OpenCV-102

BackGround Subtraction

In OpenCV we have 3 algorithms to do this operation –

1) ***BackgroundSubtractorMOG –*** *It is a Gaussian Mixture-based Background/Foreground Segmentation Algorithm.*
2) ***BackgroundSubtractorMOG2 –*** *It is also a Gaussian Mixture-based Background/Foreground Segmentation Algorithm. It provides better adaptability to varying scenes due illumination changes etc.*
3) ***BackgroundSubtractorGMG –*** *This algorithm combines statistical background image estimation and per-pixel Bayesian segmentation.*

Things to remember if you are using GMG :

1) There can be noise while using GMG ,so it is better to use morphological operation such as OPENING for noise removal as discussed in opencv_101

How to use Background Subtraction in OpenCV

Step #1 – Create an object to signify the algorithm we are using for background subtraction.
Step #2 – Apply backgroundsubtractor.apply() function on image.

MeanShift

1) Every instance of video is checked in the form of pixel distribution in that frame.
2) We define the initial window,generally a square or a rectangle for which the positions are specifies and the algorithm tracks the area of maximum pixel distribution .
3) So when we run this, it will keep shifting towards the area of maximum pixel distribution .
4) The direction of movement depends upon the difference between the center of our tracking window and the centroid of all the k-pixels inside that window .

Cons:

1) Size of the tracking window remains same irrespective of distance of object from the camera.

2) Windows will keep the track of object if there exists a object in that region.Wo we need to hardcode this very carefully.

CamShift

Camshaft -> Continuously Adaptive Mean Shift

Enhanced version of mean Shift which provides more accuracy and robustness of the model.

Size of the window keeps updating when tracking windows tries to converge.

Tracking is done by using the color information of the object.

It actually first applies mean Shift and then updates the size of window.

Optical Flow

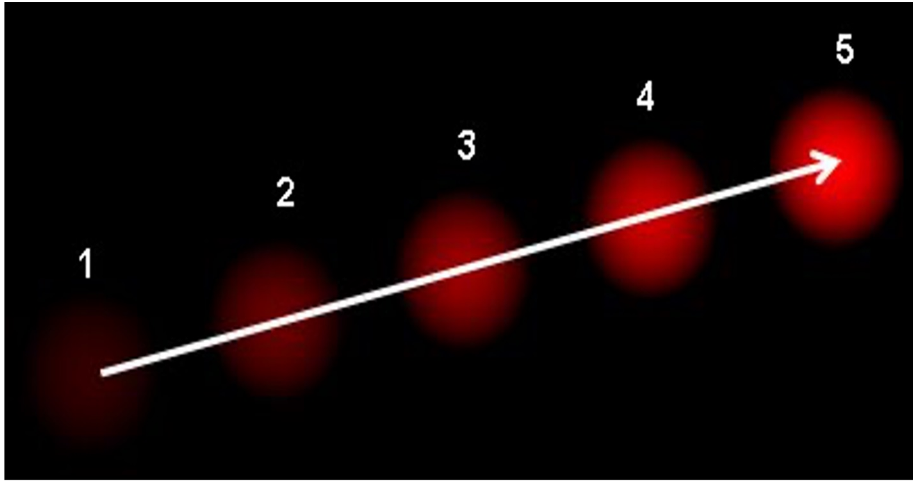
It is a task if per pixel motion estimation

Between two consecutive frames in one

video. It implies the calculation of shift

Vector for pixel as an object displacement

Difference between two neighboring .



Idea is to estimate the object’s displacement vector caused by it’s motion or camera movements.

Uses: Compression,stabilization ,slow motion etc in video editing

Also it can be used in action recognition tasks and real time tracking systems.

Lucas-Kanade Algorithm

> Used with sparse feature set ,computes the motion vector for the specific set of object (detected corners on image) .

> Using only a sparse feature set means that we will not have motion information about pixels that are not contained in it .

> This restriction can be lifted by using Dense optical flow ,which we will discuss in next video.

Dense Optical Flow

> In Dense optical Flow we calculate the motion vector for every pixel in the image.

> Farneback algorithm requires 1-dimension input image,so we convert the BRG image to gray scale.

> In dense optical flow, we look at all of the points(unlike Lucas Kanade which works only on corner points detected) and detect the pixel intensity changes between the two frames, resulting in an image with highlighted pixels, after converting to hsv format for clear visibility.

