CZ4003 Computer Vision

Camera system

Imaging device: Screen, optical system (len), Aperture, Optical axis

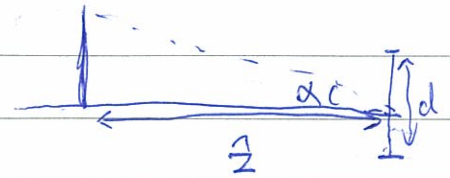
$\frac{1}{Z} + \frac{1}{z} = \frac{1}{f}$

Depth of field: range of depths that scene can be at to be acceptable

Smaller aperture: larger depth of field, more light

Scene radiance at P: $L(P) = \rho I^T n$

Image irradiance at p: $E = L \frac{\pi}{4} (d/z)^2 \cos^4 \alpha$



Exposure = irradiance × time

Spatial Enhancement: Point processing, Spatial Filtering

PP: $s = T(r)$

Image Negative: $s = L-1-r$, Contrast Stretching: $s = \begin{cases} 0 & r \le r_{min} \\ \frac{255(r-r_{min})}{r_{max}-r_{min}} & (r_{min} < r < r_{max}) \\ 255 & \text{else} \end{cases}$

Power-Law: $s = c r^\gamma$

Histogram: each bin: $\frac{MN}{L-1}$, num of pixels $\le k$, $\sum_{j=0}^{k} n_j$

bins can be filled $\left(\sum_{j=0}^{k} n_j\right) \div \frac{MN}{L-1} = \frac{L-1}{MN}\left(\sum_0^k n_j\right) = (L-1)\left(\sum_0^k P_j\right)$

SF: weighted sum: $g(x,y) = \sum_u \sum_v f(x+u, y+v) W(u,v)$ (correlation)

$= \sum_s \sum_t f(x-s, y-t) h(s,t)$ $\quad h(s,t) = W(-s,-t)$ convolution

~~Avera~~ average filter: $\frac{1}{9} \times \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$

gaussian filter: $\frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}$

median filter

ntusu voice • service • vibrancy

Frequency Enhancement : $F(u,v) = \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} f(x,y)\, e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)}$

$f(x,y) = \frac{1}{MN} \sum_{y=0}^{N-1} \sum_{u=0}^{M-1} F(u,v)\, e^{j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)}$

F : How much. $|F|$: How strong $\angle F$: How is shifted.

small and ~~high~~ large (u,v) : low frequency , median (u,v) : high f

After fftshift : small / large : high f ; median : low f

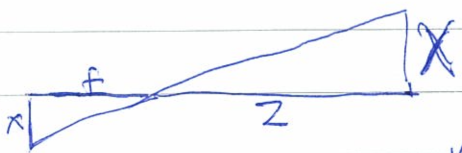
Lowpass filter : remove white noise

Gaussian lowpass filter : $e^{-\frac{u^2+v^2}{2\sigma^2}}$ σ : standard deviation , control cutoff f

Highpass filter : image sharpening, edge filtering

Gaussian highpass : $1 - e^{-\frac{u^2+v^2}{2\sigma^2}}$

Notch rejects a certain f. Bandreject rejects band



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

real 3D

$$\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} kx \\ ky \\ k \end{bmatrix}$$

image 2D

Rigid Transform : $P = R P_w + T$

rotate matrix, world vector, linear transform vector

Neater rigid transform :

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Camera to image : $\begin{cases} x_{im} = -x/s_x + O_x \\ y_{im} = -y/s_y + O_y \end{cases}$

$$\begin{bmatrix} x_{im} \\ y_{im} \\ 1 \end{bmatrix} = \begin{bmatrix} -\frac{1}{s_x} & 0 & O_x \\ 0 & -\frac{1}{s_y} & O_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Put all together :

$$\underbrace{\begin{bmatrix} kx_{im} \\ ky_{im} \\ k \end{bmatrix}}_{\text{image coordinates}} = \underbrace{\begin{bmatrix} -\frac{1}{s_x} & 0 & O_x \\ 0 & -\frac{1}{s_y} & O_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{camera to image frame}} \underbrace{\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{perspective}} \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

ntusu voice • service • vibrancy

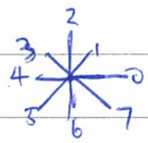
plane to plane: $\begin{bmatrix} kx_{im} \\ ky_{im} \\ k \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}$

$$\begin{bmatrix} kx_{im} \\ ky_{im} \\ k \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$ set $m_{34} = 1$

There's a scale k, so we can set one of $m_{ij} = 1$.
(Actually we can set $k=1$ ... make it more intuitive.
n points → 2n equations. Need 6 points to solve.

