FLORIDA STATE
UNIVERSITY

Efficiently Automating Self-Assembled Quantum-Dot Nanofilm Image Segmentation

Wendy Slattery - Computer Science
Florida State University
April 14, 2021

The Challenge

Researcher's Dilemma

- Chemist images a series of reactions on Quantum Dots
- He manually pixelates each image to separate the nanofilms from background
- His question is can we automate this process?
- If so, can we also apply it to a videos which have captured the entire reaction?

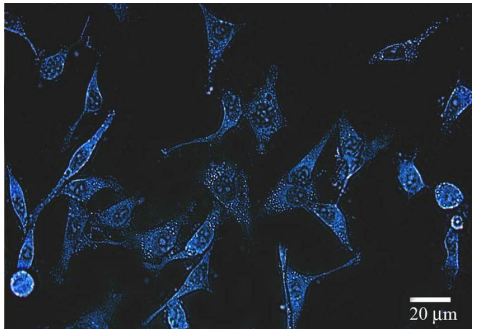
Approach & Questions

- The Chemistry: Understand researcher's challenges and goals
- Explore segmentation strategies / feature extraction
- Could Artificial Intelligence algorithms enhance outcomes?
- What mechanisms are best suited to handle feature extractions in video?

Quantum Dot Nanotechnology

https://www.graphene-info.com/graphene-quantum-dots

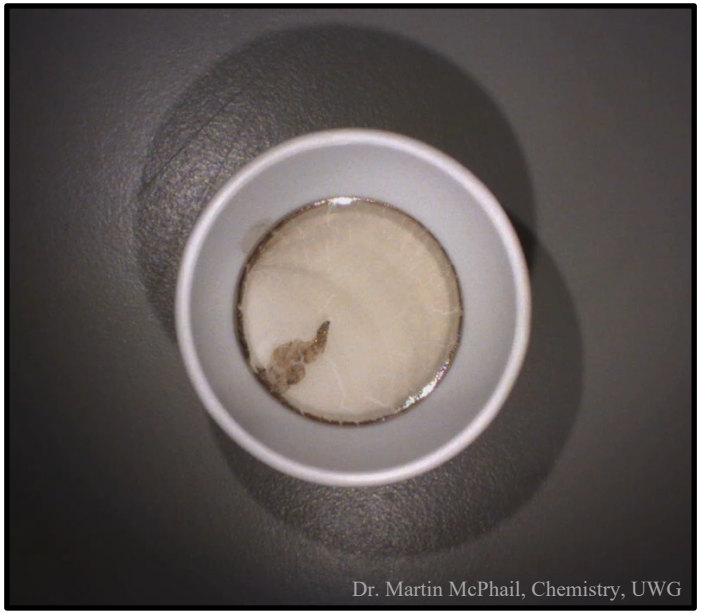
- Man-made nanoscale crystal semiconductors
- Extremely small (takes 10,000 QDs to span width of human hair)
- Convert between optical, electrical, and chemical energy
- Used in solar cells, LCD/LED displays, photocatalysts, and photovoltaics, and medical imaging and therapies



Dr. Richard Tilley: https://www.sciencelearn.org.

