Automatic Matching and 3-D Reconstruction of Free-Form Linear Features from Stereo Images
Course Project : CS646

Mohd. Dawood 10327404
Mridul Verma 10415
Supervisor : Prof. Phalguni Gupta

April 14, 2014

1 Introduction

In the given paper we have to implement the algorithm for automatically matching of free-form linear features in overlapping large-scale imagery. Firstly, conjugate entities are determined among the overlapping part of images. Secondly, determining the relative orientation parameters(ROP) that relates a couple of stereo images. Lastly, the matched entities are projected into a stereo model using the relative orientation parameters derived in second part. For the first and second part we use the MIHT technique. For the third part (3-D reconstruction) we have used the disparity map.

2 Implementation Detatils

2.1 Feature Extraction(SIFT)

First we extract the keypoints from both the images. These feature points (or keypoints) are used to determine the fundamental matrix. These feature points extracted should be localized and should be invariant to scale and lighting. So for this purpose we have used SIFT feature extraction algorithm.

SIFT features are localized and invariant to image translation , rotation and partially invariant to illumination changes. Apart from SIFT features , we also used SURF and HOG features respectively. But due to their inherent properties the results were not that good with SURF and HOG as compared to SIFT maybe due to the fact that SIFT features are invariant to scaling , lighting and rotation.

Results with SURF and HOG were not good . The number of feature points identified were low as compared to the number of feature points with SIFT.

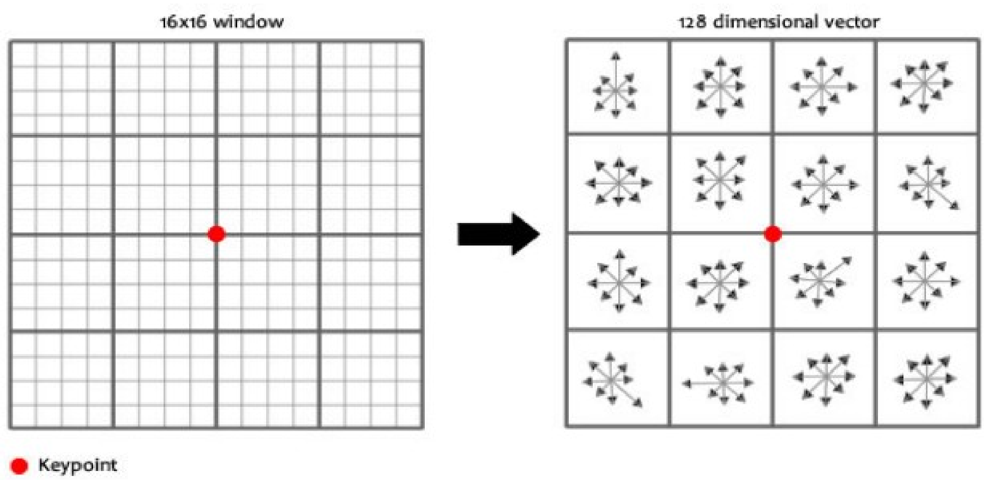

Figure 1: SIFT descriptor

After extracting the feature points we needed to identify the fundamental matrix (its parameters) ,so that we can get some relationship between the given pair of stereo images.

2.2 Fundamental Matrix calculation using MIHT

For finding the parameters of the fundamental matrix and and the matching pairs, we used the **MIHT** algorithm. Modified Iterated Hough Transform technique is used to solve some parameters given some data with some relationship amongst them.

Here we solve the fundamental matrix with the elements being the parameters and the feature points(keypoints) being the data, with the relationship between the feature points(from the stereo images) being :

$$x'T.F.x = 0 \tag{1}$$

where x' and x are the corresponding feature points on the stereo images and F being the fundamental matrix.

