

Superpixel Segmentation using Linear Spectral Clustering

Zhengqin Li

Jiansheng Chen

Department of Electronic Engineering, Tsinghua University, Beijing, China

li-zq12@mails.tsinghua.edu.cn

jschenthu@mail.tsinghua.edu.cn

Abstract

We present in this paper a superpixel segmentation algorithm called *Linear Spectral Clustering (LSC)*, which produces compact and uniform superpixels with low computational costs. Basically, a normalized cuts formulation of the superpixel segmentation is adopted based on a similarity metric that measures the color similarity and space proximity between image pixels. However, instead of using the traditional eigen-based algorithm, we approximate the similarity metric using a kernel function leading to an explicitly mapping of pixel values and coordinates into a high dimensional feature space. We revisit the conclusion that by appropriately weighting each point in this feature space, the objective functions of weighted *K*-means and normalized cuts share the same optimum point. As such, it is possible to optimize the cost function of normalized cuts by iteratively applying simple *K*-means clustering in the proposed feature space. *LSC* is of linear computational complexity and high memory efficiency and is able to preserve global properties of images. Experimental results show that *LSC* performs equally well or better than state of the art superpixel segmentation algorithms in terms of several commonly used evaluation metrics in image segmentation.

1. Introduction

Superpixel segmentation is an increasingly popular image preprocessing technique used in many computer vision applications such as image segmentation [17], image parsing [19], object tracking [22], and 3D reconstruction [9]. It provides a concise image representation by grouping pixels into perceptually meaningful small patches that adhere well to object boundaries. Comparing to the pixel-rigid image representation, superpixel is more consistent with human visual cognition and contains less redundancy. Moreover, compact and uniform superpixel segmentation can serve as the spatial support for vision feature extraction [12].

Many different superpixel segmentation algorithms have been proposed to meet the needs of various applications [17][8][5][21][11]. It is widely understood that the follow-



Figure 1. Images [13] segmented into 1000/500/200 superpixels using the proposed LSC algorithm.

ing properties of superpixel segmentation are generally desirable. First, superpixels should adhere well to the natural image boundaries and each superpixel should not overlap with multiple objects. Second, as a preprocessing technique for improving efficiency of computer vision tasks, superpixel segmentation should be of low complexity itself. Last but not the least, global image information which is important for human vision cognition should be considered appropriately. It is critical for a segmentation process to utilize the perceptually important non-local clues to group unrelated image pixels into semantically meaningful regions. Nevertheless, considering global relationship among pixels usually lead to substantial increases in computational complexity. A typical example is the eigen-based solution to the normalized cuts (Ncuts) based superpixel segmentation algorithm proposed in [17]. As a result, most practical superpixel segmentation algorithms, such as [5][21][11], are mainly based on the analysis of local image information only. These methods may fail to correctly segment image regions with high intensity variability [8].

To address this issue, we propose a superpixel segmentation algorithm, *Linear Spectral Clustering (LSC)*, which not only captures perceptually important global image properties, but also runs in linear complexity with high memory efficiency. In LSC, we map each image pixel to a point in a ten dimensional feature space in which weighted *K*-means is applied for segmentation. Non-local information is implicitly preserved due to the equivalence between the weighted *K*-means clustering in this ten dimensional feature space and normalized cuts in the original pixel space. Simple weighted *K*-means clustering in the feature space can be

used to optimize the segmentation cost function defined by normalized cuts. Figure 1 shows some superpixel segmentation results of LSC. We will demonstrate the efficiency and effectiveness of LSC through further experiments.

The rest of this paper is organized as follows. In Section 2, we briefly review existing approaches for superpixel segmentation. Section 3 presents the proposed LSC method. Experimental results are demonstrated in Section 4. The last section concludes our work.

2. Related Works

In early studies, algorithms designed for image segmentation were directly used for generating superpixels, such as FH [8], mean shift [5] and quick shift [21]. In FH, each superpixel is represented by a minimum spanning tree and two superpixels are merged if the maximum weight of edges inside the trees is larger than the minimum weight of edges connecting them. Mean shift and quick shift are two mode-seeking methods attempting to maximize a density function by shifting pixels towards areas of higher density. Pixels converging to the same mode formulate a superpixel. These algorithms offer no explicit control over the size and number of the superpixel and compactness is not considered. Superpixels thus produced are usually of irregular sizes and shapes and tend to overlap with multiple objects.

Another widely known algorithm adopts the normalized cuts formulation [18]. However, the traditional eigen-based solution is of high computational complexity which further grows when the number of eigenvectors to be computed increases. For superpixel segmentation, the number of eigenvectors equals the expected number of superpixels, which is usually much larger than the number of segments in traditional image segmentation. Therefore, to facilitate normalized cuts based superpixel segmentation, Ren and Malik proposed a two step algorithm (Ncuts) [17], in which pixels are first grouped into large regions by eigen-based normalized cuts and direct K-means clustering is then adopted to further partition these regions into small superpixels. Due to its heuristic nature, Ncuts is less effective comparing to other methods when the number of superpixel grows.

