USEQ: Ultra-Fast Superpixel Extraction via Quantization

Chun-Rong Huang, Wei-An Wang, Szu-Yu Lin
Department of Computer Science and Engineering
National Chung Hsing University
Taichung, Taiwan
crhuang@nchu.edu.tw

Yen-Yu Lin
Research Center for Information Technology Innovation
Academia Sinica
Taipei, Taiwan
yylin@citi.sinica.edu.tw

***Abstract*—We propose a novel superpixel extraction method named USEQ to generate regular and compact superpixels. To reduce the computational burden of iterative optimization procedures used in most recent approaches, the spatial and color quantizations are performed in advance to represent pixels and superpixels. *Maximum a posteriori* estimation in both pixel and region levels is then adopted to aggregate pixels into spatially and visually coherent superpixels. The resultant superpixels are extremely efficient to generate and can more precisely adhere to object boundaries. Compared to the state-of-the-art approaches to superpixel extraction, USEQ can achieve better or competitive performance in terms of boundary recall, undersegmentation error and achievable segmentation accuracy, and is significantly faster than these approaches.**

Keywords-component; superpixel; image segmentation

I. INTRODUCTION

Superpixels group perceptually similar pixels to represent image regions and adhere to intensity edges for the segmentation purpose. By using superpixels, an image is separated into small semantic regions. These regions can reduce the number of entities and provide an effective way to compute image features for reducing the complexity of image processing tasks. Many compute vision applications such as tracking [1], saliency map detection [2], and image segmentation [3] are also developed based on superpixels. Thus, superpixel generation becomes a fundamental and important task in image processing and computer vision domains.

As indicated in [4], the following three properties are generally desired for superpixel extraction. First, each superpixel should contain visually similar pixels, and adhere to image boundaries adequately. Second, because generating superpixels serves as a preprocessing step to reduce the computational complexity for successive image processing tasks, it is required to be computationally efficient. Third, the generated superpixels should increase the efficiency and improve the quality of the segmentation results. According to these properties and the growing image resolutions, effective and efficient superpixel extraction methods are always in demand. In general, increasing the number of superpixels can represent boundaries of images more precisely. However, the computation time of superpixel extraction will also significantly increase, which limits the practical usage of superpixels to high-resolution images and videos.

To solve the aforementioned problems, we propose a novel superpixel extraction approach, *USEQ*, to efficiently compile semantic regions. To reduce the computational burden of optimization processes, we apply the spatial quantization, which generates the initial superpixels based on the positions of pixels to represent the spatial relationships between pixels and initial superpixels. Then, the color space is quantized for each pixel to obtain the dominant colors within each initial superpixel. Most conventional methods such as [4] apply the averaged colors of all of the pixels as the initial colors of the superpixels. Iterative optimization procedures are then required to update the colors of the superpixels, and repeatedly assign pixels to the superpixels. However, the dominant colors obtained by quantization represent more accurate color information of the superpixels. Thus, pixels only need to be assigned to the most spatially and visually similar superpixels via *maximum a posteriori* (MAP) estimation. As a result, the iterative optimization procedure adopted in most conventional approaches is not required and can be replaced by the MAP estimation. Finally, we apply the neighborhood refinement to combine small superpixels based on MAP estimation in the region level and obtain superpixels with regular and compact shapes. In the experiments, we evaluate the performance of the proposed approach on the Berkeley segmentation benchmark [5]. Compared to the state-of-the-art approaches, our approach not only achieves better boundary recalls but also is much more computationally efficient. To the best our knowledge, our method is faster than existing methods and provides the flexibility of generating regular superpixels with different numbers of superpixels. The rest of the paper is organized as follows. We review the state-of-the-art methods in Section II. Our method is presented in Section III. The experimental results and the comparisons with the state-of-the-art approaches are shown in Section IV. Conclusions are given in Section V.

II. RELATED WORK

A. Graph-based Approaches

To construct superpixels, graph-based methods employ graphs to model the relationships between neighboring pixels.

This work was supported in part by MOST104-2221-E-005-027-MY3 and MOST105-2221-E-001-030-MY2 from the Ministry of Science and Technology, Taiwan.

As shown in a pioneering work, *normalized cuts* [6], pixels are represented as nodes with their links to the neighbors as edges in a graph. Superpixels are obtained by recursively minimizing a cost function defined on the graph.

