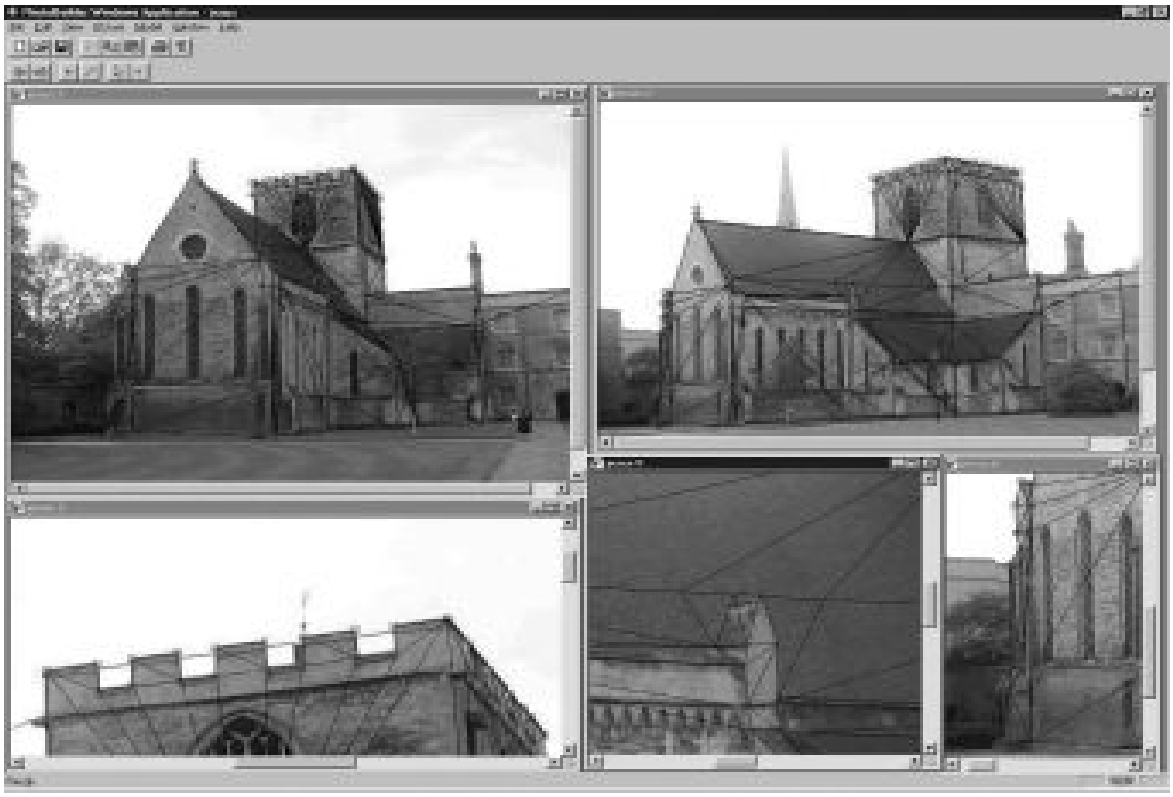
Camera Calibration using Vanishing Points

Project Report



Aditya Kusupati ***130050054***
Ritesh Theranikal ***130050069***
Sairam Bade ***130050079***

28.04.2017

CS 763 : Computer Vision

INTRODUCTION

We attempt to implement the camera calibration algorithm specified by : "*Camera calibration from vanishing points in images of architectural scenes*" authored by R. Cipolla, T. Drummond, D. Robertson. This paper presents a unique method to calibrate ea camera given two images taken by it at two different viewpoints and angles. We determine vanishing points in three mutually perpendicular directions for each image and calibrate the camera using those. The theory behind this project has been detailed in this paper and we've not included it in this for the sake of brevity.

APPROACH

1. **Test Images** :-We obtained two scenes of an Ikea store using google street view at two different viewpoints.Images are converted to grayscale.
2. We need to obtain three sets of parallel lines which are mutually perpendicular by marking two points on the line manually in each image and from this we get three corresponding vanishing points. We assume these lines correspond to X,Y,Z directions . From these vanishing points we will obtain a relation between **intrinsic matrix(K)** and **Rotation matrix(R)**.

KR = [p1 p2 p3]*diag($\lambda 1, \lambda 2, \lambda 3$) where $\lambda 1, \lambda 2, \lambda 3$ are homogenous factors.where p1,p2,p3 are the image coordinates. -(1)

3. K is dependent on optical centre and focal length assuming resolution is known and equal. Optical centre is the orthocentre of the vanishing points and focal length can be obtained from the equation of orthocentre.
4. $\lambda 1, \lambda 2, \lambda 3$ can be obtained using **orthonormality of R** in (1)

KK^T = [p1 p2 p3]*diag($\lambda 1^2, \lambda 2^2, \lambda 3^2$)* $[p1\ p2\ p3]^T$

5. **Computation of R**:R can be computed from (1) by using the lambdas obtained above.

6. **Computation of T** :Translation matrix is obtained by taking the image of world coordinate system's origin.An arbitrary reference point can be chosen as the origin's image. We can obtain T up to an arbitrary scale factor.As there is no metric information,this scale is indeterminate and can be set to 1. We need to use a fifth point correspondences in the two images to obtain T.

Results:

As a first step towards assuming that our camera matrices are legitimate,we can check the intrinsic matrices of both the images to be similar as the camera's resolution and focal length are same. They are similar.

We can find the point correspondences in both the images using SIFT and plot them to see the structure of the building.