Previous researches show that algorithms which do not consider the spatial compactness usually lead to under segmentation, especially when there is poor contrast or shadow [11]. Among the four algorithms mentioned above, Ncuts [17] is the only one that implicitly takes compactness into consideration. However, the high computational complexity has limited its applicability. To solve this problem, several other approaches have been proposed to generate compact and regular superpixels with relatively low computational complexity. The Turbopixel algorithm [11] generates highly uniform lattice-like superpixels by iteratively dilating regularly distributed seeds. However, due to the stability and efficiency issues of the level-set method, superpixels

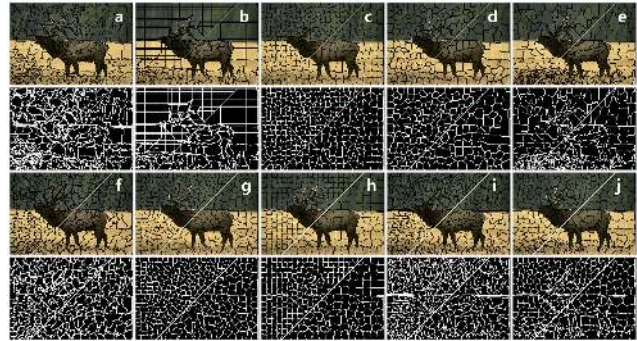


Figure 2. Comparison of different superpixel segmentation algorithms. (a)SEEDS, (b)Lattice, (c)Turbopixel, (d)EneOpt0, (e)EneOpt1, (f)quick shift, (g)Ncuts, (h)SLIC, (i) ERS and (j)LSC. The image [13] is segmented into 400/200 superpixels.

thus generated present relative low adherence to boundaries and the algorithm is slow in practice. Veksler et al. formulated superpixel segmentation as an energy optimization problem which was then solved using the min-cut/max-flow algorithm [4][3][10]. The authors further extended this algorithm to two variations (EneOpt0 and EneOpt1) by balancing between shape regularity and boundary adherence differently [20]. Moore et al. proposed an algorithm (Lattice) that preserves the topology of a regular lattice in superpixel segmentation [15][14]. Nevertheless, the quality of the superpixel relies on a pre-calculated boundary probability map. Liu et al. presented in [12] a clustering objective function that consists of the entropy rate (ERS) of a random walk and a balancing term which encourages the generation of superpixels with similar sizes. ERS is able to preserve jagged object boundaries which are likely to be smoothed by other algorithms. However, the irregular shape of ERS superpixels may become a potential drawback in feature extraction [1]. Bergh et al. proposed SEEDS in [2] by introducing an energy function that encourages color homogeneity and shape regularity. A hill-climbing algorithm is used for optimization. However, SEEDS also suffers from highly shape irregularity and the superpixel number is difficult to control. Achanta et al. proposed a linear clustering based algorithm (SLIC) which produces superpixels by iteratively applying simple K-means clustering in the combined five dimensional color and coordinate space. In spite of its simplicity, SLIC has been proved to be effective in various computer vision applications [22]. Nevertheless, as a local feature based algorithm, the relationship between SLIC and global image properties is not clear.

Another work closely related to our proposed method was introduced in [7], in which Dhillon et al. proved that K-way normalized cuts in the original pixel space is identical to the weighted K-means clustering in a high dimensional feature space by rewriting weighted K-means clustering as a trace maximization problem. However, in [7], the high

dimensional feature space is not explicitly defined and the kernel trick has to be used. The generated kernel matrix can be very large in practice. For instance, a moderate size image with $N = 10^5$ pixels will produce a 30GB kernel matrix in case that it is dense, leading to serious deterioration in both time and space complexity. Moreover, this kernel matrix has to be positive definite to guarantee convergence of iterative weighted K-means. These problems have limited the application of this algorithm in spite of its solid theoretical foundation. We will reveal that these problems can be efficiently solved by investigating the relationship between the inner product in the feature space and the similarity between image pixels. Superpixel segmentation results of different algorithms are compared in Figure 2.

3. Linear Spectral Clustering Superpixel

In this section, we will present the LSC superpixel segmentation algorithm which not only produces superpixels with state of the art boundary adherence but also captures global image properties. The LSC algorithm is proposed based on the investigation of the relationship between the objective functions of normalized cuts and weighted K-means. We find that optimizing these two objective functions are equivalent if the similarity between two points in the input space is equal to the weighted inner product between the two corresponding vectors in an elaborately designed high dimensional feature space. As such, simple weighted K-means clustering in this feature space can be used to replace the highly complex eigen-based method for minimizing the normalized cuts objective function. Comparing to the weighted kernel K-means clustering [7], LSC avoids the calculation of the large kernel matrix and the convergence condition can be naturally satisfied. By further limiting the search space of the weighted K-means, LSC achieves a linear complexity while retaining the high quality of the generated superpixels.

To facilitate the deduction, we briefly revisit the problem definitions of weighted K-means clustering and normalized cuts. For clarity, we use lowercase letters, such as \mathbf{p}, \mathbf{q} , to represent data points, or pixels in our case, to be clustered in the input space. In the weighted K-means clustering, each data point \mathbf{p} is assigned with a weight $w(\mathbf{p})$. Let K be the number of clusters; π_k be the k_{th} ($k = 1, 2, \dots, K$) cluster; and ϕ denote the function that maps data points to a higher dimensional feature space for improving linear separability. The objective function of weighted K-means is defined in (1), in which m_k is the center of π_k as is defined in (2). F_{k-m} can be efficiently minimized in an iterative manner.

$$F_{k-m} = \sum_{k=1}^K \sum_{\mathbf{p} \in \pi_k} w(\mathbf{p}) \|\phi(\mathbf{p}) - m_k\|^2 \quad (1)$$

$$m_k = \frac{\sum_{\mathbf{q} \in \pi_k} w(\mathbf{q}) \phi(\mathbf{q})}{\sum_{\mathbf{q} \in \pi_k} w(\mathbf{q})} \quad (2)$$