To calibrate cameras: Use calibration chart or cube.

Edge processing Path encoding: 1. store the pixels (x,y) in array or linked list (Edgel sequence)
2. Chain codes: 0-7 8 directions in total



Edgel sequence: Cost more space
Chain codes: Cost more time to get coordinates

Prewitt filter: $\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$ sobel filter: $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

Absolute gradient magnitude: $\nabla f = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y|$
$h(r) = -e^{-\frac{r^2}{2\sigma^2}}$ (inversed) $\nabla^2 h(r) = -\left[\frac{r^2-\sigma^2}{\sigma^4}\right] e^{-\frac{r^2}{2\sigma^2}}$
Laplacian of Gaussian (LoG) filter: find the zero-crossing in output
Less sensitive to noise due to G window
Canny Edge Detector:
1. LoG filtering
2. Non-maximal suppression: suppress the point if not local maxima in gradient direction
3. Hysteresis Thresholding: t_H: set to 1 if higher than t_H
t_L: set to 1 if between t_L, t_H and have neighbouring pixel ~~set to 1~~ perpendicular to gradient set to 1

voice • service • vibrancy
ntusu

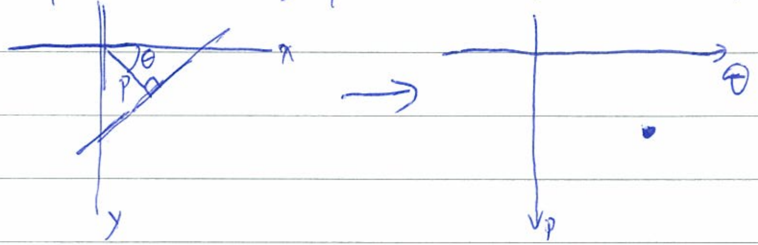
Polygonal Line Fitting:

Recursive subdivision algorithm:

1. Connect straight line section connecting end points
2. Find edgel with largest distance to any lines
3. Stop when distance is less than threshold
4. Draw new lines

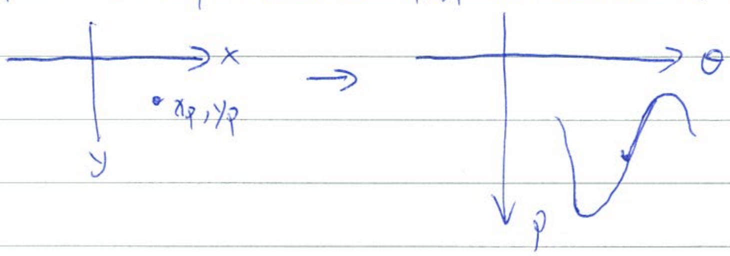
Hough Transform

~~from~~ Map a line in x,y plane to a point in θ, p plane :



Line equation: $p = x_p \cos\theta + y_p \sin\theta = \sqrt{x_p^2 + y_p^2}\,\sin(\theta + \tan^{-1}\frac{x_p}{y_p})$

Map all lines ~~p~~ intersect x_p, y_p to curve in θ, p plane:



hough transform : Transform all edge points into sin curve in p, θ plane. Find the points in p, θ plane where most of curves intersect. It corresponds to the line in x, y plane

voice • service • vibrancy
ntusu

Image Region Processing:

Region encoding $\begin{cases} \text{Labeled mask overlay} \\ \text{Quadtree representation} \\ \text{Boundary encoding} \end{cases}$

Labeled mask overlay: Map the pixel in original image to a colorful mask

Quadtree: Recursively divides a square region into 4 quadrants. Stop ~~with~~ when all pixels in ~~comm~~ subregion has common label.

