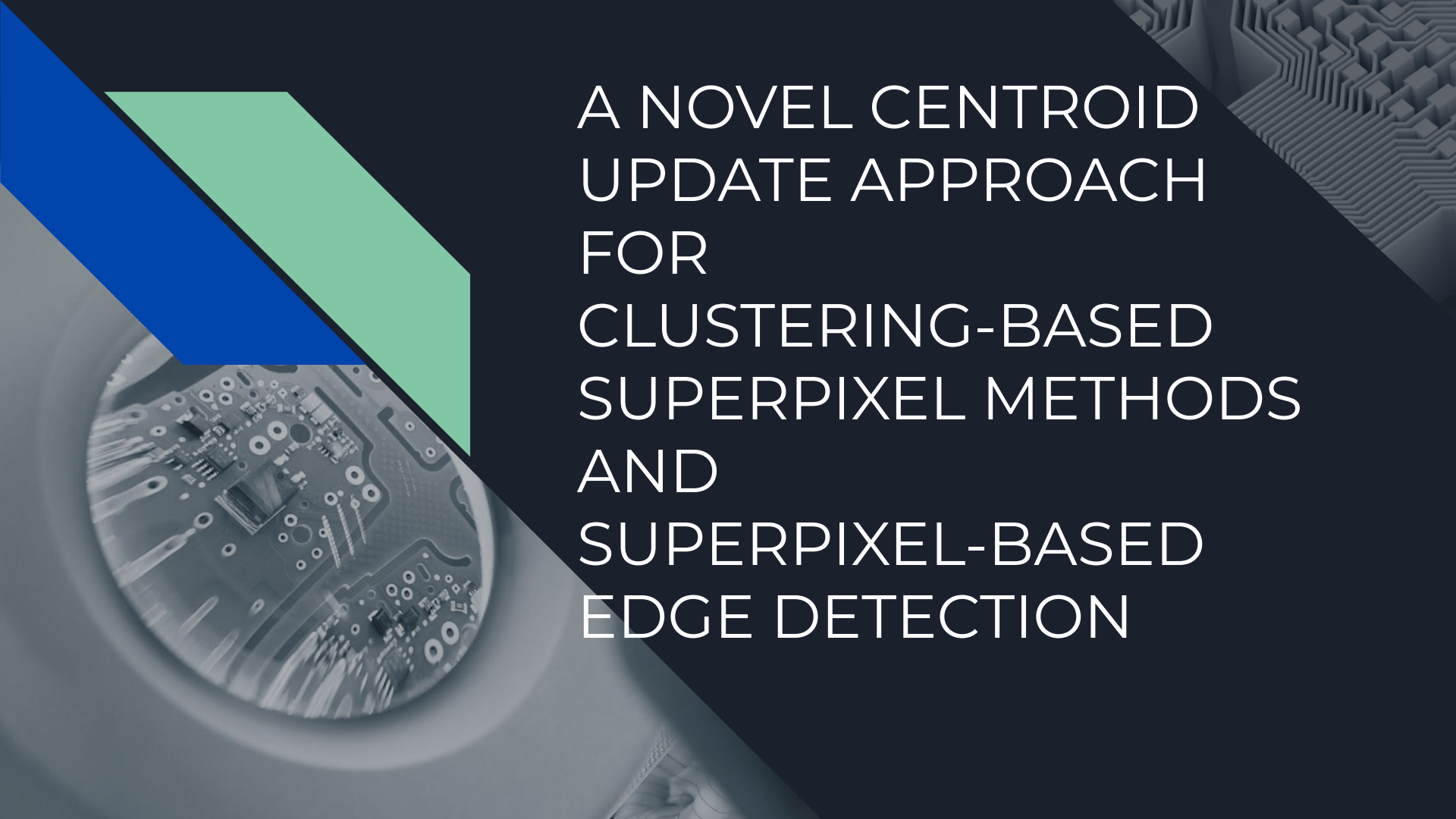


A NOVEL CENTROID UPDATE APPROACH FOR CLUSTERING-BASED SUPERPIXEL METHODS AND SUPERPIXEL-BASED EDGE DETECTION

TOC

Problem Statement

Superpixel is widely used in image processing. There are various methods for superpixel generation. The most used methods are cluster-based.