https://www.chemistryviews.org/

Single Quantum Dot (QD) Monolayers Deposited Over Large Areas



Self-Assembly is the Reorganization of QDs at the Molecular Level

Methods

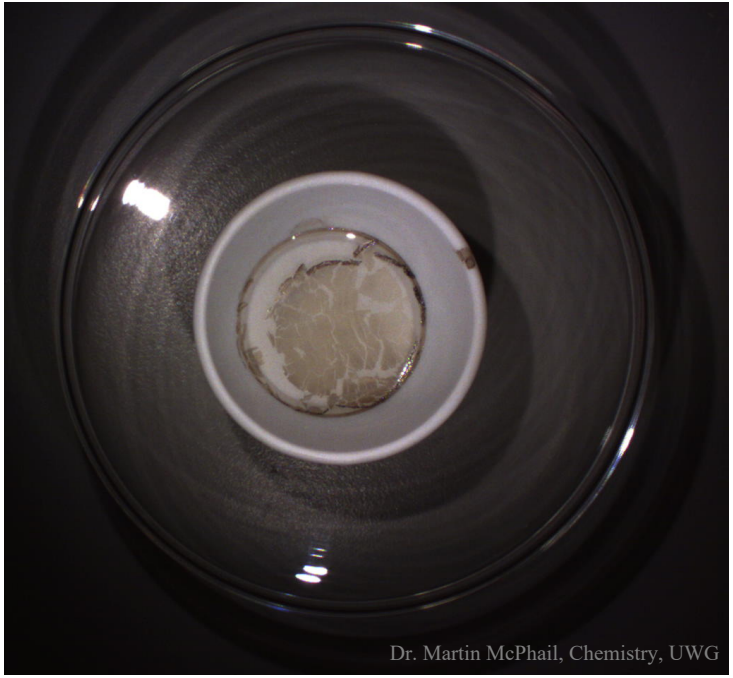
Participants

This work was a collaboration and mentorship

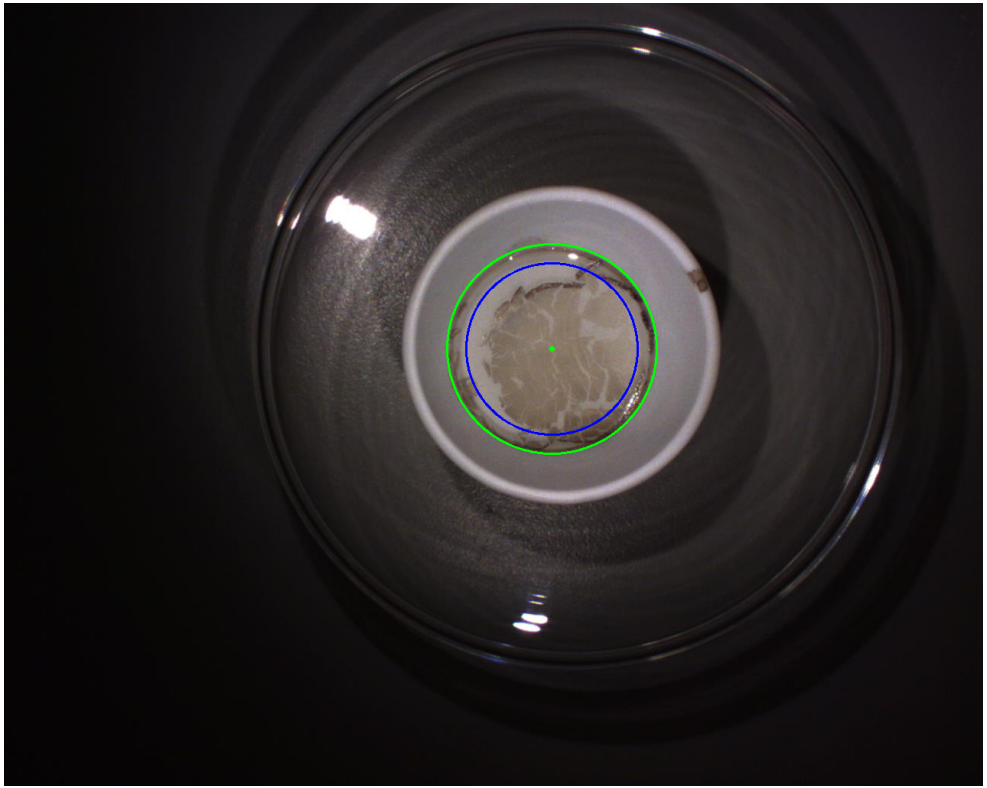
- Dr. Karen Works, FSU, Computer Science
- Dr. Martin McPhail, UWG, Chemistry

Procedure

- ❖ Isolate region of interest (ROI) & crop image
- ❖ Compare Segmentation Techniques
- ❖ Explore AI algorithms which identify and remove unnecessary pixels in images and frames thereby enhancing efficiency
- ❖ Incorporate background subtraction within videos



Hough Circle Transform



Region of Interest
(ROI)

Meniscus Ring
(Ring)

Area of Contraction
(AOC)

