# HW 1 Solution

## 1. Geometric image formation

a: P = (30, 20)

b: behind model projects an inverted image; front model projects an upright image.

The behind model corresponds better to physical pinhole. The other model has same equation except image inversion.

c: focal length gets bigger, the projection becomes bigger; distance gets bigger, the projection becomes smaller.

d: (1,1,1); another: k*(1,1,1)

e: (1/2, 1/2)

f: infinity point which represents a direction

g: In homogeneous coordinates:

$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ which is a linear equation, and u = U/W, v = V/W.

h: dimension of M is 3x4, K is 3x3, I is 3x3, O is 3x1.

i: p = (1.8, 4.6)

## 2. Modeling transformations

a: (3,4)

b: (2,2)

c: (0, $\sqrt{2}$)

d: (2, 2-$\sqrt{2}$)

e: TR

f: p is scaled by (3, 2)

g: p is translated by (1, 2)

h: $\begin{bmatrix} 1/3 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

i: T(-1,-2)R(-45)

j: find (x,y) where x + 3y = 0

k: (34/29, 85/29)

**3. General camera model**

a: The general projection matrix can help transform objects from different coordinate systems in different situations.

b: $\begin{bmatrix} R^T & -R^T T \\ 0 & 1 \end{bmatrix}$

c: R = $[\hat{x},\ \hat{y},\ \hat{z}]$

d: R* and T* are the rotation and translation of world with respect to camera.

e: $\begin{bmatrix} k0 & 0 & 512 \\ 0 & kv & 512 \\ 0 & 0 & 1 \end{bmatrix}$

f: K* contains intrinsic parameters, [R*|T*] contains extrinsic parameters.

g: To make the camera model more accurate

h: The location of the original pixel changed in a non-linear way. The camera model more scale away from center.

i: Weak-perspective camera: M∞ is approximation to perspective camera's matrix M where the last raw is [0,0,0,1]. The parallel line of object appears to parallel each other.

Affine camera: A special case of projective camera. and is obtained by constraining the matrix such that the elements in the last row of the matrix are all zeros except the last one.

**4. Color and photometric image formation**

a: The surface radiance is the power of light per surface area reflected from the surface. The image irradiance is the power of light per surface area that are received at each pixel.

b: E(p) = L(p)$\frac{\pi}{4}(\frac{d}{f})^2\ cos\ \alpha^4$

c: Defined as the ratio of irradiance reflected to the irradiance received by a surface

d: That's how human perceive colors

e: gray

f: Mapping can be done using CIE conversion.

g: Y represents the relative luminance of the color as perceived by human eye.

I.e. perceived relative brightness.

h: Lab color is approximate to human vision. Colorimetric distances between the individual colors correspond to perceived color differences.

## 5. Noise and filtering

a: SNR = Es/En = σs2/σn2,

Where σs is the variance of pixels in a sequence of images, σn is variance in uniform area.

b. The Gaussian noise is statistical noise having a probability density function equal to that of the normal distribution. The impulsive noise is caused by sharp and sudden disturbances in the image signal. Median filter is better to impulsive noise because it can avoid outliers of noise.

c. (1*2)*9 = 18

d. get the filter equivalent to the two filters, that is the derivative filter D and the other one G, the result is the convolution of the two filters D*G.

e. zero padding, repeat padding, crap and so on

f.

(1/9)*

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

The sum of all entries should be 1. Otherwise the image intensity will become darker or lighter than before.

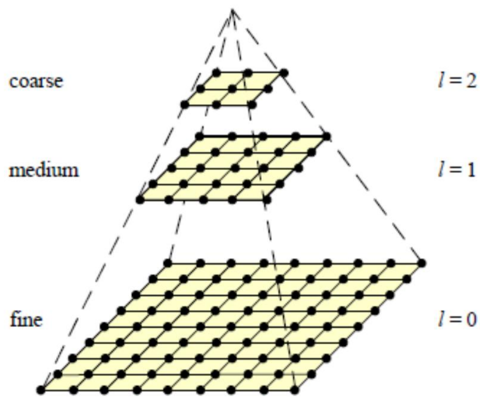
g. I * G(x,y) = G(y) * (I * G(x))

1D is more efficient. Because The 1D filter costs O(2MxWxm) instead of 2D filter costing O(MxWx$m^2$).

