http://dx.doi.org/10.5392/JKCA.2014.14.03.011

MSER을 이용한 다중 스케일 영상 분할과 응용

Multi-scale Image Segmentation Using MSER and its Application

이진선*, 오일석**
우석대학교 게임콘텐츠학과*, 전북대학교 컴퓨터공학부/영상정보신기술연구소**

Jin-Seon Lee(jslee@woosuk.ac.kr)*, Il-Seok Oh(isoh@chonbuk.ac.kr)**

요약

다중 스케일 영상 분할은 영상 스타일링과 의료진단과 같은 여러 응용에서 매우 중요하다. 이 논문은 다중 스케일 구조를 확보하며 안정적이고 효율적인 MSER에 기반을 둔 새로운 알고리즘을 제안한다. 이 알고리즘은 영상에서 MSER를 수집한 후, 이것들을 특정한 순서대로 영상에 다시 그려 넣음으로써 영상을 분할한다. 영상 경계를 평활화하고 잡음을 제거하기 위한 계층적 모폴로지 연산을 제안한다. 알고리즘의 다중 스케일 특성을 보이기 위해, 여러 종류의 상세 단계 제어의 효과를 영상 스타일링에 적용한다. 제안한 기법은 이러한 효과를 시간이 많이 걸리는 다중 가우시언 평활화없이 수행한다. 분할 품질과 계산 시간 측면에서 민쉬프트-기반 Edison 시스템과 비교 결과를 제시한다.

■ 중심어 : | 영상 분할 | 영상 스타일링 | 모폴로지 | 다중 스케일 |

Abstract

Multi-scale image segmentation is important in many applications such as image stylization and medical diagnosis. This paper proposes a novel segmentation algorithm based on MSER(maximally stable extremal region) which captures multi-scale structure and is stable and efficient. The algorithm collects MSERs and then partitions the image plane by redrawing MSERs in specific order. To denoise and smooth the region boundaries, hierarchical morphological operations are developed. To illustrate effectiveness of the algorithm's multi-scale structure, effects of various types of LOD control are shown for image stylization. The proposed technique achieves this without time-consuming multi-level Gaussian smoothing. The comparisons of segmentation quality and timing efficiency with mean shift-based Edison system are presented.

■ keyword : | Image Segmentation | Image Stylization | Morphology | Multi-scale |

I. Introduction

Image segmentation is a critical mission of most computer vision system[1][2]. The multi-scale image segmentation aims at producing a hierarchical tree-structured partitioning of image plane to analyze scene in a series of scales[3][4]. This output is useful for LOD (level-of-detail) control in various tasks such as image stylization and medical diagnosis. For example, DeCarlo stylized images by drawing fine

접수일자 : 2014년 02월 17일
수정일자 : 2014년 03월 13일
심사완료일 : 2014년 03월 13일
교신저자 : 오일석, e-mail : isoh@chonbuk.ac.kr

details at user-specified salient regions while drawing other parts coarsely[5].

Conventional algorithms construct a multi-scale representation by applying Gaussian convolutions in well-controlled multiple levels. It is well-known that the Gaussian convolution is time-consuming operation. Then a linking process is used, which finds corresponding blobs in adjacent scales[6]. Due to its sequential processing, the segmentation results of natural images failed to be qualified for a practical use. Another deficiency of conventional algorithms is inefficiency in computation time. These motivated our research to develop a more efficient and effective way of obtaining the multi-scale segmentation result.

In this paper, we propose a new approach for the multi-scale image segmentation. We are inspired by MSER (maximally stable extremal region) technique that produces blobs. The blobs are highly featured regions in the sense that they are stable and salient, and have multi-scale structure[7].

This paper proposes a method of collecting MSERs and then rearranging them onto an image plane in an appropriate order that would generate desired segmentation result. Unlike conventional multi-scale approaches, the proposed method works without time-consuming multi-level Gaussian smoothing and without cumbersome blob linking process. One of limitations of all the image segmentation algorithms is under-segmentation. With respect to under-segmentation issue, the proposed algorithm is not inferior to the smoothing-based approaches because the smoothing makes the blurred edges.