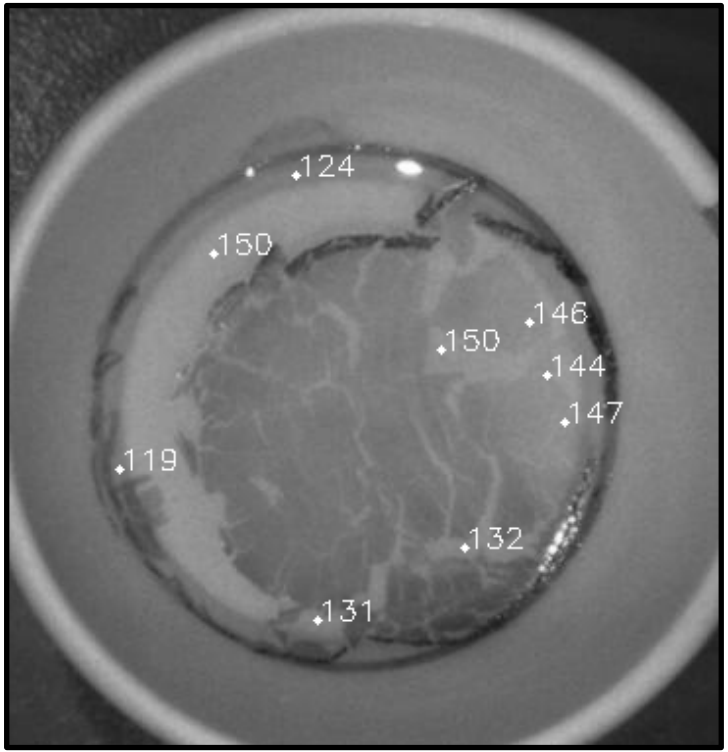
Traditional Segmentation Methods Fail

Attempted Methods

- Binary Thresholding
- Edge Detection Filters
- Dilation / Erosion
- Watershed (2D topological)
  - > poor segmentation results

Issues

- Translucency
- Low Contrast
- Discontinuous Edges
- Regions of Noise



Pixel Values
Grayscale

255 (white)

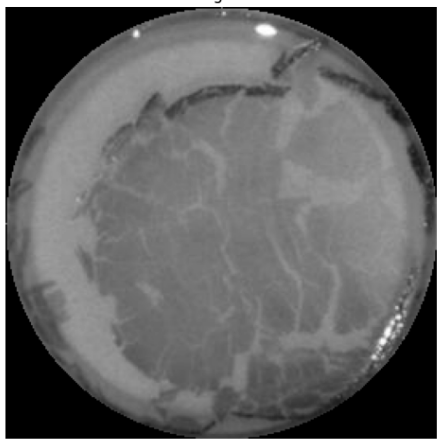
lighter

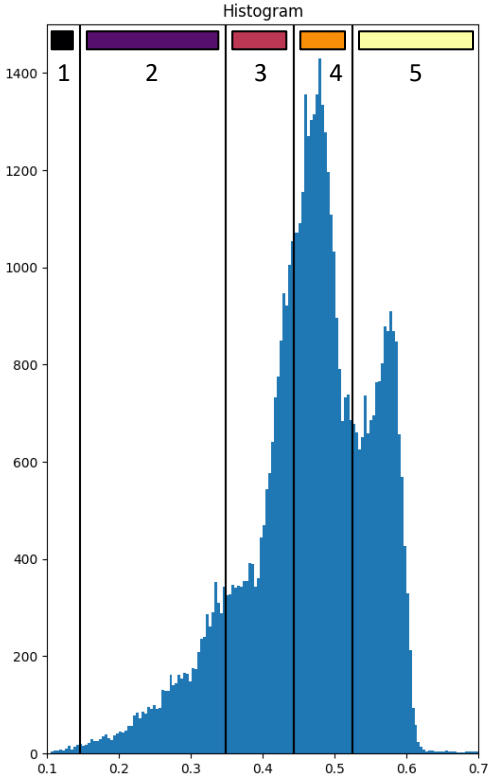
darker

0 (black)

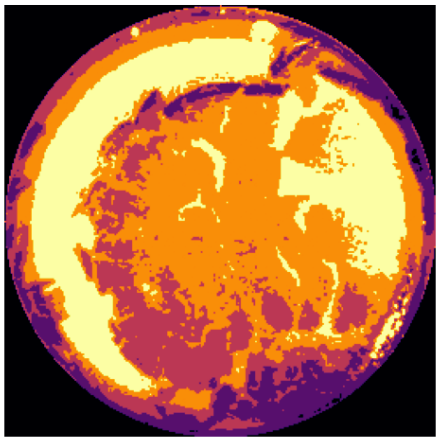
Multi-Otsu Thresholding with 5 Regions

Original



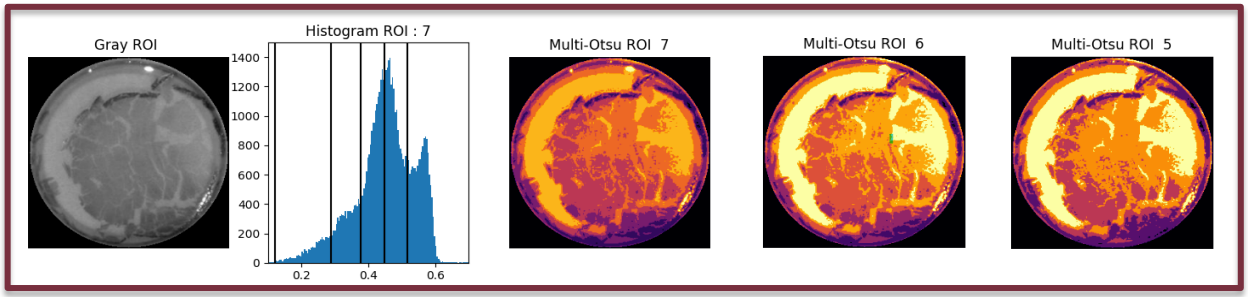


Multi-Otsu result



Identify layers of film density to discriminate between background and foreground

