Computer Vision and Image Understanding 146 (2016) 1–8

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Subsampling-based acceleration of simple linear iterative clustering for superpixel segmentation☆

Kang-Sun Choi*, Ki-Won Oh

Interdisciplinary Program in Creative Engineering, KOREATECH, 1600 Chungjeol-ro, Byeongchun, Cheonan, Chungnam 330-708, Republic of Korea

ARTICLE INFO

Article history:
Received 22 June 2015
Accepted 29 February 2016
Available online 8 March 2016

Keywords:
Superpixels
Segmentation
Acceleration
Fast implementation
Subsampling

ABSTRACT

Simple linear iterative clustering (SLIC) that partitions an image into multiple homogeneous regions, superpixels, has been widely used as a preprocessing step in various image processing and computer vision applications due to its outstanding performance in terms of speed and accuracy. However, determining a segment that each pixel belongs to still requires tedious, iterative computation, which hinders real-time execution of SLIC. In this paper, we propose an accelerated SLIC superpixel segmentation algorithm where the number of candidate segments for each pixel is reduced effectively by exploiting high spatial redundancy within natural images. Because all candidate segments should be inspected in order to choose the best one, candidate reduction significantly improves computational efficiency. Various characteristics of the proposed acceleration algorithm are investigated. The experimental results confirmed that the proposed superpixel segmentation algorithm runs up to about five times as fast as SLIC while producing almost the same superpixel segmentation performance, sometimes better than SLIC, with respect to under-segmentation error and boundary recall.

1. Introduction

Superpixel segmentation is the process of partitioning an image into multiple segments, so-called superpixels, which are homogeneous in the sense that pixels within each segment are similar with respect to certain characteristics such as color and texture. Although superpixel segmentation usually yields over-segmented results rather than object-level segments, it drastically reduces the number of image primitives with minimal loss of information and offers an easy way to extract the most likely image objects with as few segments as possible. In addition, since superpixel segmentation provides a more natural and perceptually meaningful representation of the input image, it is more convenient and effective to extract region-based visual features using superpixels [1]. This is why superpixel segmentation is widely used as a preprocessing step for various computer vision applications like foreground object detection [2,3], motion segmentation [4], object recognition [5], depth map enhancement [6,7], and saliency object detection [1].

After the superpixel concept was originally presented by Ren and Malik [8] as defining perceptually homogeneous regions using the normalized cuts (NCuts) algorithm, numerous superpixel segmentation algorithms have been proposed in the literature, for example, the mean shift algorithm [9], graph-based methods [10,11], Turbopixels [12], and simple linear iterative clustering (SLIC) [2]. Although each method has its own advantages and drawbacks that make it better suited for a particular application, SLIC, proposed by Achanta et al., has attracted a great deal of attention and has been utilized [3,6,11]. As summarized in the literature [2], SLIC outperforms the other algorithms in several desirable properties for superpixel segmentation; SLIC is capable of quickly creating compact and uniform superpixels, offers explicit control over the amount of superpixels and their compactness, and accurately aligns superpixels with object boundaries.

Thanks to the high performance of SLIC in execution time, accuracy, and compactness, SLIC has been widely employed in a wide range of practical computer vision applications. Therefore, in this paper, we only focus on how to further enhance the-state-of-the-art, SLIC.

SLIC is an iterative algorithm based on *k*-means clustering in which the best cluster for each sample is sought from among all available clusters. Because the cluster search involves an inspection process requiring distance calculations and comparisons per cluster, *k*-means clustering usually requires a huge amount of repeated computation. In contrast, a high speed-up is achieved in SLIC, such that the best cluster (i.e. the best segment for each pixel) is chosen from among only a few spatially neighboring segments, rather

☆ This paper has been recommended for acceptance by Olga Veksler.

* Corresponding author. fax: +82 41 564 3261.
E-mail address: ks.choi@koreatech.ac.kr (K.-S. Choi).
URL: http://cv3vpl.koreatech.ac.kr (K.-S. Choi)

http://dx.doi.org/10.1016/j.cviu.2016.02.018

than all segments in the image. Such a modification to k-means clustering, i.e. the reduction of candidate clusters, achieves drastic computation reduction.

Nevertheless, considering that superpixel segmentation is usually used as a preprocessing step in image processing and computer vision applications, computational challenges should be further addressed for practical real-time implementation of the applications. However, there have been, ironically, few efforts to further improve conventional SLIC due to the aforementioned somewhat-high performance improvement of SLIC in execution time, accuracy, and compactness.

Kim et al. attempted to improve segmentation accuracy with respect to boundary adherence by updating cluster representatives only with pixels having a similar luminance [13]. In Ref. [14], a GPU-based parallel implementation of SLIC was presented to speed up execution time. In Ref. [15], Borovec and Kybic presented an implementation of SLIC where repetitive distance calculation is simplified by using pre-computed look-up tables (LUTs). However, LUTs usually increase memory usage by a huge amount.

In this paper, we propose an accelerated SLIC superpixel segmentation algorithm that significantly reduces processing time without hardware parallelism while producing almost the same segmentation results as SLIC.

In the proposed algorithm, the number of candidate clusters is reduced further by exploiting interpixel redundancy in the images. The cluster search is first performed on subsampled pixels. Based on clustering results for the subsampled pixels, we can sift candidate clusters for the remaining pixels and successfully get rid of implausible candidates without any inspection computation. Consequently, the number of candidate clusters for the cluster search lessens without loss of segmentation accuracy.

This paper is organized as follows. In the next section, we review the conventional SLIC algorithm. In Section 3, both the proposed acceleration approaches are explicated in detail. Performance evaluation and comparison of the proposed algorithms are presented in Section 4. Finally, the conclusion is given in Section 5.

