold \times Scale$. Although the name *scale-insensitive* sounds redundant because MSERs are supposed to be scale-invariant by default, we can argue that such a name modification highlights improvements in the *actual* invariance of these features to rescaling (blurring) effects.

Given an image presented over a range of scales $s \in S$ (i.e. a family of images) and binarized using a range of thresholds $t \in T$, an extremal regions $Q(s, t)$ (where s defines the current scale and t indicates the current binarization threshold) is considered SIMSER, if two growth rate functions $q_1(s, t)$ and $q_2(s, t)$ defined by the partial derivatives of the region area over s and t jointly reach the local minimum there:

$$q_1(s, t) = \frac{\frac{\partial}{\partial t} \|Q(s, t)\|}{\|Q(s, t)\|}, \quad (3)$$

$$q_2(s, t) = \frac{\frac{\partial}{\partial s} \|Q(s, t)\|}{\|Q(s, t)\|}. \quad (4)$$

To illustrate the concept of region stability under blurring (scaling), Fig. 1 shows evolution of a selected dark extremal region over a number of scales.

2.3 Theoretical Advantages of SIMSERS

Advantages of SIMSERS can be preliminarily discussed using two test images, with rectangular and triangular intensity profiles, correspondingly. The images are shown in Fig. 2 (1D case is selected to simplify calculations).

MSER features are extracted from these images at minima of the growth rate function $q(t)$ (see Eq. 1), while SIMSERS are extracted at joint minima of two growth rate functions $q_1(s, t)$ and $q_2(s, t)$ (see Eqs. 3 and 4). For SIMSER



Fig. 1. A sequence of dark extremal regions over a number of neighboring scales (with the same threshold). The framed central region is maximally stable under scale changes, and may be eventually identified as SIMSER (if it is also maximally stable in the threshold dimension).

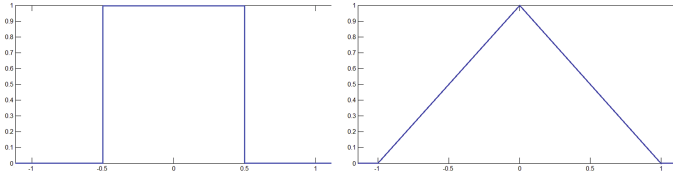


Fig. 2. Two 1D images with rectangular and triangular intensity profiles. The range of intensities is $\langle 0; 1 \rangle$.

extraction, the family of multi-scale images (where scale s ranges from 0 to ∞) is created using simple image averaging $i_s(x) = \frac{1}{2s} \int_{x-s}^{x+s} i(\zeta) d\zeta$, i.e. $s = 0$ represents the original image and larger scales correspond to more smoothing.

For the rectangular intensity profile (left image in Fig. 2):

$$q(t) = 0, \quad q_1(t, s) = \frac{4s}{1 + 2s(1 - 2t)} \quad \text{and} \quad q_2(t, s) = \frac{|2 - 4t|}{1 + 2s(1 - 2t)}. \quad (5)$$

Therefore, the number of extracted MSERs is either infinite or zero (depending on the interpretation of zero values of $q(t)$), while SIMSER is detected only once for $t = 0.5$ and $s = 0$, i.e. the image should not be smoothed and the threshold is at half of the maximum intensity. Such a result is intuitively more plausible.

For the triangular intensity profile (right image in Fig. 2), $q(t) = \frac{1}{1-t}$ reaches the minimum only once for $t = 0$. Intuitively, MSER should be rather detected somewhere at non-zero threshold. The functions $q_1(s, t)$ and $q_2(s, t)$, however, have too joint minima (details are not presented because of complex and tedious mathematical analysis). First, SIMSER is detected for $t = 0$ and $s = 0$ (which is identical to MSER above), while the second SIMSER exists for $t = 0.5$ and $s = 0.5$, i.e. the image should be slightly smoothed and thresholded at half of the maximum intensity. Again, SIMSER detection results seem to be more complete and plausible.