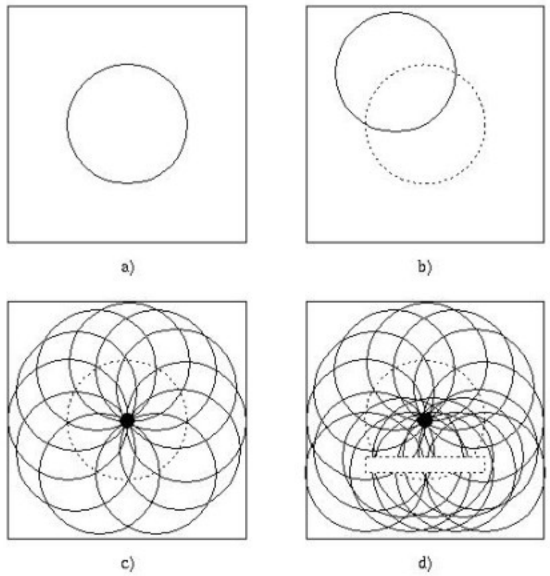

Figure 2: Hough Transform : Example of Circle Detection

To solve for the fundamental matrix :

1. We firstly initialise the matrix with some random values.
2. We also initialise an accumulator array with 5 bins for each of the parameters. We require this accumulator array for finding the bin with the most number of matching hypotheses.
3. Till now we have got n_1 points in the first image and n_2 points in the second image. We will take each pair of these points :

   for every keypoint in image 1

   for every keypoint in image 2

   (a) We calculate the value of the parameter a_{11} with the other parameters being constant.

   (b) Now we assign a11 to a bin and then increment the value of the accumulator array.

   In this way we solve for each of the parameters and assign it the value of the bin to which most number of the matching hypotheses belong.
4. We calculate each and every parameter of the matrix using the above mentioned technique.

We solve the fundamental matrix in this way. Now all the matching pairs(hypotheses) which contribute to the correct solving of the parameters are termed as the matching pairs.

Note : We have experimented with 5 to 10 bins in the accumulator array and the best results were when we took 5 bins with width ranging from $(a_i/2.5)$ where a_i is the random value allocated intially to the i^{th} parameter.

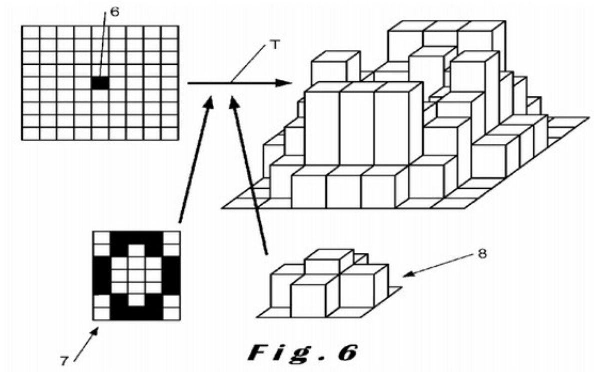

Figure 3: An example of a 2-D accumulator array

2.3 Parallelism

This is used in the step 3 of the algorithm described in section 2.2. To calculate the parameters, we run a loop from 0 to 8 with each iteration corresponding to one of the 9 parameters. In each iteration we consider each pair of points(one from keypoint set of first image and one from another) to calculate the set of values required to calculate the value of parameters. Now this[picking each pair from the set space of $n_1 \times n_2$] we consider in parallel. We calculate the desired intermediate values for each pair in parallel, thereby speeding up the whole process by a great deal.

Machine Specifications :

1. Ran all our codes on "*gpu01.cc.iitk.ac.in*" with OpenCV version-2.4.6.1 installed.
2. Took one dimensional blocks.
3. The size of each block is 256.
4. Number of blocks = $\frac{n_1 \times n_2}{256}$, where n_1 and n_2 are the number of keypoints in the respective keypoint set. keypoint sets.
5. The total number of threads will be : $n_1 \times n_2$. One thread for each pair of matching hypothesis.

2.4 3-D Reconstruction

1. Obtain the fundamental Matrix.
2. Compute the Essential Matrix and then compute the rotation and translation matrix(matrix describing the orientation between the two cameras).

   Essential Matrix:

   $$E = K'T.F.K \tag{2}$$

   More general form of fundamental matrix as essential matrix can only be used in relation to calibrated cameras since inner camera parameters must be known.
3. Then we use the inbuilt functions to obtain the disparity map of the stereo images.
4. After calculating the Essential Matrix we calculate the Rotation + Translation Matrix. This Matrix is used to defined the relative orientation between the two cameras. It is calculated by the SVD decomposition of the Essential Matrix.

$$\begin{bmatrix} R_{xx} & R_{xy} & R_{xz} & T_x \\ R_{yx} & R_{yy} & R_{yz} & T_y \\ R_{zx} & R_{zy} & R_{zz} & T_z \end{bmatrix}$$

5. After calculating the Rotation and Translation matrices , we obtain the disparity map of the stereo images from the matching points that we had calculated from the MIHT algorithm.