In normalized cuts, each data point corresponds to a node in a graph $G = (V, E, W)$ in which V is the set of all nodes; E is the set of edges; and W is a function characterizing similarity among data points. The K -way normalized cuts criterion is to maximize the objective function F_{Ncuts} defined in (3), in which $W(\mathbf{p}, \mathbf{q})$ stands for the similarity between two points \mathbf{p} and \mathbf{q} . Several solutions for solving this optimization problem have been proposed in [18][23][16]. All of these solutions are based on the eigenvalue decomposition of the large affinity matrix and are therefore intrinsically computationally complex.

$$F_{Ncuts} = \frac{1}{K} \sum_{k=1}^K \frac{\sum_{\mathbf{p} \in \pi_k} \sum_{\mathbf{q} \in \pi_k} W(\mathbf{p}, \mathbf{q})}{\sum_{\mathbf{p} \in \pi_k} \sum_{\mathbf{q} \in V} W(\mathbf{p}, \mathbf{q})} \quad (3)$$

By introducing a kernel matrix for mapping data points into a higher dimensional feature space, Dhillon et al. showed the connection between weighted K-means clustering and normalized cuts by rewriting the optimization of both F_{k-m} and F_{Ncuts} as the same matrix trace maximization problem [7]. Under such a formulation, the convergence of the iterative minimization of F_{k-m} can be guaranteed only when the kernel matrix is positive definite. However, this can not always be ensured. To solve this problem and further reveal the relationship between F_{k-m} and F_{Ncuts} , we present the following corollary. Equations (4) and (5) can also be deduced from the results in [7].

Corollary 1 *Optimization of the objective functions of weighted K-means and normalized cuts are mathematically equivalent if both (4) and (5) hold.*

$$\forall \mathbf{p}, \mathbf{q} \in V, w(\mathbf{p})\phi(\mathbf{p}) \cdot w(\mathbf{q})\phi(\mathbf{q}) = W(\mathbf{p}, \mathbf{q}) \quad (4)$$

$$\forall \mathbf{p} \in V, w(\mathbf{p}) = \sum_{\mathbf{q} \in V} W(\mathbf{p}, \mathbf{q}) \quad (5)$$

Equation (4) indicates that the weighted inner product of two vectors in the high dimensional feature space equals the similarity between the two corresponding points in the input space; and (5) indicates that the weight of each point in weighted K-means clustering equals the total weight of edges that connect the corresponding node to all the other nodes in normalized cuts. To prove **Corollary 1**, we first rewrite F_{k-m} as (6), in which $C = \sum_{k=1}^K \sum_{\mathbf{p} \in \pi_k} w(\mathbf{p}) \|\phi(\mathbf{p})\|^2$ is a constant independent of the clustering result. A detailed derivation of (6) can be found in the supplementary material.

Combining (4), (5) and (6), we have (7), from which it can be easily observed that minimizing F_{k-m} is strictly equivalent to maximizing F_{Ncuts} . In other words, by carefully constructing the high dimensional feature space defined by ϕ , the partitioning result of the normalized cuts should be identical to that of the weighted K-means clustering at their optimum points. This conclusion serves as the foundation of our LSC algorithm.

$$\begin{aligned}
F_{k-m} &= \sum_{k=1}^K \sum_{\mathbf{p} \in \pi_k} w(\mathbf{p}) \left\| \phi(\mathbf{p}) - \frac{\sum_{\mathbf{q} \in \pi_k} w(\mathbf{q}) \phi(\mathbf{q})}{\sum_{\mathbf{q} \in \pi_k} w(\mathbf{q})} \right\|^2 \\
&= \sum_{k=1}^K \sum_{\mathbf{p} \in \pi_k} w(\mathbf{p}) \|\phi(\mathbf{p})\|^2 - \sum_{k=1}^K \frac{\|\sum_{\mathbf{p} \in \pi_k} w(\mathbf{p}) \phi(\mathbf{p})\|^2}{\sum_{\mathbf{p} \in \pi_k} w(\mathbf{p})} \\
&= C - \sum_{k=1}^K \frac{\sum_{\mathbf{p} \in \pi_k} \sum_{\mathbf{q} \in \pi_k} w(\mathbf{p}) \phi(\mathbf{p}) \cdot w(\mathbf{q}) \phi(\mathbf{q})}{\sum_{\mathbf{p} \in \pi_k} w(\mathbf{p})} \quad (6)
\end{aligned}$$

Among the two sufficient conditions of **Corollary 1**, (5) can be easily fulfilled by using the sum of edge weights in normalized cuts as the point weight in weighted K-means.

$$\begin{aligned}
F_{k-m} &= C - \sum_{k=1}^K \frac{\sum_{\mathbf{p} \in \pi_k} \sum_{\mathbf{q} \in \pi_k} \mathbf{W}(\mathbf{p}, \mathbf{q})}{\sum_{\mathbf{p} \in \pi_k} \sum_{\mathbf{q} \in \mathbf{V}} \mathbf{W}(\mathbf{p}, \mathbf{q})} \\
&= C - K \times F_{Ncuts} \quad (7)
\end{aligned}$$

Fulfilling (4), however, requires a careful selection of the similarity function \mathbf{W} . Equation (4) can be rewritten as (8), in which the left hand side is the inner product of two vectors in the high dimensional feature space. In fact, (8) can also be considered as defining a symmetric kernel function, indicating that it must satisfy the positivity condition [6]. Also, to avoid the kernel matrix, \mathbf{W} must be separable to allow an explicit expression of the mapping function ϕ .

