Soliton

Vision for a Better World

Computer Vision with Machine Learning

PSG Tech - 2018

Dhivakar

Computer Vision and
Machine Learning,
Soliton Technologies

Senthil

Computer Vision and
Machine Learning,
Soliton Technologies

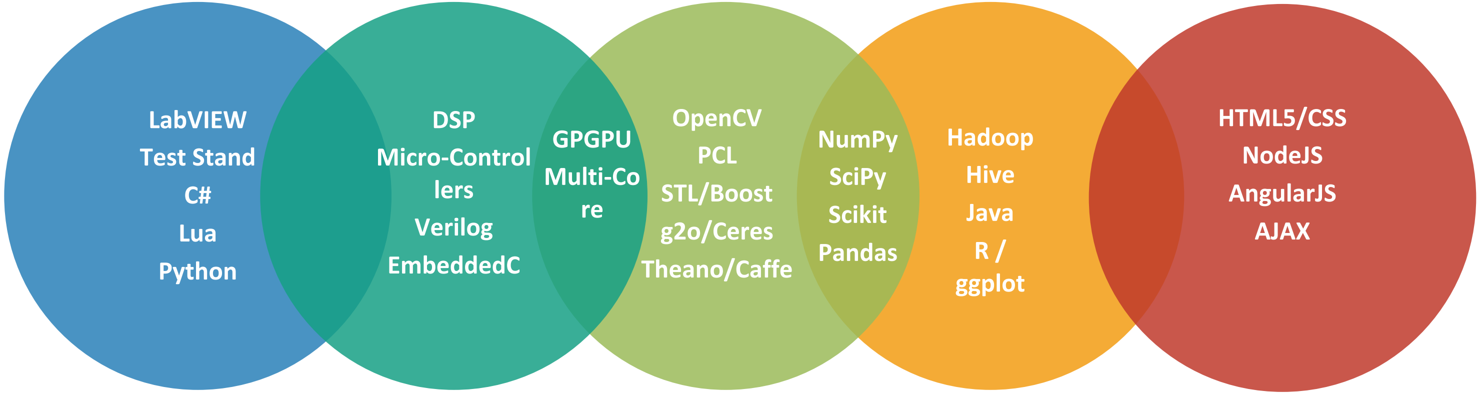
Measurement & Automation
Labview, Test Stand

Embedded Systems
FPGA, Embedded Systems, DSP, Device Driver

Computer Vision
Computer Vision, Machine Vision, Machine Learning

Data Analytics
Statistics, Machine Learning, Big Data, NLP, Text Analysis, Predictive Analytics

Web Technologies
Node.js, HTML, CSS, UI Design, C#, Java



Machine Vision Cameras

Spot It

Smart Vision Camera

DSP, Image Processing Library and Interactive GUI for programming.

Spot It V2

NEO
Smart Camera

Embedded DSP.
Fully Integrated Functionality.

NUERA

06'

07'-08

08-09

09-13'

2016 - 17

Soliton
Vision for a Better World

- ARM with 1.2 GHz processing power
- 8 GB Internal Flash memory and 1GB DDR3 RAM
- Powerful open source image processing library
- Option to interface 3 Image sensors (Parallel + MIPI)
- GIGE Ethernet, HDMI and Touch screen support
- GSM/4G support, WIFI/Bluetooth/Audio can be integrated
- RS232/CAN support to interface with PLCs
- Industrial standard with IP67 (Waterproof)
- Miniature design: 63.5 x 63.5 x 38 mm
- Interchangeable Lens
- **Deep Learning** Support

Optimization	Machine Learning	Features/ Dimn Reduction	3D Vision
MRF/DBN, Graph Cuts, Linear Prog, Convex Prog	Regression, SVM and Kernel Methods, Neural Networks, GMM, Bayesian Techniques	HOG, SIFT, SURF, LBP, PCA, Sparse-PCA, Isomap, Kernel-PCA, MDS	Structured Light, Time of Flight, Reconstruction, Structure from Motion

3D Reconstruction

Structured Lighting
Point Cloud Generation
Post-processing and Metrology

Object Recognition

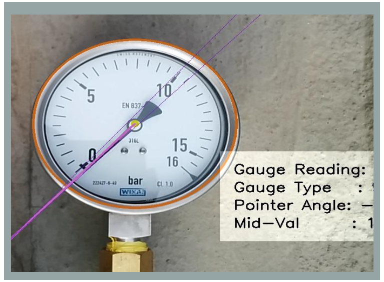
HOG, SIFT, SURF, LBP, FAST,
Boosting, SVM, ANN
PCA, Sparse-PCA, NCA, DBN

Object Tracking

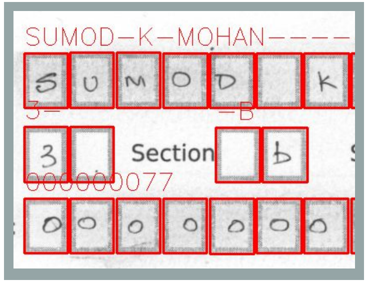
Region Based Tracking,
Kalman/Particle Filtering,
Markov Random Fields

Depth Estimation

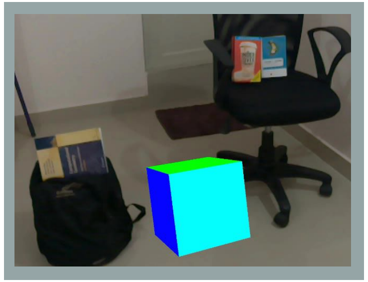
Dynamic Programming
Graph Cuts
Markov Random Fields



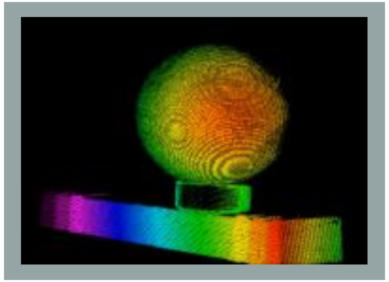
Project : Gauge Reader



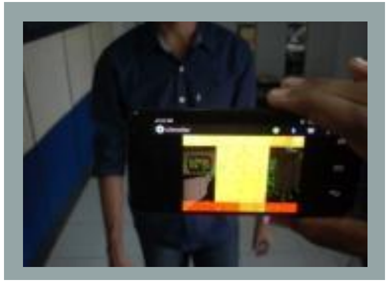
Project : HCR



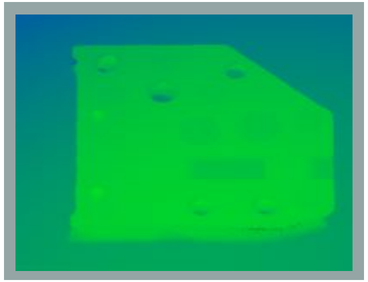
Project: Augmented Reality



Project : 3D Scanner



Project : Collision Detection

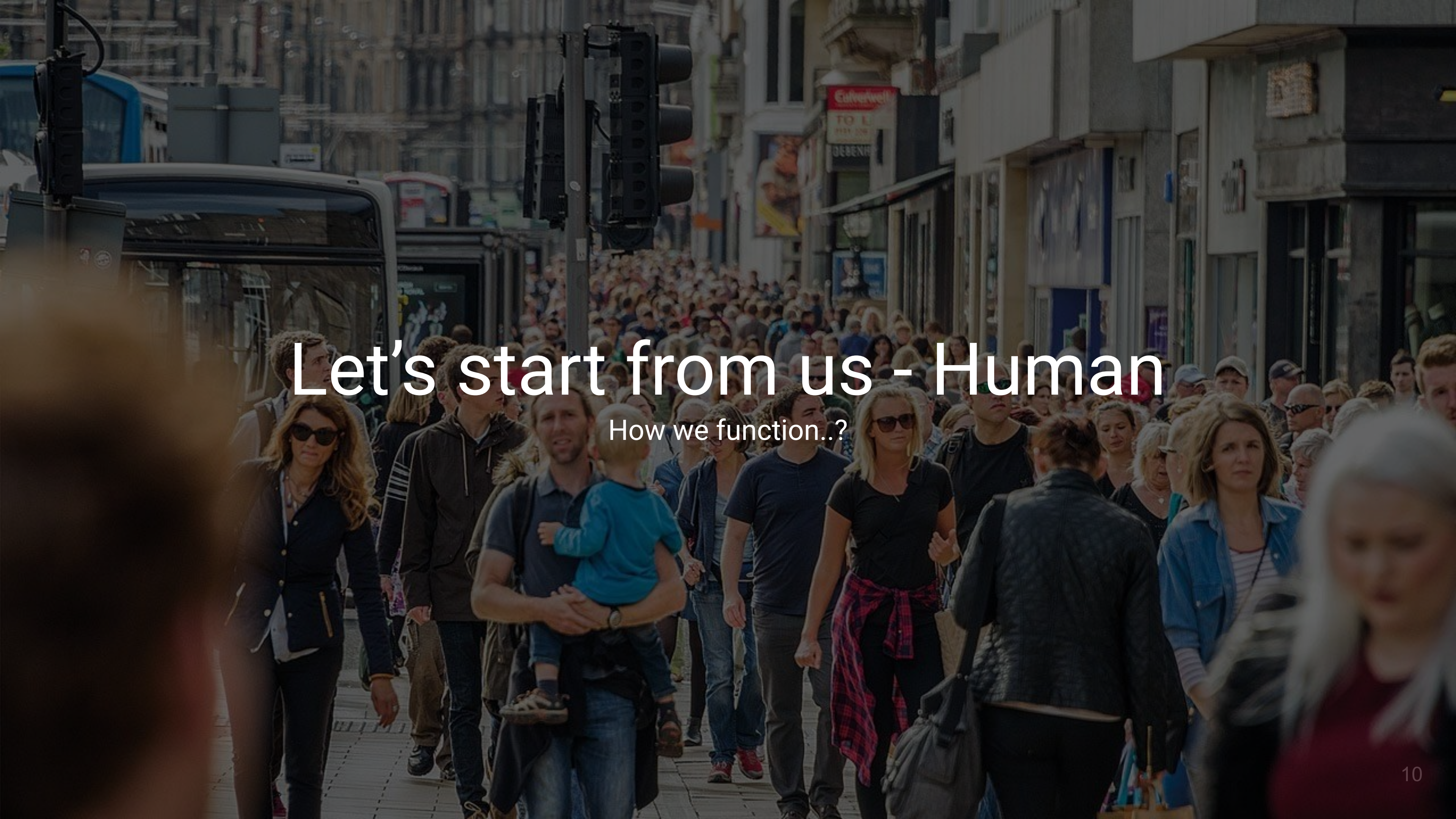


Product: 3D Pose Estimation

- We are in Machine Vision for the past 10+ years
- Deployed many multi-year projects in production
- First Smart camera from India
- Computer Vision
- 3D Vision
- Team of 12+ Engineers

Computer Vision & Machine Vision

Soliton Technologies

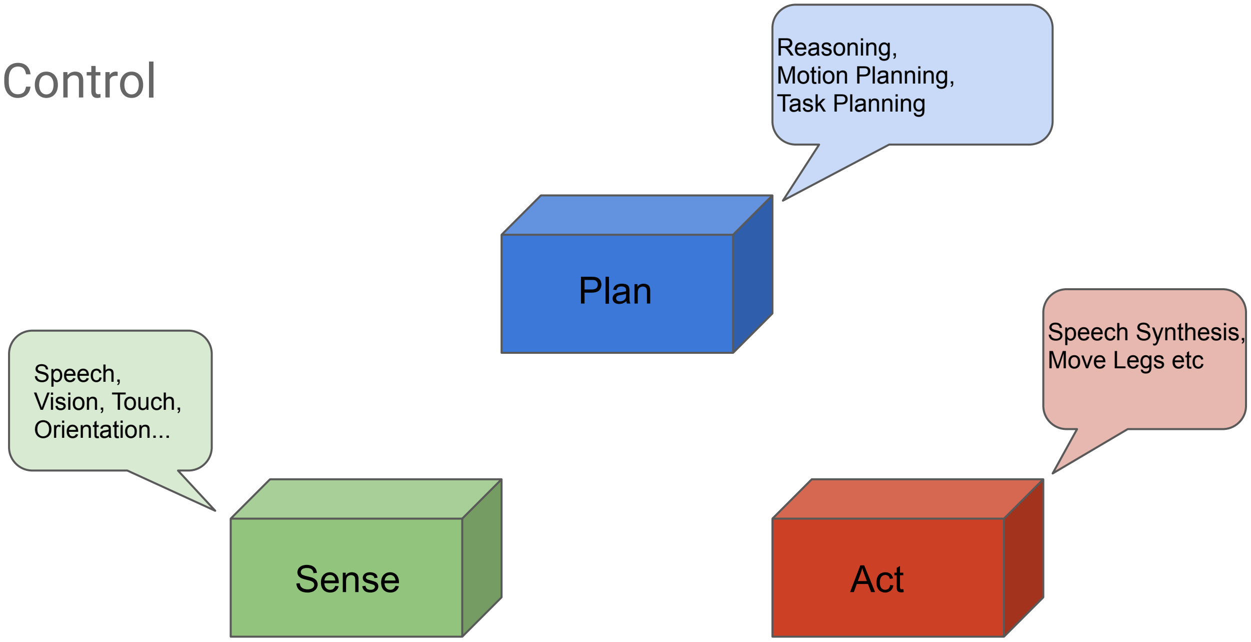


Let’s start from us - Human

How we function..?

Humans in the World

- Perceive -> Reason -> Control

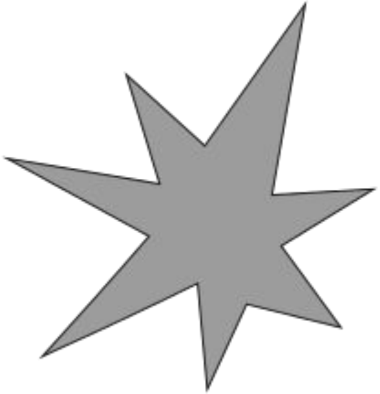


Human Vision

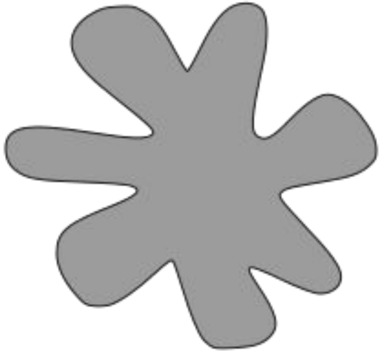
- Importance of vision for humans
  - 30% of cortex used in vision
  - McGurks Effect
- Decision makings
  - McGurk's Effect
  - Bouba / Kiki Experiment

Soliton
Vision for a Better World

Interconnections



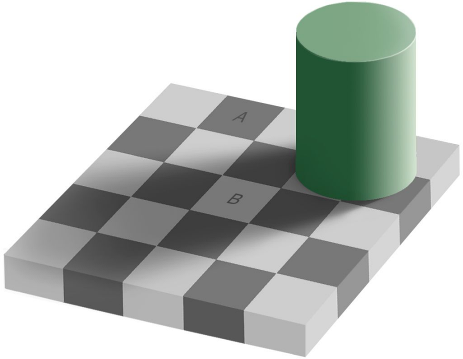
Kiki

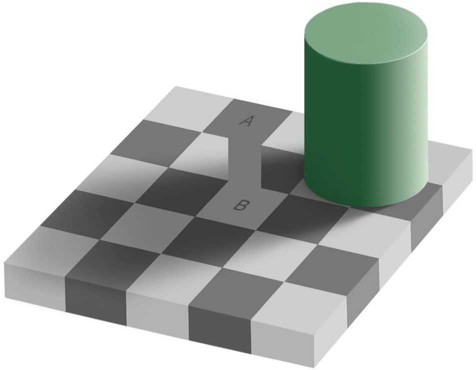


Bouba

Human Vision

- MultiSensory, High Resolution, Immersive Inner Movie
- Arguable one of the most Complex Machines
    - Color Stereo Pair: 768 x 576 @ 30 fps : 80 MB/s
    - 10^{11} Neurons
    - 10^{14}-10^{15} Synapses
    - Eons of evolution, learning right priors
    - Continuous flow of perception (with very few glitches)
- Perception
    - No eye inside an eye: just stream of electrical impulses :*Shades*
    - Brain combines them all to create our reality based on past experiences: *Sound Prediction & McGurks*
    - It is Brain's best guess of what is out there: *Ames Room*





Your Brain is Cheating you

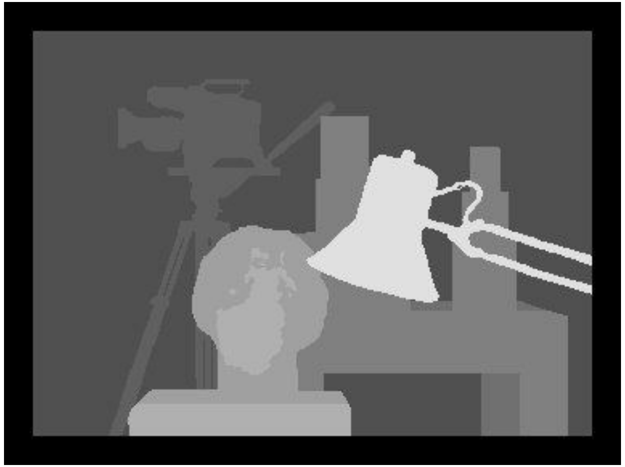
What is Computer Vision?