Multi-Otsu Thresholding Comparisons



Partitioning Adds Accuracy

~ But ~

Multi-Otsu Thresholding is Computationally Costly

Processing Speeds

Classes :	Time
4	: 5 sec.
5	: 10 - 15 sec.
6	: 10 - 15 min.
7	: 2.5 - 3 hours

Initial Comparison

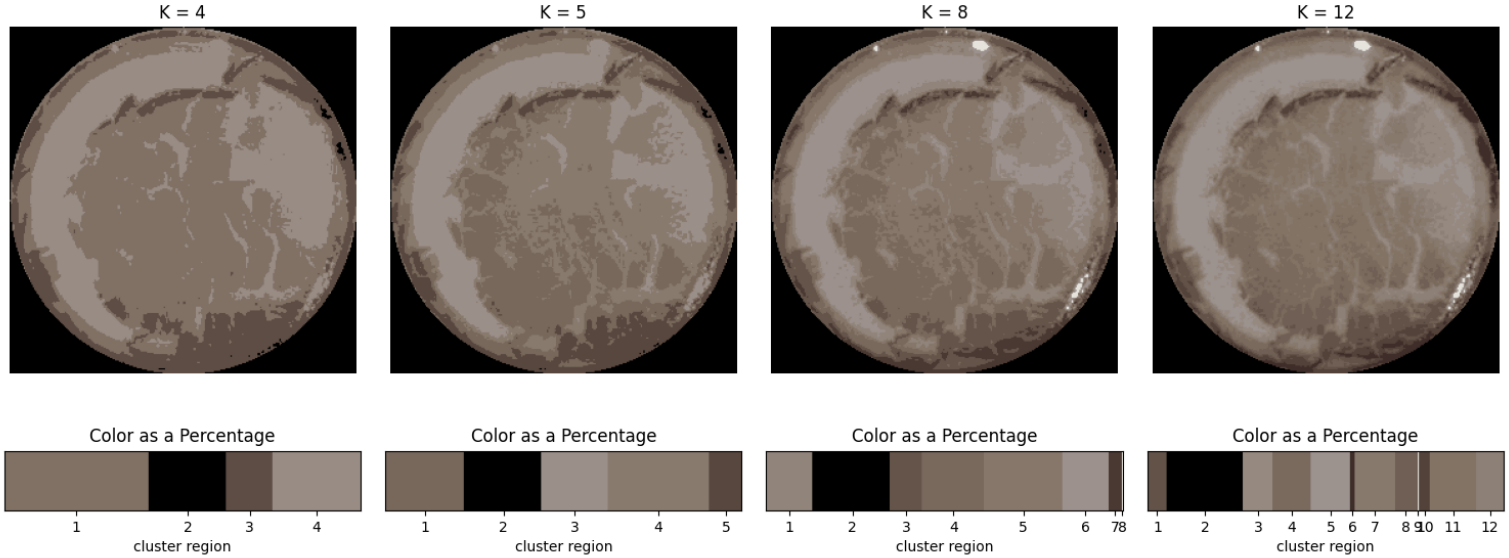
Manual Pixelation
33,919 px area
Process Time: ~ 2 hours

✓Automation
✓Accuracy

Multi-thresholding / Partitioning
35,681 px area (+5%)
Process Time: ~ 2.75 hours (if 7 classes)

K-Means Color Quantization

Unsupervised AI Learning Algorithm



Processing Time: 2 seconds!!

Background Subtraction with K-Nearest Neighbors

Frame by Frame Detection of Moving Pixels in Real-Time

Conclusions

- Nanofilm Imaging is a notoriously challenging process
- This project offers reliable and efficient solutions for these barriers
- This work is not complete and is continuing
  - Rebecca Carroll is joining the project to develop an intuitive web-based UI
  - Explore Unsupervised and Semi-Supervised learning environments to tune segmentation
  - Better define how time-series methods interact with user
  - Test-drive program in the lab
- Dr. McPhail introduced this project in his recently invited talk at the Spring National American Chemical Society Meeting for its benefits to his work and perhaps other chemists

Thank You!

