Real-Time Superpixel Segmentation by DBSCAN Clustering Algorithm

Jianbing Shen, *Senior Member, IEEE*, Xiaopeng Hao, Zhiyuan Liang, Yu Liu, Wenguan Wang, and Ling Shao, *Senior Member, IEEE*

***Abstract*— In this paper, we propose a real-time image superpixel segmentation method with 50 frames/s by using the density-based spatial clustering of applications with noise (DBSCAN) algorithm. In order to decrease the computational costs of superpixel algorithms, we adopt a fast two-step framework. In the first clustering stage, the DBSCAN algorithm with color-similarity and geometric restrictions is used to rapidly cluster the pixels, and then, small clusters are merged into superpixels by their neighborhood through a distance measurement defined by color and spatial features in the second merging stage. A robust and simple distance function is defined for obtaining better superpixels in these two steps. The experimental results demonstrate that our real-time superpixel algorithm (50 frames/s) by the DBSCAN clustering outperforms the state-of-the-art superpixel segmentation methods in terms of both accuracy and efficiency.**

***Index Terms*— Real-time, superpixel, DBSCAN, segmentation.**

I. Introduction

AS AN important preprocessing stage of many applications in the field of computer vision and image processing, superpixels generation has attracted substantial attention during the last decade. The superpixel concept was originally presented by Ren and Malik [1] as the perceptually uniform regions using the normalized cuts (NCuts) algorithm. Superpixels are clusters of pixels which share similar features, thus they can be used as mid-level units to decrease the computational cost in many vision problems, such as image/video segmentation [3]–[7], [10], [15], [40], saliency [19], [31], [36], [38], [39], [45], tracking [8], [20], [21], [37], classification [9], [11], object detection [12], [41], [46], motion estimation [14], reconstruction [16], [44], and other vision applications [13], [26], [43].

Manuscript received March 27, 2016; revised July 28, 2016 and September 29, 2016; accepted October 5, 2016. Date of publication October 11, 2016; date of current version October 28, 2016. This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2013CB328805, in part by the National Natural Science Foundation of China under Grant 61272359, in part by the Fok Ying-Tong Education Foundation for Young Teachers, and in part by the Specialized Fund for Joint Building Program of Beijing Municipal Education Commission. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Peter Tay.

J. Shen, X. Hao, Z. Liang, Y. Liu, and W. Wang are with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: shenjianbing@bit.edu.cn).

L. Shao is with the Department of Computer and Information Science, Northumbria University, Newcastle upon Tyne, NE1 8ST, U.K. (e-mail: ling.shao@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2016.2616302

There are many popular superpixel approaches such as normalized cut [17], SLIC [18], LSC [42], ERS [34], SEEDS [22], mean shift [23], Turbo-pixel [24], graphcuts [25], and pseudo-boolean optimization (PB) [32]. Each algorithm has its own advantage and disadvantage for superpixel segmentation, however, it is still very challenging to develop a high quality and real-time superpixel algorithm that exhibits the properties including good boundary adherence, compact constraints, regular shapes and low computational complexity. In order to satisfy these desired requirements, we propose a real-time superpixel segmentation algorithm by Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [27] to achieve better performance than state-of-the-art methods. This is the first time that the DBSCAN clustering algorithm is applied to superpixel segmentation.

Superpixels are used to replace pixels for a more compact visual representation together with fast computation. As an important preprocessing step of a large number of image processing applications, its computational cost is the mostly concerned issue. Among these superpixel algorithms, the SLIC algorithm becomes popular, since it can produce superpixels quickly without sacrificing much of the segmentation accuracy. But there is still much room for the improvement of superpixel in computational cost and adherence to boundaries.

A desired superpixel method needs to not only fulfil the requirement of good boundary adherence, but also be efficient. Since the superpixels are used as a preprocessing step in vision applications, the algorithm of high-quality superpixels with less computation is preferred. In this paper, we propose a new real-time superpixel method using DBSCAN [27] clustering, which inherits the advantages of existing superpixel algorithms (such as SLIC) and further carries forward beyond these advantages. Our DBSCAN superpixel segmentation algorithm is not only more efficient but also more accurate than previous superpixel algorithms.

In our paper, we bring the DBSCAN clustering algorithm to the generation of superpixels. Density-based spatial clustering of applications with noise is a data clustering algorithm proposed by Martin *et al.* [35]. It is a density-based clustering algorithm. Since DBSCAN can find arbitrarily shaped clusters, it has a good potential to segment complex and irregularly shaped objects. In order to produce regular superpixles, it is necessary to introduce extra geometric restrictions on the DBSCAN clustering algorithm. The regularity is also an important criterion for evaluating the performance of superpixels. In our superpixel algorithm, the geometric constraint

(a) (b)

Fig. 1. Results by our real-time DBSCAN superpixel method. Here the numbers of superpixels are about 250, 450, and 900 in the regions separated by two white parallel lines.

obtains the regular shape by restricting the searching paths and searching range.

As mentioned before, we adopt two fast stages in our algorithm. First, we get the initial superpixels by clustering pixels using only color and geometric restrictions by the DBSCAN cluster algorithm. In this stage, our distance function contains two items - the seed distance item and the neighbor distance item. The seed distance item ensures the pixels contained in the same superpixel should be similar. The neighbor distance item has more influence in weak boundary and flat regions. And then we merge the small initial superpixels with their neighbor superpixels through a measurement of color and spatial information. Both DBSCAN clustering and the merging procedure have low computational complexity. Our algorithm can also handle the situation of twisted objects to form the final regular superpixels. Fig. 1 gives some superpixel results with vary numbers of DBSCAN, and more quantitative results about boundary adherence and efficiency will be demonstrated in the later experiments section. Our source code will be publicly available at.[1]

The main contributions of this paper are twofold:

- We propose a real-time superpixel algorithm based on the DBSCAN clustering, achieving the state-of-the-art performance at a substantially reduced computational cost.
- Our proposed method has good performance in adherence of boundaries, even for complex and irregular objects in images that state-of-the-art superpixel algorithms cannot handle.

