SLIC- simple linear iterative clustering

Abstract—Computer vision applications have come to rely increasingly on superpixels in recent years, but it is not always clear what constitutes a good superpixel algorithm. In an effort to understand the benefits and drawbacks of existing methods, we empirically compare five state-of-the-art superpixel algorithms for their ability to adhere to image boundaries, speed, memory efficiency, and their impact on segmentation performance. We then introduce a new superpixel algorithm, simple linear iterative clustering (SLIC), which adapts a k-means clustering approach to efficiently generate superpixels. Despite its simplicity, SLIC adheres to boundaries as well as or better than previous methods. At the same time, it is faster and more memory efficient, improves segmentation performance, and is straightforward to extend to supervoxel generation.

摘要:近年来，计算机视觉应用越来越依赖于超像素，但什么是好的超像素算法并不总是很清楚。为了了解现有方法的优缺点，我们通过经验比较了五种最先进的超像素算法在坚持图像边界的能力、速度、内存效率以及它们对分割性能的影响。然后，我们引入了一种新的超像素算法，简单线性迭代聚类(SLIC)，它采用k-means聚类方法来高效地生成超像素。尽管SLIC很简单，但它遵循的边界和以前的方法一样好，甚至更好。同时，它更快，内存更高效，提高分割性能，并直接扩展到超体素生成。

1 INTRODUCTION

SUPERPIXEL algorithms group pixels into perceptually meaningful atomic regions which can be used to replace the rigid structure of the pixel grid (Fig. 1). They capture image redundancy, provide a convenient primitive from which to compute image features, and greatly reduce the complexity of subsequent image processing tasks. They have become key building blocks of many computer vision algorithms, such as top scoring multiclass object segmenta- tion entries to the PASCAL VOC Challenge [9], [29], [11], depth estimation [30], segmentation [16], body model estimation [22], and object localization [9].

超像素算法将像素分组成具有感知意义的原子区域，用于取代像素网格的刚性结构(图1)。超像素算法捕捉图像冗余，提供一个方便的原语来计算图像特征，并大大降低了后续图像处理任务的复杂性。它们已经成为许多计算机视觉算法的关键构件，如PASCAL VOC Challenge的高分多类对象分割项[9]、[29]、[11]、深度估计[30]、分割[16]、身体模型估计[22]和对象定位[9]。

There are many approaches to generating superpixels, each with its own advantages and drawbacks that may be better suited to a particular application. For example, if adherence to image boundaries is of paramount importance, the graph-based method of [8] may be an ideal choice. However, if superpixels are to be used to build a graph, a method that produces a more regular lattice, such as [23], is probably a better choice. While it is difficult to define what constitutes an ideal approach for all applications, we believe the following properties are generally desirable:

生成超像素的方法有很多，每种方法都有自己的优点和缺点，可能更适合于特定的应用程序。例如，如果遵守图像边界至关重要，那么基于图的[8]方法可能是一个理想的选择。但是，如果要用超像素来构建图，那么可以使用一种方法来生成更规则的格，比如[23]，这可能是一个更好的选择。虽然很难定义什么是所有应用程序的理想方法，但我们相信以下属性是普遍需要的:

1. Superpixels should adhere well to image boundaries.

2. When used to reduce computational complexity as a pre- processing step, superpixels should be fast to compute,

memory efficient, and simple to use.

3. When used for segmentation purposes, superpixels should both increase the speed and improve the quality of the results.

1. 超像素应该很好地贴合图像边界。

2. 当用于降低计算复杂度作为预处理步骤时，超级像素应该能够快速计算，内存高效，使用简单。

3.当用于分割目的时，超像素既要提高速度，又要提高结果的质量。

We therefore performed an empirical comparison of five state- of-the-art superpixel methods [8], [23], [26], [25], [15], evaluating their speed, ability to adhere to image boundaries, and impact on segmentation performance. We also provide a qualitative review of these, and other, superpixel methods. Our conclusion is that no existing method is satisfactory in all regards.

