Pose Estimation

By IIMP6020 teaching team

In this tutorial, you will learn how to estimate the pose of Tello with mission pads. The main procedure includes two parts:

- retrievaling and tracking:
- estimation:

Marker Retrievaling and Tracking

We need to detect markers in the captured image to determine the index of the mission pad which is called retrivaling. Usually for a video, it cost two much to retrieval markers in every frame. If a marker is retrievaled in the previous frame, then we can just track the same marker in the present frame. If the tracking is failed, then the retrivaling is executed.

Keypoint Matching By Locally Likely Arrangement Hashing

For each extracted point, its corresponding point is retrieved from the database. Because a dot does not have interior features to descriminate itself from others, we need to use geometric infromation from other potints to calculate the feature.There are several circular dots in each mision pad with different distribution. The local arrangement of neighbor dots is unique for each dot so the arrangements can be used as descriptors of dots.

Keypoint Extraction

For a captured image, keypoint extraction is first performed because the marker retrieval and tracking are based on keypoint matching. In order to utilize the center of each dot as a keypoint, we extract dot regions in the image.
Usually circular dots are black while their background is white. In this case, the dots can simply be extracted by thresholding brightness. Because the color of dots is not limited to black, other color extraction such as thresholding hue or saturation in HSV space is also applicable. After the binarization, connected regions are extracted and each center is computed as a keypoint.

Geometric Invariants

We use geometric invariants to describe the feature of a point. Geometric invariants are the values which keep unchanged under geometric transformation. There are mainly two invariants:

- **Cross-Ritio** is known as an invariant of perspective transformaiton calculated from five coplanar points ABCDE as follows:

$$\frac{P(A,B,C)P(A,D,E)}{P(A,B,D)P(A,C,E)} \tag{1}$$

  where $P(A,B,C)$ is the area of a triangle with apexes A,B and C.
- **Affine invariant** is an invariant of affine transformation which is more strictive than perspective transformation. Here we utilize four coplanar points ABCD to calulate as follows:

$$\frac{P(A,C,D)}{P(A,B,C)} \tag{2}$$

Calculation of Features

The simplest definition of the feature of a feature point p is to use f nearest feature points from p . In general, we assume that common m points exist in n nearest neighbors under some extent of perspective distortion. Common m points are obtained by examining all possible combinations $P_{m(0)}$, $P_{m(1)}$,...,$P_{m(nC_{m-1})}$ of m points from n nearist points ($nC_m = \frac{n!}{m!(n-m)!}$). As long as the assumption holds, at least one combination of m points is common. Thus a stable feature can be obtained.

The simplest way of calculating the feature from m points is to set $m = f$ and calculate the cross-ratio or the affine invariant from f points. Howerver, such a simple feature lacks the discrimination power because it is often the case that similar arrangements of f points are obtained from different feature points. In order to increase the discrimination power, we utilize feature points of a broader area. It is performed by increasing the number $m(> f)$. As m increases, the probability that different feature points have similar arrangement of m points decreases.

Registration

The index H_{index} of the hash table is calculated by the following hash function:

$$H_{index} = \left(\sum_{i=0}^{mC_{f-1}} r_{(i)} k^i \right) \ \ mod \ \ H_{size} \tag{3}$$

where $r_{(i)}$ is the discrete value of the invariant, k is the level of quantization of the invariant, and H_{size} is the size of the hash table.

The algorithm of registration of mission pads to the database is shown below.

1: **for each** $p \in \{All\ \ feature\ \ points\ \ in\ \ a\ \ database\ \ image\}$ **do**
2: $P_n \leftarrow The\ \ nearest\ \ n\ \ points\ \ of\ \ p(clockwise)$
3: **for each** $P_m \in \{All\ \ combinations\ \ of\ \ m\ \ points\ \ from\ \ P_n\}$ **do**
4: **for each** $P_f \in \{All\ \ combinations\ \ of\ \ f\ \ points\ \ from\ \ P_m\}$ **do**
5: $r_{(i)} \leftarrow The\ \ invariant\ \ caculated\ \ with\ \ P_f$
6: **end for**
7: $H_{index} \leftarrow The\ \ hash\ \ index\ \ calculated\ \ by\ \ Eq.()$
8: Register the item (document ID, point ID, $r_{(0)}$,...,r_{mC_f-1}) using H_{index}
9: **end for**
10: **end for**

Retrieval

The retrieval algorithm is shown below.