A special thank you to Drs. Works and McPhail for their mentorship and for offering such a valuable insight producing gains we did not initially foresee

Dr. Karen Works, PhD.
Computer Science
Florida State University
Project Advisor

Dr. Martin McPhail, PhD.
Chemistry
West Georgia University
Quantum Dot Research

References

Ding, Yu & Bukkapatnam, Satish. (2015). Challenges and needs for automating nano image processing for material characterization. 95560Z. 10.1117/12.2186251. Hughes A, Liu Z, Raftari M,

Reeves ME. (2014). A workflow for characterizing nanoparticle monolayers for biosensors: Machine learning on real and artificial SEM images. *PeerJ PrePrints* 2:e671v2 https://doi.org/10.7287/peerj.preprints.671v2

Huang, D.Y., Lin, T.W. and Hu, W.C. (2011) Automatic Multilevel Thresholding Based on Two-Stage Otsu's Method with Cluster Determination by Valley Estimation. ICIC International Journal of Innovative Computing, Information and Control, 7, 5631-5644.

Liao, Ping-Sung & Chen, Tse-Sheng & Chung, Pau-Choo. (2001). A Fast Algorithm for Multilevel Thresholding.. J. Inf. Sci. Eng.. 17. 713-727.

FLORIDA STATE
UNIVERSITY

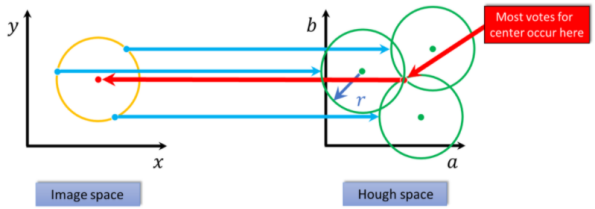
And You Thought it was the End…

Extra slides for questions of details under the hood

Feature Extraction with Hough Circle Transform and Hough Gradient

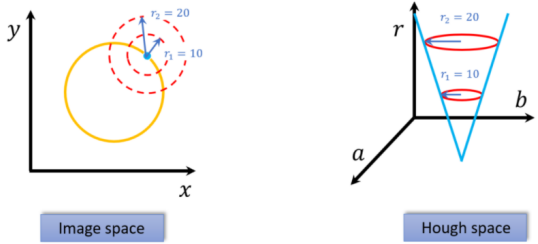
Equation for Circle: $(x_i - a)^2 + (y_i - b)^2 = r^2$

Step 1: Compute edges in image
Step 2: Map points on these edge from image space to the Hough parametric space



Mapping the Image space into a 2D Hough space

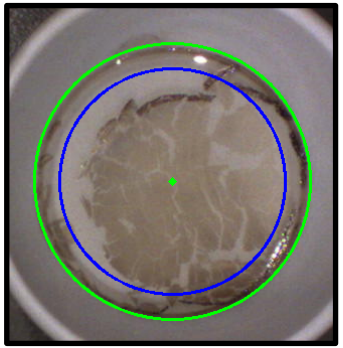
Unknown radius can still be determined by the parametric space as various radii from center create cone



Hough Circle works based on 'votes' to define the probable center of a circle.

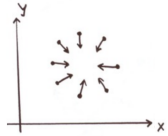
If we know the radius and the gradient of a point in an edge, then there is only one possible circle for which this would define. Any radius for that gradient would fall somewhere upon the line.

Hough Gradient increases speed of algo

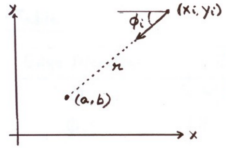


Gradient information can save lot of computation:

Edge Location (x_i, y_i)
Edge Direction ϕ_i

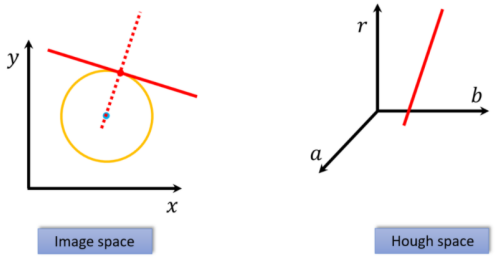


Assume radius is known:



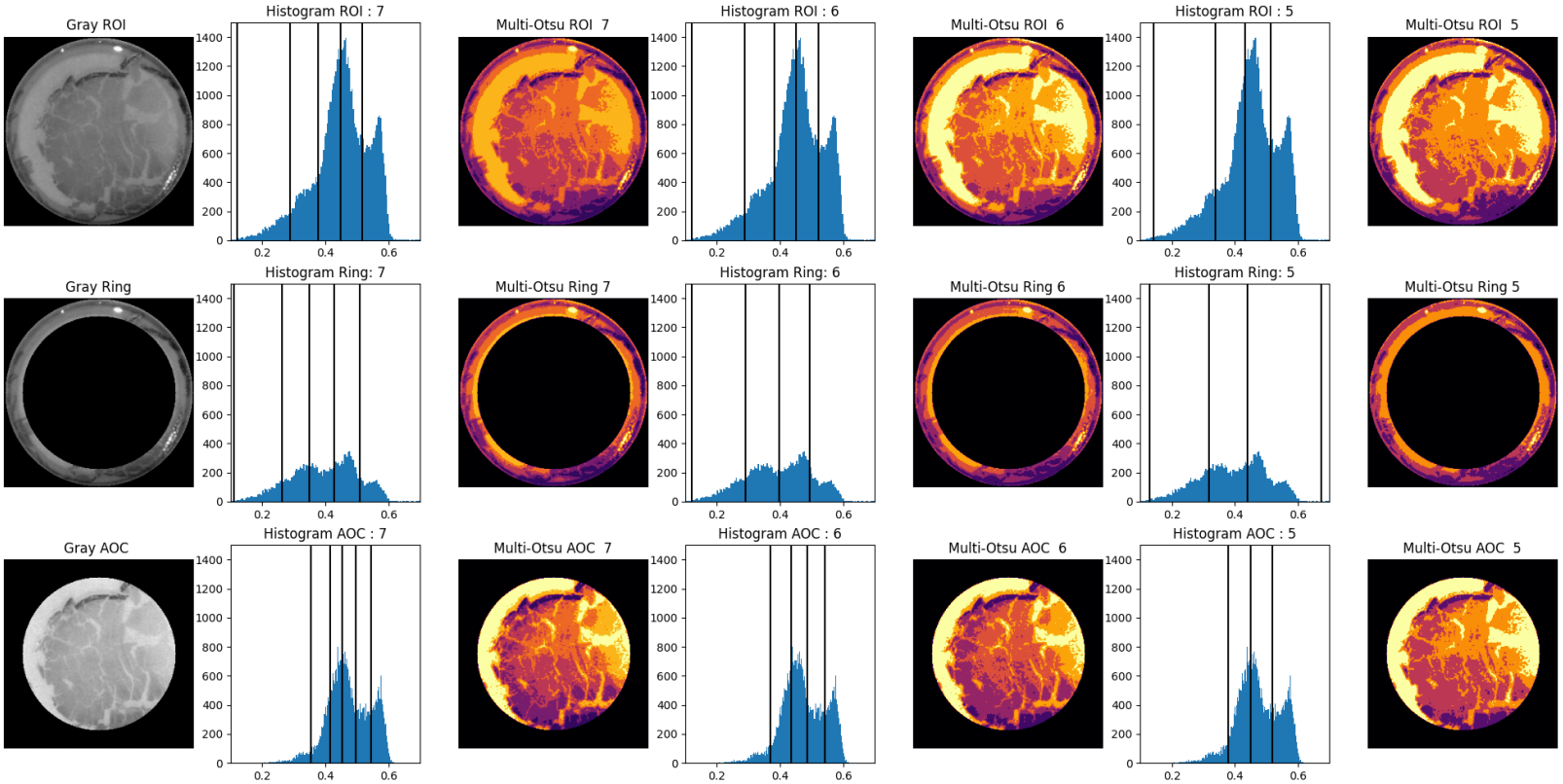
$$a = x - r\cos\phi$$
$$b = y - r\sin\phi$$

Need to increment only one point in accumulator!!

