CMSC5711 Formula sheet: Ch2. Camera model: essential Formulas
Ch3. Edge detection: essential Formulas, ***Ch4. Harris corner feature extraction***

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}_{(3\times1)} = M_{ext(3\times4)} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}_{(4\times1)}$$

$$M_{ext(3\times4)} = R_{c_{(3\times3)}} [I_3 | -T_c]_{(3\times4)}$$

$$\begin{bmatrix} s*u \\ s*v \\ s \end{bmatrix}_{(3\times1)} = M_{int(3\times3)} * M_{ext(3\times4)} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}_{(4\times1)}$$

Rodrigues formula

$$R=\cos\theta I+(I-\cos\theta)rr^T+\sin\theta \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix},$$

```
Rcamx=[1   0              0
       0   cos(theta_x)   -sin(theta_x)
       0   sin(theta_x)    cos(theta_x) ];

Rcamy=[cos(theta_y)     0   sin(theta_y)
       0                1   0
       -sin(theta_y)    0   cos(theta_y)];

Rcamz=[cos(theta_z)    -sin(theta_z)   0
       sin(theta_z)     cos(theta_z)   0
       0                0              1 ];
```

1st order edge gradient ,

edge if G(I(x,y)>> $Threshold$.

$$|G(I(x,y))| \approx \sqrt{\left(\frac{\partial I(x,y)}{\partial x}\right)^2 + \left(\frac{\partial I(x,y)}{\partial y}\right)^2}$$

2nd order edge (Laplacian operator)

$$\nabla^2 I(x,y) = \frac{\partial^2 I(x,y)}{\partial x^2} + \frac{\partial^2 I(x,y)}{\partial y^2} = 0,$$

$$2-\text{D Gaussian: G(x,y)} = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-m_x)^2+(y-m_y)^2}{2\sigma^2}}$$

Harris matrix (structure tensor): $A(x,y) =$

$$\begin{bmatrix} \sum_{i,j}^{\text{all x,y in N}} \left(\frac{\partial I(x,y)}{\partial x}\right)^2 & \sum_{i,j}^{\text{all x,y in N}} \left(\frac{\partial I(x,y)}{\partial x}\right)\left(\frac{\partial I(x,y)}{\partial y}\right) \\ \sum_{i,j}^{\text{all x,y in N}} \left(\frac{\partial I(x,y)}{\partial x}\right)\left(\frac{\partial I(x,y)}{\partial y}\right) & \sum_{i,j}^{\text{all x,y in N}} \left(\frac{\partial I(x,y)}{\partial y}\right)^2 \end{bmatrix}$$

Ch8. Stereo vision

$$[T]_x = \begin{bmatrix}\begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}\end{bmatrix}_x = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}$$

$[T_A]_x T_B = T_A \times T_B$ by definition

$X_2 = RX_1 + T$, To rotate (R) and translate (T) a vector from X1 to X2.
hence $X_2^T * E * X_1 = 0$
where essential matrix $E = [T]_x R$

For a stereo setup, a pixel at the reference (left) image is (u1,v2) , the right hand image point is (u2,v2), , F *is the* fundamental matrix, such that
$[u_2 \quad v_2 \quad 1] * F * [u_1 \quad v_1 \quad 1]^T = 0$

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}, \textit{where}$$

$$u_2u_1f_{11} + u_2v_1f_{12} + u_2f_{13} + v_2u_1f_{21} + v_2v_1f_{22} + v_2f_{23} + u_1f_{31} + v_1f_{32} + f_{33} = 0$$

If Ax=0, *[U,S,V]=SVD(A), last column of V is x.*

Normalization: Average distance of [u,v] around the center [0,0] is sqrt(2)

F*e1=0, where e1=epipole of left image
$x_1 = P_1X, x_2 = P_2X$,
$P_i = M_{int}[R_i|T_i]$,
R=cam. rotation, T=cam. translation
X=model point, x=image point
projection matrices $(P_1, P_2)$of 2 cams

$$AX = \begin{bmatrix} u_1P_1^{3T} - P_1^{1T} \\ v_1p_1^{3T} - P_1^{2T} \\ u_2P_2^{3T} - P_2^{1T} \\ v_2P_2^{3T} - P_2^{2T} \end{bmatrix} X = 0,$$

where $p^{iT} = i^{th}$ row of P

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, Z = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$[U,S,V] = SVD(E)$