The contributions of this paper are as follows:

- Extending blob detection functionality of MSER to segmentation of natural images
- Presenting tree traversal algorithms for image partitioning and boundary smoothing
- Validating the effectiveness of the multi-scale structure by showing various LOD controls for image stylization

The comparisons with mean shift-based Edison system are presented in terms of segmentation quality and timing efficiency. Additionally statistical data of component trees constructed by the proposed algorithm is given.

We start by discussing related work in Section Ⅱ following by our algorithm in Section Ⅲ. Section Ⅳ presents hierarchical morphology for denoising and boundary smoothing. In Section V various effects of multi-scale representation are shown. Experimental results and discussions on the pros and cons of the proposed approach are presented along with future directions in Section Ⅵ. Finally we summarize the work in Section Ⅶ.

II. Related works

Image segmentation has a long history in computer vision[8]. One of the most prominent recent trends is to make objective the evaluation of segmentation results[9][10]. For it, Berkeley database was constructed and made publicly available[10]. It contains human annotations of segmentation. From th ese it is evident that there is a contradiction that human decision is not at all syntactic but semantic. But automated image segmentation techniques are usually achieved without such higher level semantic information. Here we review earlier work in multi-scale segmentation and some earlier attempts to segmentation using MSER.

1. Multi-scale image segmentation

The multi-scale segmentation stems from two seminal papers[4][9] that pave a mathematically sound foundation for scale-space theory and principle of tracking *intensity extrema* in 1-D signal[4] and 2-D images[3]. In [3], algorithmic principles handle events such as blob appearance, disappearance, splitting and merging. Based on this foundation, Lindeberg solved specific issues such as selecting scale and measuring blob significance[6]. Lifshitz interpreted scale space under Morse theory and applied algorithm to medical images[11]. Further research followed this early work. Petrovic recognized the error-proneness of linear Gaussian scale-space representation and proposed a nonlinear one which better retains the image structures[12]. Arbelaez presented a unifying framework for contour detection and image segmentation and built a nested region structure by merging adjacent regions with weak contours[9]. Chen presented multi-scale edge-based image segmentation method suitable for satellite multispectral images[13].

The common aspect of all the above approaches is that they are performed in two stages - an initial segmentation using multi-resolution images and then another process to link adjacent scales. This incurs computational inefficiency and is specially prone to error in the linking stage.

2. Image segmentation using MSER

MSER was designed to be a blob detector[7]. The mission of MSER is to generate stable and repeatable blob features. When compared to the algorithms which rely on *intensity extrema*, MSER is more stable because it stems from *stability extrema* (as explained in Section III).This stability of MSER has been shown to be superior to other techniques in performance benchmarking of local features[14].

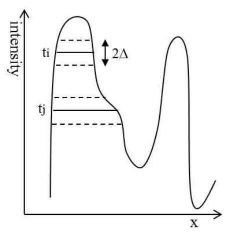
Inspired by these merits, some researches attempted to use MSER for image segmentation. Donoser proposed a color image segmentation algorithm which requires an user to indicate an ROI (region-of-interest) and uses it to order RGB values[15]. In another approach, the same authors choose the nested MSERs of a similar shape and take their boundary as final segmentation result[16]. Gui used MSER as initial segmentation tool and passed the results to spectral clustering process[17]. He applied the method to SAR (synthetic aperture radar) images.

However, all these above approaches do not address the use of multi-scale structure of MSER in designing algorithms in particular for image segmentation.

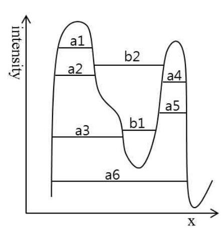
III. MSER-based multi-scale segmentation algorithms

An ER (extremal region) is defined as a connected component of pixels having higher intensity values than a threshold[7]. [Figure 1](a) shows a 1-D version of ER at two thresholds t_i and t_j . For an ER at t_i, we can define the stability as an inverse of the ratio of the change in area $\Psi(t_i)$ in Equation (1). The MSER is an ER with local maximum stability, i.e. minimumΨ. We call this property as *stability extremum*. For example, in [Figure 1], ER at ti is probable to be MSER while one at t_j is not.