To reduce the computational cost of normalized cuts, *guiding model search* is introduced in [7]. Felzenszwalb and Huttenlocher [8] present a graph-based segmentation approach, in which the agglomerative clustering is applied, so that each node in the graph forms a minimum spanning tree. Their method shows its advantage over normalize cuts in efficiency, but it leads to superpixels of less regular sizes. Moore et al. [9] generate superpixels by preserving the topology of a regular lattice. They use vertical and horizontal paths to split images, and optimize these paths by referring to the boundary cost map. Liu et al. [10] present an approach to superpixel segmentation, in which the entropy rate and a balancing function jointly constrain the compactness and sizes of each cluster, and a greedy algorithm is adopted to complete the segmentation. Their method is computationally more efficient than normalized cuts [6]. Veksler et al. [11] over-segment an image by covering it with overlapping square patches of the fixed size. They develop an energy function, which takes image gradients as the input, to guide the assignment from pixels to superpixels by using graph-cuts. Zhang et al. [12] introduce two *pseudo-Boolean* functions and model the segmentation problem as a binary labeling problem. The adopted non-iterative pseudo Boolean optimization makes their method more efficient than that in [11]. Li and Chen [13] propose the *linear spectral clustering* (LSC) to construct uniform superpixels. A normalized cuts formulation is adopted and is optimized by iteratively applying weighted *k*-means clustering.

B. Gradient-ascent-based Approaches

Unlike graph-based approaches, the gradient ascent-based approaches generate initial regions as the reference, and gradually adjust the region boundaries to yield superpixels with perceptually similar pixels. For instance, the *watershed* approach [14] considers the flooding of the water from local minima in an image to retrieve the segments of superpixels. As a result, the shapes of the superpixels may be too irregular to adhere to boundaries of objects.

Mean shift [15] searches the local maxima of a density function by using an iterative mode-seeking procedure. After convergence, pixels belonging to the same mode form a superpixel. Levinshtein et al. [16] deliver a method for compiling the *TurboPixels*. It uniformly places the initial seeds on the image and gradually expands the superpixels from the seeds by a level set based geometric flow algorithm. The method can make the sizes of the superpixels uniform, but it is less efficient compared to other gradient-ascent-based methods. Zeng et al. [17] propose structure-sensitive superpixels based on the geodesic distance computed from geometric flows. The number of superpixels is automatically adjusted by the energy functions of the structure density and compactness constraints. Achanta et al. [4] propose a method, called *simple linear iterative clustering* (SLIC), to construct superpixels. SLIC also generates initial seeds as the centers in *k*-means algorithm. The computational complexity of SLIC, mainly on running *k*-means, is dramatically reduced by considering local search regions. Although SLIC is efficient, the yielded superpixels are sensitive to the locations of initial seeds. Bergh et al. [18] extract superpixels via using an energy-driven sampling (SEEDS) method. Their method initializes the superpixels as the uniform cells, and progressively adjusts the boundaries of superpixels according to an energy function that takes the color homogeneity and shape prior of superpixel boundaries into account. The optimization of the energy function is solved by a hill-climbing algorithm. However, the shapes of the generated superpixels are often irregular. The computation time also significantly increases with respect to the number of superpixels.

More recently, Shen et al. [19] use *lazy random walk* (LRW) to represent the relationship between a seed and its neighbor pixels, and generate superpixels. To improve the performance, an energy optimization function based on texture information and object boundaries in the image is developed and adopted. However, their method is time consuming. Fu et al. [20] propose *regularity preserved superpixels* (RPS) to maintain regularity properties. Based on the initial seeds, the pixels are re-assigned based on locally maximal edge magnitudes. The shortest path algorithm retrieves local optimal boundaries. They also extend RPS to generate supervoxels.

