Particle Analysis and Classification Techniques

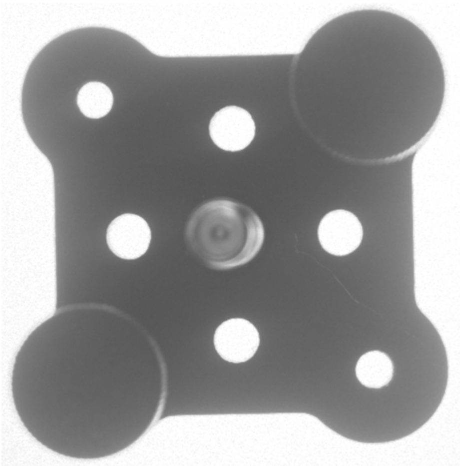
Perry C. West
President
Automated Vision Systems, Inc.

Course Outline

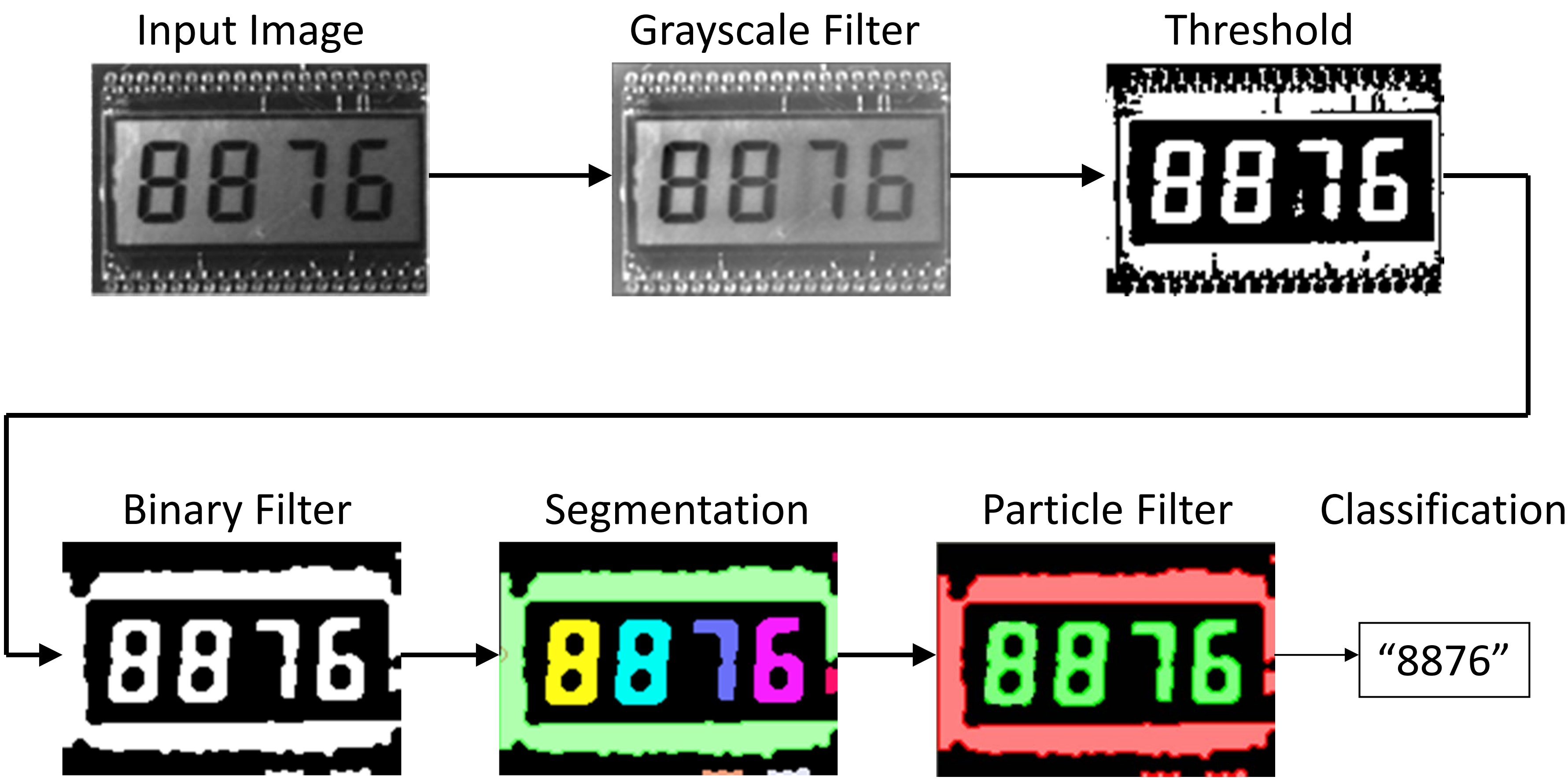
- Particle Analysis
  - Threshold Strategies
  - Segmentation
  - Morphology
  - Particle Features and Filters
- Classification Techniques
  - Feature Vectors
  - K-nearest neighbor
  - Statistical Classifiers
  - Neural Networks
- Frequency Domain
  - FFTs & filtering

What is Particle Analysis?

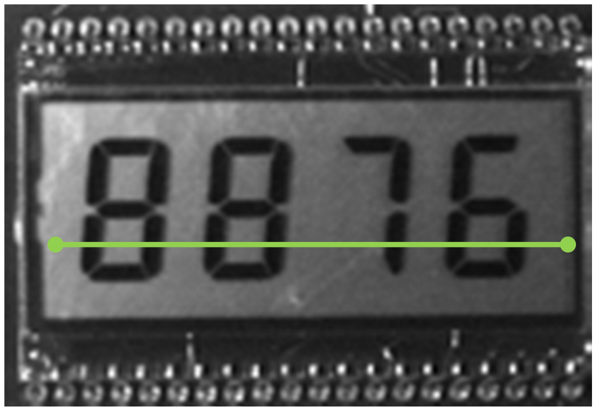
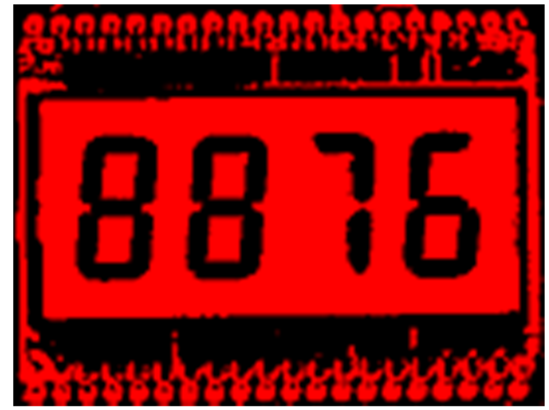
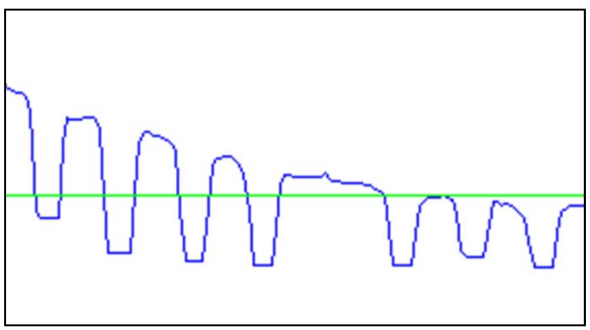
- Finding, counting, measuring, and classifying particles
  - A particle is a contiguous group of pixels that can be separated from the background by pixel intensity
- Also known as
  - Blob Analysis
  - Cell Analysis in medicine and biology
- As simple as counting holes or as complex as OCR



Steps in Particle Analysis

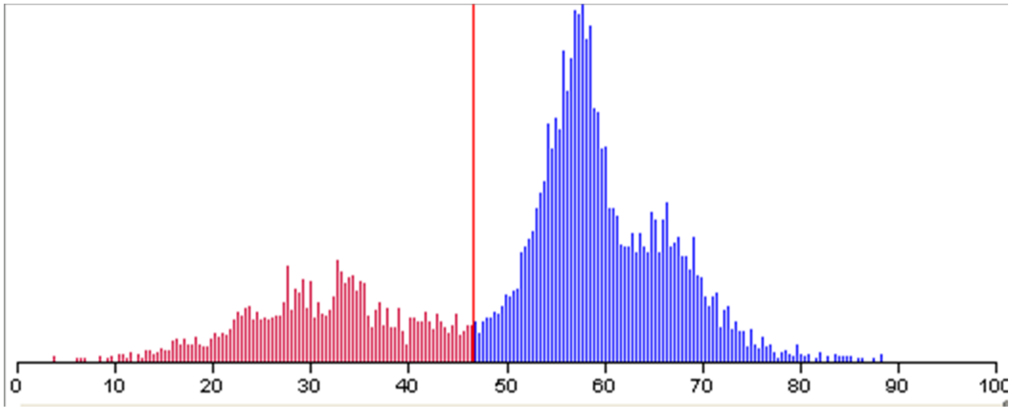


Thresholding

Original	Global Threshold	Local Threshold
		
	Fails due to light variation	aka Adaptive Threshold

Calculating A Threshold

- Histogram



$$\sigma^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t)$$

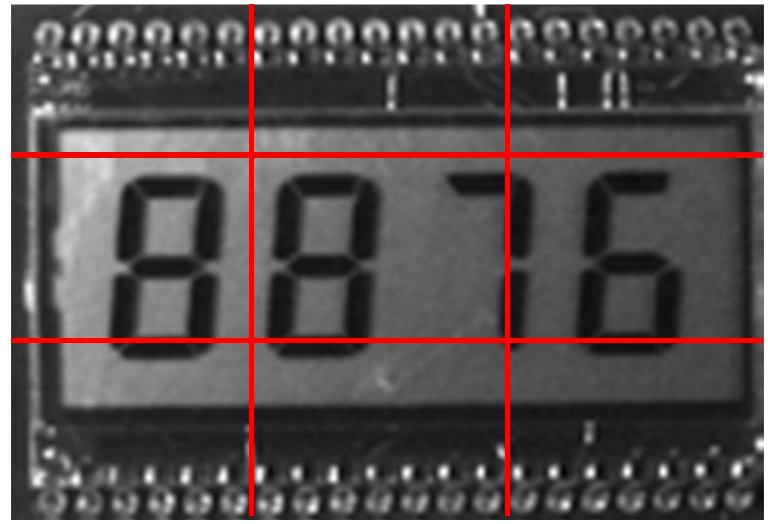
- Identifying peaks and valleys is ill-defined and unreliable
  - Are there 2, 3, or 4 peaks in this histogram?
- Otsu's method is one standard threshold calculation
  - Define a metric function, called intra-class variance

$$\sigma^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t)$$

  - Find threshold t that minimizes the metric
    - Splits the histogram into two groups with the smallest total variance

Locally Adaptive Threshold

- Chow and Kaneko method
- Divide the image into regions
- Calculate a threshold for each region, e.g., using Otsu's method
- Calculate the threshold at each pixel as the weighted average of the 4 closest regions
- Smoothly varying and efficient to calculate

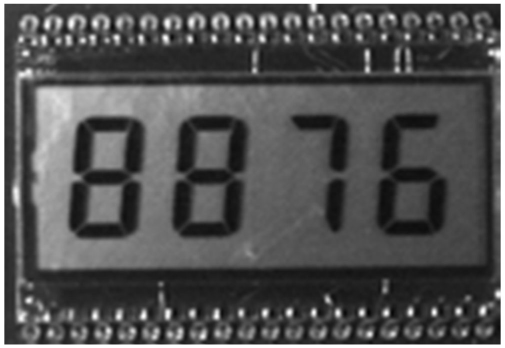


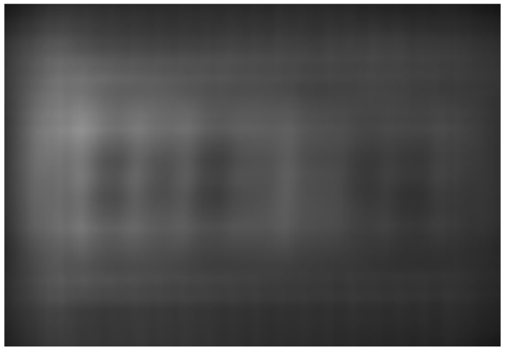
T11	T21	T31
T12	T22	T32
T13	T23	T33

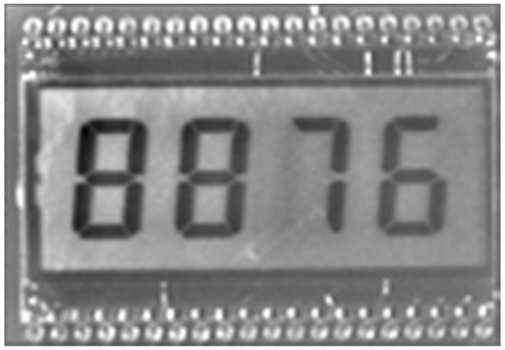
T = 0.4*T11 + 0.4*T21 + 0.1*T12 + 0.1*T22

Background Subtraction

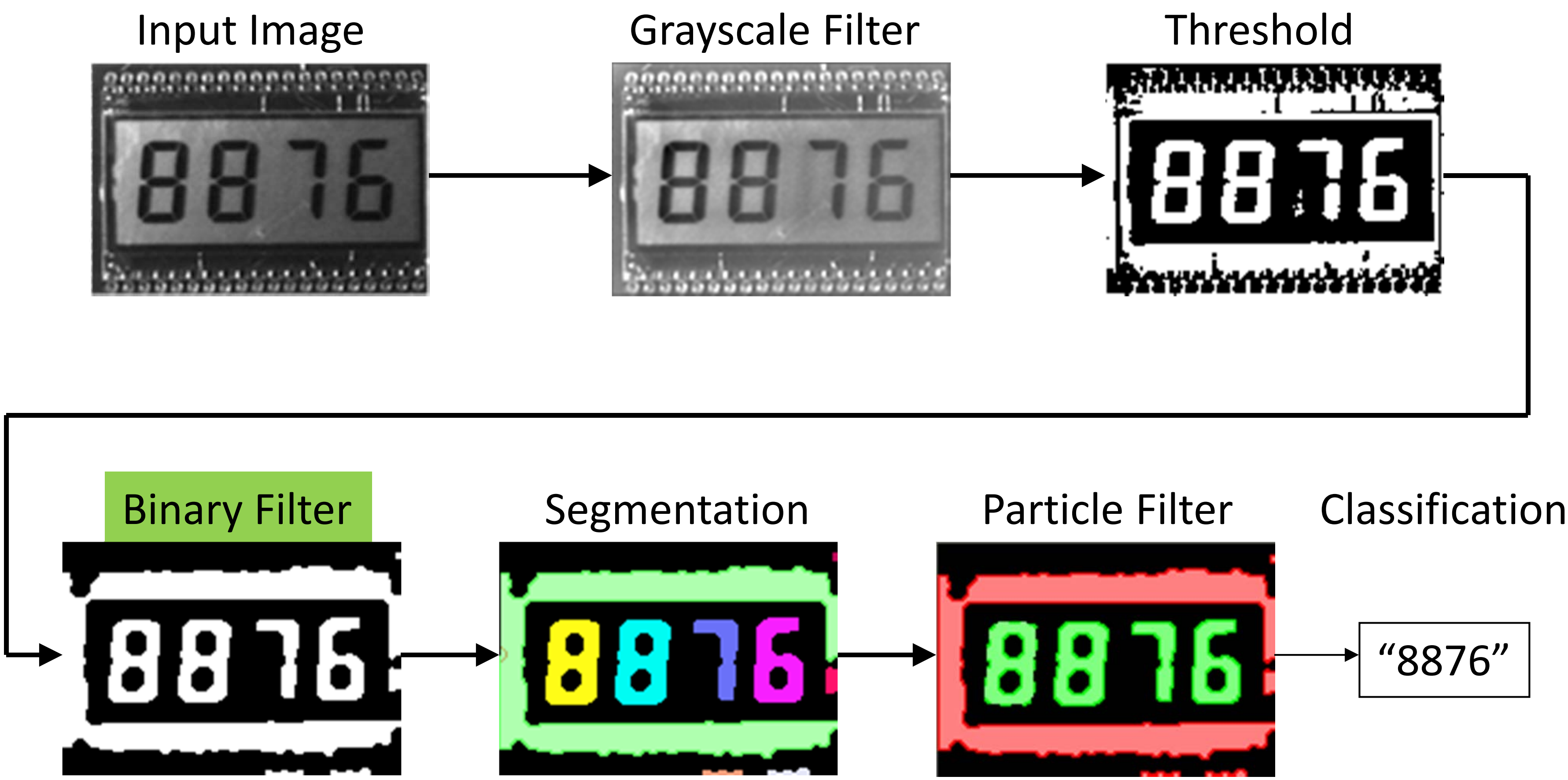
- Alternative to locally adaptive threshold
  - Removing the background variation, allows using global threshold
  - Basically a high pass filter
  - Light variation is very low frequency content to be removed
- Create background image using large smoothing filter
  - Filter should be 5 times larger than particle of interest
- Subtract the background image from the original image.
  - Add an offset so that negative pixel values can be displayed
  - Now you can use global threshold