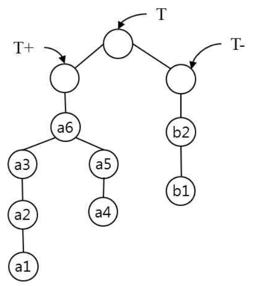
$$\Psi(t_i) = \frac{area_{i-\Delta} - area_{i+\Delta}}{area_i} \qquad (1)$$



(a) ER



(b) MSER



(c) Component tree

Figure 1. 1-D illustration of MSER+ and MSER-.

The MSERs are nested as illustrated in [Figure 1](b) and represented by a component tree shown in [Figure 1](c). It is usual to perform MSER detection twice, one for bright MSER+ and another one for dark MSER-, pointed by T_+ and T_- in the component tree, respectively. A collision between MSER+ and MSER- may happen, as exemplified by a_3 and b_2 in [Figure 1](b).

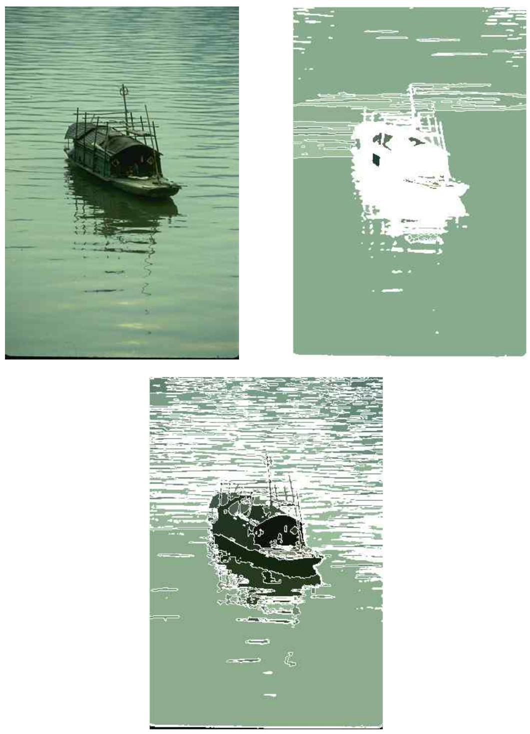


Figure 2. A natural image and segmentation results (top left: original, top right: MSER+, bottom: MSER-).

The nested structure provides multi-scale structure because we can choose any level of details by pruning the tree as we need. This property of MSER is the most powerful one from the perspective of our proposed image segmentation technique. A linear $O(n)$ algorithm to construct the component tree has been developed in [18] where n is the number of pixels. So image segmentation based on MSER can be expected to be very fast. We propose such a technique in the following section.

[Figure 2] shows a natural image taken from Berkeley database [10] and two segmentation results by MSER+ and MSER- trees. The white area indicates being not covered by MSERs.

1. Segmentation by component tree traversal

Our method is based on post-order traversal of the component tree that partitions the image. The pseudo code is as follows.

Algorithm 1: Image segmentation by post-order traversal of component tree

```
function postorder(t) {
    if(t is leaf node) partition(I, t);
    else {
        for(each of children, c of t) postorder(c);
        partition(I, t);
    }
}
```

In order to segment using MSER+, function call, postorder(T_+) is used. For MSER-, postorder(T_-) is used. The function partition(I, t) labels the pixels of image I belonging to the node t. Labeling is done only when the pixel has not been labeled before so that child node has a priority over parent node. For the residual parts not covered by MSERs(white area in middle and right images of [Figure 2]), assigning special label completes the image segmentation.

Inspection of the segmented images in [Figure 2] shows that MSER+ or MSER− alone cannot segment whole image. This is while MSER+ captures the regions of bright objects, MSER− captures the dark ones. Hence, both the MSER+ and MSER− component trees should be considered to get a complete image segmentation. To achieve this, we present a technique to perform postorder(T_+) and postorder(T_-) in the following section.