II. Related Work

In order to produce satisfying superpixels, two types of superpixel segmentation methods have been proposed in the past few years, and we briefly review them in this section. The first class is based on clustering, including the normalized cuts [17], SLIC [18], LSC [42], SEEDS [22] and Turbo [24]. In this class, superpixels are regarded as clusters of pixels. As a result, many clustering methods can be employed to produce different superpixel results. Shi and Malik [17] use the normalized cut as the clustering method to produce homogeneous superpixels. However, the normalized cut is computationally expensive, since it is a two-way clustering method and each iteration of the calculation can only produce one superpixel. Achanta *et al.* [18] change the clustering process to the k-means algorithm, which is quite efficient. Therefore, this method becomes the most popular superpixels preprocessing step in many applications. The main shortcoming of the SLIC is that it only uses the color and coordinates of each pixel as features, therefore their superpixels cannot adhere to the boundaries of objects well, especially when superpixels are not small enough. Li and Chen [42] adopt linear spectral clustering (LSC) to extract superpxiels. The LSC combines the advantages of normalized cut and k-means to get good quality superpixels. However, due to the calculations of features, the LSC is pretty slow as a preprocessing method. Levinshtein *et al.* [24] generate highly uniform lattice-like superpixels by iteratively dilating regularly distributed seeds. But, the generated superpixels present relatively low adherence to boundaries and high computational complexity, due to the stability and efficiency issues of the level-set method.

The other class is based on optimization, such as graphcuts [33], lattice cut [28], entropy rate (ERS) [34], pseudo-boolean optimization (PB) [32] and lazy random walk (LRW) [2]. Most of these methods use the graphcut as the basic optimization framework. Veksler *et al.* [33] introduced the use of both a data term and a smoothness term that are typical in graph-cut based optimization. Zhang *et al.* [32] further add more constraints to generate superpxiels between two horizontal and vertical strips. Then, their superpixels are regular and square-like, and cannot align well in some places. Shen *et al.* [2] propose a method using LRW to obtain superpixel segmentation results. The LRW algorithm is to get the probabilities of each pixel from the input image, and utilizes the probabilities and the commute time to get initial superpixels. Then their method introduces an energy function to optimize iteratively the initial superpixels, which is related to the texture measurement and the commute time. However, the LRW superpixel algorithm is very expensive, and it usually costs several seconds to generate the final superpixels. Liu *et al.* [34] formulated the superpixel segmentation problem as an objective function that consists of the entropy rate (ERS) of a random walk on a graph and a balancing term which encourages the generation of superpixels with similar sizes. The entropy rate can help to cluster the compact and homogeneous regions, which also favors the superpixels to overlap with a single object on the perceptual boundaries. However, the irregular shape of ERS superpixels may become a potential drawback in future application.

In addition, another important related work with our paper is the DBSCAN algorithm and its application in image processing [29], [30]. Density-based spatial clustering of applications with noise (DBSCAN) is a data clustering algorithm proposed by Martin *et al.* [35]. Given a set of points in some space, it groups points that are closely packed together (points with

[1] http://github.com/shenjianbing/realtimesuperpixel

many nearby neighbors), marking as outliers points that lie alone in low-density regions whose nearest neighbors are too far away. DBSCAN is one of the most common clustering algorithms and also the most cited algorithm in the leading data mining literature. Manavalan and Thangavel [29] use DBSCAN clustering for transrectal ultrasound image segmentation. The ultrasound images usually have poor image quality, such as low contrast, speckle noise, and weak boundaries, and it is very difficult to segment them by conventional clustering approaches. However, the DBSCAN algorithm is quite suitable to detect and segment most of these important regions. Their segmentation procedure can extract the prostate region efficiently and accurately through a series of experiments and evaluation. Hou *et al.* [30] propose a DSets-DBSCAN algorithm framework for image segmentation, which overcomes the drawback of the original DBSCAN clustering algorithm that is sensitive to similar measures and requires appropriate parameters to generate satisfactory clustering results. DSets-DBSCAN is able to generate clusters of arbitrary shapes and determine the number of clusters automatically.

III. DBSCAN Superpixels

In this section, we give the details of our simple algorithm framework to generate superpixels that is not only faster than existing well-known algorithms (LSC [42], SLIC [18] and SEED [22]), but also exhibits better boundary adherence. The goal of superpixels is to cluster pixels with a homogenous appearance from an image into small, compact regions. The superpixel segmentation can be considered as a clustering problem where each superpixel contains a unique feature in color and shape. The DBSCAN is a density-based clustering algorithm, and it will improve the performance of segmentation algorithms by adding geometric constraints. The proposed method includes two stages - a clustering stage and a merging stage. Firstly, we aggregate pixels to get initial superpixels by the DBSCAN algorithm. Secondly, these initial superpixels are refined to obtain final superpixel results through merging very small superpixels. The entire algorithm is summarized as follows.

In the clustering stage, we define two sets as labeled set and candidate set, respectively, then assign the top-left pixel a label for the first seed and add it into the labeled set. Now we have three kinds of pixels, i.e., the seed, labeled pixels and unlabeled ones. Firstly, we find all of the unlabeled four neighboring pixels of the labeled set, then calculate the combination distance between each unlabeled pixel of its center pixel (the unlabeled pixel is generated by the center pixel) and the seed - if the distance is less than the threshold we defined, we put it into the candidate set. Secondly, we update the labeled set through replacing it by the candidate set and give them the same label with the seed. We repeat the two steps until the termination condition is satisfied.

As shown in Fig. 2, the terminate conditions include two aspects. The first one is that the number of pixels in this cluster is more than a threshold S/N, where S represents the size of the input image, N denotes the number of superpixels that the user needs. This condition is used to control the size of the cluster. The second one is the labeled set that becomes

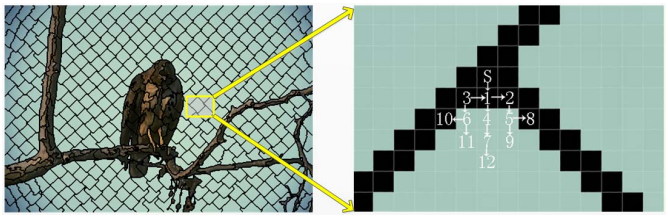
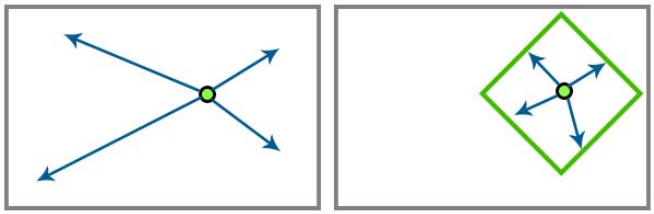


Fig. 2. Illustration of DBSCAN clustering process. The strategy of the proposed superpixel generation by DBSCAN clustering is described as the right sub-figure, where each grid in above figure represents a pixel. The searching strategy is similar with Breadth-First-Search method, where pixel S is the seed that expand the labeled set until the termination conditions are satisfied. The right sub-figure (e.g., pixels S, 1, 2, 3) demonstrate the detailed illustration about this iterative process.



