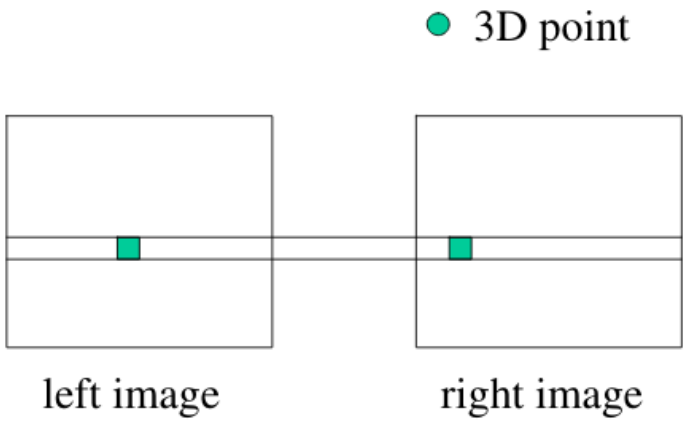
CSE463 - Vision

In stereo vision, the two images, captured by the two cameras separated by a distance, can be used to get the 3d location (x, y, z) of the points of images in real world i.e. the depth -- z location in addition to 2d -- x and y location.

The **disparity** is the difference in image location of the same 3D point when projected under perspective to two different cameras.

> Any point in the scene that is visible in both cameras will be projected to a pair of image points in the two images, called a conjugate pair. The displacement between the positions of the two points is called the disparity.



The disparity map/image is simply the image that given where each pixel gives the disparity of that 3d point.

The **depth** (the actual z location of 3d point) can be calculated by using the disparity of the corresponding point e.g. in simple cases, as follows:

```
depth = (baseline * focal length) / disparity)
```

where `baseline` is the distance b/w the cameras.

By getting the depth of every pixel, you get the depth map/image.

STEREO MATCHING EXAMPLES

https://github.com/eborboihuc/stereo-matching

https://github.com/erget/StereoVision/blob/master/stereovision/blockmatchers.py

---

Rotation matrix in stereo → 3x3 x,y,z/x,y,z/x,y,z

Correspondence için aynı doğru üzerinde + fundamental matrix

https://youtu.be/GXysgyXgRSY?t=9420 fundamental matrix correspondence

fundamental matrix diğer noktayı arayacağımız doğruyu bize söyler.

denklik = correspondence

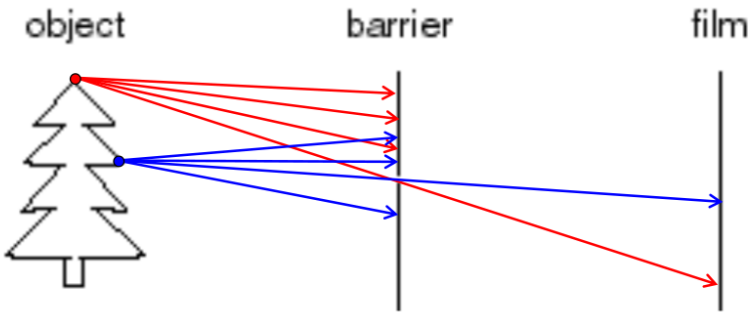
Keypointteki pikselleri kendi aralarında karşılaştırıp ordan depth map ve dispairty çıkar.

NOTES

Digimask → put your face on 3d avatar

Range image sana yakın olan daha koyu renk

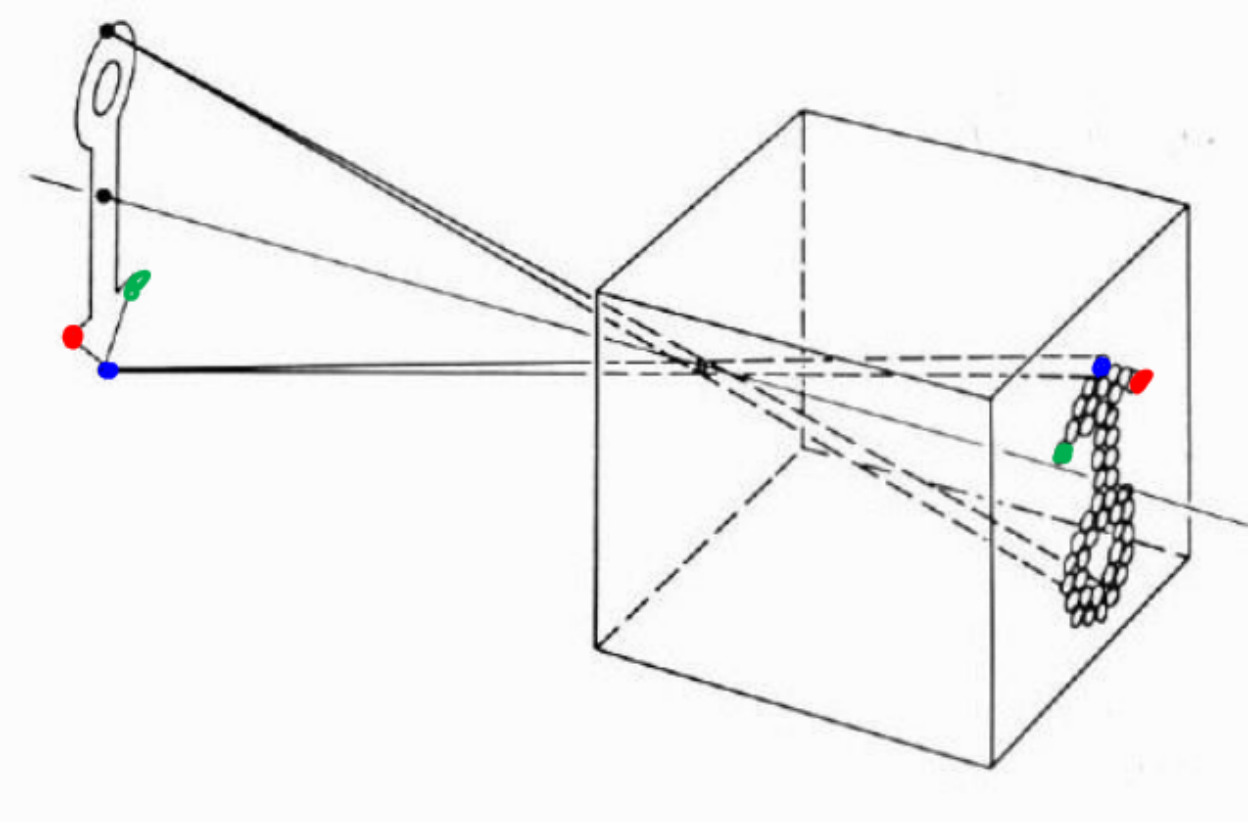
Pinhole camera



Add a barrier to block off most of the rays
- This reduces blurring
- The opening known as the **aperture**

aperture = açıklık

pupil = gözbebeği



1/focus = 1/distO + 1/distİ

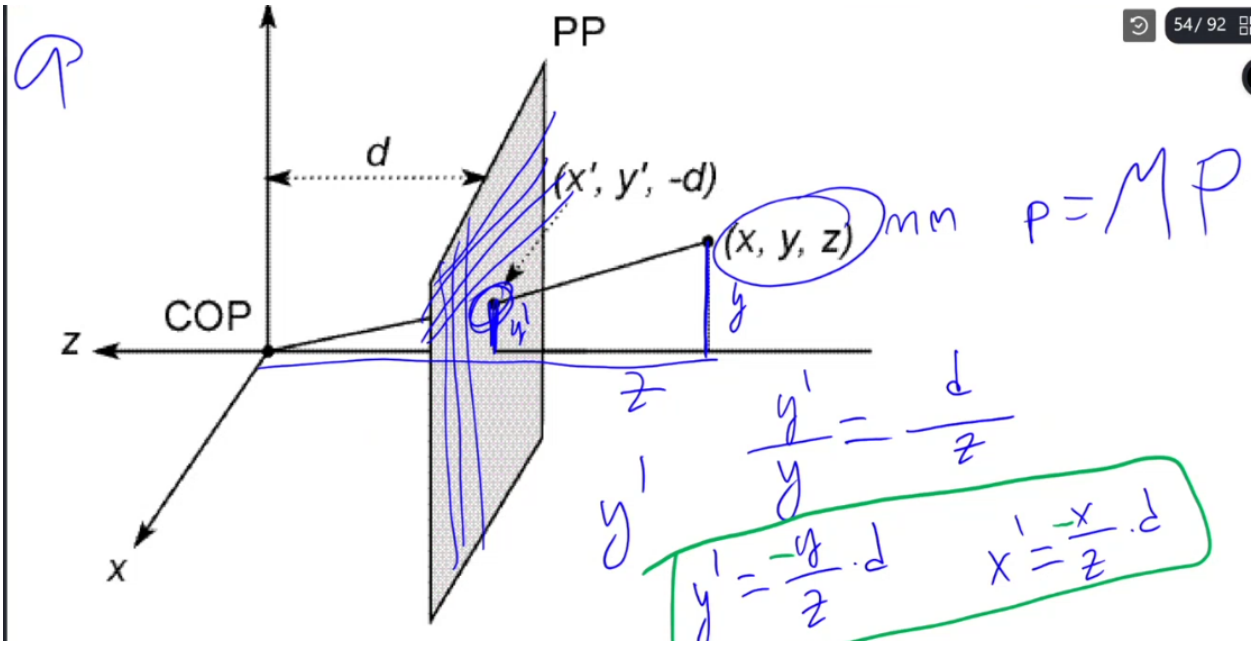
focal = odak

When intensity of lights reflected from objects is high → cone vision(else rod vision)

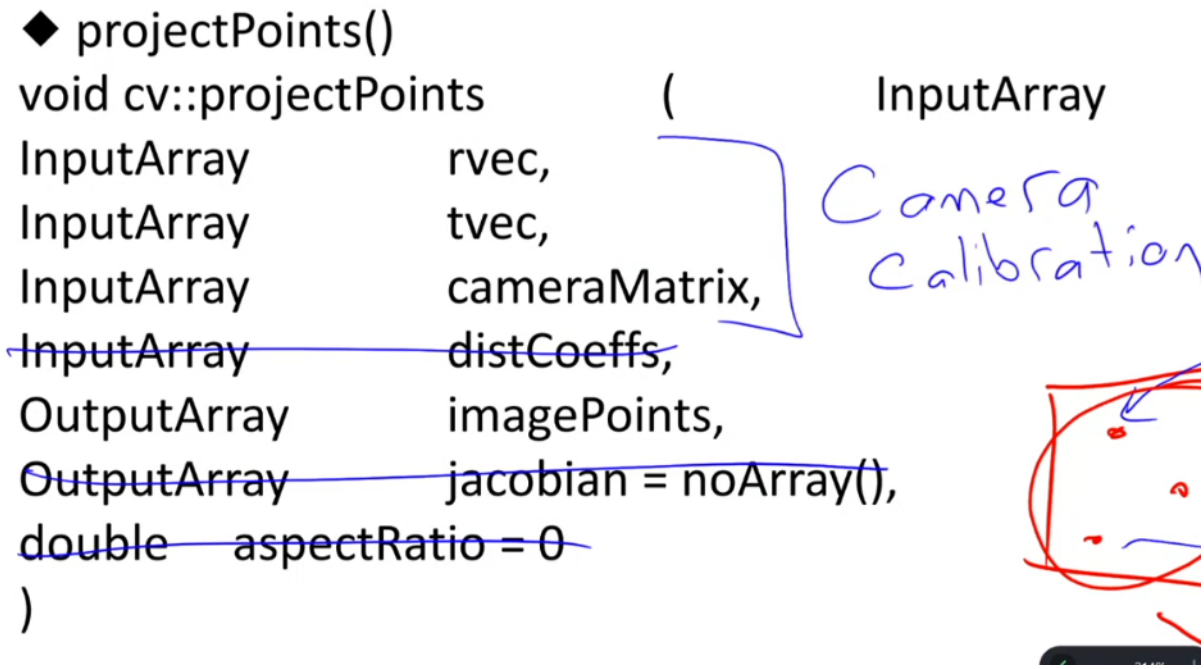
Put optical center at the origin. Image plane in front of center of projection.

r = non homojen | h = homojen

telecentric lens = make distance same



p = MP.We will multiply projection matrix M with P, and will get p in terms of image coordinates.



process of recovering these parameters is called → camera calibration.

$p_{1...10} = M.P_{1...10}$

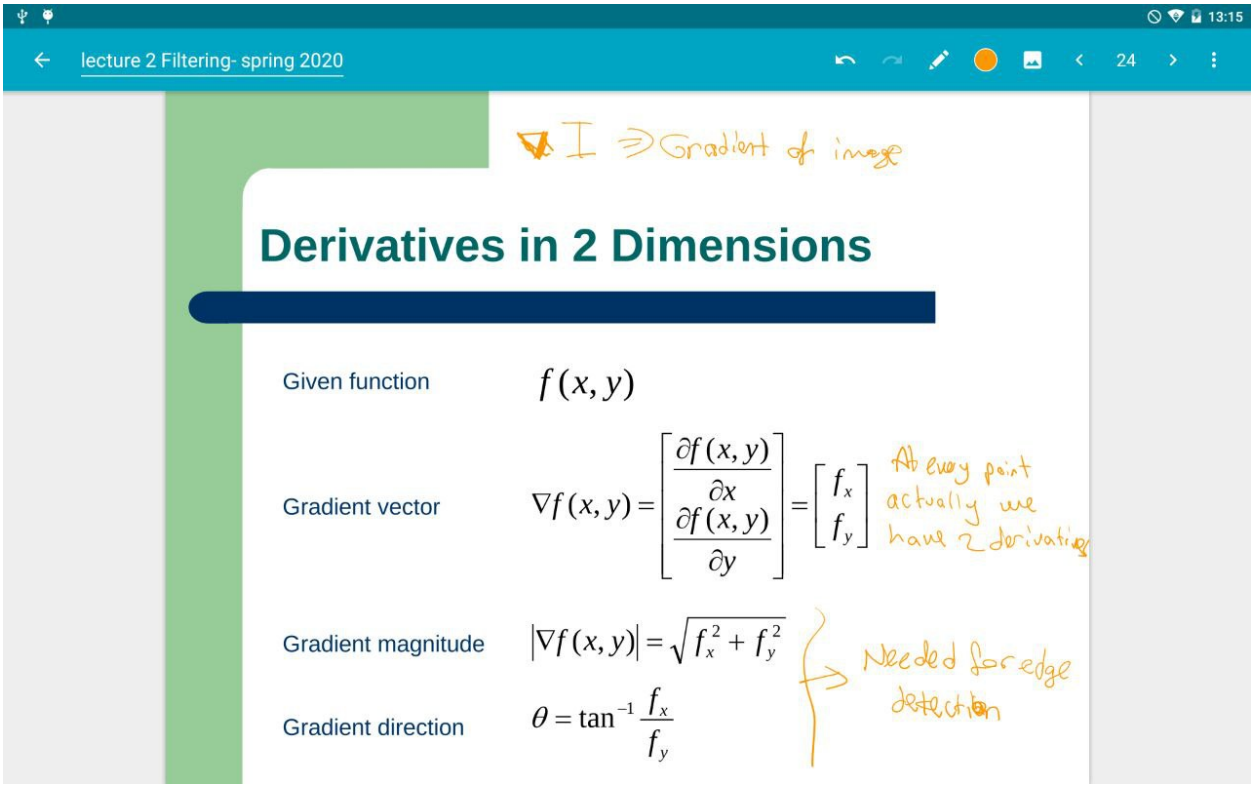
M contains 12 elements. P=3D, p=2d

barrel and cushion(yastık) distortion

Extrinsic parameters → depend on real world objects

Intrinsic parameters → They don't depend on where my camera is in real world

x'e göre kısmı türev [partial derivative] f_x X dışındakileri sabit kabul et



1 dimensional gaussian → $g(x) = e^{\frac{-x^2}{2\sigma^2}}$

2d gaussian $g(x) = e^{\frac{-(x^2+y^2)}{2\sigma^2}}$

when σ is increases, image is more blurred and noise is more supressed.

türev sayesinde kenar bulma

EDGE → Sudden changes in an image intensity [not because object borders]

For edge detection you need to find sudden changes and we need to use DERIVATIVE for this

Filters are linear operations applied on images

when derivative makes a peak, we know that there is an edge

2.türevin x ekseninden geçtiği yer [0 crossings] tepe noktasını verir (peak positions)

To find edges, look for peaks in $\frac{d(f*g)}{dx} = f * \frac{d}{dx} g$.