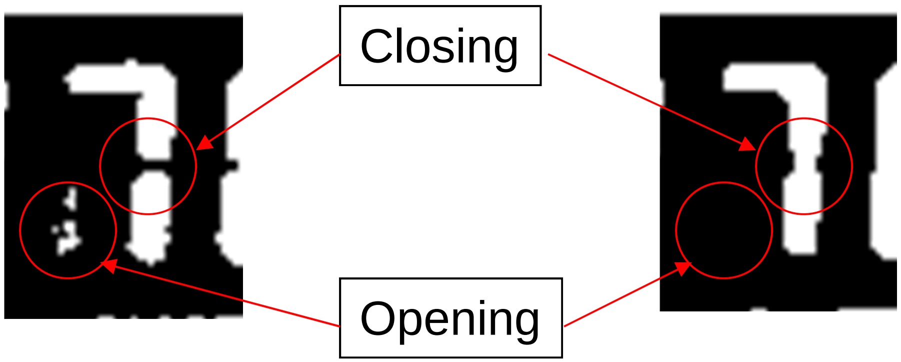


Steps in Particle Analysis

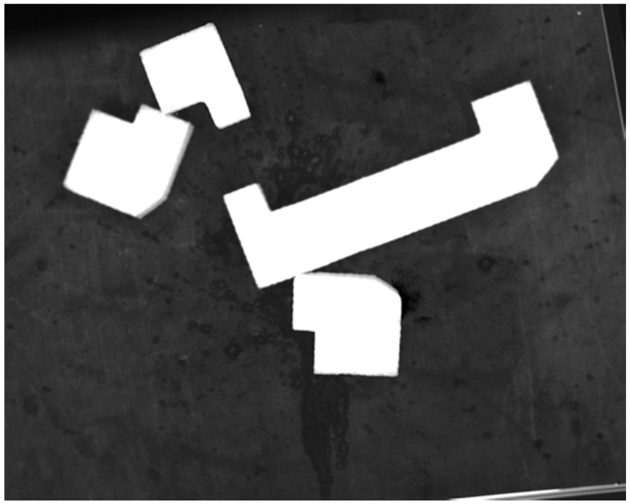


Morphology - Opening and Closing

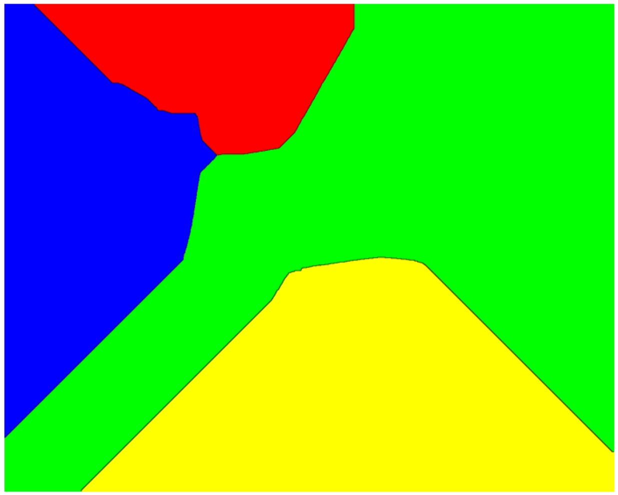
- Particles that erroneously touch or break can be filtered using morphology

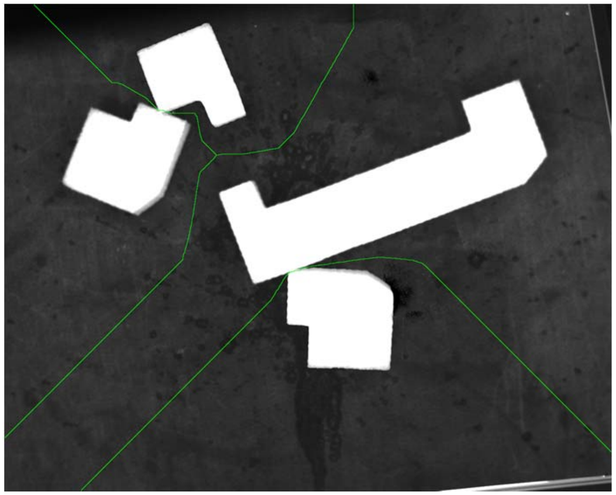


Problem: Touching Parts



- Solution – Watershed Transform

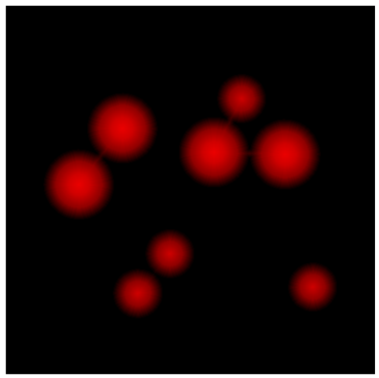




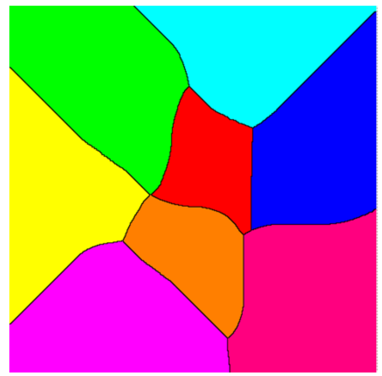
Watershed Transform

Original Image

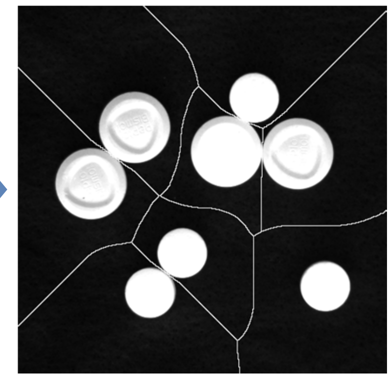
Distance Transform



Watershed Lines

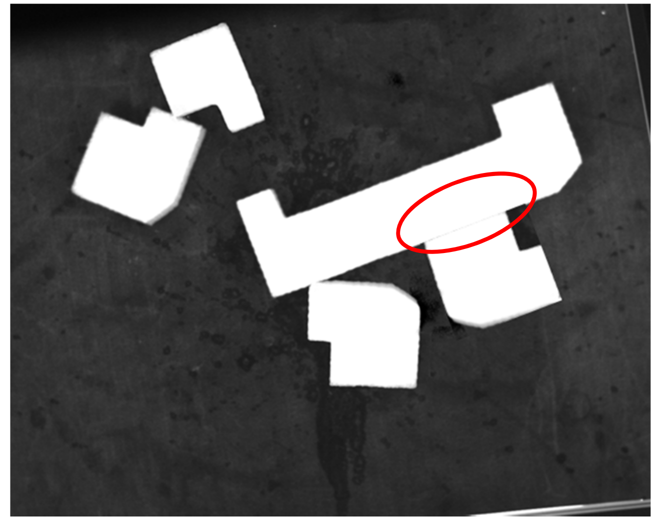


Divided Particles

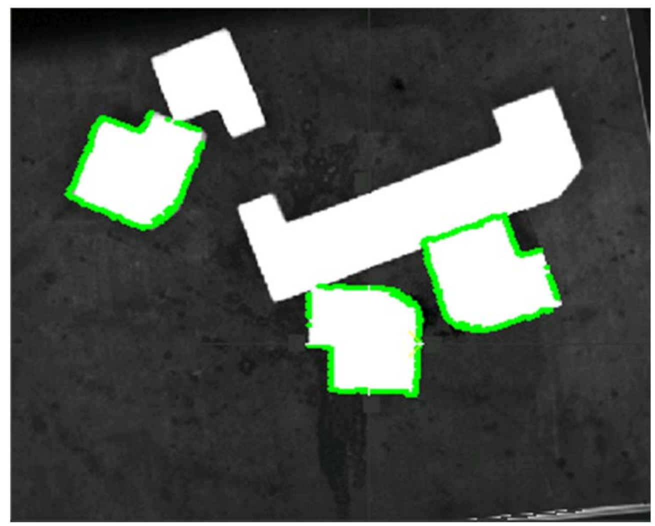


Inseparable Particles

- When flat surfaces touch, morphology and particle analysis will fail

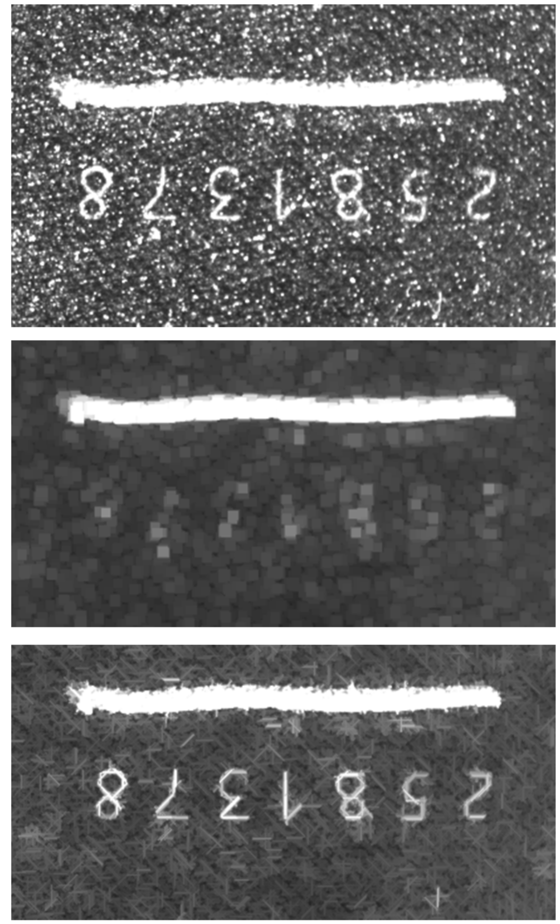


- Solution is pattern find

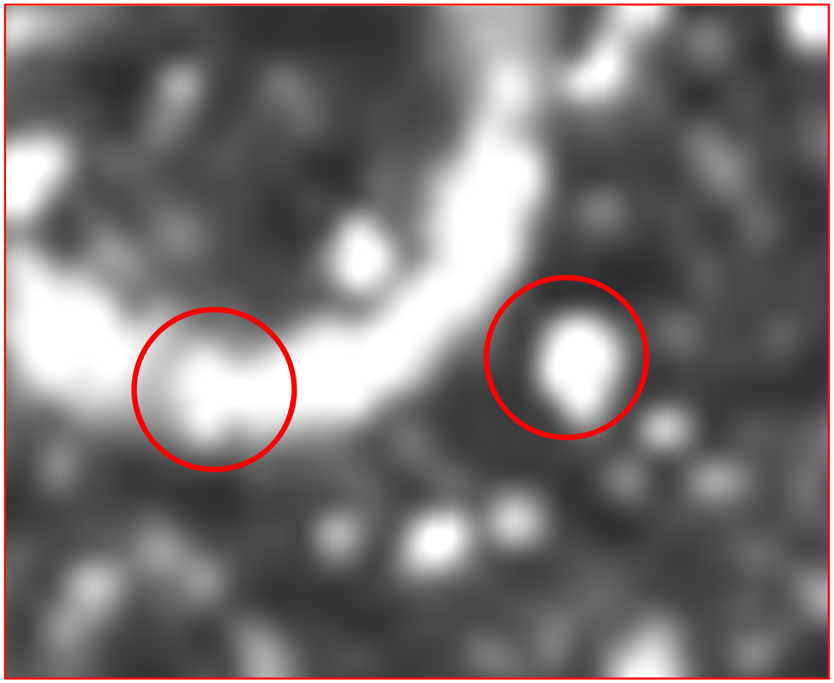


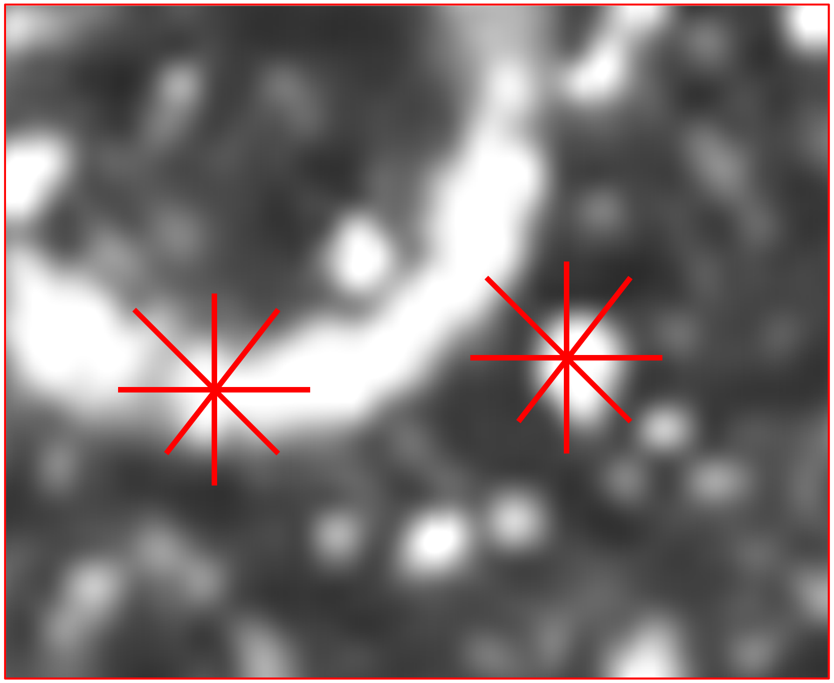
Morphology Structuring Elements

- Typically, morphology is based on symmetric circles or squares
  - Can not distinguish dots from lines
- Morphology using multiple linear structuring elements at different angles solves the problem



Morphology Structuring Elements





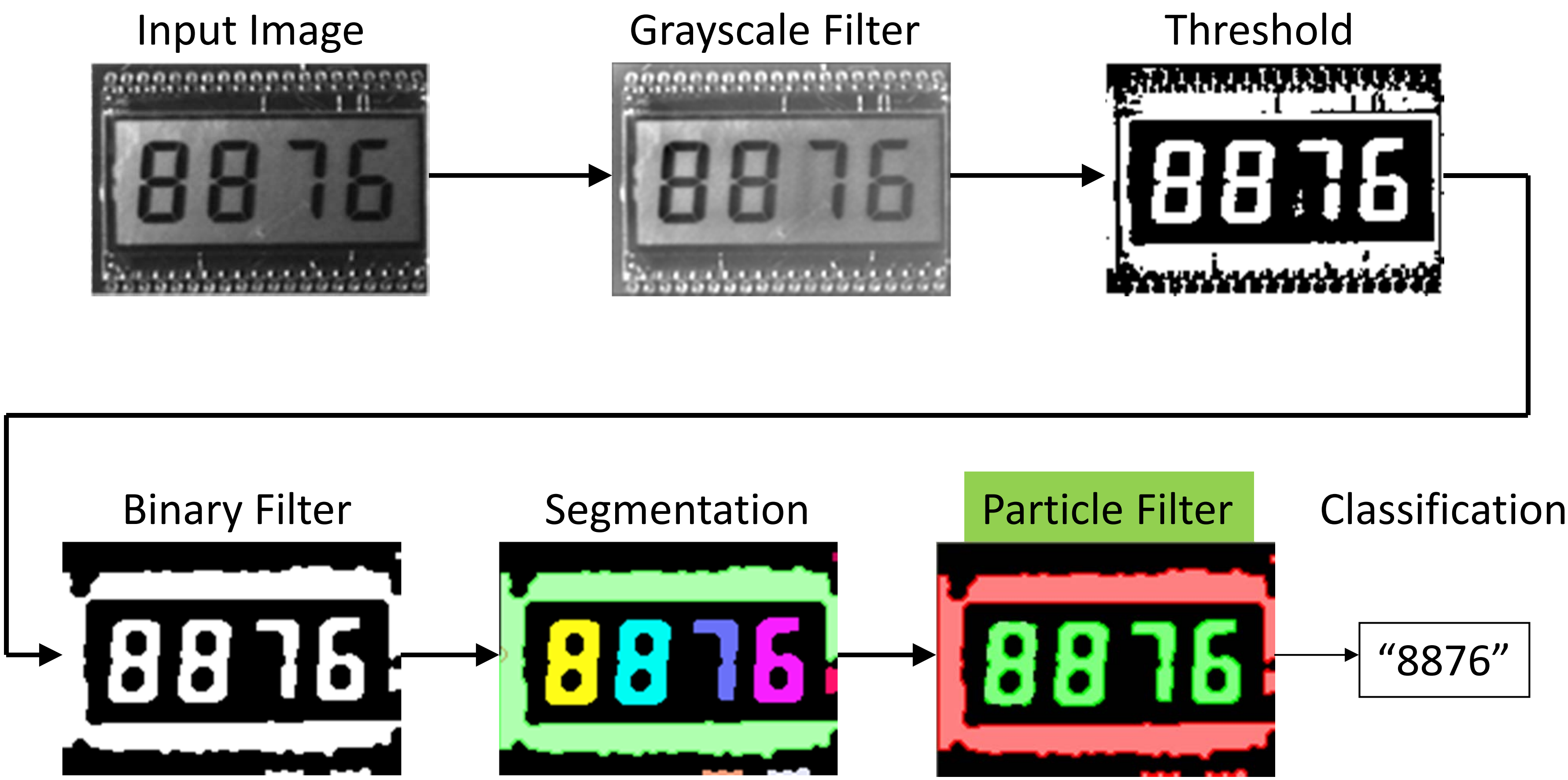
- Set center pixel to min of all pixels in circle
- Eliminates spot AND character stroke

- Set center pixel to max of min of each line
- Eliminates spot and keeps line

Segmentation

- Connected thresholded pixels are grouped together into geometric objects called particles, or blobs, or cells
- This is the step that moves from image-oriented processing to object-oriented processing
- Blobs are stored in memory as
  - Polygons or
  - Run lengths
- aka Connected Components Labeling

Steps in Particle Analysis



Particle Filters

- Features
  - Area, height, width, perimeter, centroid, circularity, …
- Filter rules
  - min < area < max   AND
  - min < width < max AND
  - …
  - If all rules pass, then the particle passes the filter

Typical Particle Features

- Area
- Perimeter
- Moments
  - Central Moments
  - Hu Moments
- Feret Diameters
  - aka Caliper diameters
- Number of Holes
- Shape Equivalences
    - Circle, Ellipse, Rectangle, Convex Hull
    - Ratios between particle feature and equivalent shape feature
- Location and Orientation
- ISO 9276-6
  - Descriptive and quantitative representation of particle shape and morphology

Feature Invariance

- Feature has same numeric value even as the object
  - Translates (moves in x and y)
  - Rotates
  - Scales
  - Flips symmetry
- May enable fast classification even with rotating and scaling objects

	Translation	Rotation	Scale
Area	x	x	
Perimeter	x	x	
Centroid		x	x
Eqv. Ellipse Axes	x		x
Max Feret Diam.	x	x	
Hu Moment	x	x	x