$$\phi(\mathbf{p}) \cdot \phi(\mathbf{q}) = \frac{\mathbf{W}(\mathbf{p}, \mathbf{q})}{w(\mathbf{p})w(\mathbf{q})} \quad (8)$$

In order to find a suitable form for $\mathbf{W}(\mathbf{p}, \mathbf{q})$, we first investigate the widely used Euclidean distance based pixel similarity measurement. For each pixel in a color image, we represent it using a five dimensional vector (l, α, β, x, y) , in which l, α, β are its color component values in the CIELAB color space; and x, y are the vertical and horizontal coordinates in the image plane. Without loss of generality, the range of each component is linearly normalized to $[0, 1]$ for simplicity. The CIELAB color space is adopted because it is believed that the Euclidean distance is nearly perceptually uniform in this space [1]. Given two pixels $\mathbf{p} = (l_p, \alpha_p, \beta_p, x_p, y_p)$ and $\mathbf{q} = (l_q, \alpha_q, \beta_q, x_q, y_q)$, a similarity measurement between them can be defined as (9), in which $\widehat{\mathbf{W}}_c$ and $\widehat{\mathbf{W}}_s$ are used to measure color similarity and space proximity respectively. Two parameters C_c and C_s are used to control the relative significance of color and spatial information. We multiply the first term of $\widehat{\mathbf{W}}_c(\mathbf{p}, \mathbf{q})$ with 2.55^2 in order to be consistent with the standard CIELAB definition.

$$\begin{aligned}
\widehat{\mathbf{W}}(\mathbf{p}, \mathbf{q}) &= C_c^2 \cdot \widehat{\mathbf{W}}_c(\mathbf{p}, \mathbf{q}) + C_s^2 \cdot \widehat{\mathbf{W}}_s(\mathbf{p}, \mathbf{q}) \\
\widehat{\mathbf{W}}_c(\mathbf{p}, \mathbf{q}) &= 2.55^2 [2 - (\alpha_p - \alpha_q)^2 - (\beta_p - \beta_q)^2] \\
&\quad + [1 - (l_p - l_q)^2] \\
\widehat{\mathbf{W}}_s(\mathbf{p}, \mathbf{q}) &= [2 - (x_p - x_q)^2 - (y_p - y_q)^2] \quad (9)
\end{aligned}$$

Although $\widehat{\mathbf{W}}(\mathbf{p}, \mathbf{q})$ has very clear physical meaning in measuring pixel similarity, it cannot be directly used in our method because it does not satisfy the positivity condition [6] required by (8). Detailed explanation can be found in the supplementary material. To solve this problem, we try to find a proper approximation of $\widehat{\mathbf{W}}(\mathbf{p}, \mathbf{q})$.

$$\begin{aligned}
\widehat{\mathbf{W}}(\mathbf{p}, \mathbf{q}) &= C_s^2 [g(x_p - x_q) + g(y_p - y_q)] + C_c^2 [g(l_p - l_q) + 2.55^2 (g(\alpha_p - \alpha_q) + g(\beta_p - \beta_q))] \\
g(t) &= 1 - t^2, \quad t \in [-1, 1] \quad (10)
\end{aligned}$$

We rewrite (9) as (10) to show that $\widehat{\mathbf{W}}(\mathbf{p}, \mathbf{q})$ is a nonnegative linear combination of a number of instances of a simple functions $g(t)$, which can be expanded as a uniformly convergent Fourier series shown in (11). The coefficients of this series converge to 0 very quickly at a speed of $(2k+1)^3$. Therefore, $g(t)$ can be well approximated by the first term in the series as is expressed in (12).

$$g(t) = \sum_{k=0}^{\infty} \frac{32(-1)^k}{[(2k+1)\pi]^3} \cos\left(\frac{(2k+1)\pi t}{2}\right), \quad t \in [-1, 1] \quad (11)$$

$$g(t) = 1 - t^2 \approx \frac{32}{\pi} \cos \frac{\pi}{2} t, \quad t \in [-1, 1] \quad (12)$$

Simply omitting the constant multiplier $32/\pi$, $\widehat{\mathbf{W}}(\mathbf{p}, \mathbf{q})$ can be approximated by $\mathbf{W}(\mathbf{p}, \mathbf{q})$ defined in (13). Unlike $g(t)$, $\cos \frac{\pi}{2} t$ is positive definite, leading to the positivity of $\mathbf{W}(\mathbf{p}, \mathbf{q})$. Actually, according to the properties of cosine function, $\mathbf{W}(\mathbf{p}, \mathbf{q})$ can be directly written in the inner product form shown in (4), in which ϕ and w are defined in (14).

$$\begin{aligned}
\mathbf{W}(\mathbf{p}, \mathbf{q}) &= C_s^2 \left[\cos \frac{\pi}{2} (x_p - x_q) + \cos \frac{\pi}{2} (y_p - y_q) \right] \\
&\quad + C_c^2 \left[\cos \frac{\pi}{2} (l_p - l_q) + 2.55^2 (\cos \frac{\pi}{2} (\alpha_p - \alpha_q) + \cos \frac{\pi}{2} (\beta_p - \beta_q)) \right] \quad (13)
\end{aligned}$$

$$\begin{aligned}
\phi(\mathbf{p}) &= \frac{1}{w(\mathbf{p})} (C_c \cos \frac{\pi}{2} l_p, C_c \sin \frac{\pi}{2} l_p, 2.55 C_c \cos \frac{\pi}{2} \alpha_p, \\
&\quad 2.55 C_c \sin \frac{\pi}{2} \alpha_p, 2.55 C_c \cos \frac{\pi}{2} \beta_p, 2.55 C_c \sin \frac{\pi}{2} \beta_p, \\
&\quad C_s \cos \frac{\pi}{2} x_p, C_s \sin \frac{\pi}{2} x_p, C_s \cos \frac{\pi}{2} y_p, C_s \sin \frac{\pi}{2} y_p) \\
w(\mathbf{p}) &= \sum_{\mathbf{q} \in \mathbf{V}} \mathbf{W}(\mathbf{p}, \mathbf{q}) = w(\mathbf{p}) \phi(\mathbf{p}) \cdot \sum_{\mathbf{q} \in \mathbf{V}} w(\mathbf{q}) \phi(\mathbf{q}) \quad (14)
\end{aligned}$$