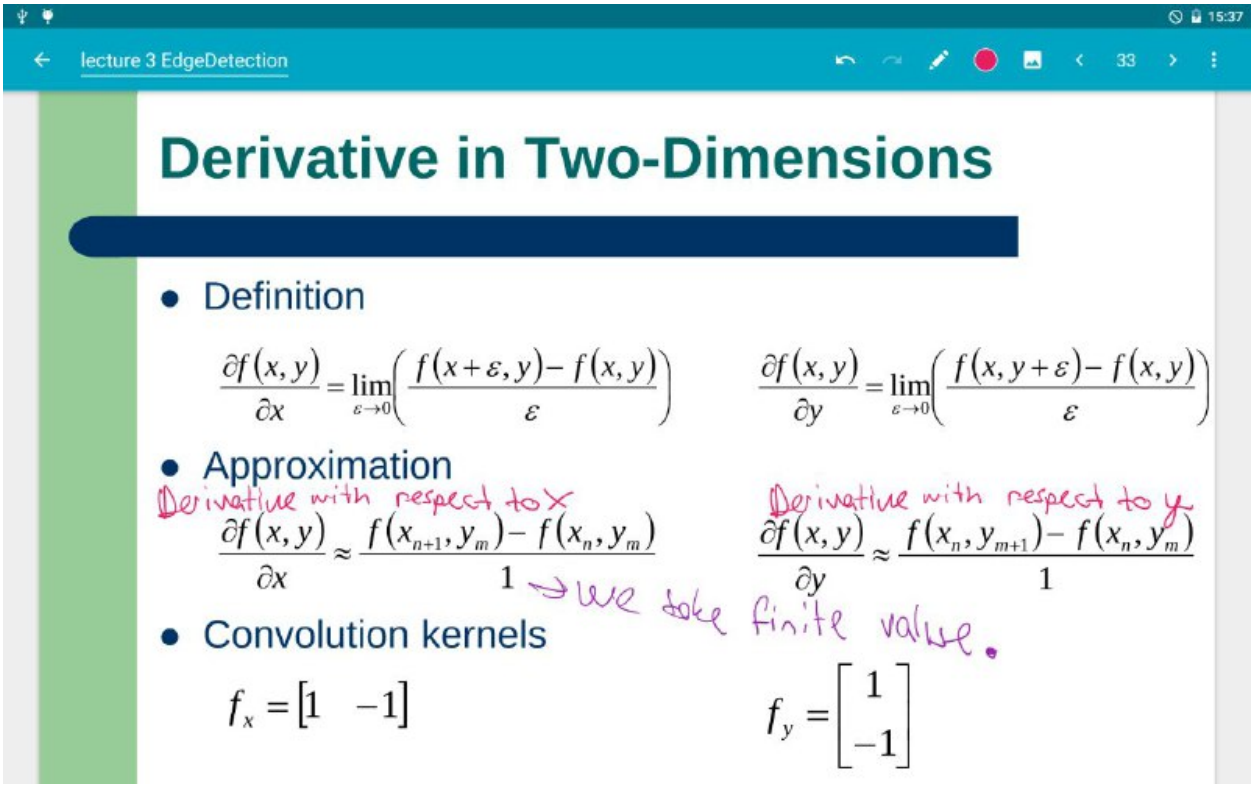


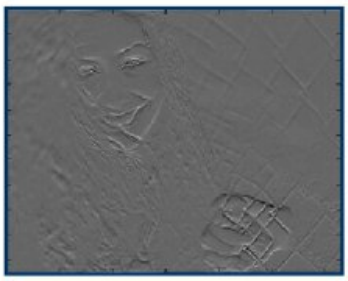
Image I

$$I_x = I * \begin{bmatrix} 1 & -1 \end{bmatrix}$$

Sensitive to the changes in X direction

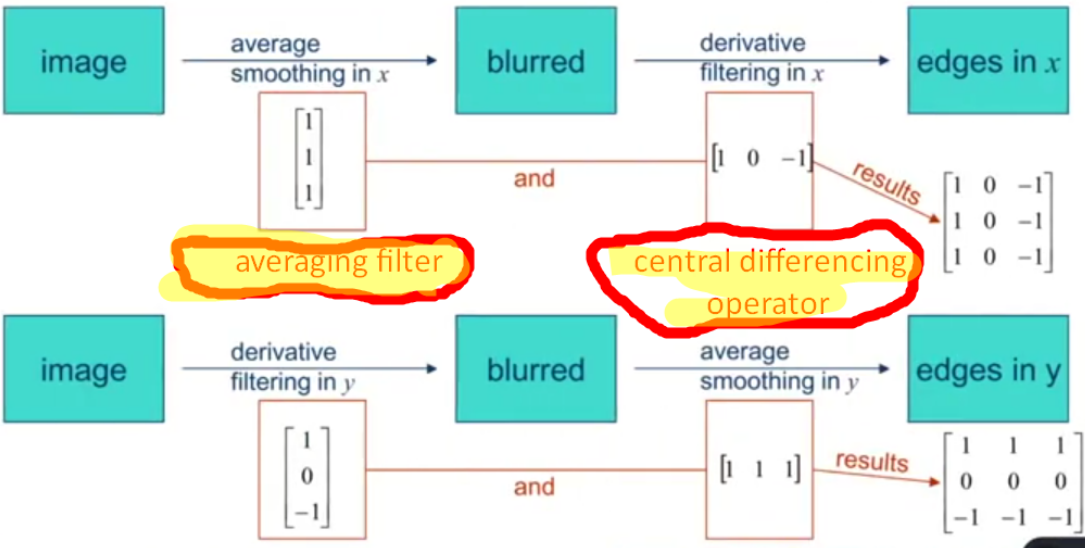
$$I_y = I * \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Sensitive to the changes in Y direction



Alper Yilmaz, Mubarak Shah Fall 2012UCF

Prewitt Edge Detector



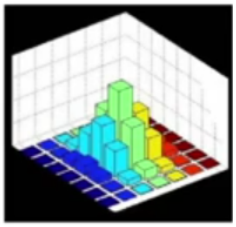
Alper Yilmaz, Mubarak Shah Fall 201

For sobel edge detector, instead [1,1,1] it's [1,2,1]

laplacian is second derivative of the gauss

- Gaussian smoothing

$$\underbrace{S}_{\text{smoothed image}} = \underbrace{g}_{\text{Gaussian filter}} * \underbrace{I}_{\text{image}} \qquad g = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



- Find Laplacian

$$\Delta^2 S = \underbrace{\frac{\partial^2}{\partial x^2} S}_{\text{second order derivative in } x} + \underbrace{\frac{\partial^2}{\partial y^2} S}_{\text{second order derivative in } y}$$

- ∇ is used for gradient (first derivative)
- Δ^2 is used for Laplacian (Second derivative)

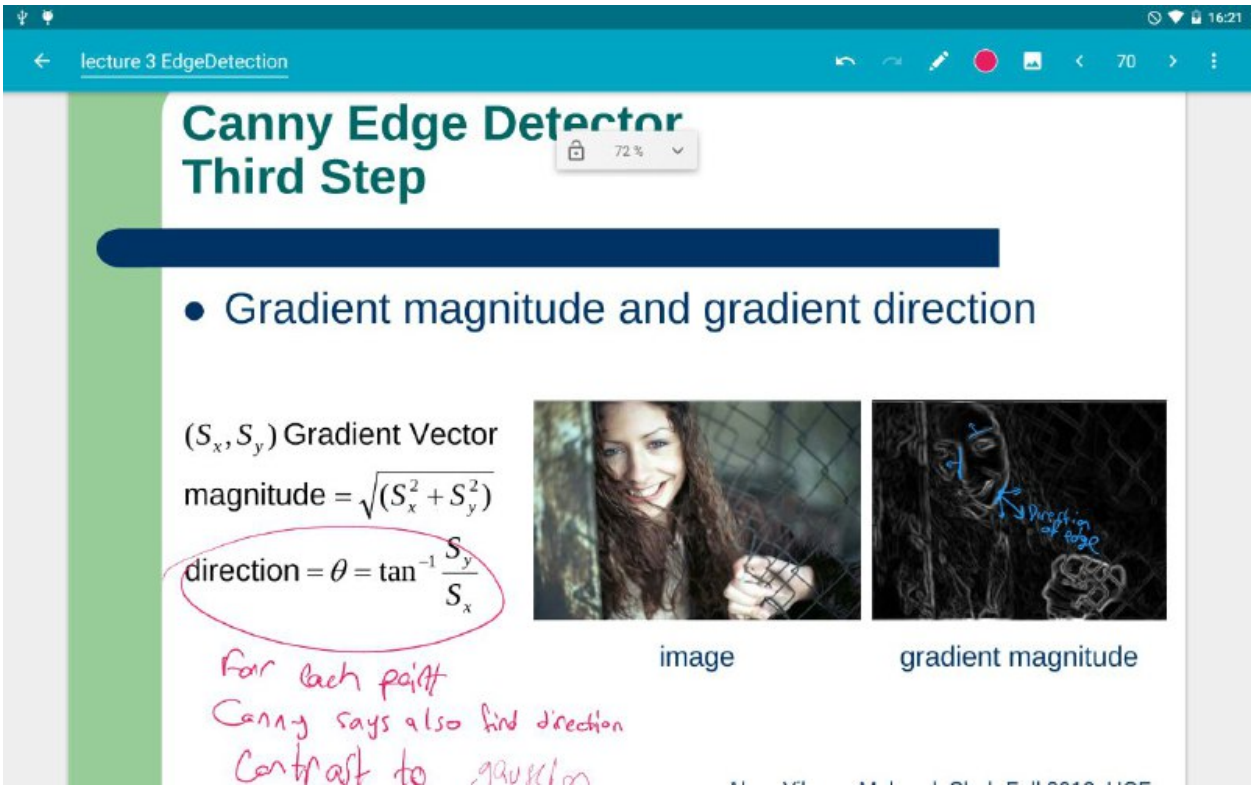
Alper Yilmaz, Mubarak Shah Fall 201

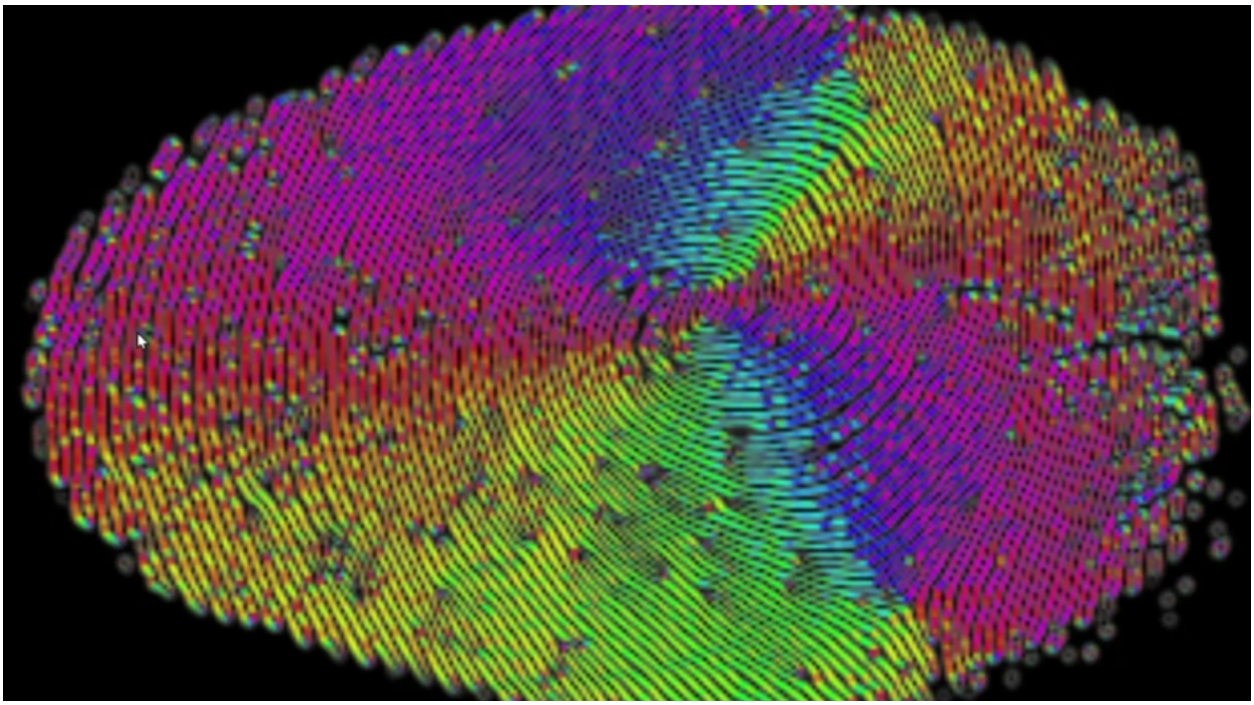
taking second derivative creates more noise in the image

Taking second derivative may not be the best option. So CANNY says take first derivative do rest yourself

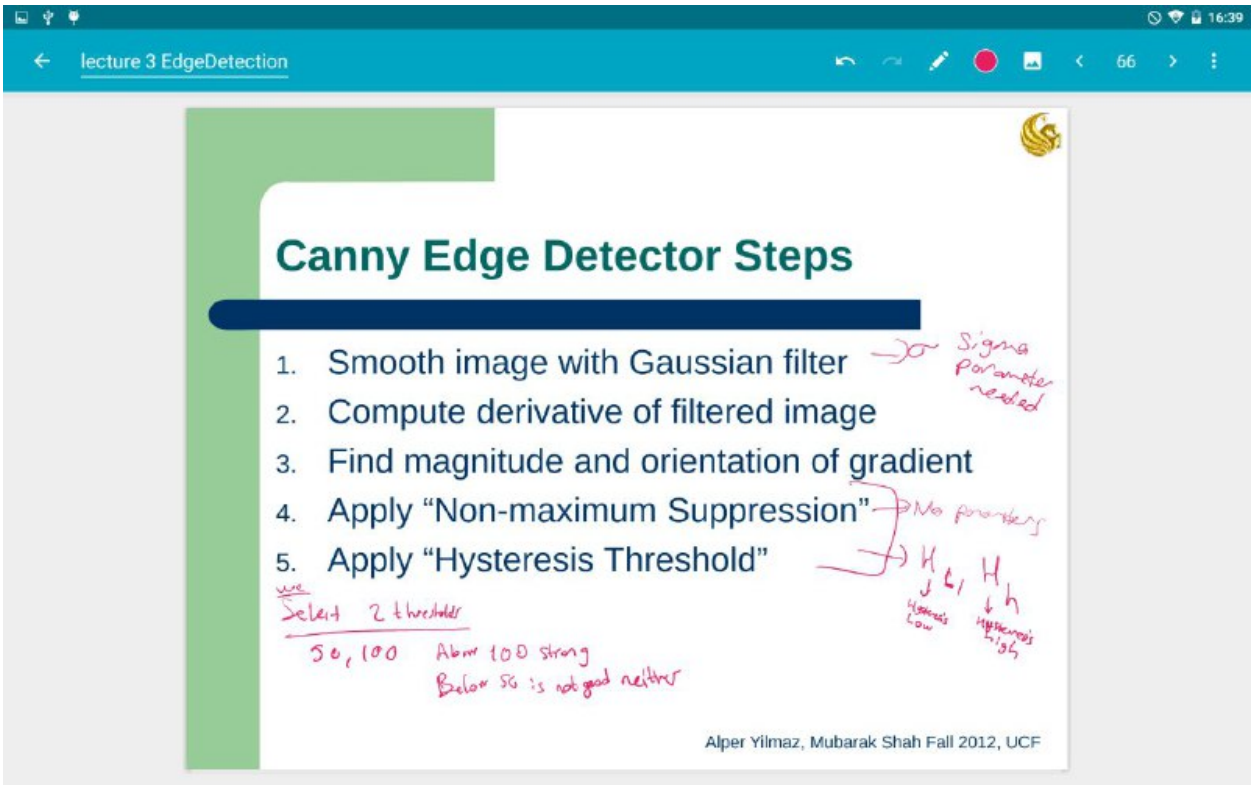
canny→ using 2 thresholds for ex. 50 and 100. above 100 is strong but also take above 50

consider neighbors iteratively and if it's connected declare it as edge pixel.





color shows orientation in each pixel



distinctive corner detectors are robust against rotation, lighting changes, scale changes

thats an edge because when you move, changes

Moving:

$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u, y+v) - I(x,y)\right]^2$$

Window function

Shifted intensity

Intensity

E(0,0) is 0

E(1,1) Moving down and right

Taylor Series Expansion

Taylor's Theorem : Suppose *f* is continuous on the closed interval [a, b] and has *n + 1* continuous derivatives on the open interval (a, b). If *x* and *c* are points in (a, b), then

The Taylor series expansion of $f(x)$ about c:

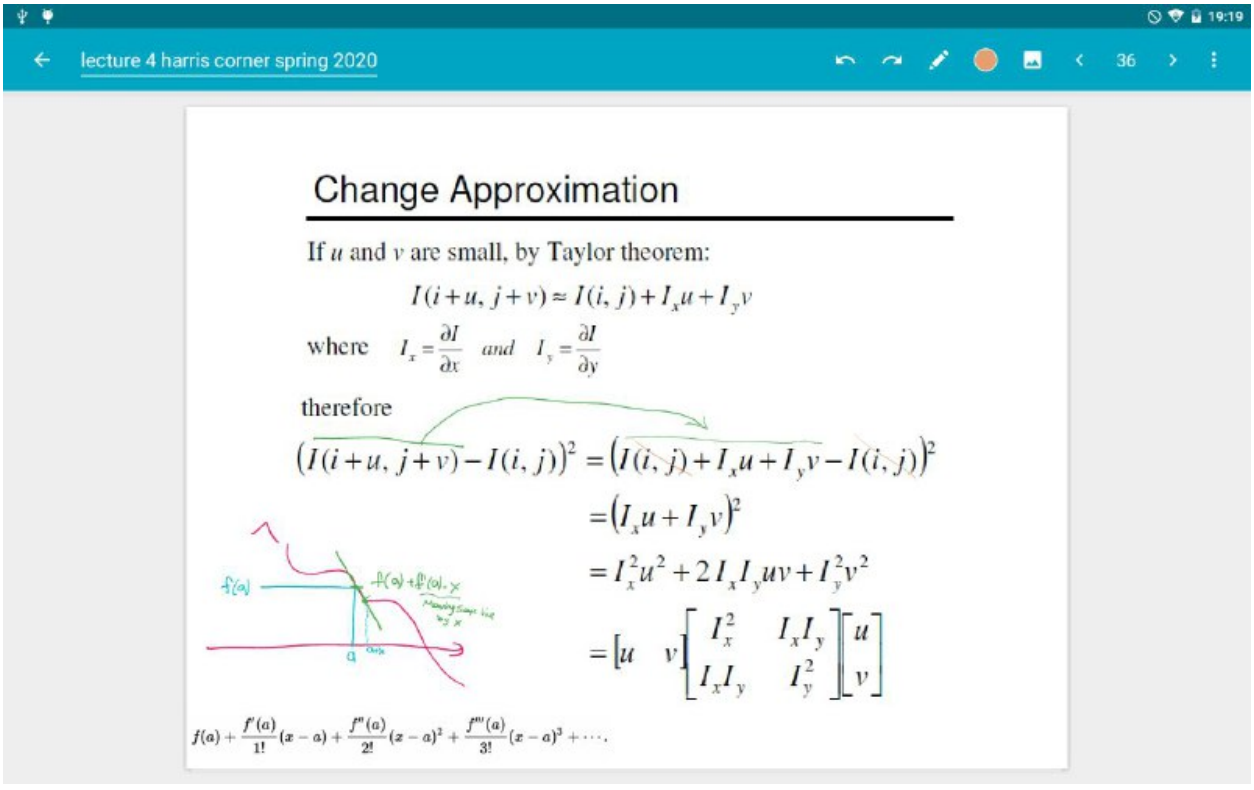
$$f(c)+f'(c)(x-c)+\frac{f^{(2)}(c)}{2!}(x-c)^2+\frac{f^{(3)}(c)}{3!}(x-c)^3+\ldots$$

or

$$Taylor\ Series=\sum_{k=0}^{\infty}\frac{1}{k!}f^{(k)}(c)\,(x-c)^k$$

If the series converge, we can write:

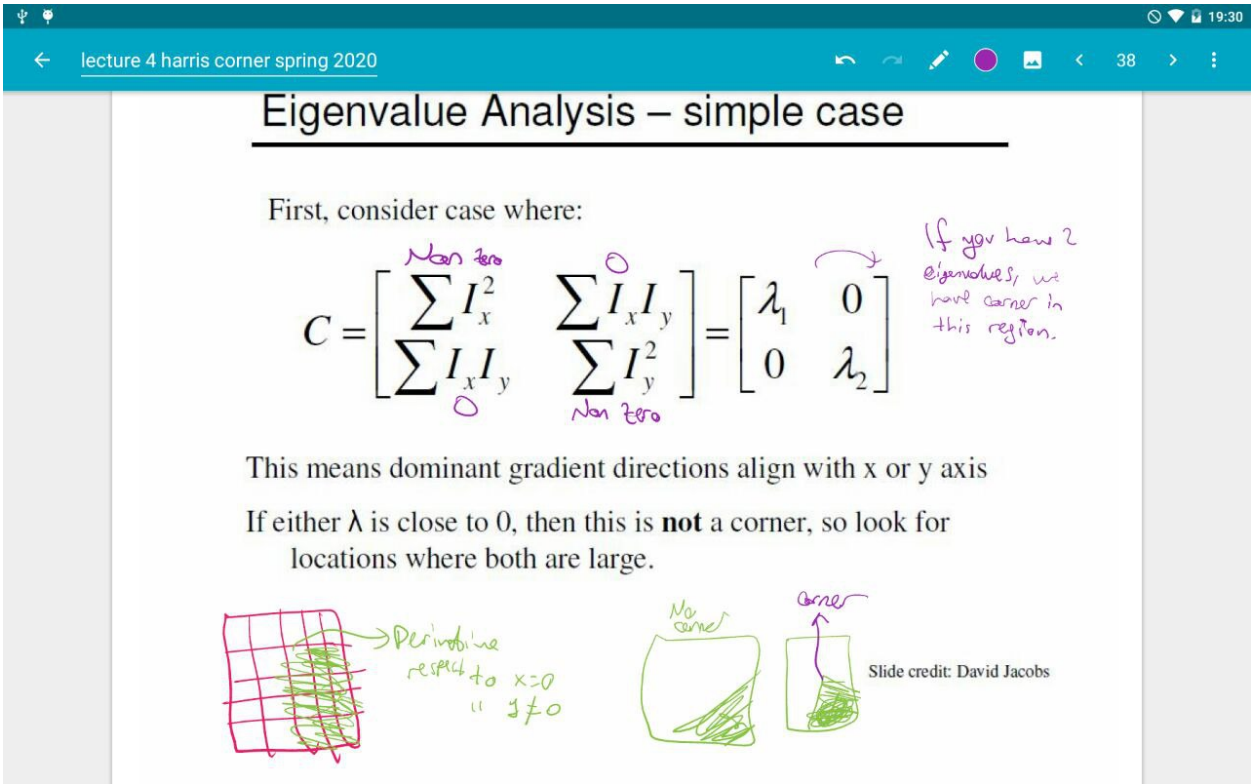
$$f(x)=\sum_{k=0}^{\infty}\frac{1}{k!}f^{(k)}(c)\,(x-c)^k$$



$Ax = \lambda x$ | x is such a dimension/vector that when its multiplied by matrix A, X direction stays same.

A - λI = 0

*****



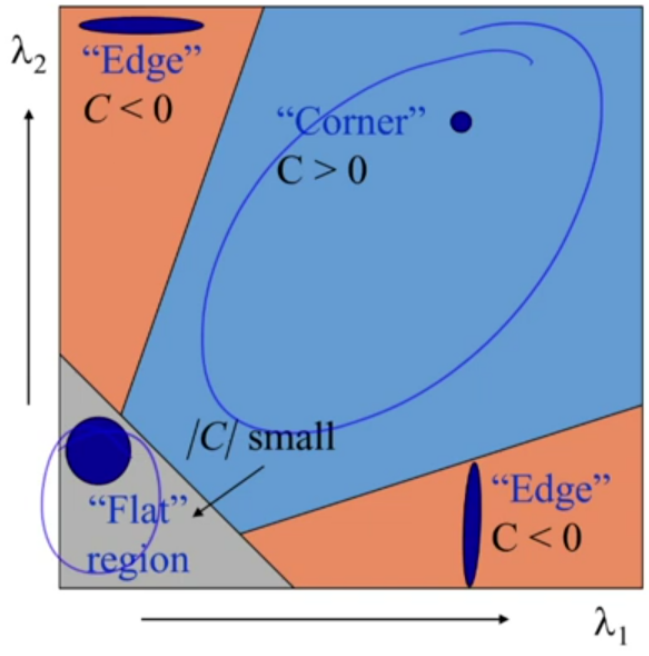
if we have 2 special direction we have corner; one→edge

If no lambda be found then there is no corner . its a flat region

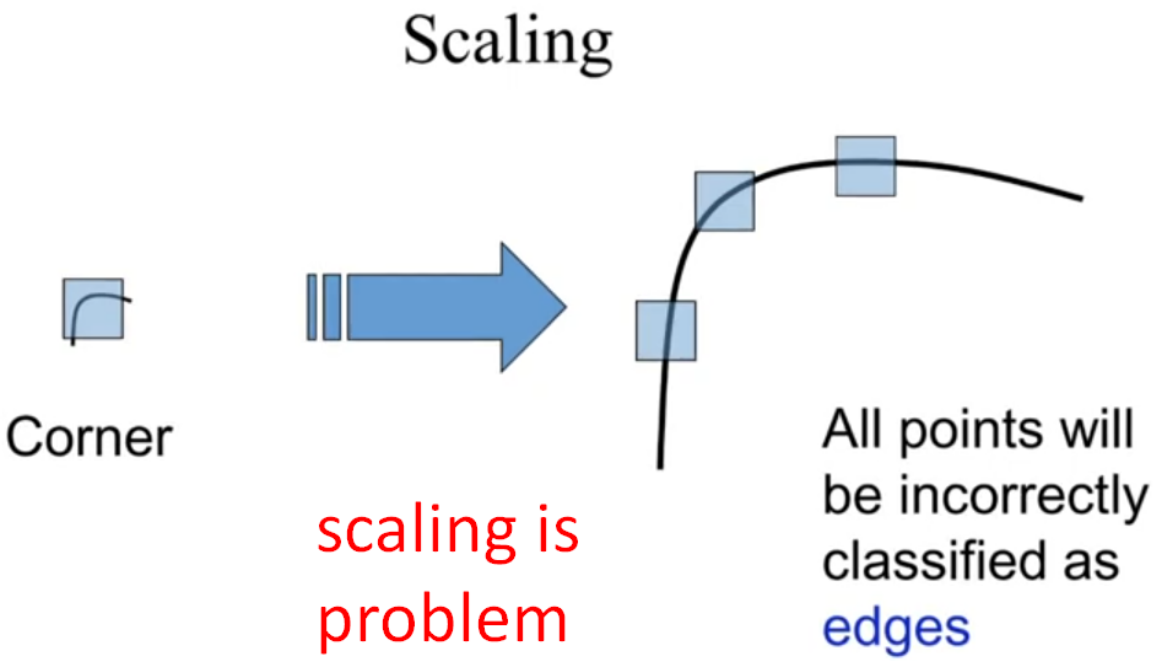
Cornerness score:

$C = \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$

α: some constant (~0.04 to 0.06)



cornerlık hesaplama = Lambda1 * lambda2 - alfa(lambda1 + lambda2)^2

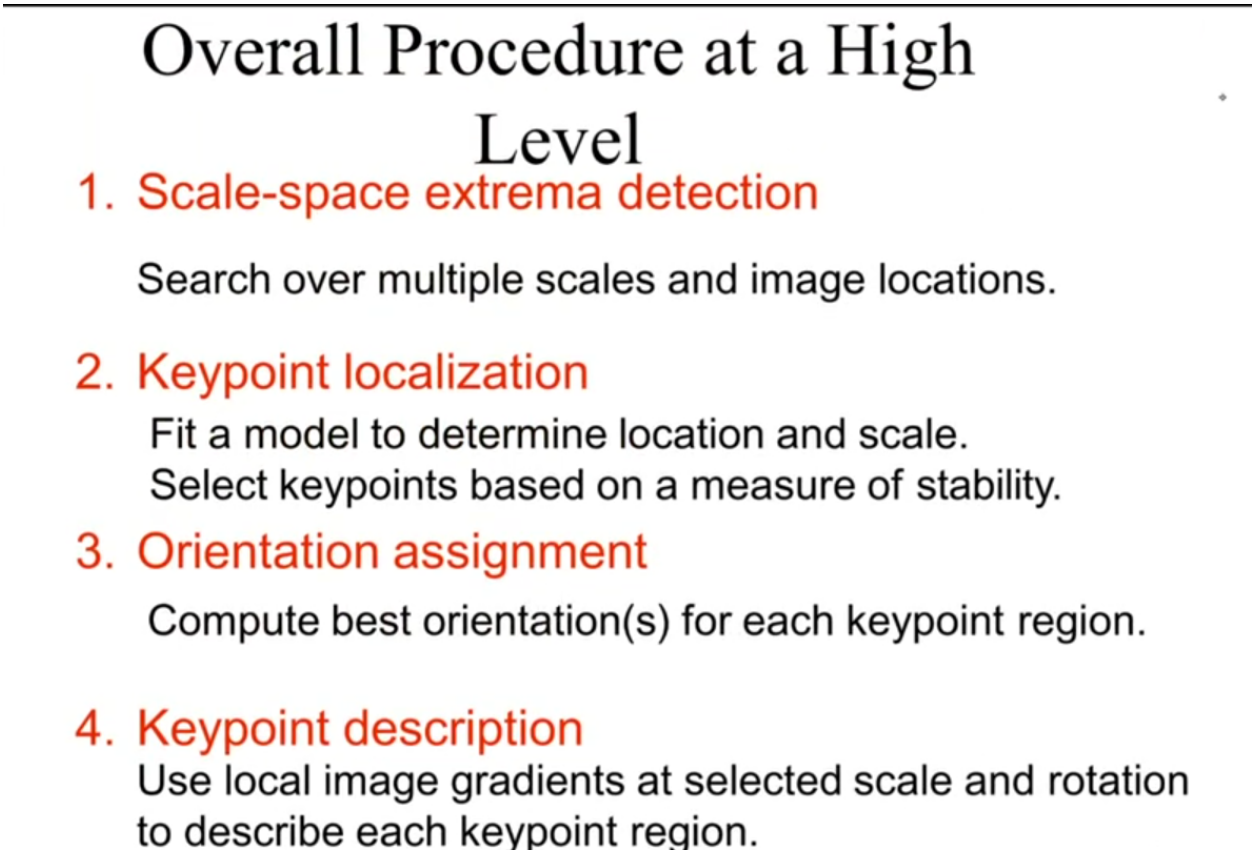


for ex.: a 128d **descriptor** can be used because robust to variations corresponding to typical ciewing conditions.

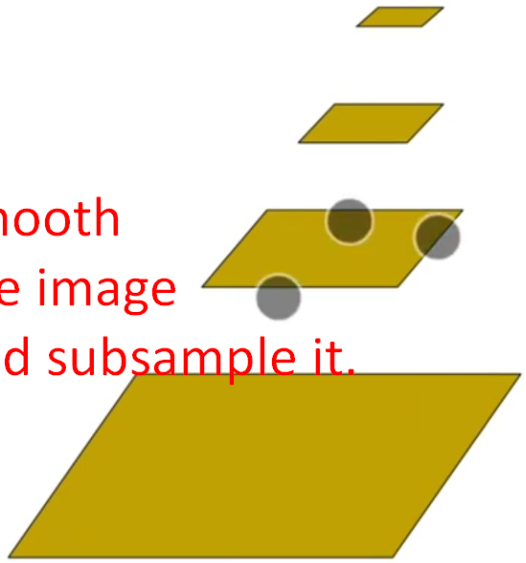
Idea of SIFT → Scale Invariant [harris is not]

with descriptor you can Match points in different scale images

1 scale independence



Pyramids



smooth
the image
and subsample it.

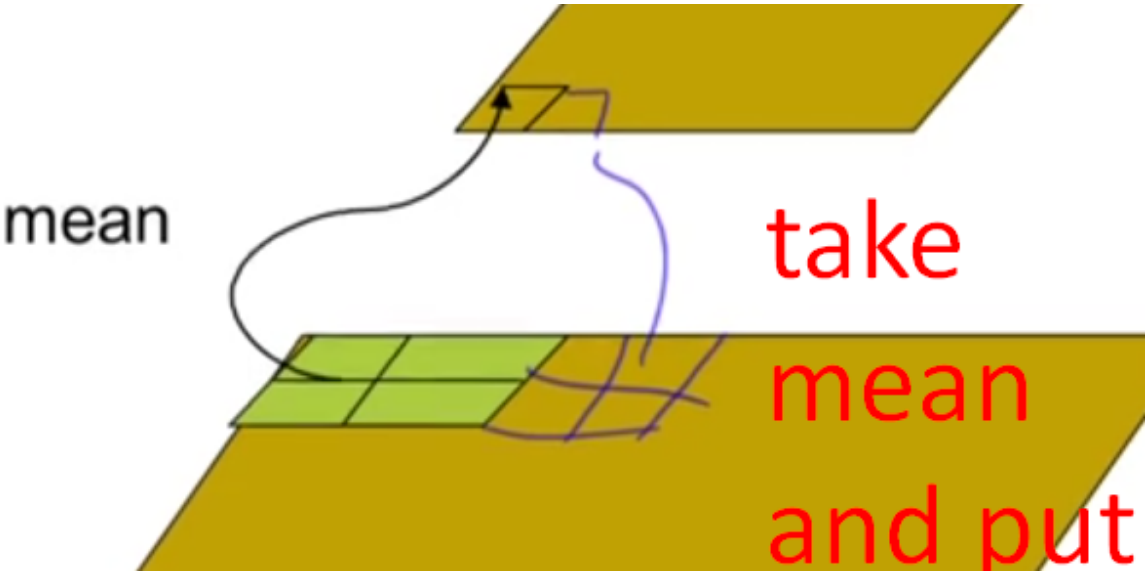
And so on.

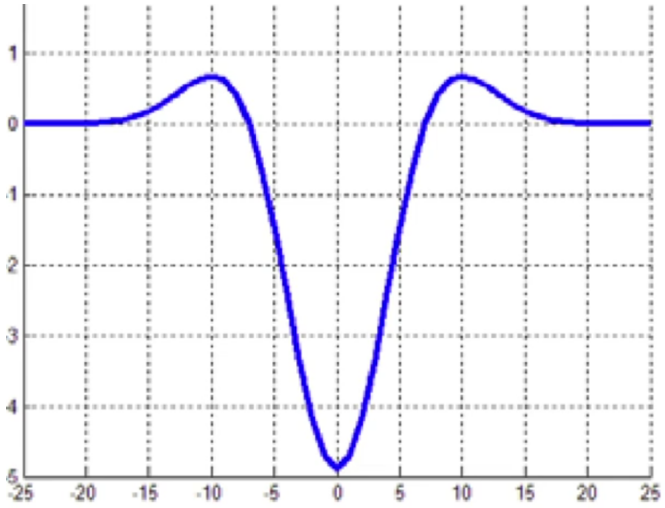
3rd level is derived from the 2nd level according to the same funtion

2nd level is derived from the original image according to some function

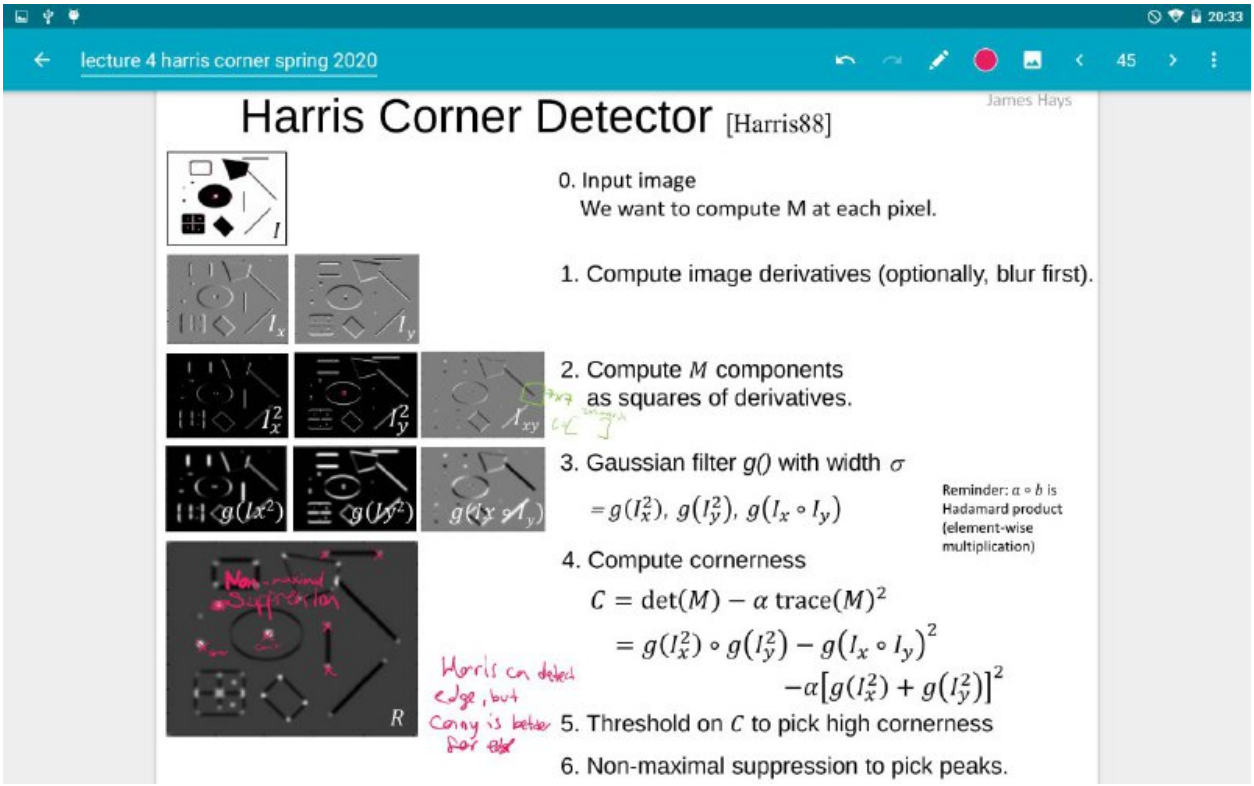
Bottom level is the original image.

