# Assignment 3. Generate Panographs

Sneha Oladhri

University of South Florida

snehaoladhri@usf.edu

**Abstract**

In this paper we will discuss about generating panograph from a series of images. First, we discuss about feature detector, descriptor, and matcher to match features between the image pairs. Second, we estimate a best fitting 2D perspective homograph between two images. Third, we stitch the images to form the panorama.

***Index Terms*—Feature detector, descriptor, homography.**

## I. Introduction

Stitching of images has become crucial in many applications like self driving cars,document mosaicing and video stitching etc. Image stitching has been needed due to limited field of view captured by the cameras. To overcome this, multiple images have been stitched together to get the wider view of the scene.To achieve the final panogram of the images, we perform following steps:

- Feature detection and Matching
- Estimating 2Dperspective
- Stitching
- Averaging of images

### *A. Feature Detection and Matching:*

For feature detection we use SIFT detector. General steps of SIFT detector are:

- Keypoint detection
- Keypoint localization
- Orientation assignment
- Feature descriptor generation.

1) Keypoint detection: SIFT is better than than harris detector due to its scale invariance. This was achieved based on LoG function used between the functions.

$$L(x,y,\sigma) = f(x,y) \star G(x,y,\sigma) \quad (1)$$
$$= \int_x \int_y f(u,v) G(u-x, v-y, \sigma) dx dy \quad (2)$$

However, the best function to create scale space variance is the gaussian function.

$$\frac{\partial}{\partial \sigma} G(x,y,\sigma) = \sigma \nabla^2 G(x,y,\sigma) \quad (3)$$
$$\approx \frac{G(x,y,k\sigma) - G(x,y,\sigma)}{(k-1)\sigma} \quad (4)$$
$$G(x,y,k\sigma) - G(x,y,\sigma) \approx (k-1)\sigma^2 \nabla^2 G(x,y,\sigma) \quad (5)$$

- Keypoint Localization: Detect maxima and minima of difference-of-Gaussian in scale space. Each point is compared to its 8 neighbors in the current image and 9 neighbors each in the scales above and below. Once a keypoint candidate has been found by comparing a pixel to its neighbors, the next step is to perform a precise fit to the nearby data for location, scale, and the ratio of principal curvatures. This information allows points to be rejected that have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge. The Hessian matrix help eliminate edge responses based on curvature. A poorly defined peak in the difference-of-Gaussian function will have a sizeable principal curvature across the edge but a small one in the perpendicular direction. The derivatives are estimated by taking differences of neighboring sample points.

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2}{\partial^2 x} D & \frac{\partial^2}{\partial x \partial y} D \\ \frac{\partial^2}{\partial y \partial x} D & \frac{\partial^2}{\partial^2 x} D \end{bmatrix} \quad (6)$$

- Orientation Assignment: The keypoint scale is used to select the Gaussian smoothed image, L , with the closest scale so that all computations are performed in a scale-invariant manner. For each image sample, L(x,y) , at this scale, the gradient magnitude, m(x,y) , and orientation, (x,y) , are precomputed using pixel differences to approximate the derivative operation. An orientation histogram is formed from the gradient orientations of sample points within a region around the keypoint. The orientation histogram has 36 bins covering the 360-degree range of directions. Each sample added to the histogram is weighted by its gradient magnitude and by a Gaussian-weighted circular window with 1.5 times that of the keypoint scale. Peaks in the orientation histogram correspond to dominant directions of local gradients. The highest peak in the histogram is detected, and then any other local peak within 80 percent of the highest peak is also used to create a keypoint with that orientation. Each key specifies stable 2D coordinates (x, y, scale, orientation).

- Feature descriptor generation: Use the normalized region about the keypoint. Compute gradient magnitude and orientation at each point in the region. Weight them by a Gaussian window overlaid on the circle. Create an orientation histogram over the 4 X 4 subregions of the window 4 X 4 descriptors over 16 X 16 sample array were used in practice. 4 X 4 times 8 directions gives a vector of 128 values.

### *B. Estimating 2D perspective:*

The most important step in this process is to estimate the 2D perspective model from target image domain to the reference image domain. Homography is a 3*3 matrix representation which contains the transformation from one plane to another.

## II. Code and Algorithm Usage

In the current experimentation I used a series of images and found keypoint interests using SIFT detector and descriptor. Then homography for two images were estimated and warped w.r.t reference domain. The resulting images were stitched and blended to form a panorama. I ran the entire experiment on colab with gpu enabled runtime.

## III. Methodology

Series of overlapping images were taken and the keypoints were detected and matched using SIFT detector,descriptor. After getting the keypoint interests, homography of the images was estimated. Then the images were attached together to form a panorama.

### *A. Keypoint Detection and Matching:*

SIFT detector and descriptor are used for keypoint detection and analysing the points to compute the detectors. Then the keypoints detected were matched using the bruteforce matcher. For each descriptor in the first set finds the closest descriptor in the second set.

### *B. Estimating 2D perspective*

Gauss newton estimate for perspective transformation has been used to find the transformation matrix. Then the images were warped using the obtained transformation matrix.

