Seam Driven Image Stitching

(PROJECT ID : 30)

TEAM MEMBERS :

1. SAMYAK JAIN (20161083)
2. MUDIT SURANA (20161100)

Github Link - Link

Introduction

Image stitching or photo stitching is the process of combining multiple photographic images with overlapping fields of view to produce a segmented panorama or high-resolution image. This is done by aligning multiple images based on the best geometric fit of feature correspondences between overlapping images. Seam-cutting is used afterwards to hide misalignment artifacts.

Goal of Seam Driven Image Stitching

The traditional method of image stitching is quite naive and does not produce the best results of panoramic images. New method of image stitching where seam cutting is applied beforehand to compute the best homography provides much better result.

Pre-requisites

- **Homography**

  Homography is a perspective transformation of a plane, i.e. a reprojection of a plane from one camera into a different camera view, subject to change in the translation (position) and rotation (orientation) of the camera.
  Perspective transformations map 3D points onto 2D image planes using the transformation matrix that incorporates the camera characteristics: focal length, optical centre, and the extrinsic parameters (rotation, translation) .

- **Scale-Invariant Feature Transform - SIFT**

  The scale-invariant feature transform (SIFT) is a feature detection algorithm in computer vision to detect and describe local features in images.
  There are mainly five steps involved in SIFT algorithm :

  1. Scale-space Extrema Detection
  2. Keypoint Localization
  3. Orientation Assignment
  4. Keypoint Descriptor
  5. Keypoint Matching

- **Random Sample Consensus - RANSAC**

  RANdom SAmple Consensus (RANSAC) a is a robust and iterative technique to estimate parameters of a mathematical model from a set of observed data that contains outliers, when outliers are to be accorded no influence on the values of the estimates. It can be seen as a resampling technique that generates candidate solutions

by using the minimum number observations (data points) required to estimate the underlying
model parameters. RANSAC uses the smallest set possible and proceeds to enlarge this set with consistent data points.

In our implementation we use ransac on the 128-dimensional feature vectors generated by **SIFT.** We try to fit our matched features (the [x, y] points matched) of both images in a model and since there are various outliers we also need to reject those outliers.

Basic Algorithm Pipeline-

1. Select randomly the minimum number of points required to determine the model parameters. (in our case we took 4 random correspondence points)

2. Solve for the parameters of the model
.
3. Determine how many points from the set of all points fit with a predefined tolerance ß.

4. If the fraction of the number of inliers over the total number points in the set exceeds a predefined threshold τ , re-estimate the model parameters using all the identified inliers and terminate.

5. Otherwise, repeat steps 1-4 for upto maximum of *N* iterations.

Traditional Method for Image Stitching

In the past implementations of Image Stitching, a naive two step pipeline was followed. The first was to compute the alignment or homography of both images using a robust algorithm such as **RANSAC** . The algorithm is used on the feature vectors to generate homographies. The homography with the best geometric fit is then used to align both images.

Implementation Pipeline

- The **SIFT** algorithm is used to generate features in both images.

```
sift = cv2.xfeatures2d.SIFT_create()
kp1, des1 = sift.detectAndCompute(im1, None)
kp2, des2 = sift.detectAndCompute(im2, None)
```

- The descriptors of both images can be used to match the features. **BFMatcher** is used to match these features. It returns a correspondence matrix with matched (x,y) of one image in queryIdx and other images matched point in trainIdx.

```
matcher = cv2.BFMatcher(cv2.NORM_L2, True)
matches = matcher.match(des1, des2)
```

- The matched feature vectors stored in as a correspondence matrix are fed into **RANSAC.** In our implementation maximum number of iterations for the algorithm to run is 1000. In every iteration four random

points are selected from the correspondence matrix. We calculate the homography matrix by using these four corresponding points -
$(x_i , y_i) \rightarrow (x_i', y_i')$ for i = 1:4
This gives homography H. Now we try to fit other matched points corresponding to this homography matrix. Geometric distance is calculated after applying homography over (x, y) gives (x1, y1) between these points. If the distance is less than threshold then these points are appended in our model as inliers and other are considered outliers.