2.4 Implementation Details and Computational Complexity

The numerical schemes for computing $q_1(s, t)$ and $q_2(s, t)$ growth rate functions in discretized *Threshold* \times *Scale* space are basically the same (following Eq. 2), i.e.

$$q_1(t_j, s_k) = \frac{\|Q(t_j, s_k) - Q(t_{j-1}, s_k)\|}{\|Q(t_j, s_k)\|} \text{ or } q_1(t_j, s) = \frac{\|Q(t_{j+1}, s_k) - Q(t_{j-1}, s_k)\|}{\|Q(t_j, s_k)\|}, \quad (6)$$

$$q_2(t_j, s_k) = \frac{\|Q(t_j, s_k) - Q(t_j, s_{k-1})\|}{\|Q(t_j, s_k)\|} \text{ or } q_2(t_j, s_k) = \frac{\|Q(t_j, s_{k+1}) - Q(t_j, s_{k-1})\|}{\|Q(t_j, s_k)\|}. \quad (7)$$

In line with recommendations from the original MSER papers and Matlab, we use in the subsequent experiments the threshold increment $\Delta t = 3$ (for images with 256 levels of intensity).

The scale-space increments follow the standards of multi-scale image processing (e.g. [10]), i.e. the original image is repetitively convolved with a smoothing filter equivalent to halving the image resolution. The minimum equivalent image size is assumed 64 because 32 pixels is a default (in Matlab) minimum size of MSER features, and we assume (somehow arbitrarily) that the largest MSER should not cover more than a half of an image. Thus, the number of scales NS is defined by

$$NS = 1 + \lfloor \log_2(n/64) \rfloor, \quad (8)$$

where n is the image resolution.

For example, for images of VGA resolution 640×480 the recommended number of scales is 13.

Based on the structure of Eqs. 6 and 7, we can preliminarily claim that computational complexity of SIMSER detection is the same as the complexity of MSER detection (subject to the multiplication by the number of scales NS , which is considered constant and as such it does not affect the theoretical estimate), i.e. $O(n \times \log(\log(n)))$ or $O(n)$ (the former based on [1], and the latter given in [2]).

The only issue is to verify whether the growth rate function $q_2(t_j, s_k)$, which does not exist in the original MSER algorithm, has the same or lower computational complexity. The problem is that the extremal regions over the sequence of threshold values are always nested, while the extremal regions over the sequence of scales generally do not nest (an illustrative example is given in Fig. 3), i.e. the topology of extremal regions may unpredictably change under image smoothing. Nevertheless, a simple algorithm has been proposed to track correspondences between extremal regions in the neighboring scales and to (simultaneously) compute the growth rate function $q_2(t_j, s_k)$. A commented pseudo-code of this algorithm is shown at the end of the paper. It is deliberately not optimized (a more



Fig. 3. Evolution of a dark extremal region over two neighboring scales (smoothing removes sharp fragments). The regions intersect significantly, but there is no nesting in either direction.

practical variant is outlined in [11]) to clearly illustrate its $O(n)$ complexity. This pseudo-code corresponds to the left expression in Eq. 7.

3 Preliminary Experimental Evaluation

SIMSER features have been experimentally compared to MSERs using several aspects of their performances. In general, the objective was to evaluate gains achieved by using SIMSERS instead of MSERs. Therefore, the results are given *relatively* to the corresponding MSER results (which are considered references with unit values).

3.1 General Properties

First, we established relations between numbers and distributions of SIMSER and MSER features in typical images. Experiments have been conducted on a large number of images (including benchmark datasets used in other experiments mentioned below), and the conclusions are as follows:

1. The numbers of SIMSER features are generally similar to the numbers of MSERs, even though SIMSERS are found from a multi-scale pyramid of images. The average number of SIMSER features is 109 % of the average number of MSERs (with very few cases outside 80 %–130 % range).
2. SIMSER features are generally better concentrated in the areas of higher visual prominence (see an example in Fig. 4) which suggests that SIMSERS are more likely to maintain their numbers under image distortions. Using a collection of near-duplicate images (including the benchmark dataset at [12]) distorted by illumination changes, blur, JPEG compression, rotation and scaling, we found that the standard deviation of SIMSER numbers within the same image under diversified distortions is, in average, lower by 38 % than the standard deviation of MSER numbers.
3. It seems the numbers of SIMSERS are less sensitive to *MaxAreaVariation* parameter which is used (see the Matlab notation at [13]) to define acceptable minima of growth rate functions.