(a) standard DBSCAN searches the entire image (b) DBSCAN searches a limited region

Fig. 3. Illustration of the searching space. (a) is the searching space of the conventional DBSCAN algorithm [27], which contains the whole image. (b) is the searching space of our algorithm. The black dot is a seed, and the rhombus region is the corresponding limited space. Moreover, if the seed locates at a flat region, the final superpixel will be a rhombus region.

an empty set. It indicates that the neighboring pixels of the previous labeled set will locate at the boundary regions, and this condition guarantees the algorithm to stop at the boundary. Then we select a new seed with a conventional order from the unlabeled pixels and repeat the above procedure until all the pixels are labeled. This conventional order is from left to right and from top to bottom for selecting the seeds. Finally, initial superpixels are produced after getting enough clusters with different labels, which can be viewed as initial superpixels. In each initial superpixel, pixels are similar in color through the distance constraint.

Our searching strategy in clustering stage is different from the conventional DBSCAN algorithm [27]. As shown in Fig. 3, the searching range of the original DBSCAN algorithm is the whole image, while our algorithm limits the searching range in the rhombus neighbor region around the seed. This local searching strategy will greatly reduce the computational complexity and also make each superpixel with a uniform shape as possible.

After the clustering, we obtain the initial superpixels $L_{(p)}$ whose boundaries align the edges of objects well in most cases. Fig. 4 gives an example of the fragments (very small initial superpixels) at some edges of objects. In our method, the clustering stage generates relatively small superpixels and fragments, and the merging stage is used to merge the initial superpixels and eliminate the small fragments produced in the clustering stage, as shown in Fig. 4. The reason of these fragments generation is the usage of distance between pixels, which is sensitive to the local color features. Therefore, we

Fig. 4. Illustrating superpixel segmentation results in different stages. The left two images are the results after the proposed clustering stage, and the right two images are the final superpixel results by the merging stage.

perform the merging stage to refine the initial superpixels by eliminating those fragments. There are two strategies for eliminating the fragments. One is to add a small fragment into its neighbor to form a larger superpixel, and the other is to merge two neighbor fragments. Here we introduce a superpixel distance to control the merging strategy. This distance combines color distance and spatial distance, which are described in the next subsection in detail.

In the merging stage, all the initial superpixels are processed in a similar order as in the first stage, which is from left to right and from top to bottom. If the number of pixels in one initial superpixel is less than a threshold, we will merge another initial superpixel from its shortest distance neighbor into this one. After the merging stage, we can obtain the final refined superpixels with the regular shapes. These two stages can be formulated as

$$L(p) = \{i \in I \mid D_1^p(i, j) < \psi, i \sim j, j \in P\} \cup L(p) \tag{1}$$

where p is a superpixel, ψ is a threshold (distance constraint), $i \sim j$ indicates pixel i around pixel j, j is in labeled set L(p), and $D_1^p(i, j)$ is the distance between pixels i and j. (1) is used to produce the initial superpixels during the clustering stage.

$$\begin{aligned} L(i) &= \text{argmin} D_2(p, i), \ i \sim p \\ L_R(p) &= L(p) \cup L(i) \end{aligned} \tag{2}$$

We use the above equation in the merging stage, where $D_2(L(p), L(i))$ is the distance between initial superpixels p and i. More details are shown in (4). Our DBSCAN superpixel segmentation is a clustering algorithm by comparing the distance between pixels in the RGB color space. The distance metric is an important step, and we define two distance metric functions in clustering and merging stages. In the clustering stage, we generate a linear combination function $D_1^k(i, j)$ by integrating two Euclidean metric functions $d_n(i, j)$ and $d_s^k(i, k)$, where $d_n(i, j)$ is the distance between two adjacent pixels in RGB color space and $d_s^k(i, k)$ is the distance between an unlabeled pixel i and a seed k. As $d_n(i, j)$ is simple but not robust enough especially in the weak boundaries, we improve this boundary discrimination ability by adding $d_s^k(i, k)$ and enhancing the superpixel internal consistency. The distance $D_1^k(i, j)$ is defined as

$$\begin{aligned} d_n(i, j) &= \sqrt{(R_i - R_j)^2 + (G_i - G_j)^2 + (B_i - B_j)^2} \\ d_s^k(i, k) &= \sqrt{(R_i - R_k)^2 + (G_i - G_k)^2 + (B_i - B_k)^2} \\ D_1^k(i, j) &= \alpha_1 d_s^k(i, C_k) + \alpha_2 d_n(i, j) \end{aligned} \tag{3}$$

where R_i, G_i, B_i represent the RGB color features of pixel i, α_1, α_2 are the weight parameters, and $\alpha_1 + \alpha_2 = 1$.

In the merging stage, the distance function D_2 is applied to calculate the distance between the initial superpixels. The function combines color distance and spatial distance. A good superpixel should be measured by not only the performance of adhering to the boundaries of superpixels to the edges of objects, but also the capability of maintaining the uniform size of superpixels. In our method, the color difference is used to determine whether two initial superpixels should be merged into one or not, and the spatial distance guarantees the regularity of final superpixels. Thus, the merging distance D_2 is defined as

$$\begin{aligned} d_c(l, p) &= \sqrt{(\bar{R}_l - \bar{R}_p)^2 + (\bar{G}_l - \bar{G}_p)^2 + (\bar{B}_l - \bar{B}_p)^2} \\ d_a(l, p) &= \sqrt{(\bar{x}_l - \bar{x}_p)^2 + (\bar{y}_l - \bar{y}_p)^2} \\ D_2(l, p) &= d_c(l, p) + \alpha_3 d_a(l, p) \end{aligned} \tag{4}$$

where $d_c(l, p)$ represents the color distance between two initial superpixels l and p, and ensures a high degree of similarity, $d_a(l, p)$ is the spatial distance that has a large influence on the shape of final superpixel results. $\bar{R}_p$, $\bar{G}_p$, and $\bar{B}_p$ are the average RGB color values for initial superpixel p, $\bar{x}_p$ and $\bar{y}_p$ are the coordinates of the centroid in superpixel p. The parameter α_3 in (4) is a positive coefficient for balancing the relative influence between $d_c(l, p)$ and $d_a(l, p)$.

Finally, we summarize these two main stages in Algorithm 1 and Algorithm 2.

IV. Experimental Results

We evaluate the performance of the proposed DBSCAN superpixel algorithm by comparing with state-of-the-art algorithms, including SLIC [18], PB [32], Ncuts [17], Tubopixel [24], ERS [34], SEEDS [22], LRW [2] and LSC [42]. Their results are generated by running publicly available implementations provided by the original authors. All the experiments are performed on the Berkeley Segmentation Database (BSD) [35], which consists of five hundred 321×481 images, together with human-annotated ground truth segmentations. We will demonstrate the effectiveness of the proposed method by first giving the visual comparison results of those methods and then providing a detailed quantitative comparison. In our experiments, we set the default parameters $\alpha_1 = 0.6$, $\alpha_2 = 0.4$, $\alpha_3 = 1.0$ and $\psi = 30$ to obtain a balanced good performance.

A. Visual Comparison

In order to obtain a fair comparison on superpixel quality, we run all of the state-of-the-art algorithms by using their suggested parameters to generate the optimal results. Fig. 5 gives

Algorithm 1 Initial Superpixel by DBSCAN Clustering

Input: Image I, superpixel p, threshold ψ, labeled set L_{set} and candidate set C_{set};
Output: Initial superpixel label $L(p)$;

```
Set initial pixel label is 0 for each image in I;
while each pixel have a new label do
    find a seed k;
    set k ∈ L_set;
    while L_set is empty or the number of pixels in p is larger than the threshold S/N do
        for each pixel j in L_set do
            for each pixel i around pixel j do
                compute the clustering distance D_1^k(i,j) with seed k and j.
                if D_1^k(i,j) < ψ then
                    set i ∈ C_set;
                end if
            end for
        end for
        set L_set = C_set;
    end while
end while
```

Algorithm 2 Superpixel Refinement by Merging Optimization

Input: Initial superpixel label $L(p)$;
Output: Refined superpixel label $L_R(p)$;

```
Set distance d(p) = 100000 for each superpixel p.
if the number of pixels in p is less than S/N then
    for each superpixel l around p do
        compute the merging distance D_2(l,p) between p and l;
        if D_2(l,p) < d(p) then
            set d(p) = D_2(l,p);
            set label j = l;
        end if
    end for
    merge two superpixels between j and p;
end if
```

the representative visual superpixel results generated by SLIC, LSC, ERS and our algorithm. It is obvious that our method obtains a better performance of image edges adherence than the other methods. This is because that our algorithm considers color information to detect the image boundaries more accurately than other algorithms. Our superpixel algorithm also produces compact and regular superpixels through the proposed extra geometric shape constraints. The ERS (Fig. 5, the third row) has the worst performance in compact and regular shape of superpixel compared with the other algorithms. The superpixel results by SLIC (Fig. 5, the top row) maintains the best performance for regular sizes of superpixels, but the adherence of boundaries is worse than the superpixel results by our algorithm and LSC (Fig. 5, the second row). Since the LSC algorithm combines the advantages of normalized cut and k-means to improve the performance of superpixels, it achieves the good property of compactness for the shape of superpixels, but LSC cannot handle the boundaries well, such as the incorrect boundaries of superpixels in the tree region (Fig. 5, the second column). The existing superpixel algorithms (e.g. SLIC and ERS) usually produce small regular superpixel results, and it will be difficult to achieve good performance of regularity and edge coherence. In contrast, our DBSCAN clustering superpixel algorithm achieves the best real-time performance, including the better boundary adherence and regular shapes (Fig. 5, the bottom row).

B. Quantitative Comparison

One of the most important requirements for superpixels is to maintain its adherence of object boundaries. Therefore, we adopt three commonly used evaluation metrics in superpixel segmentation for measuring the quality of boundary adherence: under-segmentation error (UE), boundary recall (BR) and achievable segmentation accuracy (ASA).

Boundary recall is an important metric for measuring the performance of adherence of boundaries in superpixel algorithms. It measures what fraction of the ground truth edges falls within at least two pixels of a superpixel boundary. A high BR means that very few true boundaries are missed. We use a standard measure of boundary recall (BR) [33], [34] to evaluate the performance. As shown in Fig. 6(b), it is apparent that our superpixel DBSCAN algorithm outperforms the other eight algorithms from the lower superpixel densities to the higher superpixel densities for the BR measurement. PB and Turbopixel give the worst performance compared with other algorithms, and LSC and ERS produce similar performance as our method.

Achievable segmentation accuracy (ASA) computes the highest achievable accuracy of labeling each superpixel with the label of ground truth that has the biggest overlap area. ASA is calculated as the fraction of labeled pixels that are not leaking from the ground truth boundaries. A high ASA means that the superpixels comply well with objects in the image. The ASA of each algorithm is calculated by averaging the values of ASA across all of the images in BSD [35]. Fig. 6(c) plots the average ASA result values of 500 images against the number of superpixels, and our method outperforms the other eight algorithms. Our DBSCAN superpixel approach generates the most correct overlap regions with the same label between ground truth segmentations and superpixel results.

Under-segmentation error (UE) measures the percentage of pixels that leak from the ground truth boundaries [2]. A good superpixel algorithm should try to avoid the under-segmentation areas in the segmentation results. In other words, we need to protect that a superpixel only overlaps with one object. A lower UE indicates that fewer superpixels cross multiple objects. As shown in Fig. 6(a), the UE curves are the average values for all 500 images in BSD. With the increase of superpixel numbers, our DBSCAN method demonstrates better performance. In addition, our DBSCAN method has better performance than SLIC in each number of superpixels, because SLIC generates superpixels without a compact constraint term.

Considering the aforementioned three metrics comprehensively, our DBSCAN superpixel method achieves the best performance among the state-of-the-art superpixel algorithms,