III. METHOD

A. Spatial Quantization

Given the target number δ of superpixels, we perform the spatial quantization to obtain the initial positions and sizes of each superpixel. Then, we build the spatial relationships between superpixels and pixels. Let W and H be the width and height of the image I, respectively. Let $\boldsymbol{sp}_i = [u_i\ v_i]^{\mathrm{T}}$ be the initial position of the center of the ith superpixel sp_i by uniformly sampling on a regular grid in the x and y axes of the image, respectively. Let $\boldsymbol{p}_k = [x_k\ y_k]^{\mathrm{T}}$ represent the position of the kth pixel p_k in I. In the spatial quantization, pixels belonging to a superpixel sp_i are defined as follows:

$$sp_i = \{p_k \mid \| \boldsymbol{p}_k - \boldsymbol{sp}_i \| < \| \boldsymbol{p}_k - \boldsymbol{sp}_j \| \ \forall\ j \neq i\}. \tag{1}$$

If p_k belongs to the sp_i, the initial label of p_k is l_i. To represent the spatial relationships among pixels and superpixels, the spatial neighbor relationship $e(sp_i, sp_j)$ between sp_i and sp_j is defined as follows:

$$e(sp_i, sp_j) = \begin{cases} 1, & \text{if } \boldsymbol{sp}_i \text{ and } \boldsymbol{sp}_j \text{ are neighbor girds} \\ 0, & \text{otherwise} \end{cases}. \tag{2}$$

By using the spatial quantization, we can efficiently build the spatial neighbor relationships between spatial grids.

B. Color Quantization

Because a superpixel is supposed to contain pixels with similar colors, graph-based and gradient-ascent-based approaches apply iterative processes to recursively retrieve pixels with similar colors. To reduce the computational burden in the iterative processes, we consider using binary color

quantization [21][22] to effectively retrieve homogeneous color quantization results of each initial superpixel.

Given an image I, a pixel $p_k \in I$ is represented by a three-dimensional color vector $\boldsymbol{c}_k = (r_k, g_k, b_k)^{\mathrm{T}}$, where r_k, g_k, and b_k respectively represent the red, greed, and blue values of p_k. The objective of the color quantization algorithm is to partition the original color space C into M disjoint ones $\{C_1, \ldots, C_m, \ldots, C_M\}$ containing the quantization results, where $M = 2^\theta \times 2^\theta \times 2^\theta$ is the size of the palette after quantization. For efficiency, we apply the binary tree structure to represent the partition of C. Each node is partitioned to two children nodes until reaching the leaf nodes. Each leaf node of the binary tree represents a subset C_m of C. For representing the operations the binary partition, the index of root node is 1, and the children of node n are indexed as $2n$ and $2n+1$, respectively. During the binary partitioning process, the set of pixels belonging to node n is labeled as C_n. There is no intersection between two partitions C_{2n} and C_{2n+1} of C_n. N_n is the total number of pixels residing in C_n.

Similar to [21][22], we aim to find a quantized color image I_q that minimizes total square error (TSE) between I_q and I as follows:

$$TSE = \sum_{n=1}^{M} \sum_{p_k \in C_n} \| \boldsymbol{c}_k - \boldsymbol{q}_n \|^2, \tag{3}$$

where $\boldsymbol{q}_n$ is the quantized color of C_n. The statistic $\boldsymbol{m}_n$ used to decide the binary partition is defined as follows:

$$\boldsymbol{m}_n = \sum_{p_k \in S_n} \boldsymbol{c}_k. \tag{4}$$

The quantized color $\boldsymbol{q}_n$ of C_n is defined as follows:

$$\boldsymbol{q}_n = \frac{\boldsymbol{m}_n}{2}. \tag{5}$$

To split a node n into two nodes for minimizing TSE, Orchard and Bouman [21] consider projecting the color vectors of pixels in C_n by using the eigenvector of the cluster covariance matrix. For efficiency, we consider using the $\boldsymbol{q}_n$ to split C_n into two sets C_{2n} and C_{2n+1}, respectively, as follows:

$$C_{2n} = \{p_k \in C_n : \boldsymbol{c}_k \le \boldsymbol{q}_n\}, \tag{6}$$

and

$$C_{2n+1} = \{p_k \in C_n : \boldsymbol{c}_k > \boldsymbol{q}_n\}. \tag{7}$$

The sets C_{2n} and C_{2n+1} can be split again until the target palette size M is reached. Please note that $\theta = 8$ means that the number of quantization labels is exactly the same as the number of colors of the original image I, i.e. $TSE = 0$. After color quantization, we can then generate the color maps of pixels, which records the labels of color quantized pixels.

