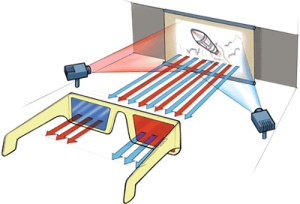
**1. How to display different images to two eyes?**

In order to see things in 3D each eye must see a slightly different picture. This is done in the real world by your eyes being spaced apart so each eye has its own slightly different view. The brain then puts the two pictures together to form one 3D image that has depth to it.

Methods:

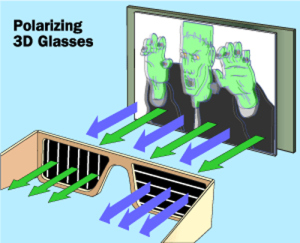
**(1) Color Filter Glasses**

The same scene is projected simultaneously from two different angles in two different colors, red and cyan (or blue or green). The color filter glasses separates the two different images so each image only enters one eye and your brain puts the two images back together. You cannot really have a color movie when you are using color to provide the separation, so the image quality is not nearly as good as with the polarized system.



**(2) Polarized Glasses**

Two synchronized projectors project two respective views onto the screen, each with a different polarization. The glasses allow only one of the images into each eye because they contain lenses with different polarization.



**(3) Synchronization between the screen and glasses (Active shutter 3D system)**

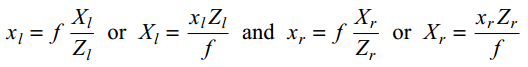
Rather than interleaving two half-resolution images in space, it interleaves two full resolution images in time. The TV rapidly alternates showing images for the left and right eye, while the glasses use electronic shutters synchronized to the TV to ensure that each eye sees a different image. This system is more complex but allows for a higher resolution 3D image.

Active shutter 3D systems generally use liquid crystal shutter glasses (also called "LCS glasses", "LCS 3D glasses", "LC shutter glasses" or "active shutter glasses"). Each eye's glass contains a liquid crystal layer which has the property of becoming opaque when voltage is applied, being otherwise transparent. The glasses are controlled by a timing signal that allows the glasses to alternately block one eye, and then the other, in synchronization with the refresh rate of the screen. The timing synchronization to the video equipment may be achieved via a wired signal, or wirelessly by either an infrared or radio frequency (e.g. Bluetooth, DLP link) transmitter.



**2. Displaying disparity**

- Consider recovering the position of P from its projections pl and pr



where f is the focal length of a camera.

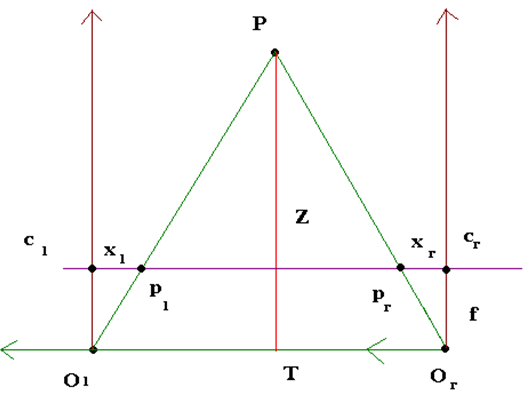
- In general, the two cameras are related by the following transformation: Pr = Pl − T

where T is distance between the two cameras.

- Using Zr = Zl = Z and Xr = Xl − T, we have:



where d = xl - xr is the disparity (i.e., the difference in the position between the corresponding points in the two images)



**3. a. Panoramic Image Stitcher - Warping(How to warp image to cylinder)**

**b. Panoramic Image Stitcher – Blending**

**4. Shape context descriptor**

**5. Eliminating False Matches through Relaxation**

**6. Chamfer Distance + Distance transform**

**Chamfer Distance:** Average distance to nearest feature

To determine something in given image, one can use mask and move it in edge image from left to right, from top to bottom, use chamfer distance to compute distance to something. The position of something gives minimum distance.



T: template shape -> a set of points

I: Image to search -> a set of points

d i(t) = min distance from a point t to some points in I

Feature can be edge points, foreground points …

**Distance transform**

Distance transform is a function that for each image pixel p assigns a non-negative number corresponding to distance from p to the nearest feature in the image I.

Distances transform (1D)

1. Initialize: For all j D[j] 1 p[j] // 0 if j is in P, identify otherwise
2. Forward : For j from 1 to n-1 D[j] min (D[j], D[j-1] + 1 )
3. Backward: For j from n-2 to 0 D[j] min (D[j], D[j+1] + 1)

Distances transform (2D) is analogous distances transform (1D)