Moments

$$M_{ij} = \sum_x \sum_y x^i y^j I(x,y)$$

- M_{00} is area
- Centroid is (M_{10}/M_{00}, M_{01}/M_{00})
- Higher order moments are usually central moments
  - Translation invariant

$$\mu_{ij} = \sum_x \sum_y (x - x_c)^i (y - y_c)^j I(x,y)$$

- Normalized moments are scale invariant

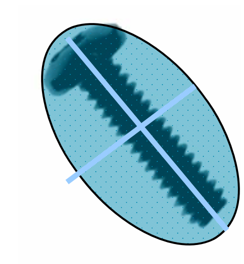
$$\eta_{ij} = \mu_{ij} / (area)^{1+(i+j)/2}$$

- Hu moments are rotationally invariant
  - 7th Hu moment is useful for detecting mirror symmetry

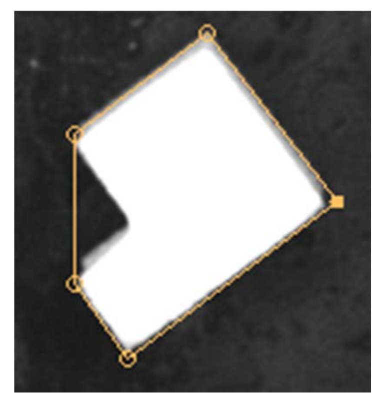
Shape Equivalences

- Equivalent Circle
  - Circle with same area as particle
  - Circularity: 1 for circle, <1 for other shapes

$$C = \sqrt{4\pi Area / Perimeter^2}$$

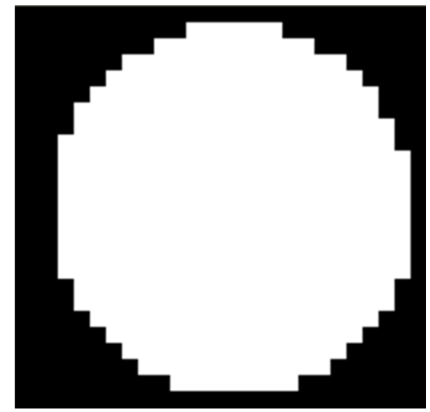


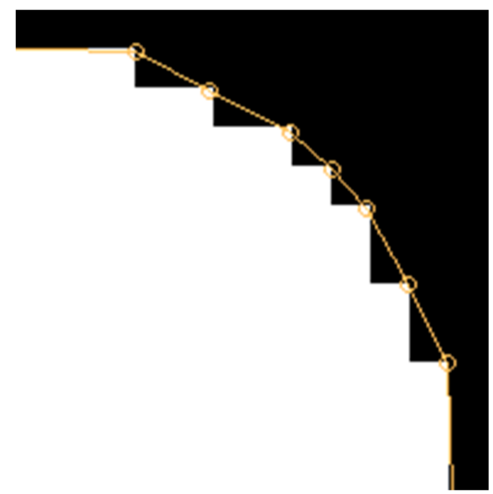
- Equivalent ellipse
  - Ellipse with same 1st and 2nd order moments as the particle
  - Orientation of major axis is useful
  - Major/minor axis ratio is useful
- Convex Hull
  - Bounding convex polygon
  - Solidity = $Area_{convex}$ / Area
  - Convexity = $Perimeter_{convex}$ / Perimeter



Perimeter is Tricky

- Discrete pixel boundaries make it difficult to define and calculate consistently
- Counting every horizontal and vertical edge on this circle gives perimeter of 4*d rather than pi*d





- Connecting the vertices gives a better result, though still not a perfect circular perimeter

Features and Holes

- Holes can be segmented as their own particles
- Holes can included or excluded from feature calculation

Centroid with hole

Centroid with hole filled in

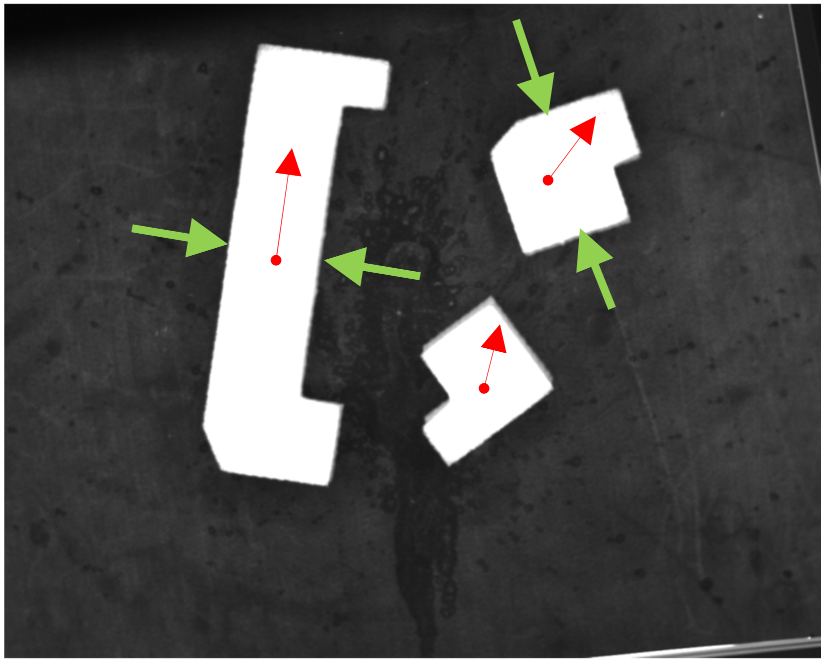
- This is where moments come in so handy!
  - Filled X centroid is just $(M_{10}+Hole\_M_{10})/(M_{00}+Hole\_M_{00})$

Problem: Find Grab Point

- Must determine position and orientation of each part

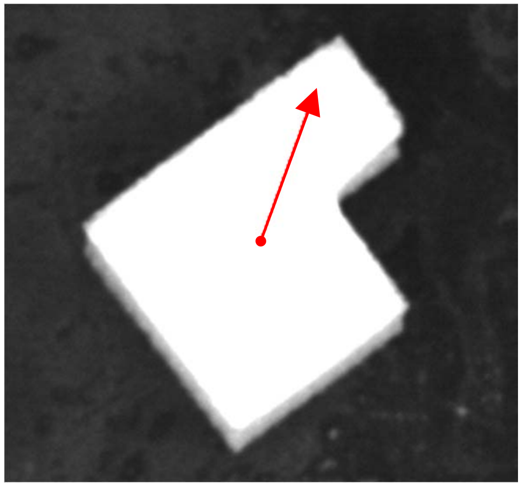
Solution: Particle Analysis

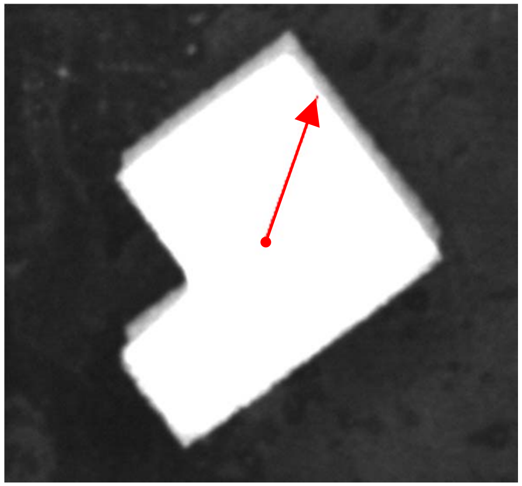
- Define a coordinate system
  - Centroid
  - Angle of major axis of equivalent ellipse
- Define grab point relative to coordinate system



Problem: Distinguish Orientations

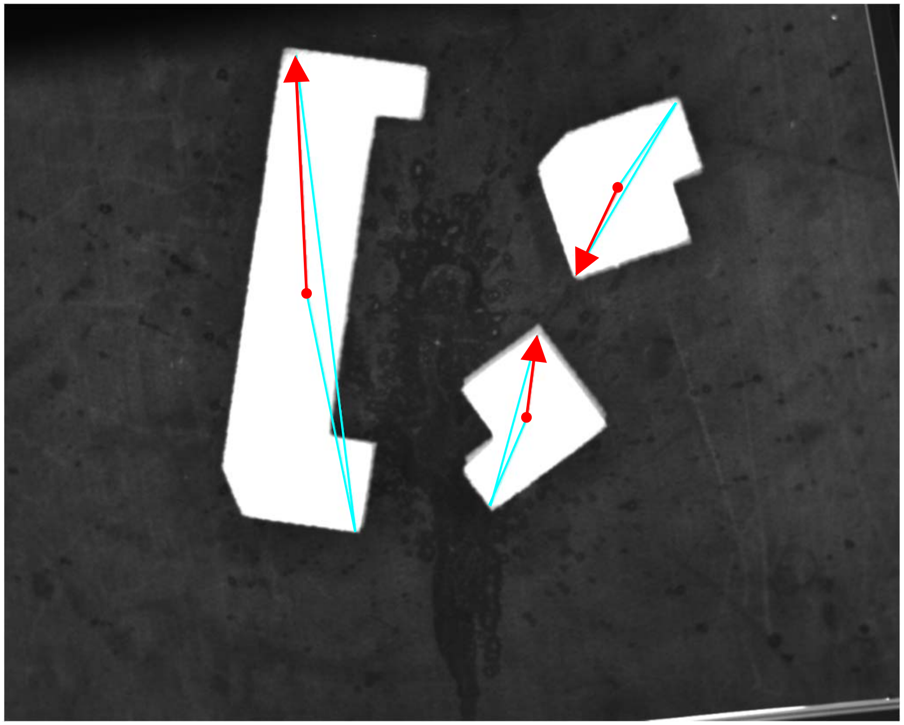
Particle Orientation based on the equivalent ellipse is limited to 0 to 180 degrees, therefore all orientations cannot be uniquely identified





Solution: Find More Features

- Is centroid to the left or right of the max Feret diameter?
- Of course Pattern Find is another solution, but is much slower.



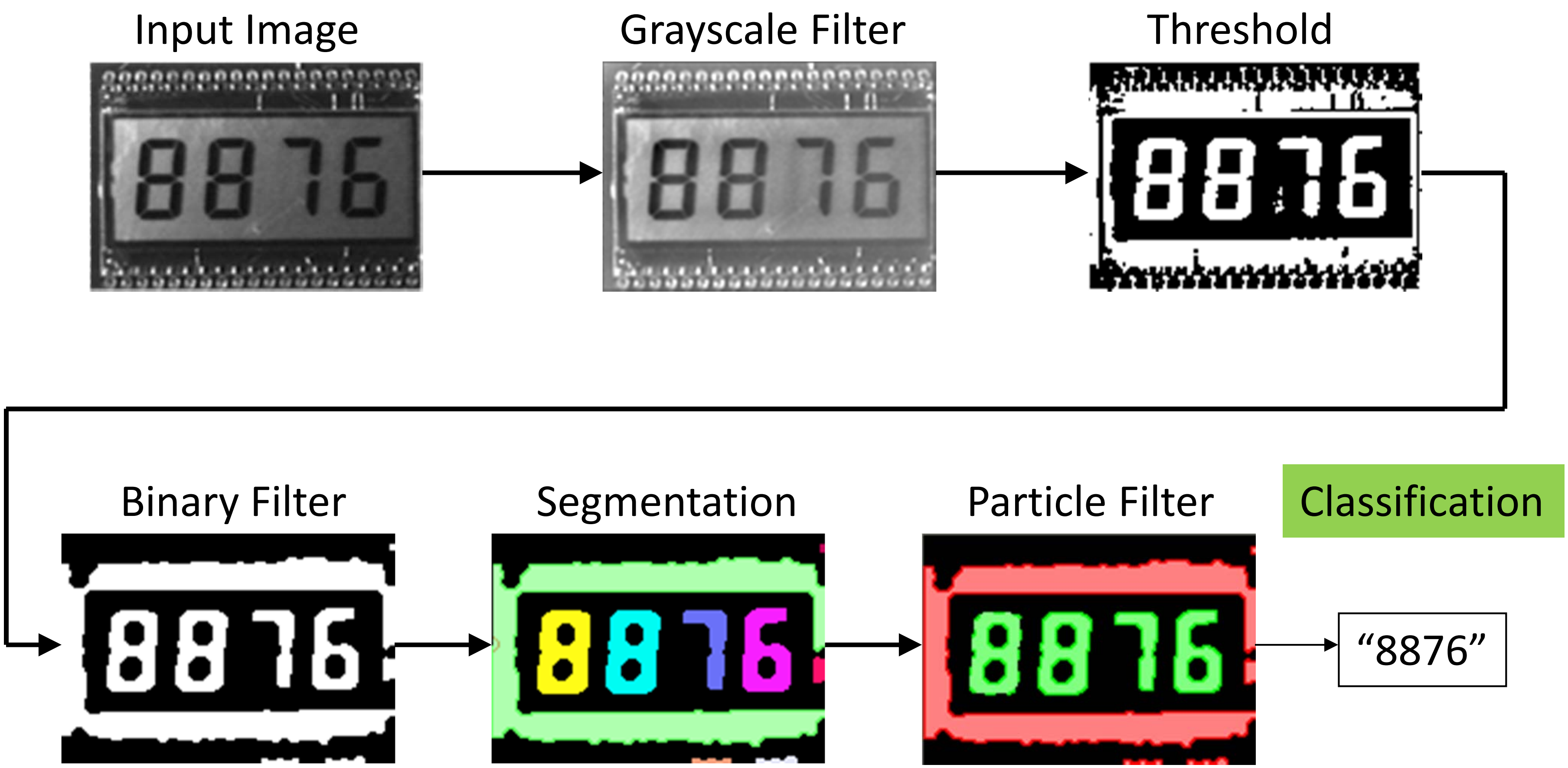
Feature Accuracy

- For an circular particle

	3 Sigma accuracy
Centroid x & y	$3\sigma = 0.6 / \sqrt{diameter}$
Diameter	$3\sigma = 0.78 / \sqrt{diameter}$

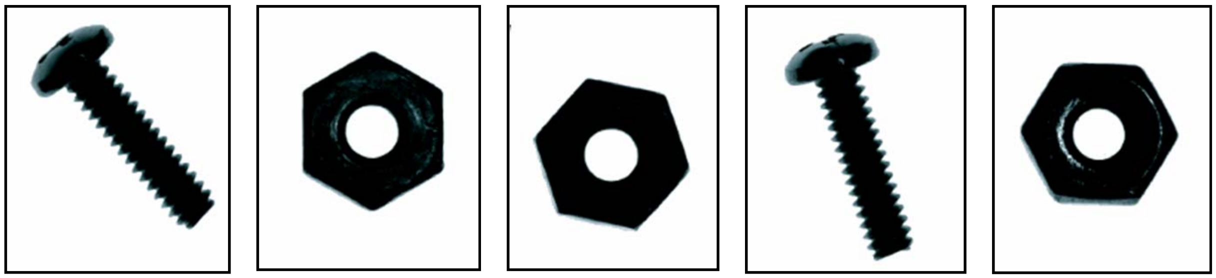
- Based on
  - +-0.5 pixel accuracy on each border point
  - Averaging all the border points
  - Central Limit Theorem

Steps in Particle Analysis

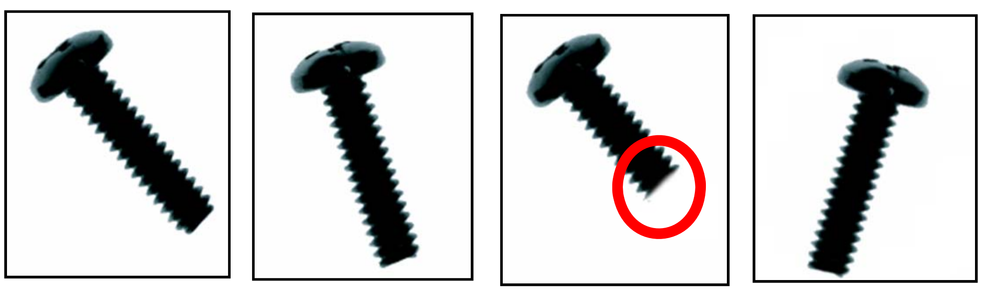


Classification Uses

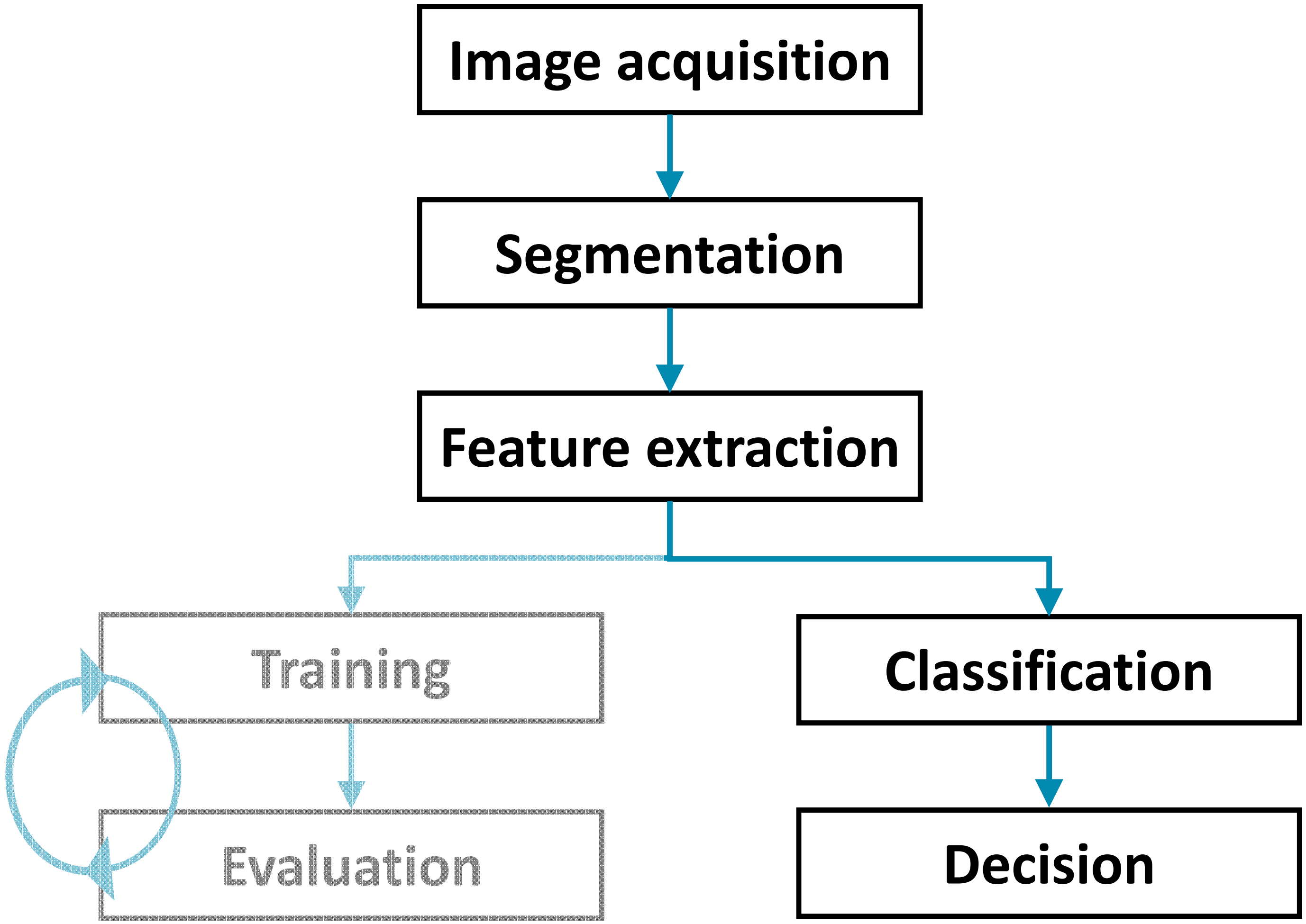
- Part sorting



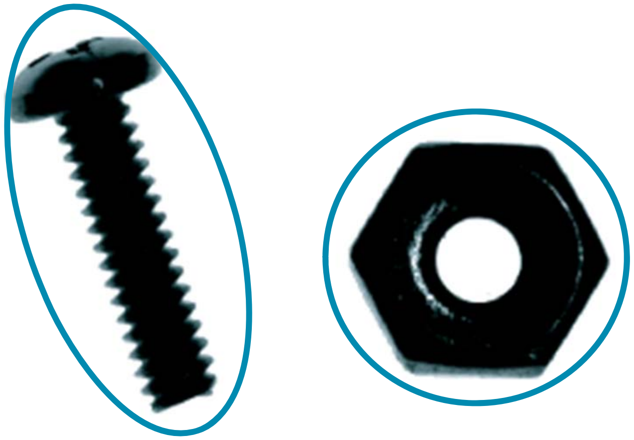
- Inspection



Classification Steps



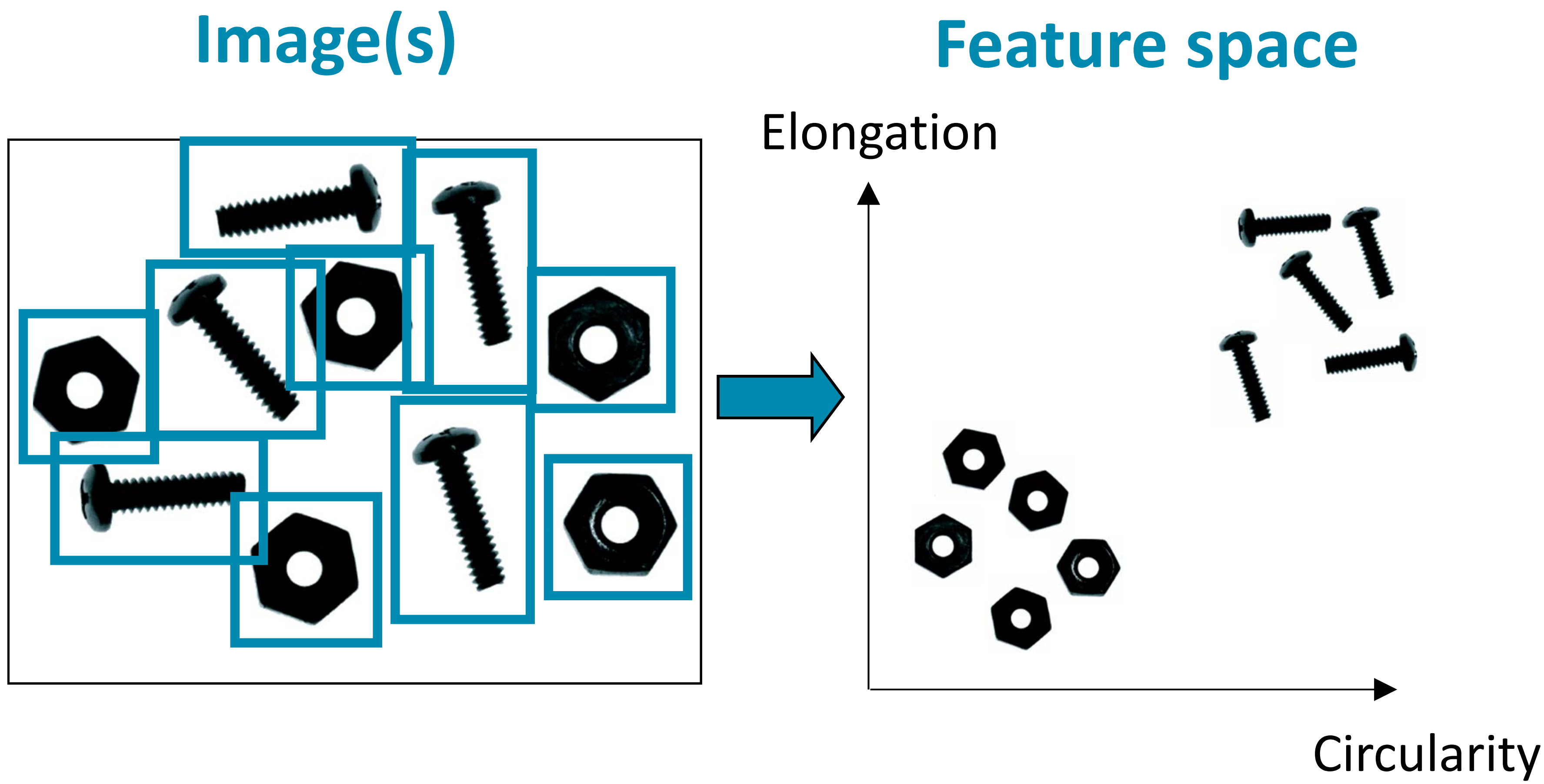
Feature Extraction



Class	Circularity	Elongation
Nut	1.1	1.0
Bolt	1.9	3.3

- Features are
  - Particle shape and size properties
  - Similar within a class
  - Different between classes
- "Class" is the label assigned to group of similar objects, such as nut or bolt

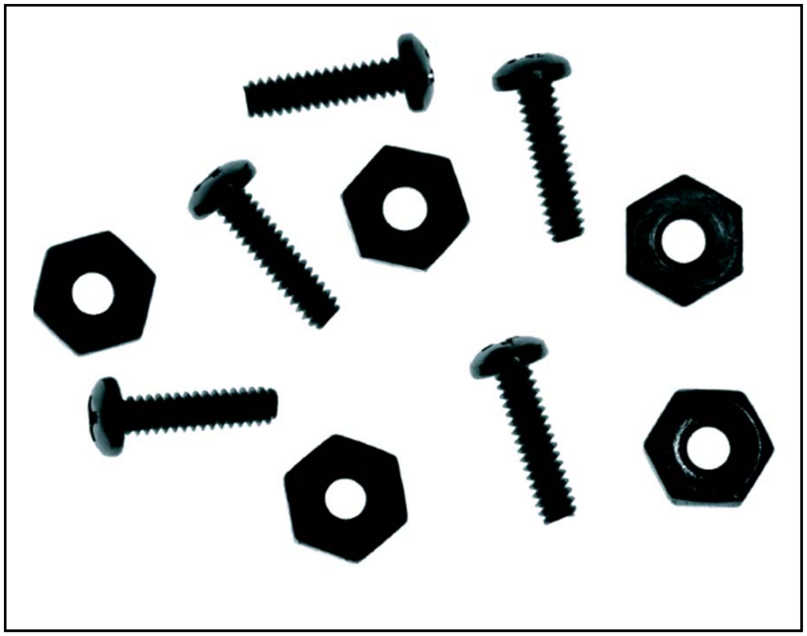
Training



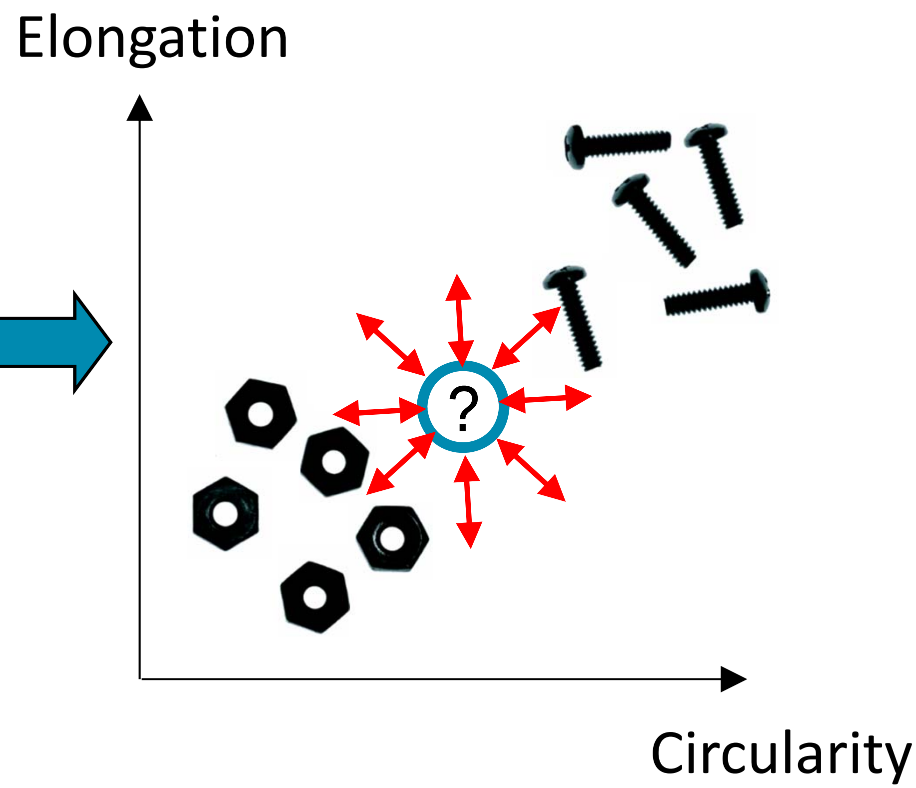
Train - Learn the range of each class in feature space

Classification

Image(s)



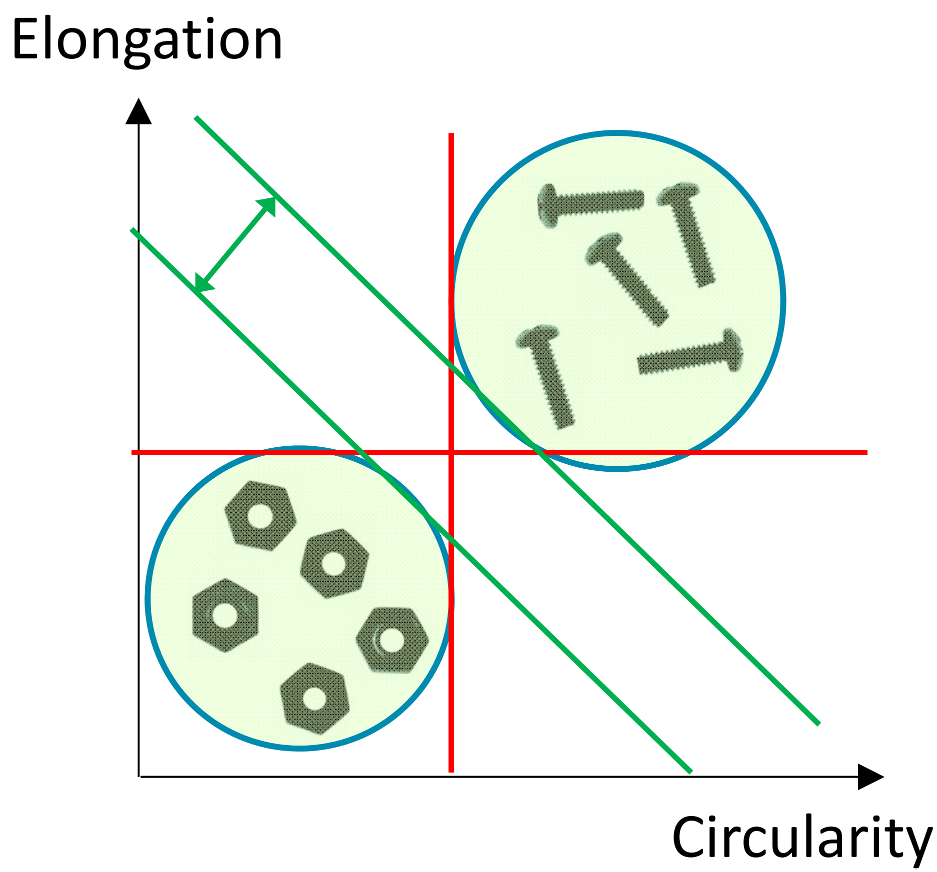
Feature space



Which class does it belong to?

Classification vs. Filtering

- Filtering – Red lines
  - Tests features one at a time
  - No separation in elongation or circularity alone
- Classification – Green Lines
  - Tests multiple features at once
  - Weighted sum of features
  - N dimensional thinking!

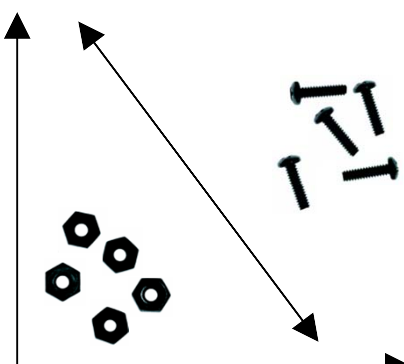


N-dimensional Feature Space

- Any number of features can be extracted from an object
- Features are grouped together into mathematical vectors
  - (feature1, feature2, feature3, …., featureN)
- A vector with N numbers is a point in an N-dimensional space
- We usually draw examples in 2-dimensional spaces, but all the ideas can be extended into N-dimensions
  - Euclidean distance

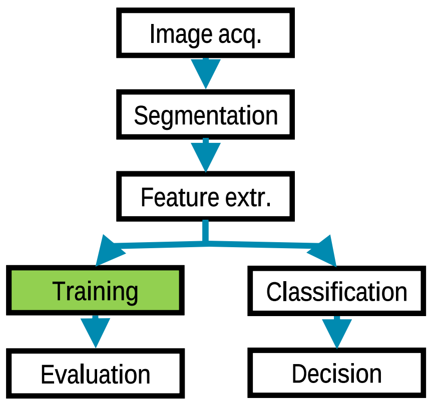
$$d_N(v1, v2) = \sqrt{\sum_{i=1}^{N} (v1_i - v2_i)^2}$$

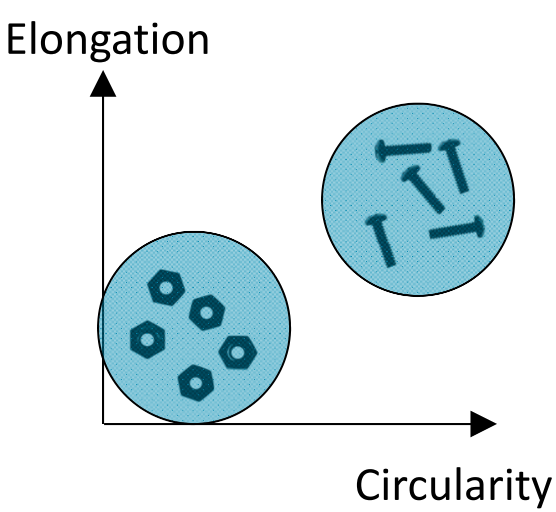
  - A line that divides two classes in 2 dimensions becomes a plane in 3 dimensions and a hyperplane in N dimensions



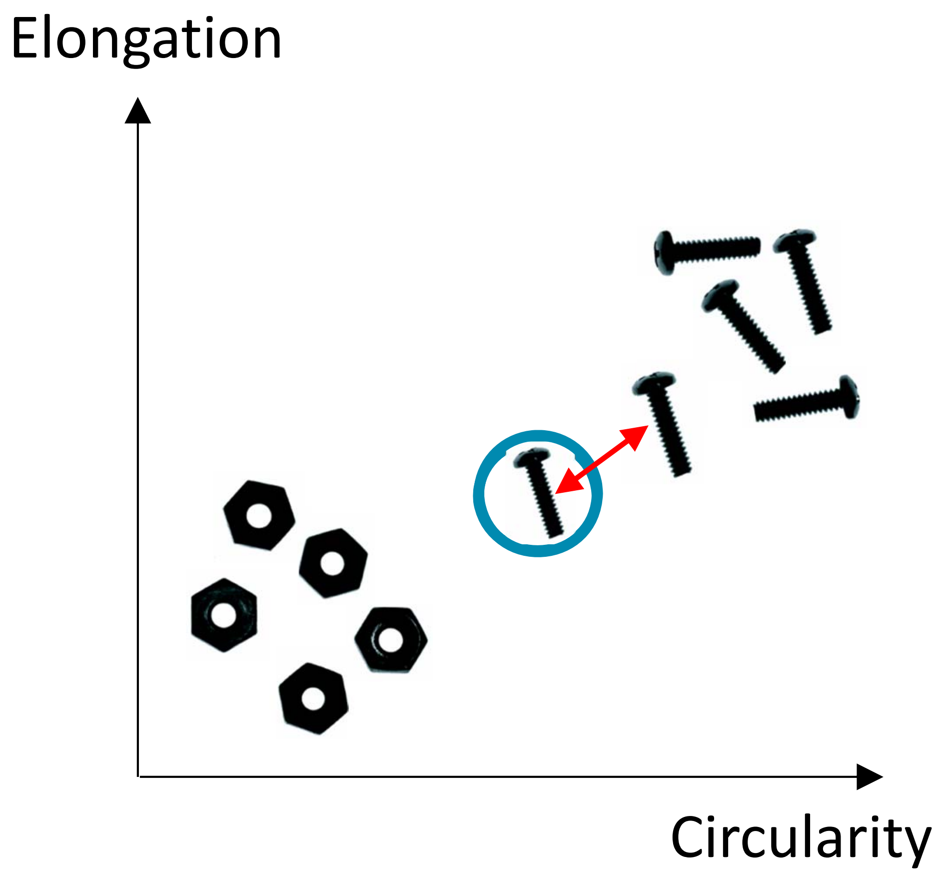
Classification Methods

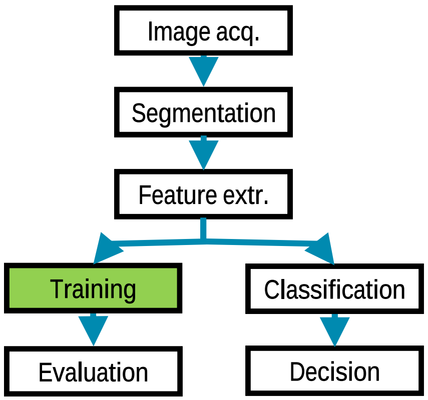
- Nearest Neighbor
- K-Nearest Neighbor
- Linear Statistical Classifiers
- Neural Network





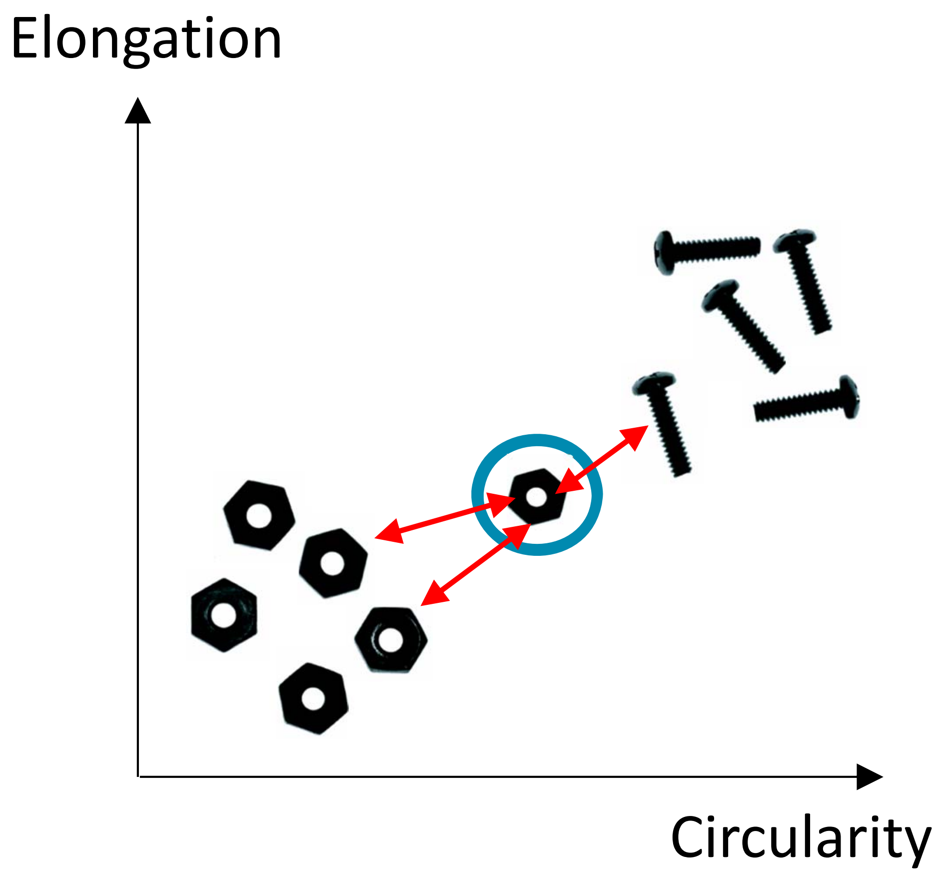
Nearest Neighbor Classifier

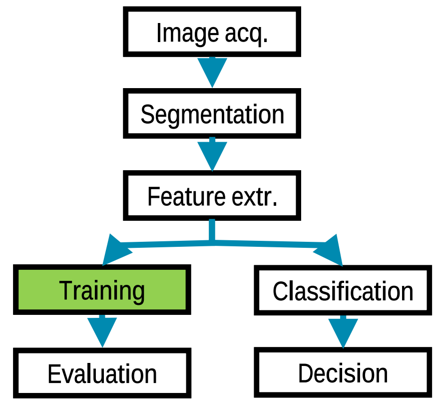




- Compare to EVERY training example to find best match
- Nearest neighbor in Euclidean distance

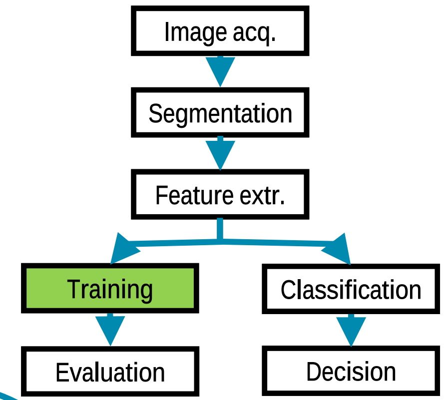
K-Nearest Neighbor Classifier



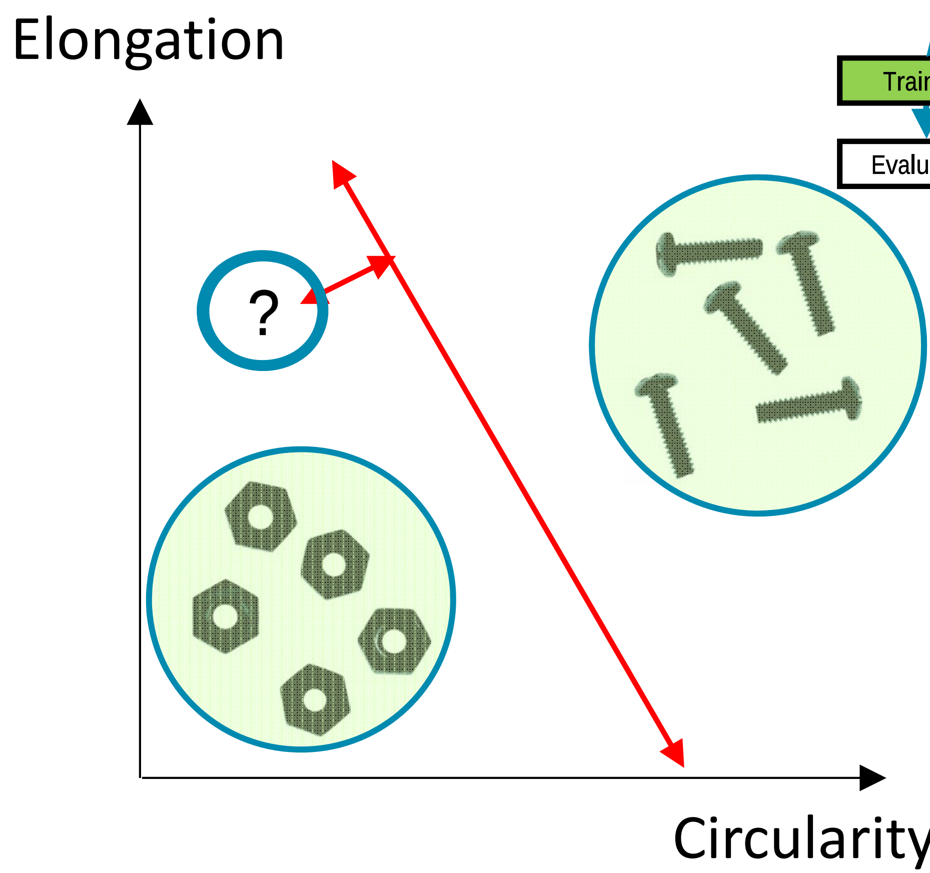


- Majority vote amongst K nearest neighbors
- Less noise sensitive than 1 nearest neighbor

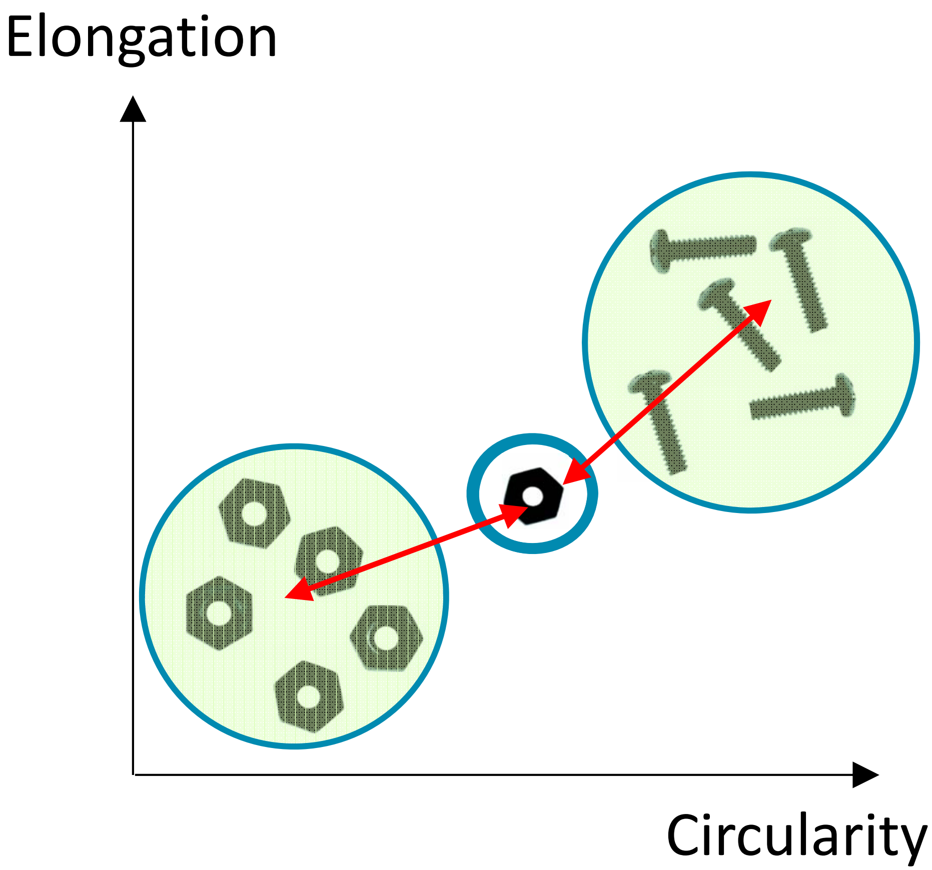
Linear Statistical Classifiers

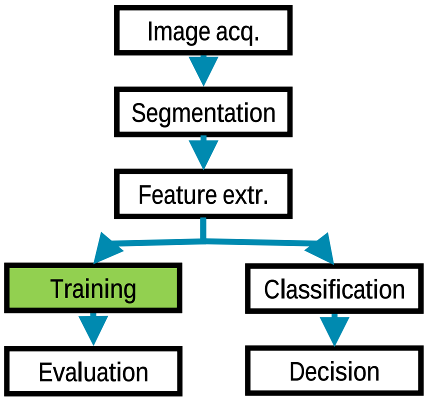


- Uses class statistics
  - Mean
  - Variance
  - Not examples
- Trains a dividing line
- Fast to compute
  - No searching
- Algorithms
  - SVM
  - Bayes
  - Fisher
  - Perceptron



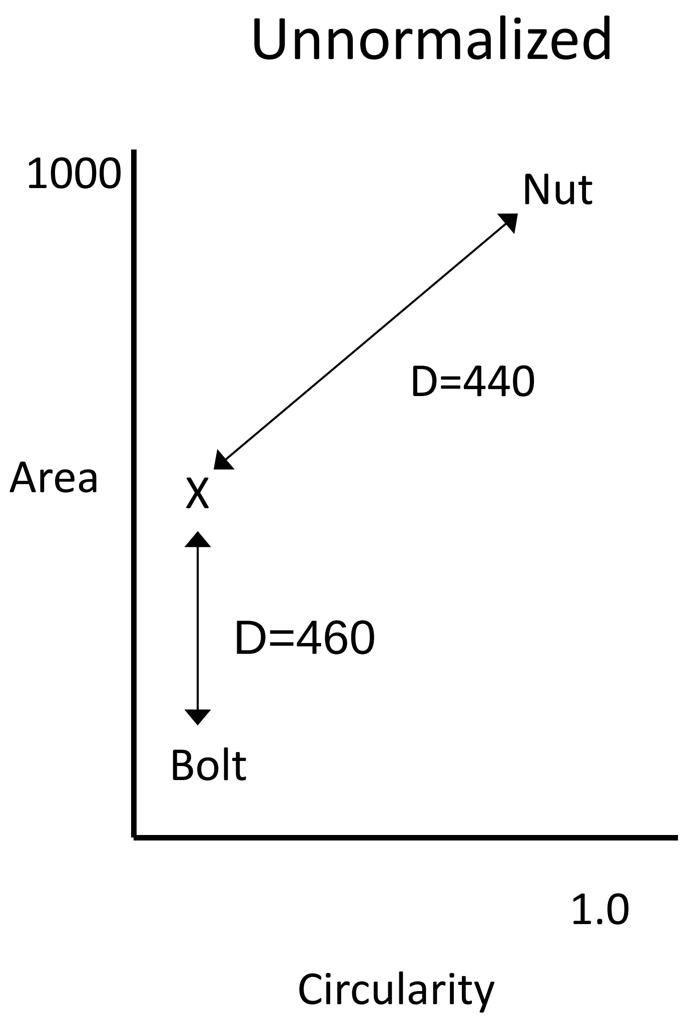
Minimum Mean Distance

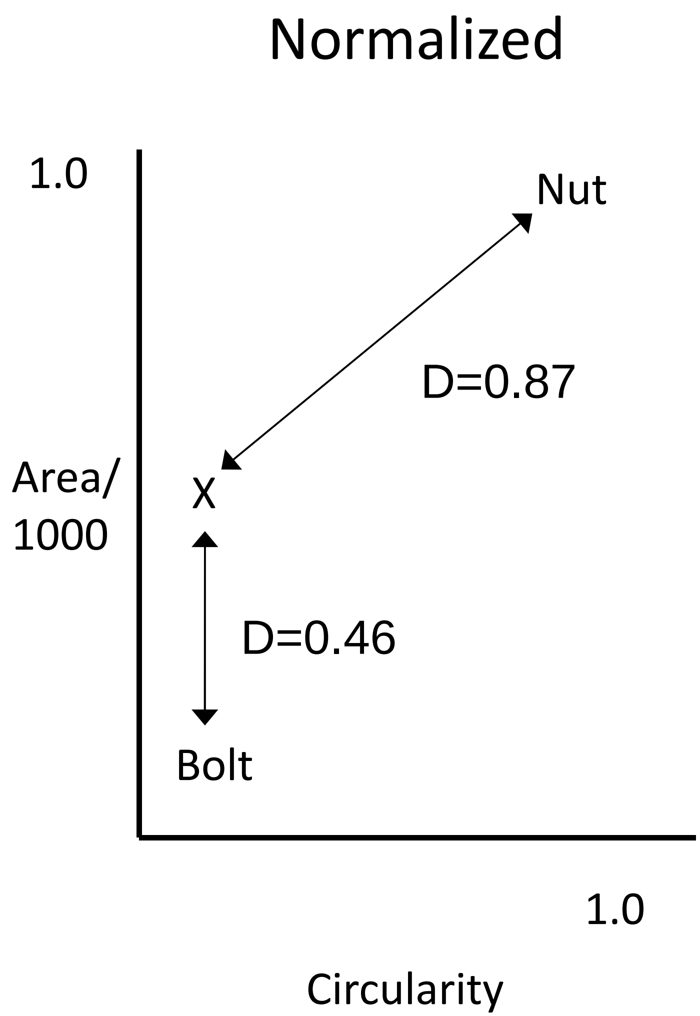




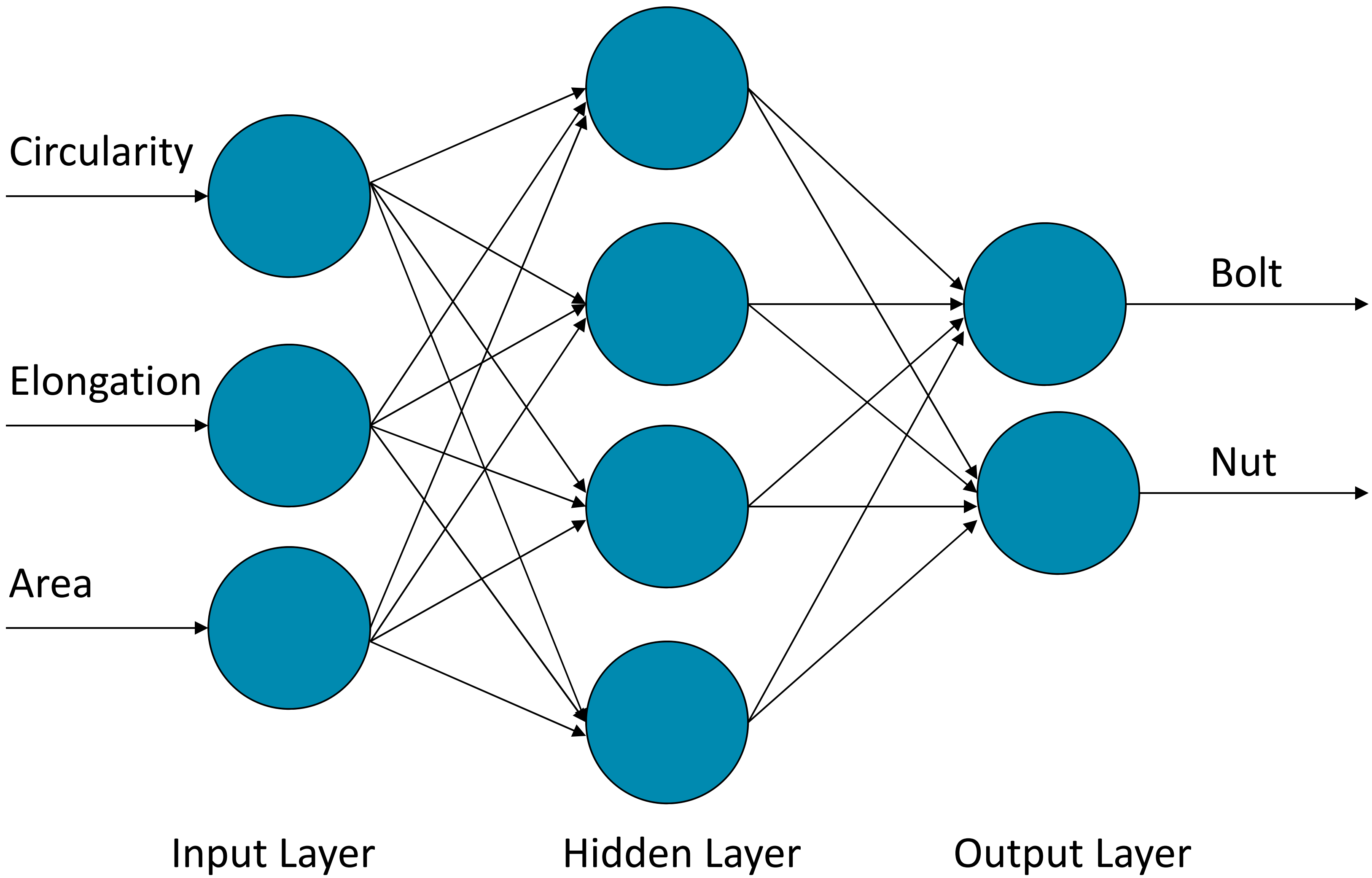
Calculate Euclidean distance to mean value of each class

Feature Normalization

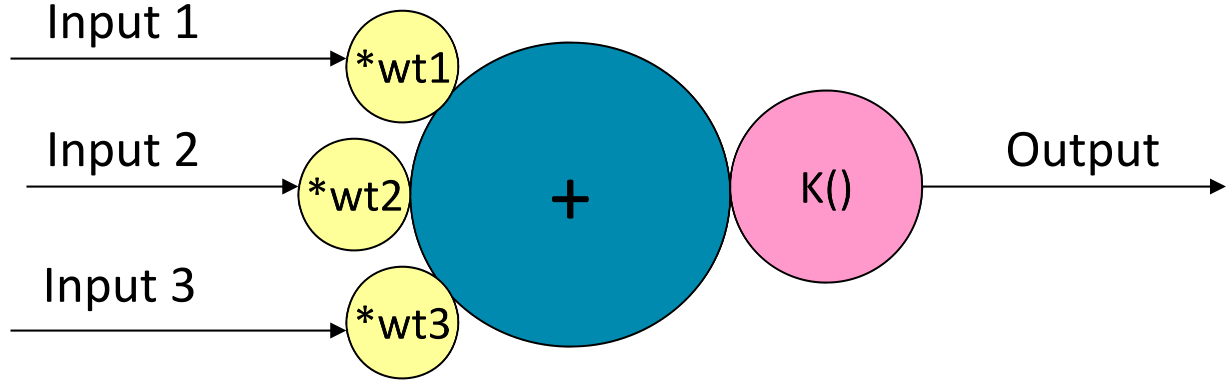




Neural Network

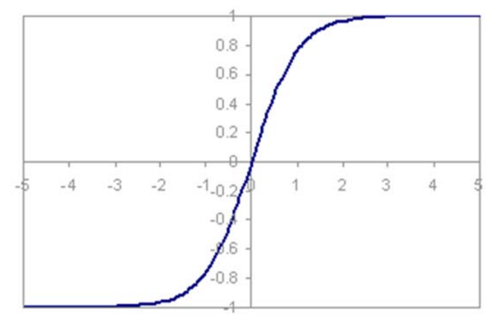


Operation of a Neuron



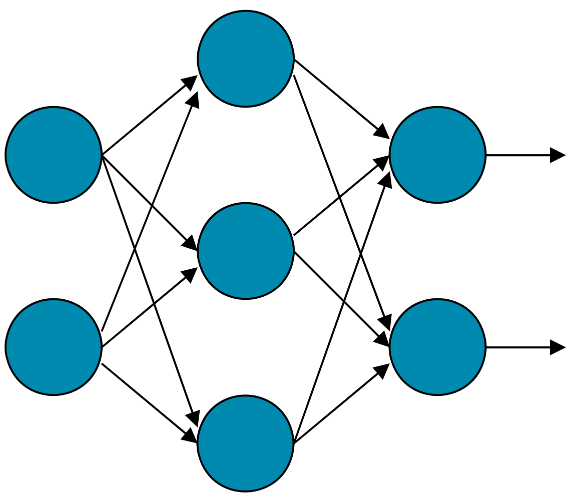
$$Output = K\left(\sum_i wt_i * Input_i\right)$$

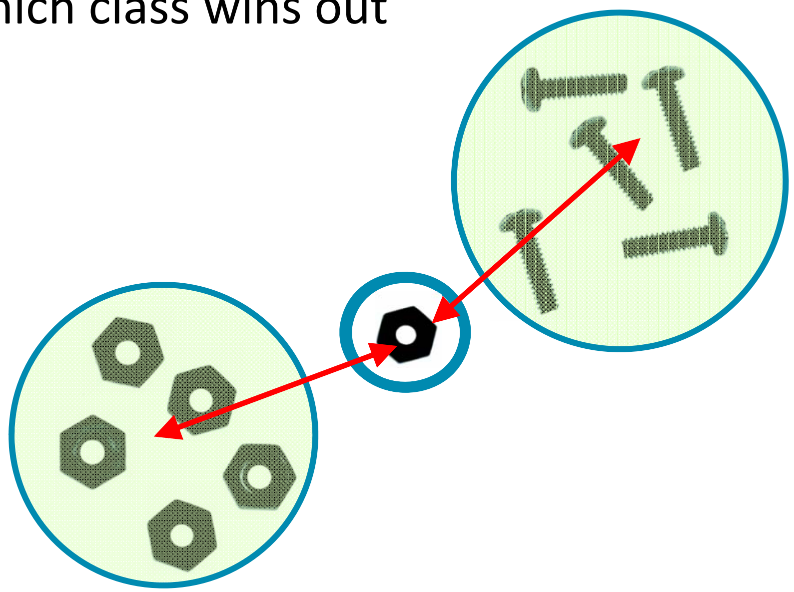
- K() is Activation Function
  - Clips output to a normalized range
  - **“Firing”** any one neuron will not overwhelm the network



Neural Networks as Statistical Classifiers

- Each neuron is its own linear statistical classifier, comparing the similarity of the inputs to some class and firing the output
- Practically, what happens is
  - Each hidden layer neuron measures similarity to a trained class
  - Each output layer neuron decides which class wins out



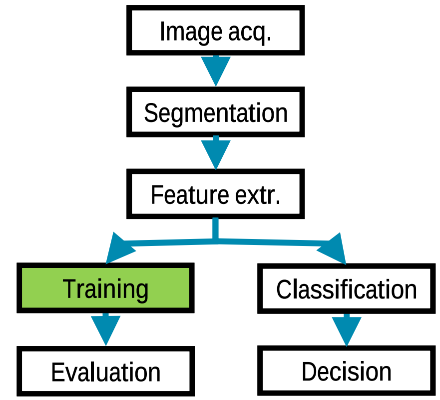


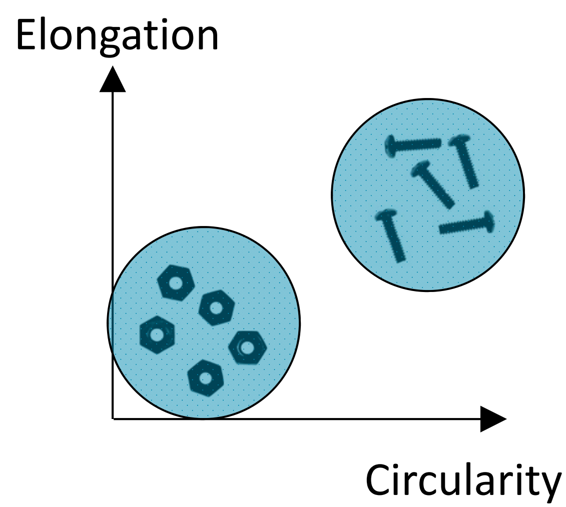
Training Neural Networks

- Trained iteratively
  - Each training example is presented to network one at a time
  - Neuron weights are adjusted to reinforce getting the correct result
  - Adjustment algorithm is “backpropogation of errors”
- Pros
  - Self-organizes into multiple versions of each class
  - Fast execution at run-time
- Cons
  - Results can be sensitive to the training set order
  - Large training set required to get convergence
  - Difficult to analyze operations (opaque)

Training - Classification Methods

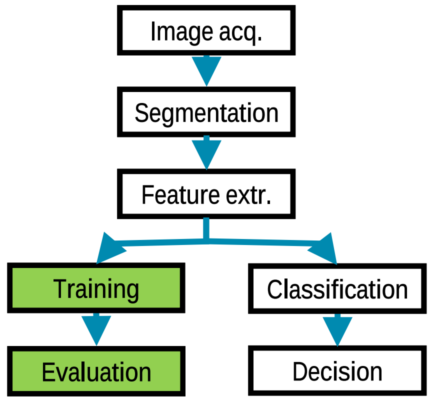
- Nearest Neighbor
  - Noise sensitive
- K-Nearest Neighbor
  - Noise filtering
  - Comparing to every trained sample is slow
- Linear Statistical Classifiers
  - Statistical data -> faster processing
  - Restricted to 1 cluster per class
- Neural Network
  - Statistical data -> faster processing
  - Multiple clusters per class
  - Sensitive to training process
  - Difficult to analyze
  - Does not output a match score





Training Process

- Divide samples into 2 data sets
  - Training data set
  - Testing data set
- Training data set
  - Learn statistics of each class
  - Learn boundary areas between classes
- Testing data set
  - Evaluate performance on unseen examples
  - New features needed?
  - New classifier algorithm needed?
  - DO NOT evaluate performance on training data set
    - Results will be too optimistic



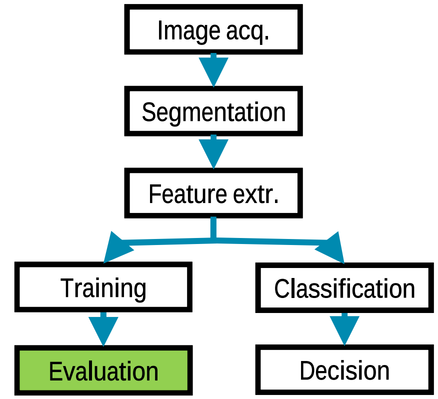
Evaluation Scoring

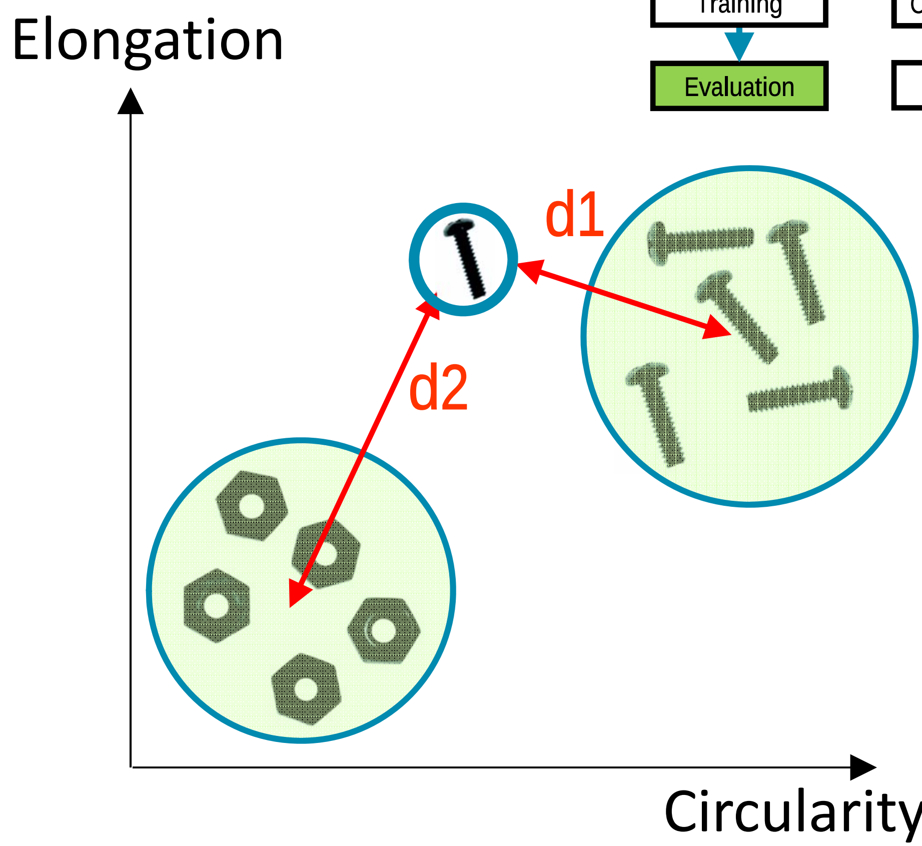
Reliability of classification

- Classification score (1..1000) =
  (1 – d1 / d2) x 1000

Similarity of sample

- Identification score (1..1000) =
  (1- d1) x 1000





Single Class – OK / NG

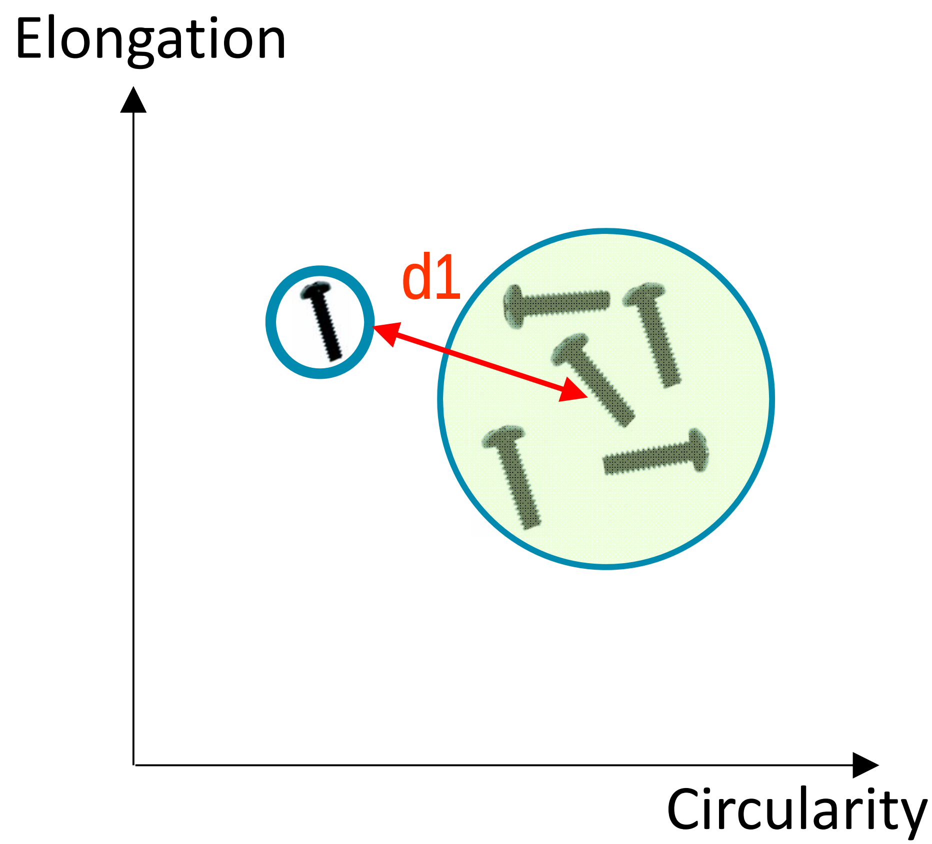
~~**Reliability of classification**~~

- ~~Classification score (1..1000) = (1 – d1 / d2) x 1000~~

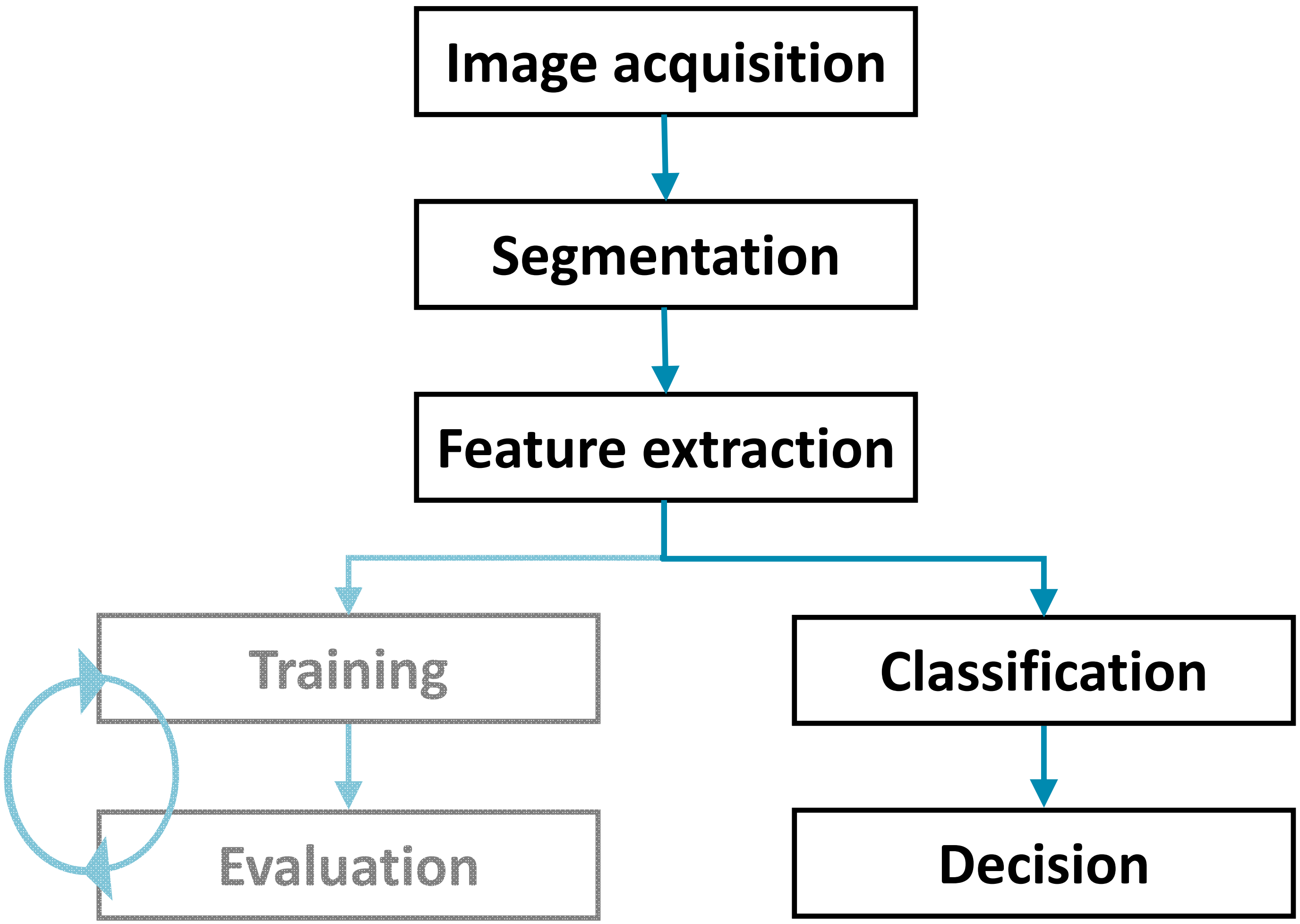
Similarity of sample

- Identification score (1..1000) = (1- d1) x 1000

OK if score > threshold



Classification Steps



Frequency Domain Processing

- Work directly with frequencies rather than pixels
- Main tool is FFT Fast Fourier Transform
- Application areas
  - Images with repeated dense patterns
  - Orientation tests
  - Spatial filtering
  - Image enhancement
  - Examples

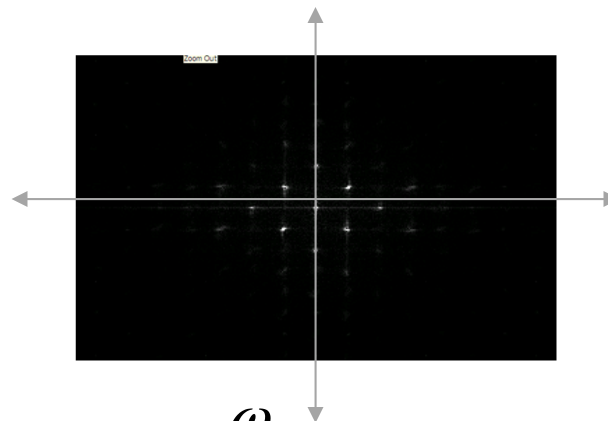
Domains & Transforms

y

x

Spatial Domain

FFT



ω_x

ω_y

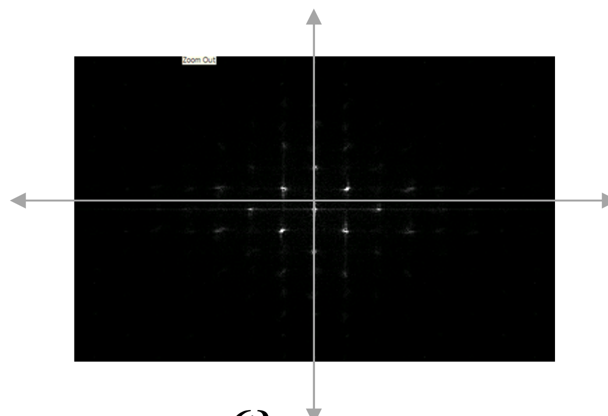
Frequency Domain

y

x

Spatial Domain

Inverse FFT

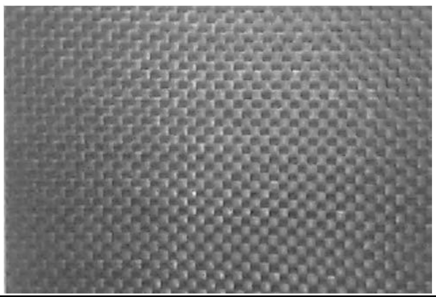
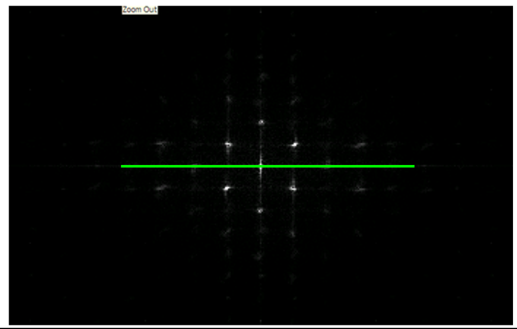
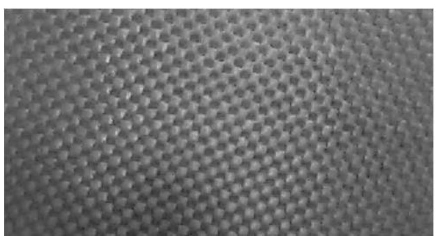
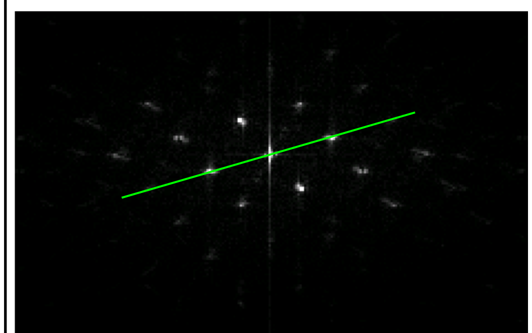


ω_x

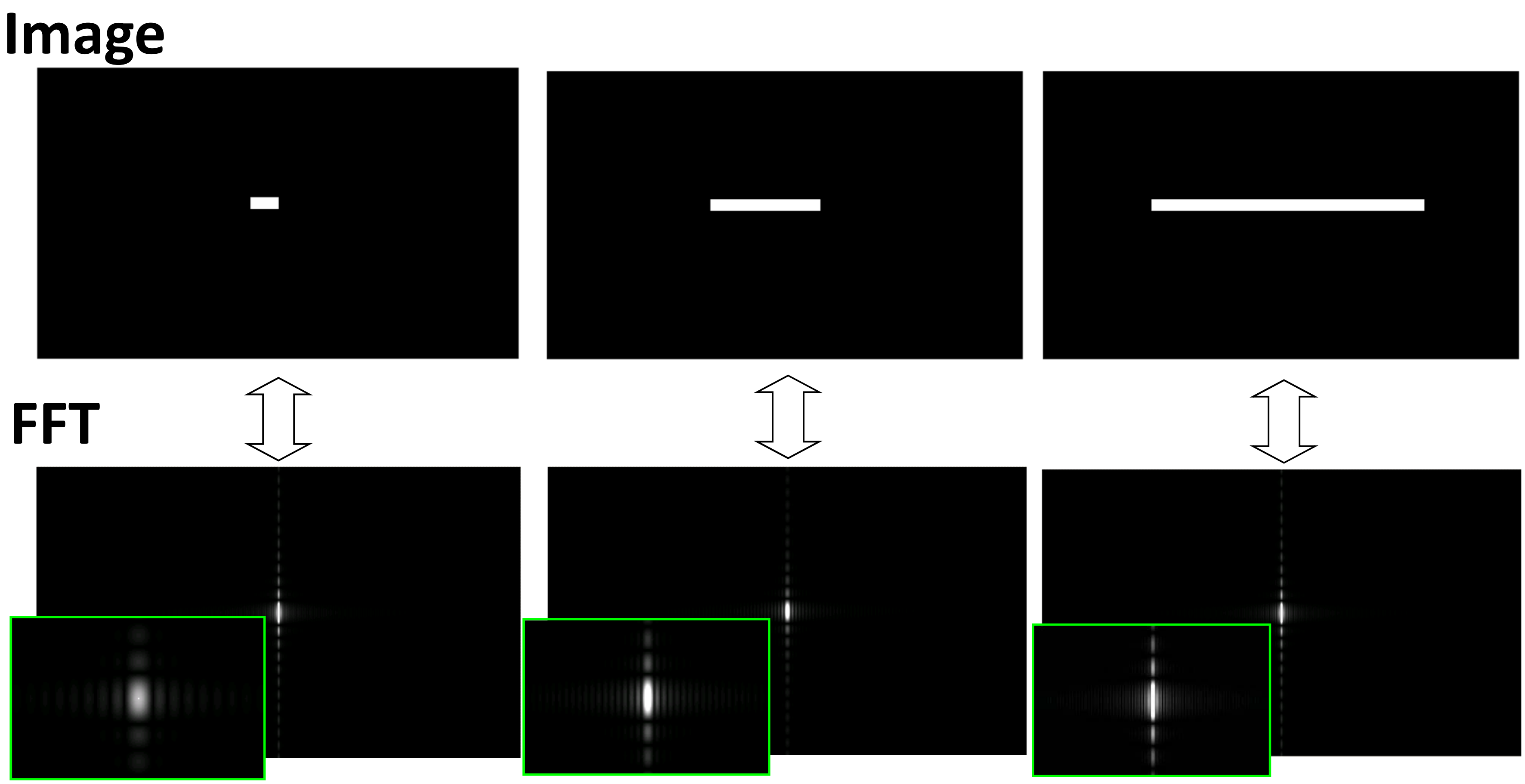
ω_y

Frequency Domain

Example - Fabric Inspection

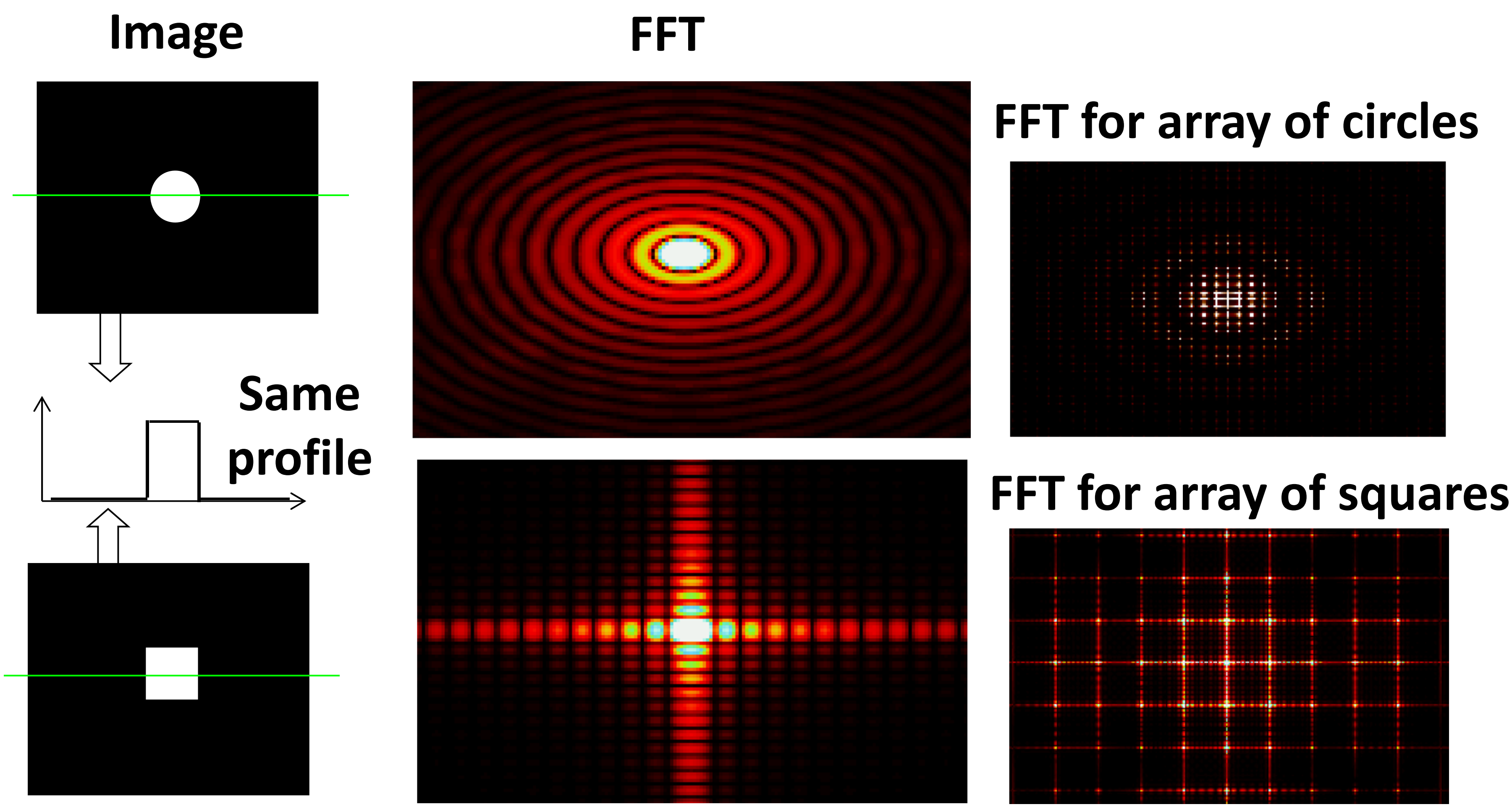
Case	Raw Image	FFT	Comments
Fabric with weave oriented horizontal			Orientation of the fiber and period of the weave is observed from the bright spots in each quadrant of FFT
Fabric with weave rotated to ≈30°			When weave is rotated, the spatial frequency is changed and FFT pattern is rotated

Inverse Size Relationships



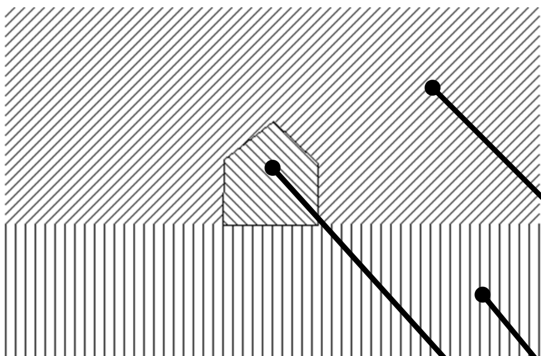
Small image features-> High Frequencies -> Away from center of FFT
Large image features -> Low Frequencies -> Center of FFT

Effects of Feature Shape on FFT

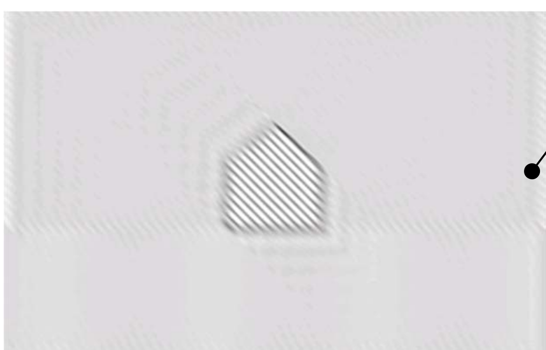


Spatial Filtering to Select Image Components

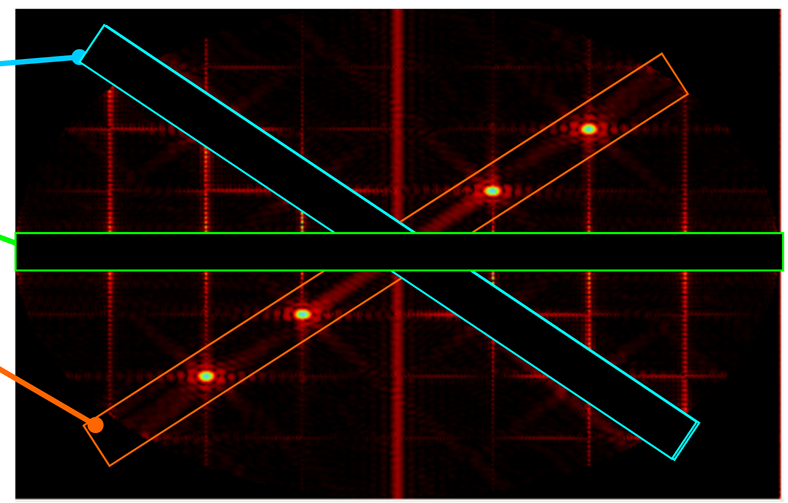
Encoded image



Filtered

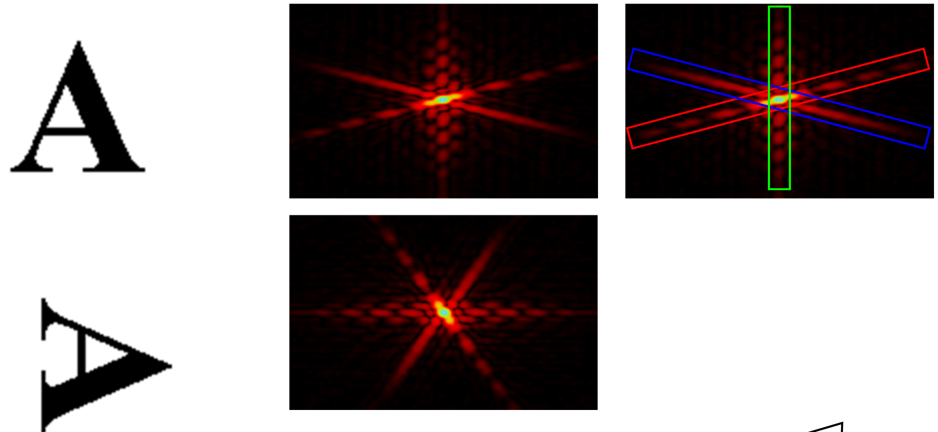


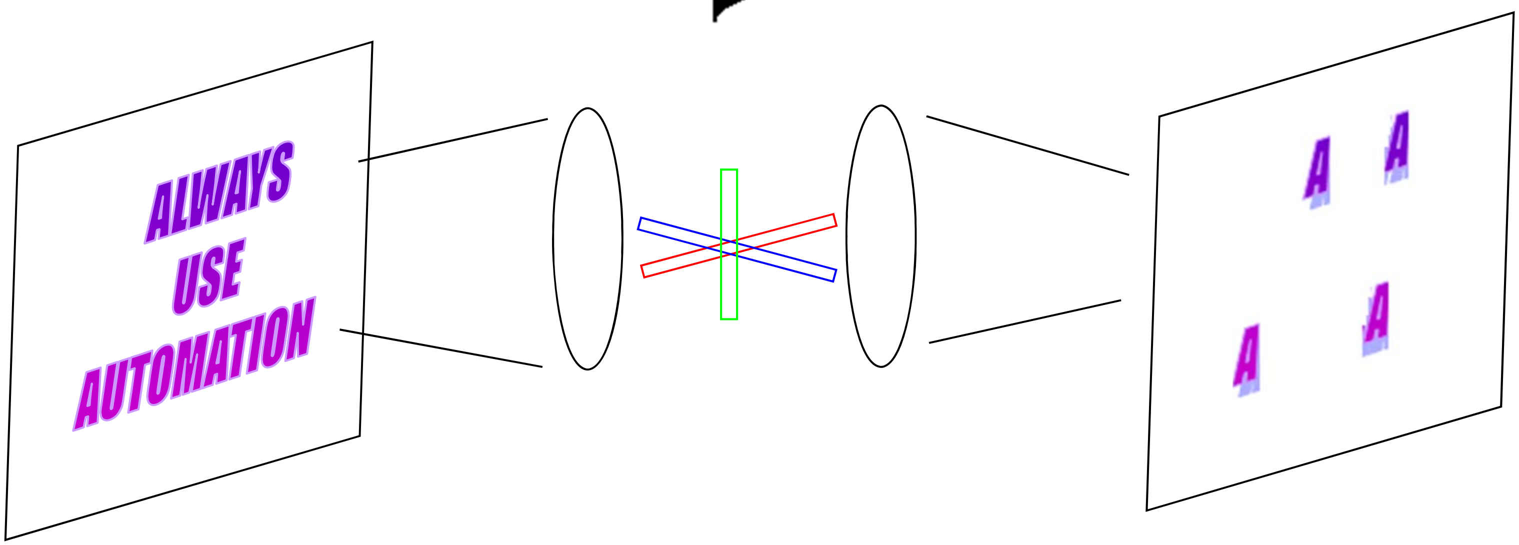
FFT with the filter masks overlaid that would be used to extract a separate part of the image.



Spatial Filtering for Character/Object Recognition

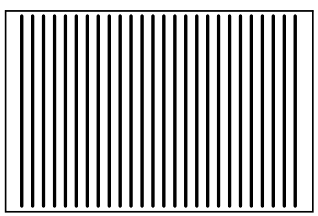
- FFT is invariant
  - Translation/position
  - Magnification/scale
  - NOT rotation



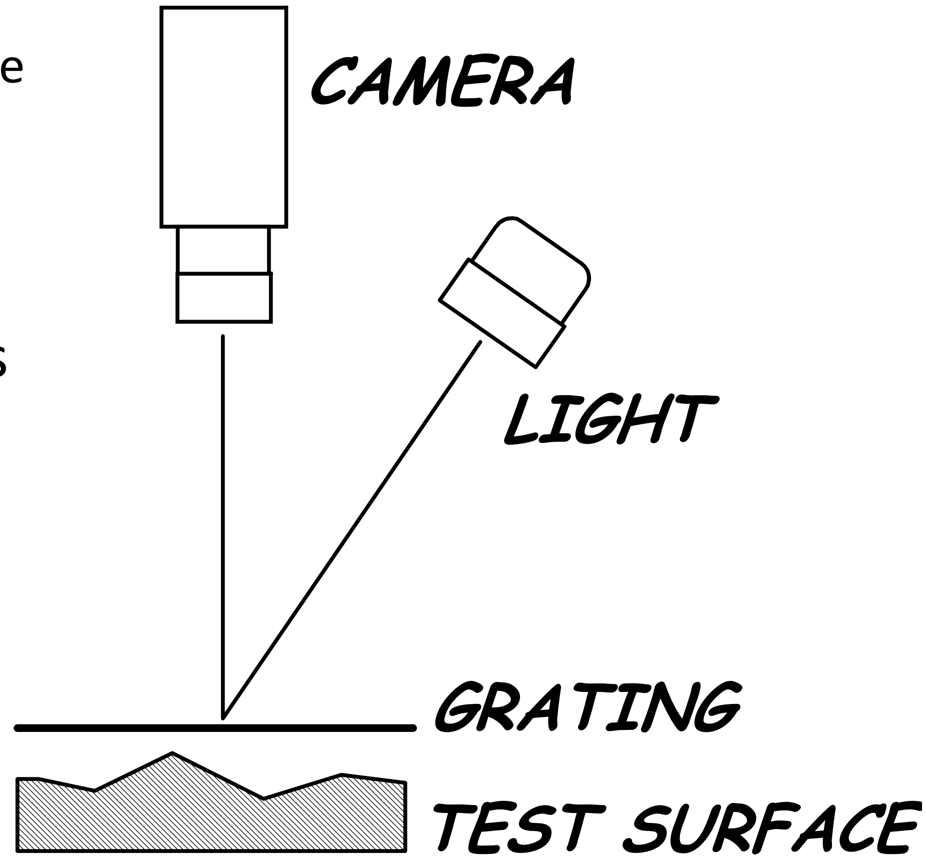


Shadow Moiré Contour Measurement

- Combination of concepts
  - Encoding of contour into image grayscale
  - Use of FFT to enhance the image
- Result is an image that gives the 3D surface shape

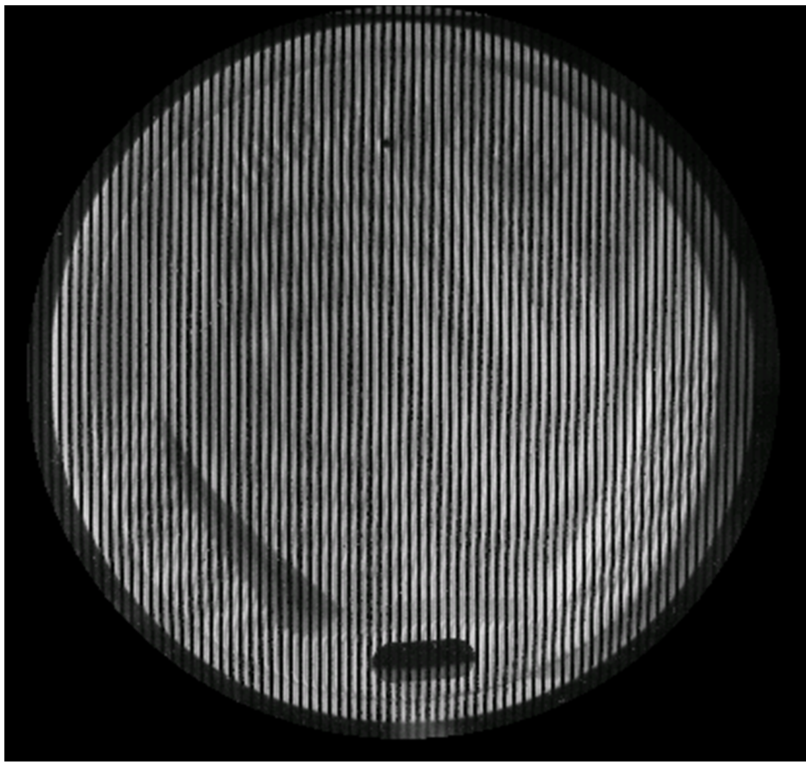


TOP VIEW OF LINE GRATING



Raw Test Image of Cup Lid

- Space of the contour lines is simple relationship



$$d = \frac{p}{\tan\alpha + \tan\beta}$$

d = distance between contour fringes

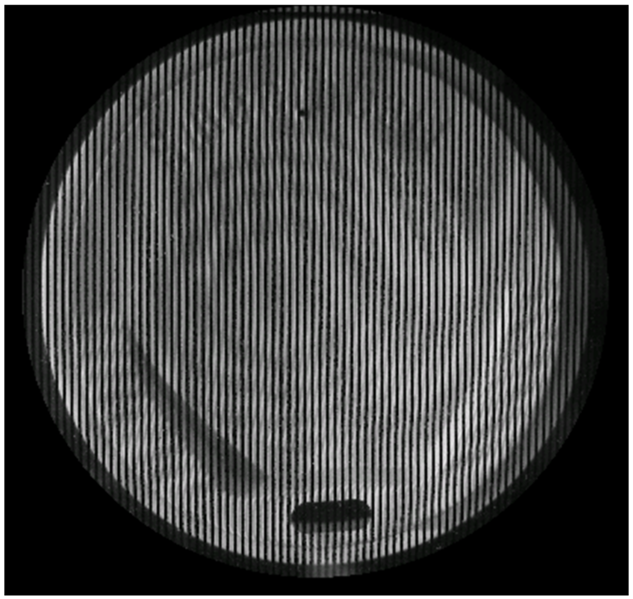
p = grating period

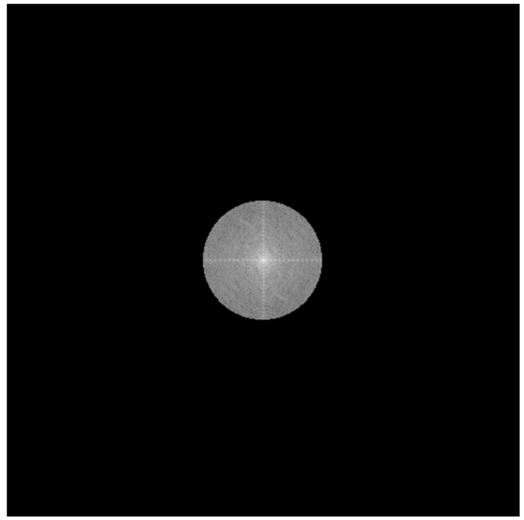
α = angle between light and grid normal

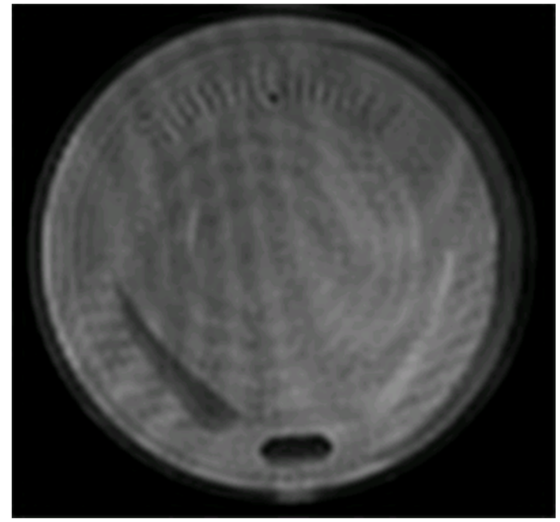
β = angle between camera and grid normal

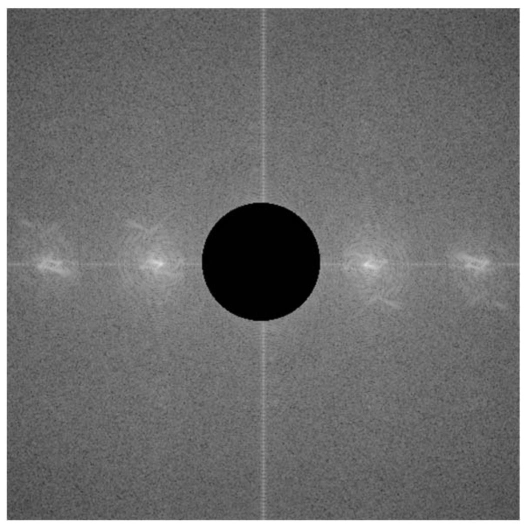
Low-pass and High-pass Filtering

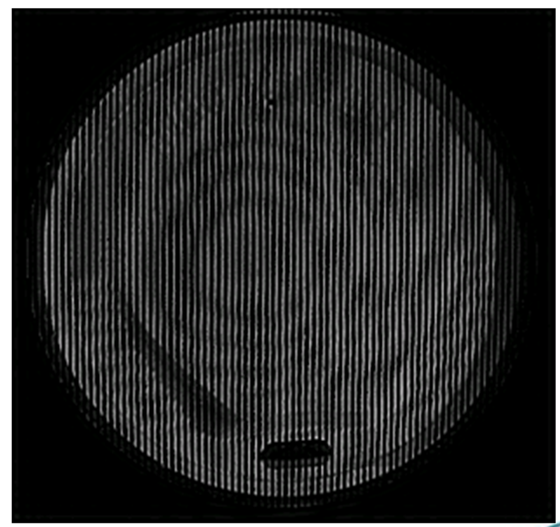
Low-pass Filtering







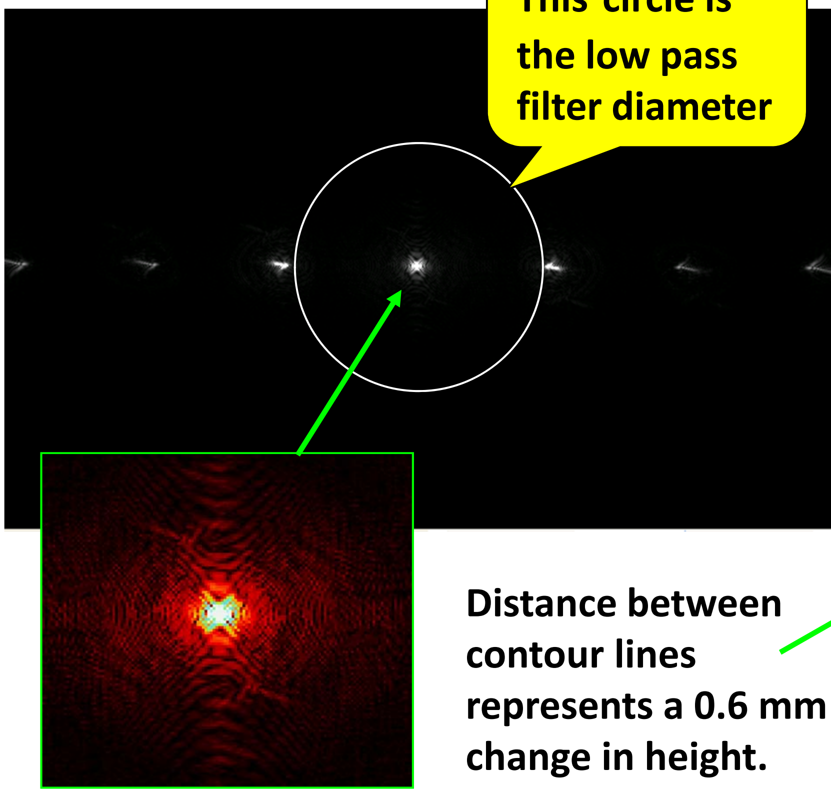




High-pass Filtering

Cup Lid with Contours Only

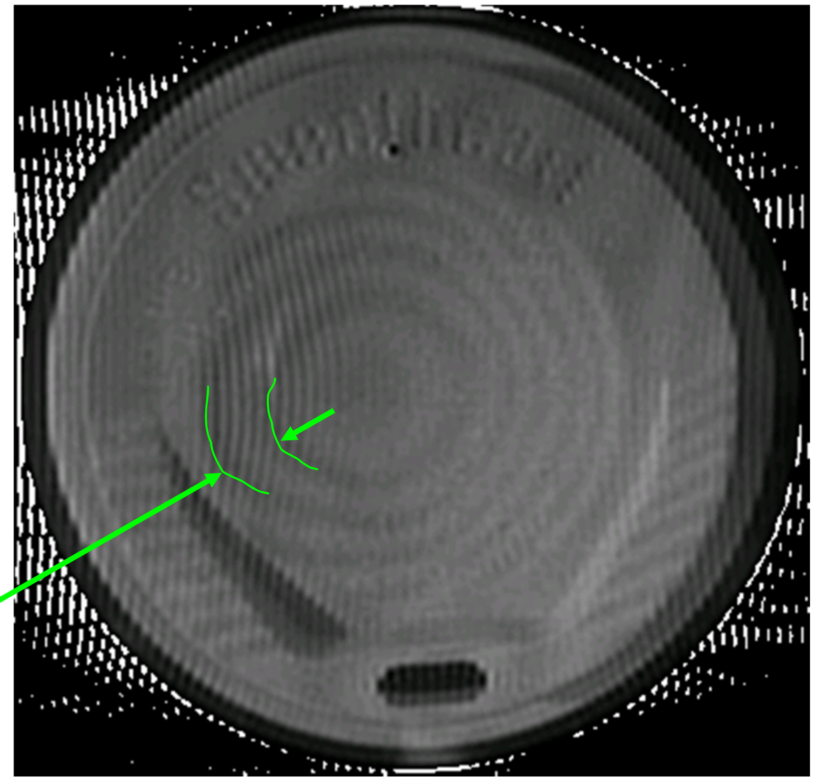
FFT



This circle is the low pass filter diameter

Distance between contour lines represents a 0.6 mm change in height.

Image of lid after removal of grating lines.



Applications

- Filtering
  - Standard low-pass/high-pass/band-pass filters
  - Non-standard filters
  - Very specific directions or frequencies
- Image encoding & compression
  - JPEG
- Image quality/inspection
  - Texture

Contact Information

Perry C. West
President

Automated Vision Systems, Inc.
4787 Calle de Lucia
San Jose, California 95124
U.S.A.

Phone: +1 408-267-1746
Email: perry@autovis.com

www.autovis.com