### *C. Stitching*

In the final step, size of the resulting composite canvas was found and all the images were resampled on that to form a panorama.

### *D. Averaging of intersection areas*

After overlapping of images, image1 completely overlaps on the image2. To overcome this we introduce translucent by averaging the pixel values of common areas.

## IV. Experiments

The proposed method was implemented in python, pytorch and used a series of overlapping images.

Using SIFT detector and descriptor, keypoints will be detected and the analysed to get the descriptors. After getting the descriptors using bfmatcher, matching keypoints were matched. below is an example of keypoint matching.

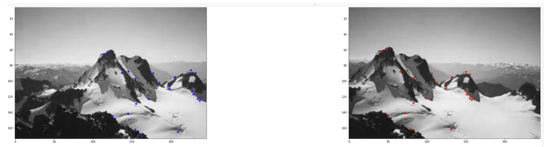

Figure 1. keypoints detection image

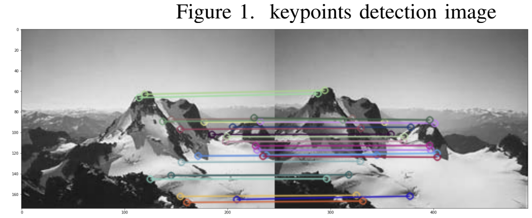

Figure 2. keypoint matching

After getting the keypoints matched, images were warped using perspective transformation. Below are examples of gauss-newton estimate for perspective transformation.

Figure 3. Perspective transformation

Then two images were stitched together to form the panorama. For making images into a single final canvas, the resulting canvas is taken of size with all images length. Then image 1 is overlapped on image 2 warped as image 1. It is repeated to multiple images too.

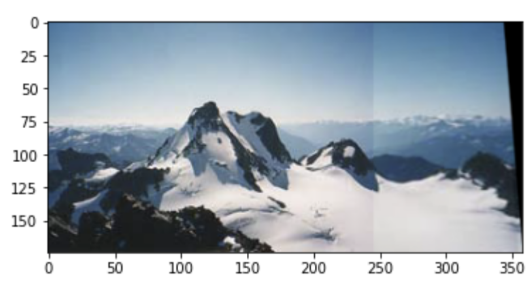


Figure 4. Resampling images into the final canvas

As the image1 is overlapped on warped image2, to get the translucent of the final image, we have done the average of pixels in common regions. To achieve this, a mask has been built for finding common region.

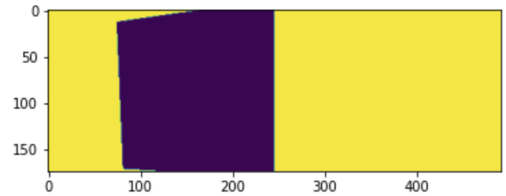


Figure 5. Mask for common regions

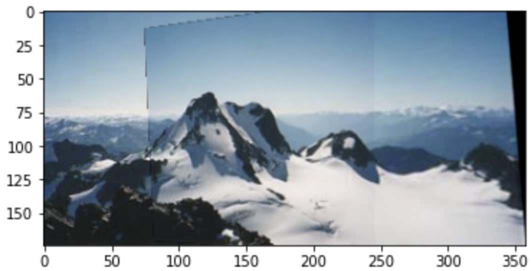


Figure 6. Final image after averaging the pixels at intersection areas.

Same experiment has been done on multiple images to form a panorama. Below are the images and final panorama.

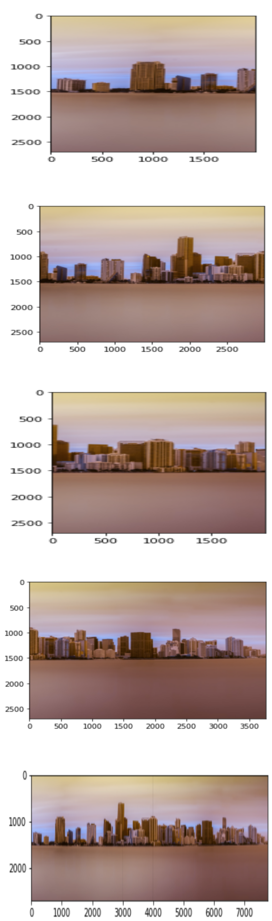

Figure 11. Final panorama.

Blending of the images was also tried on multiple images. An experiment was performed on a real time image from my community.


Figure 12. Stitching of images without blending


Figure 13. Stitching of images with blending


Figure 14. Stitching of images with averaging the pixels of common areas.

## V. Discussion

In the current experiment we have used the series of overlapping images to find the matching points in all the images. Then the images were warped using the transformation matrix obtained from gauss newton estimate for perspective transformation. Then all the images were resampled on the resulting composite canvas.

## VI. Conclusion

In this paper we discussed about Feature descriptor, detector and 2D transformations. We have seen a method to stitch multiple images to form a panorama.

## VII. References

1. Lang Nie, Chunyu Lin, Kang Liao, Yao Zhao, "Learning Edge-Preserved Image Stitching from Large-Baseline Deep Homography" .