Benjamin Scialom
A20432063

Homework 5 : Camera Calibration

CS-512

1. Robust Estimation and segmentation

a.

An outlier is an observation that lies outside the overall pattern of a distribution. In other words, it can be considered as a sabotage of the data-set.
If the value associated to the data is too far away, then it has a huge impact on the final result.

b.

Least square objective function : (for line fitting)

Given a set of matched feature points {(xi , x′i)} and a planar parametric transformation1 of the form

$$x^{'} = f(x, p)$$

Using the least squares method (i.e., to minimize the sum of squared residuals) consists in :

$$E_{LS} = \sum_i ||r_i||^2 = \sum_i ||f(x_i, p) - x_i^{'}||^2$$

Robust estimation :

While regular least squares is the method of choice for measurements where the noise follows a normal (Gaussian) distribution, more robust versions of least squares are required when there are outliers among the correspondences (as there almost always are).
It involves applying a robust penalty function ρ(r) to the residuals :

$$E_{RLS} = \sum_i \rho(r_i) \ , \ r_i = f(x_i, p) - x_i^{'}$$

The robust norm $\rho(r_i)$ is a function that grows less quickly than the quadratic penalty associated with least squares.

c.

However, since this robust estimation function is not differentiable at the origin, it is not well suited for every set of data. So, a smoothly varying function that is quadratic for small values but grows more slowly away from the origin is often used such as the Geman-McClure function :

$$\rho_{GM}(x) = \frac{x^2}{1 + \frac{x^2}{a^2}}$$

a is a constant that can be thought of as an *outlier threshold*. An appropriate value for the threshold can itself be derived using robust statistics.

d.

Ransac (Random Sample Consensus).

The algorithm's outlines are :

- Perform multiple experiments
- Choose the best result

1) Repeat S times :

- Drawn n points random and uniformly
- fit model to points
- Find inliners in entire data-set (distance < t)
- Recompute model
- update parameters S et t

2) Choose the best solution :

- Large set consensus
- Small error solution

Which one is better ?
Large set consensus gives more support so it is the one use most of the time (more sample points).

The number of trials grows quickly with the number of sample points used. This provides a strong incentive to use the minimum number of sample points n possible for any given trial, which is how RANSAC is normally used in practice.

e.

Parameters :

- n = number of points drawn at each evaluation (2 or 3)
- d = min number of points needed to estimate model (2)
- S = number of trials
- t = distance threshold identify inlines

To ensure that the random sampling has a good chance of finding a true set of inliers, a sufficient number of trials S must be tried.
Let p be the probability that any given correspondence is valid and P (user selected, p=099) be the total probability of success (does not have outliers) after S trials.
The likelihood in one trial that all n random samples are inliers is p^n(estimated). Therefore, the likelihood that S such trials will all fail is :

$$(1-p) = (1-p)^n S$$

and the required minimum number of trials is :

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

f.

The term image segmentation refers to the partition of an image into a set of regions that cover it. The goal in many tasks is for the regions to represent meaningful areas of the image, such as the crops, urban areas, and forests of a satellite image.
In other analysis tasks, the regions might be sets of border pixels grouped into such structures as line segments and circular arc segments in images of 3D industrial objects.
Regions may also be defined as groups of pixels having both a border and a particular shape such as a circle or ellipse or polygon.
When the interesting regions do not cover the whole image, we can still talk about segmentation, into foreground regions of interest and background regions to be ignored.

Merge and split approaches :

For the merge approach, **we start with each pixel** in a particular cluster and then we merge clusters with small distance and we repeat that until clusters are not satisfactory. Whereas, in the split approach, **we start with all pixels** and split the clusters until there are not satisfactory.

g.

We define feature vector at each pixel x by :

$$f(x) = \begin{bmatrix} x -> location \\ I(x) -> intensity \\ L(x) -> local\ characteristics \end{bmatrix}$$

K-MEANS :

- Select k
- Select initial guess of means :$m_1, \ldots, m_k$
- Repeat while m_j change :

$l_i = argmin\ ||f_i - m_j||^2$ (for each pixel)
$S_j = \{i | l_i = j\}$

So, it's the solution of two problems : find a cluster center and assign point to cluster. What we can do is hold one parameter fixed and change other to minimize the error:

$$E(\{l_i\}, \{m_j\}) = \sum ||f_i - m_{e_i}||^2$$

Mixture of gaussians :

Like K-Means with additional parameters. We replace :
$d = ||f_i - m_{e_i}||^2$ with $d = (f_i - me_i)^T (\sum e_i)^{-1} (f_i - me_i)$

Which is literally equivalent to :
mahalanobis distance = mean of the cluster (transpose) x covariance matrix of the cluster x mean of the cluster

h.

Mean shift :

Similar approach as K-Means but the mean computation is replaced with a weighted sum based on distance from center.

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

It needs iterative computation because m_j is not known.

Note that : $w(f_i - m_j) = exp(-c||f_i - m_j||)$

---

2. Camera Calibration

a.