3 Experimental Data and Results

3.1 On images from the given dataset

Running the code on **22.tiff** and **21.tiff** from the given dataset we will get the following :

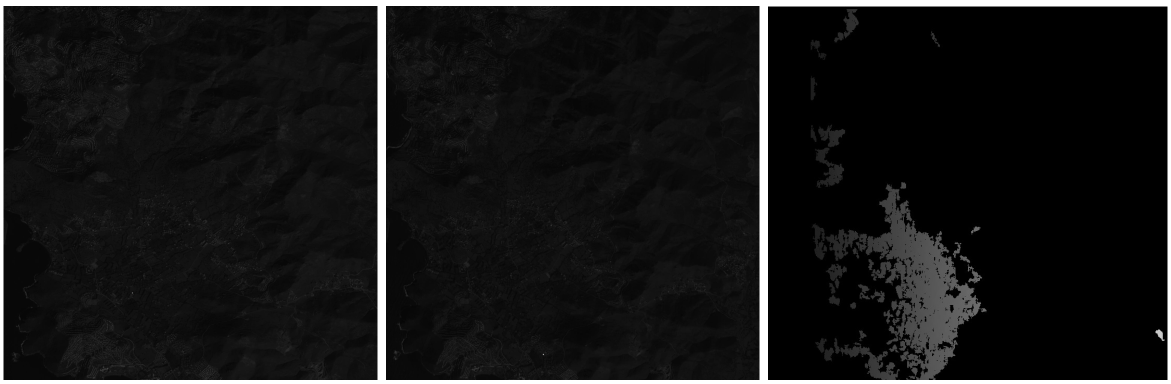

Figure 4: The first[from left to right] image is 22.tiff, then 21.tiff and then the disparity map that we get in the end.

The SIFT parameters were set as follows :

1. nFeatures = 0
2. nOctaveLayers = 4
3. contrastThreshold = 0.02
4. edgeThreshold = 10
5. sigma = 1.6

The number of keypoints detected were **454 in first image** and **252 in second image**. The parameter calculation part when done completely sequentially took 0.22 sec, while when the parallelizeable part was done in parallel it took 0.09 sec and the feature detection took 0.01 sec and the disparity map generation took even less than 0.01 sec. So, when done completely sequentially it took 0.23 sec, while parallelizeable part was done in parallel it took 0.09 sec.

Part	sequential	parallel
Feature Extraction	0.01 s	0.01 s
Parameter Estimation_Parallel Part	0.14 s	<0.01 s
Parameter Estimation_Sequential Part	0.08 s	0.08 s
Disparity Map Generation	<0.01 s	<0.01 s
Total	0.23 s	0.09 s

3.2 On images from outside the given dataset

Running the code on **w1.tiff** and **w2.tiff** from the EPFL website we will get the following :

Figure 5: The first[from left to right] image is w1.tiff, then w2.tiff and then the disparity map that we get in the end.

The SIFT parameters were set as follows :

1. nFeatures = 0
2. nOctaveLayers = 4
3. contrastThreshold = 0.09
4. edgeThreshold = 10
5. sigma = 1.6

The number of keypoints detected were **642 in first image** and **389 in second image**. The number of keypoints detected were **454 in first image** and **252 in second image**. The parameter calculation part when done completely sequentially took 0.39 sec, while when the parallelizeable part was done in parallel it took 0.15 sec and the feature detection took 0.06 sec and the disparity map generation took 0.01. So, when done completely sequentially it took 0.46 sec, while parallelizeable part was done in parallel it took 0.22 sec.

Part	sequential	parallel
Feature Extraction	0.06 s	0.06 s
Parameter Estimation_Parallel Part	0.24 s	<0.01 s
Parameter Estimation_Sequential Part	0.15 s	0.15 s
Disparity Map Generation	0.01 s	0.01 s
Total	0.46 s	0.22 s

4 References

1. Automatic Matching and Three-Dimensional Reconstruction of Free-Form Linear Features from Stereo Images, Ayman F. Habib, Young-ran Lee, and Michel Morgan, 2003.
2. Open CV Documentation : `http://docs.opencv.org/`
3. Cuda Documentation : `http://docs.nvidia.com/cuda/`