Ability of Computers to See and Understand from the Scene

- Saying if an object is present in the image
- Saying what are all the things present in the image
- Describe the scene

Computer Vision

- Image Processing, Computer Vision, Machine Vision
- Low Level: Edge Detection, Filtering..
- Mid Level: Segmentation..
- High Level: Object Detection, Stereo



Soliton
Vision for a Better World

Computer Vision in Games

Image Captioning

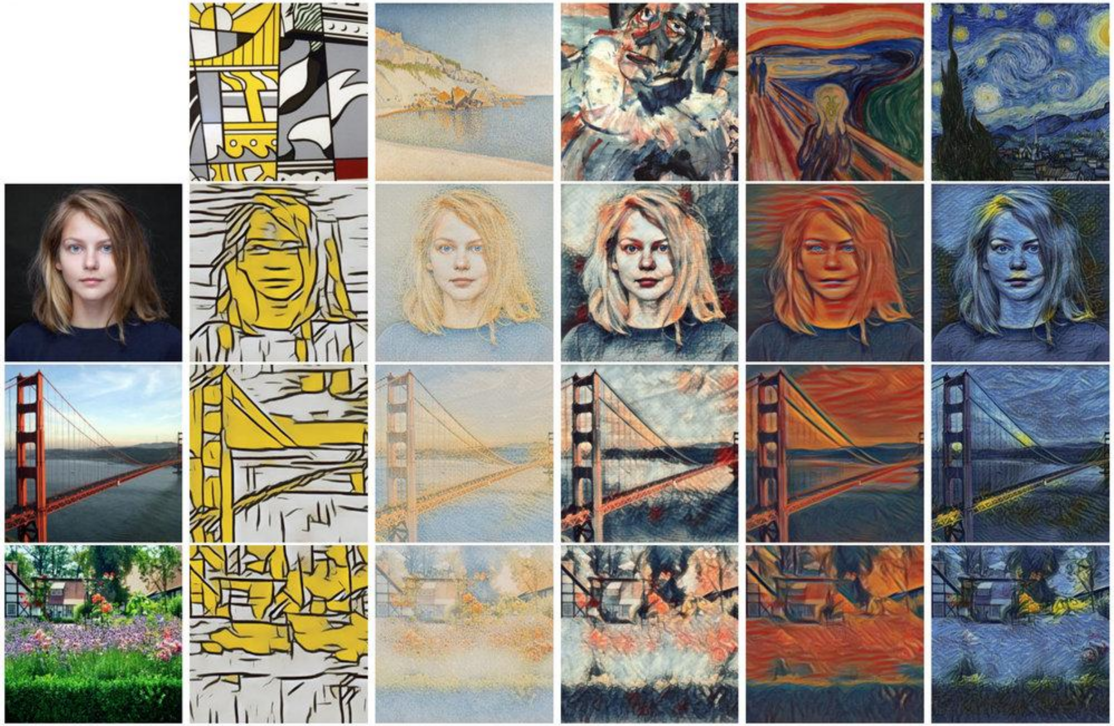
A person riding a motorcycle on a dirt road.

A group of young people playing a game of frisbee.

A herd of elephants walking across a dry grass field.

Deep Learning + Computer Vision in Prisma

Style transfer



Structure from Motion

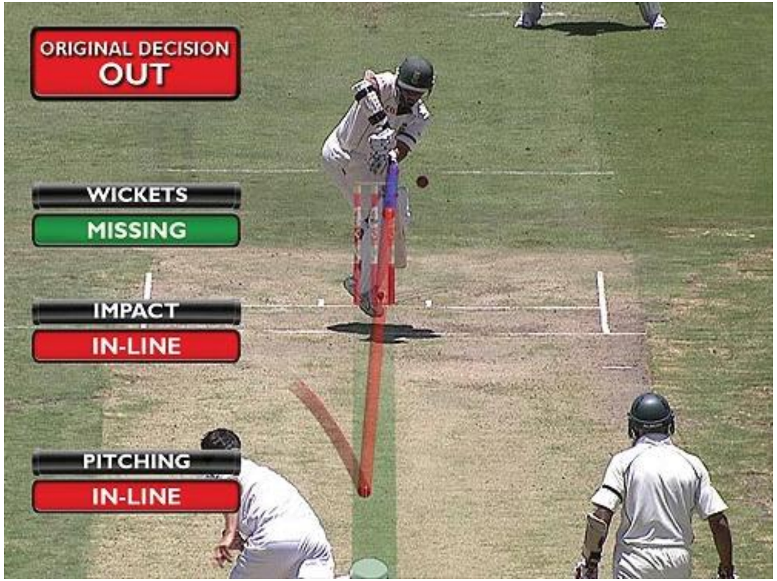
Seam Carving

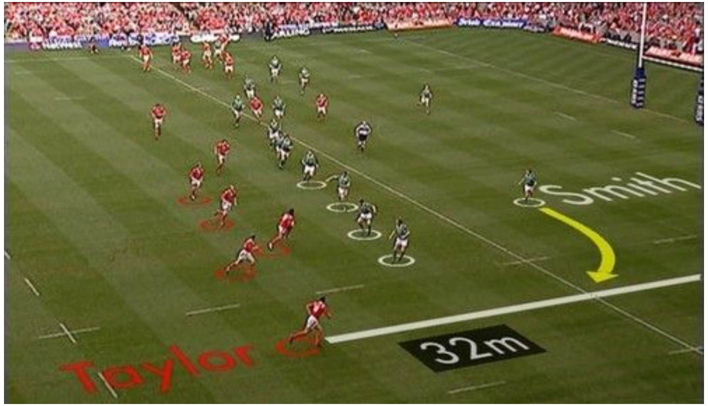
Soliton
Vision for a Better World

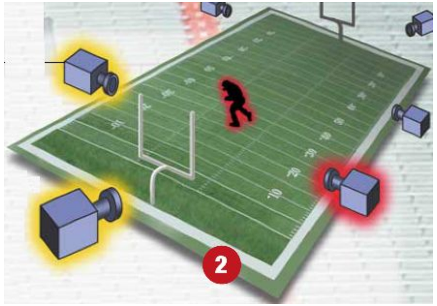
In Movies

- Bullet Time special-effect in The Matrix is an example of the application of SFM ideas.
- Linear array of cameras replaces moving camera.
- Green screen makes segmentation easy.

In International Games







Soliton
Vision for a Better World

Goals

Help building an understanding of How to solve real world problems using Computer Vision with ML and DL

Understand the Math and appreciate the beauty of it as opposed to simply Applying the technology

We can't teach you Vision in a 2 day Workshop. But we will try our best to Motivate you towards learning

Computer Vision

Digital image capture, storage and process.
Traditional Object detection and Limitations

Machine Learning

Evolution of ML. Understanding how Machines learn.
Limitations

Deep Learning

History and Biological motivation. Dive into Black Box (DL). Best practices

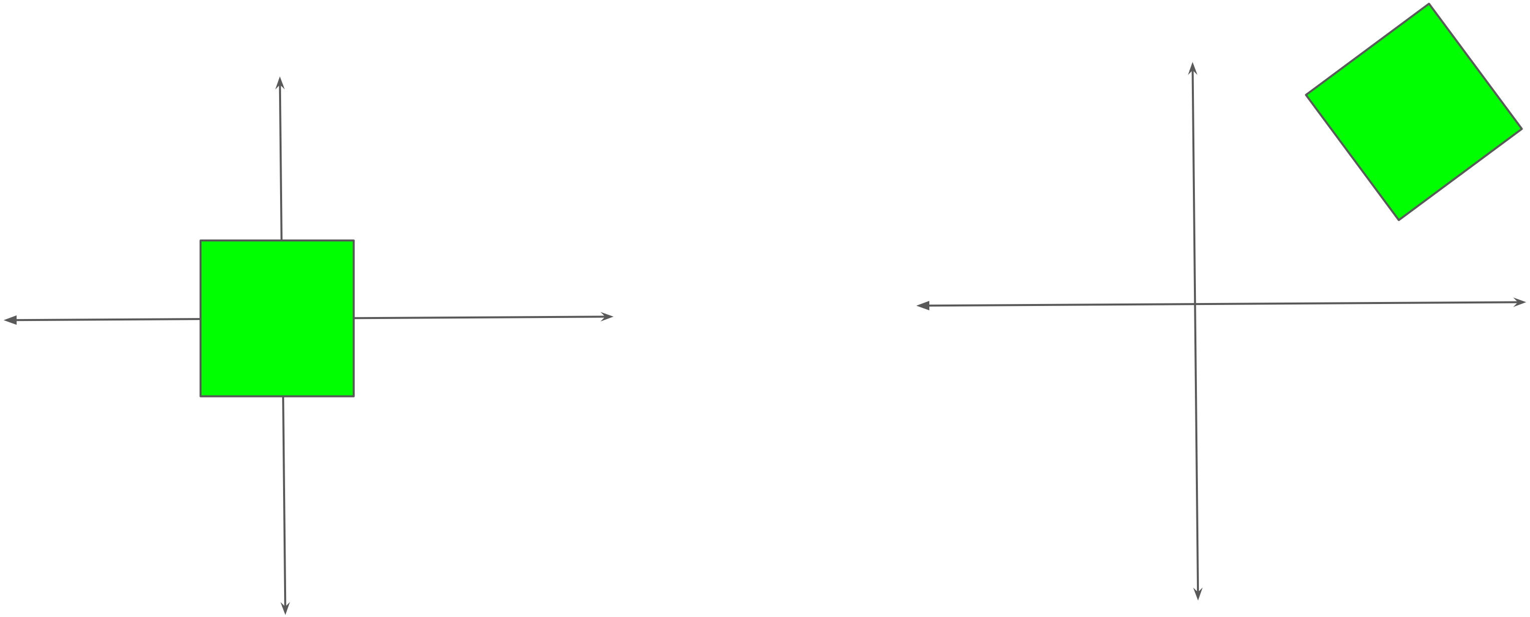
Soliton
Vision for a Better World

Euclidean Geometry?

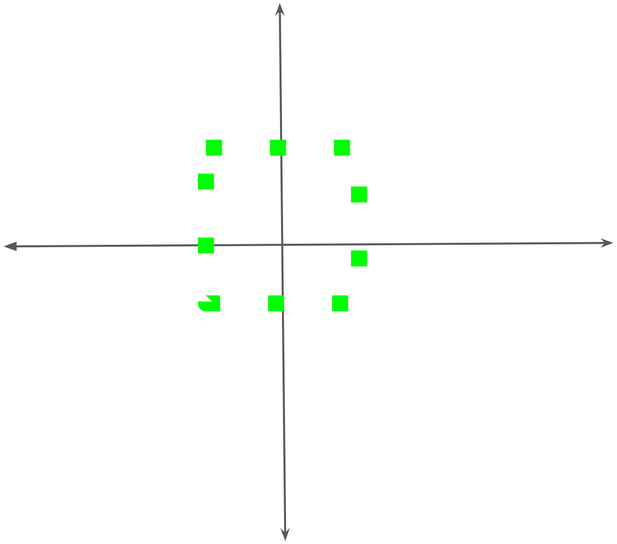
Projective Geometry

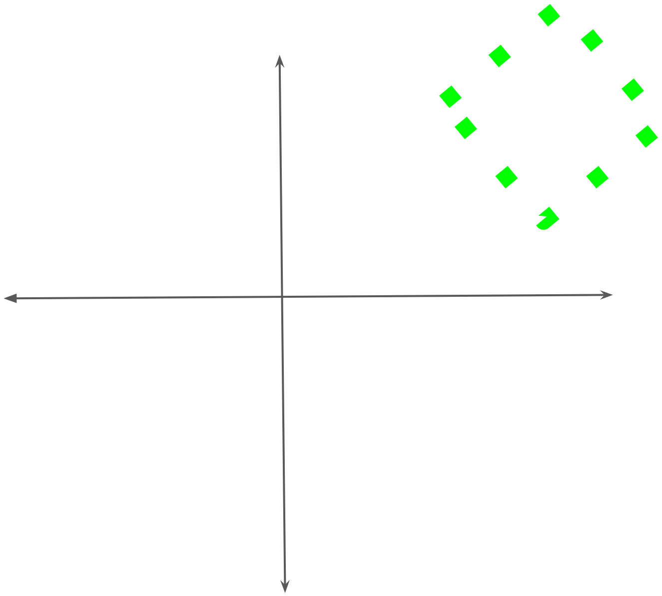
Soliton
Vision for a Better World

Motivation example - Euclidean Transform

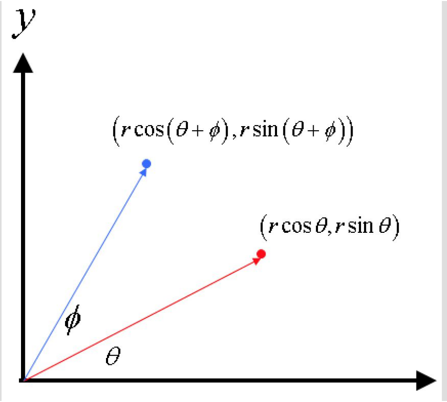


Shapes are Complicated. Focus on Points





Transformation Matrices



Let’s represent a point x,y in polar coordinates

$x = r\cos\phi$ $y = r\sin\phi$

Let θ be the rotation angle.Let x’ and y’ represent the rotated point.The rotated point can be expressed in polar coordinates as

$$x' = r\cos(\theta + \phi)$$
$$= r\cos\theta\cos\phi - r\sin\theta\sin\phi$$
$$= x\cos\theta - y\sin\theta$$

$$y' = r\sin(\theta + \phi)$$
$$= r\sin\theta\cos\phi + r\cos\theta\sin\phi$$
$$= x\sin\theta + y\cos\theta$$

Soliton
Vision for a Better World

Can we represent the same in Matrix?

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{bmatrix}$$

Lets see how we can represent translation

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} tx \\ ty \end{bmatrix}$$

But can we represent both translation and rotation in a single compact matrix, rather than splitting into a matrix multiplication and vector translation?

Soliton
Vision for a Better World

Solution - Homogeneous Coordinates

Lets represent each point by appending a one

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} tx \\ ty \end{bmatrix} = \begin{bmatrix} x\cos\theta - y\sin\theta + tx \\ x\sin\theta + y\cos\theta + ty \end{bmatrix}$$

Soliton
Vision for a Better World

Solution - Homogeneous Coordinates

$$\begin{bmatrix} \cos\theta & -\sin\theta & tx \\ \sin\theta & \cos\theta & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x\cos\theta - y\sin\theta + tx \\ x\sin\theta + y\cos\theta + ty \\ 1 \end{bmatrix}$$

Class I: Euclidean

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \varepsilon \cos\theta & -\sin\theta & t_x \\ \varepsilon \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

(*iso*=same, *metric*=measure)

$\varepsilon = \pm 1$

orientation preserving: $\varepsilon = 1$

orientation reversing: $\varepsilon = -1$

$$\mathrm{x}' = \mathbf{H}_E\, \mathrm{x} = \begin{bmatrix} \mathbf{R} & \mathrm{t} \\ 0^\mathsf{T} & 1 \end{bmatrix} \mathrm{x} \qquad \mathbf{R}^\mathsf{T}\mathbf{R} = \mathbf{I}$$

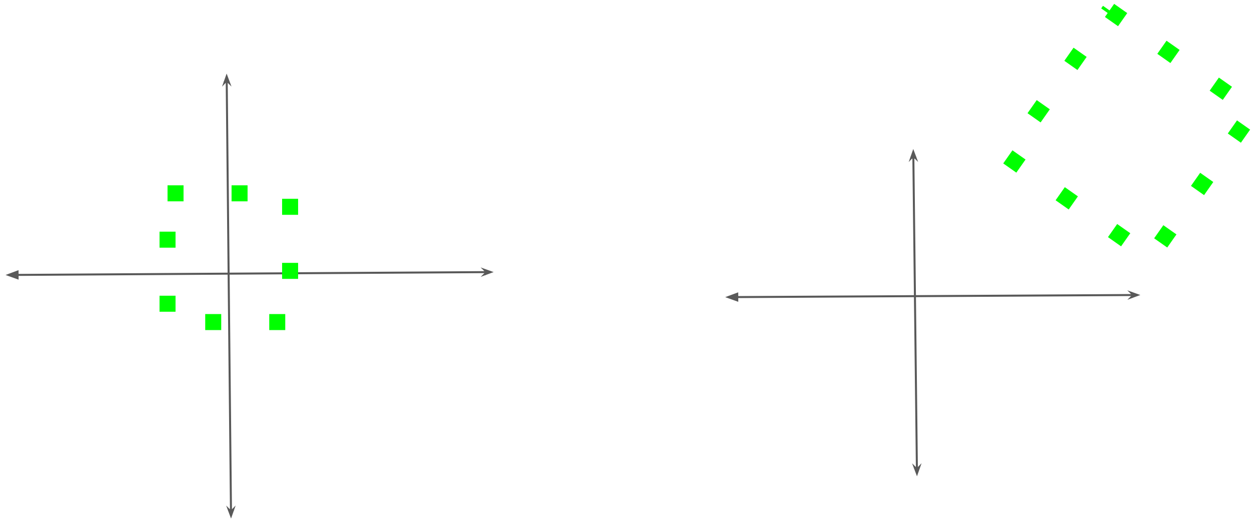
3DOF (1 rotation, 2 translation)

special cases: pure rotation, pure translation

Invariants: length, angle, area

Soliton
Vision for a Better World

Ability to Scale up or Scale down



$$\begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} s_x x \\ s_y y \\ 1 \end{pmatrix}.$$

Soliton
Vision for a Better World

Ability to Scale up / Scale down

$$\begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} sx \\ sy \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \cos\theta & -\sin\theta & tx \\ \sin\theta & \cos\theta & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & tx \\ s\sin\theta & s\cos\theta & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Class II: Similarities

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

(*isometry* + *scale*)

also known as *equi-form* (shape preserving)

$$\mathrm{x}' = \mathbf{H}_S\,\mathrm{x} = \begin{bmatrix} s\mathbf{R} & \mathrm{t} \\ 0^\mathsf{T} & 1 \end{bmatrix} \mathrm{x}$$

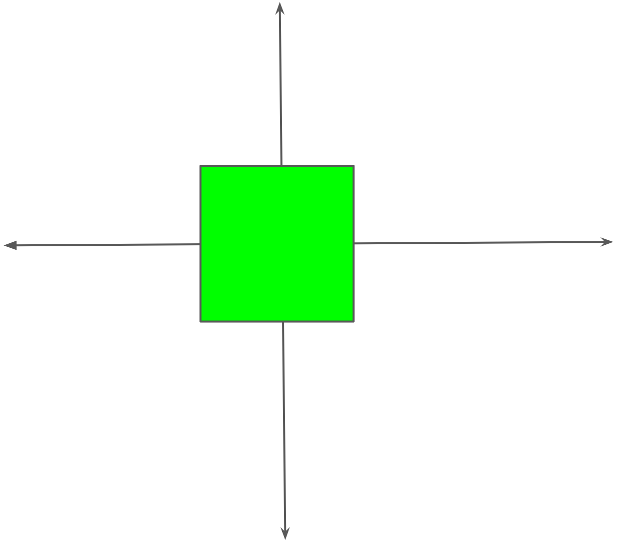
$$\mathbf{R}^\mathsf{T}\mathbf{R} = \mathbf{I}$$

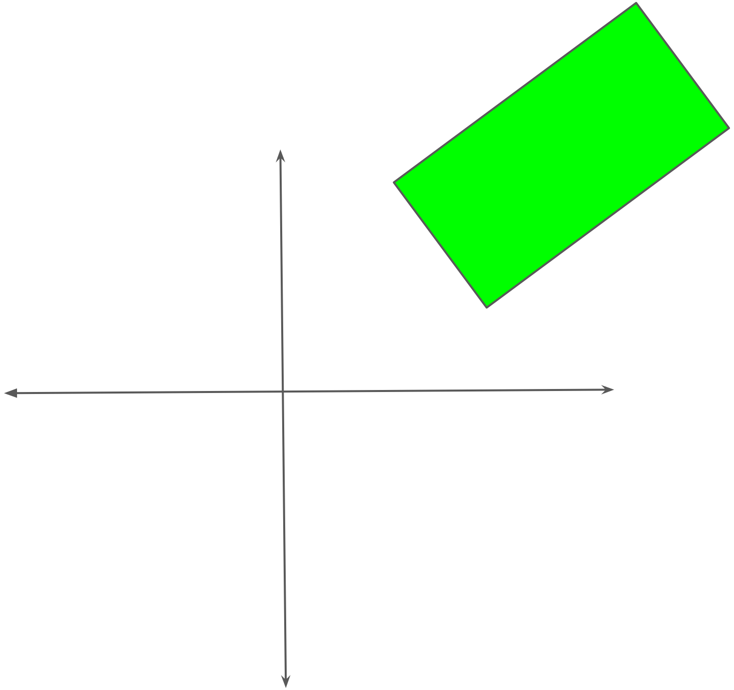
4DOF (1 scale, 1 rotation, 2 translation)

metric structure = structure up to similarity (in literature)

Invariants: ratios of length, angle, ratios of areas, parallel lines

For Non-Uniform scaling?