Besides the pixels, the dominant color of sp_i is also computed based on the color quantization results. A superpixel usually contains perceptually similar pixels in a region. Namely, the pixels in a superpixel should have similar colors. To generate a representative color for each superpixel sp_i, we compute the color histogram using quantization labels of pixels in that superpixel, and find the dominant label, which represents the dominant color of the initial superpixel. Let $h_i(C_m)$ be the value of the label C_m of the color histogram in sp_i. The dominant label C_{m*} is defined as the label, which contains the maximal value of $h_i(C_m)$ in sp_i as follows:

$$C_{m*} = \arg\max_{C_m} h_i(C_m). \tag{8}$$

The dominant color $\boldsymbol{c}_i$ of sp_i is then computed as follows:

$$\boldsymbol{c}_i = \frac{\sum_{p_k \in sp_i, \boldsymbol{c}_k \in C_{m*}} \boldsymbol{c}_k}{\sum_{p_k \in sp_i, \boldsymbol{c}_k \in C_{m*}} 1}. \tag{9}$$

With the spatial and color quantization results, we then propose a non-iterative MAP pixel label assignment method to retrieve the candidate boundary of superpixels.

C. Non-Iterative MAP Pixel Label Assignment

To avoid the time-consuming iterations of most recent superpixel approaches and efficiently assign pixels to correct superpixels, we propose a non-iterative MAP pixel label assignment method by considering both the spatial and color quantization results. Given $p_k \in sp_i$, the initial label of p_k is l_i. Because the boundaries of initial superpixels may cover multiple objects, the initial superpixels may not adhere to the true boundaries of objects. To better adhere to intensity edges, the labels of pixels need to be reassigned.

Given the color $\boldsymbol{c}_k$ and the location $\boldsymbol{p}_k$ of a pixel p_k, we aim to retrieve the most plausible superpixel sp_{i*} for p_k by using the *maximum a posteriori* (MAP) estimation of the posterior probability function $p(sp_i|p_k)$. Based on the formula of Bayes' theorem, the posterior probability function is derived as follows:

$$p(sp_i \mid p_k) \propto p(p_k \mid sp_i) p(sp_i), \tag{10}$$

where $p(p_k|sp_i)$ is the likelihood function representing the similarity between the pixel p_k and the superpixel sp_i, and $p(sp_i)$ is the prior probability function of sp_i, which represents the possibility of sp_i as a suitable superpixel. Because each superpixel is represented by the spatial and the color quantization results, the likelihood probability function $p(p_k|sp_i)$ can be represented as follows:

$$p(p_k \mid sp_i) = p(\boldsymbol{p}_k, \boldsymbol{c}_k | \boldsymbol{sp}_i, \boldsymbol{c}_i), \tag{11}$$

where $\boldsymbol{c}_i$ and $\boldsymbol{sp}_i$ are the dominant color and the initial location of sp_i, respectively. Since the spatial and color quantization results are independent, (11) can be rewritten as follows:

$$p(p_k \mid sp_i) = p(\boldsymbol{p}_k \mid \boldsymbol{sp}_i) p(\boldsymbol{c}_k | \boldsymbol{c}_i), \tag{12}$$

where $p(\boldsymbol{p}_k|\boldsymbol{sp}_i)$ and $p(\boldsymbol{c}_k|\boldsymbol{c}_i)$ are the spatial and color likelihood functions of spatial quantization and color quantization results of p_k and sp_i, respectively. To represent the spatial quantization, we consider the similarity between the positions of p_k and sp_i. The spatial likelihood function $p(\boldsymbol{p}_k|\boldsymbol{sp}_i)$ is defined as follows:

$$p(\boldsymbol{p}_k \mid \boldsymbol{sp}_i) = e^{-\omega\|\boldsymbol{p}_k - \boldsymbol{sp}_i\|}. \quad (13)$$

To avoid the effects of different sizes of images, we normalize $\boldsymbol{p}_k$ and $\boldsymbol{sp}_i$ with respect to the image width and height in advance.

To represent the color likelihood function, we consider the similarity between the color of p_k and the dominant color of sp_i. The likelihood function $p(\boldsymbol{c}_k|\boldsymbol{c}_i)$ is defined as follows:

$$p(\boldsymbol{c}_k \mid \boldsymbol{c}_i) = e^{-(1-\omega)\|\boldsymbol{c}_k - \boldsymbol{c}_i\|}, \quad (14)$$

where $\boldsymbol{c}_k$ and $\boldsymbol{c}_i$ are normalized by 255. Constant ω in Eq. (13) and (14) represents the weight between the spatial and color quantization results.

