**Superpixels**

Rewritten by Han

SLIC:

SLIC was first proposed by Achanta et. al. [10], and their motivation was to find an algorithm to compute nearly equal-sized superpixels efficiently for an image. Intuitively, for all pixels, the algorithm performs local clustering of a 5-D space data defined by *L, a, b*, values of the CIEAB color standard and the *x, y* coordinates. Suppose an image has *N* total pixels, and a user sets *K* as the number of total superpixels to have in the image. Then, there will be around pixels for each superpixel. If our goal is to make each superpixel’s shape similar to a perfect square, then the length of its side will be . Thus, this value will be a good estimate for the interval distance between each pair of initial superpixel centers on the image. After that, we can use to denote each superpixel center, with *k* as the index from 1 to *K*. Here, Achanta et. al. [10] made an assumption that the search area for the pixels associated with a superpixel center will be within a area around the center, since the area of each superpixel is about .

Then according to [10], the *L, a, b* values are used to calculate the color distances from neighboring temporal superpixel centers, while the *x, y* coordinates are used to estimate the Euclidean distances from the center. The following formula from [10] was intended to combine both color and geometric distances for each pixel from a nearby superpixel center:

Here, and are the Euclidean distances for *L, a, b* values and *x, y* coordinates, respectively. is a linear combination of and . Specifically, , and the *m* value is chosen by the user. Since will be fixed for fixed *N* and *K* values, the *m* value will determine how much weight we put on the distance for *x, y* coordinates.

Then, a minor adjustment will be done for the locations of initial superpixel centers to prevent them from sitting on any edges [10]. We ignored this minor detail here, but if you are interested in this, please do not hesitate to check the original paper for the detail. Finally, the formal algorithm proposed by [10] will be to first initialize the superpixel centers on the grid of an image by the interval distance of . Then, each pixel in the image will be assigned to its nearest superpixel center. After this step, a new superpixel center will be calculated by averaging all the of each pixel belonging to that center. Then, it will repeat the above two steps, until the results converge.

The advantage of using SLIC to compute superpixels is that its efficiency is outstanding. According to [10], its efficiency is , where *N* is the total number of pixels in an image. However, since the SLIC algorithm is a special case of the K-means algorithm in the unsupervised learning field, it may inherit some disadvantages from the K-means algorithm family.

NCuts:

Turbopixels: