Assignment - 5

* Name = Parthkumar Patel
* Student ID = A20416508
* Course Number = CS - 512
* Semester = Fall 2018

Que - 1

(a) Outlier is a noise point that is distant from other points. Points which are closely to each other, they fit nicely to model. The problem associated with outlier is that when the model is fit considering the outlier it results in a wrong solution.

(b)

→ Objective function used for robust estimation is =

$$E(\theta) = \sum_{i=1}^{N} \rho_\sigma(d(x_i, \theta))$$

In the standard least square objective function is -

$$E(\theta) = \sum_{i=1}^{N} d(x_i, \theta)^2.$$

We take the individual errors and square them $\rho_\sigma(x) = x^2$ where as in the robust estimation $\rho_\sigma(x) = \frac{x^2}{x^2+\sigma^2}$

(c)

→ German-Mcclure function $\rho_\sigma(x) = \frac{x^2}{x^2+\sigma^2}$.

* Advantage =

→ In german-Mcclure function has a upper bound. In german-mcclure in the begining it matters if you give bigger value outlier. it will give bigger penalty but after some point, if give higher value outliers. but it gives maximum value that we kept. Your point which has more error will not make contribution more than 1 to the total sum.

→ if we choose a large σ we may include outliers but if it is too small we may not include enough points so we can estimate σ as $\sigma = 1.5$.

(d) principle of the RANSAC algorithm is to use minimum number of points to fit the model and repeat this process several times and choose the best model after many trials.
number of ~~attempts~~ points drawn at each attempt should be small because there are less chances of getting outliers and at least in one of ~~any~~ many trials will lead to a better model.

(e)
→ The parameter of the RANSAC are
n = number of points at each evaluation
d = minimum number of points needed to estimate model.
K = number of trials.
t = distance threshold to identify inliers.
w = probability that a point is inlier.

$$K = \frac{\log(1-p)}{\log(1-w^n)}$$

(f) segmentation separate foreground and background.. In other words it is the process of partitioning a digital image into multiple segments.

→ Merge approach = start with each pixels in separate cluster iteratively merge cluster.

→ split approach = start with all pixels in one cluster iteratively split cluster.

(g)

→ K-means =

→ select k.

→ select intial guess of k-means; $m_1, \ldots, m_k$

→ Assign $l_i = \underset{j \in [1,K]}{\operatorname{argmin}} \|f_i - m_j\|^2$ for each pixels & assign correspondent cluster.

→ recompute mean. $m_j = \dfrac{\sum_{i \in j} f_i}{\# S_j}$.

→ Stop when m_j doesnot change.

Dt: / / Pg.no:

* Mixture of gaussians segmentation =

→ It is like k-means with additional parameters.

→ Replacing $d = \|f_i - m_j\|^2$ with $d = (f_i - m_j)^T \cdot \Sigma_j^{-1} (f_i - m_j)$.

$$m_j = \frac{\sum_{i \in j} f_i}{\# S_j} \qquad \Sigma_j = \frac{\Sigma (f_i - m_j)(f_i - m_j)}{\# S_j}.$$

$$\underset{\text{mahalnobis distance}}{d} = (f_i - \underset{\text{mean of cluster}}{m_j})^T \underset{\text{covariance of cluster.}}{\Sigma_j^{-1}} (f_i - m_j)$$

(h)

*) mean-shift algorithm for segmentation

→ mean-shift algorithm is similar to k-means.

→ give a weight to each sample.

$$m_j = \frac{\sum_{i \in j} w(f_i - m_j) f_i}{\sum_{j \in j} w(f_i - m_j)}$$

→ The closest a sample is to the mean, more weighted ~~or~~ effect to the mean. Mean shift find cluster centers as peaks of histogram.

Que (2)

(a)

→ **Forward projection** = given a 3D world point P project into the image using the projection matrix M.

→ **Calibration** = given P_i world points & corresponding p_i image points find M & the intrinsic ~~of~~ and extrinsic parameters.

→ **Reconstruction** = given image point p_i, compute world point P.

→ Forward projection is easiest.
→ Reconstruction is most difficult.

(b)

→ A set of 3D world points and its image 2D corresponding points.

(c)

→ (1) estimate projection matrix M.
→ (2) find parameters (K^*, R^*, T^*)

(d)

$\rightarrow P_i = MP_i = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 0 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 18 \\ 14 \\ 7 \end{bmatrix}$

$= \begin{bmatrix} 18/7 \\ 14/7 \end{bmatrix}$

$= \begin{bmatrix} 18/7 \\ 2 \end{bmatrix}$

(e)

$\rightarrow P_i = MP_i = \underset{2DH}{\begin{bmatrix} 100 \\ 200 \\ 1 \end{bmatrix}} = \underset{3\times 4}{\begin{bmatrix} 10 & 20 & 10 & 20 \\ 20 & 20 & 40 & 20 \end{bmatrix}} \underset{3DH}{\begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix}}$

(f)

→ we need atleast 6 points to get matrix m because we need to solve ~~twelve~~ 11 unknown. in order to solve it we need 6 points to get 12 equations.

(g)

→ The principal that is used to extract the unknown parameters from the projection matrix m is,

$m = R^*[K^* | T^*]$

we use the fact that the rotation matrix ~~has~~ has the orthogonal vectors along the rows.

→ we exploit this fact by taking the dot product and cross product of the rows n the M, thereby cancelling out some unknowns.

(h)

→ $$E(K^*, R^*, T^*) = \sum_{i=1}^{n} \left(x_i - \frac{M_1^T P_i}{M_3^T P_i} \right)^2 + \left(y_i - \frac{M_2^T P_i}{M_3^T P_i} \right)^2$$

(i)

→ Planner contribution steps =

(1) estimate 2D Homography between calibration plane and image.

(2) estimate intrinsic parameters.

(3) Compute extrinsic parameters for view of interest

→ planner solve 2DH points. Non-planner solve 3DH points.

(j)

→ $$P_i^{\to 2DH} = K^* [R^* | T^*] P_i^{\to 3DH}$$

$$\begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix}^{2DH} = \underbrace{K^* \begin{bmatrix} r_1 & r_2 & T^* \end{bmatrix}}_{m' \text{ Homography}} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}$$

$$P_i = \begin{bmatrix} X_i \\ Y_i \\ 0 \\ 1 \end{bmatrix}$$ assume Z coordinate is zero.