The prior function $p(sp_i)$ here is used to reduce the computation time for the MAP estimation. When a superpixel sp_i is not a spatial neighbor of pixel p_k, p_k should not belong to sp_i. Thus, the prior function $p(sp_i)$ is defined as follows:

$$p(sp_i) = \begin{cases} 1, & \text{if } e(sp_i, sp_j) > 0, \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

To achieve non-iterative pixel label assignment, we search the most possible sp_{i*} for p_k by maximizing the posterior probability function $p(sp_i|p_k)$ as follows:

$$sp_{i*} = \arg\max_{sp_i} p(sp_i \mid p_k), \quad (16)$$

where sp_{i*} with the maximal posterior probability is the most similar superpixel for p_k. Because the MAP process is performed once for each pixel p_k, the candidate label of p_k is l_{i*} without performing the iterative optimization procedure.

D. Neighborhood Refinement

After the pixel assignment, small objects in complex scenes may be assigned to different labels, because their colors are different from each other. Although keeping these small regions help to represent the detailed shapes of small objects, the shapes of the generated superpixels will become irregular and incompact. To solve the problem, the neighborhood refinement process is applied to merge small superpixels into spatially connected and visually similar superpixels.

To retrieve small superpixels, we apply the flood filling for obtaining connected components of the pixel assignment results. Then, we update the spatial correlation $e(sp'_i, sp'_j)$ between candidate superpixels sp'_i and sp'_j as follows:

$$e(sp'_i, sp'_j) = \begin{cases} 1, & N(sp'_i, sp'_j) \neq \Phi \\ 0, & \text{otherwise} \end{cases}, \quad (17)$$

where Φ means the null set, and $N(sp'_i, sp'_j)$ is used to decide if sp'_i and sp'_j are spatially connected as follows:

$$N(sp'_i, sp'_j) = \{sp'_i | \exists p_k \in sp'_i, \exists p_l \in sp'_j, \|\boldsymbol{p}_k - \boldsymbol{p}_l\| = 1\}. \quad (18)$$

Similar to the pixel assignment, the small superpixel assignment is also achieved by using the MAP estimation of the posterior probability function $p(sp'_i|sp'_j)$. Based on the formula of the Bayes' theorem, the posterior probability function is derived as follows:

$$p(sp'_i|sp'_j) \propto p(sp'_j|sp'_i)p(sp'_i), \quad (19)$$

where $p(sp'_j|sp'_i)$ is the likelihood function representing the similarity between superpixel sp'_j and sp'_i, and $p(sp'_i)$ is the prior probability function of sp_i, which represents the possibility of sp'_i as a suitable superpixel.

Because two superpixels need to be visually similar enough to be merged, the likelihood function $p(sp'_j|sp'_i)$ is defined as follows:

$$p(sp'_j \mid sp'_i) = e^{-\|\boldsymbol{c}'_j - \boldsymbol{c}'_i\|}, \quad (20)$$

where $\boldsymbol{c}'_j$ and $\boldsymbol{c}'_i$ are the dominant colors of sp'_j and sp'_i, respectively. The prior function $p(sp'_i)$ is used to measure if sp'_j and sp'_i should be merged. For the compactness of superpixels, we only merge spatially connected superpixels represented by G'_{sp}. Thus, $p(sp'_i)$ is then defined as:

$$p(sp'_i) = \begin{cases} 1, & \text{if } e(sp'_i, sp'_j) > 0, \\ 0, & \text{otherwise.} \end{cases} \quad (21)$$

The small superpixel sp'_j is then merged to superpixel sp'_{i*} based on the MAP estimation as follows:

$$sp'_{i*} = \arg\max_{sp'_i} p(sp'_i \mid sp'_j). \quad (22)$$

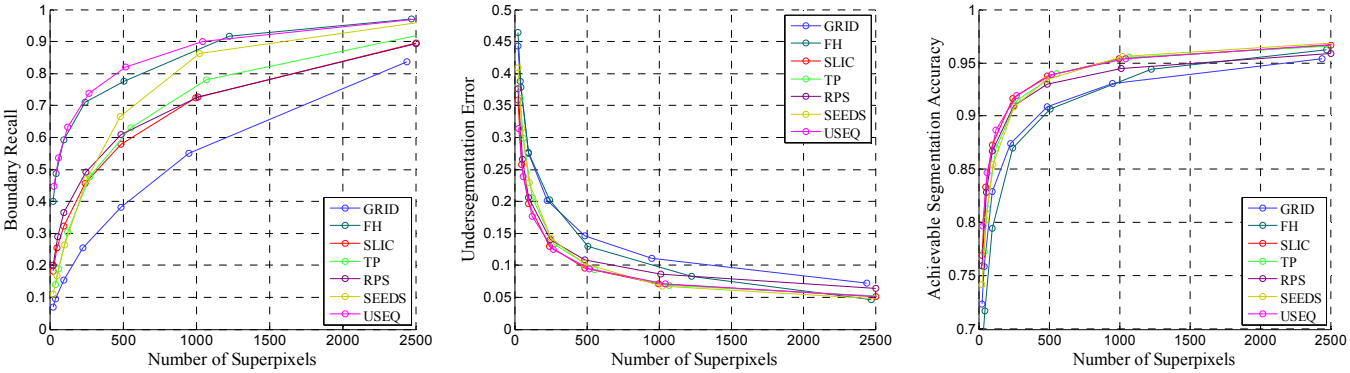
As a result, the constructed superpixels can have more regular and compact shapes compared to the candidate superpixels after pixel label assignment.

IV. Experiments

In the experiments, we applied the Berkeley segmentation benchmark BSDS500 [2], which contains 500 manually labelled results for evaluation. The evaluation metrics include boundary recall (BR), undersegmentation error (UE), and achievable segmentation accuracy (ASA) which are commonly used in most recent state-of-the-art papers. Among these three metrics, BR represents the correctness of adhering the true boundaries of objects. The high BR indicates that the extracted superpixels adhere to boundaries of objects better. UE measures the superpixel overlapping with multiple objects by the percentage of pixels that leak from the ground truth boundaries. Thus, the low UE indicates better adherence of boundaries of objects. By matching the labels of each superpixel with respect to the labels of ground truths, ASA is computed to evaluate the highest achievable object segmentation accuracy. Similar to BR, the high ASA indicates better object representation in the image. Besides the performance evaluation, we also list the average computation time for comparison. The USEQ code is available at http://cvml.cs.nchu.edu.tw/USEQ.html.

A. Quantitative Comparisons

We compare USEQ to five state-of-the-art superpixel extraction approaches, including FH [8], SLIC [4], Turbopixel (TP) [16], RPS [20], and SEEDS [18] on the BSDS500 dataset. The parameters of USEQ is empirically set as $\theta = 3$, and $\omega = 0.01$ in the following comparisons. Because FH cannot control a fixed output number of superpixels, we adjust the parameters of FH to extract superpixels of the desired numbers. For



(a) (b) (c)

Figure 1. Quantitative evaluation results of the state-of-the-art methods and USEQ. (a) BR, (b) UE, and (c) ASA.

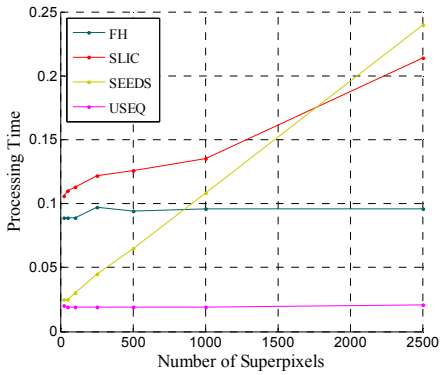


Figure 2. Processing time of the state-of-the-art methods and USEQ with respect to the numbers of superpixels.

comparison, we also show the performance of the spatial quantization grid (GRID) as a baseline. To provide fair comparisons, the results of all of the compared approaches were obtained from the codes released by the original authors.

As shown in Figure 1(a), USEQ owns the best BR curve compared to state-of-the-art approaches when increasing the number δ of superpixels. When colors of pixels of objects are different, these pixels are quantized to different labels in USEQ. Then, based on the MAP pixel label assignment, pixels of different objects with different quantized labels are assigned to different superpixels. Thus, USEQ can precisely adhere to the boundaries of objects. FH owns the second best results compared to remaining approaches. Such results are consistent to the results reported in [4][18]. Nevertheless, FH has higher UE as shown in Figure 1(b). In contrast, our USEQ still has low UE and has the best ASA as shown in Figure 1(b) and (c), respectively.