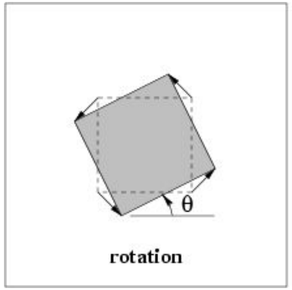


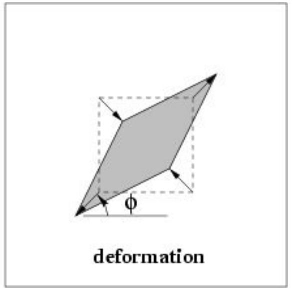
$$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \cos\theta & -\sin\theta & tx \\ \sin\theta & \cos\theta & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x\cos\theta & -s_y\sin\theta & tx \\ s_x\sin\theta & s_y\cos\theta & ty \\ 0 & 0 & 1 \end{bmatrix}$$

Class III: Affine transformations

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$





$$\mathbf{x}' = \mathbf{H}_A\,\mathbf{x} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \mathbf{x}$$

$$\mathbf{A} = \mathbf{R}(\theta)\mathbf{R}(-\phi)\mathbf{D}\mathbf{R}(\phi) \quad \mathbf{D} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

6DOF (2 scale, 2 rotation, 2 translation)

non-isotropic scaling! (2DOF: scale ratio and orientation)

Invariants: parallel lines, ratios of parallel lengths, ratios of areas

General Affine Transform

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ rotation

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$ translation

$$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ scaling

$$\begin{bmatrix} 1 & s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ skew

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}$$ general affine transform

Projective Transform

All transforms learnt till now are planar transform

What happens when we try to map between two different planes

Why is this necessary? Why is it necessary to learn a transform that maps between planes?

Optical illusions - Inferring 3D world from 2D planes

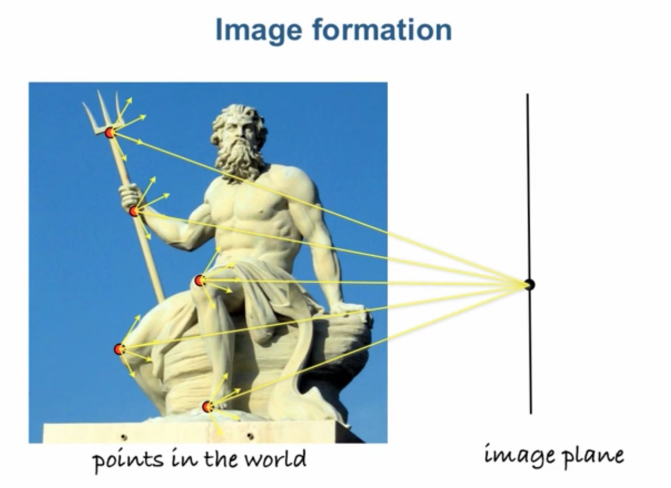


Image formation
points in the world
image plane

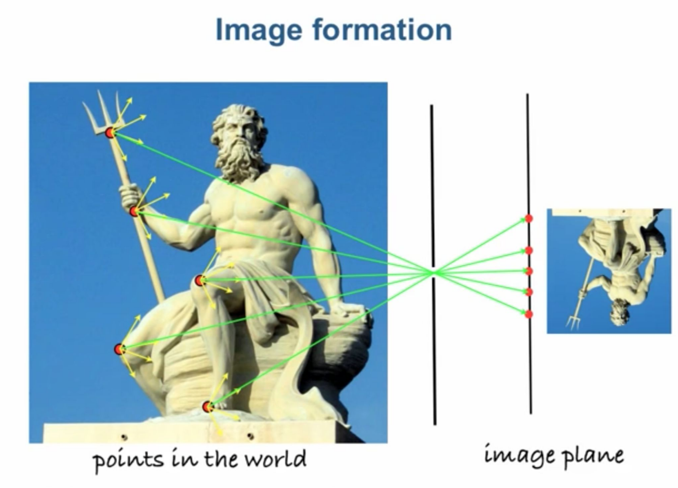
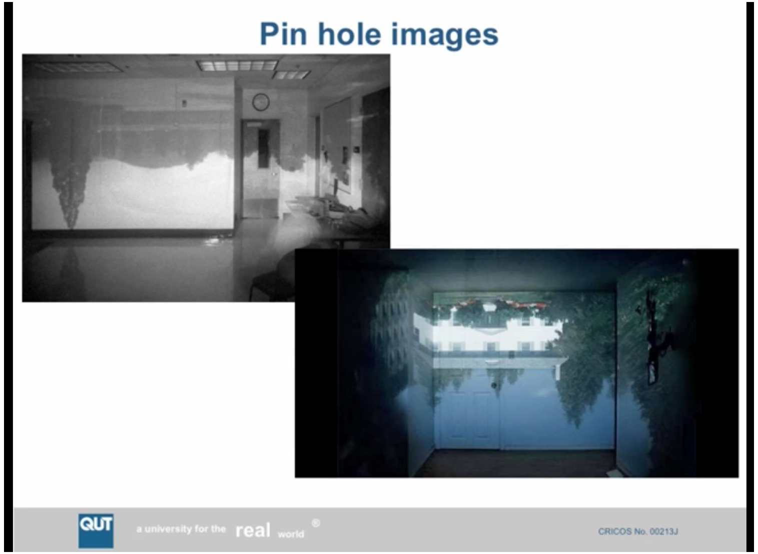


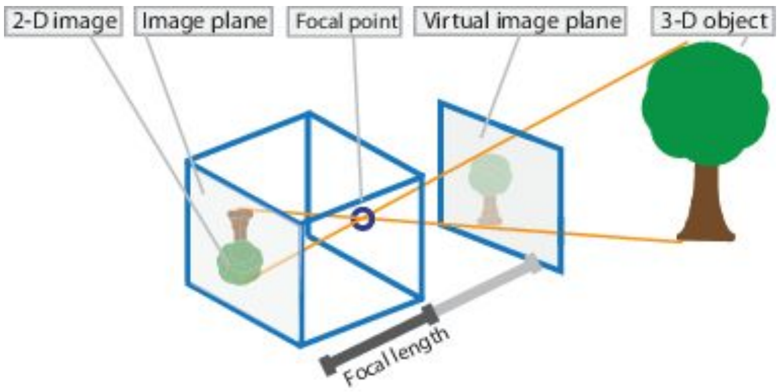
Image formation
points in the world
image plane

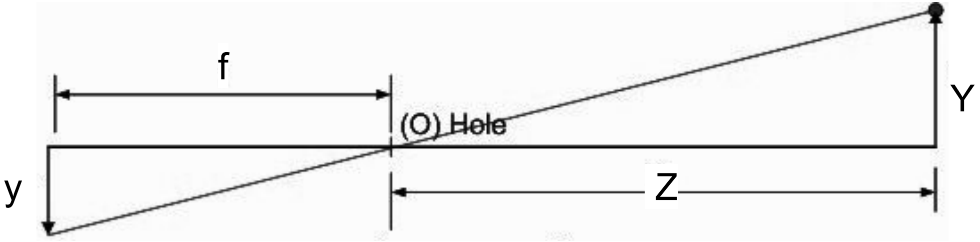

Pin hole images
QUT
a university for the real world
CRICOS No. 00213J


The world's largest pin hole camera
http://www.legacyphotoproject.com
QUT
a university for the real world
CRICOS No. 00213J

Pinhole Camera Model

- Simplest model of imaging process





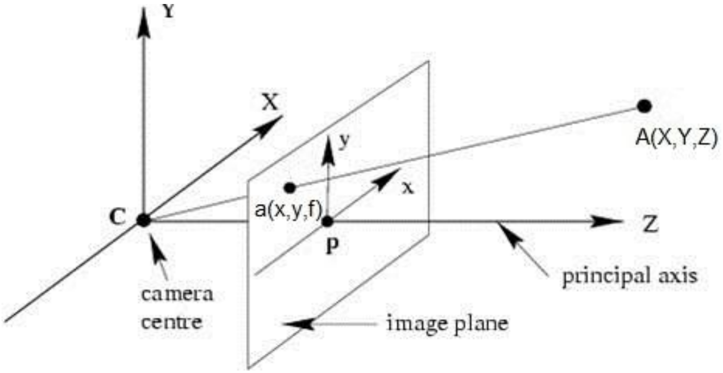
$$\frac{y}{f} = \frac{Y}{Z} \longrightarrow y = \frac{fY}{Z}$$

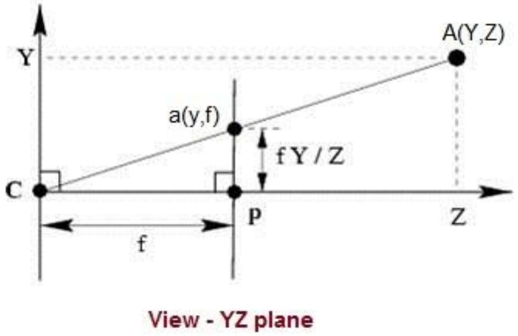
$$\frac{x}{f} = \frac{X}{Z} \longrightarrow x = \frac{fX}{Z}$$

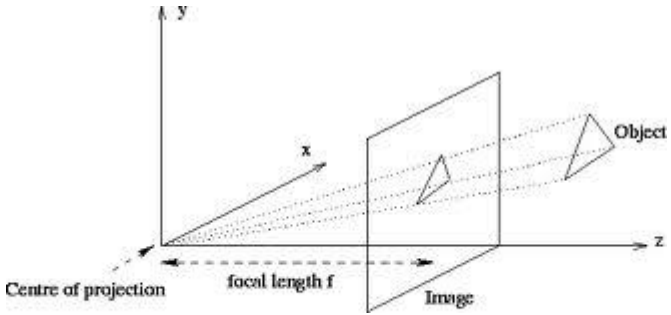
Ref : "A Flexible New Technique for Camera Calibration", Zhengyou Zhang, Technical Report, August 13, 2008

Soliton
Vision for a Better World

Pinhole Camera Model- Another Representation (Use of virtual planes)

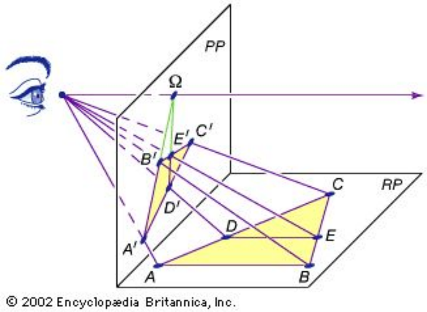


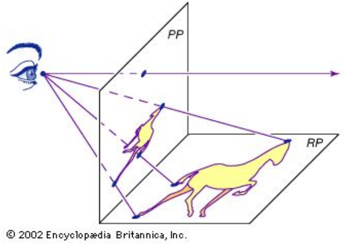


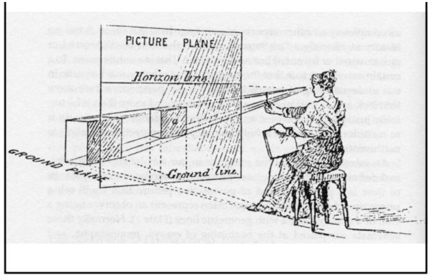


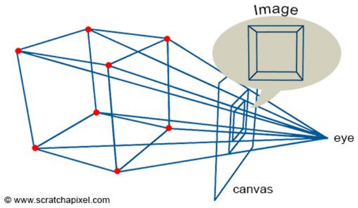
Soliton
Vision for a Better World

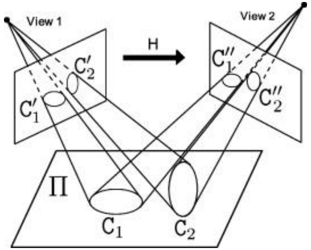
How human eye perceives the world?





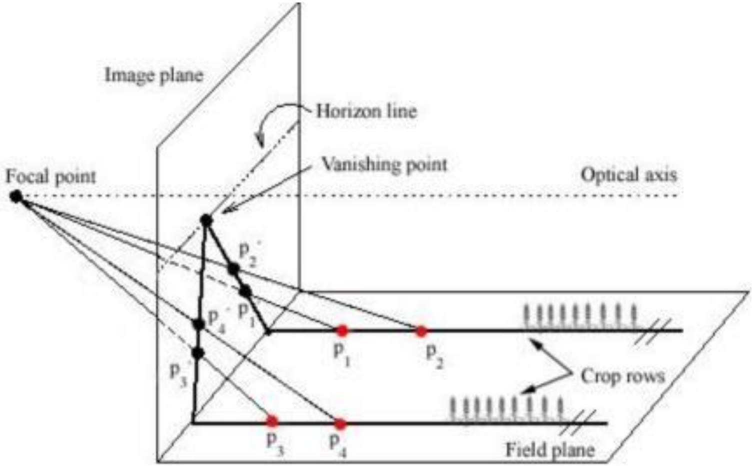




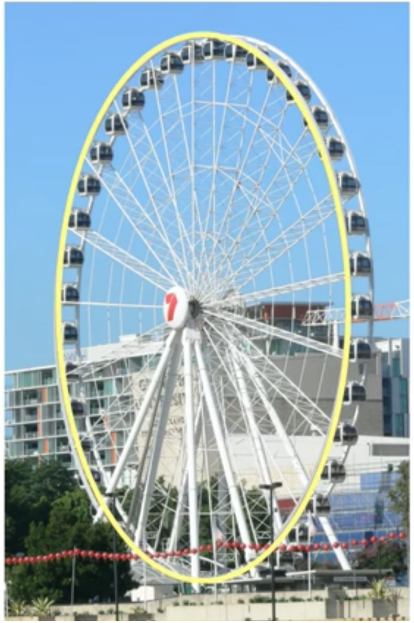


What is lost in Projective transforms?

Parallel lines meet



Distortion of Shapes



Construction of Perspective images

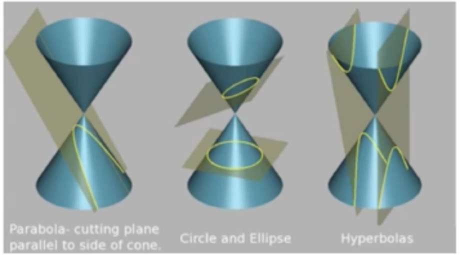
https://www.geogebra.org/classic

Soliton
Vision for a Better World

What is preserved in Projective Transform?

Perspective projection

- Maps
  - Lines → lines
    - parallel lines not necessarily parallel
    - angles are not preserved
  - Conics → conics



Significance of Previous statement

Every family of parallel lines should be changed into a family of intersecting lines and as a consequence a few of the intersecting lines may change into parallel lines

All previous transforms were dealing with points and hence were mathematically easy to derive.

This kind of projective transform actually requires one to analyse lines and hence mathematically involved?

Can we find an easier way out?

Ideal points and line at infinity

Let us assume that all parallel lines actually meet at an infinite point called the "ideal point".

All the ideal points lie on a line called the "line at infinity"

Then our transformation should focus on transforming the "ideal points" into real points

This gives rise to the two axioms of projective geometry

Axiom 1: Any two distinct points are incident with exactly one line

Axiom 2: Any two distinct lines are incident with at least one point

Supreme power that Projective transform bestows

Map an ideal point into a real point (Thus transforms parallel lines into intersecting lines)

Map the line at infinity into a “horizon line”

How to realise ideal points in practice?