> It computes the magnitude and direction of optical flow from an array of the flow vectors, i.e., *(dx/dt, dy/dt)*. Later it visualizes the angle (direction) of flow by hue and the distance (magnitude) of flow by value of HSV color representation.For visibility to be optimal, strength of HSV is set to 255. OpenCV provides a function **cv2.calcOpticalFlowFarneback** to look into dense optical flow.

HDR Imaging(High Dynamic Range)

> Most digital images and devices used 8 bits per channel,thus limiting the dynamic range of device to two orders of magnitude .

> When we take photographs of a real world scene,bright regions may be overexposed and the dark ones may be underexposed ,so we can’t capture all details using a single exposure .

> HDR imaging works with images that use more than 8 bits per channel.

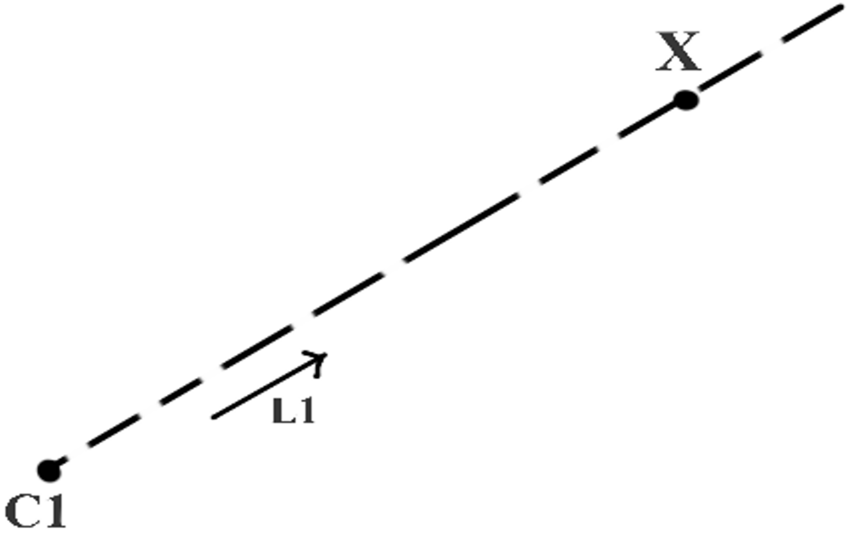
> The way to take HDR images ,is to be use photographs of the scene taken with different exposure values.

> Tonemapping : After the conversion of HDR image, we need to convert that to 8-bit view to view on usual displays.That is called as Tonemapping.

Epipolar Geometry

> Do we need more than 1 image to calculate depth ?

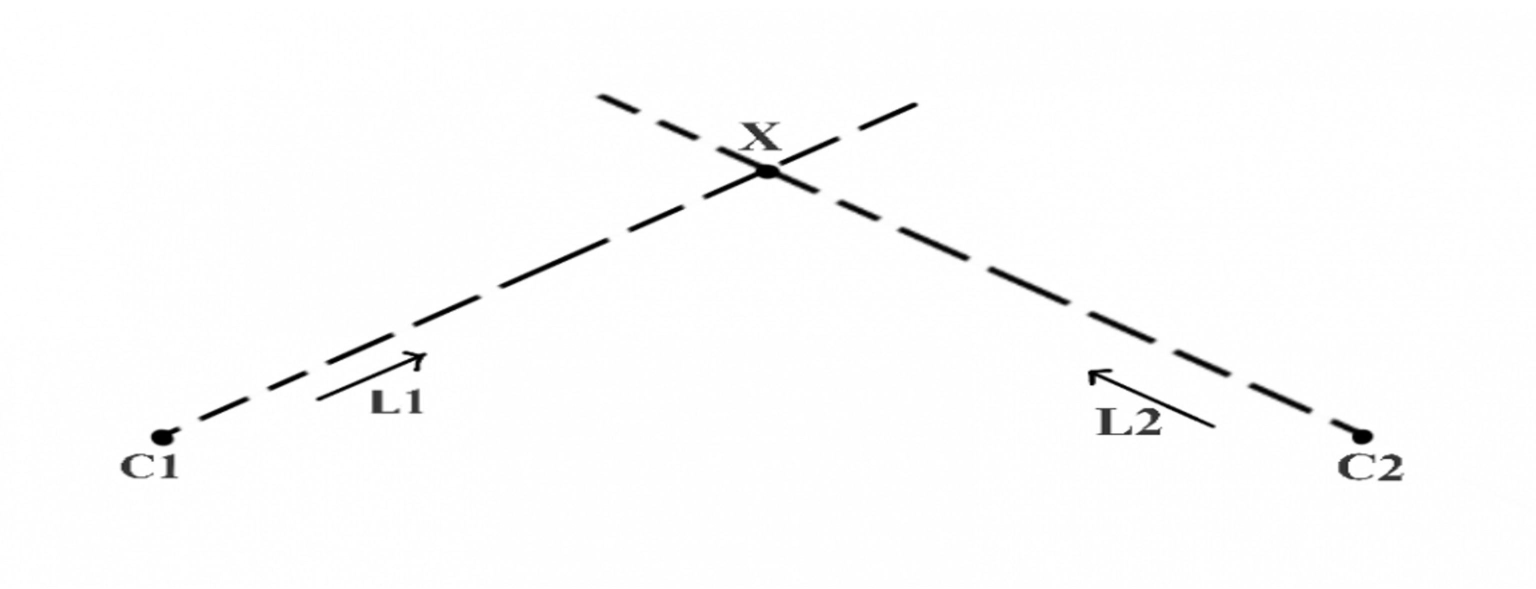
> If so,why ?