11/1/2010



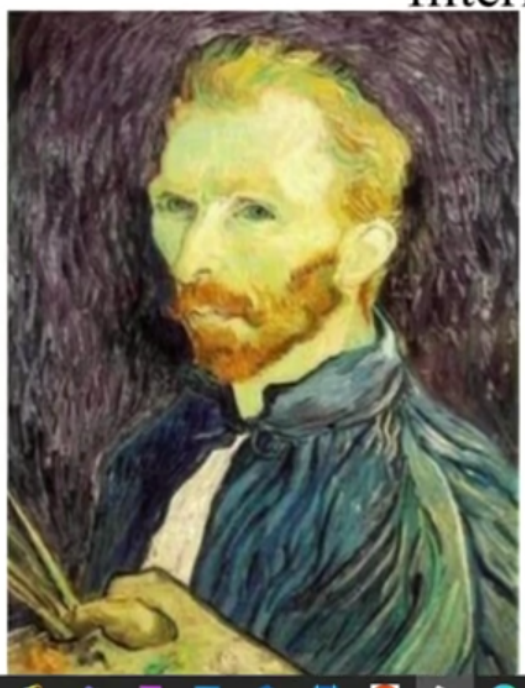


Mexican hat: second derivstive of gaussian



occlusion → engelleme / tıkama

Example: Subsampling with Gaussian pre-filtering



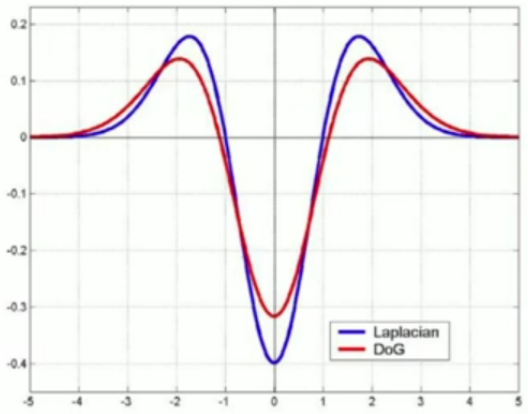
G 1/4

G 1/8

first→ apply gaussian and then take average of and put that position

57/

Lowe's Scale-space Interest Points: Difference of Gaussians



- Gaussian is an ad hoc solution of heat diffusion equation

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G.$$

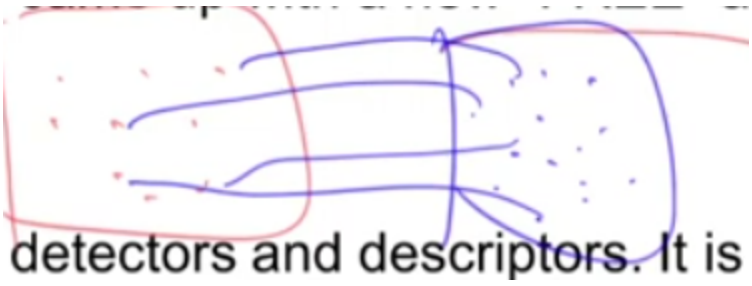
- Hence

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma^2 \nabla^2 G.$$

- k is not necessarily very small in practice

Instead of taking laplacian(second derivative), for k>1 substract Gaussians aobve and it'S similar with laplacian.

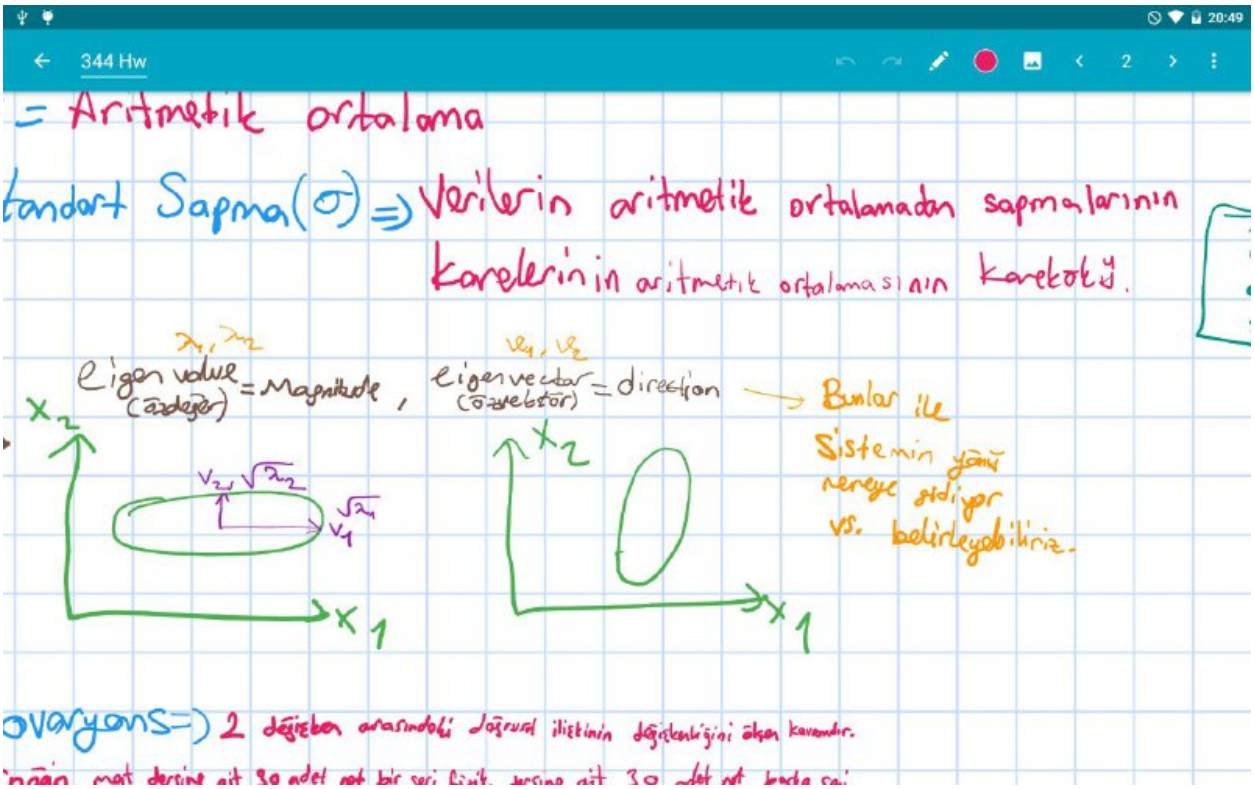
```cpp
//-- Step 1: Detect the keypoints using SURF Detector, compute the descriptors
int minHessian = 400;
Ptr<SURF> detector = SURF::create( minHessian );
std::vector<KeyPoint> keypoints1, keypoints2;
Mat descriptors1, descriptors2;
detector->detectAndCompute( img1, noArray(), keypoints1, descriptors1 );
detector->detectAndCompute( img2, noArray(), keypoints2, descriptors2 );

//-- Step 2: Matching descriptor vectors with a FLANN based matcher
// Since SURF is a floating-point descriptor NORM_L2 is used
Ptr<DescriptorMatcher> matcher = DescriptorMatcher::create(DescriptorMatcher::FLANNBASED);
std::vector< std::vector<DMatch> > knn_matches;
matcher->knnMatch( descriptors1, descriptors2, knn_matches, 2 );

//-- Filter matches using the Lowe's ratio test
const float ratio_thresh = 0.7f;
std::vector<DMatch> good_matches;
for (size_t i = 0; i < knn_matches.size(); i++)
{
    if (knn_matches[i][0].distance < ratio_thresh * knn_matches[i][1].distance)
    {
        good_matches.push_back(knn_matches[i][0]);
    }
}

//-- Draw matches
Mat img_matches;
drawMatches( img1, keypoints1, img2, keypoints2, good_matches, img_matches, Scalar::all(-1),
             Scalar::all(-1), std::vector<char>(), DrawMatchesFlags::NOT_DRAW_SINGLE_POINTS );

//-- Show detected matches
imshow("Good Matches", img_matches );
```

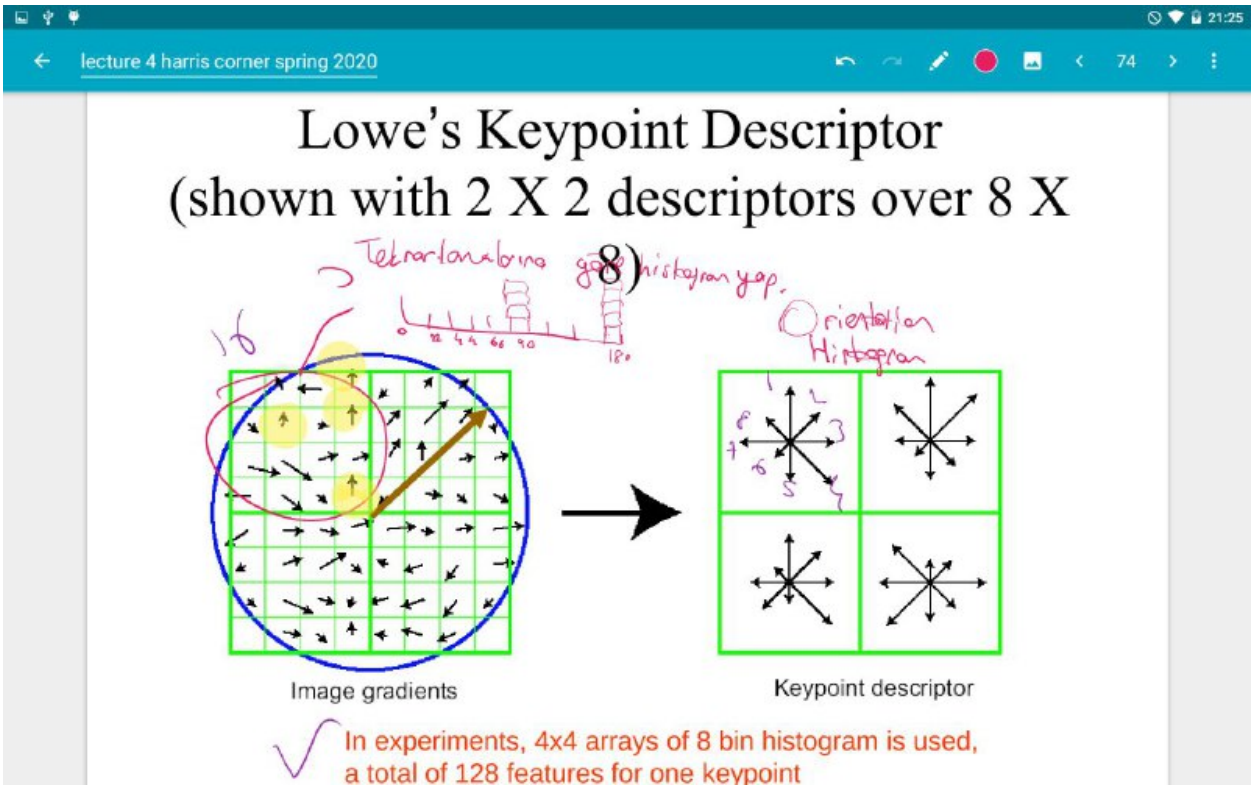


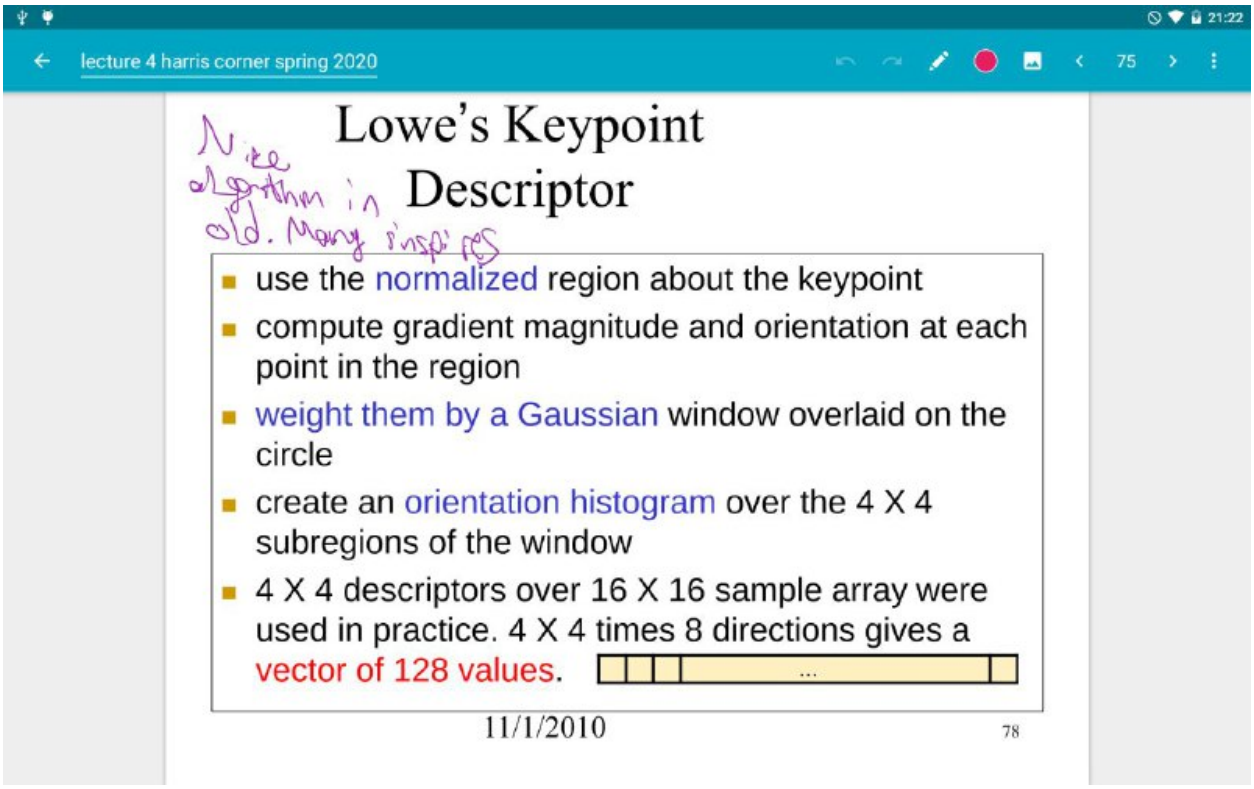
Assign random match, remove bad, do enough times and that’s ***RANSAC***

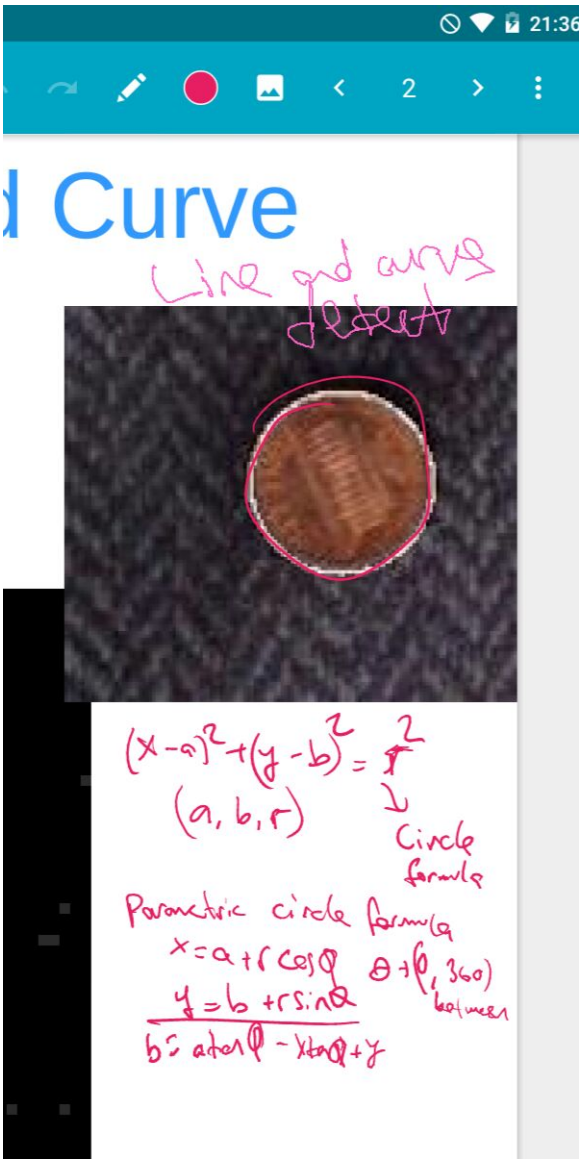
Flann is feature of Ransac[Feature Matching]

Non maximum supression → Edge’e dik yürü, intensity(yoğunluğun) peak noktasını al. Gauss grafiği / ters paraboldaki max nokta.