Solution : Homogeneous coordinates

(2, 4, 1) ----> (2, 4)

(4, 8, 2) -----> (2,4,1) -----> (2,4)

Ideal points can be represented in the form (x, y, 0). A set of parallel lines are hypothesised to meet in an ideal point

Now we should have the ability to transform (x, y, 0) into a real point (a, b, c) where c != 0

Soliton
Vision for a Better World

Class IV: Projective transformations

$$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^{\mathsf{T}} & v \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = \begin{pmatrix} \mathtt{A} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ v_1 x_1 + v_2 x_2 \end{pmatrix}.$$

$$\mathrm{x}' = \mathbf{H}_P\,\mathrm{x} = \begin{bmatrix} \mathbf{A} & \mathrm{t} \\ \mathrm{v}^{\mathsf{T}} & v \end{bmatrix} \mathrm{x} \qquad \mathrm{v} = (v_1, v_2)^{\mathsf{T}}$$

$$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = \begin{pmatrix} \mathtt{A} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ 0 \end{pmatrix}.$$

8DOF (2 scale, 2 rotation, 2 translation, 2 line at infinity)

Action non-homogeneous over the plane

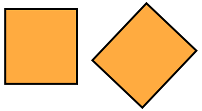
Invariants: cross-ratio of four points on a line (ratio of ratio)

Soliton
Vision for a Better World

Overview Transformations

Euclidean
3dof

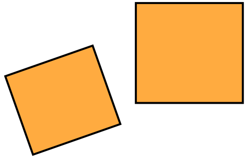
$$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



lengths, areas.

Similarity
4dof

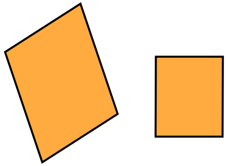
$$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



Ratios of lengths, angles.
The circular points I,J

Affine
6dof

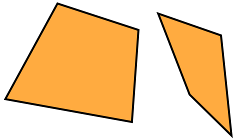
$$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



Parallelism, ratio of areas, ratio of lengths on parallel lines (e.g midpoints), linear combinations of vectors (centroids). **The line at infinity l_∞**

Projective
8dof

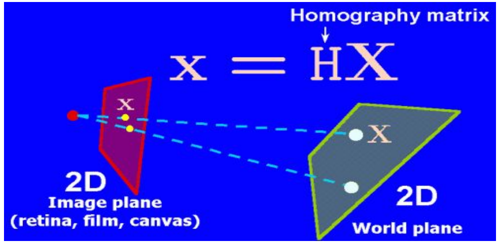
$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$



Concurrency, collinearity, order of contact (intersection, tangency, inflection, etc.), cross ratio

Soliton
Vision for a Better World

Projective Distortion



Hands - On

Demo

1. Geometric transforms ipython notebook
2. Projective transform image plane change
3.

Soliton
Vision for a Better World

Removing projective distortion

select four points in a plane with known coordinates

$$x' = \frac{x'_1}{x'_3} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \qquad y' = \frac{x'_2}{x'_3} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

(2 constraints/point, 8DOF ⇒ 4 points needed)

$$x'\left(h_{31}x + h_{32}y + h_{33}\right) = h_{11}x + h_{12}y + h_{13}$$
$$y'\left(h_{31}x + h_{32}y + h_{33}\right) = h_{21}x + h_{22}y + h_{23}$$

(linear in h_{ij})

Remark: no calibration at all necessary

Application - License plate recognition

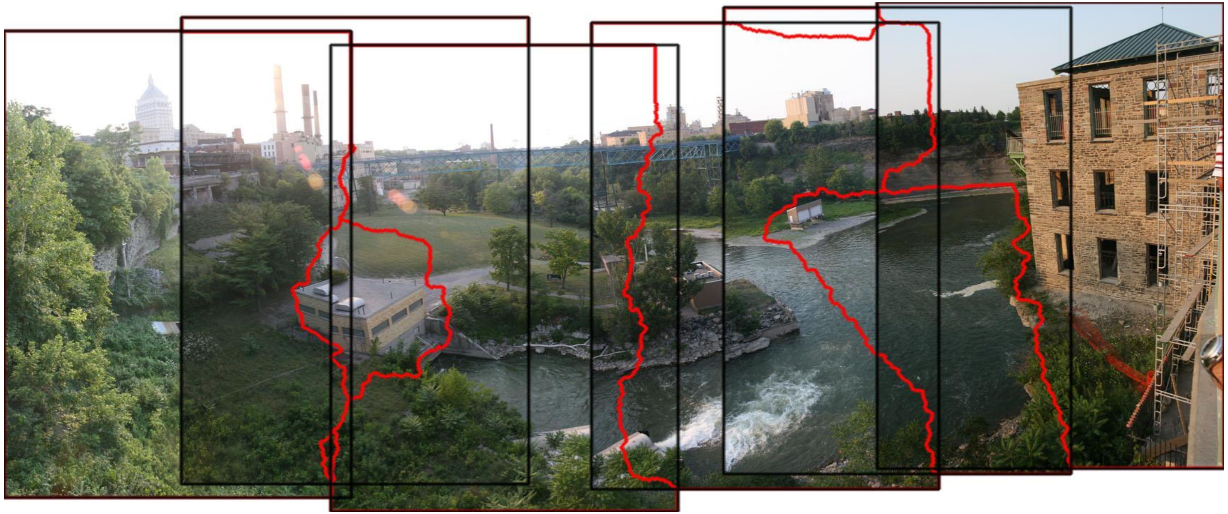
Soliton
Vision for a Better World

Image Stitching

Soliton
Vision for a Better World

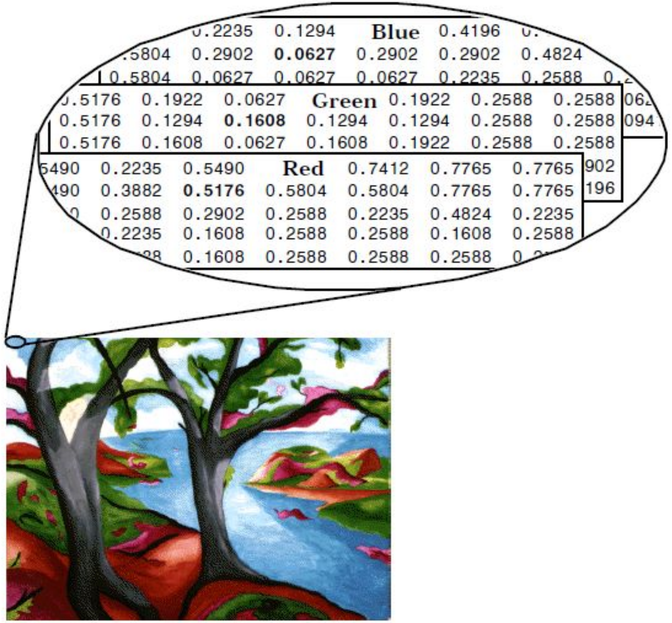
Image Stitching

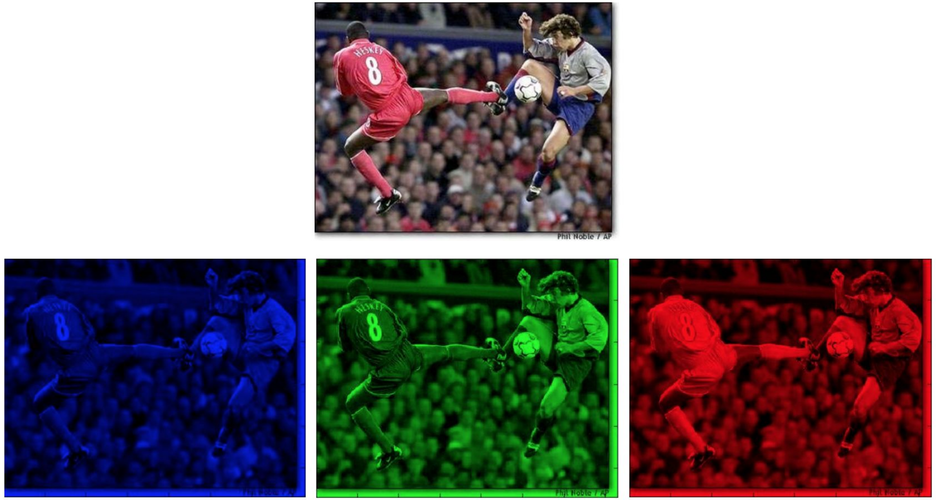
1. Repeat for all images
    a. Detect 2D Features points in both the images
    b. Find match between 2D features of both images
    c. Find Homography between two images using matched points (Do RANSAC to reduce effect of outliers)
    d. Warp the image w.r.t any particular one image
    e. Merge the images
1. Color Blend the final image



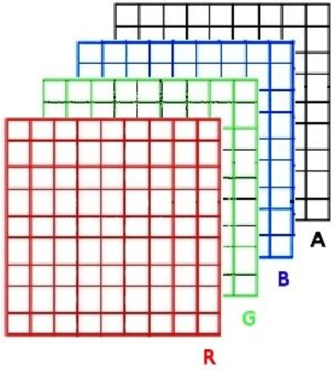

15
Coffee
Est. 2016

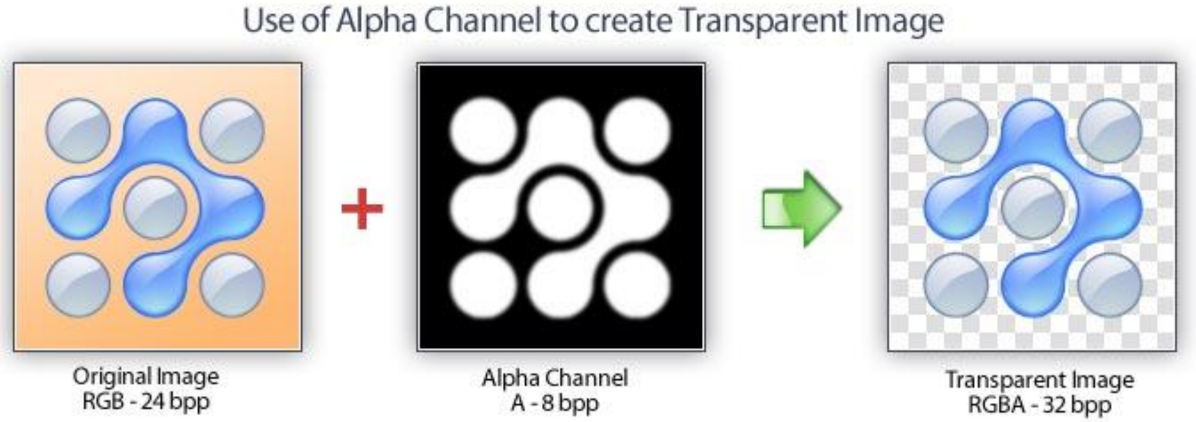
Image Representation



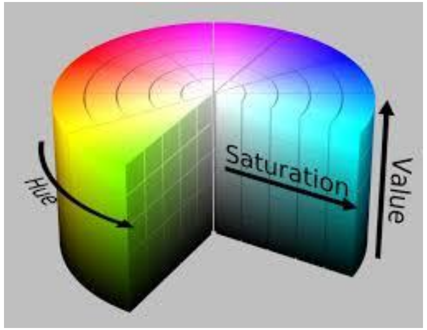


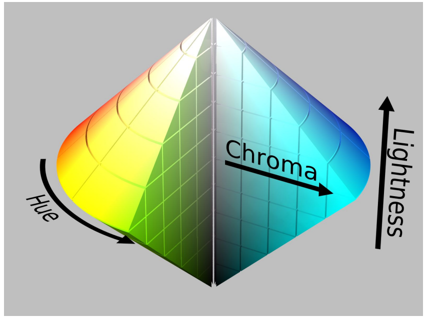
Transperant Images

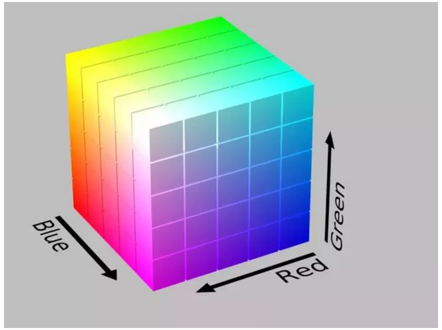


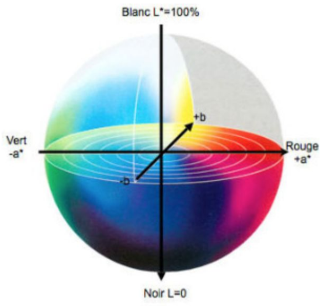


Different Color Spaces









Lab color space

Basic terminologies in images

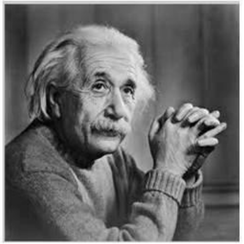
Brightness - Determines the intensity of the color presented by a pixel in a color image

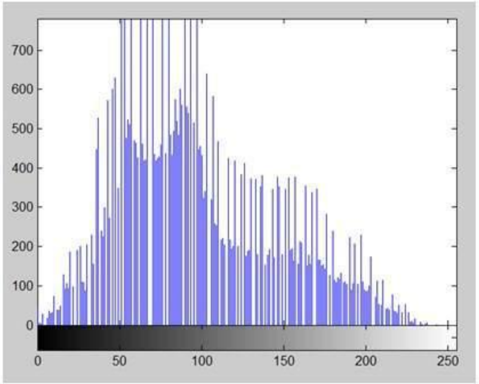
Contrast - The difference between the lightest and darkest regions of an image.

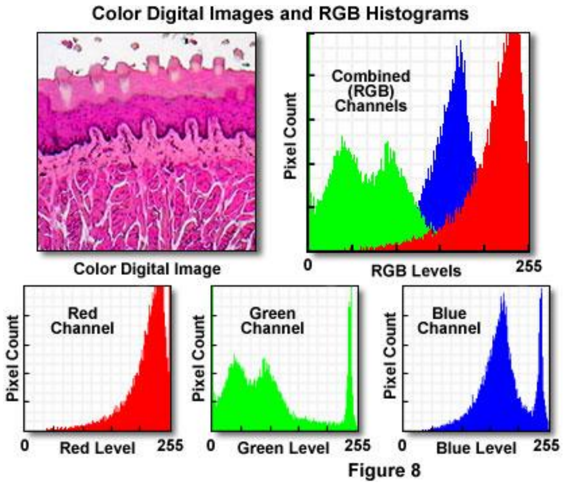
Saturation - The component of the HSV color model that controls the amount of white mixed into the hue.

Histograms

Histogram is a measure of the frequency of occurrence of each pixel value in the image



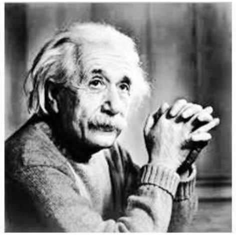




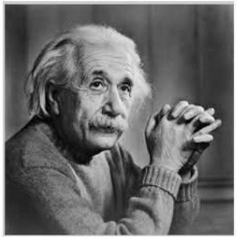
Histogram equalisation

Histogram equalization is a technique for adjusting image intensities to enhance contrast.
A very good preprocessing step to reduce the effect input image lightness varaiations

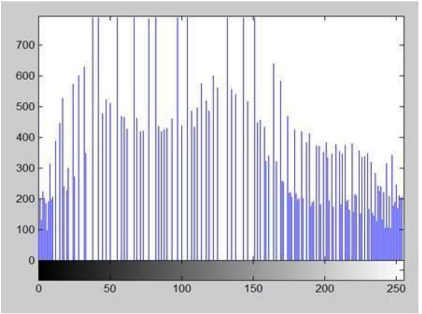
New Image



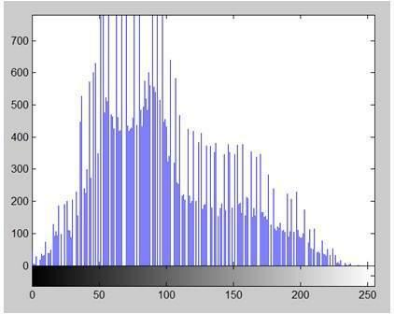
Old image



New Histogram

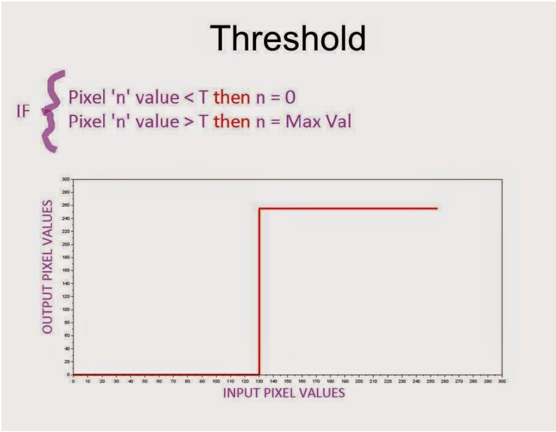


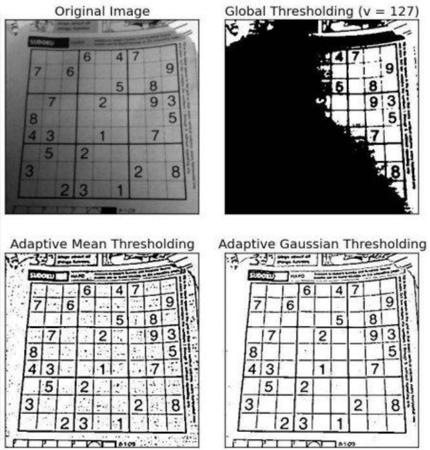
Old Histogram



Thresholding

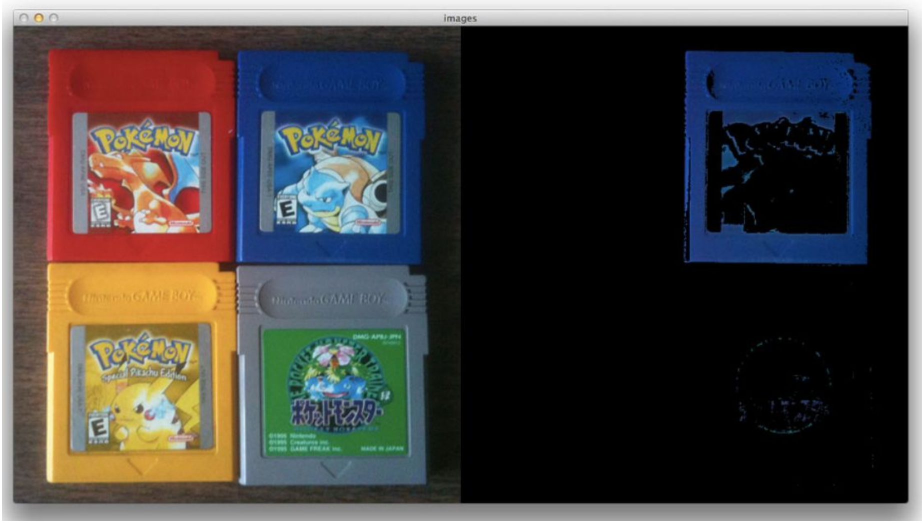
1. Converts a grayscale image to a binary image by thresholding all values below the threshold (a max limit) to zero and all values above the threshold (a max limit) to the highest value
   a. Binary
   b. Binary inverted
   c. Adaptive





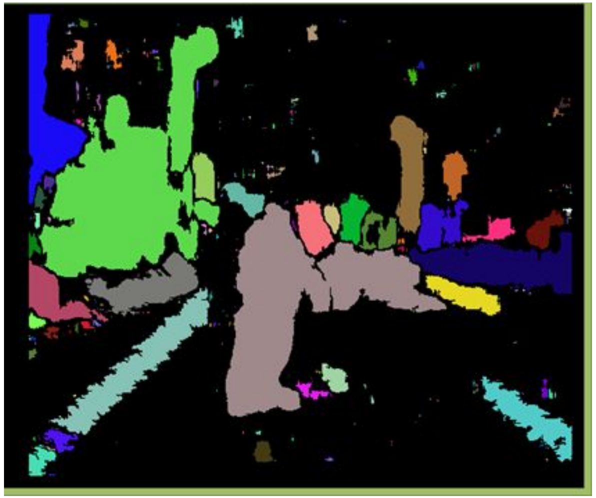
Masking a color in an image

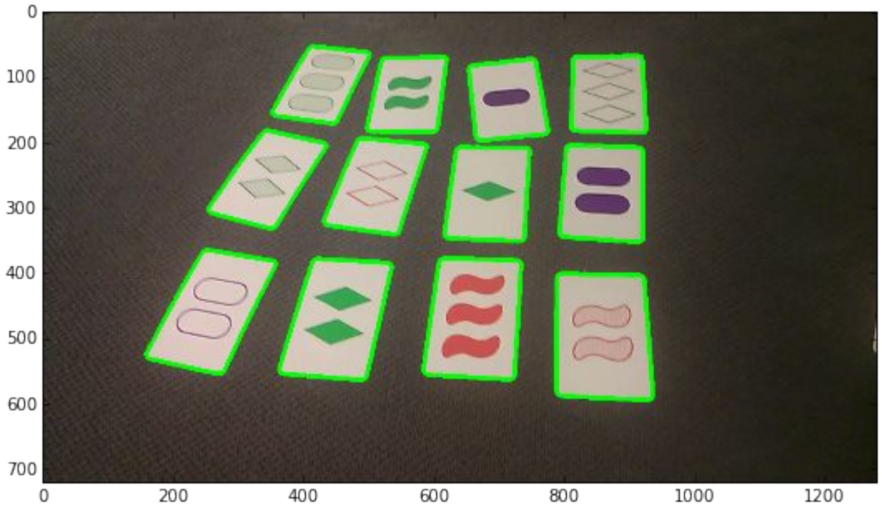
1. Find the desired range for the color in HSV
2. Get / Convert image to HSV format.
3. Keep all pixels within the range as it is and replace other pixels by zero



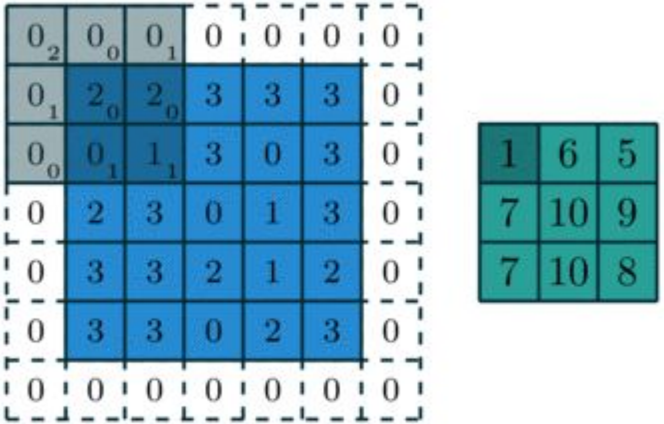
Contour detection

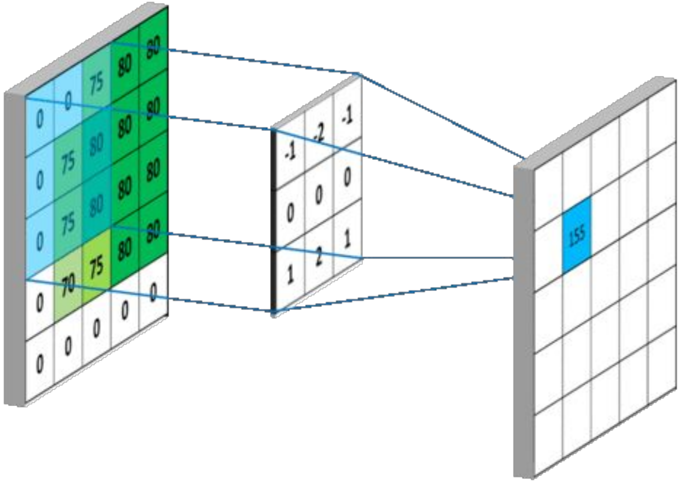
1. Contour is a curve joining all points in a boundary.
2. Contour are useful for shape analysis, object detection and analysis





Convolution





Smoothing

1. Smoothing removes noise
2. Different types of smoothing are available
   a. Gaussian
   b. Median
   c. Average
   d. Weighted average

Box filter

1/9

1	1	1
1	1	1
1	1	1

Gaussian filter

$\frac{1}{273}$

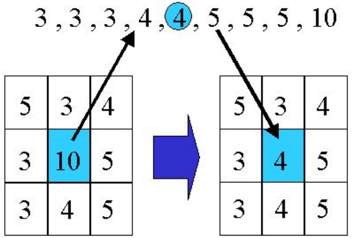
1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Weighted average filter

$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

Median filter

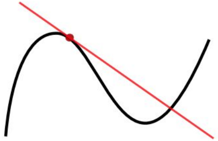


Soliton
Vision for a Better World

How to find gradients?

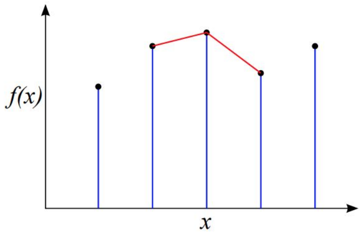
Given a function $f : \mathbb{R} \to \mathbb{R}$, its derivative is defined as

$$\frac{df}{dx}(x) = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon}$$



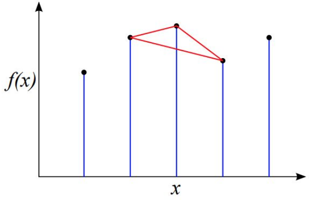
Derivative of f is the slope of the tangent to the graph of f

Derivatives of Discrete Functions



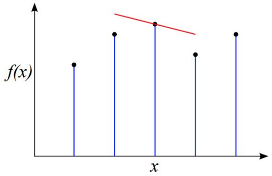
Slopes (derivatives) don’t match on left and right

Derivatives of Discrete Functions



Instead take the average of the two (or secant)

Derivatives of Discrete Functions



Instead take the average of the two (or secant)

What is an edge?

What is an Edge?

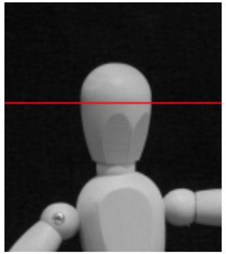
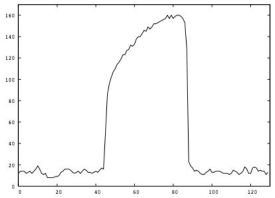


Image Value vs X-Position



An abrupt transition in intensity between two regions

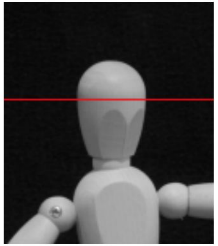


Image X-Derivative vs X-Position

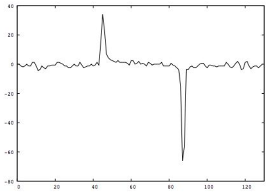
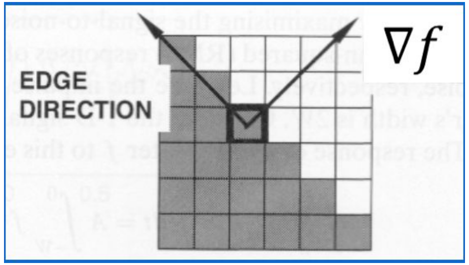


Image derivatives are high (or low) at edges



Soliton
Vision for a Better World

How to find gradients?

Finite Differences

Forward Difference

$$\Delta_+ f(x) = f(x+1) - f(x)$$ right slope

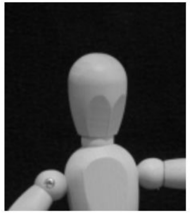
Backward Difference

$$\Delta_- f(x) = f(x) - f(x-1)$$ left slope

Central Difference

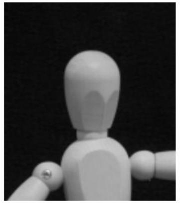
$$\Delta f(x) = \frac{1}{2}\left(f(x+1) - f(x-1)\right)$$ average slope

x-derivative using central difference:



$$* \quad \left[\tfrac{1}{2} \;\; 0 \;\; -\tfrac{1}{2}\right] \quad =$$

y-derivative using central difference:



$$* \begin{bmatrix} 0.5 \\ 0 \\ -0.5 \end{bmatrix} =$$

How to find gradients?

Finite Differences

Forward Difference

$$\Delta_+ f(x) = f(x+1) - f(x)$$ right slope

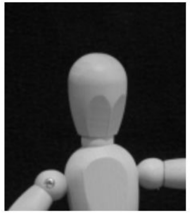
Backward Difference

$$\Delta_- f(x) = f(x) - f(x-1)$$ left slope

Central Difference

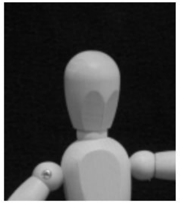
$$\Delta f(x) = \frac{1}{2}\left(f(x+1) - f(x-1)\right)$$ average slope

x-derivative using central difference:



$$* \quad \left[\tfrac{1}{2} \;\; 0 \;\; -\tfrac{1}{2}\right] \quad =$$

y-derivative using central difference:



$$* \quad \begin{bmatrix} 0.5 \\ 0 \\ -0.5 \end{bmatrix} \quad =$$

But let's smooth our gradients a little.

Prewitt

$$H_x^P = \frac{1}{3}\begin{bmatrix}1\\1\\1\end{bmatrix} * [0.5 \quad 0 \quad -0.5] = \frac{1}{6}\begin{bmatrix}1 & 0 & -1\\1 & 0 & -1\\1 & 0 & -1\end{bmatrix}$$

What is $\mathbf{H_y}^{\mathbf{P}}$ in Prewitt**?**

Sobel

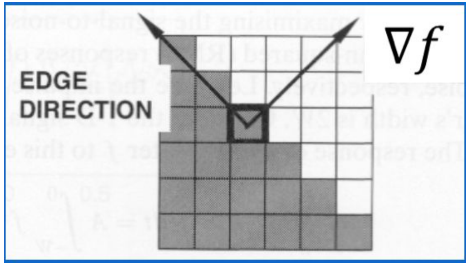
$$H_x^S = \frac{1}{4}\begin{bmatrix}1\\2\\1\end{bmatrix} * [0.5 \quad 0 \quad -0.5] = \frac{1}{8}\begin{bmatrix}1 & 0 & -1\\2 & 0 & -2\\1 & 0 & -1\end{bmatrix}$$

$$H_y^S = \frac{1}{4}[1 \quad 2 \quad 1] * \begin{bmatrix}0.5\\0\\-0.5\end{bmatrix} = \frac{1}{8}\begin{bmatrix}1 & 2 & 1\\0 & 0 & 0\\-1 & -2 & -1\end{bmatrix}$$

Now we are set to calculate our gradients?

Derivatives in 2 Dimensions

Given function	$f(x,y)$
Gradient vector	$\nabla f(x,y) = \begin{bmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$
Gradient magnitude	$\lvert \nabla f(x,y) \rvert = \sqrt{f_x^2 + f_y^2}$
Gradient direction	$\theta = \tan^{-1} \dfrac{f_x}{f_y}$



Morphological operations

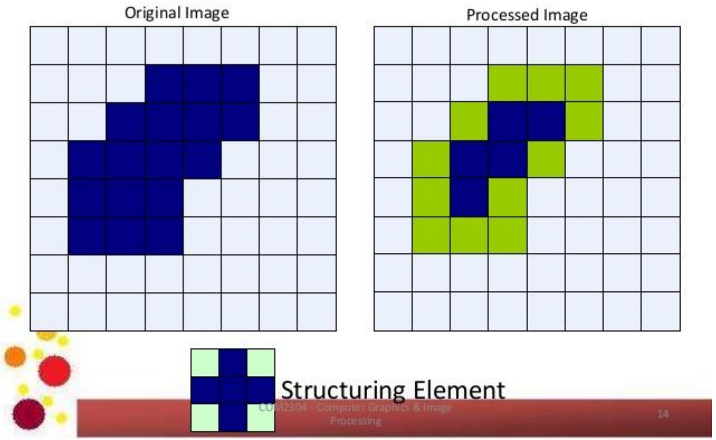
A collection of non-linear operations that is related to shape or morphology of features in an image.

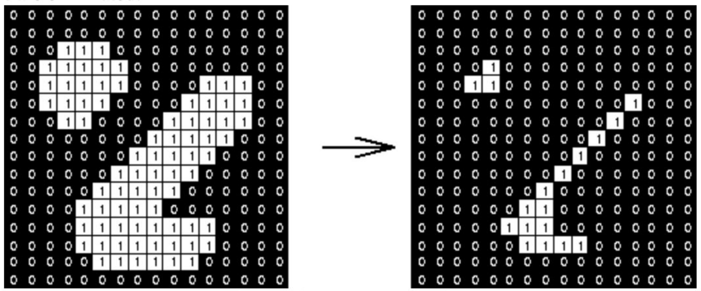
There are four kinds of morphological operations in an image

1. Erosion
2. Dilation
3. Opening
4. Closing

Erosion

The structuring element is superimposed on every pixel in the image. If all pixels within the structuring element are foreground, then the input pixel retains it value. If any pixel within the structuring element is background, then the input pixel is replaced by background.





Erosion: a 3×3 square structuring element (www.cs.princeton.edu/~pshilane/class/mosaic/).

Greyscale image | Binary image by thresholding | Erosion: a 2×2 square structuring element

Dilation

The structuring element is superimposed on every pixel in the image. If all pixels within the structuring element are background, then the input pixel retains it value. If any pixel within the structuring element is foreground, then the input pixel is replaced by foreground.

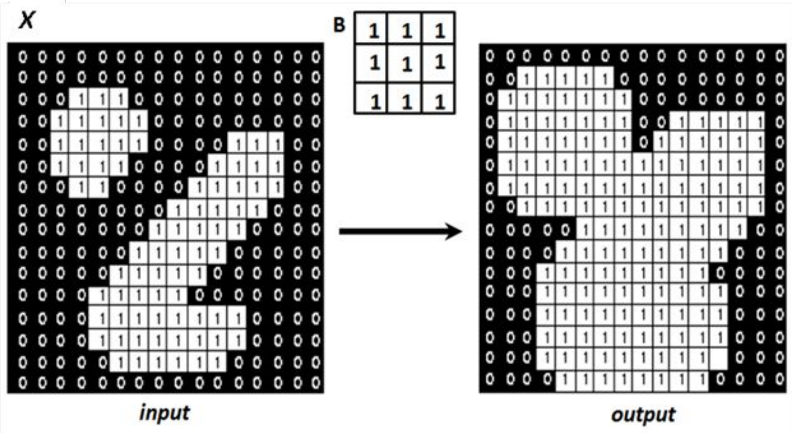
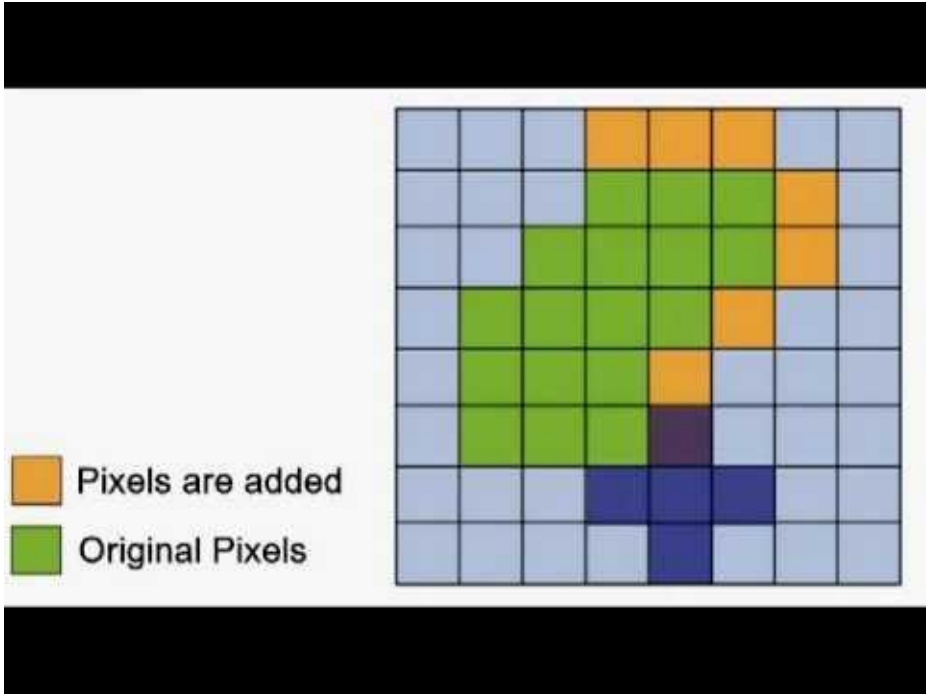
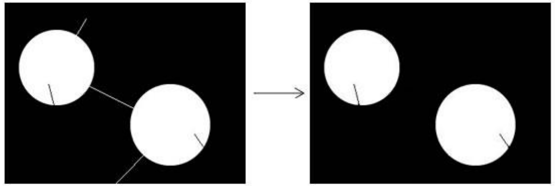


Figure 1. Effect of dilation using a 3X3 square structural element B.