$[T]_x = UZU^T$

$R = UWV^T$

Ch4: Normalized cross correlation coefficient

$$r_{f,f'} = \frac{\sum_{x,y\in s}\left(f_{x,y} - \bar{f}\right)\left(f'_{x,y} - \overline{f'}\right)}{\sqrt{\sum_{x,y\in s}\left(f_{x,y} - \bar{f}\right)^2 \sum_{x,y\in s}\left(f'_{x,y} - \overline{f'}\right)^2}}$$

$$s = all\_range, \bar{f} = mean(f), \overline{f'} = mean(f')$$

Ch5. Hough transform: $y = mx + c$

$$m = \left(-\frac{\cos(\theta)}{\sin(\theta)}\right), c = \left(\frac{r}{\sin(\theta)}\right),$$

hence $y_i = \left(-\frac{\cos(\theta)}{\sin(\theta)}\right)x_i + \left(\frac{r}{\sin(\theta)}\right)$,

so $r = \sin(\theta)y_i + \cos(\theta)x_i$

Ch6: ---- Histogram equalization -------

$$p(r_k) = \frac{n_k}{MN}, s_k = \frac{L-1}{MN}\sum_{j=0}^{k} n_j$$

-----Color model: RGB to HSV,HSL,HIS-----

$$S_{HSV} = \begin{cases} 0, & if\ M = 0 \\ C/V, & otherwise \end{cases}$$

$$S_{HSL} = \begin{cases} 0, & if\ C = 0 \\ C/2L, & if\ L \le 1/2 \\ C/(2-2L), & if\ L \rangle 1/2 \end{cases}$$

$$S_{HSI} = \begin{cases} 0, & if\ I = 0 \\ 1 - \frac{m}{I}, & otherwise \end{cases}$$

$M = Max(R,G,B)$
$m = Min(R,G,B)$
$C = M - m$
$I = \left({}^1/_3\right)(R + G + B)$
$V = M$
$L = \left({}^1/_2\right)(M + m)$

Ch7: The Epanechnikov $k_E\left(\left\|\frac{x-x_i}{h}\right\|^2\right) = \begin{cases} c\left(1 - \left\|\frac{x-x_i}{h}\right\|^2\right) & \left\|\frac{x-x_i}{h}\right\| \le 1 \\ 0 & \text{otherwise} \end{cases}$

$$P(x) = \frac{c_{k,d}}{nh^d}\sum_{i=1}^n K\left(\frac{x-x_i}{h}\right),\ \nabla P(x) = \frac{2c_{k,d}}{nh^{d+2}}\left[\sum_{i=1}^n g_i\left(\left\|\frac{x-x_i}{h}\right\|^2\right)\right]\left[\frac{\sum_{i=1}^n x_i g_i\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^n g_i\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} - x\right]$$

Where $g_i\left(\left\|\frac{x-x_i}{h}\right\|^2\right) = \begin{cases} c & \text{if } \left\|\frac{x-x_i}{h}\right\| \le 1 \\ 0 & \text{otherwise} \end{cases}$

Ch9. Factorization

$$\text{Affine } P_{affine} = \begin{bmatrix} a_1 & a_2 & a_3 & t_1 \\ b_1 & b_2 & b_3 & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} M_{2\times3} & t_{2\times1} \\ 0 & 1 \end{bmatrix}$$

(u,v)=real pixel position, (u'v')= affine translation eliminated pixel.

(X Y Z)' = model point, *x*= projected image.

M = affine rotation , *t*=affine translation,

$$x = \begin{pmatrix} u \\ v \end{pmatrix} = M\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + t,\ \min_{M^i, X_j} \sum_{i,j}\left\|W - (M^iX_j)\right\|^2$$

$$W_{(2\Gamma\times n)} = \begin{bmatrix} u'^1_1 & u'^1_2 & \dots & u'^1_n \\ \vdots & \vdots & \vdots & \vdots \\ u'^\Gamma_1 & u'^\Gamma_2 & \dots & u'^\Gamma_n \\ v'^1_1 & v'^1_2 & \dots & v'^1_n \\ \vdots & \vdots & \vdots & \vdots \\ v'^\Gamma_1 & v'^\Gamma_2 & \dots & v'^\Gamma_n \end{bmatrix}_{(2\Gamma\times n)}$$

where $W_{(2\Gamma\times n)} = \quad MX =$

$$= \begin{bmatrix} a_{1,i=1} & a_{2,i=1} & a_{3,i=1} \\ \vdots & \vdots & \vdots \\ a_{1,i=\Gamma} & a_{2,i=\Gamma} & a_{3,i=\Gamma} \\ b_{1,i=1} & b_{2,i=1} & b_{3,i=1} \\ \vdots & \vdots & \vdots \\ b_{1,i=\Gamma} & b_{2,i=\Gamma} & b_{3,i=\Gamma} \end{bmatrix}_{2\Gamma\times3}$$

Ch10 linear pose estimation, Barycentric homogenous coord.