344 Hw
= Aritmetik ortalama
tandart Sapma(σ) ⇒ Verilerin aritmetik ortalamadan sapmalarının
karelerinin aritmetik ortalamasının karekökü.
eigen value (özdeğer) = Magnitude ,
eigen vector (özvektör) = direction
Bunlar ile
Sistemin yönü
nereye gidiyor
vs. belirleyebiliriz.
x2
x1
varyans ⇒ 2 değişken arasındaki doğrusal ilişkinin değişkenliğini ölçen kavramdır.


lecture 4 harris corner spring 2020
74
Lowe's Keypoint Descriptor
(shown with 2 X 2 descriptors over 8 X 8)
Tekrarlanabilirlik için histogram yap.
Orientation Histogram
Image gradients
Keypoint descriptor
In experiments, 4x4 arrays of 8 bin histogram is used,
a total of 128 features for one keypoint


lecture 4 harris corner spring 2020
75
Lowe's Keypoint Descriptor
use the normalized region about the keypoint
compute gradient magnitude and orientation at each point in the region
weight them by a Gaussian window overlaid on the circle
create an orientation histogram over the 4 X 4 subregions of the window
4 X 4 descriptors over 16 X 16 sample array were used in practice. 4 X 4 times 8 directions gives a vector of 128 values.
11/1/2010
78



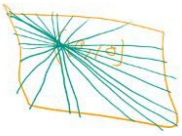
3D voting (a,b and r)

Line Detection by Hough Transforms (HT)

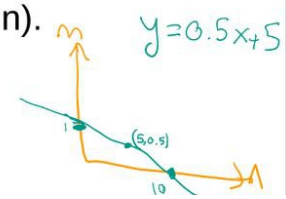
- Instead of going directly for the solution (as in template matching), we can use a voting scheme.
- Each possible point (edge) votes in the parameter space of a line. A line can be specified uniquely by two parameters m and n.

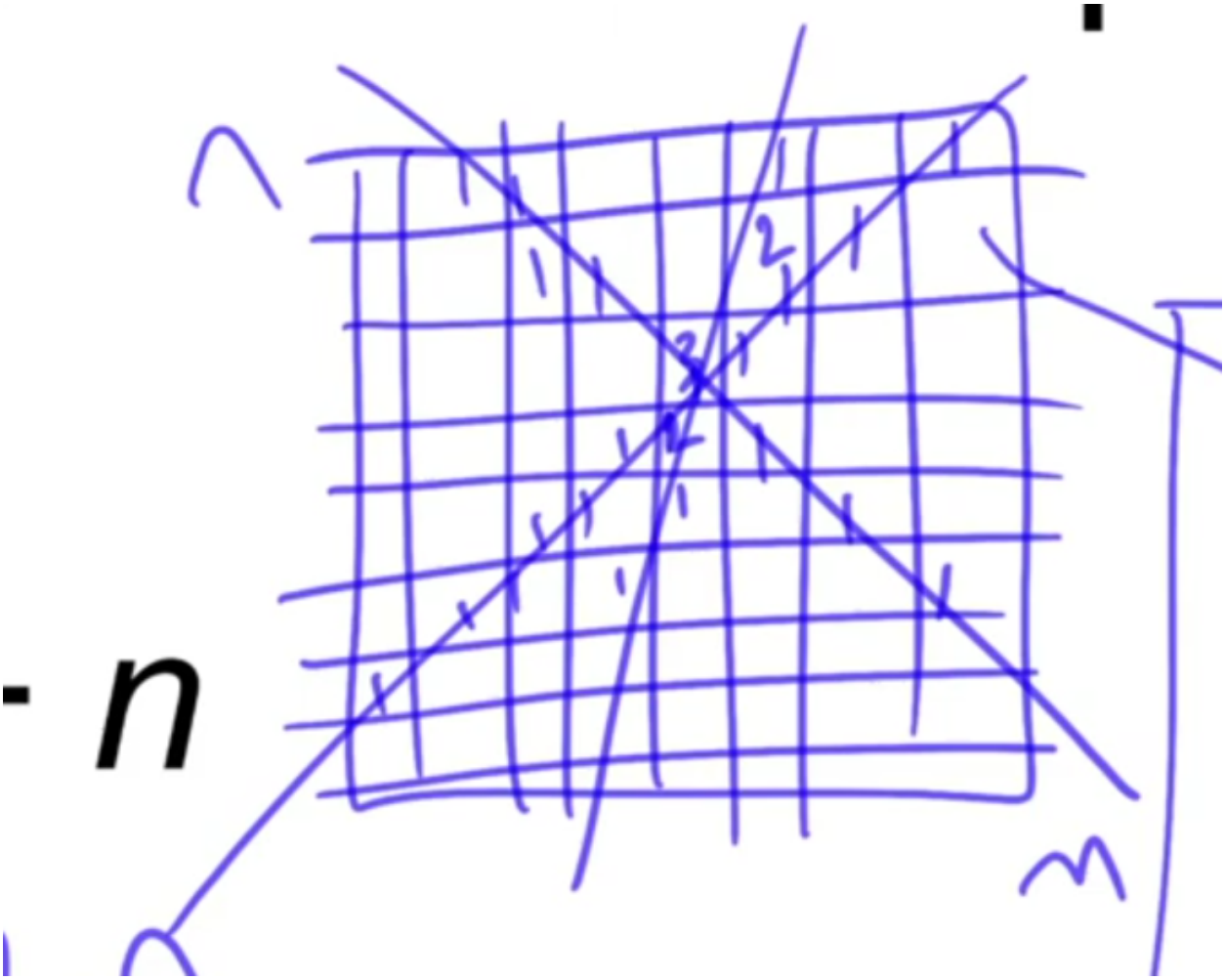
$$y = m.x + n$$

$10 = m10 + n$

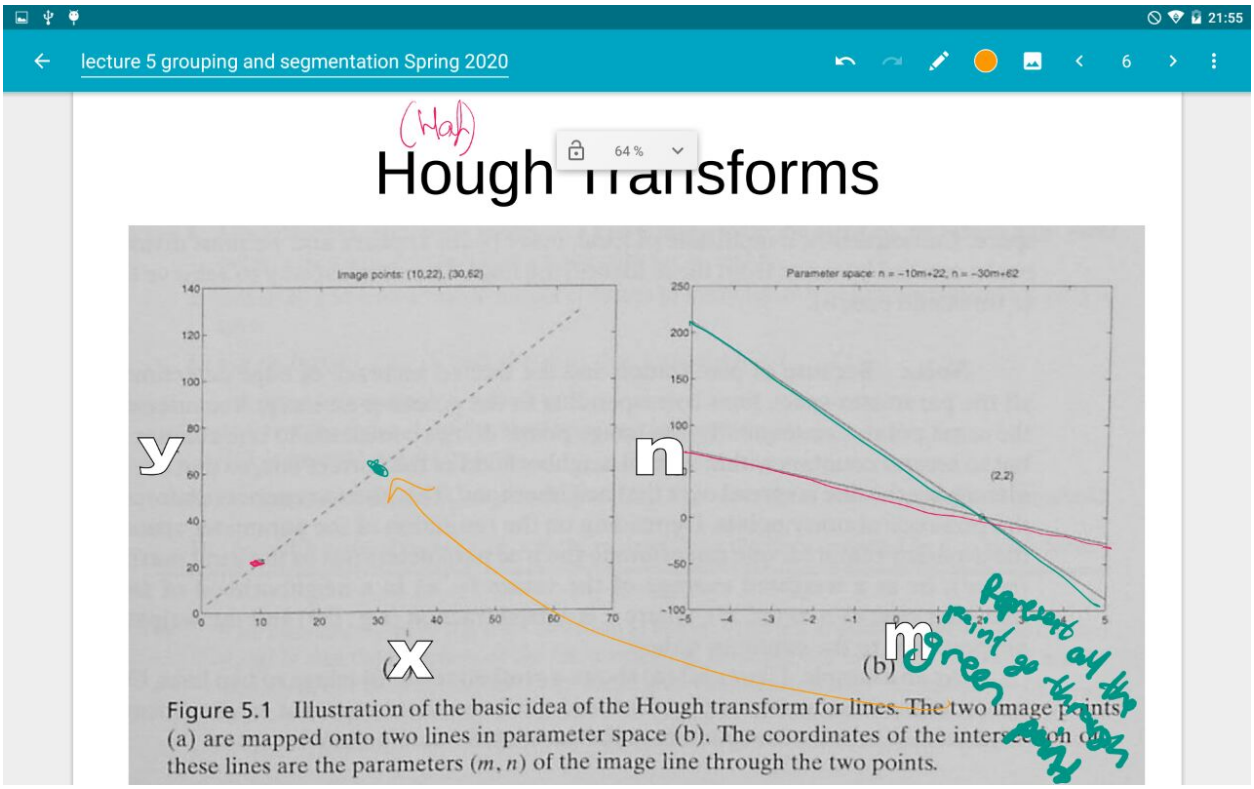


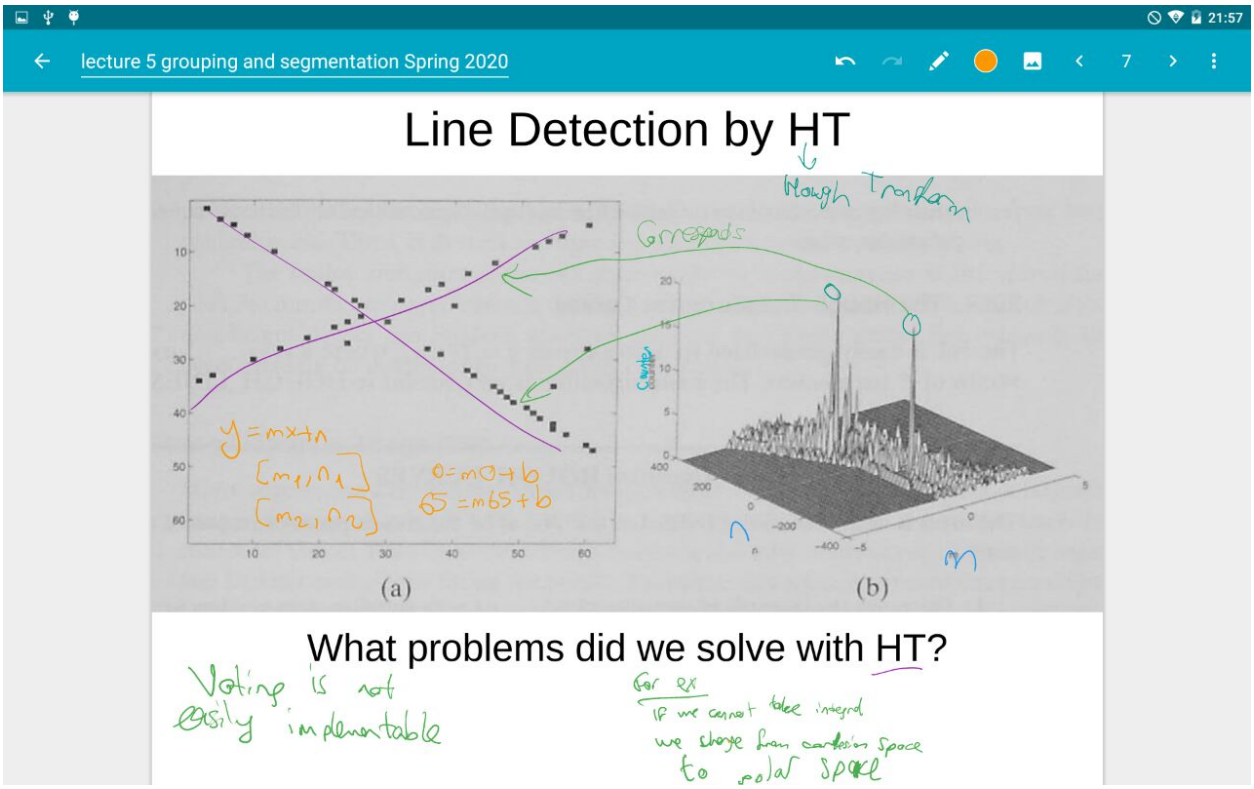
Parameter space of 2 dimensions (m and n).





Voting the highest number. When many lines cross the same point, then increment the counter for it. (LINE DETECTION BY HOUGH TRANSFORMS)





hough sayesinde eğer integral alamıyosak, kartezyen uzaydan polar uzaya geçebilirz

Circle Detection With Hough Transforms

$$(x-a)^2 + (y-b)^2 = r^2$$

is the formula for a circle.

We can use the parametric version

$$x = a + r \cos \theta \quad \rightarrow \theta = 0 \rightarrow 360$$

$$y = b + r \sin \theta$$

When we eliminate r from the above formula, we would have

$$b = a \tan \theta - x \tan \theta + y$$

Circle Detection With Hough Transforms

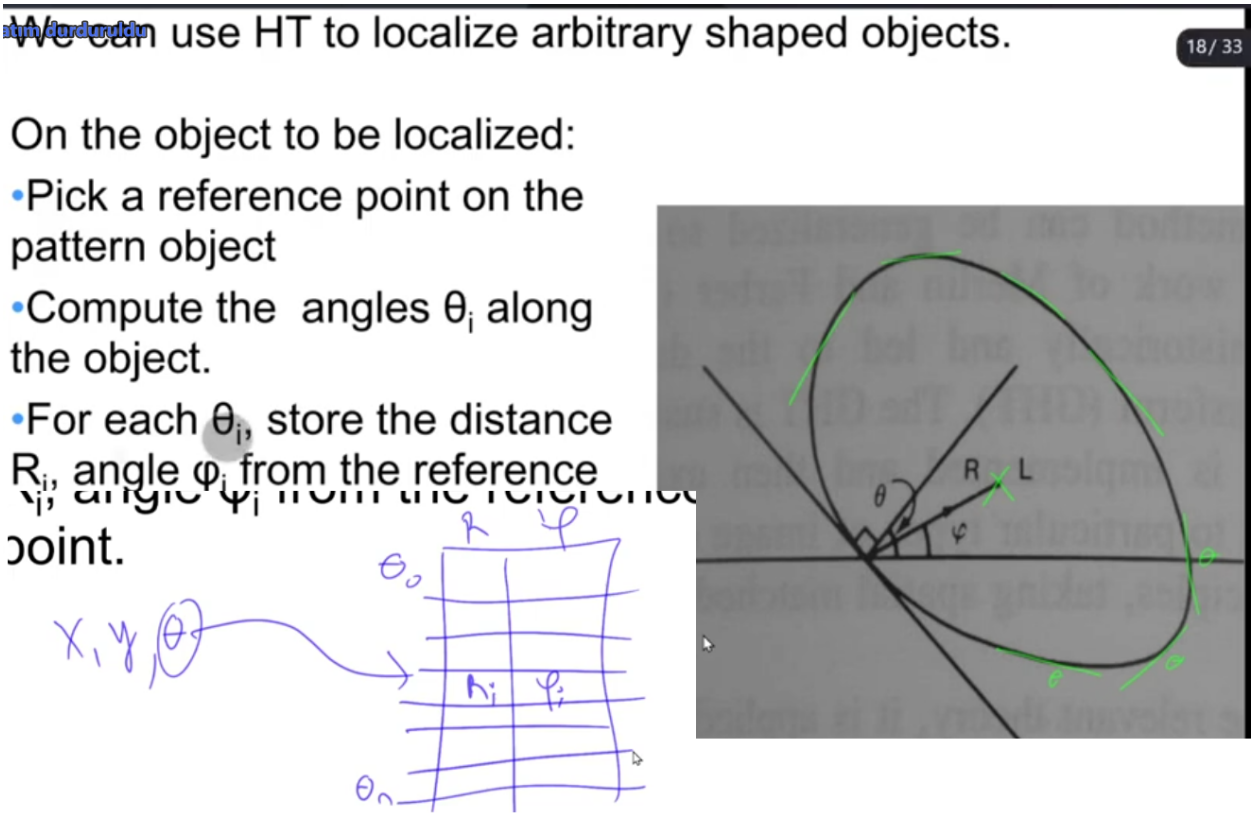
nicelendir $b = a \tan \theta - x \tan \theta + y$

1. Quantize the parameter space for parameters a and b
2. Zero the accumulator array M(a,b)
3. Compute the edges in the image and their angles.
4. For each edge point, increment all points in the accumulator array M(a,b) along the line

$$b = a \tan \theta - x \tan \theta + y$$ voting

1. Local maxima in the accumulator array will give you the centers of the circles. biriktirici

bir fonksiyonun kısmi türevlerinin vektörüne gradyan denir. türev, fonksiyonun bir noktadaki eğimini bulmak için kullanılıyordu. birden fazla türev de yönlü, yani vektörel bir eğim belirtir. dolayısıyla bu tip vektörler, "bir tepeden en hızlı şekilde aşağı inmek için hangi yönlerde ilerlemeliyim?" gibi sorularda işimize yarar.