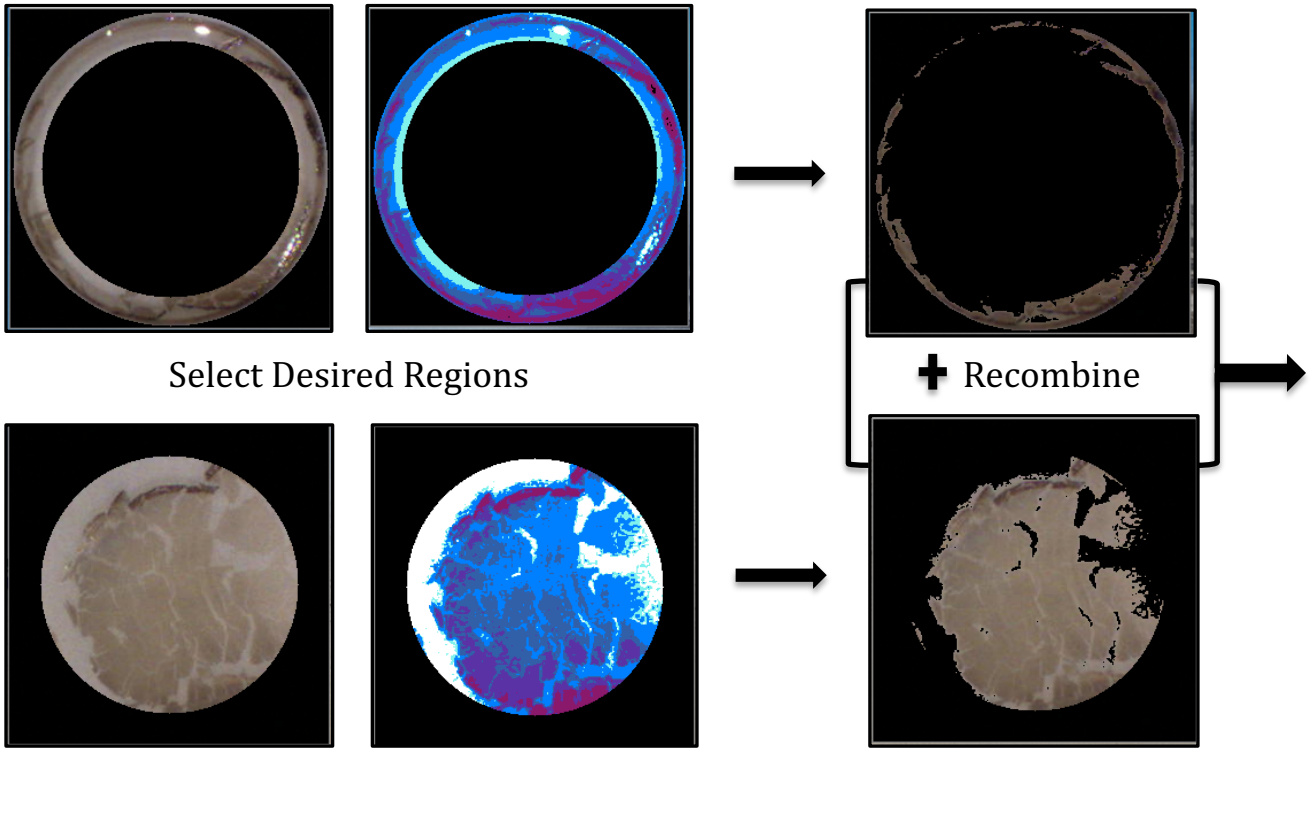


http://datahacker.rs/

Multi-Otsu Thresholding with Histograms



Manual Adjustment of Multi-Thresholding Technique



Multi-Otsu Thresholding

The algorithm exhaustively searches for the threshold that minimizes the intra-class variance, defined as a weighted sum of variances of the two classes:

$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t)$$

Weights ω_0 and ω_1 are the probabilities of the two classes separated by a threshold t ,and σ_0^2 and σ_1^2 are variances of these two classes.

The class probability $\omega_{0,1}(t)$ is computed from the L bins of the histogram:

$$\omega_0(t) = \sum_{i=0}^{t-1} p(i)$$

$$\omega_1(t) = \sum_{i=t}^{L-1} p(i)$$

For 2 classes, minimizing the intra-class variance is equivalent to maximizing inter-class variance:[2]

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2$$
$$= \omega_0(t)\omega_1(t)[\mu_0(t) - \mu_1(t)]^2$$

which is expressed in terms of class probabilities ω and class means μ, where the class means $\mu_0(t)$, $\mu_1(t)$ and μ_T are:

$$\mu_0(t) = \frac{\sum_{i=0}^{t-1} ip(i)}{\omega_0(t)}$$

$$\mu_1(t) = \frac{\sum_{i=t}^{L-1} ip(i)}{\omega_1(t)}$$

$$\mu_T = \sum_{i=0}^{L-1} ip(i)$$

The following relations can be easily verified:

$$\omega_0\mu_0 + \omega_1\mu_1 = \mu_T$$
$$\omega_0 + \omega_1 = 1$$

The class probabilities and class means can be computed iteratively. This idea yields an effective algorithm.

Algorithm [edit]

1. Compute histogram and probabilities of each intensity level
2. Set up initial $\omega_i(0)$ and $\mu_i(0)$
3. Step through all possible thresholds $t = 1, \ldots$ maximum intensity
    1. Update ω_i and μ_i
    2. Compute $\sigma_b^2(t)$
4. Desired threshold corresponds to the maximum $\sigma_b^2(t)$

Multi-Otsu Thresholding tends to run so many more iterations due to summation calculations and k-means limits unnecessary iterations.

The number of objects that change cluster is smaller than a given threshold

wikipedia.org

K-Means Color Quantization

3. K-Means

The K-Means algorithm is an unsupervised clustering algorithm that automatically clusters based on the similarity of individual data points. The K-Means algorithm is simple and easy to implement, and it still has good clustering effect and high processing efficiency for large data sets.