C1 and X are points in 3D space, and the unit vector L1 gives the direction of the ray from C1 through X. Now, can we find X if we know the values of point C1 and direction vector L1? Mathematically it simply means to solve for X in the equation

$$X = C1 + k(L1)$$

Now, as the value of k is not known, we cannot find a unique value of X

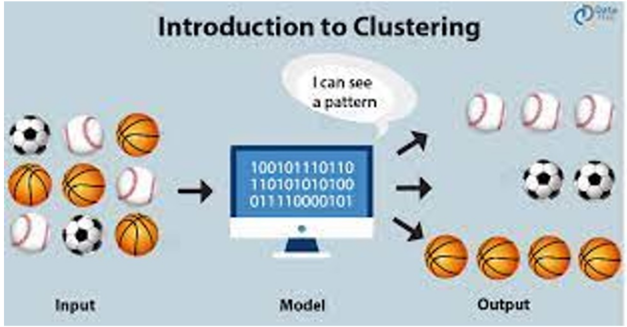


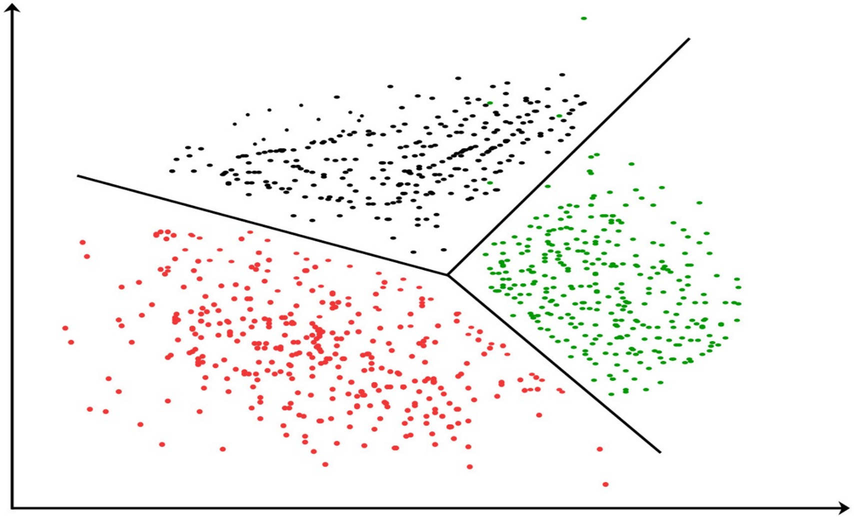
Locating a 3D point (X), at an unknown depth, with two known 3D points (C1 and C2) and direction vectors (L1 and L2) – Triangulation.

Yes! Because the rays originating from C1 and C2 clearly intersect at a unique point, point X itself. This is called triangulation. We say we triangulated point X.

Clustering

Clustering is one of the most common exploratory data analysis technique used to get an intuition about the structure of the data. It can be defined as the task of identifying subgroups in the data such that data points in the same subgroup (cluster) are very similar while data points in different clusters are very different.





K-means Clustering

Step 1: Choose the number of clusters *k*

Step 2: Select k random points from the data as centroids

Step 3: Assign all the points to the closest cluster centroid

Step 4: Recompute the centroids of newly formed clusters

Step 5: Repeat steps 3 and 4

Stopping Criteria for K-Means Clustering

There are essentially three stopping criteria that can be adopted to stop the K-means algorithm:

1. Centroids of newly formed clusters do not change
2. Points remain in the same cluster
3. Maximum number of iterations are reached