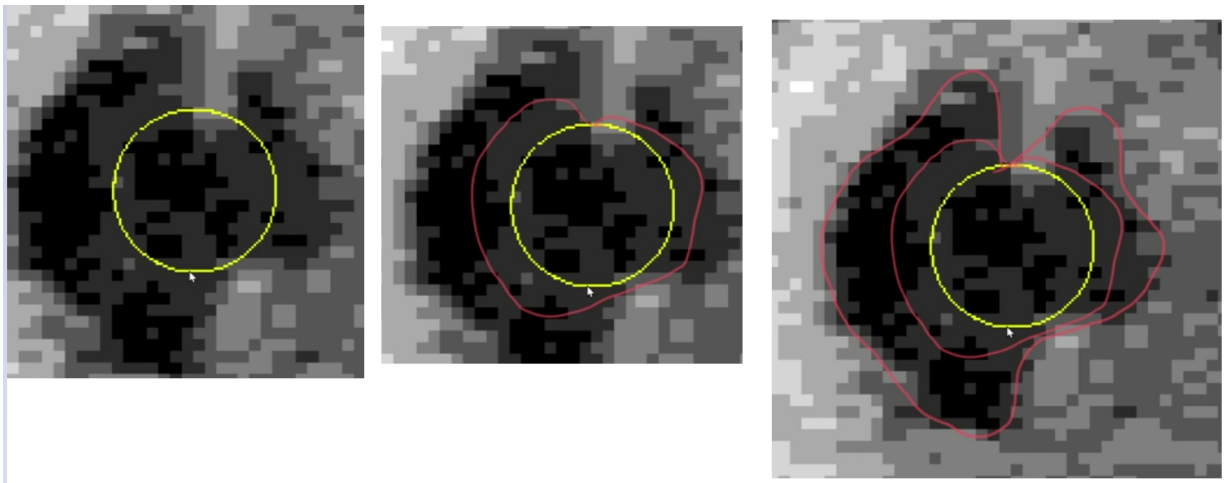


•For each edge point (x,y) with gradient angle θ, the possible locations of the reference point are given by

$$a = x - R(\theta)\cos(\varphi(\theta))$$

$$b = y - R(\theta)\sin(\varphi(\theta))$$

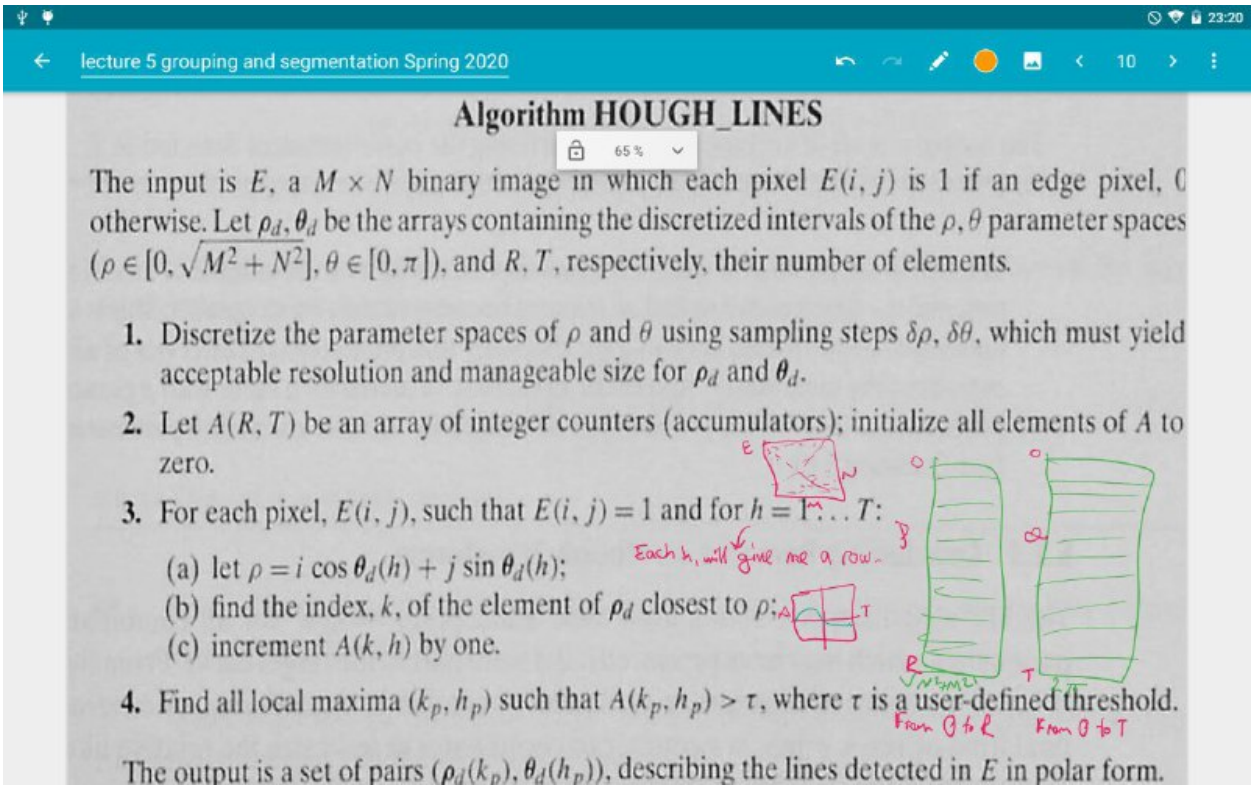
•The peaks on the voting space is the location of the reference point.



IT DOESN'T KNOW WHICH OBJECT LOOKING FOR

[DEFORMABLE CONTOUR SNAKE OPTIMIZATION]

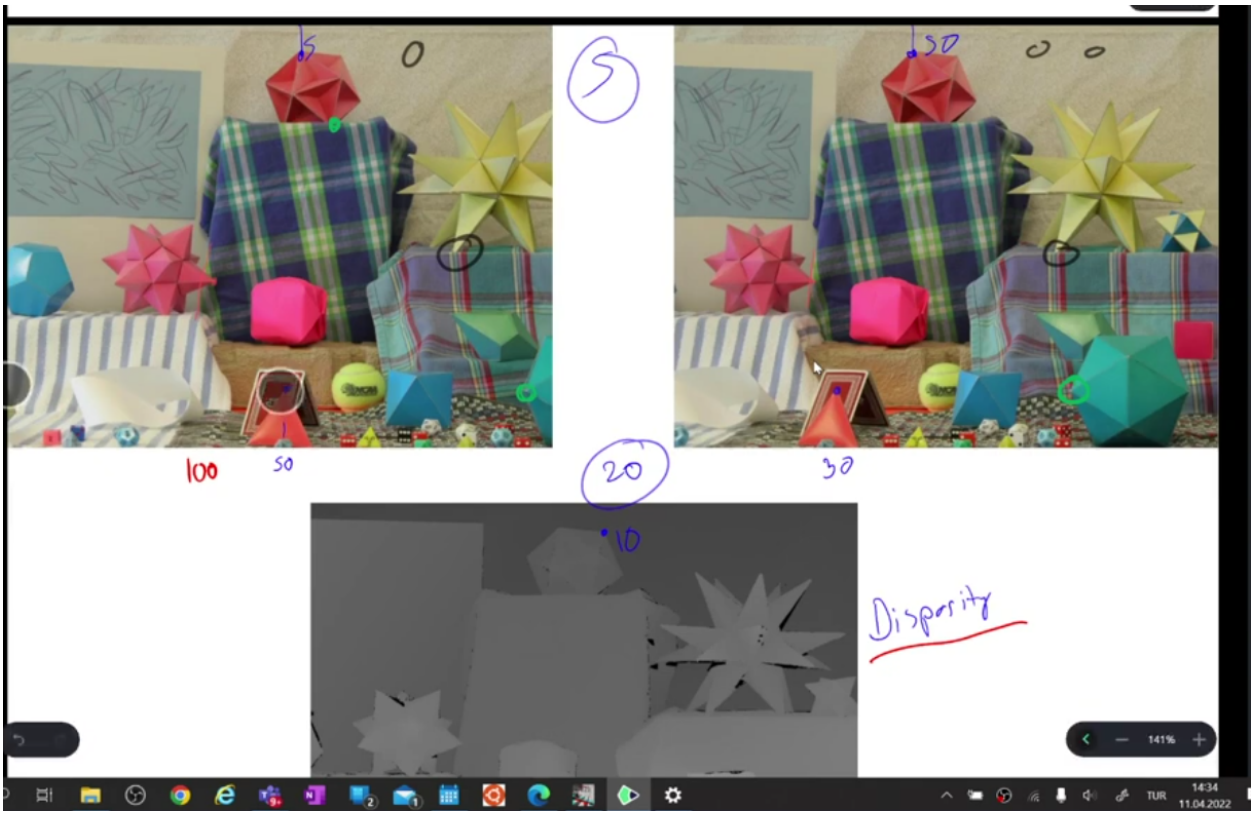
You release a snake and it tries to expand . when sees high gradient image areas, expand



disparity = uyumsuzluk

Computer stereo vision

Bilgisayar stereo görüşü, bir CCD kamera tarafından elde edilenler gibi dijital görüntülerden 3B bilgilerin çıkarılmasıdır. Bir sahne hakkındaki bilgileri iki bakış noktasından karşılaştırarak, iki paneldeki nesnelerin göreceli konumlarını inceleyerek 3B bilgiler çıkarılabilir.



All correspondences are on the same line. **[Epipolar constraint]**

DISPARITY: let’s say that’s my left and right eye. disparity shows how they differ.

If points are close to us, disparity will be larger. Calculating disparity between left and right images will give depth from stereo.

Thus both the Essential and Fundamental matrices completely describe the geometric relationship between corresponding points of a stereo pair of cameras. The only difference between the two is that **the former deals with calibrated cameras, while the latter deals with uncalibrated cameras**.

http://robotics.stanford.edu › ~birch › projective › node20

Essential and fundamental matrices

Öne çıkan snippet'ler hakkında • Geri bildirim

Derivative in Two-Dimensions

- Definition

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\varepsilon \to 0}\left(\frac{f(x+\varepsilon, y) - f(x,y)}{\varepsilon}\right) \qquad \frac{\partial f(x,y)}{\partial y} = \lim_{\varepsilon \to 0}\left(\frac{f(x, y+\varepsilon) - f(x,y)}{\varepsilon}\right)$$

- Approximation

$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x_{n+1}, y_m) - f(x_n, y_m)}{1} \qquad \frac{\partial f(x,y)}{\partial y} \approx \frac{f(x_n, y_{m+1}) - f(x_n, y_m)}{1}$$

- Convolution kernels

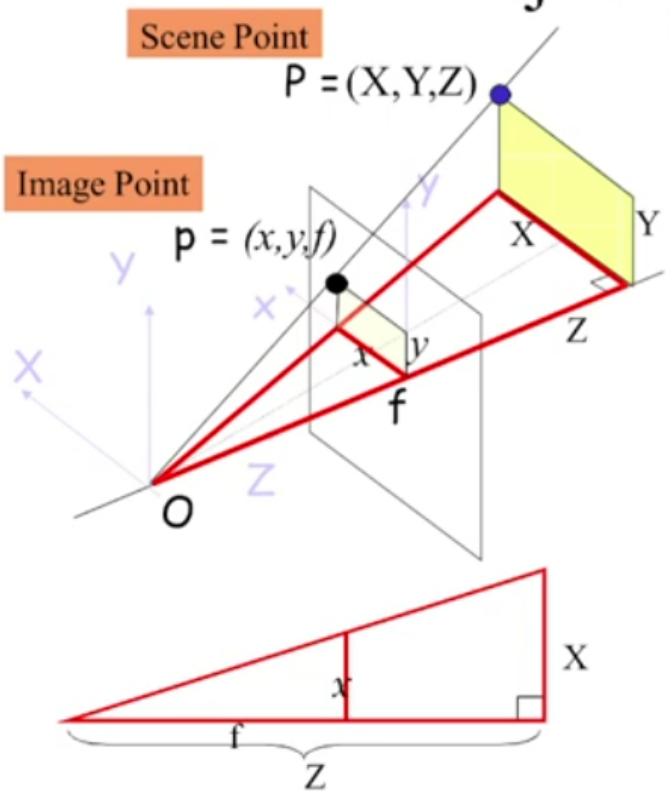
$$f_x = \begin{bmatrix} 1 & -1 \end{bmatrix} \qquad f_y = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Alper Yilmaz, Mubarak Shah Fall 2012UCF

When object that we are looking is far away, disparity is less.

Robert Collins
CSE486, Penn State

Basic Perspective Projection

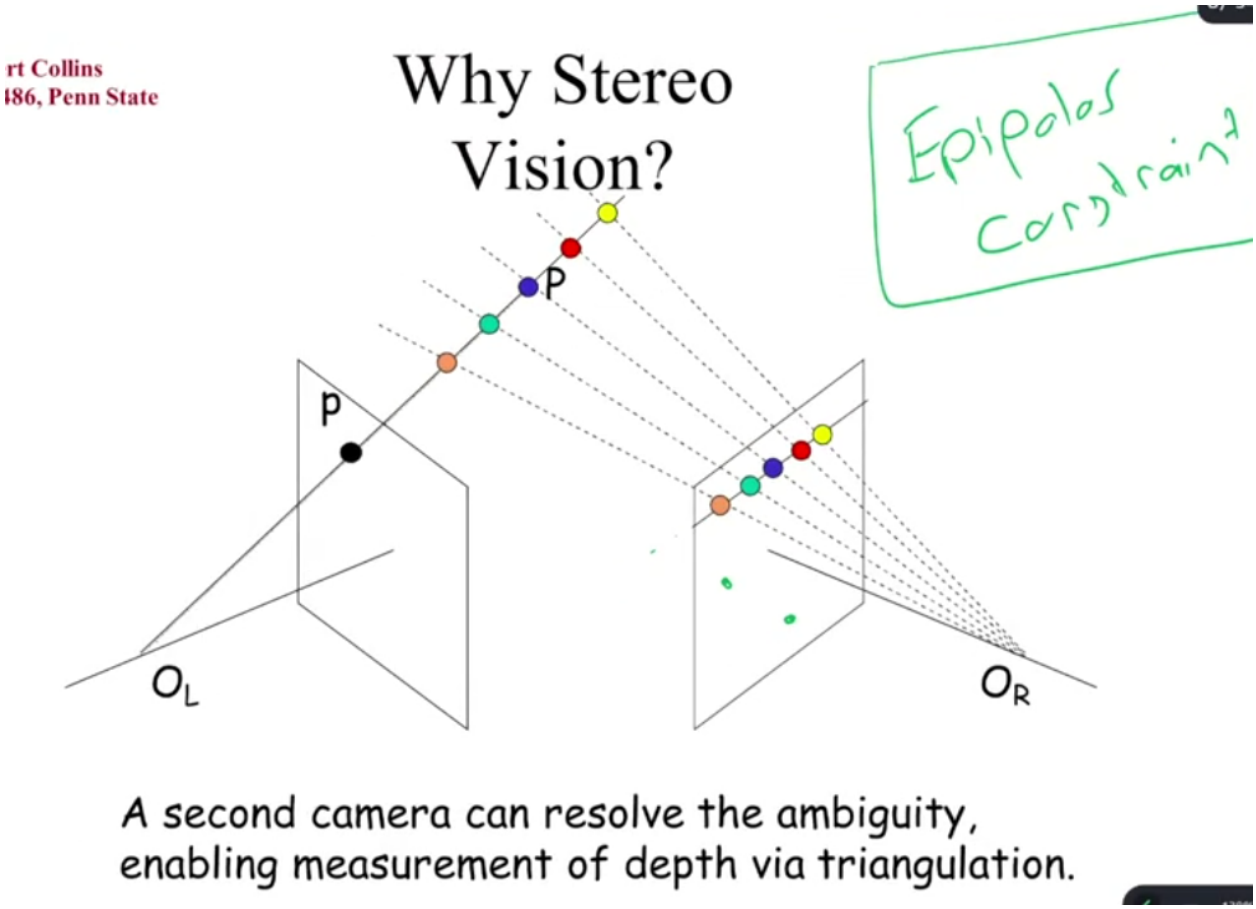


Perspective Projection Eqns

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

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

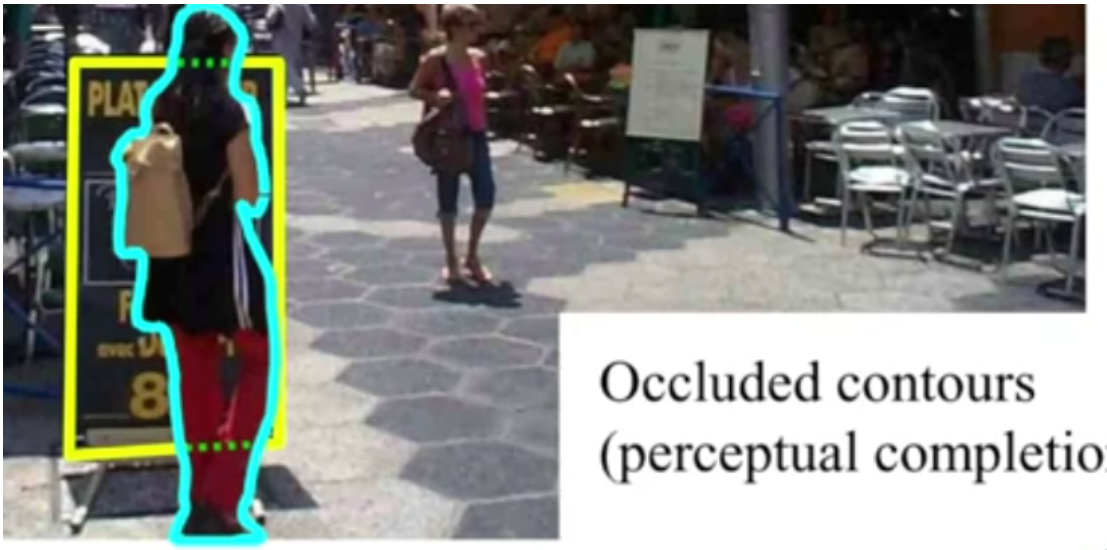
f is like focal length



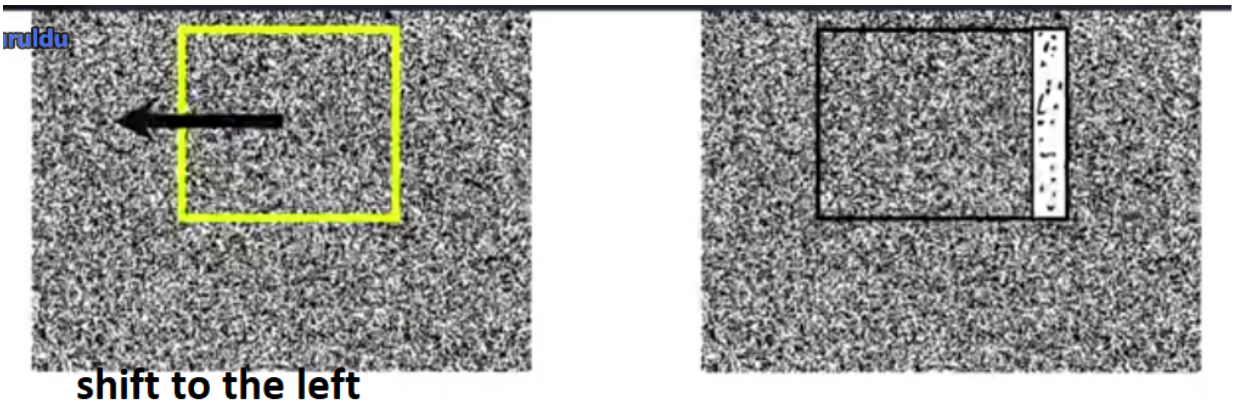
Epipolar constraint → we don't have to search all image for P. one line is enough.

