Computer Vision
Homework -5
CWID: A20424771
Rakshith Churchagundi Amarnath

---

(1)
a) the outlier points are very different than the other points. the problem with the outliers is that when the model is fit considering the outlier, it gives a wrong solution.

---

b) Robust estimation's objective function is as follows,

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

In the standard least square objective function,

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

we take the individual errors and square them,

$$\left(\rho(x) = \frac{x^2}{x^2+\sigma^2} \right)$$

(c) Geman - McClure: $\rho_\sigma(x) = \frac{x^2}{x^2+\sigma^2}$, when $x >> 6$, $\rho_\sigma = 1$
$x << 6$, $\rho_\sigma = \frac{x^2}{\sigma^2}$

The advantage of this function is its not affected by outliers. using this function, the maximum weight the outlier can get is 1. where as in the standard least square function the weight given to outlier is x^2.

the bandwidth parameter can be adjusted in an iterative manner.

* Draw a large subset of points uniformly at random.
* fit model using robust estimation. (θ_n)
* Compute $\sigma_n = 1.5 *$ median $(d(x_i, \theta_n))$
* Repeat the process while $(\theta_n - \theta_{n-1}) >$ threshold.

In the process, we started with a large σ.

i.e $\sigma_n = 1.5 *$ median $(d_i(x_i, \theta_n))$

and as fitting better, the median of the points decreases and inturn σ decreases as we estimate σ ~~dec~~ as $1.5 *$ median $(d(x_i, \theta))$.

(d) Principal of RANSAC algorithm is to use max. no/: of points to fit the model and repeat this process several times and choose the best model after many trails.

Number of points drawn at each attempt should be small as there are less chance of getting outliers and atleast we will get a best fit in one of these attempts.

(e) Parameters of RANSAC algorithm

n → number of points to draw at each evaluation.

d → minimum number of point needed

K → Number of trails.

t → distance of identity outliers.

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

where, P = Probability that atleast one of the trails will succeed.

w = Probability than a point is an outlier

n = number of points to drawn at each trail.

W.O.T,

update w on each iteration, $\frac{\text{Number of inliers}}{\text{Number of points}}$.

(f) objective of image sequentation is to seperate foreground from the background.

In merge approach, we start with each pixel in a different cluster and merge iteratively based on distance of feature vectors.

the dense clusters are formed by merging similar pixel together.

Scanned with CamScanner

In split approach, ~~the~~ Every pixel is a single cluster and iteratively split the cluster by looking at the distance of pixel.

we can reduce the size of cluster by removing the pixel which do not belong to a particular cluster.

---

(9) K-mean algorithm for segmentation:

* Select K, the number of cluster to be formed.
* Start with initial guess of K means $\{m_i\}_{i=1}^{k}$.

Random points are chosen some pixels to be the mean. Here we make sure that means are separated enough to space images

* Repeat until stopping condition is meet. i-e its constant.

for each pixel, assign the pixel to cluster nearest to it.

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

f_i = feature Vector of the i^{th} pi

m_j = Mean of j^{th} cluster

* calculate the new mean of the cluster as

$$m_j = \frac{\sum_{i \in S_j} f_i}{\text{no. of pixel in } S_j} \quad ; \quad S_j \text{ is all pixels named } l_j.$$

mixture of gaussian algorithm for segmentation.

the process in mixture of gaussian is same as that of K-means, other than the distance measure used to assign pixels to the cluster centers.

computed by:-

$$d = (f_i - m_j)^T \Sigma_j^{-1} (f_i - m_j)$$ where Σ_j is the covariance matrix.

and $$\Sigma_j = \frac{\sum_{i \in S_j} (f_i - m_j)(f_i - m_j)^T}{\text{number of pixels in } S_j} \quad ; \quad m_j = \frac{\sum_{i \in S_j} f_i}{\# S_j}$$

(h) Mean-shift algorithm for segmentation:-

The major difference is in calculating the mean of ~~the~~ cluster.

$$m_j = \frac{\sum_{i \in S_j} w(f_i - m_j) f_i}{\sum_{i \in S_j} w(f_i - m_j)} \quad ; \quad w(f_i \cdot m_j) = \text{exponent}(-\| f_i - m_j)$$

In Recomputation, the means of cluster, we assign weight for each pixel belonging to the cluster based on its distance to the previous mean of the cluster.

