Notes on Multiple View Geometry in Computer Vision

Rom Parnichkun

June 17, 2022

1 Background

- Homogenous representation of 2D points is (x, y, w), where the corresponding point in Euclidean space is $(x/w, y/w)$.
- Homogenous matrix $\mathbf{H}$ transforms homogenous points. $\mathbf{x}' = \mathbf{H}\mathbf{x}$.
- An image can be "warped" using $\mathbf{H}$.
- Hierarchy of transformations:
  - Class 1: **Isometries**: Transformations that preserve euclidean distance.
  $$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \epsilon\cos\theta & -\sin\theta & t_x \\ \epsilon\sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad (1)$$
  where $\epsilon = \pm 1$.
  - Class 2: **Similarity Transformations**: An isometry composed with scaling.
  $$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \quad (2)$$
  - Class 3: **Affine Transfomrations**: A non-singular linear transformation followed by a translation.
  $$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_1 & a_2 & t_x \\ a_3 & a_4 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad (3)$$
  in which a_1 to a_4 are the components of an affine matrix A.
  - Class 4: **Projective Transfomrations**: A general non-singular linear transformation.
  $$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_1 & a_2 & t_x \\ a_3 & a_4 & t_y \\ v_1 & v_2 & v_3 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \quad (4)$$
- Different cost functions.
  1. **Algebraic distance**: Is the algebraic residual of $||A\mathbf{h}||$ which ideally optimizes to satisfy $Ah = 0$. Problem is that this error doesn't really correspond to any real measurable quantity.
  2. **Geometric distance**: Is the geometric distance in the image, such as the difference between the measured and estimated image coordinates.
  3. **Reprojection error**: Is the geometric distance between ideal points and actual points such that the ideal points corresponds with a perfect homography.
- Generally speaking optimization takes these steps outlined below.
  1. Pose the problem as something like a non-linear least squares problem.
  2. Normalize coordinates/image.
  3. Use the direct linear transform (DLT) algorithm to find a starting point for a non-linear optimizer (which are generally iterative).
  4. Use a non-linear least squares optimizer such as the Levenberg-Marquardt algorithm.

  Optionally, to make the algorithm more robust, **RANSAC** can be utilized, in which, the minimum amount of functions required for solving the list of parameters are queried, and used to find the number of correspondences that are approximately satisfied. The set of minimal functions that satisfies the largest amount of correspondences is then picked as the final solution to the optimization problem. This acts as a method of robust to outliers.

2 Camera Geometry and Single View Geometry

- Steps to convert 3D point to points in the camera's coordinate:
  1. Projecting 3D point to a 2D point:
  $$\begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \quad (5)$$
  In which f is the focal length.
  2. Offsetting the point so that 0,0 lies on the top left of the image.
  $$\begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \quad (6)$$
  In which p_x and p_y is the offset in pixels. Here we can set
  $$K = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix}, \quad (7)$$
  and call K the *camera calibration matrix*.