2. Simple linear iterative clustering

Before describing the proposed accelerated SLIC algorithm in detail, we review the conventional SLIC algorithm [2] after defining some notations.

A pixel i in an input image $\mathbf{I}$ of size $L \times M = N$ is represented as a feature vector consisting of both its color in the CIELAB color space and its position: $\mathbf{f}_i = [l_i, a_i, b_i, x_i, y_i]^T$. Similarly, cluster C_k is represented with the mean color and the center of mass of the cluster: $\mathbf{f}_{C_k} = [l_{C_k}, a_{C_k}, b_{C_k}, x_{C_k}, y_{C_k}]^T$. Let $\mathcal{D}(i, C_k)$ denote a distance metric between i and C_k, which is defined as

$$\mathcal{D}(i, C_k) = \sqrt{\mathcal{D}_c(i, C_k)^2 + \lambda^2 \cdot \mathcal{D}_s(i, C_k)^2}, \tag{1}$$

where $\lambda = r/S$, and r and S represent a regularization factor controlling compactness and an initial clustering sampling interval, respectively. In [2], a default value of 10 was used for r.

$$\mathcal{D}_c(i, C_k) = \sqrt{\left(l_i - l_{C_k}\right)^2 + \left(a_i - a_{C_k}\right)^2 + \left(b_i - b_{C_k}\right)^2} \tag{2}$$

and

$$\mathcal{D}_s(i, C_k) = \sqrt{\left(x_i - x_{C_k}\right)^2 + \left(y_i - y_{C_k}\right)^2} \tag{3}$$

indicate Euclidean distances in color and spatial domains, respectively.

The procedure of the conventional SLIC algorithm is described below. SLIC consists of four steps; initialization, cluster assignment, update, and postprocessing.

For color images in the CIELAB color space, K cluster centers are initially sampled on a regular grid S. To prevent the initial cluster centers from being located on object boundaries, the initial cluster centers are re-located to the lowest gradient position within their 3×3 neighborhood.

In the cluster assignment step, each pixel i is associated with the nearest cluster center, in the sense of (1), by inspecting neighboring cluster centers whose search regions overlap the position of i. Here, cluster inspection involves a distance calculation in (1) and a comparison. In [2], the size of the search region was set to $2S \times 2S$ around the cluster center.

Let Ω_i, $\mathbf{L}(i)$, and Π_{C_k} denote a set of cluster centers whose search regions contain i, a label associated with i, and a set of pixels constituting C_k, respectively. That is, $\mathbf{L}(i) = C_k \in \Omega_i$ and $\Pi_{C_k} = \{i | \mathbf{L}(i) = C_k\}$.

Even though K clusters exist in the image, the cluster inspection for each pixel i is performed only with neighboring candidate clusters. The number of candidate clusters to be inspected, $|\Omega_i|$, is usually much smaller than K, which leads to a significant speed advantage over the conventional k-means clustering, where each pixel is compared with all K clusters.

In the update step, each $\mathbf{f}_{C_k}$ is updated in accordance with the Π_{C_k} that is determined in the cluster assignment step. The cluster assignment and update steps are repeated until the segmentation converges. At the end of this clustering procedure, orphaned pixels forming small-sized clusters are merged into the nearest cluster to enforce connectivity.

3. Proposed accelerated simple linear iterative clustering algorithm

3.1. Acceleration exploiting interpixel redundancy of images

Because images are generally characterized by large regions of constant or near-constant pixel values, there is considerable spatial correlation between adjacent pixels, so-called *interpixel redundancy*, which is the most important characteristic of images for image compression.

In the context of superpixel segmentation, interpixel redundancy implies that if a pixel i is associated with a certain cluster, its neighboring pixels also highly tend to belong to the cluster. In order to exploit interpixel redundancy, in the proposed algorithm, a subsampling method is applied to the SLIC cluster assignment step.

For clear notation, let Λ and Λ^C be a set of 2:1 subsampled pixels, i.e. every second pixel, both horizontally and vertically, and the remaining pixels except Λ, respectively. Therefore, the number of pixels belonging to Λ, $|\Lambda|$, becomes $N/4$, and $|\Lambda^C|$ becomes $(3N)/4$.

Specifically, after the SLIC initialization step is executed, pixels on Λ are first associated with their closest clusters through the cluster search, as in SLIC. At that point, the remaining pixels in Λ^C have not been associated with any cluster yet. However, it is guaranteed that, for a pixel $i \in \Lambda^C$, some of its 8-connected neighboring pixels are already labeled. Letting Φ_i denote the set of 8-connected neighboring pixels around i, the cluster search for $i \in \Lambda^C$ is performed with the candidate clusters in $\Omega_{\Phi_i} = \{\mathbf{L}(j) | j \in \Lambda \cap \Phi_i\}$ in the proposed algorithm. It is noteworthy that $|\Omega_{\Phi_i}|$ is usually less than $|\Omega_i|$. In consequence, computation reduction on cluster inspection can be achieved.

The process of the proposed cluster assignment algorithm using interpixel redundancy is demonstrated in Fig. 1, where small squares indicate pixels and the subsampled grid Λ is represented as thick outlined squares. As described before, for $i \in \Lambda$, Ω_i contains all the clusters within a region of $2S \times 2S$, and thus, becomes $\{C_1, C_2, C_3, C_4\}$ in this example.

Assume that the closest clusters for the subsampled pixels are determined as marked inside the pixels. Then, for the in-between yellow pixels (except the central one), Ω_{Φ_i} can have two different