因此，我们对五种最先进的超像素方法[8]、[23]、[26]、[25]、[15]进行了经验比较，评估了它们的速度、坚持图像边界的能力以及对分割性能的影响。我们还对这些超像素方法以及其他超像素方法进行了定性的综述。我们的结论是，现有的方法在所有方面都不令人满意。

To address this, we propose a new superpixel algorithm: simple linear iterative clustering (SLIC), which adapts k-means clustering to generate superpixels in a manner similar to [30]. While strikingly simple, SLIC is shown to yield state-of-the-art adherence to image boundaries on the Berkeley benchmark [20], and outperforms existing methods when used for segmentation on the PASCAL [7] and MSRC [24] data sets. Furthermore, it is faster and more memory efficient than existing methods. In addition to these quantifiable benefits, SLIC is easy to use, offers flexibility in the compactness and number of the superpixels it generates, is straightforward to extend to higher dimensions, and is freely available.1

为了解决这一问题，我们提出了一种新的超像素算法:简单线性迭代聚类(simple linear iterative clustering, SLIC)，该算法采用k-means聚类生成超像素，其方式类似于[30]。尽管非常简单，但在Berkeley基准[20]上，SLIC显示出了对图像边界的最先进的坚持，并在PASCAL[7]和MSRC[24]数据集上用于分割时优于现有的方法。此外，它比现有的方法更快，内存效率更高。除了这些可量化的好处之外，SLIC易于使用，在其生成的超像素的紧凑性和数量上提供了灵活性，可以直接扩展到更高的维度，并且可以免费使用

2 EXISTING SUPERPIXEL METHODS

Algorithms for generating superpixels can be broadly categorized as either graph-based or gradient ascent methods. Below, we review popular superpixel methods for each of these categories, including some that were not originally designed specifically to generate superpixels. Table 1 provides a qualitative and quanti- tative summary of the reviewed methods, including their relative performance.

2种现有的超像素方法

生成超像素的算法可以大致分为基于图的方法和梯度上升方法。下面，我们将回顾这些类别中流行的超像素方法，包括一些最初不是专门为生成超像素而设计的方法。表1提供了被审查方法的定性和定量总结，包括它们的相对性能。

2.1 Graph-Based Algorithms

Graph-based approaches to superpixel generation treat each pixel as a node in a graph. Edge weights between two nodes are proportional to the similarity between neighboring pixels. Superpixels are created by minimizing a cost function defined over the graph.

2.1基于算法

基于图的超像素生成方法将每个像素视为图中的一个节点。两个节点之间的边缘权值与相邻像素之间的相似度成正比。超像素是通过最小化图上定义的代价函数来创建的。

NC05. The Normalized cuts algorithm [23] recursively parti- tions a graph of all pixels in the image using contour and texture cues, globally minimizing a cost function defined on the edges at the partition boundaries. It produces very regular, visually pleasing superpixels. However, the boundary adherence of NC05 is relatively poor and it is the slowest among the methods (particularly for large images), although attempts to speed up the algorithm exist [5]. NC05 has a complexity of O（N的3/2次方） [15], where N is the number of pixels.

NC05。归一化切割算法[23]使用轮廓和纹理线索递归地对图像中所有像素进行分割，全局最小化在分割边界边缘定义的代价函数。它产生了非常规则的，视觉上令人愉悦的超像素。然而，NC05的边界粘附性相对较差，是所有方法中最慢的(特别是对于大的图像)，尽管有尝试加快算法的[5]。NC05的复杂度为N的3/2次方[15]，其中N为像素数。

GS04. Felzenszwalb and Huttenlocher [8] propose an alter- native graph-based approach that has been applied to generate superpixels. It performs an agglomerative clustering of pixels as nodes on a graph such that each superpixel is the minimum spanning tree of the constituent pixels. GS04 adheres well to image boundaries in practice, but produces superpixels with very irregular sizes and shapes. It is O（N log N） complex and fast in practice. However, it does not offer an explicit control over the amount of superpixels or their compactness.

GS04。Felzenszwalb和Huttenlocher[8]提出了另一种基于图的方法，该方法已被应用于生成超像素。它将像素作为图上的节点进行聚合聚类，使每个超像素是组成像素的最小生成树。GS04在实践中很好地坚持了图像的边界，但产生的超像素的大小和形状非常不规则。它是O（N log N）复杂和快速的实践。然而，它并没有对超像素的数量或其紧凑性提供明确的控制。

SL08. Moore et al. propose a method to generate superpixels that conform to a grid by finding optimal paths, or seams, that split the image into smaller vertical or horizontal regions [21]. Optimal paths are found using a graph cuts method similar to Seam Carving [1]. While the complexity of SL08 is o（N 的3/2次方\* log N ） according to the authors, this does not account for the precomputed boundary maps, which strongly influence the quality and speed of the output.