Boundary encoding: Define the boundary of each region

Mean - average pixel gray-level: $\mu = \frac{1}{N} \sum f(x,y) = \sum_{0}^{255} r\, p(r)$

Variance - $\sigma^2 = \frac{1}{N} \sum (f(x,y) - \mu)^2 = \sum_{r=0}^{255} (r-\mu)^2 p(r)$

Naïve Gray-level thresholding: use average value or 127

Otsu Method: Minimizing withing group variance thresholding

Within-group variance: $\sigma_w^2 = q_L(t)\sigma_L^2(t) + q_H(t)\sigma_H^2(t)$

$$q_L(t) = \sum_{r=0}^{t} p(r) \qquad q_H(t) = \sum_{r=t+1}^{255} p(r)$$

Texture representation: structural and ~~statiscal~~ statiscal

Structral: hard to extract texture.

Statiscal: neighborhood mean / variance

Co-occurrence Matrix C_d: dimensions = number of gray-levels

d: displacement vector indicating pairwise relationship

$C_d[r,c]$ = number of a pixel with gray-level g_r occurs ~~toget~~ at (x,y) together with a pixel with gray-level g_c at $(x+dx, y+dy)$

Normalized C_d: ~~N_d~~ $N_d[r,c] = \frac{C_d[r,c]}{\sum_R \sum_C C_d[R,C]}$

Energy $= \sum_r \sum_c N_d^2[r,c]$, Entropy $= \sum_r \sum_c N_d[r,c] \log_2 N_d[r,c]$

Contrast $= \sum_r \sum_c (r-c)^2 N_d[r,c]$, Homogeneity $= \sum_r \sum_c \frac{N_d[r,c]}{1+(r-c)}$

Gabor filters bank to get different statistical features and texture vector

Euclidean distance $D(T_1, T_2) = \|T_1 - T_2\|$

K-means and Quadtree Decomposition for clustering

Covariance of feature vectors: mean $\mu = \frac{1}{N} \sum_1^N T_i$

$$S = \frac{1}{N} \sum_1^N (T_i - \mu)(T_i - \mu)^T$$

The largest eigenvalue e_{max} of S is used to measure ~~the~~ distance between vectors.

After spliting, merge!

Object Recognition :

Levels of recognition : Image categorization, detection / localization, segmentatio

Challenges : view point variation, illumination, ~~occal~~ occlusion, scale, deformation, background clutter, intra-class variation

K-means :

select k initial points

repeat :

Assign all points to closest centroid

Recompute centroid to each cluster

until centroids don't change much

Solution to initial points :

1、 multiple runs

2. hierarchical clustering

3、 select more than k centroids, then select k from them

Evaluation for K-means :

Cohesion, measured by ~~cluster~~ within-cluster sum of square :

$$WSS = \sum_i \sum_{x \in C_i} (x - m_i)^2$$

Separation, measured by between-cluster sum of square :

$$BSS = \sum_i |C_i| (m - m_i)^2$$

Classification :

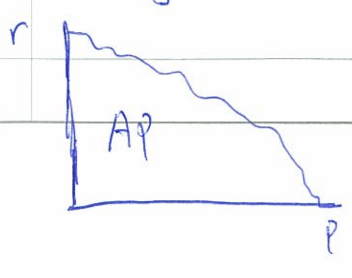
precision = $TP / (TP + FP)$ recall = $TP / (TP + FN)$

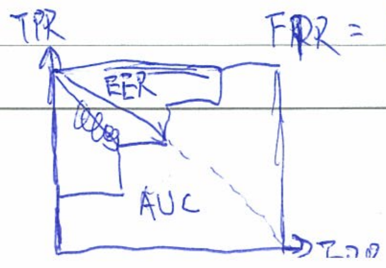
F-measure = $2pr / (p + r)$

Average Precision

ROC curve : TPR = recall = $TP / (TP + FN)$

FPR = $FP / (FP + TN)$





voice • service • vibrancy
ntusu

(Equal Error Rate)

EER: The point on ROC having same false positive and false ~~neg~~ negative rate

Area ~~Ut~~ Under Curve (AUC): The higher the better

Bag-of-words model:

Region selection → region description → vector quantization → histogram → classifier

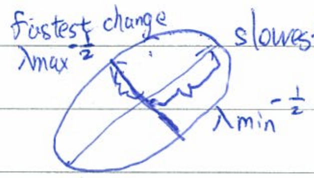
Harris Corner Detector:

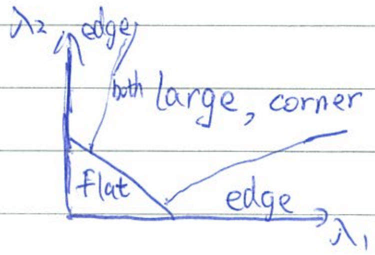
$$E(u,v) = \sum_{x,y} w(x,y)\,[I(x+u, y+v) - I(x,y)]^2 \quad \left(w(x,y) = e^{-\frac{r^2}{2\sigma^2}}\right)$$

$$\approx \sum_{x,y} w\,[uI_x + vI_y]^2$$

$$= [u, v]\; M \begin{bmatrix} u \\ v \end{bmatrix} \quad \left(M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}\right)$$

λ_1, λ_2 eigenvalues of M,





$R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2 \quad (k = 0.04 - 0.06)$

large positive: corner small: flat

large negative: edge

Algorithm: Find points with large R, take local maxima

Properties: Rotation invariance. Intensity ~~invariance~~ change invariance

non-invariant to image scale change

SIFT: divide sample image into 4X4 subregions.

For each region, generate the histogram of gradient in 8 orientation

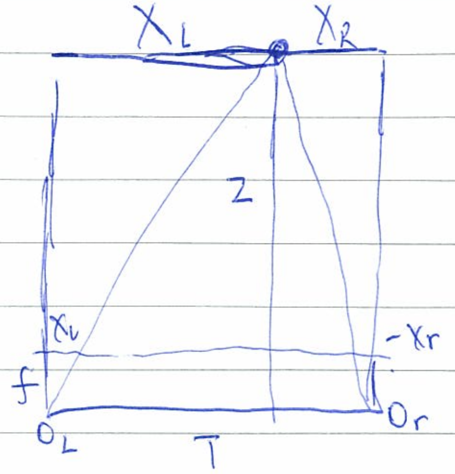
Bag-of-words pros ~~and cons~~:

flexible to geometry / deformation / view points

compact summary, vector representation, good result

cons: ignore geometry, optimal formation remains unclear

3D stereo vision :



$$x_L = X_L \frac{f}{Z}, \quad x_r = X_R \frac{f}{Z}, \quad (x_l - x_r) = (X_L - X_R)\frac{f}{Z} = \frac{Tf}{Z}$$

$$Z = \frac{Tf}{d}, \quad \left|\frac{\delta Z}{\delta d}\right| = \left|\frac{Tf}{d^2}\right| = \frac{Z^2}{Tf} \Rightarrow \infty \text{ when } T \Rightarrow 0$$

Matching : Appearance based , feature based

~~Mini~~ Appearance - based : Minimize sum-of-square Difference (SSD)

$$\arg\min_{(x,y)} \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} \left(I(x+u, y+v) - g(u,v)\right)^2$$

$$\Downarrow$$

$$\arg\max_{(x,y)} 2 \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} I(x+u, y+v)\, g(u,v) - \sum\sum I(x+u, y+v)^2$$

partition left image into patches, find corresponding patches in right image minimizing SSD

Feature based matching :

Find candidates by special property

Use heuristic to find correct solution

Reconstruction :

L cam : $\begin{bmatrix} kx_l \\ ky_l \\ k \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_5 & a_6 & a_7 & a_8 \\ a_9 & a_{10} & a_{11} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$

R cam : $\begin{bmatrix} mx_r \\ my_r \\ m \end{bmatrix} = \begin{bmatrix} b_1 & \cdots & \\ & \cdots & \\ & \cdots & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$

$$\underset{(3\times3)}{W} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \underset{(4\times1)}{q}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = W^+ q = (W^T W)^{-1} W^T q$$

SVM Target : find $y = f(x)$ classify dataset $\{X_i\}$

Expected risk $R[f] = C(f(x), y)\, d(x, y)$ C could be squared error

Empirical risk $R_{emp}[f] = \frac{1}{N} \sum_{i=1}^{N} C(f(x_i), y_i)$

~~SW~~ SVM : define a hyperplane to seperate dataset

Margin : minimum distance of a point to the plane, idealy being, as large as possible

Larger margin : over learning, cannot tune

minimize $J(w) = \frac{1}{2} ||w||^2$ using lagrangian optimization