Normalization for the Eight-Point Algorithm

Owen Jow

November 17, 2019

1 Motivation

In the eight-point algorithm, we set up a matrix equation $\mathbf{Af} = \mathbf{0}$ based on a reformulation of the epipolar constraint, where $\mathbf{A}$ is derived from the coordinates of corresponding points and $\mathbf{f}$ consists of the entries of the fundamental matrix. However, in its original form, $\mathbf{A}$ will typically be ill-conditioned because it contains both pixel coordinates (with a maximum value of, say, 1920) and final homogeneous coordinates (say, 1).

To circumvent this, we *normalize* the image coordinates upon which $\mathbf{A}$ is based. For each image, we construct a transformation which

1. translates the centroid of the image points to the origin, and
2. uniformly rescales the image points so that the mean L_2-distance from the origin is $\sqrt{2}$.

2 Implementation

First, rescale the image points so that the third (homogeneous) coordinate is 1 in all cases.

For the first transform, we simply subtract the current centroid point $(\mu_x, \mu_y) = \left(\frac{1}{n}\sum_{i=1}^{n} x_i, \frac{1}{n}\sum_{i=1}^{n} y_i\right)$ from each of the n image points (x_i, y_i).

For the second transform, we divide all points by the standard deviation (making the mean distance from the origin 1) and then multiply by $\sqrt{2}$. The standard deviation should be computed as the square root of the expected squared L_2-distance from the current centroid point:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[(x_i - \mu_x)^2 + (y_i - \mu_y)^2\right]}$$

which is the same as separately computing the standard deviation σ_x across x-coordinates and the standard deviation σ_y across y-coordinates and then combining them using $\sqrt{\sigma_x^2 + \sigma_y^2}$:

$$\begin{aligned}\sqrt{\sigma_x^2 + \sigma_y^2} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_x)^2 + \frac{1}{n}\sum_{i=1}^{n}(y_i - \mu_y)^2} \\ &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[(x_i - \mu_x)^2 + (y_i - \mu_y)^2\right]}\end{aligned}$$

Overall, the second transform comes together as the following scale factor:

$$s = \frac{\sqrt{2}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[(x_i - \mu_x)^2 + (y_i - \mu_y)^2\right]}}$$

2.1 Affine Transformation Matrix

As a single affine matrix, the transformation is

$$\mathbf{T} = \begin{bmatrix} s & 0 & -s\mu_x \\ 0 & s & -s\mu_y \\ 0 & 0 & 1 \end{bmatrix}$$

Note that the translations need to be multiplied by s because the scaling happens first.

3 De-Normalization

After computing $\mathbf{F}$ according to the normalized image points, we will have to de-normalize it. If the transformation for the first set of image points is $\mathbf{T}_1$ and the transformation for the second set of image points is $\mathbf{T}_2$, then de-normalization for $\mathbf{F}$ will be

$$\mathbf{T}_1^T \mathbf{F} \mathbf{T}_2$$

assuming the $\mathbf{x}_1^T \mathbf{F} \mathbf{x}_2$ form of the epipolar constraint.