SL08。Moore等人提出了一种方法，通过寻找最优路径或接缝，将图像分割成更小的垂直或水平区域[21]，从而生成符合网格的超像素。使用类似于Seam Carving[1]的图形切割方法可以找到最优路径。虽然SL08的复杂度是o（N 的3/2次方\* log N ），但这并没有考虑到预先计算的边界图，它强烈地影响了输出的质量和速度。

GCa10 and GCb10. In [26], Veksler et al. use a global optimization approach similar to the texture synthesis work of [14]. Superpixels are obtained by stitching together overlapping image patches such that each pixel belongs to only one of the overlapping regions. They suggest two variants of their method, one for generating compact superpixels (GCa10) and one for constant- intensity superpixels (GCb10).

GCa10 GCb10。在[26]中，Veksler等人使用了一种类似于[14]纹理合成工作的全局优化方法。超像素是通过将重叠的图像块拼接在一起，使每个像素只属于一个重叠区域而得到的。他们提出了两种方法的变体，一种用于生成紧凑的超像素(GCa10)，另一种用于生成恒定强度的超像素(GCb10)。



Images segmented using SLIC into superpixels of size 64, 256, and 1,024 pixels (approximately).

使用SLIC将图像分割成大小为64、256和1024像素(大约)的超像素。

2.2 Gradient-Ascent-Based Algorithms

Starting from a rough initial clustering of pixels, gradient ascent methods iteratively refine the clusters until some convergence criterion is met to form superpixels.

2.2 Gradient-Ascent-Based算法

梯度上升方法从初始像素的粗略聚类开始，迭代细化聚类，直到满足一定的收敛准则，形成超像素。

MS02. In [4], mean shift, an iterative mode-seeking procedure for locating local maxima of a density function, is applied to find modes in the color or intensity feature space of an image. Pixels that converge to the same mode define the superpixels. MS02 is an older approach, producing irregularly shaped superpixels of nonuniform size. It is O（N的2次方） complex, making it relatively slow, and does not offer direct control over the amount, size, or compactness of superpixels.

MS02。在[4]中，均值漂移是一种定位密度函数局部极大值的迭代模态寻找过程，用于在图像的颜色或强度特征空间中寻找模态。汇聚到相同模式的像素定义了超像素。MS02是一种较老的方法，它会产生形状不规则、大小不一致的超像素。它是O（N的2次方）复杂的，这使得它相对缓慢，并且不能提供对超像素的数量、大小或紧凑性的直接控制。

QS08. Quick shift [25] also uses a mode-seeking segmentation scheme. It initializes the segmentation using a medoid shift procedure. It then moves each point in the feature space to the nearest neighbor that increases the Parzen density estimate. While it has relatively good boundary adherence, QS08 is quite slow, with an （dN的2次方）complexity (d is a small constant [25]). QS08 does not allow for explicit control over the size or number of superpixels. Previous works have used QS08 for object localization [9] and motion segmentation [2].

QS08。快移[25]也使用了模式寻找分割方案。它使用中间移位程序初始化分割。然后，它将特征空间中的每个点移动到最近的邻居，从而增加Parzen密度估计。虽然它具有相对良好的边界遵循，但QS08相当慢，具有O（dN的2次方）复杂度(d是一个小常数[25])。QS08不允许明确控制超像素的大小或数量。之前的工作使用QS08进行目标定位[9]和运动分割[2]。

WS91. The watershed approach [28] performs a gradient ascent starting from local minima to produce watersheds, lines that separate catchment basins. The resulting superpixels are often highly irregular in size and shape, and do not exhibit good boundary adherence. The approach of [28] is relatively fast (O（NlogN） complexity), but does not offer control over the amount of superpixels or their compactness.

WS91。分水岭方法[28]从局部最小值开始进行梯度上升，以产生分水岭，即分隔集水区流域的线。得到的超像素通常在大小和形状上高度不规则，并且不表现出良好的边界粘附。[28]的方法相对较快(O（NlogN）复杂性)，但不能控制超像素的数量或它们的紧凑性。

TP09. The Turbopixel method progressively dilates a set of seed locations using level-set-based geometric flow [15]. The geometric flow relies on local image gradients, aiming to regularly distribute superpixels on the image plane. Unlike WS91, TP09 superpixels are constrained to have uniform size, compactness, and boundary adherence. TP09 relies on algorithms of varying complexity, but in practice, as the authors claim, has approximately OðNÞ behavior [15]. However, it is among the slowest algorithms examined and exhibits relatively poor boundary adherence.