Even though this method is very fast, it is very susceptible to noise. In this project, we will analyze different noises and suggest a new centroid update approach for clustering-based superpixel methods and superpixel-based edge detection.

Goals and approach

- Analyze why clustering based superpixel algorithms won’t work in noisy environments
    - According to the feature of the noise, we propose a novel centroid based approach for superpixel generation
        - Based on superpixels we propose a new edge detection method

Effects of noise

- When noise exists it tends to be singular, but when we gradually form a superpixel by taking average the noise accumulate
- If we use this noisy superpixel segmentation for edge detection it will cause false borders
- We are assuming the noise to be gaussian

$$Noise_{(z)} = \frac{1}{\sqrt{2\pi}\delta} exp[\frac{-(z-u)^2}{2\delta^2}]$$

z is the value of the pixel and u is the mean value of z

Simple Linear Iterative Clustering

SLIC performs a local clustering of pixels in 5-D space defined by the L, a, b values of the CIELAB colorspace and x, y coordinates of the pixels.

Few points before the algorithm:

- Each Super Pixel has five dimensions [l, a, b, x, y]
- **Initially Each super pixel can be imagined as square segment of side S**
- **Distance metrics for SLIC are given as**
  - $d_{lab} = \sqrt{((l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2)}$
  - $d_{xy} = \sqrt{((x_k - x_i)^2 + (y_k - y_i)^2)}$
  - $D_s = d_{lab} + (m / S)* d_{xy}$
- Where m = distance emphasis desired by user and S is the side of each super pixels

Source: https://www.iro.umontreal.ca/~mignotte/IFT6150/Articles/SLIC_Superpixels.pdf

The SLIC algorithm

1. Take 3 inputs from the user
    a. Input Image
    b. K = desired number of super pixels
    c. M = Distance Emphasis factor
2. Approximate number of pixels in a super pixel = A = N/K pixels, N= total number of pixels
3. S = √(N/K) is the distance at which each super pixel center is present
4. Gradient is applied across the super pixel to prevent the ambiguity of a pixel being on a edge
5. Now for every pixel search a window of 2S and assign the pixel to a super pixel which has least distance Ds
6. Now calculate the mean L, A, B, X, Y values of the super pixel
7. Now the steps 5 and 6 are iterated for multiple times to get a better accuracy

Source: https://www.iro.umontreal.ca/~mignotte/IFT6150/Articles/SLIC_Superpixels.pdf