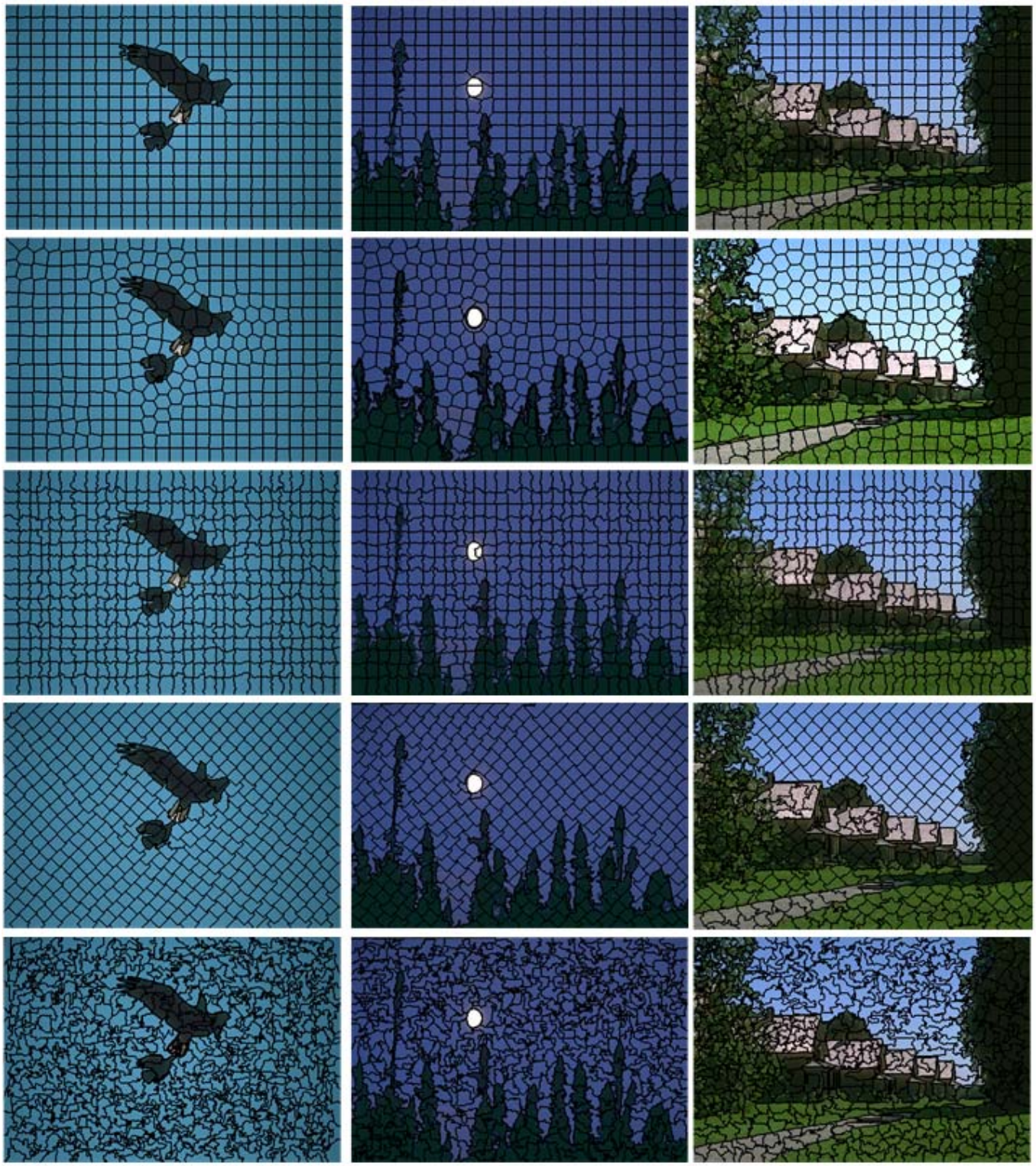
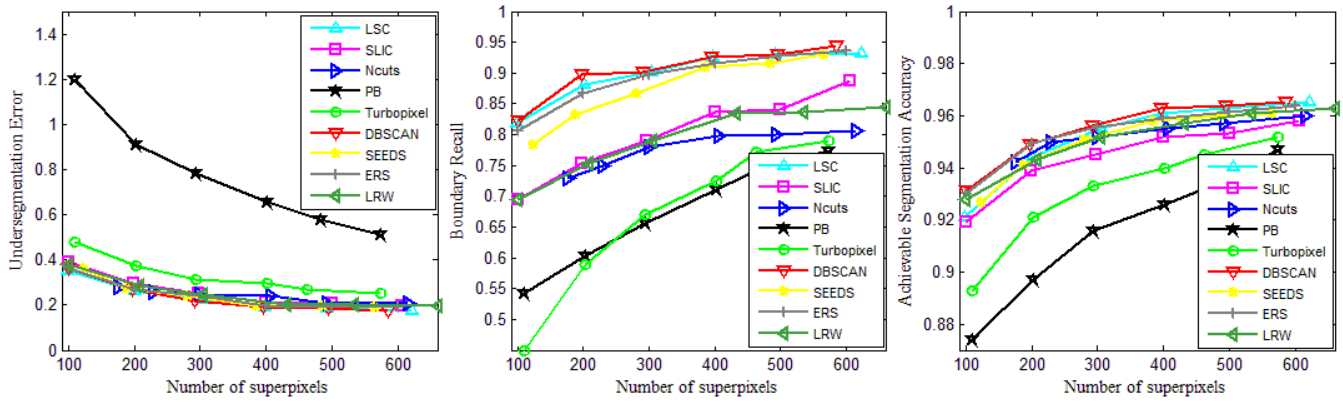