$P_i^w = \sum_{j=1}^{4} \alpha_{ij}\, c_j^w$, with $\sum_{j=1}^{4} \alpha_{ij} = 1$,

$$\begin{bmatrix}\begin{bmatrix}\alpha_{i=1,j=1}\\ \alpha_{i=1,j=2}\\ \alpha_{i=1,j=3}\\ \alpha_{i=1,j=4}\end{bmatrix} \begin{bmatrix}\alpha_{i=2,j=1}\\ \alpha_{i=2,j=2}\\ \alpha_{i=2,j=3}\\ \alpha_{i=2,j=4}\end{bmatrix} \;..\; \begin{bmatrix}\alpha_{i=n,j=1}\\ \alpha_{i=n,j=2}\\ \alpha_{i=n,j=3}\\ \alpha_{i=n,j=4}\end{bmatrix}\end{bmatrix}_{(4\text{x}n)} = \begin{bmatrix} c^w_{x,j=1} & c^w_{x,j=2} & c^w_{x,j=3} & c^w_{x,j=4}\\ c^w_{y,j=1} & c^w_{y=2} & c^w_{y,j=3} & c^w_{y,j=4}\\ c^w_{z,j=1} & c^w_{z,j=2} & c^w_{z,j=3} & c^w_{z,j=4}\\ 1 & 1 & 1 & 1\end{bmatrix}^{-1}_{(4\text{x}1)} \begin{bmatrix}\begin{bmatrix}P^w_{x,i=1}\\ P^w_{y,i=1}\\ P^w_{z,i=1}\\ 1\end{bmatrix}\begin{bmatrix}P^w_{x,i=2}\\ P^w_{y,i=2}\\ P^w_{z,i=2}\\ 1\end{bmatrix} \cdots \begin{bmatrix}P^w_{x,i=n}\\ P^w_{y,i=n}\\ P^w_{z,i=n}\\ 1\end{bmatrix}\end{bmatrix}_{(4\text{x}n)}$$

α_{ij} = homogeneous barycentric coord. can be estimated

2D case $\forall i, (i = 1,2,..,n$ and $j = 1,2,3,4)$. A = camera intrinsic parameters

$$w_i \begin{bmatrix}U_i\\1\end{bmatrix} = AP_i^c = A\sum_{j=1}^{4} \alpha_{ij}\, c_j^c, \text{where } U_i = \begin{bmatrix}u_i\\v_i\end{bmatrix}, A = \begin{bmatrix} f_u & & u_{c\_ox}\\ & f_v & v_{c\_oy}\\ & & 1\end{bmatrix}, \text{then}$$

$$w_i \begin{bmatrix}u_i\\v_i\\1\end{bmatrix} = \begin{bmatrix} f_u & & u_{c\_ox}\\ & f_v & v_{c\_oy}\\ & & 1\end{bmatrix} \sum_{j=1}^{4} \alpha_{ij} \begin{bmatrix}x_j^c\\ y_j^c\\ z_j^c\end{bmatrix},$$

$\sum_{j=1}^{4} \alpha_{ij}\, f_u x_j^c + \alpha_{ij}(u_{c\_ox} - u_i) z_j^c = 0,\;\; \sum_{j=1}^{4} \alpha_{ij}\, f_v y_j^c + \alpha_{ij}(v_{c\_oy} - v_i) z_j^c = 0,$

For feature *i, a1=Alph(i,1); a2=Alph(i,2); a3=Alph(i,3); a4=Alph(i,4); where Alph=* α

See code compute_M_ver2.m of https://www.epfl.ch/labs/cvlab/wp-content/uploads/2018/08/EPnP_matlab.zip