In the clustering process, the similarity between any two data points is measured by distance. Common distance measurement methods include Euclidean distance, Chebyshev distance, Manhattan distance, etc. The distance metric used in this paper is the Euclidean distance. The Euclidean distance is the linear distance between two points in the metric space. The calculation method is shown in formula (3):

$$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (3)$$

The basic idea of K-Means is introduced as follows:

(1) Select k data points from the data set containing n data points as the initial cluster center C_1, where the data points are the pixels in the image;

(2) Calculate the distance d between each data point and the cluster center separately. Suppose the data point coordinates are (x_1, y_1, z_1) and the cluster center coordinates are (x_2, y_2, z_2). The calculation method is shown in formula (4), and the data point with the smallest distance from the cluster center is classified as the same category as the cluster center;

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (4)$$

(3) Calculate the mean according to the existing data points of each category, and re-select the cluster center of each category according to formula (5);

$$C_i = \frac{1}{n_j}\sum_{x_j \in C_i} x_j \quad (5)$$

(4) This is repeated until the cluster center C_i does not change.

In summary, the purpose of K-means clustering is to divide the raw data into k classes $S = \{S_1, S_2 \ldots S_k\}$ given the number of classification groups k ($k \leqslant n$), on the numerical model, which is to find the minimum value of the Formula (6):

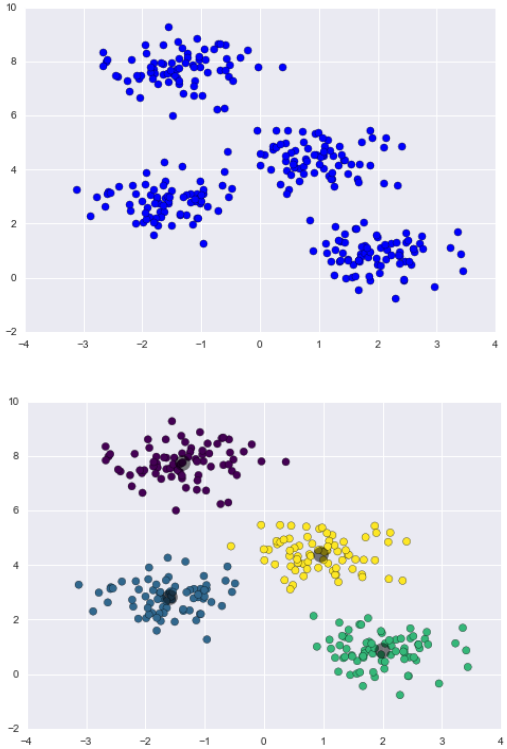
$$\arg\min \sum_{i=0}^{k} \sum_{x_j \in S_i} \| x_j - \mu_i \|^2 \quad (6)$$

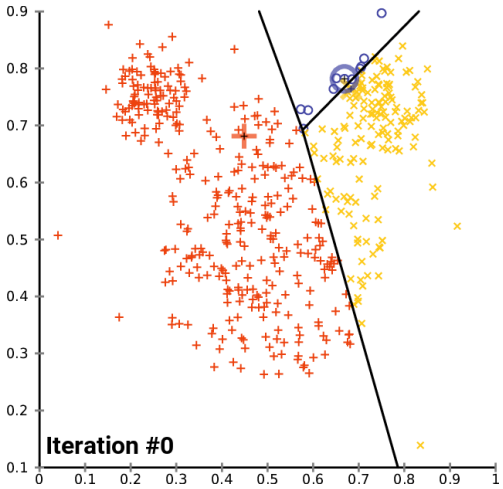
Where u_i represents the average of S_i

This paper mainly analyzes the clustering of pixels in RGB remote sensing images. The cluster center k represents the number of colors. This parameter needs to be set by yourself. The value of k must be smaller than the number of colors of the original RGB remote sensing image. In the two experiments in this paper, k is set to 64 and 32 respectively.

Guojian Cheng and Junjie Wei 2019 *J. Phys.: Conf. Ser.* **1213** 042012

Color Quantization with k-means





https://upload.wikimedia.org/wikipedia/commons/e/ea/K-means_convergence.gif

Step 1: define k value of clusters

Step 2: k number of random (or not) pixel values are chosen (centroids)

Step 3: Iterate through each data point (pixel) and determine through variances in color which cluster that pixel belongs and designate it.

Step 4: For each cluster, now calculate the average color and this becomes the new centroid / center for that cluster.

Step 5: Iterate through pixels with newly defined centroids and make sure each pixel is still within its proper cluster.

This process continues through a determined number of iterations or until the accuracy (epsilon) is reached stopping the program