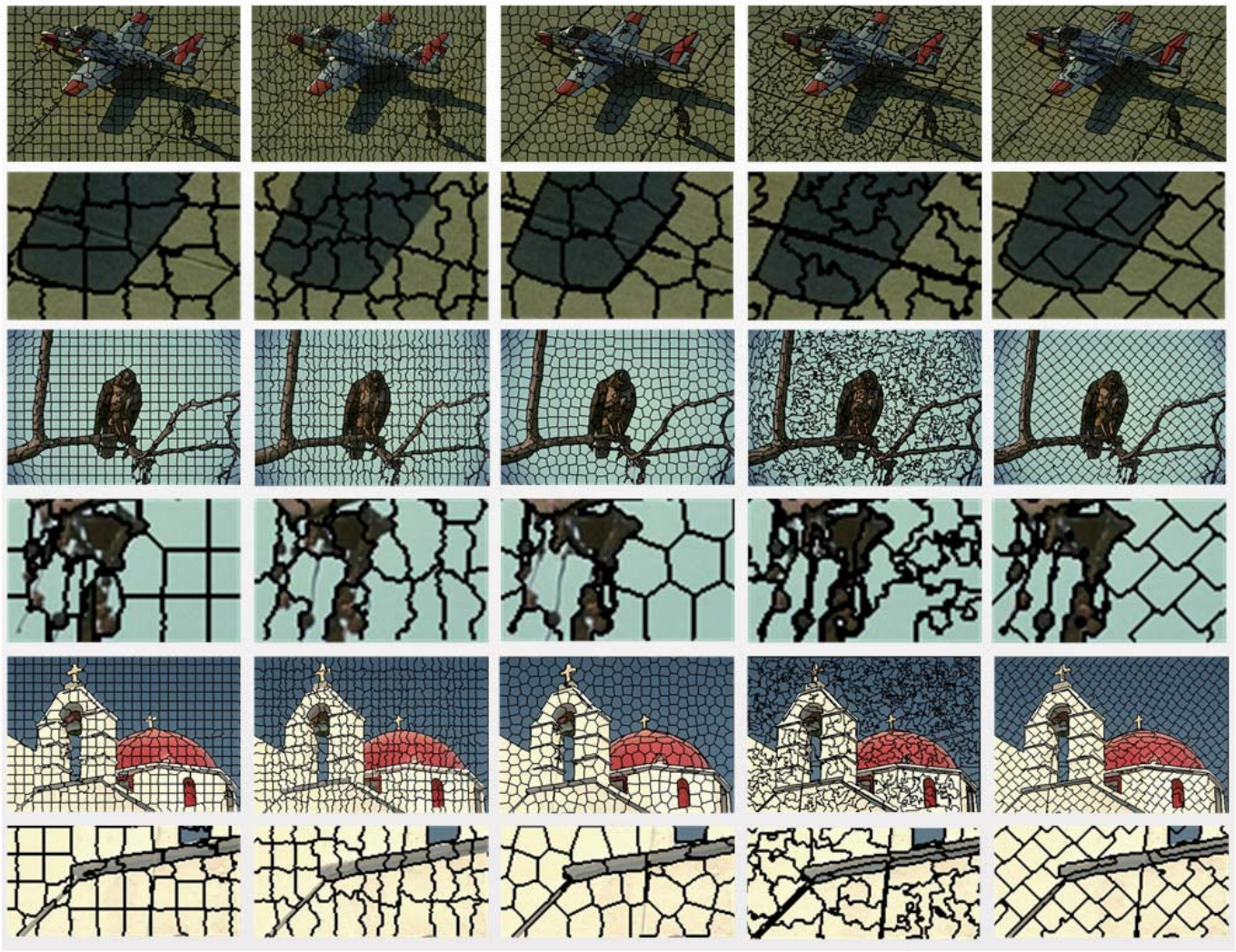
Fig. 5. Visual comparison of superpixel segementation results when the number of superpixels is 500. From top to bottom, the results are obtained by SLIC [18], LSC [42], ERS [34] and our DBSCAN method, respectively.

as shown in Fig. 6. Tubopixel and PB show the worst performance in terms of boundary adherence among all tested algorithms. LSC uses spectral clustering with an iterative weighted K-means clustering process and incorporates a local feature into a global objective optimization framework. Thus, it gives better boundary adherence than SLIC, but it costs more computational time. ERS and SEEDS have similar boundary adherence performance with LSC by sacrificing the regularity and perceptual satisfaction, and SEEDS is the fastest superpixel segmentation algorithm before our method.



(a) (b) (c)

Fig. 6. Quantitative evaluation on BSDS500: (a) the curves of UE; (b) the curves of BR; (c) the curves of ASA. Our real-time DBSCAN algorithm performs better than the state-of-the-art algorithms (LSC [42], SLIC [18], Ncuts [17], PB [32], Tubopixel [24], SEEDS [22], ERS [34] and LRW [2]).



(a) SLIC [15] (b) PB [26] (c) LSC [16] (d) ERS [28] (e) our DBSCAN

Fig. 7. Visual comparison of superpixel segmentation results by SLIC [18], PB [32], LSC [42], ERS [34] and our DBSCAN method. The number of superpixels is about 500.

C. More Superpixel Comparisons

Fig. 7 shows more visual comparison results between our DBSCAN algorithm and the state-of-the-art algorithms such as SLIC [18], LSC [42], ERS [34], and PB [32]. In each group, the top row is the superpixel results generated by each algorithm and the bottom row is the magnified regions.

TABLE I
PERFORMANCE METRICS OF SUPERPIXEL SEGMENTATION ALGORITHMS AS THE SUPERPIXEL NUMBER IS 400

	SLIC [15]	PB [26]	LSC [16]	SEEDS [17]	ERS [28]	LRW [2]	our DBSCAN
Adherence to boundaries							
Under-segmentation error (UE)	0.222	0.654	0.191	0.198	0.195	0.201	0.189
Boundary recall (BR)	0.832	0.724	0.924	0.917	0.923	0.851	0.923
Achievable segmentation accuracy (ASA)	0.954	0.927	0.961	0.959	0.960	0.957	0.964
Segmentation speed							
Computational complexity	O(N)	O(N)	O(N)	O(N)	$O(nN^2\lg N)$	$O(nN^2)$	O(N)
Average time per image	0.075s	0.5s	0.503s	0.061s	0.973s	3.894s	0.019s

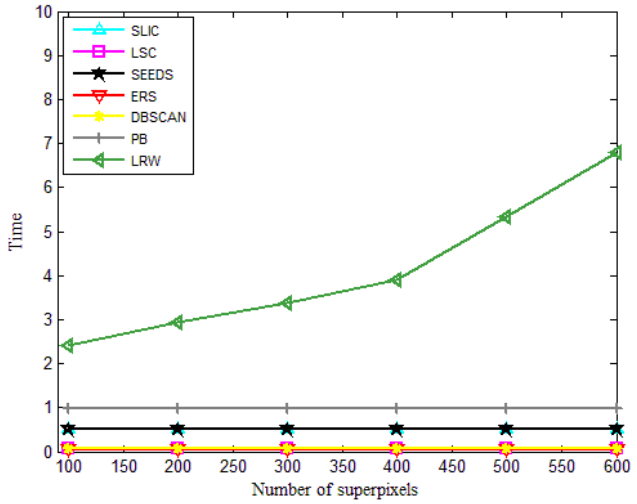


Fig. 8. Average run time with varying numbers of superpixels (from 100 to 600) by state-of-the-art superpixel algorithms (SLIC [18], PB [32], LSC [42], SEEDS [22], ERS [34], LRW [2] and our DBSCAN).

We can clearly see that our superpixel result achieves the better adherence to object edges and keeps the superpixels homogeneous in both weak boundaries and complex texture regions. Our method produces superpixels that align the object boundaries well for weak edges. SLIC cannot segment the ridge on the wall of church well, while LSC cannot do well in complex and irregular regions, such as tree branches. In contrast, our superpixel results preserve important regions like the slit on the ground, the ridge on the wall of church and the tree branches, while SLIC as well as LSC results do not show the same distinction between objects. PB exhibits significant holes to make it hard to use for later analysis. ERS cannot provide a regular and compact shape of each superpixel. Generally, superpixel algorithms should have a low processing cost, because it is usually employed as a preprocessing step in image or video applications. Our real-time DBSCAN superpixel method is the fastest superpixel algorithm among the state-of-the-art algorithms.