The use of a calibration pattern or set of markers is one of the more reliable ways to estimate a camera's intrinsic parameters. In photogrammetry, it is common to set up a camera in a large field looking at distant calibration targets whose exact location has been precomputed using surveying equipment .
In this case, the translational component of the pose becomes irrelevant and only the camera rotation and intrinsic parameters need to be recovered.
If a smaller calibration rig needs to be used, it is best if the calibration object can span as much of the workspace as possible as planar targets often fail to accurately predict the components of the pose that lie far away from the plane. A good way to determine if the calibration has been successfully performed is to estimate the covariance in the parameters and then project 3D points from various points in the workspace into the image in order to estimate their 2D positional uncertainty.

The easiest problem to tackle is the forward projection and the most difficult is reconstruction. Indeed, Due to the loss of one dimension in the projection process, the estimation of the true 3D geometry is difficult.

b.

The necessary input for camera calibration are intrinsic and extrinsic parameters of the matrix M in the projection equation.
so, we need to find :

K*, R*, T* => R, T, f, K_u, K_v, U_0,$V_{0,}$ $\tan(\theta)$

c.

Given ,for i=1 to m, $\{p_i\} < - > \{P_i\}$. There are two steps in non-planar calibration :

1. Find projection matrix M

- Using the projection equation, we can generate 2 equations for each point. There are 12 unknown parameters, so we need at least 6 points.
- By gathering the equations from the first step, we have a new equation to solve : Ax = 0.
- We find Ax=0 with SVD method (singular value decomposition) : $A = UDV^T$
- From the previous step we can find M

2. Find parameters from M

- From the previous step :

$$\hat{M} = \begin{bmatrix} \dots m_1^T \dots \\ \dots m_2^T \dots \\ \dots m_3^T \dots \end{bmatrix}$$, note that the solution for $\hat{M}$ is not unique.

- We need to find S so that : $M = k^*[R^* \,|\, T^*] = S\hat{M}$, where S in unknown
- Finally, find parameters and S from $\hat{M}$. Note that the extraction of parameters will be done thanks to the orthogonality property of r_1, r_2, r_3 which are the vectors of the rotation matrix R^*.

d.

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

So, $p_i = \begin{bmatrix} 18/7 \\ 2 \\ 1 \end{bmatrix}$ because we needed to marginalize p_i'.

e.

We have from the projection equation and the marginalized equation:

$$p_i' = \begin{bmatrix} \dots m_1^T \dots \\ \dots m_2^T \dots \\ \dots m_3^T \dots \end{bmatrix} P_i, \; x_i = \frac{x_i'}{w_i'} \text{ and } y_i = \frac{y_i'}{w_i'}$$

Thanks to this equation we can provide two equations :

- $x_i = \frac{m_1^T P_i}{m_3^T P_i}$ which involves $m_1^T P_i - x_i m_3^T P_i = 0$
- $y_i = \frac{m_2^T P_i}{m_3^T P_i}$ which involves $m_2^T P_i - y_i m_3^T P_i = 0$

Using the corresponding world-image point : (1, 2, 3) <-> (100, 200)

- $m_1^T \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix} - 100 * m_3^T \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix} = 0$
- $m_1^T \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix} - 200 * m_3^T \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix} = 0$

f.

For each point, we have 2 equations according to what is written above. Moreover, we have 12 unknown parameters. So we need at least 6 points to find a unique solution for the projection matrix M.

g.

From the question c of this part, $M = k^*[R^* \mid T^*] = S\hat{M}$.
it can be written with the following format :

$$M = [k^* R^* \mid k^* T^*] = S\hat{M} = S \begin{bmatrix} \ldots a_1^T \ldots & b1 \\ \ldots a_2^T \ldots & b2 \\ \ldots a_3^T \ldots & b3 \end{bmatrix}.$$

From this equation we can extract the for following equations :

- $\alpha_u r_1^T + tan(\theta) r_2^T + U_0 r_3^T = S a_1^T$
- $\alpha_v r_2^T + V_0 r_3^T = S a_2^T$
- $r_3^T = S a_3^T$
- $k^* T^* = Sb$

Then, to extract parameters, we have to use the orthogonality of r_1, r_2, r_3 and the equations just upper.

First, we find S, then U_0,V_0. And we continue with α_u, α_v and S. Then, we have to determine the sign of S by look at the third component of the vector b.
To conclude, we find the parameters from M.

h.

Given ,for i=1 to m, $\{p_i\} < - > \{P_i\}$
We assess the quality of fit using the error function that follows :

$$E = \frac{1}{m}(||x_i - \frac{m_1^T P_i}{m_3^T P_i}||^2 + ||y_i - \frac{m_2^T P_i}{m_3^T P_i}||^2)$$

Which is the error between known and predicated positions.

i.

The approach for planar calibration is the following :

1. Estimate 2D homography (projective maps) between calibration plane and image (for several images)
2. Estimate intrinsic parameters
3. Compute extrinsic parameters for view of interest

It differ from the non-coplanar one because we try to compute a 2D generated map.
For example, the units of the main equation change like this :

$p_i' = MP_i$ => (2DH) = (3x3) x (2DH)

j.

To get a 2D projective map, we assume that $\{P_i\}_z = 0$ which is equivalent to $P_i = (x_i, y_i, 0)$.

2D projective map H:

$$p_i' = MP_i$$

2DH 3X3 2DH

Projection matrix M :

$$p_i' = MP_i$$

2DH 3X4 3DH