2. Core node−first traversal

For this, we first define a merger node and core node. A merger node is a node with more than or equal to two children. A *core* node is a node in the component tree that is between a leaf node and the first merger node encountered while moving upwards from the leaf node. In [Figure 1], a_6 is merger while others are core nodes. The regions of core nodes can be regarded as *photometrically salient* since they have distinctive intensities from surrounding regions and are called core regions. To find the core regions, we present another traversal algorithm which endows priority to the core nodes.

Algorithm 2: Image segmentation by core-first traversal

```
function image_segmentation ( ) {
    core_first(T+);
    core_first(T-);
    postorder(T+);
    postorder(T-);
}
function core_first(t) {
    if (t is leaf node) {
        repeat {
            partition(I, t);
            prune t;    // make it dead
            t=parent(t);
        } until(t is not merger node);
    }
    else
        for (each of children, c of t) core_first(c);
}
```

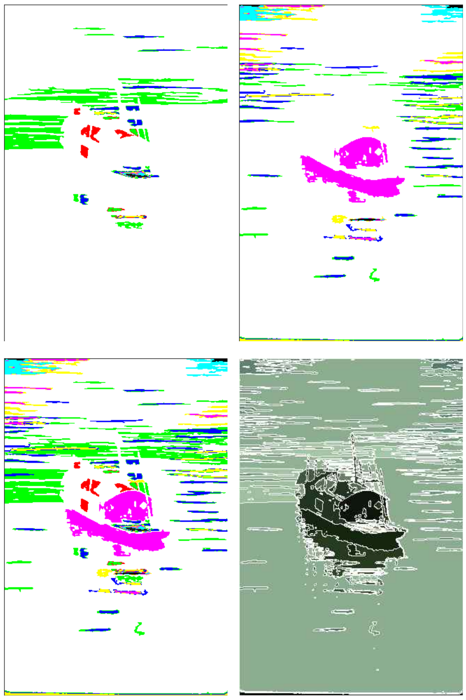


Figure 3. Core−first segmentation (top left: cores from MSER+, top right: cores from MSER−, bottom left: both, bottom right: results of Algorithm 2, Colors encoding node's depth in tree, red (depth 2), green(3), blue(4), ⋯).

Algorithm 2 applies the function core_first() for each of two trees successively, MSER+ and MSER−. Successive applications of postorder() for two trees complete the segmentation process. The pruned dead node is excluded in the process of postorder(). [Figure 3] shows the core regions thus deciphered from MSER+ and MSER−. For this example image, no collision has occurred between them. Observation over lots of images revealed that no or only a slight collision happens. Currently occupation of the collision zones is determined arbitrarily between MSER+ and MSER- by first-come-first-occupy rule. An elaboration can be made in the future, for example by exploiting edge information as [13].

IV. Hierarchical morphology

Like any other segmentation methods, our method also needs denoising and boundary smoothing. We consider morphological operations for this purpose. However since the segmentation result is not binary but encoded in a hierarchical manner with the child-parent relationship, standard operations are not readily applicable. Hence, we propose the following hierarchical morphology algorithm:

Algorithm 3: Child-first dilation

```
function postorder_dil(t) {
    if(t is leaf node) dilation(t);
    else {
        for(each of children, c of t) postorder_dil(c);
        dilation(t);
    }
}
function dilation(t) { // S: structuring element
    for(each pixel, p of node t)
        for(each pixel, q in S[p])
            if(q is ancestor of p) I[q]=I[p];
}
```

[Figure 4] explains the above dilation operation on a hierarchical map where node 6 is the parent of node 5 and node 7 is the parent of node 6. This example uses 3*3 structuring element. The net effect of this dilation is the expansion of children region into ancestor regions about one pixel wide.

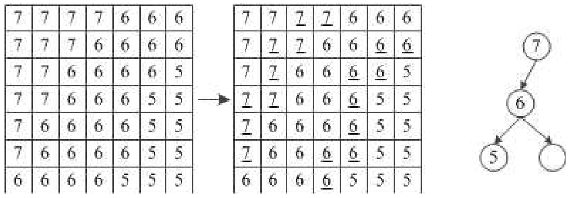


Figure 4. Hierarchical morphology
(The label of underlined pixels should be changed into their children's label).