D. Analysis of Computational Complexity and Discussions

The computational efficiency is also an important factor for evaluating the performance of superpixel segmentation algorithms. We will analyze the algorithm complexity and compare the computational costs of all the superpixels algorithms. We perform all the experiments on a desktop PC equipped with an Intel 8-core $3.4GH_Z$ processor with 4GB RAM. We do not use any parallelization, GPU or dedicated hardware. According to Table I, PB is based on global optimization to produce superpixels with the computational complexity of $O(N)$. The complexity of the LRW is $O(nN^2)$ and ERS is $O(nN^2\lg N)$, these two methods will spend much more time in obtaining the superpixel results. The complexity of our DBSCAN superpixel method is $O(N)$ without an iteration process, which is faster than all the state-of-the-art superpixel algorithms. Though the complexities of SLIC and LSC algorithms are also $O(N)$, their core algorithm requires many iterations. In addition, LSC needs more computational time for feature spaces under the iterative weighted K-means clustering than SLIC. The reason is that their algorithm iterates many times to find the suitable seeds for getting a good performance. The widely used superpixel segmentation algorithm such as SLIC makes a good balance between time costs and performance of boundaries adherence, and it has a good speed among our compared algorithms. Our algorithm only processes once in both the DBSCAN clustering step and the optimization step, therefore, our algorithm is real-time with the better performance compared to most of the state-of-the-art algorithms in our experiments.

Table I lists the computational statistics between the well-known superpixel algorithms and the proposed method, where the number of superpixels is about 400. The computational complexity of each method is also listed in Table I. The average computing time of the proposed algorithm achieves the speed of 50fps, where the clustering and merging stages cost about 0.011s and 0.007s, respectively. Our experiments are performed on BSDS500 with the typical image size of 320×480. According to the average time per image in Table I, it is obvious that our real-time DBSCAN algorithm is the fastest superpixel method among all these algorithms. It is very important for superpixel segmentation to have the real-time performance to provide a fast preprocessing step in many image processing and vision applications. In order to demonstrate the real-time performance of our DBSCAN algorithm, we have further performed more experimental results for the computing time using varying numbers of superpixels with state-of-the-art superpixel algorithms in Fig. 8.

It is very important to set a correct number of superpixels for a superpixel segmentation algorithm. The current superpixel algorithms can be roughly classified into two main types

regarding setting the number of superpixels. The first type is based on graphs by gradually adding cuts, and the other type starts from an initial seed to gradually grow superpixels. SLIC *et al.* belong to the second type, and they have a seed initialization strategy on the input image before the starting of the superpixel segmentation algorithm, which will help these methods to get an approximate number of superpixels. Our method belongs to the first type, and we also obtain the number of superpixels approximately. In our method, the number of superpixels is set by changing the size of superpixels through the geometric constraint in the clustering stage and merging stage. The threshold S/N in Algorithms 1 and 2 controls the number of superpixels in our method. Since all the superpixels have the regular shapes and similar size, the number of pixels in each superpixel is also similar. Then our algorithm adopts threshold S/N to control the size and shape of superpixels with geometric restrictions so as to obtain an approximate number N of superpixles with the size S of input image.

There are many issues that we should deal with in designing a high quality superpixel algorithm, such as the computational time, uniform compact shape and boundary adherence. Our real-time algorithm with 50fps has the best performance compared with other algorithms in the aspects of computational time, uniform shape and boundary adherence. However, the proposed DBSCAN superpixel algorithm still has some limitations. For example, a potential limitation is that the current DBSCAN superpixel algorithm cannot handle the compactness property perfectly. The reason is that DBSCAN is a local optimum method in both the clustering and merging stages.

V. Conclusion

We proposed a novel image superpixel segmentation algorithm using DBSCAN clustering in this paper. Our DBSCAN superpixel segmentation algorithm produces regular shaped superpixels in 50fps. Our proposed superpixel segmentation first produces the initial superpixel results with the similar colors by performing the DBSCAN clustering algorithm, and then combines the small initial superpixels with their nearest neighbor superpixels by considering their color and spatial information. Evaluation was conducted on the public Berkeley Segmentation Database with using three evaluation metrics. Our algorithm achieves the state-of-the-art performance at a substantially smaller computation cost, and significantly outperforms the algorithms that require more computational costs even for the images including complex objects or complex texture regions. In future work, we will obtain better compactness of superpixels by developing a new DBSCAN algorithm that has the global optimum property. We also plan to extend the current superpixel framework to real-time video supervoxel segmentation for maintaining spatial-temporal compact shapes.

References

[1] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 1. Oct. 2003, pp. 10–17.

[2] J. Shen, Y. Du, W. Wang, and X. Li, "Lazy random walks for superpixel segmentation," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1451–1462, Apr. 2014.

[3] X. Dong, J. Shen, L. Shao, and L. Van Gool, "Sub-Markov random walk for image segmentation," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 516–527, Feb. 2016.

[4] D. Giordano, F. Murabito, S. Palazzo, and C. Spampinato, "Superpixel-based video object segmentation using perceptual organization and location prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4814–4822.

[5] W. Wang and J. Shen, "Higher-order image co-segmentation," *IEEE Trans. Multimedia*, vol. 18, no. 6, pp. 1011–1021, Jun. 2016.

[6] Z. Li, X.-M. Wu, and S.-F. Chang, "Segmentation using superpixels: A bipartite graph partitioning approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 789–796.

[7] Y. Liang, J. Shen, X. Dong, H. Sun, and X. Li, "Video supervoxels using partially absorbing random walks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 5, pp. 928–938, May 2016.

[8] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1323–1330.

[9] J. Cheng *et al.*, "Superpixel classification based optic disc and optic cup segmentation for glaucoma screening," *IEEE Trans. Med. Imag.*, vol. 32, no. 6, pp. 1019–1032, Jun. 2013.

[10] W. Wang, J. Shen, X. Li, and F. Porikli, "Robust video object cosegmentation," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 3137–3148, Oct. 2015.

[11] B. Liu, H. Hu, H. Wang, K. Wang, X. Liu, and W. Yu, "Superpixel-based classification with an adaptive number of classes for polarimetric SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 2, pp. 907–924, Feb. 2013.

[12] G. Shu, A. Dehghan, and M. Shah, "Improving an object detector and extracting regions using superpixels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3721–3727.

[13] J. Shen, D. Wang, and X. Li, "Depth-aware image seam carving," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1453–1461, Oct. 2013.

[14] J. Lim and B. Han, "Generalized background subtraction using superpixels with label integrated motion estimation," in *Proc. ECCV*, 2014, pp. 173–187.