Parallax → Points at different depths displace differently

Anaglyphs (3d gözlük)→ are a way of encoding parallax in a single picture. actually there is no color[just channels] info but we see it as red and cyan(green+blue)

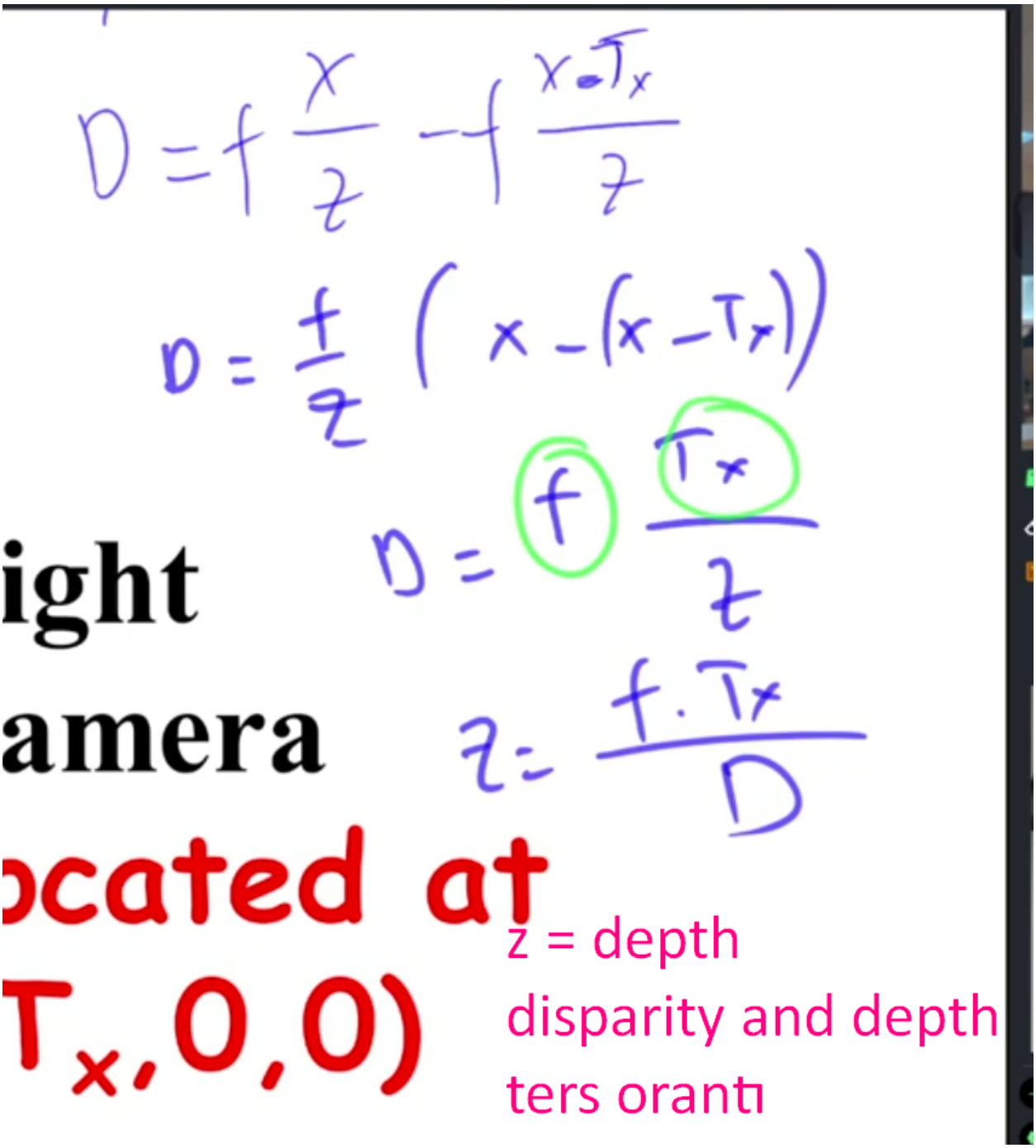


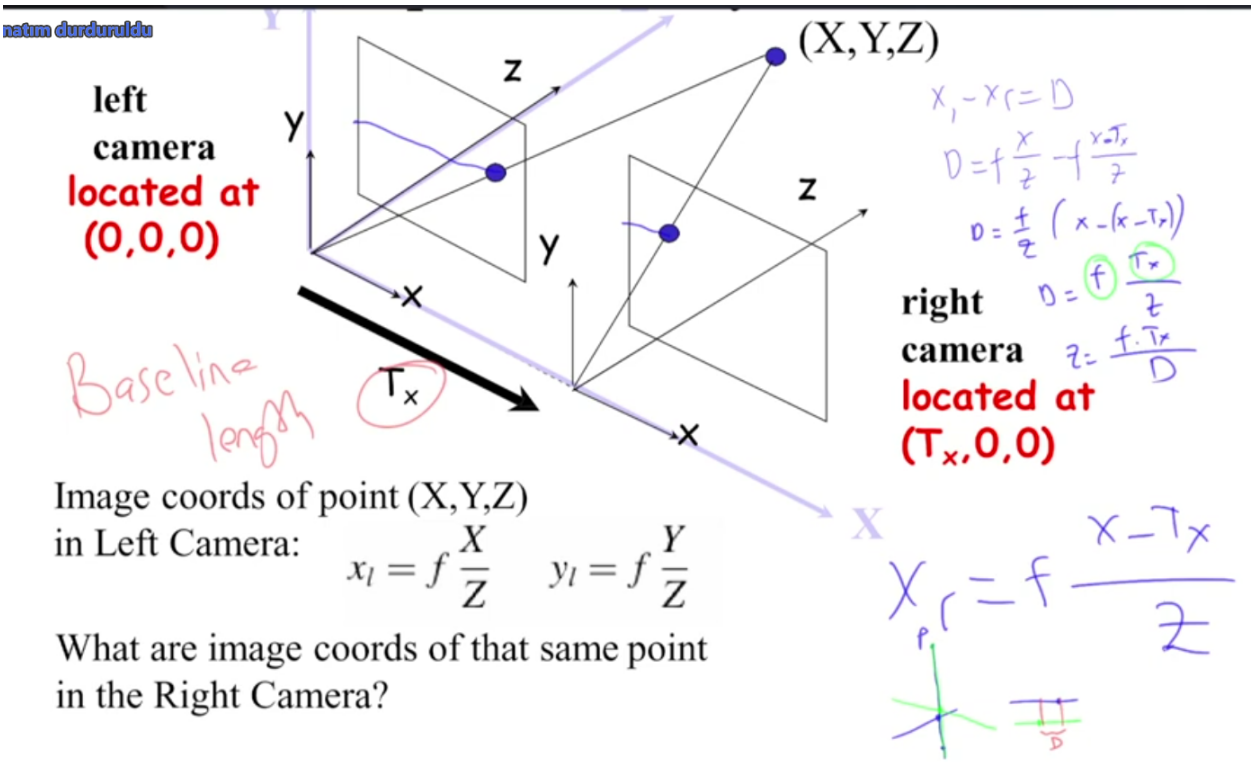
Occlusion gives info about which one is closer.



shift to the left

when we wear glasses
feel 3d
[without using vanishing points etc.]

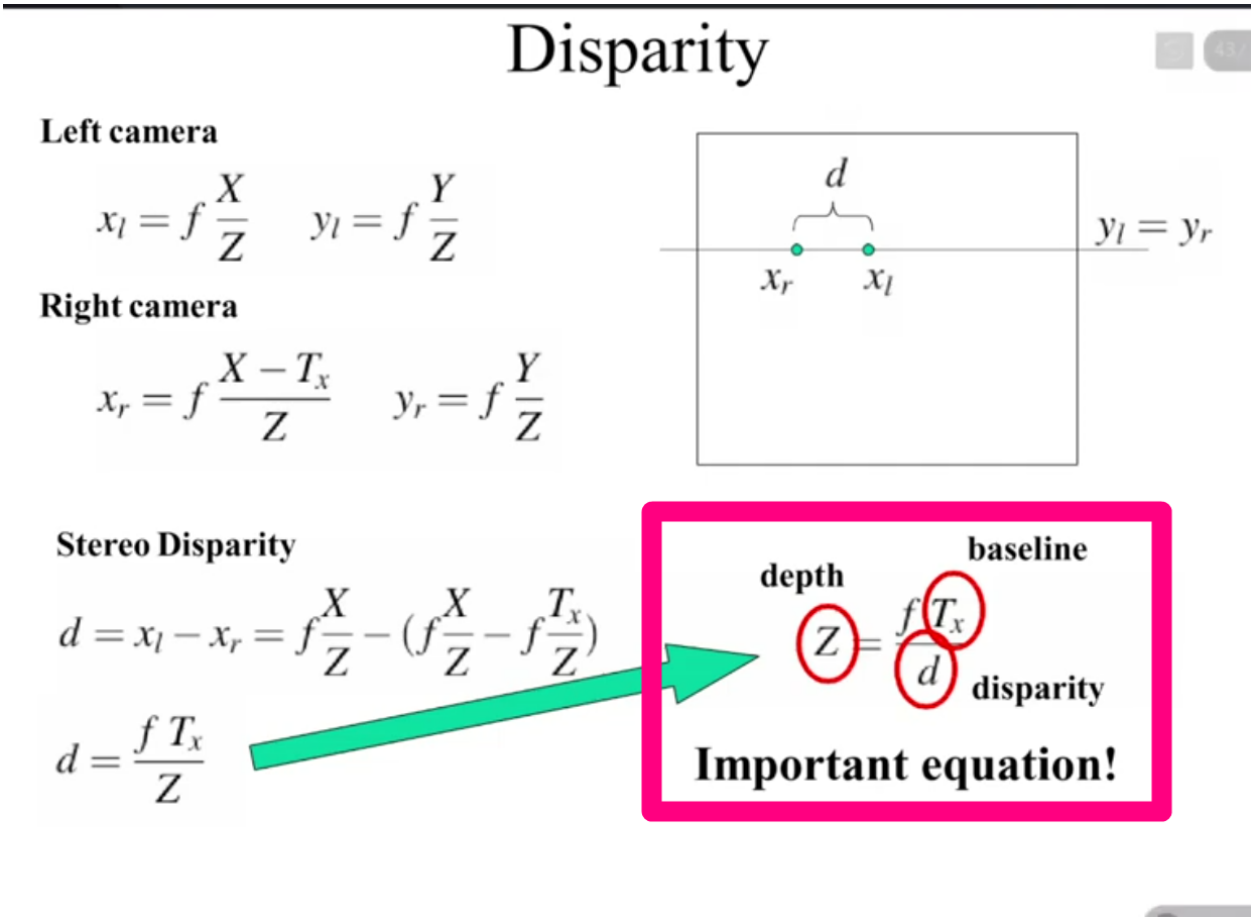

$D = f\frac{x}{z} - f\frac{x - T_x}{z}$
$D = \frac{f}{z}(x - (x - T_x))$
$D = f\frac{T_x}{z}$
$z = \frac{f \cdot T_x}{D}$
ight
amera
ocated at
$T_x, 0, 0)$
z = depth
disparity and depth
ters oranti



$T_x = Baseline\ length$
$Z = depth$
$D = disparity$

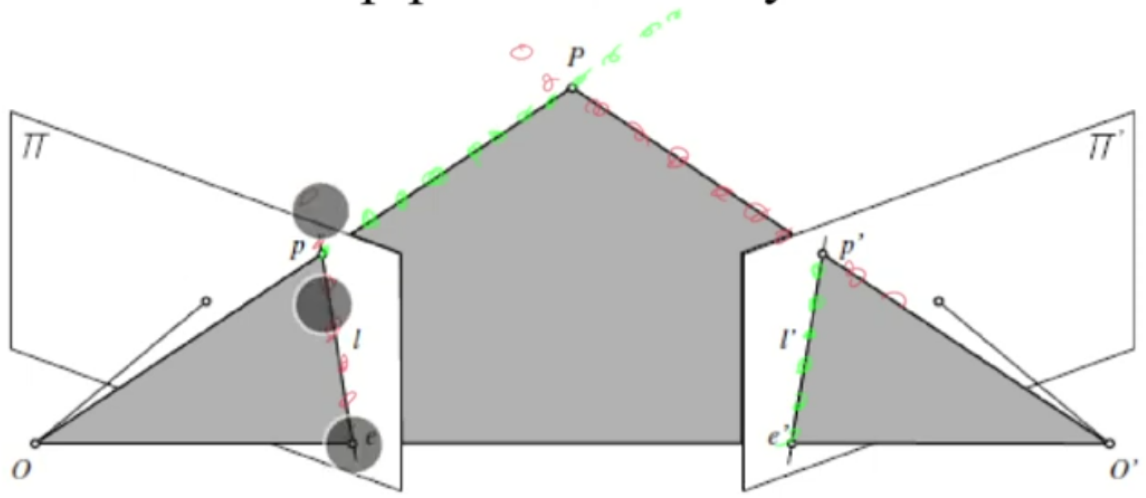
$$Z = \frac{f.T_x}{D}$$

When baseline is large, it will cause more depth and more disparity and baseline will be away from 0. [better Z]



Epipolar geometri stereo görüşün geometrisidir. İki kamera bir 3D sahnesini iki farklı konumdan görüntülediğinde, 3B noktalar ile bunların 2B görüntülere yansımaları arasında görüntü noktaları arasında kısıtlamalara yol açan bir dizi geometrik ilişki vardır.

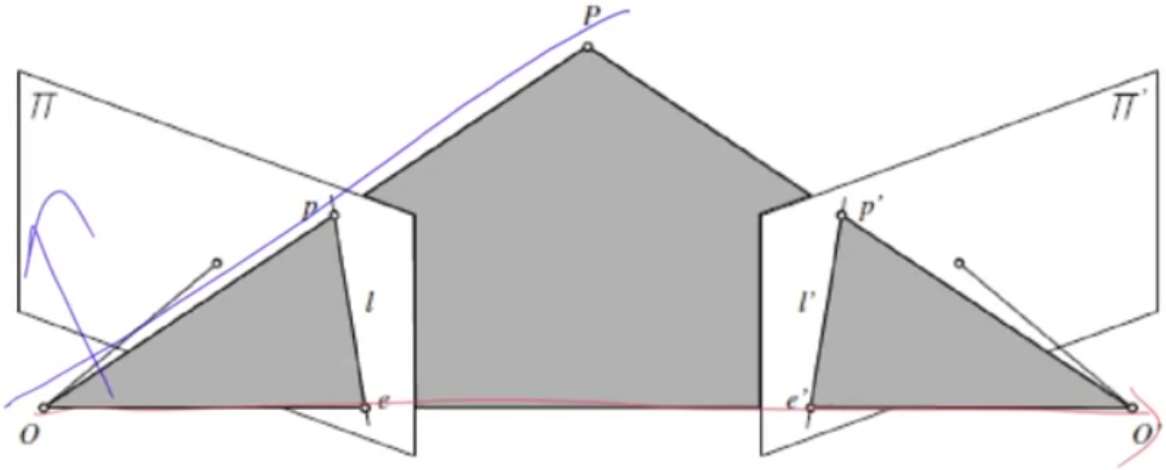
Epipolar Geometry



Epipolar plane, conjugated epipolar lines, epipoles, baseline

konjuge = eşlenik

Epipolar Geometry

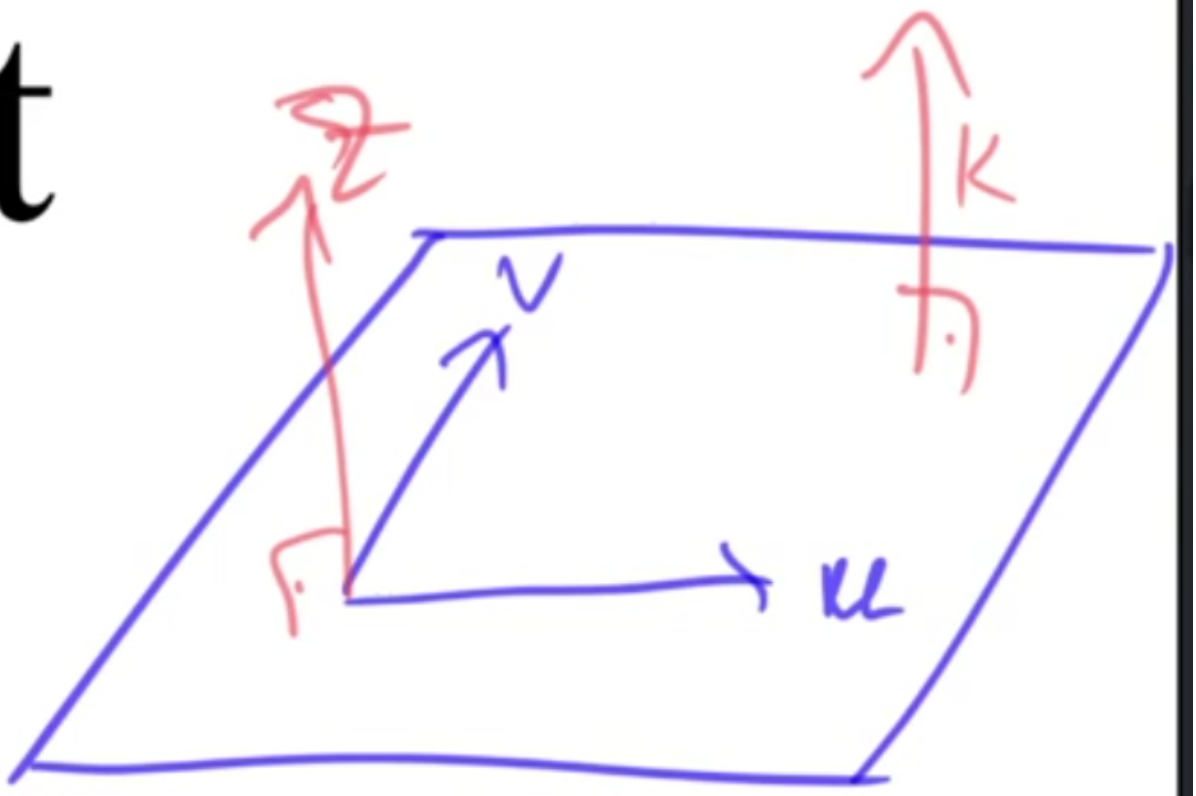


Epipolar plane, conjugated epipolar lines, epipoles, baseline

$(OO' \times OP) \cdot O'P = 0$

we like where epipoles intersect at infinity (like this birthday image.)