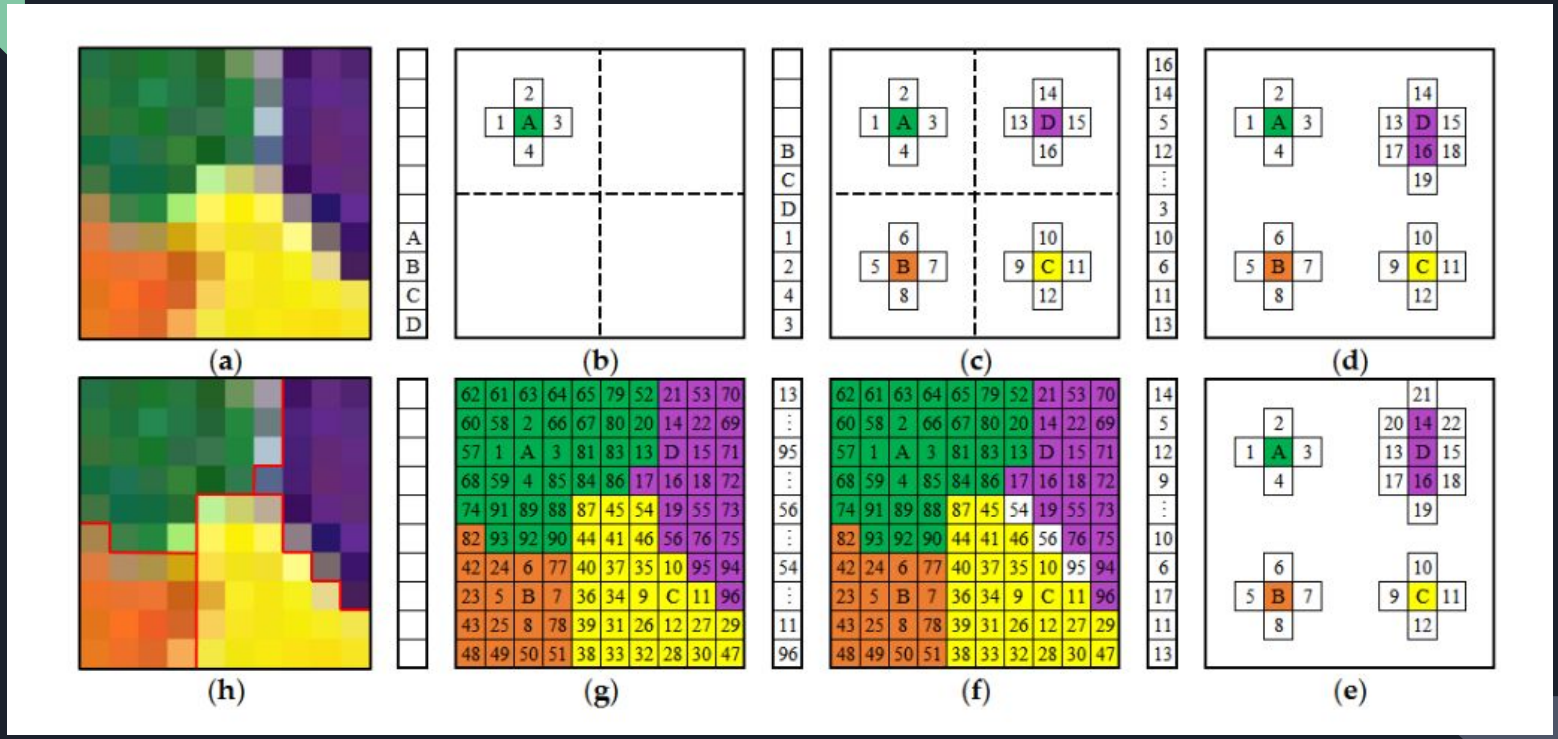
SNIC (Simple Non-Iterative Clustering)

- First the image is segmented into x_seg x y_seg pieces and a seed in defined inside each of those segments
- We'll keep a list of centroids corresponding to each superpixel. Each centroid will have 5 parameters : [x, y, l, a, b]
- We define a priority queue that sorts based on the D (The distance between two pixels)

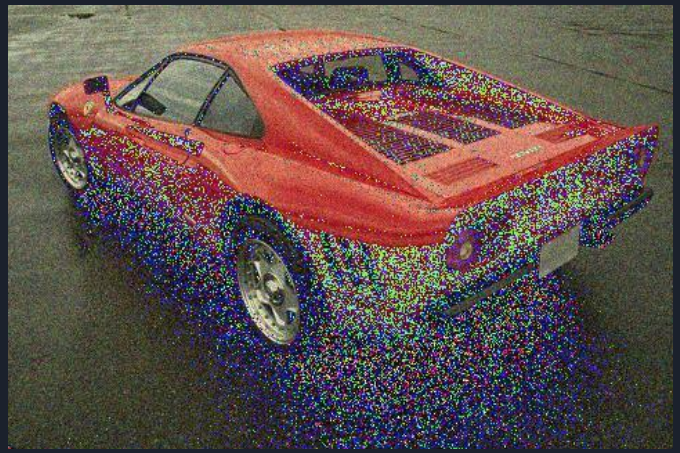
$$D(I_i, S_k) = \sqrt{\|C(I_i) - C(S_k)\|_2^2 + \lambda^2 \|P(I_i) - P(S_k)\|_2^2},$$

- All the seeds are pushed into the queue (seeds will have their D = 0)
- Then we run a loop until the queue is empty. In each iteration we pop the first pixel in the queue and allot it to corresponding superpixel (and update the centroid)
- Now this pixel is taken as the new frontier and all its neighbors are pushed into the queue after calculating D with the centroid of corresponding superpixel

SNIC (Simple Non-Iterative Clustering)



