classmate
Date
Page

Homework 5

Name: Karan Bhatiya
Student Id: A20424290
Course Number: CS512
Semester: Fall-18

Q.1

a) Outliers :→
These are the points which are different from all the other points.

Problem :→
The problem is that whenever we think the model is fit considering the outliers it results in a wrong direction.

b) Objective function used for robust estimation

$$\boxed{E(\theta) = \sum_{i=1}^{n} \rho_\sigma(d(x_i;\theta))}$$

Standard least square Objective function is:

$$\boxed{E(\theta) = \sum_{i=1}^{n} d[x_i, \theta]^2}$$

robust estimator = $\boxed{\rho_\sigma(x) = \frac{x^2}{x^2+\sigma^2}}$

c) Geeman Mcclure Objective function is as follows:

$$g_{\sigma}(x) = \frac{x^2}{x^2 + \sigma^2} \begin{cases} x >> \sigma & g_{\sigma} = 1 \\ x << \sigma & g_{\sigma} = x^2 \end{cases}$$

Advantage:

The advantage is that it doesn't get affected by the outliers. We can get the maximum outlier 1 by using this function.

Standard least square function the weight given to the outlier is x^2.

Bandwidth parameter (σ) can be adjusted in a iterative manner by using following steps:

1) Draw a large subset of points uniformly random
2) Fit model using robust estimation given by θ_n
3) Now, compute

$$\sigma_n = 1.5 * \text{median}(d(x_i, \theta_n))$$

4) Repeat the whole process till we get $(\theta_n - \theta_{n-1}) > \gamma$ (threshold)

c) In this process, we start with a large value of σ

i.e. $\sigma = 1.5 * \text{median}(d(x_i, \theta_n))$

as we ~~&~~ start fit the model better & better, the ~~value~~ median of point decreases so σ also decrease.

d) Principle of RANSAC Algorithm:

IS to use ~~the~~ maximum no. of points to fit the model and repeat these process several time and choose the best fit model.

The no. of points drawn at each points should be small, because there are very less chance of getting outliers and one of many trials will lead us to better model.

e) Parameters used for finding RANSAC algorithms as follows:

1) n : no. of points to drawn at each evaluation
2) d : d represents minimum no. of points needed
3) k : k represents number of trials
4) t : t represents distance to identify outliers

Scanned by CamScanner

Formula for estimating no. of trials as follows:

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

where,

P : is probability that at least one trial will succed,

w : is probability that ~~particular~~ point is inlier

n : no. of points to be drawn at each trial

we start $w = 0.5$

At each ~~it~~ iteration of trial the value of w gets updated

i.e. $\frac{\text{No. of inliers}}{\text{No. of points}}$

f) The main objective of the image segmentation is to seprate foreground from the background

Agglomerative Approach (Merge)

⇒ In this, we start with each pixel from different cluster and we merge iteratively based on the distance similarity of the feature vector.

Finally we make the cluster dense by merging similar ~~features~~ pixels together classmate

Date ______
Page ______

Divisive Approach (Split) :-

In this approach, we take all pixels in single cluster and we split them iteratively by looking at distance of pixels or feature vectors.

By using this method, we decrease the size of the cluster by removing pixels which donot belong to that cluster.

9) 1) K-means Algo. for Segmentation:

1) First of all select k i.e. the no. of clusters to be formed.

2) Now, start with initial given ~~value~~ of k means.

We randomly choose some pixel to be mean. So, we have to make sure that means are seprated enough to space the image.

3) Now, repeat the process untill the mean do not change.

- Now, we assign each pixel to the nearest cluster

$$l_i = \underset{j \in (1, k)}{\operatorname{argmin}} \|f_i - m_j\|^2$$

where,

$f_i \Rightarrow$ is feature vector of each pixel

& $m_j \Rightarrow$ is mean of the j^{th} cluster.

Now, calculate the new mean of the cluster

i.e. $m_j = \dfrac{\sum_{i \in S_j} f_i}{\text{no. of pixels at } S_j}$

& S_j is all the pixels we labeled by l_i

2) Mixture of gaussian algorithm for Segmentation: →

The process is very much similar to the k-means, but the change is only the distance measure we used to assign the pixels to the clusters.

The distance measure is given as follows

$$d = (f_i - m_j)^T \vec{\Sigma}_j (f_i - m_j)$$

Here Σ_j is the covariance matrix.

& we calculate it by as follows:

$$\Sigma_j = \frac{\sum_{i \in S_j} (f_i - m_j)(f_i - m_j)^T}{\text{no. of pixels in } S_j}$$

& mean calculation is same as kmeans

$$\therefore m_j = \frac{\sum_{i \in S_j} f_i}{\text{no. of pixels in } S_j}$$

b) Mean-Shift Algorithm

The algorithm we used for calculating the mean-shift is very similar to the kmeans, the only difference is calculating the mean cluster.

Here, mean calculation is given as follows.