1. Initialization
2. Forward pass finds closest above and to left.
3. Backward pass finds closest below and to right.

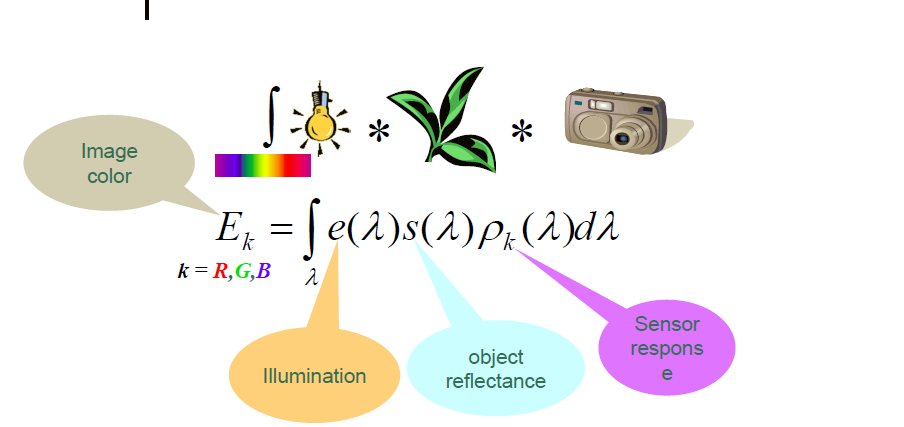


**7. Color basic and Color constancy**

**a. Color basic and Color constancy**

**Color basic:**

Color is important factor. Color allows people to identify a lot of thing or check people’s health, fitness … In computer vision; color is used recognition and segmentation. Object color depends on illumination. Color is perceptual property. Color seems to fade in low light. Color has infinite number of frequencies.

Apply formula to compute image color

**Color constancy**: is a part of visual perceptual system which allows people perceive color under different illumination and to see the consistency of color. An apple has red color at the midday and also red at sunset although the illumination has different wave length.

**b. Color constancy by Gray World**

The Gray World Assumption is a white balance method that assumes that given an image with sufficient amount of color, the average value of the R, G, and B components of the image should average to a common gray value. In the case an image is taken by a digital camera under a particular lighting environment, the effect of the special lighting cast can be removed by enforcing the gray world assumption on the image. As a result of approximation, the color of the image is much closer to the original scene.

Example of Gray World balance:

P1 (190, 210, 195)

P2 (210,205,185)

P3 (200,200,205)

R av = = 200

G av = = 205

B av =  = 195

∝ =  = 205/200

β =  = 205/195

* R’ =  = 205
* G’ = G av = 205
* B’  =  = 205

**c. Color constancy by Retinex**

These algorithms receive as input the red, green, blue values of each pixel of the image and attempt to estimate the reflectance of each point. Algorithm operates as follows: the maximal read value r max of all pixels is determined, and also the maximal green value g max and the maximal blue value b max. Assuming that the scene contains objects which reflect all red light, and (other) objects which reflect all green light and still others which reflect all blue light, one can then deduce that the illuminating light source is described by (r max, g max, b max ). For each pixel with values(r, g, b) its reflectance is estimated as (r/r max, g/max, b max).Although retinex models are used in computer vision, they have been shown not accurately model human color perception.

Example of Retinex

P1 (190, 210, 195)

P2 (210,205,185)

P3 (200,200,205)

R max = max (190,210,200) = 210

G max = max (210,205,200) = 210

B max = max (195,185,205) = 205

Scale use Reference White (255,255,255)

Scale R = R max/ 255 = 210/255

Scale G = G max/ 255 = 210/255

Scale B = B max/ 255 = 205/255

Apply this formula to have P’1, P’2; P’3



**8. Energy minimization**

**9. C- rule algorithm + Gamut Mapping**

**Gamut Mapping**

Gamut is central notion of color constancy algorithm.

Gamut is convex. Approximate Gamut by a convex hull.

In real world images, for a given illumination, one observes only a limited number of colors. This limited set of colors that can occur under a given illuminant is call canonical gamut and the known light source can be called “canonical illuminant”.

The Gamut Mapping algorithm consists of three steps as follow:

1. Estimate the gamut of the unknown light source
2. Determine the set of feasible mappings, i.al. All mappings can be applied to the gamut of the input image and result in a gamut that lies completely within the canonical gamut. Under the assumption of the diagonal mapping, a unique mapping exists that converts the gamut of the unknown light source to the canonical gamut.
3. Apply an estimator to select one mapping from the set of feasible mappings. The selected mapping can be applied to the canonical illumination to obtain an estimate of the unknown illuminant.

A convex combination of reflectance functions is a valid reflection function.

Assume that P1 is coordinating of P under the sun: P1 (Ri, Gi, Bi )

P2 is coordinating of P under the lamp: P2 (ri, gi, bi)

Find relationship between 2 coordinate:

Exist: fx = ∝ a + β b + γ c (x = r , g, b)

fr(Ri, Gi, Bi )= ri => ri = ∝r Ri + βr G + γrB

fg(Ri, Gi, Bi )= gi => gi = ∝g Ri + βg G + γgB

fb(Ri, Gi, Bi )= bi => bi  = ∝b Ri + βb G + γbB

Matrix: A =  

Ci’ = A Ci

* Ci = A-1 Ci’

**C rule algorithm**

Training: compute canonical gamut

Give a new image

1. Find mappings which map each pixel to inside canonical gamut
2. Choose one such mapping
3. Compute new RGB values

Example:

P1 (100,200,200)

P1 (200,200,200)

Eliminate blue component:

r1 = 100/200 = 0.5; g1 = 200/200 = 1;

r2= 200/200 = 1; g2 = 200/200 = 1;

k1 = r2/r1 = 2

k2 = g2/g1 = 1

Find the matrix with maximum trace (i.e. max (k1+ k2))



New value R B G by applying formula

x   