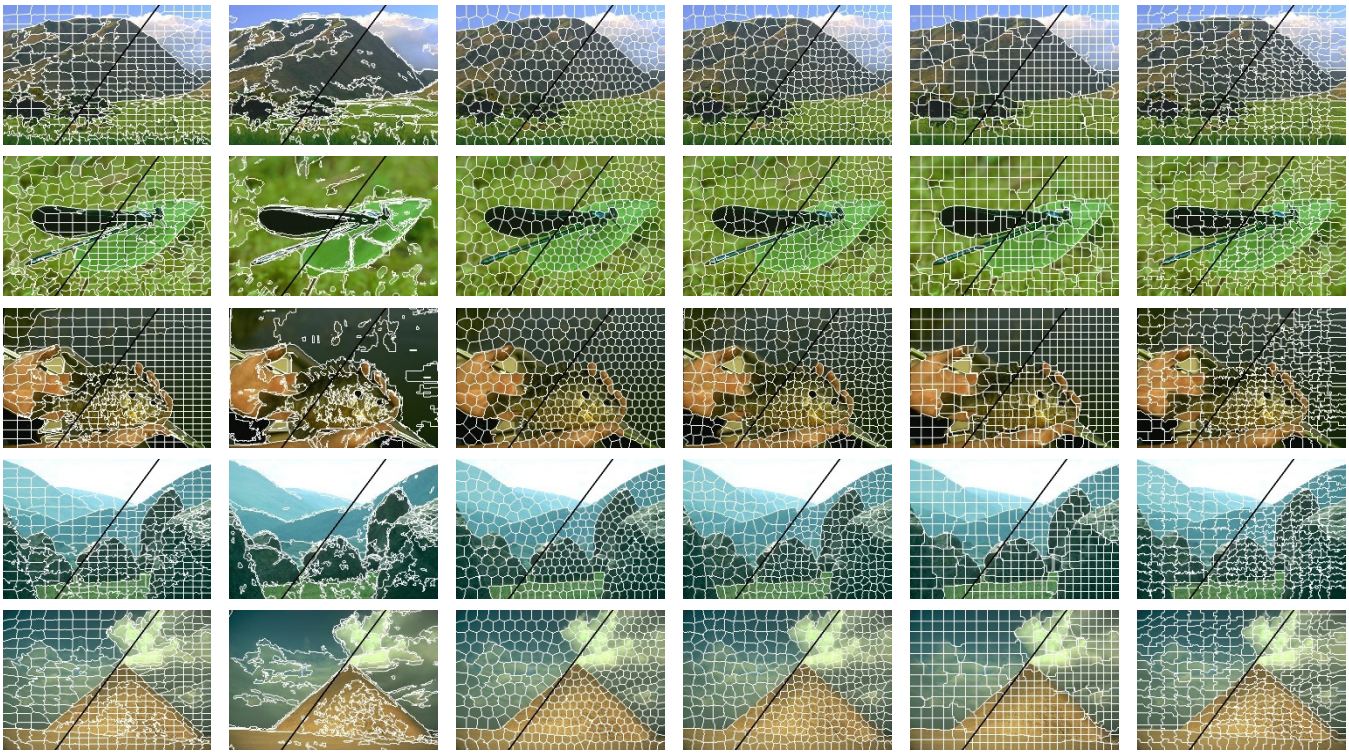
Let the number of pixels be N. The complexity of spatial and color quantization processes are $O(N)$, because all pixels are processed sequentially. Each pixel is also assigned once in the MAP estimation process of the pixel and region levels. Thus, the complexity of USEQ is $O(N)$. As for the computational efficiency evaluation, all of the approaches were run under an Intel Core i7 3.40GHz computer with 8G memory and no GPU accelerators were applied. Figure 2 shows the average computation time of the top three efficient superpixel approaches including FH, SLIC and SEEDS. Our USEQ owns the fastest computation time compared to these approaches. Although the complexity of SLIC and SEEDS is also $O(N)$, their iterative procedures performed on superpixel generation lead to significantly increasing time with respect to the number of superpixels. In contrast, the computational complexity of FH is based on the number of graph edges and thus, is not correlated with the number of superpixels. Because no iterative optimization procedure is required in USEQ, the computation time only slightly increases with respect to the number of superpixels.