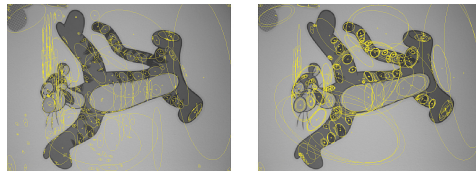


Fig. 4. MSER (left) and SIMSER (right) detection in an exemplary image. To maximize the number of features, *MaxAreaVariation* parameter was not restricted.

3.2 Keypoint Detection and Matching

We have used three popular metrics to compare performances of SIMSERS and MSERs. First, keypoint *repeatability* is evaluated. Subsequently, reliability of keypoint matching is estimated using *precision* and *recall* parameters. Although these two parameters evaluate primarily performances of keypoint descriptors, they can be instrumental in assessing keypoint detectors as well. *Precision* and *recall* can be compared using the same descriptor over keypoints extracted by alternative detectors. We have adopted this approach, with SIFT descriptor computed over MSER and SIMSER features. SIFT (in RootSIFT variant) has been selected because of its popularity and good performances. Results based both on matching SIFT descriptors (vectors) and matching SIFT-based *visual words* are discussed; the latter for the practical importance.

Repeatability. Repeatability of MSERs and SIMSERS was compared on a popular dataset [12] which provides homographies between *the-same-category* images, so that the ground-truth keypoint correspondences can be identified similarly to [5]. It was found that both types of features have practically the same repeatability (actually, repeatability of SIMSERS is slightly higher by a statistically negligible margin of 2.4%).

Matching (Keypoint Descriptors). *Precision* and *recall* of keypoint matching was evaluated on the same dataset [12]. First, SIFT descriptors are matched by the *one-to-one* (O2O) method (using the *mutual nearest neighbor* approach) which is considered a recommended setup returning the most credible matches, e.g. [14].

As seen in the first row of Table 1, SIMSERS outperform MSERs by a wide margin (both in *recall* and *precision* values). An illustrative example given in Fig. 5 shows a larger number of true correspondences (and fewer incorrect matches) for SIMSERS than for MSERs.

Matching (Visual Words). In actual applications (e.g. CBVIR) keypoints are represented by visual words so that *precision* and *recall* based on matching visual words are more significant in practice. Various sizes of visual vocabularies can be

Table 1. Comparative performances of matching SIMSER and MSER features using SIFT descriptors/words (the results obtained for MSERs are represented by the reference value 1.0).

Method	Precision	Recall
O2O - SIFT descriptors	2.42	2.34
M2M - 32M SIFT words	0.39	11.22
M2M - 1G SIFT words	0.46	12.78

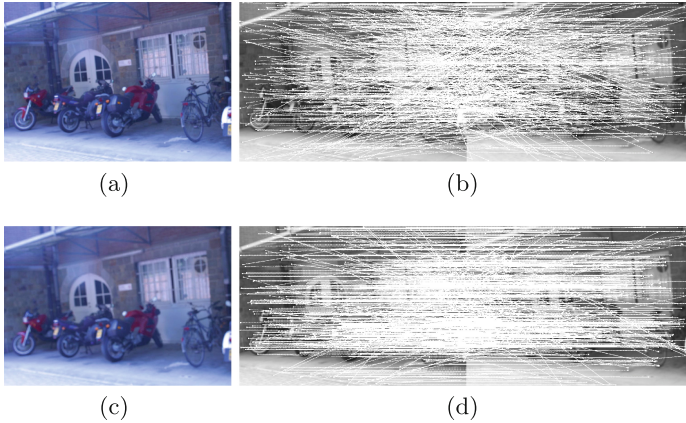


Fig. 5. A pair of near-duplicate images (a, c), and their O2O matching results for SIFT descriptors over MSERs (b) and SIMSERS (d).

used, but large vocabularies (at least a few million words) are recommended in several important works, e.g. [15, 16], to provide sufficient *precision* (even though *recall* may suffer). Therefore, we performed tests focusing on very large vocabularies (vocabularies of 32 million and 1 billion words are used as examples). To minimize the computational costs of vocabulary building and word assignment, such large vocabularies are defined using a technique somehow similar to simplified *binary* embedding, where the numerical value of the code is considered the word number. Somehow surprising results (showing huge improvements in *recall* values) for those two exemplary vocabulary sizes are given in the lower part of Table 1.