TP09。Turbopixel方法使用基于水平集的几何流[15]逐步扩展一组种子位置。几何流依赖于局部图像梯度，目标是在图像平面上有规律地分布超像素。与WS91不同，TP09超像素被限制为具有均匀的大小、紧凑性和边界粘附性。TP09依赖于不同复杂度的算法，但在实践中，正如作者所声称的，它大约有O（N）行为[15]。然而，它是最慢的算法检查和显示相对较差的边界粘附。

3 SLIC SUPERPIXELS

We propose a new method for generating superpixels which is faster than existing methods, more memory efficient, exhibits state-of-the-art boundary adherence, and improves the perfor- mance of segmentation algorithms. Simple linear iterative clustering is an adaptation of k-means for superpixel generation, with two important distinctions:

3 SLIC SUPERPIXELS

我们提出了一种新的方法来生成超像素比现有的方法更快，更有效的内存，展示了最先进的边界粘附，并提高了分割算法的性能。简单线性迭代聚类是k-means对超像素生成的一种适应，有两个重要的区别:

1. The number of distance calculations in the optimization is dramatically reduced by limiting the search space to a region proportional to the superpixel size. This reduces the complexity to be linear in the number of pixels N—and independent of the number of superpixels k.

2. A weighted distance measure combines color and spatial proximity while simultaneously providing control over the size and compactness of the superpixels.

1. 通过将搜索空间限制在与超像素大小成比例的区域，优化中的距离计算数量大大减少。这降低了复杂度，使其与像素n成线性关系，而与超像素k无关。

2. 加权距离度量结合了颜色和空间邻近性，同时提供了对超像素大小和紧凑性的控制。

SLIC is similar to the approach used as a preprocessing step for depth estimation described in [30], which was not fully explored in the context of superpixel generation.

SLIC类似于[30]中描述的用于深度估计的预处理步骤的方法，这在超像素生成的背景下没有得到充分的探讨。

3.1 Algorithm

SLIC is simple to use and understand. By default, the only parameter of the algorithm is k, the desired number of approximately equally sized superpixels.2 For color images in the CIELAB color space, the clustering procedure begins with an initialization step where k initial cluster centers Ci 1⁄4 1⁄2li ai bi xi yi􏰀T are sampled on a regular grid spaced S pixels apart. To produce roughly equally sized superpixels, the grid interval is S 1⁄4 N=k. The centers are moved to seed locations corresponding to the lowest gradient position in a 3 􏰁 3 neighborhood. This is done to avoid centering a superpixel on an edge and to reduce the chance of seeding a superpixel with a noisy pixel.

Next, in the assignment step, each pixel i is associated with the nearest cluster center whose search region overlaps its location, as depicted in Fig. 2. This is the key to speeding up our algorithm because limiting the size of the search region significantly reduces the number of distance calculations, and results in a significant speed advantage over conventional k-means clustering where each pixel must be compared with all cluster centers. This is only possible through the introduction of a distance measure D, which determines the nearest cluster center for each pixel, as discussed in Section 3.2. Since the expected spatial extent of a superpixel is a region of approximate size S \* S, the search for similar pixels is done in a region 2S \* 2S around the superpixel center.

3.1算法

SLIC易于使用和理解。默认情况下，算法的唯一参数是k，即期望的大约相等大小的超像素的数量对于CIELAB颜色空间中的彩色图像，聚类过程从初始化步骤开始，在间隔为S像素的规则网格上采样k个初始聚类中心Ci 1 / 4 1 / 2li ai xi yi􏰀T。为了产生大致相等大小的超像素，网格间隔为S =N/k。这些中心被移动到对应于3\*3网格中最低梯度位置的种子位置。这样做是为了避免超像素在边缘上居中，并减少用噪声像素播种超像素的机会。

接下来，在分配步骤中，将每个像素i与最近的搜索区域与其位置重叠的聚类中心关联，如图2所示。这是加快算法速度的关键，因为限制搜索区域的大小大大减少了距离计算的数量，并导致与传统的k-means聚类(每个像素必须与所有聚类中心进行比较)相比，具有显著的速度优势。这只能通过引入距离度量D来实现，它确定每个像素最近的聚类中心，如第3.2节所述。由于超像素的期望空间范围是一个近似大小为S \* S的区域，所以在超像素中心周围的2S \* 2S区域内进行相似像素的搜索。

Once each pixel has been associated to the nearest cluster center, an update step adjusts the cluster centers to be the mean 1⁄2l a b x y􏰀T vector of all the pixels belonging to the cluster. The L2 norm is used to compute a residual error E between the new cluster center locations and previous cluster center locations. The assignment and update steps can be repeated iteratively until the error converges, but we have found that 10 iterations suffices for most images, and report all results in this paper using this criteria. Finally, a postprocessing step enforces connectivity by reassigning disjoint pixels to nearby superpixels. The entire algorithm is summarized in Algorithm 1.