$M\_i_{(2x12)}$=

[a1*fu, 0, a1*(u0-ui), a2*fu, 0, a2*(u0-ui), a3*fu, 0, a3*(u0-ui), a4*fu, 0, a4*(u0-ui);

0, a1*fv, a1*(v0-vi), 0, a2*fv, a2*(v0-vi), 0, a3*fv, a3*(v0-vi), 0, a4*fv, a4 *(u0-ui)]

$M_{(2nx12)}=[M\_i=1_{(2x12)}, M\_i=2_{(2x12)},..., M\_i=n_{(2x12)}]^T$

$M{*}C^c_{all\ j}=0$, where

$C^c_{all\ j} = [x^c_{j=1}\;\; y^c_{j=1}\;\; z^c_{j=1}\;\; x^c_{j=2}\;\; y^c_{j=2}\;\; z^c_{j=2}\;\; x^c_{j=3}\;\; y^c_{j=3}\;\; z^c_{j=3}\;\; x^c_{j=4}\;\; y^c_{j=4}\;\; z^c_{j=4}]^T$

Ch11. Iterative pose estimation

Ch12 Bundel adjustment

- $\begin{bmatrix}u_i\\v_i\end{bmatrix} = \left[f\left(\frac{X_{i'}}{Z_{i'}}\right), f\left(\frac{Y_{i'}}{Z_{i'}}\right)\right]^T$, f =focal length
- $u_i = f\frac{r_{11}X_i + r_{12}Y_i + r_{13}Z_i + T_1}{r_{31}X_i + r_{32}Y_i + r_{33}Z_i + T_3}$
- $v_i = f\frac{r_{21}X_i + r_{22}Y_i + r_{23}Z_i + T_2}{r_{31}X_i + r_{32}Y_i + r_{33}Z_i + T_3}$
- Rotation is R = $\begin{bmatrix} r_{11} & r_{12} & r_{13}\\ r_{21} & r_{22} & r_{23}\\ r_{31} & r_{32} & r_{33}\end{bmatrix} \approx \begin{bmatrix} 1 & -\varphi_3 & \varphi_2\\ \varphi_3 & 1 & -\varphi_1\\ -\varphi_2 & \varphi_1 & 1\end{bmatrix}$,

  $\varphi_1, \varphi_2, \varphi_3$ are small angles rotate aginst X, Y, Z axes resp.
- Translation T =[T1,T2,T3]^T
- $[X_i', Y_i', Z_i']$ *is a 3D poiin, image is at* $\begin{bmatrix}u_i\\v_i\end{bmatrix}$

Ch14: Camera calibration and Homography

$$\begin{bmatrix}u'_{i,j}\\ v'_{i,j}\\ s\end{bmatrix} = \begin{bmatrix}H_{1,1} & H_{1,2} & H_{1,3}\\ H_{2,1} & H_{2,2} & H_{2,3}\\ H_{3,1} & H_{3,2} & H_{3,3}\end{bmatrix}\begin{bmatrix}X_j\\Y_j\\1\end{bmatrix}, \begin{bmatrix}X_j\\Y_j\\1\end{bmatrix}$$

where $[X_j, Y_j]^T$ is on a planner surface, $[u',v']^T$ is the 2D projection

$$u_{i,j} = \frac{u'_{i,j}}{s} = \frac{H_{1,1}\cdot X_j + H_{1,2}\cdot Y_j + H_{1,3}\cdot 1}{H_{3,1}\cdot X_j + H_{3,2}\cdot Y_j + H_{3,3}\cdot 1}$$

$$v_{i,j} = \frac{v'_{i,j}}{s} = \frac{H_{2,1}\cdot X_j + H_{2,2}\cdot Y_j + H_{2,3}\cdot 1}{H_{3,1}\cdot X_j + H_{3,2}\cdot Y_j + H_{3,3}\cdot 1}$$

Eliminate the term s

$u_{i,j}(H_{3,1}\cdot X_j + H_{3,2}\cdot Y_j + H_{3,3})\text{-}(H_{1,1}\cdot X_j + H_{1,2}\cdot Y_j + H_{1,3}) = 0$

$v_{i,j}(H_{3,1}\cdot X_j + H_{3,2}\cdot Y_j + H_{3,3})\text{-}(H_{2,1}\cdot X_j + H_{2,2}\cdot Y_j + H_{2,3}) = 0$