B. *Qualitative Comparisons*

Figure 3 shows the superpixel extraction results of the state-of-the-art approaches and USEQ. To demonstrate the effects of different numbers of superpixels, we segmented images into 250/500 superpixels. Figure 3(a) shows the results of our USEQ. When image regions are smooth, USEQ can generate regular superpixels and precisely adhere to boundaries when the shapes of objects are irregular. In contrast, the generated shapes and sizes of superpixels using FH are very irregular as shown in Figure 3(b). SLIC, TP, and RPS generate more regular superpixels compared to FH as shown in Figure 3(c), (d), and (e), respectively, but fail to correctly adhere to the detailed boundaries of objects. The results of SEEDS shown in Figure 3(f) achieve better boundary adherence compared to SLIC, TP and RPS. Nevertheless, the shapes of the superpixels of SEEDS are not as regular as those of USEQ. For demonstration of high resolution images and more number of superpixels, please refer to the demo video in the supplementary material.

V. Conclusions

We propose a novel superpixel extraction approach based on the spatial and color quantization. The proposed bottom up procedure assigns labels of pixels by grouping visually similar pixels and then merge small fragments to generate regular and compact superpixels without iterative optimization procedures applied in most conventional approaches. As a result, our approach is significantly faster than the state-of-the-art approaches. Experimental results on the BSDS500 dataset have demonstrated the effectiveness and efficiency of the proposed USEQ in terms of both quantitative and qualitative criteria. In the future, we will extend the image based



(a) USEQ (b) FH (c) SLIC (d) TP (e) RPS (f) SEEDS

Figure 3. The qualitative results of (a) USEQ, (b) FH, (c) SLIC, (d) TP, (e) RPS, and (f) SEEDS, respectively.

superpixel framework to extract supervoxels for video processing.

REFERENCES

[1] F. Yang, H. Lu, and M.-H. Yang, "Robust superpixel tracking," *IEEE Trans. Image Processing*, vol. 23, no. 4, pp. 1639–1651, Apr. 2014.

[2] Z. Liu, X. Zhang, S. Luo, and O. L. Meur, "Superpixel-based spatiotemporal saliency detection," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 24, no. 9, pp. 1522–1540, Sep. 2014.

[3] Z. Li, X.-M. Wu, and S.-F. Chang, "Segmentation using superpixels: A bipartite graph partitioning approach," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 789–796, Jun. 2012.

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

[5] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, May 2011.

[6] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[7] G. Mori, "Guiding model search using segmentation," in *International Conf. on Computer Vision*, vol. 2, pp. 1417–1423, Oct. 2005.

[8] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[9] A. P. Moore, S. J. D. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1–8, Jun. 2008.

[10] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2097–2104, Jun. 2011.

[11] O. Veksler, Y. Boykov, and P. Mehrani, "Superpixels and supervoxels in an energy optimization framework," in *European Conf. on Computer Vision*, pp. 211–224, 2010.

[12] Y. Zhang, R. Hartley, J. Mashford, and S. Burn, "Superpixels via pseudo-boolean optimization," in *International Conf. on Computer Vision*, pages 1387–1394, Nov. 2011.

[13] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1356–1363, Jun. 2015.

[14] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–598, Jun. 1991.

[15] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.

[16] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2290–2297, Dec. 2009.

[17] G. Zeng, P. Wang, P. Wang, R. Gan, and H. Zha, "Structure sensitive superpixels via geodesic distance," in *International Conf. on Computer Vision*, pp. 447–454, Nov. 2011.

[18] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool, "Seeds: Superpixels extracted via energy-driven sampling," in *European Conf. on Computer Vision*, pp. 13–26, 2012.

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

[20] H. Fu, X. Cao, D. Tang, Y. Han, and D. Xu, "Regularity preserved superpixels and supervoxels," *IEEE Trans. Multimedia*, vol. 16, no. 4, pp. 1165–1175, Jun. 2014.

[21] M. Orchard and C. Bouman, "Color quantization of images," *IEEE Trans. Signal Processing*, vol. 39, no. 12, pp. 677–2690, Dec. 1991.

[22] X. Chen, S. Kwong, and J.-F. Feng, "A new compression scheme for color-quantized images," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 10, pp. 904–908, Oct. 2002.