The results of Table 1 can be interpreted in the context of Fig. 4 which shows that SIMSER regions (i.e. their best-fit ellipses) tend to nest more frequently than MSER ellipses (especially in most contrasted parts of images).

While nested MSERs are undesirable because they create a number of near-identical descriptors, nested SIMSERS are more useful since they represent visual data in diversified scales. Thus, their descriptors are usually sufficiently distinctive to be quantized into different words. As a result, several alternative words are found to represent what effectively is the same visual content. When near-duplicate image fragments are matched, chances are much higher that some of those alternative words are identical. An illustrative example is given in Fig. 6.

Those alternative visual words may, nevertheless, cause a drop of *precision*, as seen in Table 1 and Fig. 6. In our opinion, this is acceptable because for large vocabularies *precision* is usually so high that even if it drops by half (or more) it can still be considered satisfactory (especially if combined with a huge improvement in *recall*).

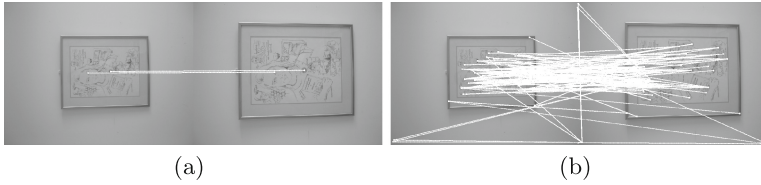


Fig. 6. In (a), MSER features of two images are matched using a vocabulary of 32M words. *Precision* is 100 %, but *recall* is very low. In (b), the same vocabulary is used to match SIMSERS. *Precision* is lower, but *recall* improves dramatically.

3.3 Image Matching

Finally, performances of SIMSERS and MSERs have been compared in a typical CBVIR task, i.e. retrieval of near-duplicate images. A popular UKB dataset (see [17, 18]) was selected because of its regularity (classes of similar images do not intersect, and each class consists of exactly 4 images).

Using the same very large vocabularies of SIFT-based words of 32 M and 1G words (mentioned in Sect. 3.2), *bag-of-words* (BoW) histograms are built over MSER and SIMSER features and used for pre-retrieval of images ranked by the similarity of their BoW's to the query BoW. Individual images randomly selected from each class are used as queries.

Because our analysis is not targeting any particular database, BoW normalization requiring database statistics (e.g. *td-idf*, [19]) cannot be applied, and we use histograms of *absolute* word frequencies. We selected a simple *histogram intersection* measure of histogram similarities (proposed in [20]), where the distance between two histograms H_1 and H_2 over Voc vocabulary is defined by

$$d(H_1, H_2) = \sum_{w \in Voc} \min(H_1(w), H_2(w)), \quad (9)$$

which nicely corresponds to the intuitive notion of similarity between images.

The results shown in Table 2 give the relative values *mean average precision* (*mAP*) for SIMSER-based approach (MSER-based results are considered unit-valued references). Again, two vocabulary sizes (i.e. 32 M and 1G words) are used, and *mAP* is computed for two different scenarios. First, *mAP* is evaluated from all retrieved images, and in the second scenario only 20 top-ranked images are taken into account.

Table 2. Relative *mean average precision* (*mAP*) of UKB image retrieval using two large vocabularies over SIMSER features (the corresponding MSER-based results are consider references with 1.0 value).

Scenario	<i>mAP</i> for 32M vocabulary	<i>mAP</i> for 1G vocabulary
All retrieved images	1.57	1.63
Only 20 top-ranked images	2.25	2.36

The results indicate significant improvements, in particular for fixed numbers of pre-retrieved images. This is because for large vocabularies *precision* of SIMSERS is much lower than for MSERS (see Table 1) so that the total number of pre-retrieved *false-positive* images would be larger for SIMSERS.

As an example, 8 top-ranked images retrieved by BoWs built over MSERS and SIMSERS are shown in Fig. 7 (using the first image of UKB database as a query).