No, if the filter is not separable.

h. Size ≥ 5σ = 10

i.




The process of each layer is image ⟶ smoothing ⟶ downsample.

We can't make sure the size of window in detection, however, if we use Gaussian pyramid, we can use the same size of window with different size of image.

The cost of Gaussian pyramid is $m^2$ + $m^2/4$ + $m^2/16$ < $4m^2/3$, the cost of single image is $m^2$. So the amount of additional processing is $m^2/3$.

j. A Laplacian pyramid is very similar to a Gaussian pyramid but saves the difference image of the blurred versions between each level. Only the smallest level is not a difference image to enable reconstruction of the high resolution image using the difference images on higher levels. This technique can be used in image compression.

## 6. Edge detection

a. Because it can detect the change of image which can help get feature of image.

Properties: Edges are corresponding to scene elements. Invariment. Consistent detection.

b. Smoothing: smoothing filters convolve with image to reduce noise in image.

Enhancement: emphasizing pixels where there is a significant change in local intensity values and is usually performed by computing the gradient magnitude.

Localization: find the exact location of edge

c. forward difference filter and central difference filter, or Ix and Iy, or Sobel and LOG.

An image gradient is a directional change in the intensity or color in an image. The image gradient can detect the change of image such as edge detection, corner detection.

d. $I_x = I * G * (\partial/\partial x) \Rightarrow I * (G * (\partial/\partial x))$

$$\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & 1 \\ \hline \end{array} * \begin{array}{|c|c|}\hline -1 & 1 \\ \hline -1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$$

Gaussian smoothing x direction derivative x direction of sobel filter

e. Suppose we have the discrete 1D image f[x], we want to calculate the f'[x] which is the derivative of image. However, this image is discrete, we reconstruct this function to a continuous function f(x) = f[x]*h(x), we then calculate the derivative of this function as f[x]*h'(x). Finally we sample f[x]*h'(x) which is the derivative of image. h'(x) = Gx' which is the derivative of Gaussian function.

Gx = Gy =

(1/4.98)*

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.04 | 0.14 | 0.32 | 0.61 | 0.88 | 1 | 0.88 | 0.61 | 0.32 | 0.14 | 0.04 |

Gx’ = Gy’=

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.055 | 0.135 | 0.243 | 0.303 | 0.221 | 0 | -0.221 | -0.303 | -0.243 | -0.135 | -0.055 |

f. The first order derivative that is evaluate across an edge will capture a significant rise or fall in pixel values. The derivative will be at a max or min around this edge. A threshold can be used to identify that an edge is somewhere in the vicinity. Depending upon the threshold, it can be a rather wide vicinity.

The second order derivative captures the actual peak of the change of value. It is observed as a root, or zero-crossing, of the 2nd derivative. This gives a reliable and standardized way of declaring exactly where the edge is.

g.

| | | | | |
|---|---|---|---|---|
| 0.110 | 0.246 | 0.271 | 0.246 | 0.110 |
| 0.246 | 0 | -0.607 | 0 | 0.246 |
| 0.271 | -0.607 | -2 | -0.607 | 0.271 |
| 0.246 | 0 | -0.607 | 0 | 0.246 |
| 0.110 | 0.246 | 0.271 | 0.246 | 0.110 |

(1) compute LOG(convolve with LOG)

(2) Threshold to classify the value = 1 when $I*LOG > 0$ or value = 0 when $I*LOG \leq 0$

(3) Mark edge at transitions 0 to 1 and 1 to 0.

h. The Canny detection uses the directional derivative of image, and the standard detection uses the gradient of image.

If $|n| > \tau$, then detect edges as max of $\partial(I*G)/\partial n$.

Where n is $\nabla(I*G)$, $\tau$ is the threshold we set.

i. Non-Maximum Suppression: The image is scanned along the image gradient direction, and if pixels are not part of the local maxima they are set to zero. This has the effect of suppressing all image information that is not part of local maxima.

Hysteresis thresholding: we set two thresholds at first, τH and τL, where tH > τL.

(1) make array of visited pixels, v(i, j) = 0

(2) Scan image from left to right and top to bottom.

If !v(i,j) && $\nabla(I*G) > \tau H$:

track an edge using τL, set v(i,j) = 1

Track means searching neighbors orthogonal to gradient.