- Various homographies are generated as it is an iterative algorithm. The homography with minimum number of outliers is selected as the best model fit. This was the traditional approach. In our approach every other homography generated is evaluated based on better alignment.

- Seam cutting is applied to the overlapping regions of pairs of images (**I1, I2**) aligned with the candidate homographies. The seam computation can be formulated as a labeling problem on a **Markov Random Field (MRF)** which minimizes a global energy with the following form:

$$E = \sum_{p} E_d + \lambda \sum_{(p,q)\in N} E_s$$

Where E_d is data-cost energy reflecting the saliency of a pixel, p, with label l_p and E_s is the smoothness energy which measures the discontinuity of adjacent pixels, p and q, defined over a 4-connected neighborhood **N**. Data cost energy is defined as -

$$E_d(p,\ l_p) = -\nabla I_{(l_p)}$$

l_p represents which image to choose **I1** or **I2** to choose from to (**i.e. ∇I 1 or ∇I 2**). The smoothness energy is defined as -

$$E_s(p,\ l_p,\ q,\ l_q)\ =\ \left|l_p\ -\ l_q\right|.(D(p)\ +\ D(q))$$

The smoothness cost is defined as the difference D of the overlapped pixels, where D is:

$$D(\mathrm{v})\ =\ \|I1\ (\mathrm{v})-I2(\mathrm{v})\|^{\ 2}\ +\ \alpha\,\|\nabla I1(\mathrm{v})-\nabla I2(\mathrm{v})\|^{\ 2}$$

In our implementation α is taken to be 2. This step generates the stitched image with values at common pixels decided by the energy minimized from both the images. This decides which image pixel to be used at the common pixel.

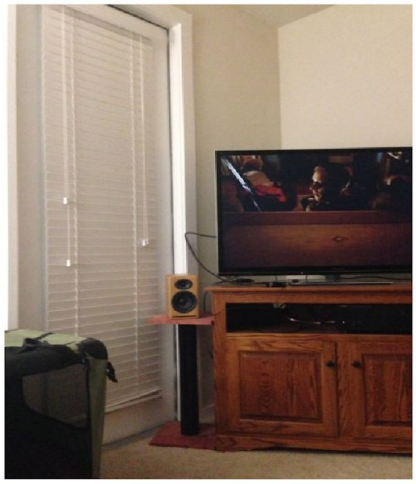
- The final step is to evaluate the seam cut. This step is the deciding factor as to which homography would be best the align the images. For each pixel, p, along the seam, we estimate an error value, E(p) by extracting a 17 × 17 patch, P, centered at p, and searching for its most similar patch in either I1 or I2 . This can be expressed as:

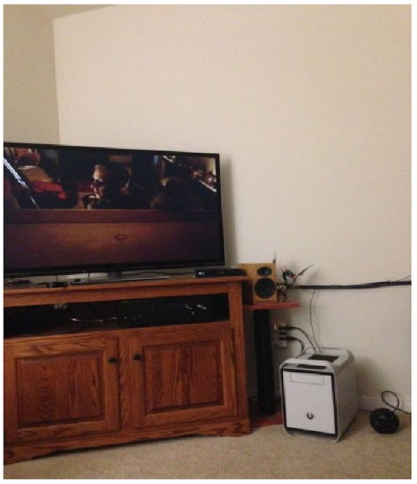
$$E(p)\ =\ min\ \left\|P\ -\ S_i\right\|^{\ 2}\ \forall\ S_i\ \in I_1,\ I_2$$

- The best homography matrix is now computed and used over any one of the image and warped over another image.

```
out = cv2.warpPerspective(im2,
                          scipy.linalg.inv(out_ransac),
                          (im1.shape[1],  im1.shape[0])
                          )
```