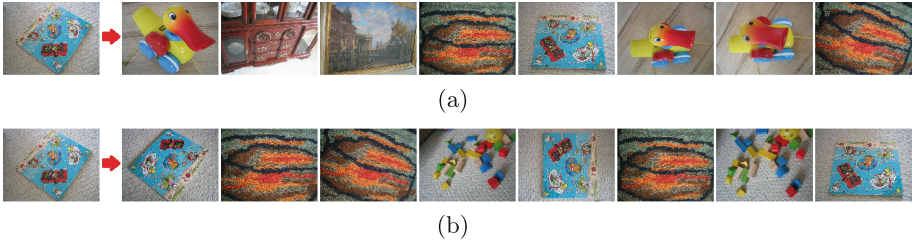


Fig. 7. Top-ranked retrievals for the first UKB image (note that in UKB each query has only three relevant images). In (a), BoW built over MSERS is applied, while in (b) BoW is built over SIMSERS. The same vocabulary of 1G words is used in both cases.

4 Conclusions

In the paper, MSER features are redefined to improve their performances, primarily in matching and retrieval tasks. Novel SIMSER features (i.e. *scale-insensitive* maximally stable extremal regions) are presented as an alternative to MSERS. SIMSERS are detected at joint local minima of two growth rate functions (in the *threshold* dimension and in the *scale* dimension), i.e. they are extremal regions which are maximally stable not only under the *threshold* changes (like MSERS) but, additionally, under *image rescaling* (smoothing). The proposed feature detector has the same complexity as MSER detector (subject to the multiplicative factor corresponding to the number of employed scales).

It has been verified that other important characteristics, namely the average numbers of features in images and repeatability of features, are practically the same for SIMSERS and MSERS.

However, CBVIR-related performances, i.e. *recall* and *precision* of key-point matching, and *mean average precision* of image retrieval, are significantly improved for SIMSERS. In particular, results on a benchmark dataset indicate that SIMSERS are recommended in conjunction with huge-size visual vocabularies, for which they achieve a tenfold increase of *recall* over MSERS. This is an important property, because dramatically lowering *recall* has been detrimental to the usage of huge visual vocabularies (which, otherwise, are recommended for large-scale CBVIR applications).

Apart from typical CBVIR tasks, SIMSERS can prospectively replace MSERS in other applications. For example, we envisage that image segmentation could

be one of such areas, because multiple-scale MSERs are explicitly used there (e.g. [21]) or MSER-like structures are exploited as a supplementary tool (e.g. [22]).

Pseudo-code for computing $q_2(t_j, s_k)$ growth rate function

```

Input: Im1(M,N), Im0(M,N)
% two binary images of M*N size (at two neighbouring scales)
Input: Rs1(M,N), Rs0(M,N)
% two labeled images of M*N size
Input: SoR1(K1), SoR0(K0)
% list of region sizes (K1 regions in Im1 and K0 regions in Im0)
Storage: Ints(K1,K0) <- zeros
% size of intersections between regions from Im1 and Im0
Output storage: q2(K1) <- large value
% q2 function for Im1 regions initialized with very large values
Output storage: previous(K1) <- zeros
% the numbers of the corresponding regions from the previous scale

for i = 1:M
    for j = 1:N
        if Im1(i,j)==Im0(i,j)
            Ints(Rs1(i,j),Rs0(i,j))++;
        endif
    endfor
endfor
for i = 1:M
    for j = 1:N
        temp = SoR1(Rs1(i,j))+SoR0(Rs0(i,j)) - 2*Ints(Rs1(i,j),Rs0(i,j));
        temp = temp/SoR1(Rs1(i,j));
        if temp < q2(Rs1(i,j)) && Ints(Rs1(i,j),Rs0(i,j)) > 0
            q2(Rs1(i,j)) = temp; previous(Rs1(i,j)) = Rs0(i,j);
        endif
    endfor
endfor

