

CSci 5563 Assignment 1

Luis Guzman

Friday February 4, 2020

Algorithm 1 Create Panoramic Image

```

1: Read all images.
2:  $\mathbf{R}_1 = \mathbf{I}_3$  ▷ Rotation for the first image.
3: for  $i^{\text{th}}$  image in all images except the last do
4:   Load consecutive images  $\mathbf{I}_i$  and  $\mathbf{I}_{i+1}$ .
5:   Extract SIFT feature using OpenCV for each image
6:   Find the matches between two images ( $\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$ ) ▷ MatchSIFT
7:   Estimate the homography between images using RANSAC ▷ EstimateH
8:   Compute the relative rotation matrix,  ${}^{i+1}\mathbf{R}_i$  ▷ EstimateR
9:   Compute the  $(i+1)^{\text{th}}$  rotation matrix,  $\mathbf{R}_{i+1}$ .
10: end for
11: Generate 3D points on the cylindrical surface ▷ ConstructCylindricalCoord
12: for  $i^{\text{th}}$  image in all images do
13:   Project the 3D points to the  $i^{\text{th}}$  camera plane ▷ Projection
14:   Warp the image to the cylindrical canvas. ▷ WarpImage2Canvas
15:   Update the canvas with the new warped image. ▷ UpdateCanvas
16: end for

```

Figure 1: Panorama stitching algorithm

The goal of this assignment is to stitch a series of photos into a 360 degree panorama. The algorithm is shown in figure 1. I first use OpenCV's SIFT feature extraction and match the keypoints between two images with the ratio test. I then use RANSAC to compute the homography between the two images. At each RANSAC iteration, the homography is calculated from a random sample of four keypoints, and then the number of inliers is computed to determine the best fit homography. The below figure shows the system of equations for calculating this homography, which can be solved directly with $H = (A^T A)^{-1} A^T b$.

$$\begin{bmatrix}
 u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1 u'_1 & -v_1 u'_1 \\
 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1 v'_1 & -v_1 v'_1 \\
 & & & & & & \vdots & \\
 u_4 & v_4 & 1 & 0 & 0 & 0 & -u_4 u'_4 & -v_4 u'_4 \\
 0 & 0 & 0 & u_4 & v_4 & 1 & -u_4 v'_4 & -v_4 v'_4
 \end{bmatrix}
 \begin{bmatrix}
 h_{11} \\
 h_{12} \\
 h_{13} \\
 h_{21} \\
 h_{22} \\
 h_{23} \\
 h_{31} \\
 h_{32}
 \end{bmatrix}
 =
 \begin{bmatrix}
 u'_1 \\
 v'_1 \\
 \vdots \\
 u'_4 \\
 v'_4
 \end{bmatrix}$$



Figure 2: Results of homography matching. The left image is warped to appear from the same perspective as the middle image.

The next step is to estimate the relative rotation matrix between the two image's cameras. This can be calculated from the intrinsic camera parameters K and the homography matrix H we computed in RANSAC.

$$R = K^{-1}HK$$

In order to ensure that R is a valid rotation matrix, I perform SVD clean up as follows

$$U, S, V^T = SVD(R)$$

$$R = UV^T$$

Lastly, I make sure that $Det(R) = 1$ to preserve a right-handed coordinate system. If $Det(R) = -1$, then I multiply R by -1 to compensate. Since each rotation matrix is relative to the previous camera, we must compose them to get the global rotation matrix.

$${}^w R_0 := \mathbb{1}$$

$${}^w R_i = {}^w R_{i-1} {}^{i-1} R_i$$

The last steps require projecting our images on to a cylinder for the panorama. I create a panorama canvas indexed by cylindrical coordinates (θ, h) . These points map to a 3D location

$$x = \begin{bmatrix} f \sin \theta \\ h - H/2 \\ f \cos \theta \end{bmatrix}$$

where H is the total height of the source image and f is the focal length ($f = K[0,0]$). We can then use each camera's rotation matrix to map the 3D points to the pixel coordinates of each image.

$$x = \lambda K^{-1} R^T u \implies u = \begin{bmatrix} u/\lambda \\ v/\lambda \\ \lambda \end{bmatrix} = KRx$$

By requiring that $\lambda > 0$, we can filter out the solution that corresponds to points behind the camera. Further, I check that each 3D point maps to a valid pixel location in the original image and mask out the points that fall outside the source image. Lastly, I perform inverse transformation by filling in each canvas pixel with the mapped location in the source image. The final panorama canvas is updated for each additional image.

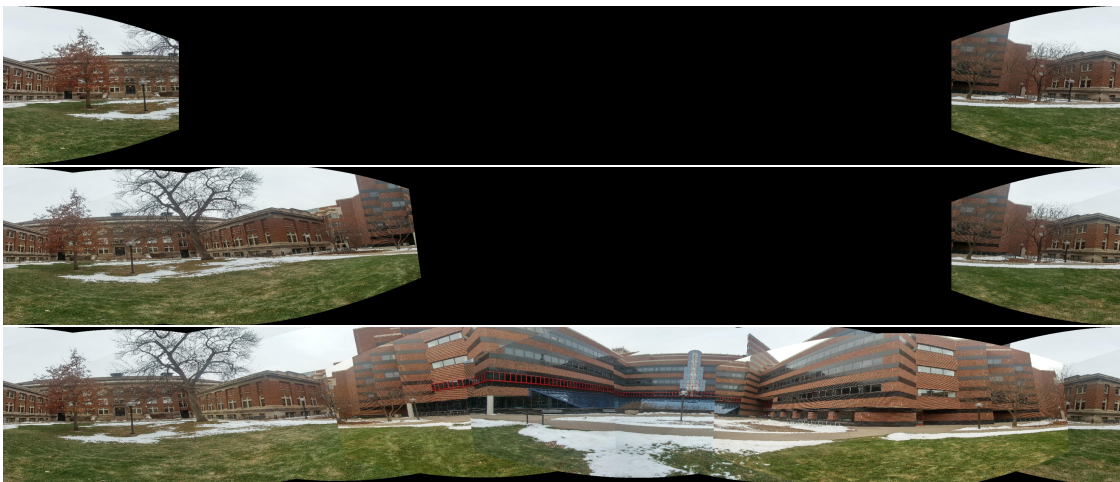


Figure 3: The final panorama (bottom) and the intermediate steps at the first image step (top) and third image step (middle).