[15] J. Shen, Y. Du, and X. Li, "Interactive segmentation using constrained Laplacian optimization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 7, pp. 1088–1100, Jul. 2014.

[16] D. Hoiem, A. A. Efros, and M. Hebert, "Automatic photo pop-up," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 577–584, Jun. 2005.

[17] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[18] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[19] W. Wang, J. Shen, L. Shao, and F. Porikli, "Correspondence driven saliency transfer," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5025–5034, Nov. 2016.

[20] B. Ma, J. Shen, Y. Liu, H. Hu, L. Shao, and X. Li, "Visual tracking using strong classifier and structural local sparse descriptors," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1818–1828, Oct. 2015.

[21] B. Ma, H. Hu, J. Shen, Y. Liu, and L. Shao, "Generalized pooling for robust object tracking," *IEEE Trans. Image Process.*, vol. 25, no. 9, pp. 4199–4208, Sep. 2016.

[22] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool, "Seeds: Superpixels extracted via energy-driven sampling," in *Proc. ECCV*, 2012, pp. 13–26.

[23] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[24] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "TurboPixels: Fast superpixels using geometric flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2290–2297, Dec. 2009.

[25] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.

[26] J. Shen, X. Yang, Y. Jia, and X. Li, "Intrinsic images using optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 3481–3487.

[27] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining (KDD)*, 1996, pp. 226–231.

[28] A. P. Moore, S. J. D. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[29] R. Manavalan and K. Thangavel, "TRUS image segmentation using morphological operators and dbscan clustering," in *Proc. World Congr. Inf. Commun. Technol.*, Dec. 2011, pp. 898–903.
[30] J. Hou, C. Sha, L. Chi, Q. Xia, and N.-M. Qi, "Merging dominant sets and dbscan for robust clustering and image segmentation," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 4422–4426.
[31] W. Wang, J. Shen, and L. Shao, "Consistent video saliency using local gradient flow optimization and global refinement," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4185–4196, Nov. 2015.
[32] Y. Zhang, R. Hartley, J. Mashford, and S. Burn, "Superpixels via pseudo-Boolean optimization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1387–1394.
[33] O. Veksler, Y. Boykov, and P. Mehrani, "Superpixels and supervoxels in an energy optimization framework," in *Proc. ECCV*, 2010, pp. 211–224.
[34] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 2097–2104.
[35] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2. Jul. 2001, pp. 416–423.
[36] W. Wang, J. Shen, and F. Porikli, "Saliency-aware geodesic video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3395–3402.
[37] B. Ma, L. Huang, J. Shen, and L. Shao, "Discriminative tracking using tensor pooling," *IEEE Trans. Cybern.*, vol. 46, no. 11, pp. 2411–2422, Nov. 2016.
[38] J. Han, D. Zhang, S. Wen, L. Guo, T. Liu, and X. Li, "Two-stage learning to predict human eye fixations via SDAEs," *IEEE Trans. Cybernetics*, vol. 46, no. 2, pp. 487–498, Feb. 2016.
[39] D. Zhang, J. Han, C. Li, J. Wang, and X. Li, "Detection of co-salient objects by looking deep and wide," *Int. J. Comput. Vis.*, vol. 120, no. 2, pp. 215–232, Nov. 2016.
[40] J. Peng, J. Shen, and X. Li, "High-order energies for stereo segmentation," *IEEE Trans. Cybernetics*, vol. 46, no. 7, pp. 1616–1627, Jul. 2016.
[41] X. Lu, X. Li, and L. Mou, "Semi-supervised multitask learning for scene recognition," *IEEE Trans. Cybernetics*, vol. 45, no. 9, pp. 1967–1976, Sep. 2015.
[42] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)* Jun. 2015, pp. 1356–1363.
[43] B. Du, W. Xiong, J. Wu, L. Zhang, L. Zhang, and D. Tao, "Stacked convolutional denoising auto-encoders for feature representation," *IEEE Trans. Cybernetics*, to be published. doi: 10.1109/TCYB.2016.2536638.2016.
[44] A. Bódis-Szomorú, H. Riemenschneider, and L. Van Gool, "Superpixel meshes for fast edge-preserving surface reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2011–2020.
[45] W. Wang, J. Shen, Y. Yu, and K.-L. Ma, "Stereoscopic thumbnail creation via efficient stereo saliency detection," *IEEE Trans. Vis. Comput. Graph.*, to be published. doi: //10.1109/TVCG.2016.2600594.2016.
[46] J. Yan, Y. Yu, X. Zhu, Z. Lei, and S. Z. Li, "Object detection by labeling superpixels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5107–5116.

Jianbing Shen (M'11–SM'12) is currently a Full Professor with the School of Computer Science, Beijing Institute of Technology, Beijing, China. He has authored over 60 journal and conference papers, such as the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE Conference on Computer Vision and Pattern Recognition, and the IEEE International Conference on Computer Vision. His current research interests include computer vision, intelligent systems, and multimedia processing. He has also obtained many flagship honors, including the Fok Ying Tung Education Foundation from the Ministry of Education, the Program for Beijing Excellent Youth Talents from the Beijing Municipal Education Commission, and the Program for New Century Excellent Talents from the Ministry of Education. He is on the Editorial Board of *Neurocomputing*.

Xiaopeng Hao received the B.S. degree in computer science from the China University of Mining and Technology in 2014. He is currently pursuing the M.S. degree with the School of Computer Science, Beijing Institute of Technology, Beijing, China. His current research interests include image superpixel and video supervoxel segmentation.

Zhiyuan Liang received the B.A. degree in computer science from the Beijing Institute of Technology in 2016. He is currently pursuing the M.S. degree with the School of Computer Science, Beijing Institute of Technology, Beijing, China. His current research interests include deep learning for video tracking.

Yu Liu received the M.S. degree in computer science from the Beijing Institute of Technology in 2014. He is currently pursuing the Ph.D. degree with the School of Computer Science, Beijing Institute of Technology, Beijing, China. His current research interests include video reconstruction and motion analysis.

Wenguan Wang received the B.S. degree in computer science and technology from the Beijing Institute of Technology in 2013. He is currently pursuing the Ph.D. degree with the School of Computer Science, Beijing Institute of Technology, Beijing, China. His current research interests include salient object detection and object segmentation for image and video.

Ling Shao (M'09–SM'10) is currently a Full Professor and the Head of the Computer Vision and Artificial Intelligence Group, Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, U.K. His research interests include computer vision, image processing, pattern recognition, and machine learning. He is an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEM, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and other journals. He is a fellow of the British Computer Society, the IET, and a Life Member of the ACM.