Until now, we have explicitly define a ten dimensional feature space in (14) such that weighted K-means clustering in this feature space is approximately equivalent to normalized cuts in the input space. Noticing that under the similarity function defined in (13), both the kernel matrix for

weighted kernel K-means and the affinity matrix in the normalized cuts will be highly dense, leading to high computational complexity when using existing methods. In contrast, by directly applying weighted K-means in the ten dimensional feature space, the objective function of the normalized cuts can be efficiently optimized.

Based on the above analysis, we propose the LSC superpixel segmentation algorithm which takes as input the desired number of superpixels, K . In LSC, image pixels are first mapped to weighted points in the ten dimensional feature space defined by (14). K seed pixels are then sampled uniformly over the whole image with horizontal and vertical intervals v_x and v_y , while v_x/v_y equals the aspect ratio of the image. After slight disturbances for avoiding noisy and boundary pixels [1], these seeds are used as the search centers and their feature vectors are used as initial weighted means of the corresponding clusters. Each pixel is then assigned to the cluster of which the weighted mean is closest to the pixel's vector in the feature space. After pixel assignment, the weighted mean and search center of each cluster will be updated accordingly. The above two steps are iteratively performed until convergence. Pixels assigned to the same cluster form a superpixel.

Theoretically, the search space of each cluster should cover the whole image to satisfy **Corollary 1**. However, for superpixels, local compactness is a common prior. In other words, it may not be favorable to assign pixels far away from each other to the same superpixel in term of human perception. Hence, we adopt the common practice [20][1] in superpixel segmentation by limiting the search space of each cluster to the size of $\tau v_x \times \tau v_y$, in which $\tau \geq 1$ is a parameter for balancing local compactness and global optimality. We simply choose $\tau = 2$ for implementation.

The above process offers no enforcement on the connectivity of superpixels, meaning that there is no guarantee that pixels in the same cluster form a connected component. To address this problem, we empirically merge small isolated superpixels which are less than one fourth of the expected superpixel size to their large neighboring superpixels. When there are more than one candidates for merging, we choose the closest one in the ten dimensional feature space. The algorithm is summarized in **Algorithm 1**.

Suppose the number of image pixels is N . The complexity of the feature mapping is obviously $\mathcal{O}(N)$. By restricting the search space of each cluster, the complexity of pixel assignment is reduced from $\mathcal{O}(KN)$ to $\mathcal{O}(N)$ in each iteration. The complexity of updating the weight means and search centers is also $\mathcal{O}(N)$. The merging step requires $\mathcal{O}(nz)$ operations, in which z represents the number of small isolated superpixels to be merged and n is the average number of their adjacent neighbors. As such, the overall complexity of LSC is $\mathcal{O}(\kappa N + nz)$, in which κ is the number of iterations. In practice, $nz \ll NI$ and $\kappa = 20$

Algorithm 1 LSC Superpixel Segmentation

- 1: Map each point $\mathbf{p} = (l_{\mathbf{p}}, \alpha_{\mathbf{p}}, \beta_{\mathbf{p}}, x_{\mathbf{p}}, y_{\mathbf{p}})$ to a ten dimensional vector $\phi(\mathbf{p})$ in the feature space.
 - 2: Sampling K seeds over the image uniformly at fixed horizontal and vertical intervals v_x and v_y .
 - 3: Move each seed to its lowest gradient neighbor in the 3×3 neighborhood.
 - 4: Initialize weighted mean m_k and search center c_k of each cluster using the corresponding seed.
 - 5: Set label $L(\mathbf{p}) = 0$ for each point \mathbf{p} .
 - 6: Set distance $D(\mathbf{p}) = \infty$ for each point \mathbf{p} .
 - 7: **repeat**
 - 8: **for** each weighted means m_k and search center c_k **do**
 - 9: **for** point \mathbf{p} in the $\tau v_x \times \tau v_y$ neighborhood of c_k in the image plane **do**
 - 10: $D =$ Euclidean distance between $\phi(\mathbf{p})$ and m_k in the feature space.
 - 11: **if** $D < d(\mathbf{p})$ **then**
 - 12: $d(\mathbf{p}) = D$
 - 13: $L(\mathbf{p}) = k$
 - 14: **end if**
 - 15: **end for**
 - 16: **end for**
 - 17: Update weighted means and search centers for all clusters.
 - 18: **until** weighted means of K cluster converge.
 - 19: Merge small superpixels to their neighbors.
-

will be enough for generating superpixels with the state of art quality. Therefore, LSC is of a linear complexity $\mathcal{O}(N)$ and experiments will show that LSC is among the fastest superpixel segmentation algorithms.