一旦将每个像素关联到最近的集群中心，更新步骤将集群中心调整为属于该集群的所有像素的平均1 / 2l a b x y􏰀T向量。L2范数用于计算新的集群中心位置和以前的集群中心位置之间的残差E。赋值和更新步骤可以迭代地重复，直到误差收敛，但我们发现，对于大多数图像，10次迭代就足够了，本文使用这个准则报告所有结果。最后，后处理步骤通过将不相交的像素重新分配给邻近的超像素来增强连接性。整个算法在算法1中进行了总结。

3.2 Distance Measure

SLIC superpixels correspond to clusters in the labxy color-image plane space. This presents a problem in defining the distance measure D, which may not be immediately obvious. D computes the distance between a pixel i and cluster center Ck in Algorithm 1. A pixel’s color is represented in the CIELAB color space 1⁄2l a b􏰀T, whose range of possible values is known. The pixel’s position position 1⁄2x y􏰀T, on the other hand, may take a range of values that varies according to the size of the image.

Simply defining D to be the 5D euclidean distance in labxy space will cause inconsistencies in clustering behavior for different superpixel sizes. For large superpixels, spatial distances outweigh color proximity, giving more relative importance to spatial proximity than color. This produces compact superpixels that do not adhere well to image boundaries. For smaller superpixels, the converse is true.

To combine the two distances into a single measure, it is necessary to normalize color proximity and spatial proximity by their respective maximum distances within a cluster, Ns and Nc. Doing so, D0 is written

3.2距离测量

SLIC超像素对应labxy彩色图像平面空间中的聚类。这在定义距离度量D时提出了一个问题，这个问题可能不是很明显。D计算算法1中像素i到聚类中心Ck的距离。像素的颜色在CIELAB颜色空间1 / 2l A b􏰀T中表示，其可能值的范围是已知的。另一方面，像素的位置1⁄2x y􏰀T可能会根据图像的大小而变化。

简单地将D定义为labxy空间中的5D欧氏距离会导致不同超像素大小的聚类行为不一致。对于大的超像素，空间距离比颜色接近更重要，空间接近比颜色更重要。这将产生紧凑的超像素，不能很好地粘附在图像边界上。对于较小的超像素，情况正好相反。

为了将两个距离合并为一个度量，需要将颜色接近度和空间接近度归一化，即它们在一个簇内的最大距离n和Nc。这样，D0就被写入

The maximum spatial distance expected within a given cluster should correspond to the sampling interval, NS 1⁄4 S 1⁄4 ðN=KÞ. Determining the maximum color distance Nc is not so straightfor- ward, as color distances can vary significantly from cluster to cluster and image to image. This problem can be avoided by fixing Nc to a constant m so that (1) becomes

给定集群内期望的最大空间距离应对应于采样间隔，NS 1 / 4 S 1 / 4 ðN=KÞ。确定最大颜色距离Nc不是那么简单，因为颜色距离在不同的簇和不同的图像之间会有很大的差异。这个问题可以通过将Nc固定到一个常数m来避免，使(1)变成

这就简化了我们在实践中使用的距离度量:

通过这样定义D, m也允许我们权衡颜色相似度和空间接近度之间的相对重要性。当m较大时，空间邻近性更重要，得到的超像素更紧凑(即，它们的面积与周长比更低)。当m较小时，生成的超像素更紧密地贴合图像边界，但其大小和形状不那么规则。当使用CIELAB颜色空间时，m可以在范围【1，40】。通过设置，可以使式(3)适用于灰度图像

3.3后处理

也可以扩展到处理3D超体素，如图3所示，在(3)的空间接近项中加入深度维度:

与其他一些超像素算法[8]一样，SLIC并不显式强制连接性。在聚类过程的最后，一些“孤立的”像素可能仍然存在，它们不属于与它们的聚类中心相同的连接组件。为了纠正这个问题，使用连接组件算法为这些像素分配最近的聚类中心的标签。

当采用k-means加速方法时，SLIC的复杂度是线性的

3.4复杂性

通过在聚类过程中定位搜索，SLIC避免了执行数千个冗余的距离计算。实际上，一个像素位于少于8个集群中心的附近，这意味着SLIC是OðNÞ复合体。而经典k-means算法的平凡上界为OðkNÞ[17]，实际时间复杂度为OðNkIÞ[6]，其中I为收敛所需的迭代次数。虽然已经提出了一些降低k-means复杂度的方案，如素数长度抽样[27]、随机抽样[13]、局部簇交换[12]和设置上界和下界[6]，但这些方法在本质上是非常通用的。SLIC是专门针对超像素聚类问题的。最后，不像前面提到的大多数超像素方法