We can easily change Algorithm 3 into a parent-first version as follows. Here the traversal is changed from post-order to pre-order.

Algorithm 4: Parent-first dilation

```
function preorder_dil(t) {
    if(t is leaf node) dilation(t);
    else {
        dilation1(t);
        for(each of children, c of t) preorder_dil(c);
    }
}
function dilation1(t) {
    for(each pixel, p of node t)
        for(each pixel, q in S[p])
            if(p is ancestor of q) I[q]=I[p];
}
```

Note that in function dilation1(), the condition of if statement has been changed by reversing roles of p and q. Now we may apply various combinations of C (Child-first) and P (Parent-first) such as C, CC, CP, CPC. [Figure 5] shows effects of CPC scheme. Experiments with many natural images showed that CPC scheme produced good results. So in the remainder of this paper, we will use CPC scheme.

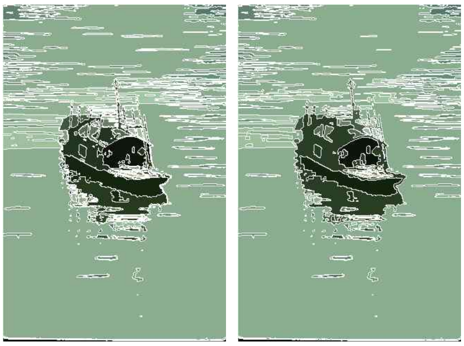


Figure 5. Results before (left) and after (right) hierarchical morphology.

V. Image tooning using multi-scale structure

The multi-scale structure embedded in the component tree can be used in a variety of purposes. In this section, we use this to present effects of image tooning[19] that uses a simple technique of assigning a region its average color. We illustrate this in [Figure 6]. The top left image is the original image from Berkeley database. Pruning nodes of surfer's body resulted in top right image in which details disappeared. Two images in bottom were produced by pruning nodes for sky and big waves.

Figure 6. Styling using multi-scale structures (top: original, the remainings: stylized).

[Figure 7] illustrates the use of the multi-scale structure for a stylization in which core regions and non-core regions are treated differently. [Figure 7](c)-(e) is an image where boundaries are emphasized by black lines. [Figure 7](d) was generated by decreasing intensities of non-core regions by multiplying by 0.8 while increasing the intensities of cores by multiplying by 1.2. To accomplish this RGB is first converted into CIELab color space and the multiplication is done for L (luminance) component. Then the image is converted back to RGB. The effect is contrast enhancement between core and non-core. [Figure 7](e) is of reverse effect by increasing intensities of non-core regions while decreasing the intensities of cores.

(a)

(b)

(c)

(d)

(e)

Figure 7. Contrast enhancing of core regions ((a) and (b): original and core regions (non-core region is white), (c): core 1.0 and non-core 1.0, (d): core 1.2 and non-core 0.8, (e): core 0.8 and non-core 1.2).

VI. Experimental results and discussions

All the programming is done in C++ under the OpenCV environment. The CPU is 1.7GHz Pentium chip and main memory is 4Giga bytes. All the images except the skating woman are obtained from Berkeley segmentation database. The skating woman image is from Wikimedia archive. The segmentation quality will be compared in terms of whether the multi-scale property is provided or not.

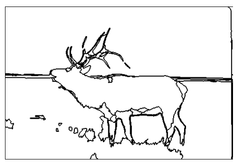
[Figure 8] compares the proposed algorithm with mean shift algorithm. We used the EDISON software which is an improvement of the mean shift[20]. A rigorous and objective comparison is out of scope of this paper. The purpose is to qualitatively identify the advantages and limitations of the proposed algorithm. The strongest property of the proposed algorithm is multi-scale functionality as illustrated in Section V. The face and legs of woman in top image and deer body of middle image of [Figure 8] reveal the multi-scale structure of the proposed algorithm clearly. A limitation is under-segmentation. A part of face of skating woman in bottom image is typical example. In the discussion of Section 2, we will address this issue.