4. Experiments

We compare LSC to eight state of the art superpixel segmentation algorithms including SLIC [1], SEEDS [2], Neuts [23], Lattice [15], ERS [12], Turbopixel [11], EneOpt1 and EneOpt0 [20]. For all the eight algorithms, the implementations are based on publicly available code. Experiments are performed on the Berkeley Segmentation Database [13] consisting of three hundred test images with human segmented ground truth. The boundary adherence of superpixels generated by different algorithms are compared using three commonly used evaluation metrics in image segmentation: under-segmentation error (**UE**), boundary recall (**BR**) and achievable segmentation accuracy (**ASA**).

Among the three metrics, **UE** measures the percentage of pixels that leak from the ground truth boundaries. It actually evaluates the quality of superpixel segmentation by penalizing superpixels overlapping with multiple objects. The definition of **UE** used in [1] is adopted here. Lower **UE** indicates that fewer superpixels straddle multiple objects. **BR**

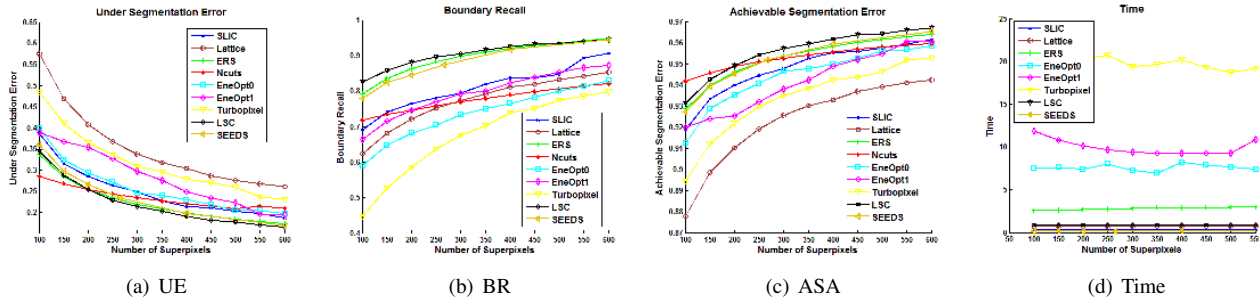


Figure 3. Quantitative evaluation of different superpixel segmentation algorithms.

measures the fraction of ground truth boundaries correctly recovered by the superpixel boundaries. A true boundary pixel is regarded to be correctly recovered if it falls within 2 pixels from at least one superpixel boundary point. A high **BR** indicates that very few true boundaries are missed. **ASA** is defined as the highest achievable object segmentation accuracy when utilizing superpixel as units [12]. By labeling each superpixel with the ground truth segments of the largest overlapping area, **ASA** is calculated as the fraction of labeled pixels that are not leaked from the ground truth boundaries. A high **ASA** indicates that the superpixels comply well with objects in the image. Figure 3 shows the experimental results which are average values over all the 300 test images in the Berkeley segmentation database.

Computational efficiency is also an important factor for evaluating the performance of superpixel segmentation algorithms. In our experiment, we calculate the average running time for different algorithms and the results are shown in Figure 3(d). All the experiments are performed on a desktop PC equipped with an Intel 3.4 GHz dual core processor and 2GB memory. The time consumption of the Ncuts algorithm [23] is much higher than that of the other methods and is therefore omitted in Figure 3(d).

To be more clear, we also list the numeric values of the metrics when the number of superpixels $K = 400$ in Table 1 which also summarizes the computational complexity of different algorithms. From Figure 3 and Table 1, it can be observed that in terms of the boundary adherence, the proposed LSC is comparable to the state of art the algorithms. For relative large number of superpixels, LSC performs the best. Also, LSC is of linear complexity and is among the algorithms with the highest time efficiency. In addition, qualitative experiments demonstrate that LSC performs the best. We select the five algorithms (SEEDS, Ncuts, SLIC, ERS and LSC) that achieve the lowest **UE** values when $K = 400$ for visual comparison. According to Figure 3, these five algorithms generally outperform the remaining three algorithms in terms of **UE**, **BR** as well as **ASA**. Figure 4 shows some typical visual results of superpixel segmentation using these algorithms. Some detail segmentation results are emphasized to facilitate close visual inspection. Intuitively,

LSC has achieved the most perceptually satisfactory segmentation results for different kind of images.

According to Figure 3, EneOpt0 performs the worst in terms of boundary adherence among the five selected algorithms, probably because that it uses a variation of the minimum cut strategy which may suffer from the bias of cutting out small set of pixels, leading to under segmentation errors in practice as is shown in Figure 4(a). Actually, EneOpt0 indirectly controls the superpixel density by setting the upper bound of the superpixel size. However, it may be difficult to produce a desirable number of superpixels using EneOp0 especially for small K because large superpixels tend to be split into small patches at regions with high variabilities. As for Ncuts, a major drawback is its extremely low time efficiency. The two-step heuristic algorithm proposed in [17] for acceleration has caused Ncuts to become ineffective in terms of boundary adherence as K increases. Even though, Ncuts is still the slowest algorithm as is shown in Table 1. As a local feature based method, SLIC is the second fastest among the selected algorithms according to our experimental results. The superpixels generated by SLIC are also perceptually satisfactory for most of the cases. However, compared to the proposed LSC algorithm, the boundary adherence of SLIC is less competitive according to Figure 3. Actually, the major difference between SLIC and LSC is that the iterative weighted K-means clustering is performed inside different feature spaces. However, this difference is critical because unlike SLIC which rely on local features only, LSC successfully connects a local feature based operation with a global optimization objective function by introducing ϕ so that the global image structure is implicitly utilized to generate more reasonable segmentation results. In terms of boundary adherence, ERS and SEEDS are very close to LSC and SEEDS is probably the fastest existing superpixel segmentation algorithm. However, this is achieved by sacrificing the regularity and perceptual satisfaction of the generated superpixels as is shown in Figure 4(d).