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

In kmeans & mixture of gaussian, all pixels belonging to the cluster get equal weight, whereas in the mean-shift algorithm we weighted pixels are higher which are closer to the mean.

Whenever we recompute the mean of cluster, we give weight for the each pixels belonging to the cluster based on its distance to the previous mean of cluster

Q 2
a)

Projection equation is given by
$p = MP$

The problems of
1) Forward Projection :->
we have to find the image coordinates of the given 3D object. Given is the coordinates of the object point & projection matrix M.

2) Camera Calibration :->
Here, we have to calculate the Camera parameters ~~(intrinsic)~~, by using the image coordinate and the world coordinates

3) Reconstruction :->
Here we have to find world coordinate (3D) of the object, given is the image coordinate of object (p) & the projection matrix M.

Easiest one is forward projection, because there is no ambigious decision to make, it means that each point in 3D corresponds to a single point in 2D.

Hardest one is the Reconstruction, because we need to add the information that we already lost from going to 2D from 3D. As well as each point in 2D can represent a line in 3D, which makes it ambigious.

b) Necessary input for Camera Calibration

$$\{p_i\}_{i=1}^{n} \longleftrightarrow \{P_i\}_{i=1}^{n}$$

i.e Another representation

$$\{x_i, y_i\}_{i=1}^{n} \longleftrightarrow \{x_i, y_i, z_i\}_{i=1}^{n}$$

for camera calibaration, we need the corresponding points in both 2D & 3D.

~~c) Steps in~~

c) Non-Coplanar Calibration Algorithm.

1) Estimate the projection Matrix M using $p = MP$

where $p \Rightarrow$ image points } given
$P \Rightarrow$ world points }

2) Now, find the camera parameters i.e. K^* (intrinsic) & (R^* & T^*) extrinsic. using the projection matrix $\underline{\underline{M}}$

$$M = K^* [R^* \mid T^*]$$

Scanned by CamScanner

d) Projection Matrix

$$M = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 0 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\underset{3D}{P_i = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}} \Rightarrow \underset{3DH}{P_i = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix}}$$

$$p_i = MP_i = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 0 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix}$$

$$= \underset{2DH}{\begin{bmatrix} 18 \\ 14 \\ 7 \end{bmatrix}} = \underset{2D}{\begin{bmatrix} 18/7 \\ 2 \end{bmatrix}}$$

e) Given

world points (1, 2, 3) & image points (100, 200)

$$\begin{bmatrix} P_i^T & 0^T & -x_i P_i^T \\ 0^T & P_i^T & -y_i P_i^T \\ & & \\ & & \end{bmatrix} \begin{bmatrix} [m_1] \\ [m_2] \\ [m_3] \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 1 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & -100 & -200 & -300 & -100 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & -200 & -400 & -600 & -89 \\ & & & & & & & & & & & \\ & & & & & & & & & & & \end{bmatrix}$$

f) The minimal or optimal no. of points is 6 to find the unique solution of $\underline{M}$

We can obtain it by using SVD (Singular Value Decomposition) on the $2n * 12$ matrix & taking the last column of the matrix V

where

$A = UDV^T$ $\quad\therefore\quad$ A is $2n \times 12$ matrix

g) ~~The~~ principal that is used is as follows:

$m = R^* [K^* \mid T^*]$

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

We exploit the fact by taking dot product & cross product of the rows in the M.

h)

h) 1) We use the $\{P_i\}_{i=1}^{n} \leftrightarrow \{p\}_{i=1}^{n}$ correspondance of the image & world points given as i/p.

2) ~~the~~ Now, we use the M (projection matrix) & formula $p = MP$ to ~~calculate~~ find ~~to~~ the image points & corresponding world points & compare with then known/correct point.

$$\& \quad (K^*, R^*, T^*) = \sum_{i=1}^{n} \left(x_i - \frac{m_1^T P_i}{m_3^T P_i}\right)^2 + \left(y_i - \frac{m_2^T P_i}{m_3^T P_i}\right)^2$$

~~we want~~

The error should be low.

i) Principal of Planar Calib Camera Calibaration.

First of all we show the calibaration target to the camera, but we have to show it to the camera at least 3 times, because one view is not enough to do calibration.

Approach:

1) first, estimate the 2D projection map betn calibaration target & image.

2) Now, estimate the intrinsic camera parameter from several views.

3) Finally, Compute extrinsic parameter for any of the view.

In non coplanar calibration, only one view of the calibaration target is enough to calibrate camera parameters

j) The 2D Homograph transform the 2DH point to 2DH itself &

$$H = K[\vec{r}_1 \; \vec{r}_2 \; \vec{T}]$$

while, M (projection matrix) M transform the 2DH to 3DH

& M is 3x4

$$M = K^{*}\left[r_1 \; r_2 \; r_3 \; T^{*} \right]$$

The assumption we used to make sure that we deal with Homography matrices

i.e. z ~~points~~ coordinates of points = 0

$$P_i = \begin{bmatrix} x_i \\ y_i \\ 0 \\ 1 \end{bmatrix}$$