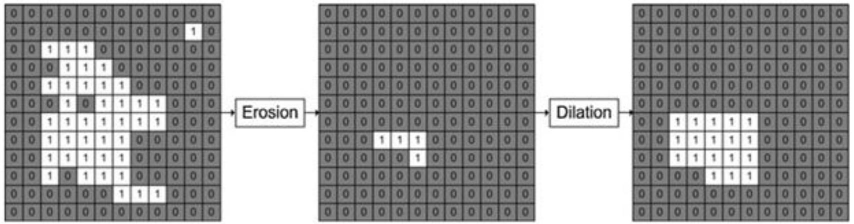


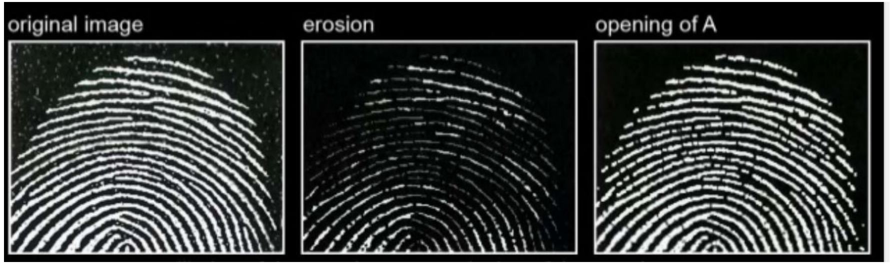
Opening



Morphological Opening

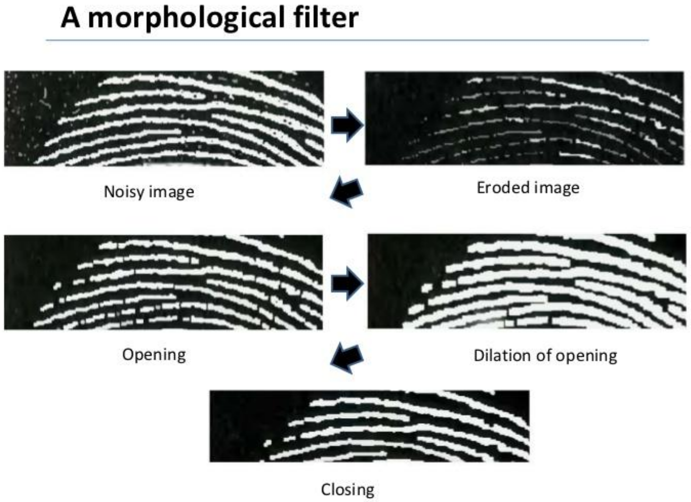
1. Erosion followed by dilations
2. Smooths contours and eliminates protrusions

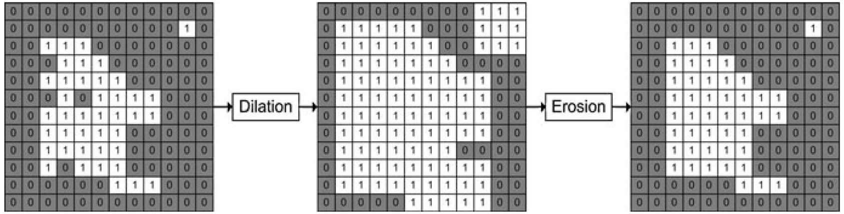




Closing

1. First dilate and then erode
2. This smooths sections of contours, fuses narrow breaks.





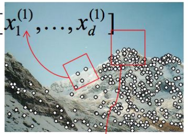
Establishing correspondence

Local features: main components

1) **Detection:** Identify the interest points

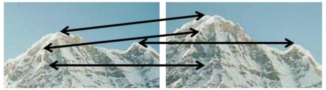
2) **Description** :Extract feature vector descriptor surrounding each interest point.

$\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$



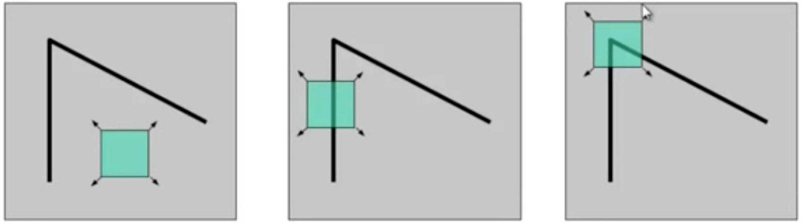
$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$

3) **Matching:** Determine correspondence between descriptors in two views



Kristen Grauman

Properties of an interest point



"flat" region: no change in all directions

"edge": no change along the edge direction

"corner": significant change in all directions

Properties of Interest Point Detectors

- Detect all (or most) true interest points
- No false interest points
- Well localized.
- Robust with respect to noise.
- Efficient detection

Some of the commonly used interest point detector

1. Harris point detector
2. Shi-Tomasi detector
3. Features from accelerated Segment Test (FAST)

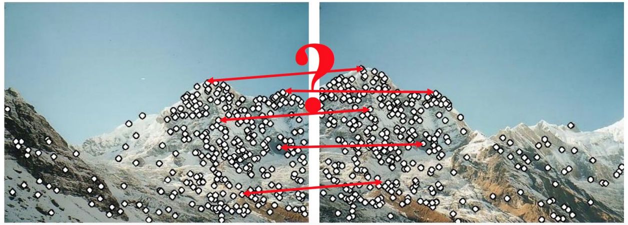
Feature descriptor

Few of the desired properties of key point descriptors

1. Rotation Invariance
2. Scale Invariance

Feature descriptors

We know how to detect good interest points
Next question:
How to match image regions around interest points?
Answer: **Need feature descriptors**



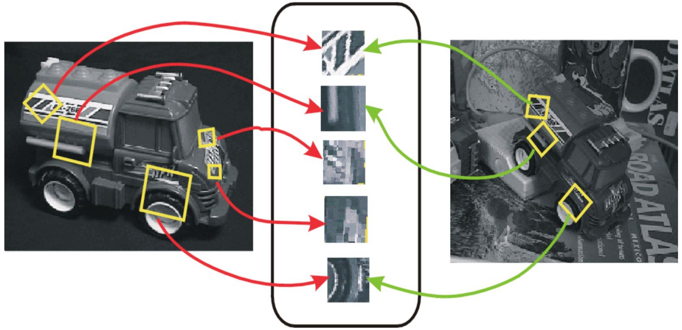
Some of the commonly used interest point descriptors

1. Scale Invariant Feature Transform (SIFT)
2. Speeded Up Robust Features (SURF)
3. Binary Robust Independent Elementary Features (BRIEF)
4. Oriented FAST and rotated BRIEF (ORB)

Feature matchers

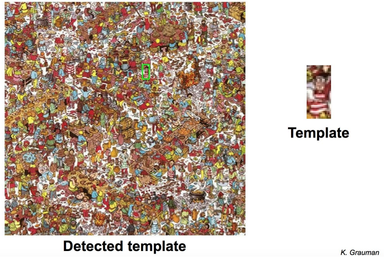
Process of feature matching is as follows

1. Find the distance of each descriptors in the given image to every descriptor in the template image
2. Find the best match for each descriptor and its corresponding distance.
3. Remove the pairs which are below a threshold
4. Still a few bad pairs will be left. To work around this, use louse's ratio. Find the ratio of the distance of the first best match to the second best match. If the ratio > 0.6, eliminate the pair



Template matching

image



1. Given a template image and an input image to search for, search where exactly is the template image located in the input image.
2. Only translation and intensity variations are allowed. No other geometric operation is allowed
3. But how do we measure similarity between two patches of images.
4. Searching through the whole image might be time consuming?

How do we measure similarity between two patches

The similarity between two patches g1 and g2 can be found by the following ways

1. Sum of squared differences (SSD)

   $\sum(g_2(m) - g_1(m))^2$

1. Sum of absolute difference (SAD)

   $\sum |g_2(m) - g_1(m)|^2$

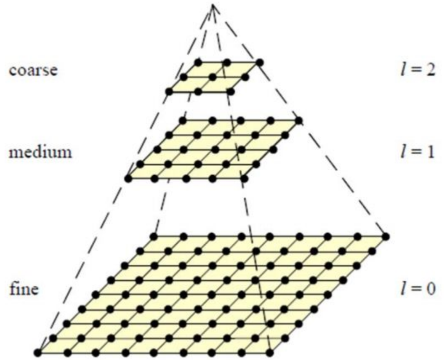
1. Maximum of difference

   $\text{Max} = \max_m |g_2(m) - g_1(m)|$

But these provide no invariance against brightness and contrast !!!!

Solution - Normalised Correlation (Convolution) And Image pyramids

$$\frac{\sum_{i=-N}^{N}\left(F(i)I(x+i)\right)}{\sqrt{\sum_{i=-N}^{N}\left(I(x+i)\right)^2}\sqrt{\sum_{i=-N}^{N}\left(F(i)\right)^2}}.$$



Day 1 - FN 2 (Basic Image Processing)

Lunch Break

Feed yourselves well to feed the Machines more..!

Day 1 - AN 1 (Basic Image Processing)


15
Coffee
Est. 2016

Day 1 - AN 2 (Machine Learning Intro)

Session II (ML)

- ➔ Object Classification
- ➔ Classification: Simple Version
- ➔ Features
- ➔ Classification: CV based
- ➔ Linear classifiers
- ➔ SVM

Soliton
Vision for a Better World

What is Machine Learning?

- Arthur Samuel (1959): Field of study that gives computers the ability to learn without being explicitly programmed
- A well-posed learning problem (1998): A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance measure P, as measured by P, improves with experience E.

Soliton
Vision for a Better World

Credits: cs229, Andrew Y. Ng

What is Learning?

- Set of training samples : $[x_1, \ldots x_n]$ & $[y_1 \ldots y_n]$
- Hypothesis function ($w1x1 + w2x2 = y$)
- Optimize the loss function
  - Update / change the coefficients such that they minimize the error
  - Perform till convergence
- Getting optimal values for '*w*' is called learning
- Now for a new x, predict it's y

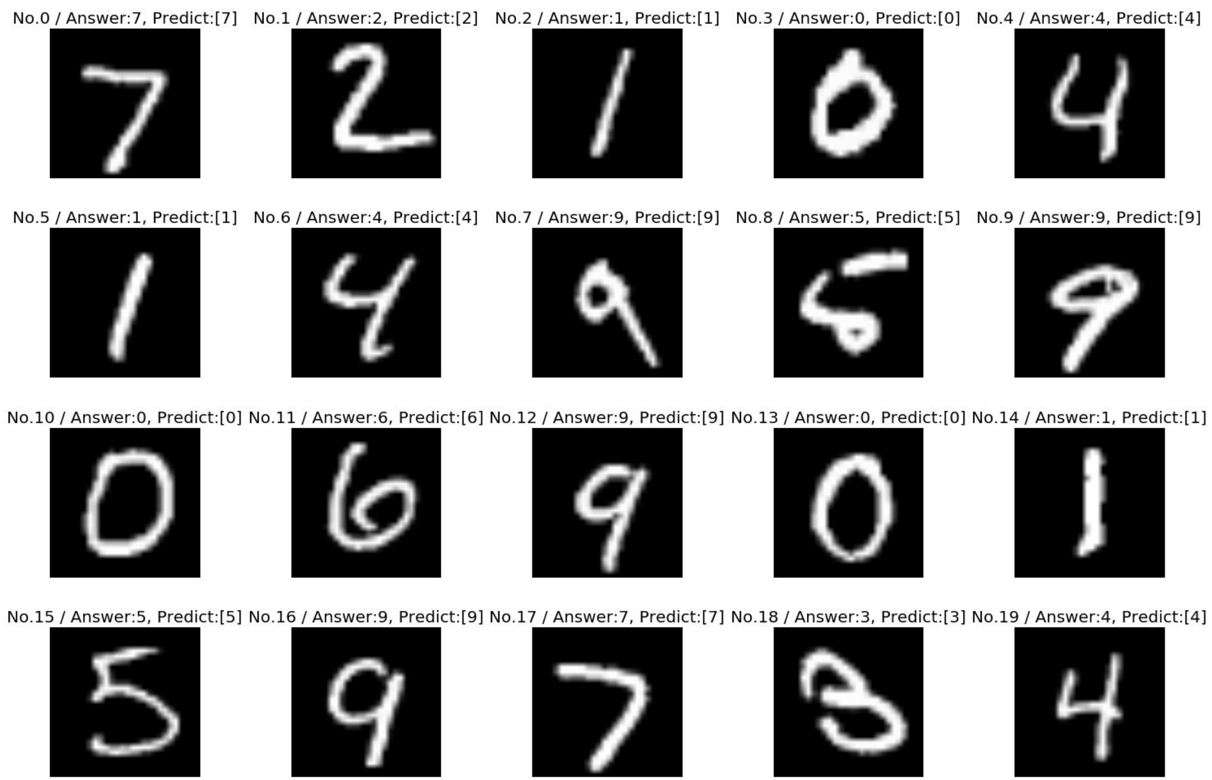
Soliton
Vision for a Better World

Credits: cs229, Andrew Y. Ng

Classification

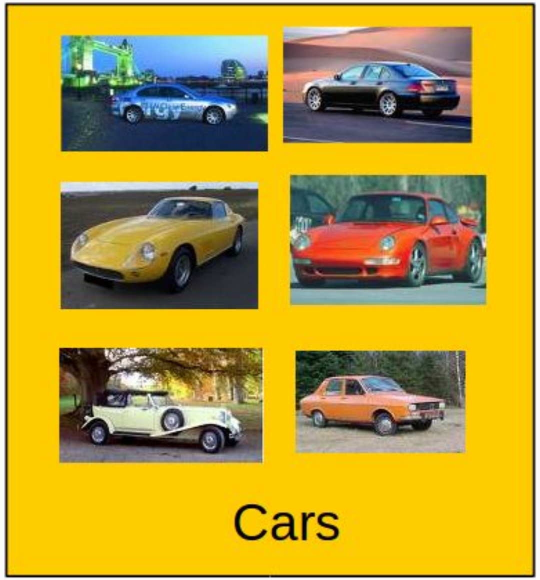
Handwritten character Classification

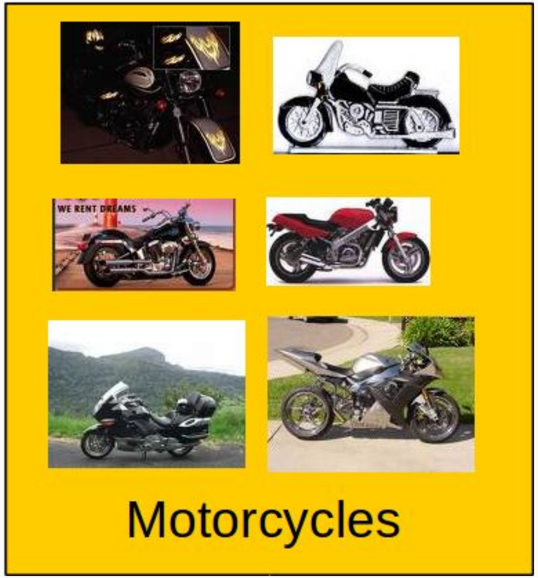
Q: As a Human, how do I learn to classify?



Classification

1. Learn Patterns
2. Learn Rules





What is this?

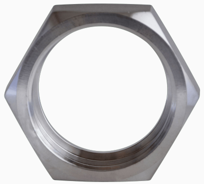
Soliton
Vision for a Better World

Simple Problem

How to classify a new image into any of the two categories?

Simplified Problem

- Conveyor Belt: 2D Image of Nut & Bolt
- Supervised Learning Problem
    - Train on available data
    - Test on new data



Nut

Bolt

Rules

We can classify a new image into any of these two categories by forming a set of rules.