LSC uses two parameters C_s and C_c to control the relative significance of the color similarity and space proximity in measuring similarity between pixels. In fact, what is truly meaningful is the their ratio $r_c = C_s/C_c$. Generally,

Table 1. Performance metrics of superpixel segmentation algorithms at $K = 400$

| | EneOpt0 | SEEDS | ERS | Lattices | Ncuts | SLIC | Turbo | LSC |
|---|--------------------------------|------------------|--------------------------|--------------------------------------|--------------------------------|------------------|------------------|------------------|
| Adherence to boundaries | | | | | | | | |
| <i>Under segmentation error</i> | 0.230 | 0.197 | 0.198 | 0.303 | 0.220 | 0.213 | 0.277 | 0.190 |
| <i>Boundary recall</i> | 0.765 | 0.918 | 0.920 | 0.811 | 0.789 | 0.837 | 0.739 | 0.926 |
| <i>Achievable segmentation accuracy</i> | 0.950 | 0.960 | 0.959 | 0.933 | 0.956 | 0.956 | 0.943 | 0.962 |
| Segmentation speed | | | | | | | | |
| <i>Computational complexity</i> | $\mathcal{O}(\frac{N^3}{K^2})$ | $\mathcal{O}(N)$ | $\mathcal{O}(N^2 \lg N)$ | $\mathcal{O}(N^{\frac{3}{2}} \lg N)$ | $\mathcal{O}(N^{\frac{2}{3}})$ | $\mathcal{O}(N)$ | $\mathcal{O}(N)$ | $\mathcal{O}(N)$ |
| <i>Average time per image</i> | 8.22s | 0.213s | 2.88s | 0.748s | 273s | 0.314s | 20.2s | 0.919s |

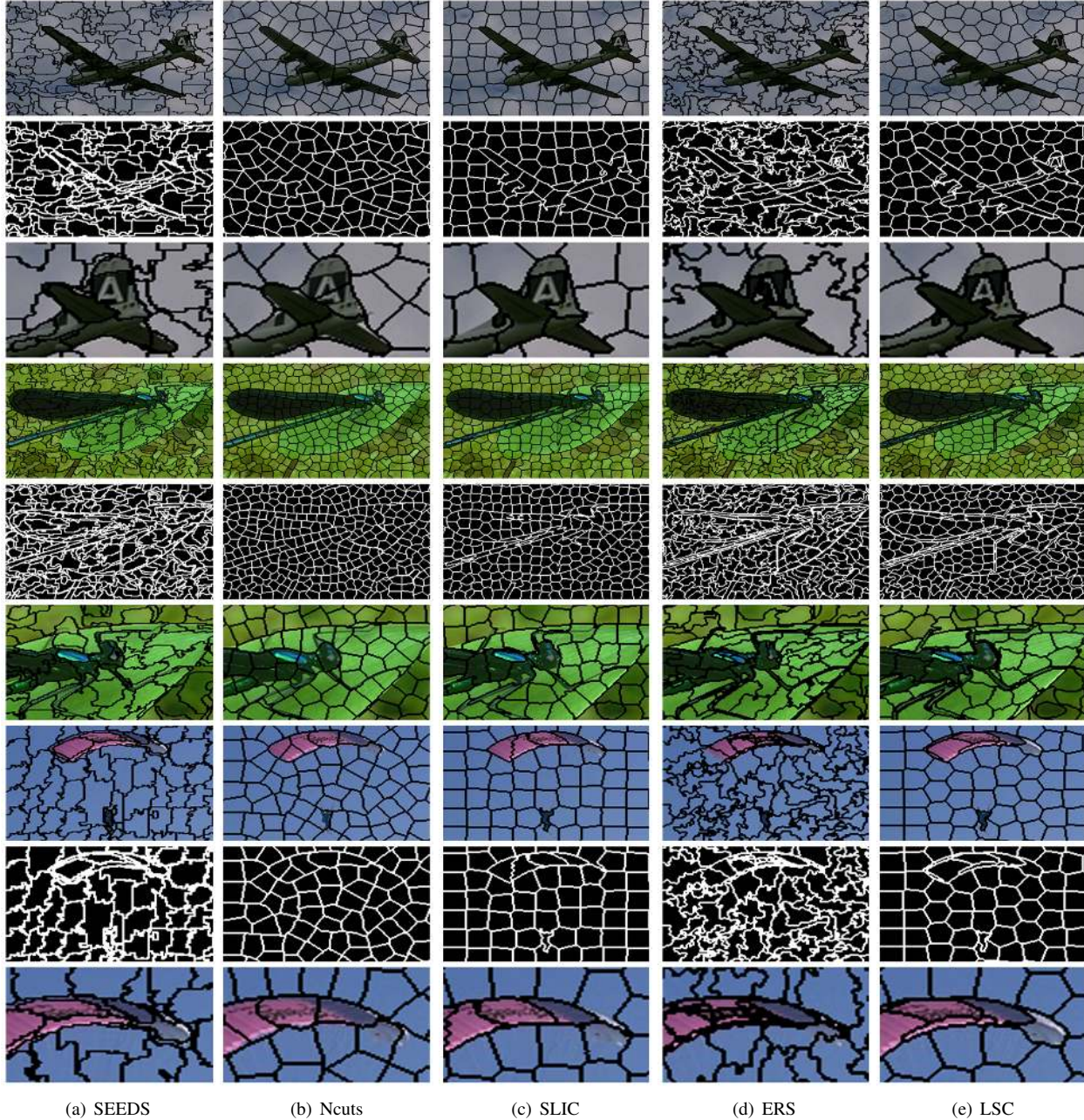
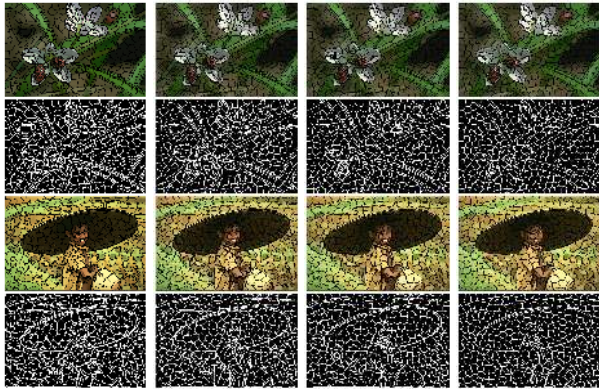