t



$v \times u = z$

$v \cdot k = 0$

$a \cdot k = 0$

if perpendicular

cross product is 0

Cross product in matrix form

$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix}.$$

a x a

$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} a_2 & a_3 \\ b_2 & b_3 \end{vmatrix} \mathbf{i} - \begin{vmatrix} a_1 & a_3 \\ b_1 & b_3 \end{vmatrix} \mathbf{j} + \begin{vmatrix} a_1 & a_2 \\ b_1 & b_2 \end{vmatrix} \mathbf{k}$$

The vector cross product also can be expressed as the product of a skew-symmetric matrix and a vector:

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_\times \mathbf{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad [\mathbf{a}]_\times \stackrel{\text{def}}{=} \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

Calibrated Epipolar Equations

Clearly, the epipolar constraint implies that the three vectors $\overrightarrow{Op}$, $\overrightarrow{O'p'}$, and $\overrightarrow{OO'}$ are coplanar. Equivalently, one of them must lie in the plane spanned by the other two, or

$$\overrightarrow{Op} \cdot [\overrightarrow{OO'} \times \overrightarrow{O'p'}] = 0.$$

We can rewrite this coordinate-independent equation in the coordinate frame associated to the first camera as

$$\boldsymbol{p} \cdot [\boldsymbol{t} \times (\mathcal{R}\boldsymbol{p}')],$$

where p and p' denote the homogenous image coordinate vectors, t is the coordinate vector of the translation, and R is the rotation matrix such that a free vector with coordinates w in the second coordinate system has coordinates Rw in the first one.

coplanar = aynı düzlemde

Mathematics [edit]

The gradient of an image is a vector of its partials:[2]: 165

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix},$$

rectification (doğrultma/düzeltme) = özel bir işlemden geçirilerek correspondencelerin aynı satır hizasına gelmesini sağlama

Görüntünün büyüklüğünü ne kadar hızla değiştiğine bölersen **TIME TO COLLISION** bulunur.

$$\frac{D(t)}{v} = T$$

Görüntünün veya objenin ne kadar uzakta olduğunu bilmeye gerek yok, ne de objenin ne hızla yaklaştığını bilmeye.

Projection eşitliği

l(t) = fL / D(t)

imagelar arası stitch olabilmesi için aralarında baseline olmaması lazım

Motion field→ gerçek dünyadaki cisimlerin hız vektörlerinin görüntü üzerine izdüşümü

direkt izdüşüm hesaplamak çok zor bu yüzden optical flow kullan (OF)

Optical flow → görüntü üzerindeki parlaklıkların yer değişimi

Aperture problem

$$(\nabla E)^T.v + E_t = 0$$

Eğer v'nin yönüyle gradyanın yönü çok benzeşiyorsa o zaman elimde image gradient'i olduğu, yönleri aynı olduğu ve uzunlugunu da buldugum için o zaman v'yi elde edebilirim.

Optical flow - Optik akış, video kareleri arasındaki içerik farklılıklarını analiz ederek hareketli nesnelerin bilgisayarlı izlemesini açıklar. Bir videoda, hem nesne hem de gözlemci hareket halinde olabilir; bilgisayar, fotoğrafların sınırlarını, kenarlarını ve bölgelerini işaretleyen ipuçlarını bulabilir. İlerlemelerini tespit etmek, bilgisayarın zaman ve uzayda bir nesneyi takip etmesini sağlar.

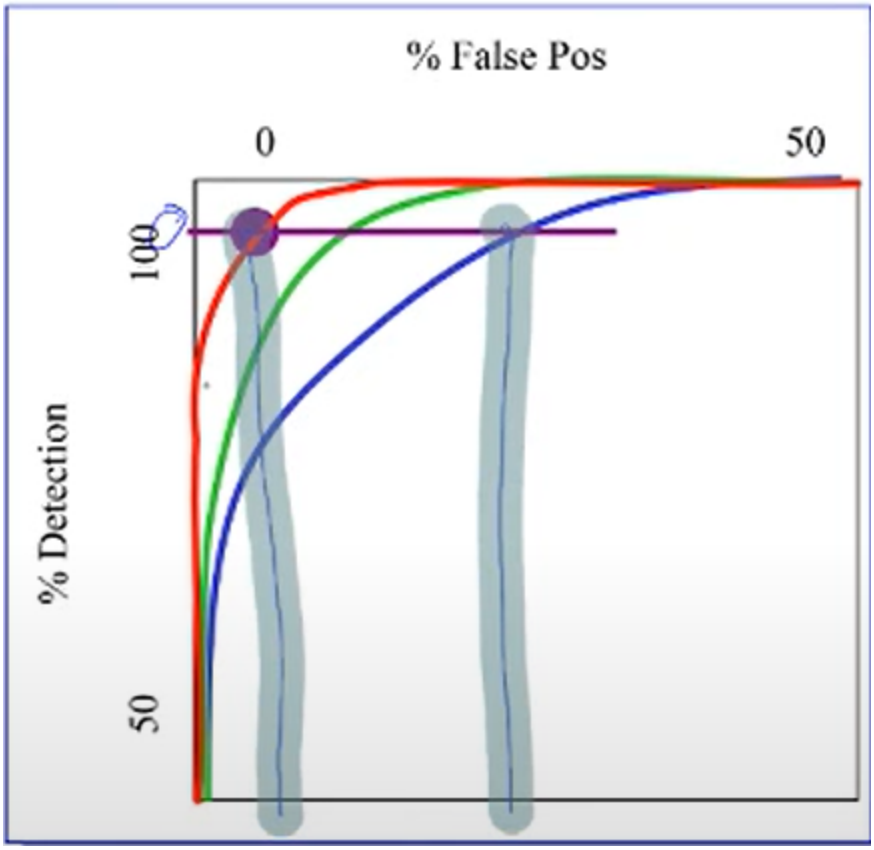
Okların büyüklükleri optic flowun vektörlerinin büyüklüklerini gösteriyor. renkler de yön gösteriyor.

AdaBoosting

Diyelim ki elimde 8-10 sınıflandırıcı var

Her bir weak classifier ile sınıflandır, kötü sınıflandırmaların weight'ini artır böylece sonraki sınıflandırıcı ile bir daha sınıflandırılabilsin. (diğer sınıflandırıcı ağırlığı fazla olanda hata yaparsa cezalandırılacak gibi)

Cascading classifier

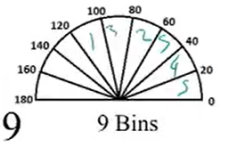


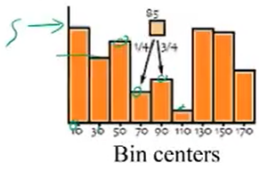
Yüz var mı diye maviye sor, mavi evet derse yeşile sor, yeşil evet derse kırmızıya sor.

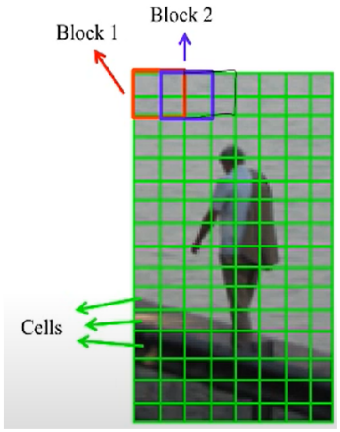
Kırmızı güvenilir ama maviye göre daha yavaş

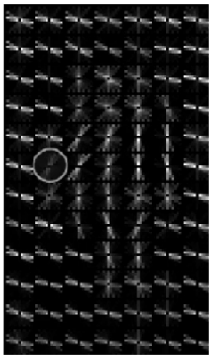
Tri-linear Interpolation

- Each block consists of 2x2 cells with size 8x8
- Quantize the gradient orientation into 9 bins (0-180)
  - The vote is the gradient magnitude









Evaluation Metric

$$\text{precision} = \frac{GT \cap RM}{RM} = \frac{TP}{RM}$$

$$\text{recall} = \frac{GT \cap RM}{GT} = \frac{TP}{GT}$$

$$F\text{-score} = \frac{2PR}{P+R}$$

F-score ‘un iyi olması iyi.

Precision ve recall en fazla 1.0 olabilir ve istediğimiz de bu.

Aperture problem - Açıklık sorunu, bir çubuk veya kenar gibi tek boyutlu bir uzaysal yapının hareketinin, uyaran uçlarının görülemeyeceği şekilde küçük bir açıklıktan bakıldığında açık bir şekilde belirlenemeyeceği gerçeğine atıfta bulunur.

In general, corners can be used as informative
features that are consistently identifiable over time. For motion estimation, the presence of the two intersecting edges
that define a corner disambiguates the unobservable motion
in the direction parallel to a single edges orientation, i.e. it
solves the aperture problem.

Weak perspective, what we are saying is that **the difference of depth over an object or over the range of an object, is very small compared to the difference in depth from the object to the center of projection**. And if that's true, then you can basically say that each object has its own scale factor. In weak perspective we assume that all of our objects are z_0 away from us.

With using SVD, we can reduce redundancy im struct from motion matrices.

W = M*S (Camera motion * scene structure)

Optical flow
• Definition: optical flow is the apparent motion of brightness patterns in the image
• Ideally, optical flow would be the same as the motion field

Geometric camera calibration, estimates the parameters of a lens and image sensor of an image or video camera. You can use these parameters to correct for lens distortion, measure the size of an object in world units, or determine the location of the camera in the scene. These tasks are used in applications such as machine vision to detect and measure objects. They are also used in robotics, for navigation systems, and 3-D scene reconstruction.

Formally, an image gradient is defined as **a directional change in image intensity.** Or put more simply, at each pixel of the input (grayscale) image, a gradient measures the change in pixel intensity in a given direction.

A sift keypoint is circular image region with an orientation. Keypoint center coordinates x and y, its scale and its orientation. It’s local feature of image . Keypoints are scale & rotation invariant

Disadvantage of Harris corner
detector is that it is not scale invariant and though it is rotational invariant it is not that much reliable[9]. There are some scale and rotation invariant feature detector or descriptors for ex: SIFT, SURF, BRIEF etc. whose descriptor can be used for further feature matching.

Also, Lowe aimed to create a descriptor that was robust to the variations corresponding to typical viewing conditions. The descriptor is the most-used part of SIFT.

The **rectification** process can be implemented by projecting
the original pictures onto the new image plane .

With an appropriate choice of coordinate system, the rectified epipolar lines are scanlines of the new images, and they are also parallel to the baseline

---

The algorithm consists of four steps:

- Rotate the left camera so that the epipole goes to infinity along the horizontal axis.
- Apply the same rotation to the right camera to recover the original geometry.
- Rotate the right camera by R.
- Adjust the scale in both camera reference frames.

Algorithm RECTIFICATION

The input is formed by the intrinsic and extrinsic parameters of a stereo system and a set of points in each camera to be rectified (which could be the whole images). In addition, Assumptions 1 and 2 above hold.

1. Build the matrix R_{rect} as in (7.22);
2. Set $R_l = R_{rect}$ and $R_r = RR_{rect}$;
3. For each left-camera point, $\mathbf{p}_l = [x, y, f]^\top$ compute

$$R_l \mathbf{p}_l = [x', y', z']$$

and the coordinates of the corresponding rectified point, $\mathbf{p}'_l$, as

$$\mathbf{p}'_l = \frac{f}{z'}[x', y', z'].$$

4. Repeat the previous step for the right camera using R_r and $\mathbf{p}_r$.

The output is the pair of transformations to be applied to the two cameras in order to rectify the two input point sets, as well as the rectified sets of points.

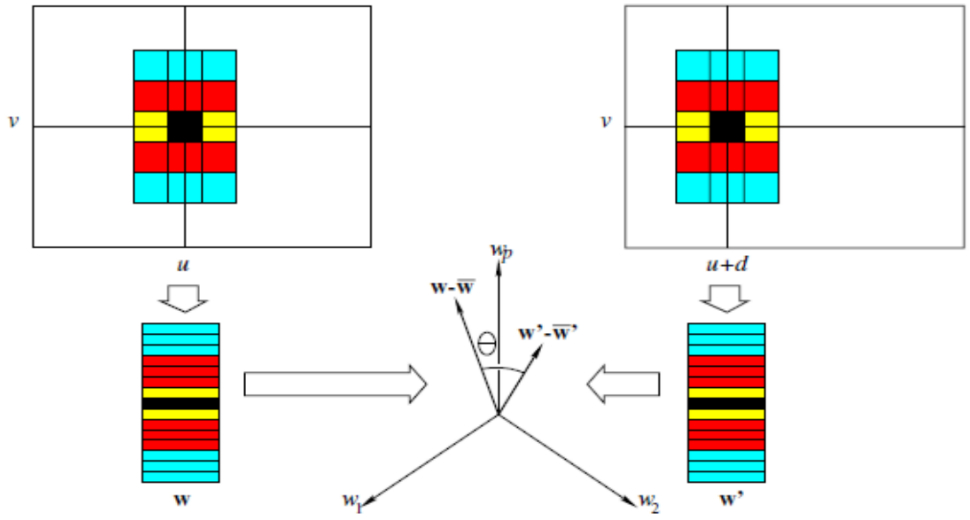
Fundamental matrix 3x3

7 degrees of freedom

rank = 2

$x'^T F x$ = 0

Normalized Cross Correlation



Line detection

ro = x.cos teta + y.sin teta

Görüntünün büyüklüğünü ne kadar hızla değiştiğine bölersen **TIME TO COLLISION** bulunur.

$\frac{D(t)}{v} = T$

The image of the object has size l(t):

$$l(t) = \frac{fL}{D(t)}$$

HARRIS CORNER DETECTOR

1 - compute image derivatives

2 - compute M components as squares of derivatives

3 - Gaussian filter g() with width σ.

g(Ix^2) g(Iy^2) g(Ix o Iy)

4 Croner score lambda1*lambda2 - alfa (lambda1+lambda2)^2

5 threshold on score

6 Then apply non maximal supression.

Aperture Problem describes a situation where the size of a window is too small to obtain useful information about changes in the image. The way I like to think about it is that there is an appropriate window size for every feature and image.
https://elvers.us/perception/aperture/

Corners are interest points which satisfy some of the above properties. They also help solve another problem called the Aperture problem. Given a region in the image with just Edge or flat region it is impossible to identify it uniquely. There could be multiple matching patches to this given patch.

HOUGH TRANSFORM

uniform değil

$$P^T F p' = 0$$

$$K^{-T} \epsilon K'^{-1}$$

ESSENTİAL = T_x* R

Thus both the Essential and Fundamental matrices completely describe the geometric relationship between corresponding points of a stereo pair of cameras. The only difference between the two is that **the former deals with calibrated cameras, while the latter deals with uncalibrated cameras**

Both matrices relate corresponding points in two images. The difference is that in the case of the Fundamental matrix, the points are in pixel coordinates, while in the case of the Essential matrix, the points are in "normalized image coordinates". Normalized image coordinates have the origin at the optical center of the image, and the x and y coordinates are normalized by Fx and Fy respectively, so that they are dimensionless.

The two matrices are related as follows: E = K' * F * K, where K is the intrinsic matrix of the camera.

F has 7 degrees of freedom, while E has 5 degrees of freedom, because it takes the camera parameters into account. That's why there is an 8-point algorithm for computing the fundamental matrix and a 5-point algorithm for computing the essential matrix.

One way to get a 3D position from a pair of matching points from two images is to take the fundamental matrix, compute the essential matrix, and then to get the rotation and translation between the cameras from the essential matrix. This, of course, assumes that you know the intrinsics of your camera. Also, this would give you up-to-scale reconstruction, with the translation being a unit vector.

—

In a stereo image pair, the fundamental matrix encodes the rigidity constraint of the scene. It
combines the internal parameters of both cameras (which can be the same) and their relative
position and orientation. It associates to image points in one view the so-called epipolar line in
the other view, which is the locus of projection of the same 3D point, whose particular position
on the straight line is determined by its depth. Reducing the correspondence search to a 1D line
instead of the 2D image is a large benefit enabling the computation of the dense 3D scene. The
estimation of the matrix depends on at least seven pairs of corresponding points in the images.

—

The epipolar geometry depends on the relative pose and calibration of the cameras
This can be computed using the fundamental matrix
Once this is computed, we can use the epipolar lines to restrict the search space to a 1D search

The fundamental matrix expresses the same constraint, but when two cameras are characterized by general (and unknown) camera matrices K

and K’, respectively. Therefore the derivation of F includes the camera matrices for both cameras.