The Problem



Input

SLIC Output

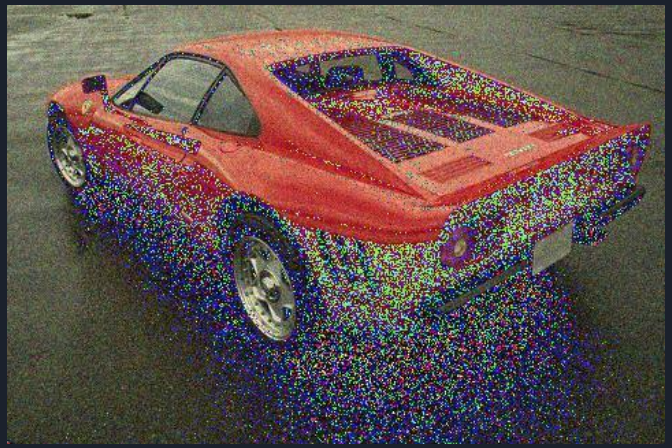
New approach

- The new approach can be used along with any currently available superpixeling algorithms
- But the way we calculate the centroid is different
- For each superpixel the coordinate centroid is the mean of the coordinates of all the pixels and the color centroid is found by taking the mean of all the colors present is a square of adaptive size

$$\sqrt{\frac{M \times N}{k \times 2}}$$

- Here M and N are rows and columns in the image and k is the number of super pixels

Super Pixel Segmentation with Noise



Input

SLIC with Centroid X

Superpixel Based Edge Detection

- Once we generate the superpixels, we can make a k x k matrix (where k is the number of superpixels) which stores the distance between two superpixels defined as :

$$D_{i,j} = \left|\bar{L}_i - \bar{L}_j\right| + \left|\bar{A}_i - \bar{A}_j\right| + \left|\bar{B}_i - \bar{B}_j\right|,$$
$$\bar{L}_i = \sum_{c \in S_i} \frac{l_c}{|S_i|}, \bar{A}_i = \sum_{c \in S_i} \frac{a_c}{|S_i|}, \bar{B}_i = \sum_{c \in S_i} \frac{b_c}{|S_i|},$$

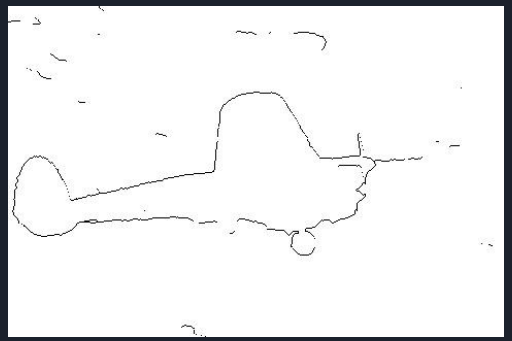
- This will be an upper triangular matrix
- Now we find a threshold distance by taking the average of all non zero elements in this matrix
- We will only mark borders for those superpixels with distance more than the threshold

Input

SNIC output

Centroid SNIC output

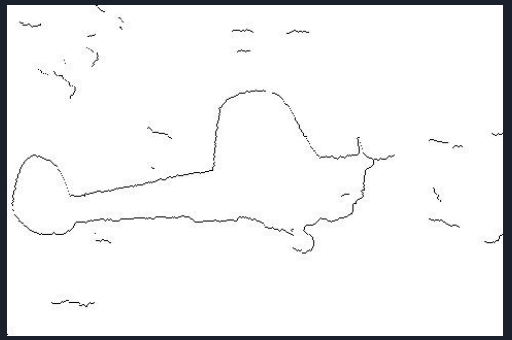
SNIC edge detection



Input(with gaussian noise)