Q: List down few rules by which we can classify

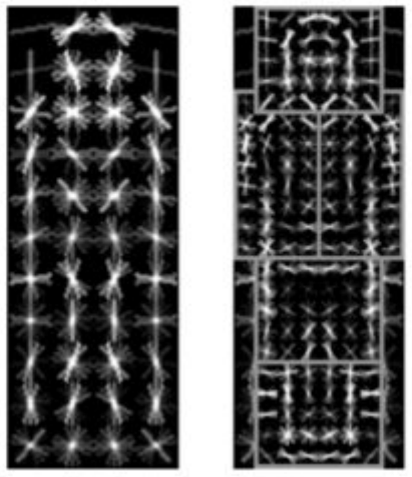
Bolts	Nuts
1. Longer 2. Thinner 3. Cylindrical in shape 4. More compact	1. Circular 2. Has cavity in the centre 3. Less area

Soliton
Vision for a Better World

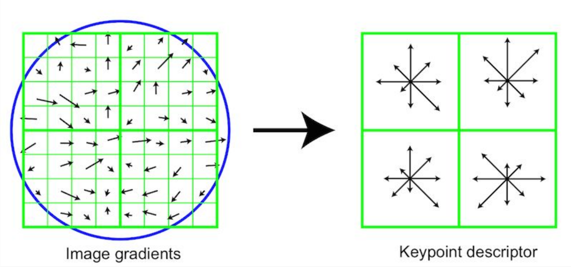
Features

In Machine Learning terms, we call them as features

Low-level Features



HoG



SIFT

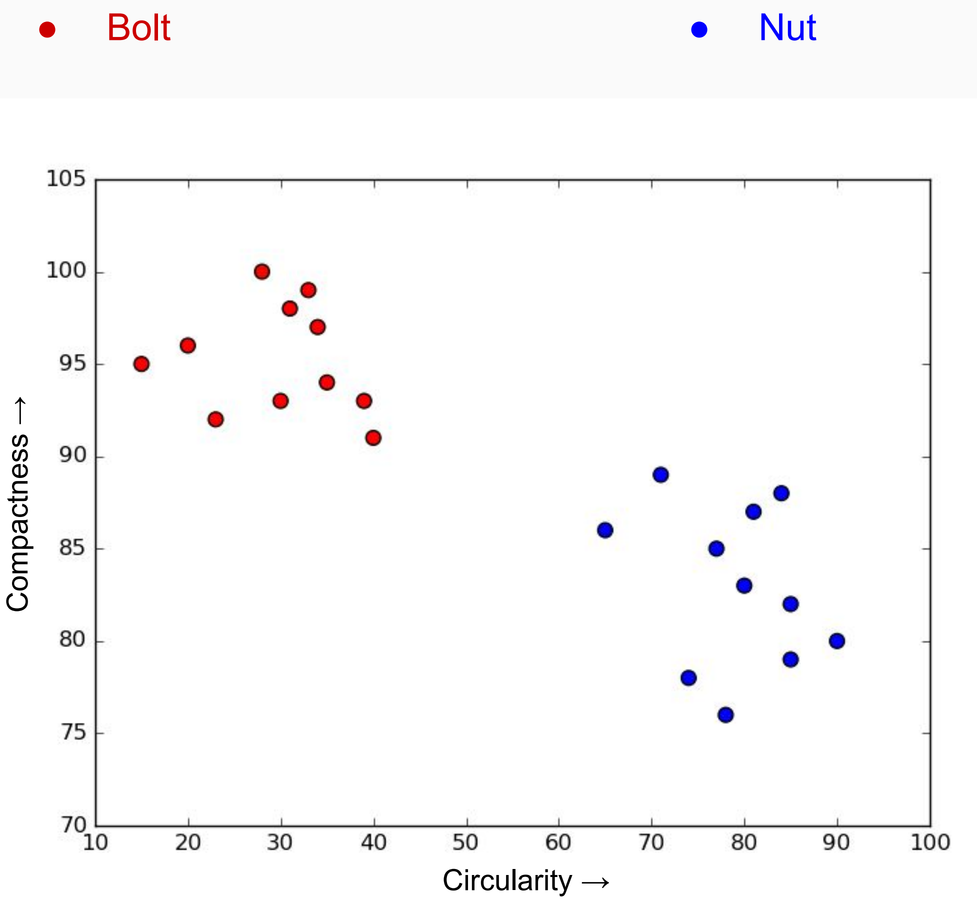
Data

The graph shows the distribution of data with respect to the below features

Features

1. Circularity
2. Compactness

Now, How can you classify the data?



Soliton
Vision for a Better World

Classification

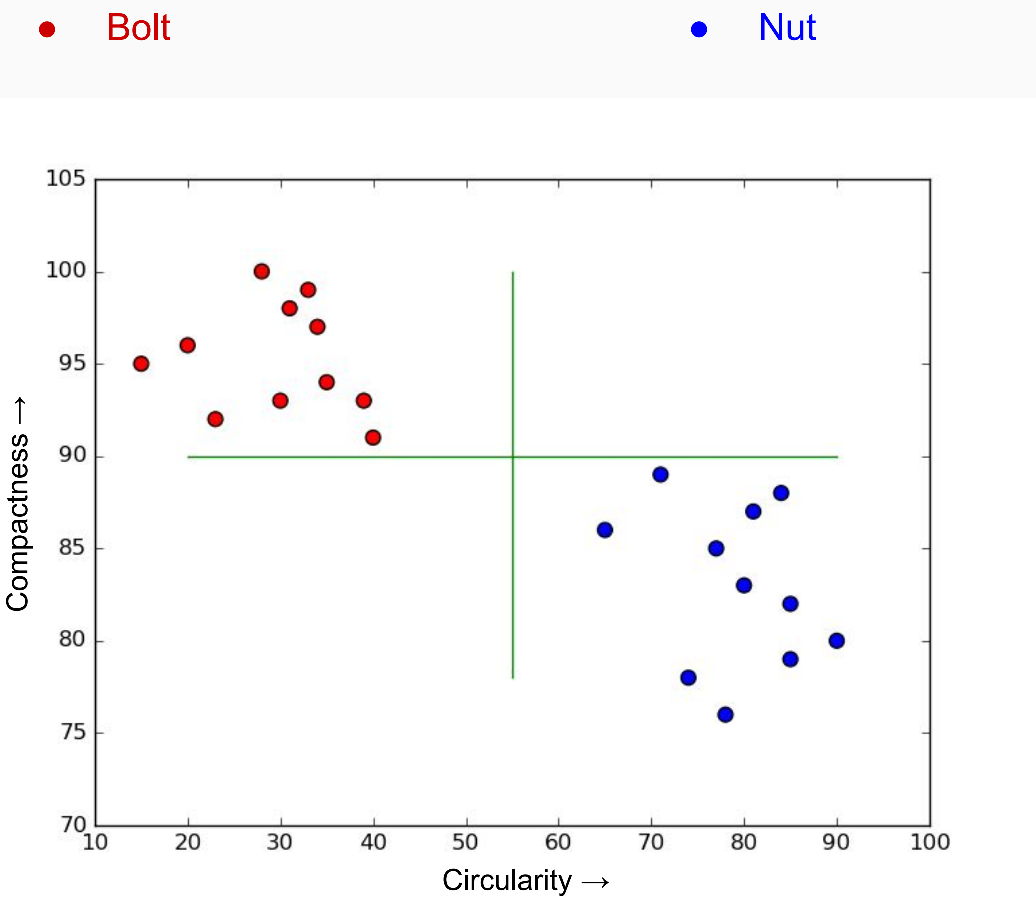
It is a simple if with two conditions

```
if Circularity > 55 and
Compactness < 90:

    Nut

elif Circularity < 55 and
Compactness > 90:

    Bolt
```



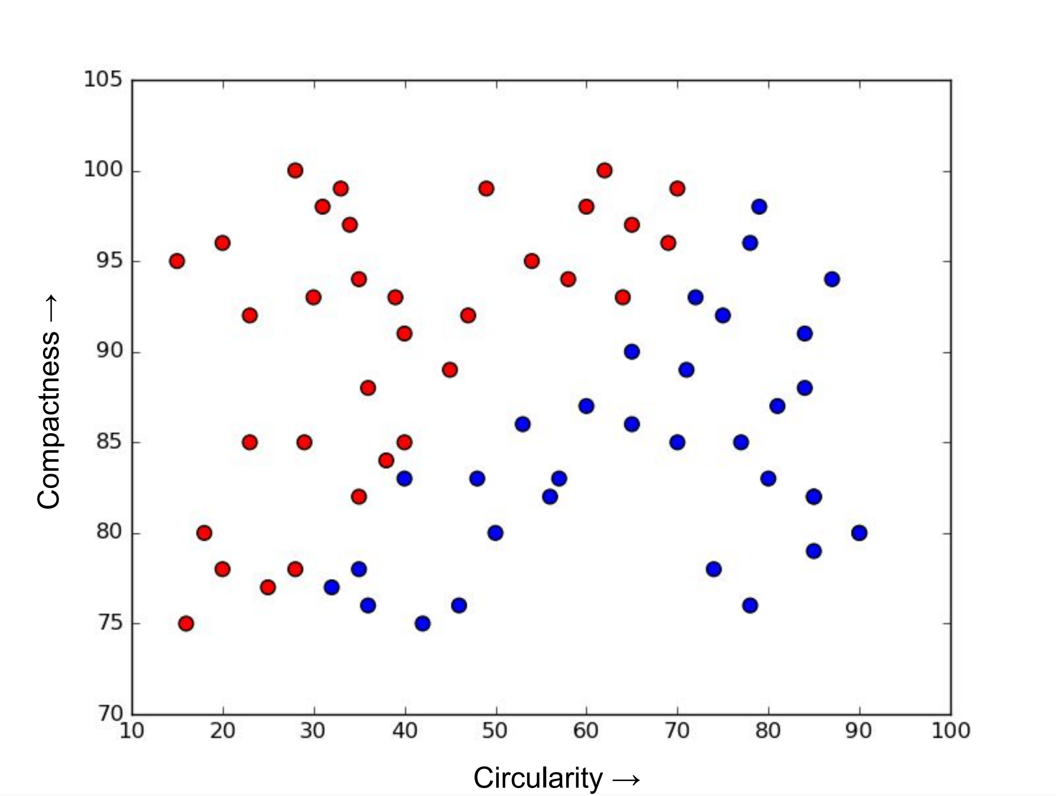
Soliton
Vision for a Better World

Complex Data

The Data becomes more complex to classify using a simple If Classifier

Now, How can you classify the data?

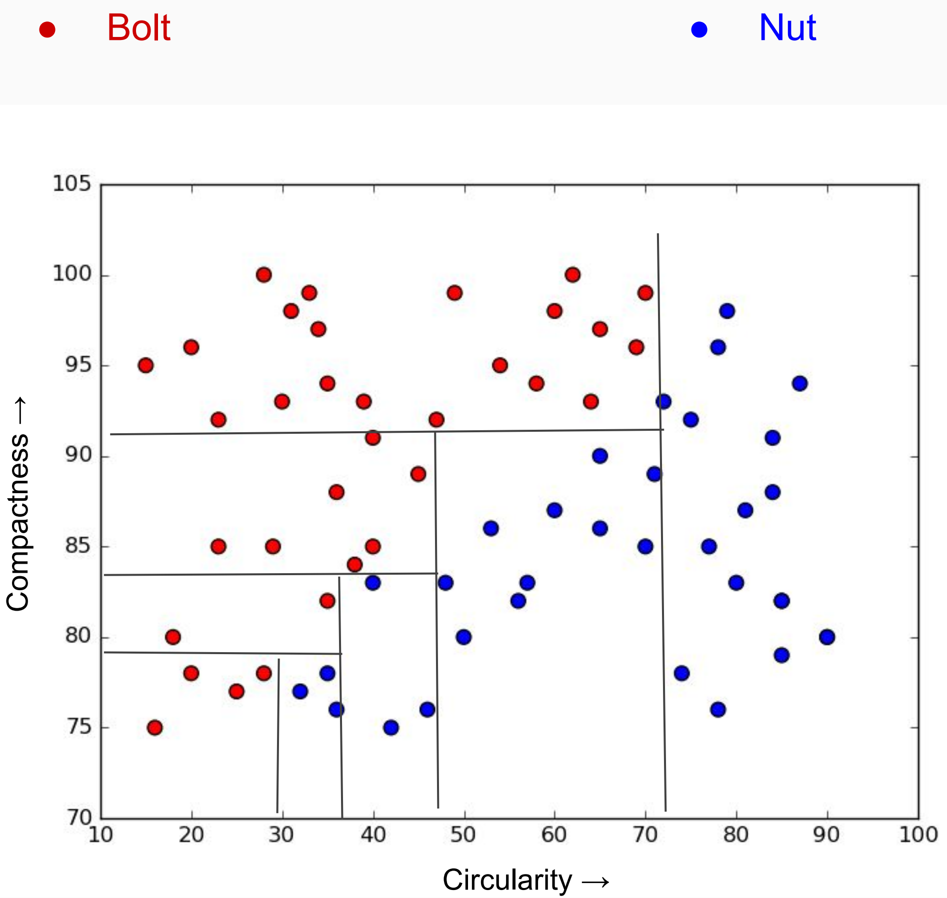
● Bolt ● Nut



Soliton
Vision for a Better World

Decision Tree

- Non-parametric Supervised Learning
- Approximating data by a set of if-then-else decision rules
- Greedy algorithm
- Whitebox Model :)



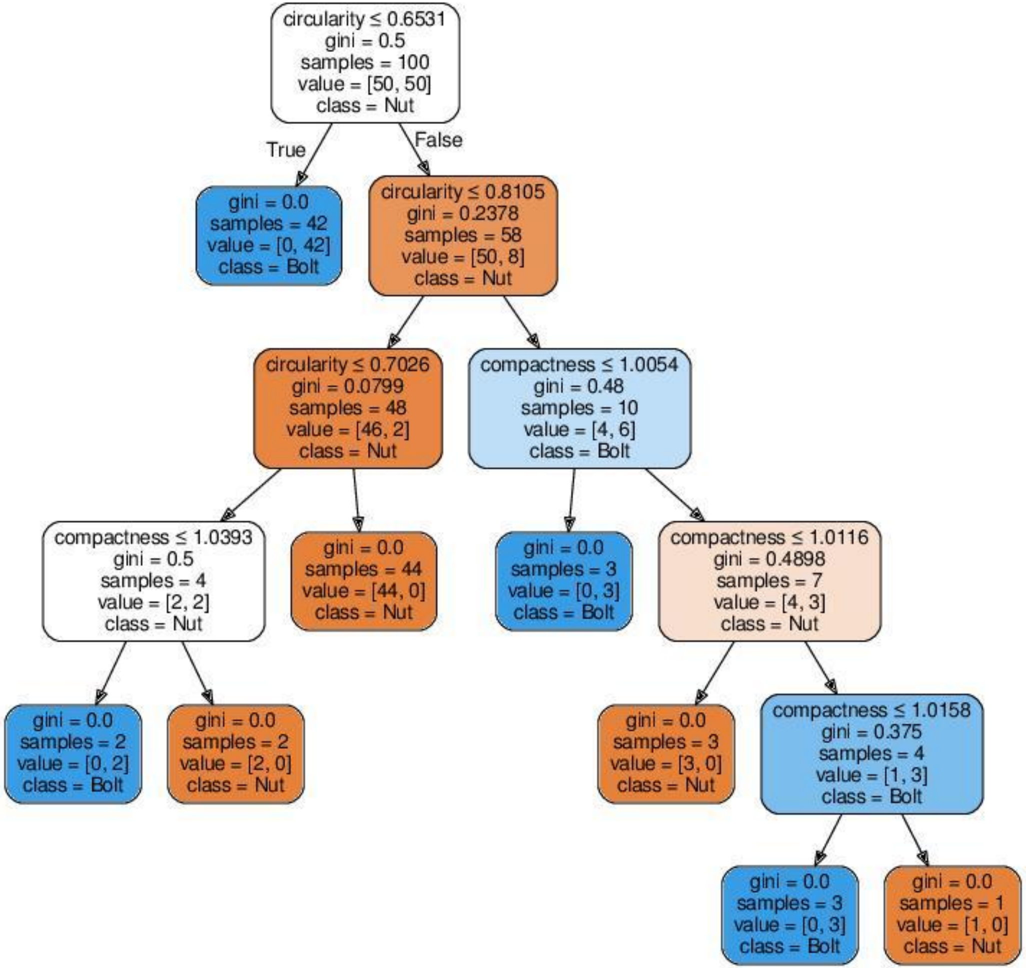
Soliton
Vision for a Better World

Decision Tree

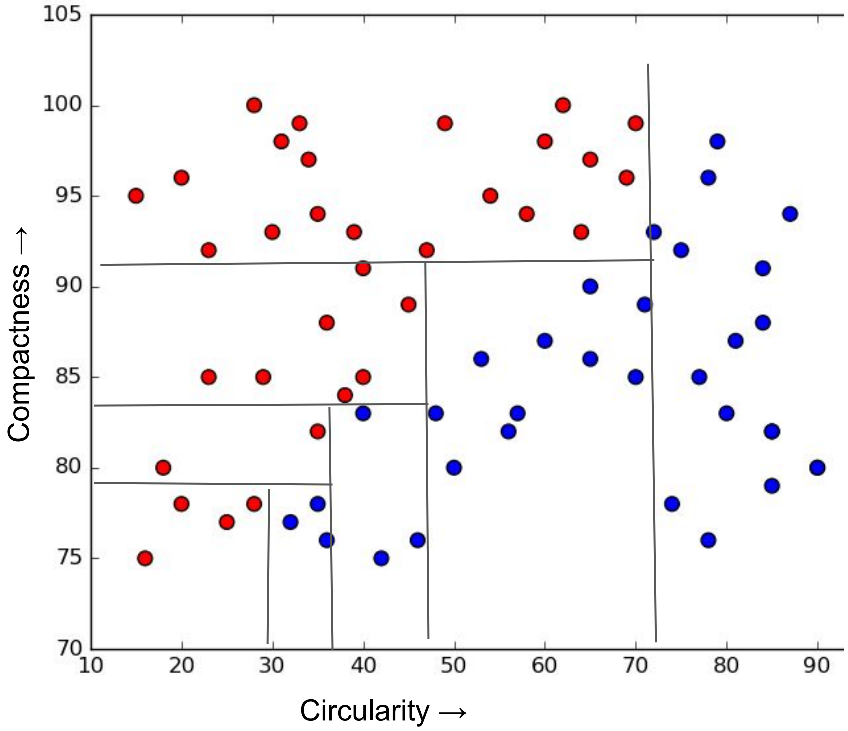
Decision Tree generated for the Nuts and Bolts classification problem

Cons

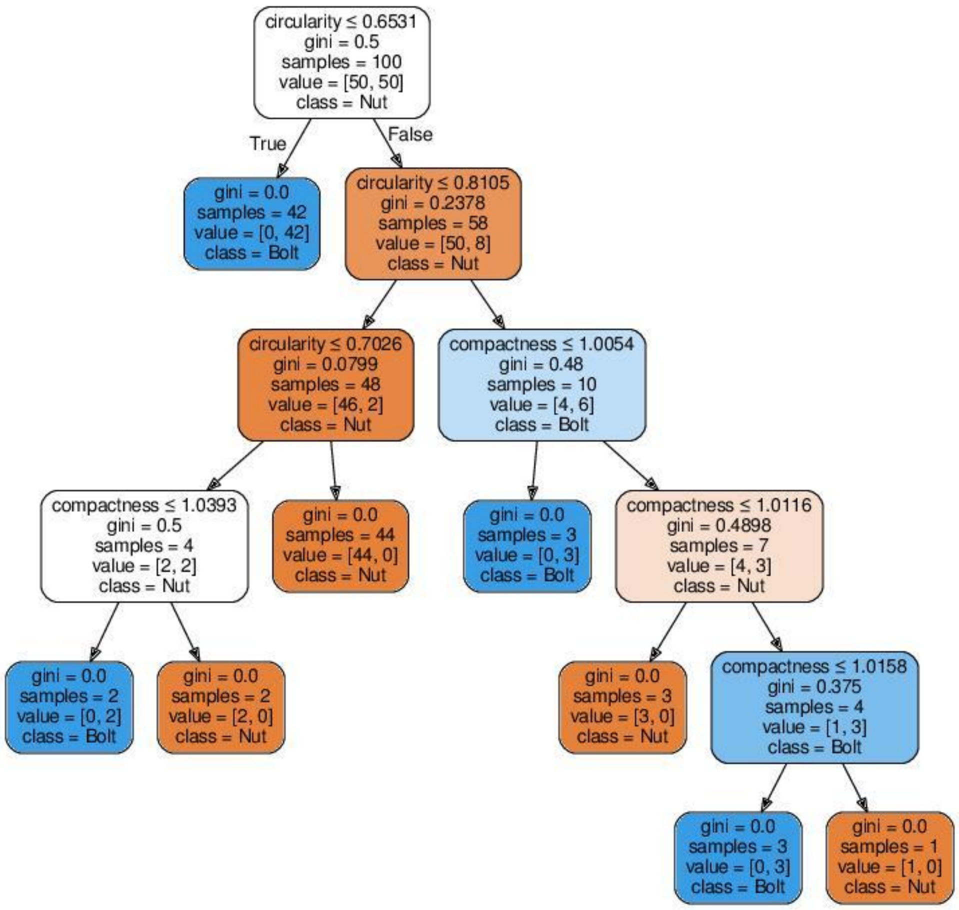
- May grow huge and hard to manage / visualize
- Problem of overfitting