original

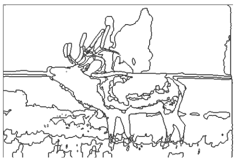
EDISON

proposed

original



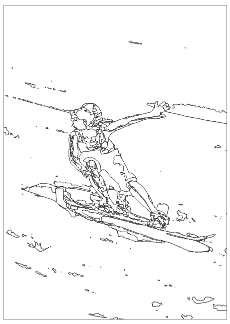
EDISON



proposed

original

EDISON



proposed

Figure 8. Comparison with mean shift algorithm

Table 1. Timing analysis and comparison
(segmentation + postprocessing in second)

	EDISON	Proposed
Boat (481*321)	1.68+0.16	0.21+0.19
Surfing (321*481)	1.72+0.05	0.17+0.31
Paprika (481*321)	1.47+0.08	0.36+0.36
Girl (481*321)	1.42+0.10	0.20+0.28
Deer (321*481)	1.50+0.11	0.25+0.24
Skating (599*428)	2.93+0.18	0.46+0.41

[Table 1] presents timing analysis of two algorithms. The proposed algorithm is about 2 to 4 times faster than EDISON. Two other algorithms, graph-cut and normalized-cut that are popularly used due to their performance and publicly available source codes were already evaluated to be slower than the mean shift by [8].

1. Statistics of component trees

Some statistics on the component trees are helpful in understanding the behaviors of the proposed algorithm and in gaining some insights for future works. [Table 2] illustrates the statistics.

Table 2. Statistics of component trees

	Boat	Surfing	Paprika
#nodes[1]	56/123	59/27	196/116
#leaf/core[2]	92/164	31/67	143/265
depth[3]	8/3.0	9/4.5	13/5.3
#children[4]	40/6.7	5/2.6	27/4.1
size[5]	64/141181/ 2349	65/142140/ 6001	63/132834/ 3415

1: number of MSER+ nodes/MSER- nodes
2: number of leaf/core
3: maximum of depth/average
4: maximum number of children/average computed only for merger
5: minimum area/maximum/average measured in number of pixels

[Table 2] showed that tree depth is quite large (around 10) and the trees are rather skewed. The leaf nodes are 30~50% of total set of nodes and the core nodes are 80~90%. As seen in [Figure 3], core nodes are smaller than non-cores.

2. Discussions

The main theme of this paper is to extend MSER as a blob detector to image segmentor. In spite of good properties of MSER, some limitations still exist. We summarize them along with important future directions.

- MSER uses only gray-scale information and hence the under-segmentation. Though a color MSER is available [21], experiments showed that color version is much inferior in boundary localization. Color ordering which is appropriate for the image segmentation purpose is one of the critical future works. Though it is restricted to skin color, approach of color ordering using

Mahalanobis distance in [22] can be considered as an attempt of such color ordering.

- The proposed algorithm has no process for optimizing boundary localization accuracy and smoothness. New formulation of MSER extraction by embedding edge-guided boundary tuning [13] and/or smoothness prior would make MSER to be utilized very successfully as a true image segmentor. The revision should be done in the realm of retaining MSER's nature of seeking stability extrema and multi-scale structure.
- The proposed algorithm is expected to be advantageous in video segmentation due to its nested structure and stability. Constructing and segmenting 3-D volumes ((x,y) for position and t for time) using MSERs is an important future work. For this purpose, faster and more robust MSER tracking algorithm proposed in [23] can be used.

VII. Conclusions

The paper described an approach to extend MSER's blob detection functionality to the image segmentation. Appropriate traversal algorithms of component trees were presented and evaluated. The results revealed both the possibilities and limitations. We validated the effectiveness of multi-scale structure by showing various LOD control for image tooning. The paper emphasized that this was achieved without multi-level Gaussian smoothing.

참 고 문 헌

[1] K. A. Tran and G. Lee, "Text segmentation from images with various light conditions based on Gaussian mixture model," International Journal of Contents, Vol.9, No.1, pp.1-5, 2013.

[2] I. S. Na, K. H. Oh, and S. H. Kim, "Unconstrained object segmentation using GrabCut based on automatic generation of initial boundary," International Journal of Contents, Vol.9, No.1, pp.6-10, 2013.

[3] J. J. Koenderink, "The structure of images," Biological Cybernetics, Vol.50, pp.363-370, 1984.

[4] A. P. Witkin, "Scale space filtering," IJCAI, pp.1019-1023, 1983.

[5] D. DeCarlo and A. Santella, "Stylization and abstraction of photographs," SIGGRAPH, pp.769-776, 2002.

[6] T. Lindeberg, "Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method of focus-of-attention," International Journal of Computer Vision, Vol.11, No.3, pp.283-318, 1993.

[7] J. Matas et al. "Robust wide baseline stereo from maximally stable extremal regions," British Machine Vision Conference, pp.384-396, 2002.

[8] F. J. Estrada and A. D. Jepson, "Benchmarking image segmentation algorithms," International Journal of Computer Vision, Vol.85, pp.167-181, 2009.

[9] P. Arbelaez, "Contour detection and hierarchical image segmentation," IEEE Tr. PAMI, Vol.33, No.5, pp.898-916, 2011.

[10] D. Martin, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," ICCV, pp.1-8, 2001.

[11] L. M. Lifshitz and S. M. Pizer, "A multiresolution hierarchical approach to image segmentation based on intensity extrema," IEEE

Tr. PAMI, Vol.12, No.6, pp.529-540, 1990.

[12] A. Petrovic, "Multiresolution segmentation of natural images: from linear to nonlinear scale-space representations," IEEE Tr. Image Processing, Vol.13, No.8, pp.1104-1114, 2004.

[13] J. Chen, "Edge-guided multiscale segmentation of satellite multispectral imagery," IEEE Tr. Geoscience and Remote Sensing, Vol.50, No.11, pp.4513-4520, 2012.

[14] K. Mikolajczyk,, "A comparison of affine region detectors," International Journal of Computer Vision, Vol.65, No.1/2, pp.43-72, 2005.

[15] M. Donoser, H. Bischof, and M. Wiltsche, "Color blob segmentation by MSER analysis," IEEE International Conference on Image Processing, pp.757-760, 2006.

[16] M. Donoser, H. Riemenschneider, and H. Bischof, "Linked edges as stable region boundaries," IEEE International Conference on Computer Vision and Pattern Recognition, pp.1665-1672, 2010.

[17] Y. Gui, X. Zhang, and Y. Shang, "SAR image segmentation using MSER and improved spectral clustering," EURASIP Journal on Advances in Signal Processing, Vol.2012, No.1, pp.1-9, 2012.

[18] D. Nister and H. Stewenius, "Linear time maximally stable extremal regions," ECCV, pp.183-196, 2008.

[19] J. E. Kyprianidis, et al, "State of the art: a taxonomy of artistic stylization techniques for images and video," IEEE Tr. Visualization and Computer Graphics, Vol.19, No.5, pp.866-885, 2012.

[20] C. M. Christoudias, B. Georgescu, and P. Meer, "Synergism in low level vision," International Conference on Pattern Recognition, Vol.4, pp.150-155, 2002.

[21] P. E. Forssen, "Maximally stable color regions for recognition and matching," IEEE International Conference on Computer Vision and Pattern Recognition, pp.1-8, 2007.

[22] E. Murphy-Chutorian and M. Trivedi, "N-tree disjoint-set forests for maximally stable extremal regions," British Machine Vision Conference, pp.739-748, 2006.

[23] M. Donoser and H. Bischof, "Efficient maximally stable extremal region (MSER) tracking," IEEE International Conference on Computer Vision and Pattern Recognition, pp.553-560, 2006.

저 자 소 개

이 진 선(Jin-Seon Lee) 정회원

- 1985년 : 전북대학교 전산통계학과(이학사)
- 1995년 : 전북대학교 전자계산기공학과 박사
- 1995년 3월 ~ 현재 : 우석대학교 게임콘텐츠학과 교수

<관심분야> : 멀티미디어, 패턴인식

오 일 석(Il-Seok Oh) 정회원

- 1984년 : 서울대학교 컴퓨터공학과(공학사)
- 1992년 : KAIST 전산학과 박사
- 1992년 9월 ~ 현재 : 전북대학교 컴퓨터공학부 교수

<관심분야> : 컴퓨터비전, 패턴인식