1: **for each** $p \in \{All\ \ feature\ \ points\ \ in\ \ a\ \ query\ \ image\}$ **do**
2: $P_n \leftarrow The\ \ nearest\ \ n\ \ points\ \ of\ \ p(clockwise)$
3: **for each** $P_m \in \{All\ \ combinations\ \ of\ \ m\ \ points\ \ from\ \ P_n\}$ **do**
4: **for each** $P'_m \in \{Cyclic\ \ permutationsof P_m\}$ **do**
5: **for each** $P_f \in \{All\ \ combinations\ \ of\ \ f\ \ pointsfrom\ \ P'_m\}$**do**
6: $r_{(i)} \leftarrow The\ \ invariant\ \ calculated\ \ with\ \ P_f$
7: **end for**
8: $H_{index} \leftarrow The\ \ hash\ \ index\ \ calculated\ \ by\ \ Eq.(3)$
9: Look up the hash table using H_{index} and obtain the list.
10: **for each** Item of the list **do**
11: **if** Conditions 1 to 3 are satisfied **then**
12: Vote for the document ID in the voting table.
13: **end if**
14: **end for**
15: **end for**
16: **end for**
17: **end for**
18: Return the document image with the maximum votes.

In order to remove items with different sequences of invariants, the following condition is employed.

Condition 1: All values of $r_{(0)}...r_{(mC(f-1))}$ in the item are equal to those calculated at the lines 5 to 7 for P'_m

Condition 2: It is the first time to vote for the document ID with the point p.

Condition 3: It is the first time to vote for the point ID of the document ID.

Marker Tracking

After markers are retrieved, they are tracked until failing the tracking. When we use geometric descriptors for keypoint matching, the tracking has an important aspect for handling wide range of view-points . Thanks to the tracking, augmentation of markers is possible even when a camera is tilted. In our work, instead of updating the descriptor database, we keep the descriptors of only previous frame because the updating procedure may produce an adverse affect such that the discriminative ability of the descriptors is degraded as the number of updated descriptors increases. In t-1th frame, the keypoints in the retrieved or tracked refer-ences are projected onto the captured image by using each homography to find more correspondences between keypoints in the image and keypoints projected from the references. For t th frame, key-point matching by LLAH is performed with t-1 th frame. Because the keypoints in t-1 th frame have already had the correspondences with those in the reference, the keypoints in t th frame can also have the correspondences with those in the reference. For the correspon-dences of each marker, the homography between t th frame and the references in the database is computed as in the retrieval.
In every frame, the tracking is performed at first. In the retrieval, tracked markers are removed from the candidates of retrieved mark-ers.

Estimation by PnP

Now we assume that we have known the 2D-3D correspondences. A simple definition of pose estimaiton could be: "given a set of correspondences between 3D features and their projections in the images plane, pose estimation consists in computing the position and orientation of the camera".

Camera Model

Let us denote $\mathcal{F}_c$ the camera frame and $\mathbf{T}_w^c$ the transformation that fully defines the position of $\mathcal{F}_w$ with respect to $\mathcal{F}_c$. $\mathbf{T}_w^c$ is a homogeneous matrix defined such that:

$$\mathbf{T}_w^c = \begin{pmatrix} \mathbf{R}_w^c & \mathbf{t}_w^c \\ \mathbf{0}_{3\times 1} & 1 \end{pmatrix} \tag{4}$$

where $\mathbf{R}_w^c$ and $\mathbf{t}_w^c$ are the rotation matrix and translation vector that define the position of the camera in the world frame.
The perspective projection $\bar{\mathbf{x}} = (u, v, 1)^T$ of a point $\mathbf{X}^w = (X^w, Y^w, Z^w, 1)^T$ will be given by

$$\bar{\mathbf{x}} = \mathbf{K}\mathbf{\Pi}\mathbf{T}_w^c\mathbf{X}^w \tag{5}$$

where $\bar{\mathbf{x}}$ are the coordinates, expressed in pixel, of the point in the image. $\mathbf{K}$ is the camera intrinsic parameters and is defined by:

$$\mathbf{K} = \begin{pmatrix} p_x & 0 & u_0 \\ 0 & p_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6}$$

where $(u_0, v_0, 1)^T$ are the coordinates of the principal point(the intersection of the optical axes with the image plane) and p_x(resp p_y) is the ratio between the focal length of the lens f and the size of the pixel l_x:$p_x = f/l_x$(resp, l_y being the height of a pixel,$p_y = f/l_y$). Π is the projection matrix given, in the case of a perspective projection model, by:

$$\mathbf{\Pi} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{7}$$

If we have N points $\mathbf{X}_i^w, i = 1, ..., N$ whose coordinates expressed in $\mathcal{F}_w$ are given by $\mathbf{X}_i^w = (X_i^w, Y_i^w, Z_i^w, 1)^T$, the projection $\mathbf{x}_i = (x_i, y_i, 1)^T$ of these points in the image plane is given by:

$$\mathbf{x}_i = \mathbf{\Pi} \mathbf{T}_c^w \mathbf{X}_i \tag{8}$$

Several(equal or greater than 3) paris of 2D-3D point correspondences $\mathbf{x}_i$ and $\mathbf{X}_i^w$ can provide a set of equations to solve $\mathbf{T}_w^c$. This is an inverse problem that is known as the Perspective from N points problem or PnP(Perspective-n-point).

Perspective-n-point

PnP considered an over-constrained and generic solution to the pose estimation problem from 2D-3D point correspondences.
For the 12 parameters of the matrix $\mathbf{T}_w^c$, considering that the homogeneous matrix to be estimated is defined by:

$$\mathbf{T_w^c} = \begin{pmatrix} \mathbf{r}_1 & t_x \\ \mathbf{r}_2 & t_y \\ \mathbf{r}_3 & t_z \\ \mathbf{0}_{3\times1} & 1 \end{pmatrix} \tag{9}$$

where $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$ are the rows of the rotation matrix $\mathbf{R}_w^c$ and position matrix $\mathbf{t}_w^c = (t_x, t_y, t_z)$. From the euquation, we can obtain a system to be sovlved:

$$\mathbf{A}\mathbf{h} = \begin{pmatrix} \vdots \\ \mathbf{A}_i \\ \vdots \end{pmatrix} \mathbf{h} = 0 \tag{10}$$

where $\mathbf{A}_i$ is given by:

$$\mathbf{A}_i = \begin{pmatrix} X_i^w & Y_i^w & Z_i^w & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & X_i^w & Y_i^w & Z_i^w & 1 \\ -x_i X_i^w & -x_i Y_i^w & -x_i Z_i^w & -x_i \\ -y_i X_i^w & -y_i Y_i^w & -y_i Z_i^w & -y_i \end{pmatrix} \tag{11}$$

and

$$\mathbf{h} = (\mathbf{r_1}, t_x, \mathbf{r}_2, t_y, \mathbf{r}_3, t_z)^T \tag{12}$$

is a vector representation of $\mathbf{T}_w^c$. The solution of this homoge-neous system is the eigenvector of $\mathbf{A}$ corresponding to its minimal eigenvalue (computed through a Singular Value Decomposition of $\mathbf{A}$). An orthonormalization of the obtained rotation matrix is then necessary.

Random Sample Consensus (RANSAC) to increase robustness

Assignment

1. Summarize the process of retrivaling and tracking using a block diagram.
2. Given a image of a mission pad, compute the cross-ratio and affine invariant of a feature point.
3. Given a group of 2d-3d point correspondences, compute the pose estimation.

Reference

The most content of the tutortial is borrowed from the following papers:
[1] Uchiyama, H. and Saito, H., 2011, March. Random dot markers. In 2011 IEEE Virtual Reality Conference (pp. 35-38). IEEE.
[2] Nakai, T., Kise, K. and Iwamura, M., 2006, February. Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval. In International Workshop on Document Analysis Systems (pp. 541-552). Springer, Berlin, Heidelberg.

[3] Marchand, E., Uchiyama, H. and Spindler, F., 2015. Pose estimation for augmented reality: a hands-on survey. IEEE transactions on visualization and computer graphics, 22(12), pp.2633-2651.

We provide the copies of these papers in the `related papers` folder for you to refer to.