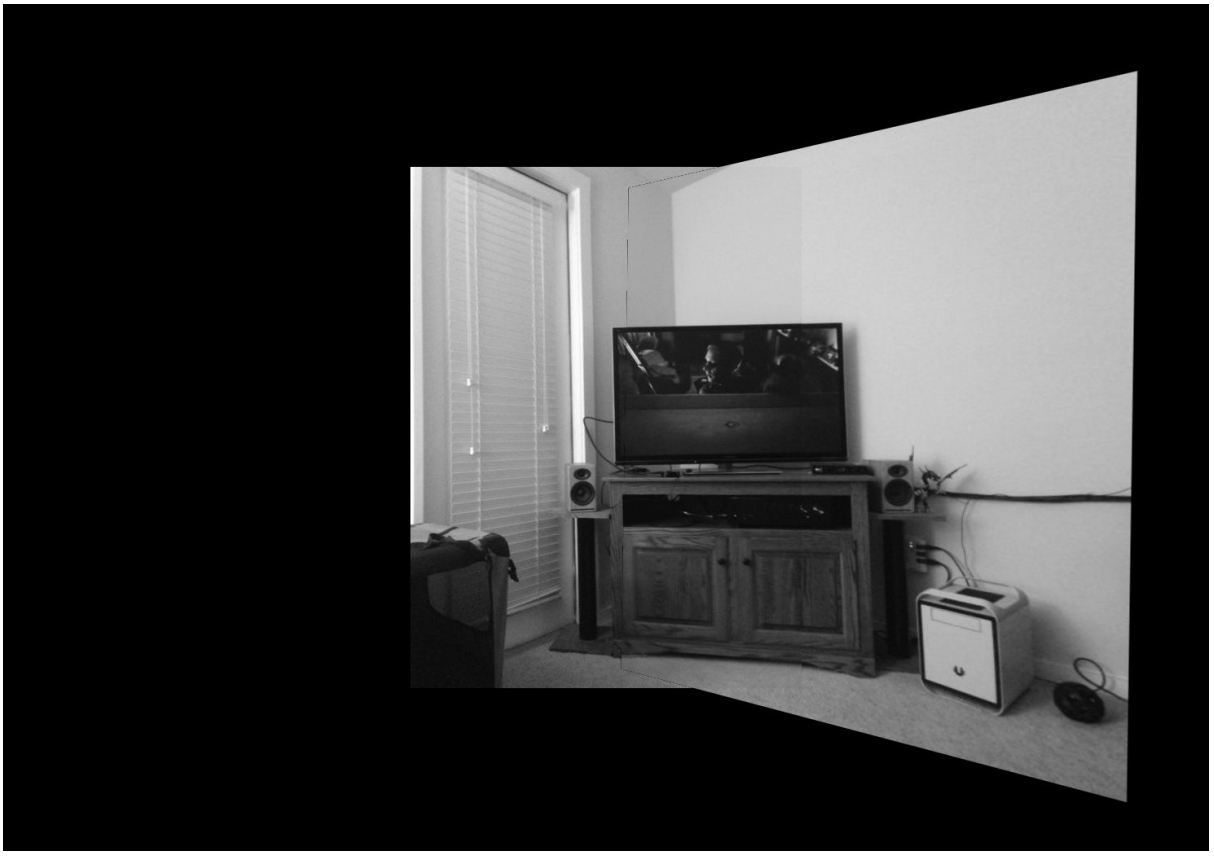
Results

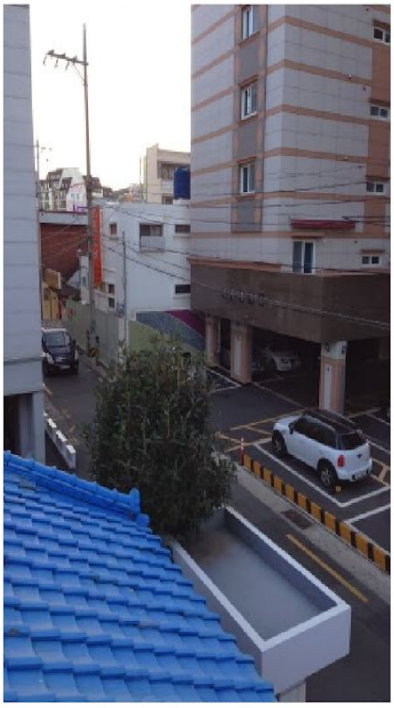




Result obtained by traditional method

Result obtained by Seam-cut algorithm





Result obtained by traditional method

Result obtained by Seam-cut algorithm