There are many problems in finding the dataset as we need images taken from the same camera from different viewpoints, so we used google street view to obtain the input images. In many images one of the three directions is parallel to the image making the calculation of vanishing points not possible.

Input images:

Scene1

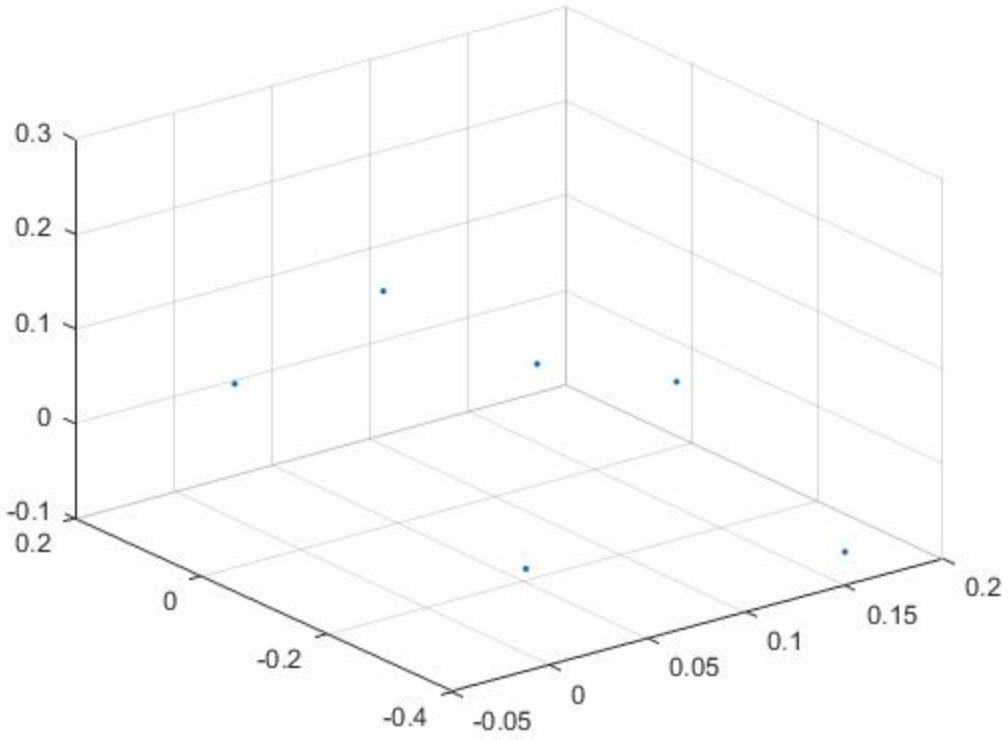
Scene2

We mark the correspondences in the two scenes and then plot that on a 3D plot to analyze the structure. Here we took points on the glass windows which are forming two perpendicular rectangular plane. In order to verify the plot, we've checked manually to ensure that the coordinates of the points are indeed vertices of perpendicular rectangles in 3D space.

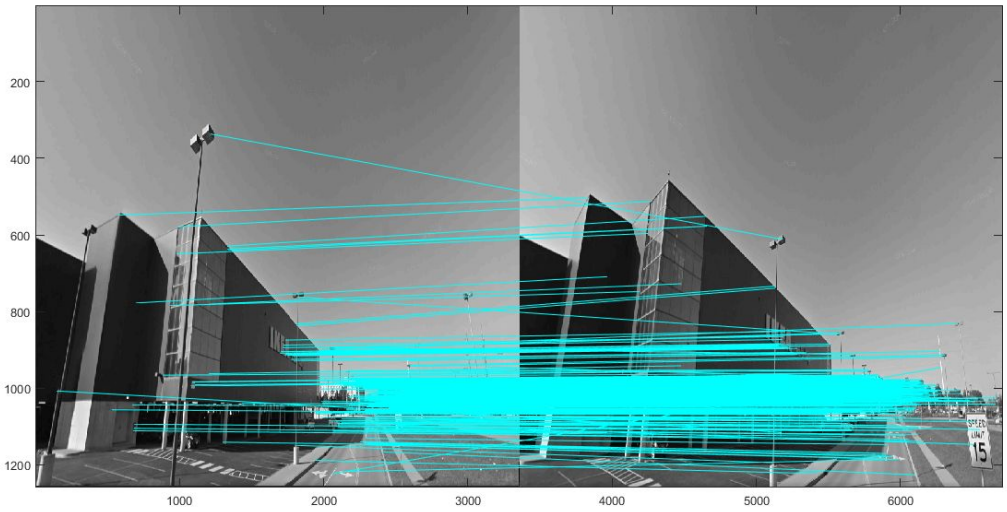
Scene1_features

Scene2_features

3Dplot obtained from these correspondences



There is also a problem with SIFT for getting the corresponding points in the image and plotting them as SIFT is giving a lot of correspondences which are not corners making us unable to get the structure from the plot .



Note: All the SIFT files are used from David Lowe's distribution and manipulated it to fit our need for testing

References:

1) http://www.cs.ubc.ca/~lowe/keypoints/
2) Images: https://www.google.co.in/maps/place/Mall+of+America/@44.8585871,-93.2433954,3a,60y,292.14h,85.41t/data=!3m6!1e1!3m4!1sJEXsO1NNedasjYYuclqa0w!2e0!7i13312!8i6656!4m5!3m4!1s0x87f62f6c393c612d:0xb3c6f1806e78286b!8m2!3d44.8548651!4d-93.2422148!6m1!1e1?hl=en
3) https://in.mathworks.com/matlabcentral/fileexchange/24445--center---calculates-and-plots-centers-of-a-triangle?focused=5136886&tab=function
4)