Set $h = [H_{1,1}, H_{1,2}, H_{1,3}, H_{2,1}, H_{2,2}, H_{2,2}, H_{3,1}, H_{3,2}, H_{3,3}]^T$,

$$\begin{bmatrix} -X_j & -Y_j & -1 & 0 & 0 & 0 & u_{i=1,j}X_j & u_{i=1,j}Y_j & u_{i=1,j}\\ 0 & 0 & 0 & -X_j & -Y_j & -1 & v_{i=1,j}X_j & u_{i=1,j}Y_j & v_{i=1,j}\\ -X_j & -Y_j & -1 & 0 & 0 & 0 & u_{i=2j}X_j & u_{i=2,j}Y_j & u_{i=2,j}\\ 0 & 0 & 0 & -X_j & -Y_j & -1 & v_{i=2,j}X_j & u_{i=2,j}Y_j & v_{i-=2,j}\\ & & & & & \vdots & & & \\ -X_j & -Y_j & -1 & 0 & 0 & 0 & u_{i=N,j}X_j & u_{i=N,j}Y_j & u_{i=N,j}\\ 0 & 0 & 0 & -X_j & -Y_j & -1 & v_{i=N,j}X_j & u_{i=N,j}Y_j & v_{i=N,j} \end{bmatrix}_{2N\times 9} \cdot h = \begin{bmatrix}0\\0\\ \vdots\\0\end{bmatrix}_{2N\times 1}$$

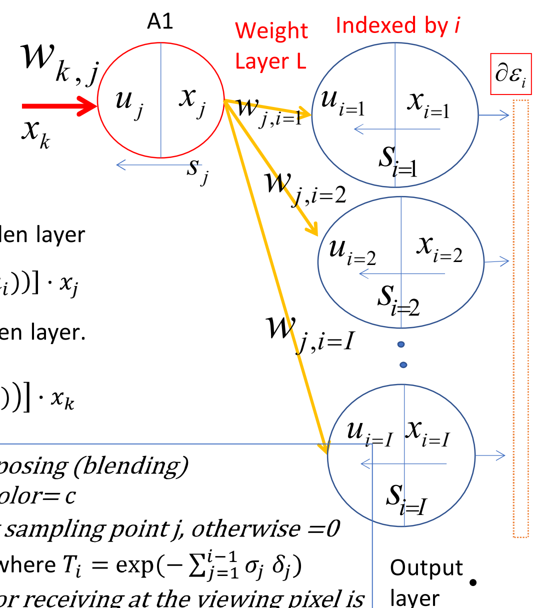
Ch15 : Neural networks useful formulas

$\text{Sigmoid\_}f(u) = \frac{1}{1+e^{-u}}$

$\text{Tanh\_}(u) = \frac{\sinh(u)}{\cosh(u)} = \frac{e^{u}-e^{-u}}{e^{u}+e^{-u}},$

$\text{Relu\_}f(u) = \max(0, u),$

$y_i = \text{SoftMax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^{K}\exp(z_j)},$



Case 1: neuron in between output and hidden layer

$$\frac{\partial\varepsilon}{\partial w_{j,i}} = s_i\cdot x_j = [(x_i - t_i)\cdot f(u_i)(1 - f(u_i))]\cdot x_j$$

Case 2: neuron between a hidden and hidden layer.

$$\frac{\partial\varepsilon}{\partial w_{k,j}} = \left(\sum_{i=1}^{i=I}(s_i\cdot w_{j,i})\right)\cdot[f(u_j)(1 - f(u_j))]\cdot x_k$$

Ch18 :Radiance field rendering, Alpha composing (blending)

Opacity= α*, Density=* σ*, Transmission=t, Color= c*

$\alpha = 1 - \exp(-\sigma\cdot\delta_j)$; $\alpha = 1 - T$; $\delta_j = 1$ *at sampling point j, otherwise =0*

$C(ray) = \sum_{i=1}^{N=1} T_i * (1 - \exp(-\sigma_j\, \delta_j)) * c_i$, where $T_i = \exp(-\sum_{j=1}^{i-1}\sigma_j\, \delta_j)$

At sampling point i=1,2,...N of a ray, the color receiving at the viewing pixel is

$C_{total} = \alpha_1 c_1 + (1-\alpha_1)\alpha_2 c_2 + (1-\alpha_1)(1-\alpha_2)\alpha_3 c_3 \ldots\ldots (1-\alpha_{N-1}) c_N$