Figure 4. Visual comparison of superpixel segmentation results when $K = 400$.



(a) $r_c = 0.05$ (b) $r_c = 0.075$ (c) $r_c = 0.1$ (d) $r_c = 0.15$

Figure 5. Superpixel segmentation results for different r_c .

larger values of r_c indicate a tendency towards generating superpixels with higher shape regularity while smaller r_c usually lead to better boundary adherence as is shown in Figure 5. In our experiments, we set $r_c = 0.075$. We have also verified the validity of approximating $\widehat{W}(p, q)$ using $W(p, q)$. For more than 98.8% practical cases, the relative error caused by this approximation does not exceed 0.5%.

5. Conclusions

We present in this paper a novel superpixel segmentation algorithm, LSC, which produces compact and regular shaped superpixels with linear time complexity and high memory efficiency. The most critical idea in LSC is to explicitly utilize the connection between the optimization objectives of weighted K-means and normalized cuts by introducing an elaborately designed high dimensional space. As such, LSC achieves both boundary adherence and global image structure perseverance through simple local feature based operations. Experimental results show that LSC generally over-performs most state of the art algorithms both quantitatively and qualitatively.

This work was supported by the Beijing Higher Education Young Elite Teacher Project (YETP0104), the Tsinghua University Initiative Scientific Research Program (20131089382), and the National Natural Science Foundation of China (61101152),.

References

- [1] 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. on PAMI*, 34(11):2274–2281, 2012. 2, 4, 5
- [2] M. Bergh, X. Boix, G. Roig, B. Capitani, and L. V. Gool. Seeds: Superpixels extracted via energy-driven sampling. *Proc. of ECCV*, 7578:13–26, 2012. 2, 5
- [3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. on PAMI*, 26(9):1124–1137, 2001. 2
- [4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on PAMI*, 23(11):1222–1239, 2001. 2
- [5] D. Comaniciu and P. Meer. Mean shift: a robust approach towards feature space analysis. *IEEE Trans. on PAMI*, 24(5):603–619, 2002. 1, 2
- [6] N. Cristianini and J. Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press New York, NY, USA, 2000. 4
- [7] I. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors: a multilevel approach. *IEEE Trans. on PAMI*, 29(11):1944–1957, 2007. 2, 3
- [8] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004. 1, 2
- [9] D. Hoiem, A. Efros, and M. Hebert. Automatic photo pop-up. *ACM Trans. on Graphics*, 24(3):577–584, 2005. 1
- [10] V. Kolmogorov. What energy functions can be minimized via graph cuts? *IEEE Trans. on PAMI*, 26(2):147–159, 2004. 2
- [11] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi. Turbopixel: fast superpixels using geometric flow. *IEEE Trans. on PAMI*, 31(12):2209–2297, 2009. 1, 2, 5
- [12] M. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy rate superpixel segmentation. *Proc. of CVPR*, pages 2097–2104, 2011. 1, 2, 5, 6
- [13] 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. *Proc. of ICCV*, 2:416–423, 2001. 1, 2, 5
- [14] A. Moore, S. Prince, and J. Warrell. Lattice cut - constructing superpixels using layer constraints. *Proc. of CVPR*, pages 2117–2124, 2010. 2
- [15] A. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones. Superpixel lattices. *Proc. of CVPR*, pages 1–8, 2008. 2, 5
- [16] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. *Proc. of NIPS*, pages 849–856, 2001. 3
- [17] X. Ren and J. Malik. Learning a classification model for segmentation. *Proc. of ICCV*, 1:10–17, 2003. 1, 2, 6
- [18] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on PAMI*, 22(8):888–905, 2000. 2, 3
- [19] J. Tighe and S. Lazebnik. Superparsing: scalable non parametric image parsing with superpixel. *Proc. of ECCV*, 5:352–365, 2010. 1
- [20] O. Veksler, Y. Boykov, and P. Mehrani. Superpixels and supervoxels in an energy optimization framework. *Proc. of ECCV*, pages 211–224, 2010. 2, 5
- [21] A. Veldadi and S. Soatto. Quick shift and kernel methods for mode seeking. *Proc. of ECCV*, pages 705–718, 2008. 1, 2
- [22] S. Wang, H. Lu, F. Yang, and M. Yang. Superpixel tracking. *Proc. of ICCV*, 1:1323–1330, 2011. 1, 2
- [23] S. Yu and J. Shi. Multiclass spectral clustering. *Proc. of ICCV*, 1:313–319, 2003. 3, 5, 6