* In K-means and mixture of gaussian, all pixels belonging to the cluster get equal weights, whereas in the mean-shift algorithm. Pixels closer to the mean are weighted higher than the one farther. The closer the pixel/point is to the mean, the more it should affect the mean, than the one farther.

* Mean-shift find cluster center as peak of histogram

---

(2)

a) Projection Equation $p = MP$.

<u>Forward equation</u>: To find out the image coordinate of the 3D object given the co-ordinate of the object in world (3D) and the projection matrix M.

<u>camera calibration</u>: Given the image co-ordinate and the world co-ordinate of the object, find the camera parameters (internal & external) used in

projection.

<u>Reconstruction:-</u> given the image co-ordinates of object P, and the projection matrix M. determine the world co-ordinates (3D) of the object.

<u>Forward projection:-</u> No ~~dile~~ ambiguity in decision making. i.e, each point in 3D corresponds to a single point in 2D.

Reconstruction is ~~was~~ most difficult, because we need to add the information we already lost from going to 2D from 3D. Each point in 2D can be represented in a line in 3D. which makes it ambiguous

---

b) $\{p_i\}_{i=1}^{n} \leftrightarrow \{P_i\}_{i=1}^{n} \leftarrow$ i/P for camera calibration.

$\{x_i, y_i\}_{i=1}^{n} \leftrightarrow \{X_i, Y_i, Z_i\}_{i=1}^{n}$

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

(c) Steps in Non-coplanar calibration algorithm.

* Given the image points (p) & world points (P). Estimate the (3x4) projection matrix M, using $p = MP$.

* Find the camera parameters, internal (K^*), external (R^* and T^*). using the estimated projection matrix in step 1. as we know that $M = K^*[R^* | T^*]$

(d) $M = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 0 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{bmatrix}$ $\quad P_i = \underset{3D}{\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}} \Rightarrow P_i = \underset{3DH}{\begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix}}$

$p_i = MP_i = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 0 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1+4+9+4 \\ 1+0+9+4 \\ 1+2+3+1 \end{bmatrix}$

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

(e) $\begin{bmatrix} P_i^T & 0^T & -x_i \, P_i^T \\ 0^T & P_i^T & -y_i \, P_i^T \\ & & \end{bmatrix} \begin{pmatrix} [m_1] \\ [m_2] \\ [m_3] \end{pmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

Given $(1,2,3) \leftrightarrow (100,200)$

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

(f) The minimum no. of points that's necessary to be able to find a unique solution for M is 6.
The solution is obtained by performing singular value decomposition on the $2n \times 12$ matrix and taking the last column of the matrix V.
where, $A = UDV^T$; a 12×12 matrix formed by the equations.

(g) The principle used to extract the unknown parameters from the projection matrix M is:-

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

The rotation matrix has the orthogonal vectors along the rows, so we take dot Product of the rows in M, thereby cancelling out some unknown

---

(h) To compute the validity of the projection matrix M that we estimated, we use the $\{P_i\}_{i=1}^{n}$ ↔ $\{P_i\}_{i=1}^{n}$ correspondance of the image & world points given as input. we use the estimated projection matrix M and $p_i = MP_i$ to find the image points corresponding to the world points and compare them with the known point.

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

Error has to minimized as possible.

(i) Principle of Planar calibration

* Estimate 2D {homography} (2D projection map) between calibration target and image.
* Estimate the intrensic camera parameters from several views.
* Compute extrensic parameters for any view.

In non-coplanar calibration one view of the calibration target is enough to calibrate the camera parameters, whereas for planar calibration we need atleast 3 different view of calibration target

(j) 2D homography (2D projective map) It transforms the 2DH point to 2DH itself and H is a 3x3 matrix.

$$M = K^* [\gamma_1 \ \gamma_2 \ T^*]$$

whereas projection matrix M transform the 2DH point to 3DH and M is a 3x4 matrix.

$$M = K^* [\gamma_1 \ \gamma_2 \ \gamma_3 \ T^*]$$

The assumption that is used to make sure that we deal with homography matrices is that the z co-ordinate of the points are 0.

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

3DH