```

References

1. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proceedings of British Machine Vision Conference, pp. 384–393 (2002)
2. Nistér, D., Stewénus, H.: Linear time maximally stable extremal regions. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5303, pp. 183–196. Springer, Heidelberg (2008). doi:[10.1007/978-3-540-88688-4_14](https://doi.org/10.1007/978-3-540-88688-4_14)
3. Kristensen, F., MacLean, W.: Real-time extraction of maximally stable extremal regions on an FPGA. In: Proceedings of IEEE Symposium on ISCAS 2007, pp. 165–168 (2007)
4. Salahat, E., Saleh, H., Sluzek, A., Al-Qutayri, M., Mohammed, B., Ismail, M.: A maximally stable extremal regions system-on-chip for real-time visual surveillance. In: Proceedings of 41st IEEE Industrial Electronics Society Conference, IECON 2015, pp. 2812–2815 (2015)

5. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. *Int. J. Comput. Vis.* **65**, 43–72 (2005)
6. Lindeberg, T.: Scale-space theory: A basic tool for analyzing structures at different scales. *J. Appl. Stat.* **21**, 224–270 (1994)
7. Kimmel, R., Zhang, C., Bronstein, A.M., Bronstein, M.M.: Are MSER features really interesting? *IEEE PAMI* **33**(11), 2316–2320 (2011)
8. Forssén, P.E., Lowe, D.G.: Shape descriptors for maximally stable extremal regions. In: *Proceedings of 11th IEEE Conference on ICCV 2007*, pp. 1–8 (2007)
9. Martins, P., Carvalho, P., Gatta, C.: On the completeness of feature-driven maximally stable extremal regions. *Pattern Recogn. Lett.* **74**, 9–16 (2016)
10. Lindeberg, T.: Feature detection with automatic scale selection. *Int. J. Comput. Vis.* **30**, 77–116 (1998)
11. Sluzek, A., Saleh, H.: Algorithmic foundations for hardware implementation of scale-insensitive MSER features. In: *Proceedings of 59th IEEE MWSCAS Symposium, Abu Dhabi (2016, accepted)*
12. Mikolajczyk, K.: <http://www.robots.ox.ac.uk/~vgg/research/affine/>
13. MathWorks®: <http://www.mathworks.com/help/vision/ref/detectmserfeatures.html>
14. Zhao, W.L., Ngo, C.W., Tan, H.K., Wu, X.: Near-duplicate keyframe identification with interest point matching and pattern learning. *IEEE Trans. Multimed.* **9**(5), 1037–1048 (2007)
15. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: *Proceedings of IEEE Conference on CVPR 2007*, pp. 1–8 (2007)
16. Stewénus, H., Gunderson, S., Pilet, J.: Size matters: exhaustive geometric verification for image retrieval. In: *Proceedings of 12th European Conference on ECCV 2012, Florence, vol. II*, pp. 674–687 (2012)
17. Stewénus, H.: <http://www.vis.uky.edu/stewe/ukbench/>
18. Nistér, D., Stewénus, H.: Scalable recognition with a vocabulary tree. In: *Proceedings of IEEE Conference on CVPR 2006, vol. 2*, pp. 2161–2168 (2006)
19. Sivic, J., Zisserman, A.: Video google: a text retrieval approach to object matching in videos. In: *Proceedings of 9th IEEE Conference on ICCV 2003, Nice, vol. 2*, pp. 1470–1477 (2003)
20. Swain, M., Ballard, D.: Color indexing. *Int. J. Comput. Vis.* **7**(1), 11–32 (1991)
21. Oh, I.-S., Lee, J., Majumder, A.: Multi-scale image segmentation using MSER. In: Wilson, R., Hancock, E., Bors, A., Smith, W. (eds.) *CAIP 2013. LNCS, vol. 8048*, pp. 201–208. Springer, Heidelberg (2013). doi:[10.1007/978-3-642-40246-3_25](https://doi.org/10.1007/978-3-642-40246-3_25)
22. Wang, G., Gao, K., Zhang, Y., Li, J.: Efficient perceptual region detector based on object boundary. In: Tian, Q., Sebe, N., Qi, G.-J., Huet, B., Hong, R., Liu, X. (eds.) *MMM 2016. LNCS, vol. 9517*, pp. 66–78. Springer, Heidelberg (2016). doi:[10.1007/978-3-319-27674-8_7](https://doi.org/10.1007/978-3-319-27674-8_7)