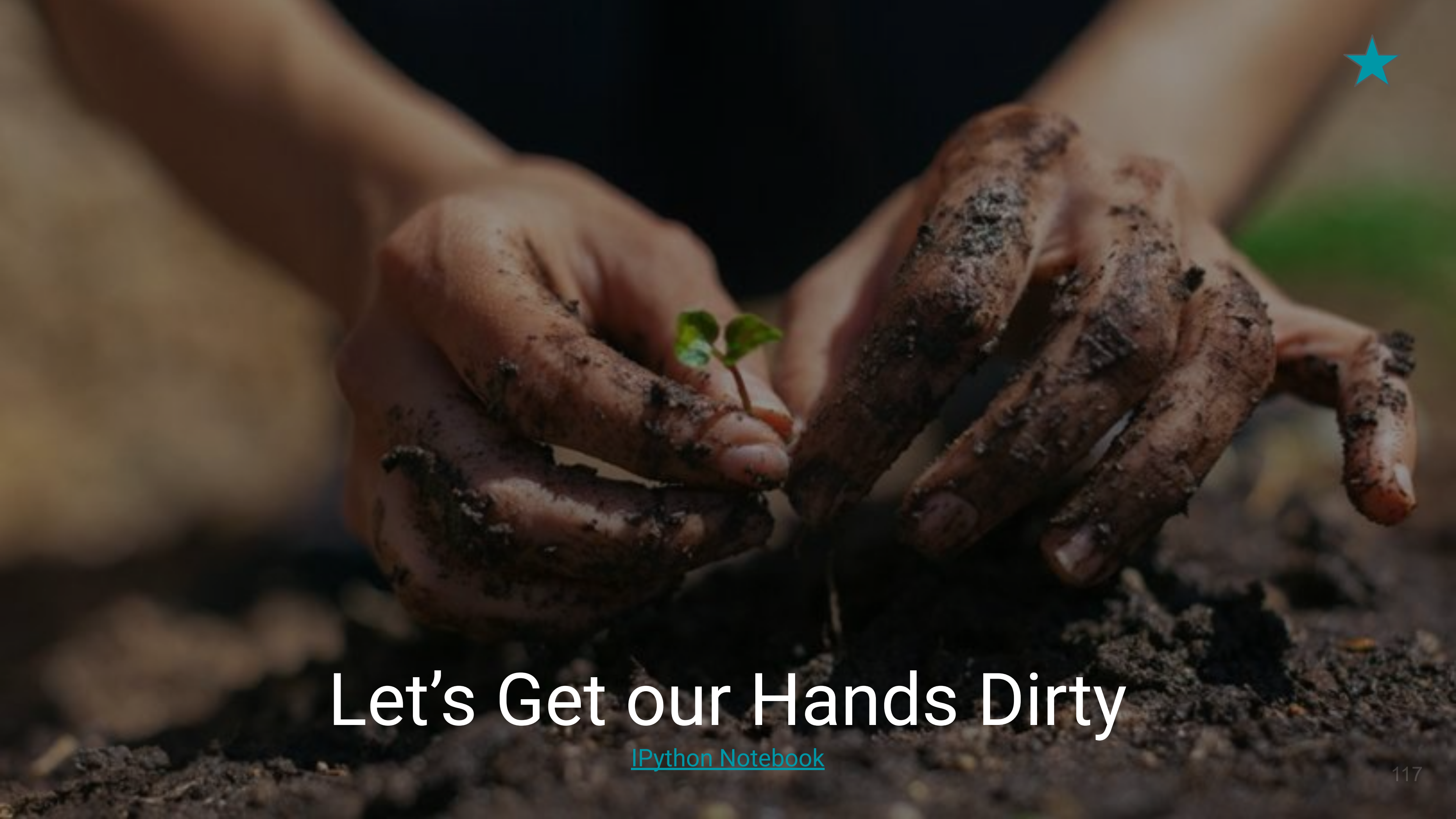
Soliton
Vision for a Better World



105
100
95
90
85
80
75
70
10
20
30
40
50
60
70
80
90
Compactness →
Circularity →



circularity ≤ 0.6531
gini = 0.5
samples = 100
value = [50, 50]
class = Nut
True
False
gini = 0.0
samples = 42
value = [0, 42]
class = Bolt
circularity ≤ 0.8105
gini = 0.2378
samples = 58
value = [50, 8]
class = Nut
circularity ≤ 0.7026
gini = 0.0799
samples = 48
value = [46, 2]
class = Nut
compactness ≤ 1.0054
gini = 0.48
samples = 10
value = [4, 6]
class = Bolt
compactness ≤ 1.0393
gini = 0.5
samples = 4
value = [2, 2]
class = Nut
gini = 0.0
samples = 44
value = [44, 0]
class = Nut
gini = 0.0
samples = 3
value = [0, 3]
class = Bolt
compactness ≤ 1.0116
gini = 0.4898
samples = 7
value = [4, 3]
class = Nut
gini = 0.0
samples = 2
value = [0, 2]
class = Bolt
gini = 0.0
samples = 2
value = [2, 0]
class = Nut
gini = 0.0
samples = 3
value = [3, 0]
class = Nut
compactness ≤ 1.0158
gini = 0.375
samples = 4
value = [1, 3]
class = Bolt
gini = 0.0
samples = 3
value = [0, 3]
class = Bolt
gini = 0.0
samples = 1
value = [1, 0]
class = Nut

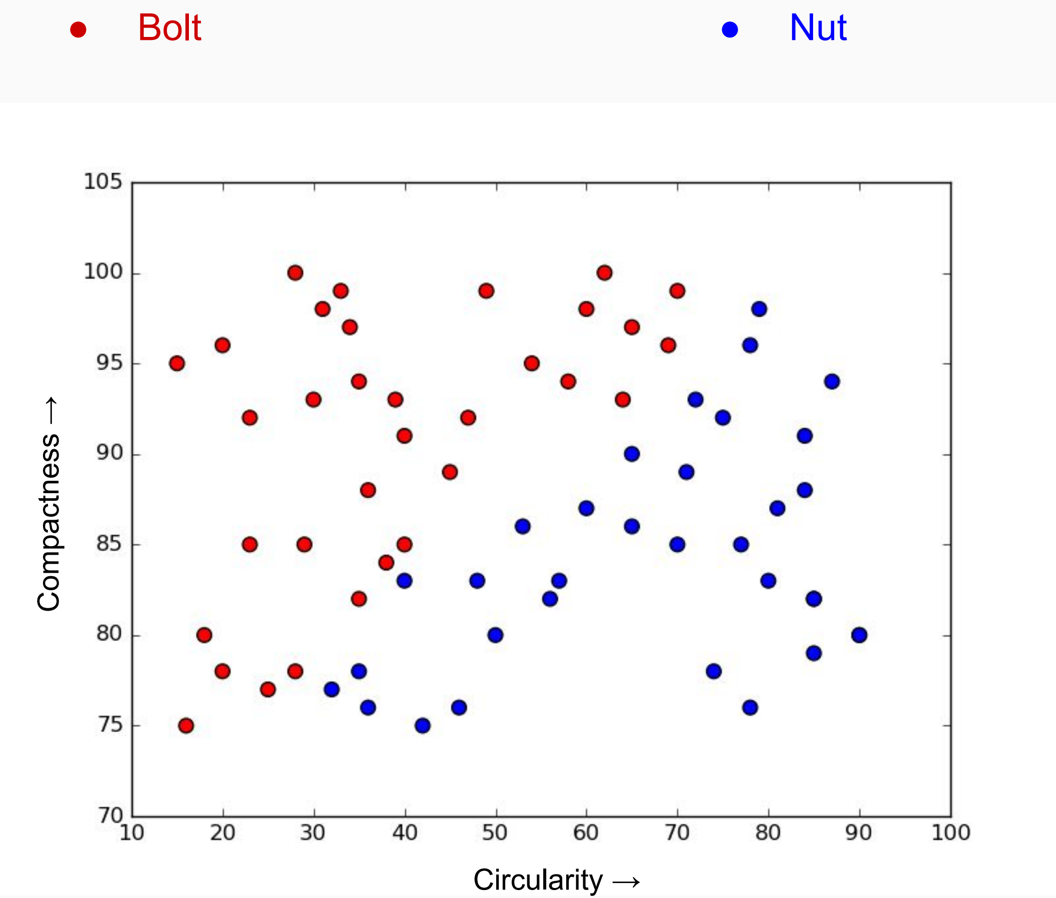


Let’s Get our Hands Dirty

[IPython Notebook]

Closing Thoughts

What is the Cleaner way of Solving this problem?

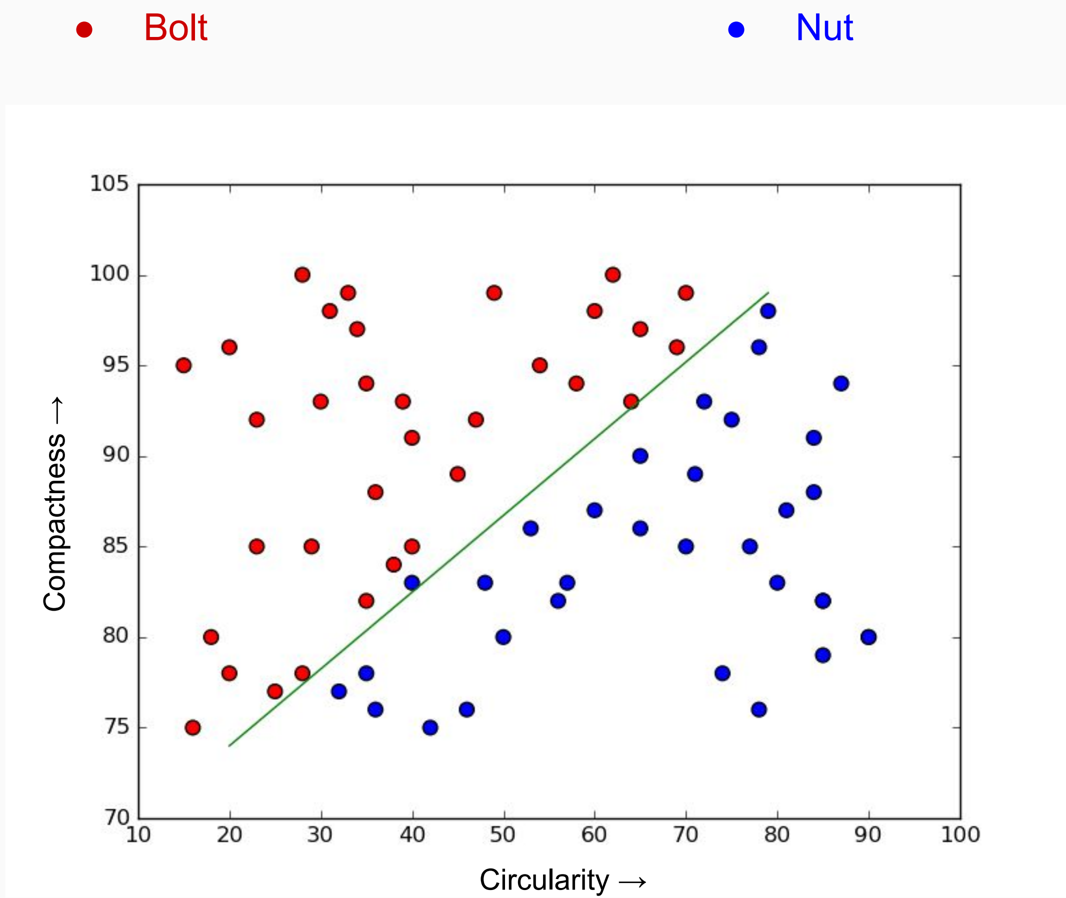


Soliton
Vision for a Better World

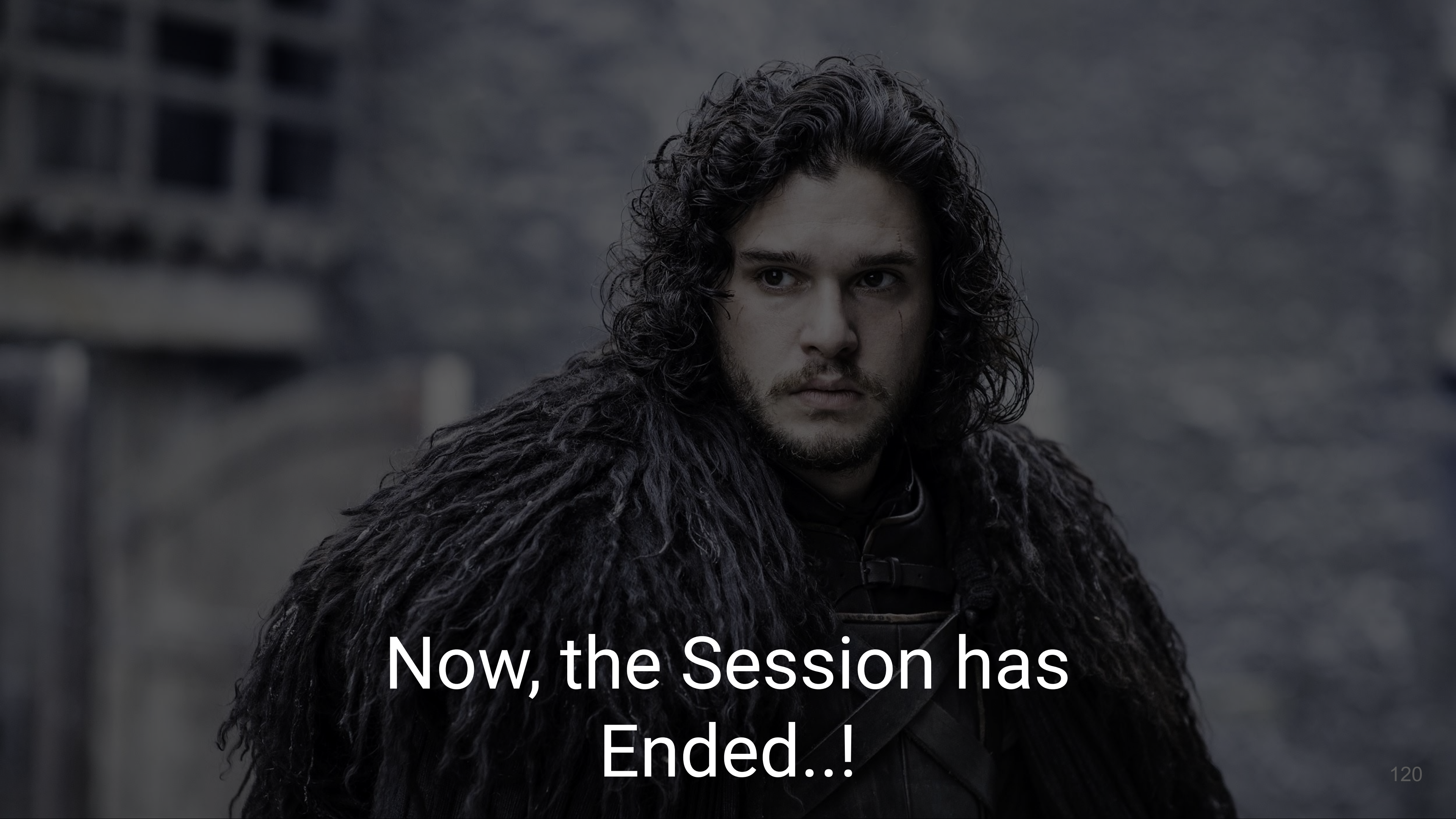
Linear Classifier

A line should classify the data

But, How do I come up with a Line which separates both the classes of data optimally?



Soliton
Vision for a Better World


Now, the Session has
Ended..!


thank you
danke
謝謝
ngiyabonga
teşekkür ederim
gracias
merci
obrigado
bedankt
dziękuję
go raibh maith agat
mochchakkeram
sukriya
kop khun krap
terima kasih
감사합니다
спасибо
рахмат
Баярлалаа
mersi
barka
welalin
tack
spas
dank je
misaotra
matondo
paldies
grazzi
mahalo
tapadh leat
хвала
asante
manana
obrigada
tenki
faafetai lava
vinaka
спасибі
blagodaram
kiitos
dankie
dhanyavad
nanni
nandri
maururu
hvala
köszönöm
bayarlalaa
grazie
enkosi
akun
dankon
djiere dieuf
tau
дякую
mamnun
murakoze
chokrane
chnorakaloutioun
grafias ago
gràcies
sulpáy
taiku
sobodi
děkuji
sagolun
arigatô
takk
dakujem
trugarez
mèsi
ありがとう
najis tuke
tanemirt
rahmet
didi madloba
kam sah hamnida
rahmat
ευχαριστώ
diolch
dhanyavadagalu
shukriya
mercé
мерси
xiexie
তোমাকে ধন্যবাদ