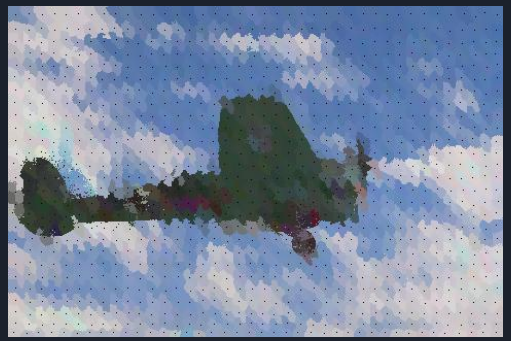
SNIC output

SNIC edge detection with noise



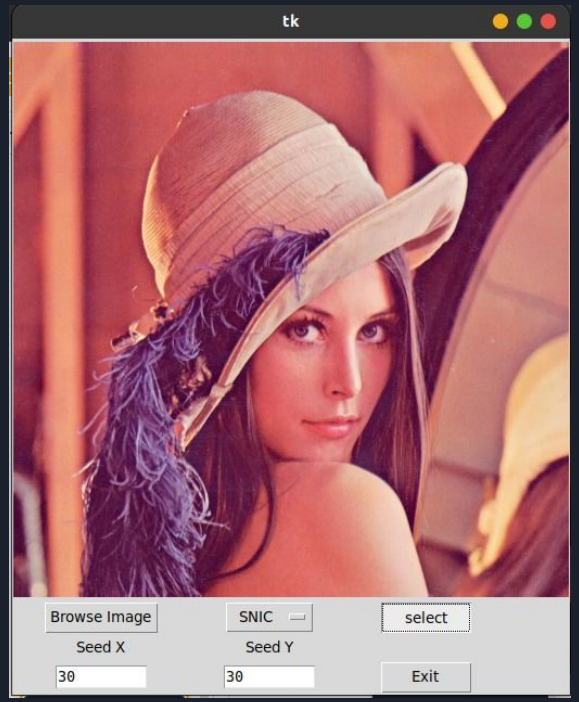
SLIC with noise

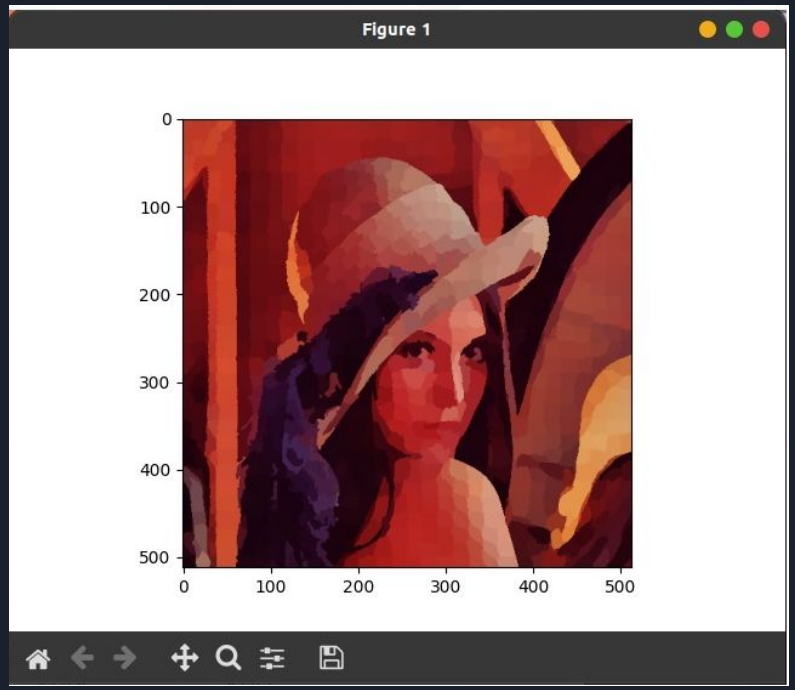
SLIC centroid with noise



GUI

Used tkinter and pillow libraries





References

[1] A NOVEL CENTROID UPDATE APPROACH FOR CLUSTERING-BASED SUPERPIXEL METHODS AND SUPERPIXEL-BASED EDGE DETECTION

[2] Radhakrishna Achanta and Sabine Susstrunk, “Super-pixels and polygons using simple non-iterative cluster-ing,” in Proceedings of the IEEE Conference on Com-puter Vision and Pattern Recognition, 2017, pp. 4651–4660.

[3] NICE: Superpixel Segmentation Using Non-Iterative Clustering with Efficiency Cheng Li 1 , Baolong Guo 1, *, Geng Wang 1 , Yan Zheng 1 , Yang Liu 2 and Wangpeng He


thank
you