3D Vision

Dilip Puri, Murtuza Bohra, Sonu Patidar
Mentor: Prof. **Gautam Dutta** DAIICT
Indian Institute of Information Technology, Vadodara

Abstract—This project is based on Stereo Vision, a very well studied and still evolving field of Computer Vision. The aim is to extract 3D information of objects from a given pair of stereo images. The intuitive motivation is human vision system in which two images(stereo images) are formed in our eyes on retinas. Our brain does lots of computations on these images so that we are able to sense the depth of objects in the scene therefore having a 3D experience of the outside world.

Keywords—Stereo images, Projection, Line of sight, Field of view, Distortion, Calibration.

1 Introduction

The problem of Stereo Vision is highly important in Robotics. It is used to extract information about the relative positions of 3D objects, object recognition and path resolution problem. The depth information allows robots to distinguish various objects like a chair placed in front of other chair. Photos from aerial surveys can be used to make contour map of a geographical region, the obtained 3D information allows us to make a 3D view of a building to make better plan for maintenance and repairing. With the help of 3D information we can make perspective views of the building to analyze the repairing works.
Extracting 3D information from given stereo images involves finding common points in different images which are images of a same scene point. Since computations are performed on matrices containing pixel information of images, we need a mapping from outside world to the world inside camera, which requires knowledge of Projective geometry and Homogeneous coordinate system. The computations involved can be made efficient and simple using knowledge of Epipolar geometry. Extraction of 3D information broadly involves following steps:

1.1 Distortion Removal

Image might get radially distorted in wide angle and zoom lenses. Radial distortion primarily dominated by low order radial components, can be corrected using Browns' distortion model [1].

2 Details

This section would give an overview of the processes involved to accomplish the project:

2.1 Rectification

Image plane of both the camera taking pictures will not be same. The process of rectification will reproject both of these planes to a common plane. In our project we have assumed that images are free of any type of distortion and they are rectified.

2.2 Correspondence

Once we have rectified images we need to find out the disparity, which is actually the position difference of a common point in a image with respect to the other image. For these purpose we need a mapping from one image to the another. This mapping will transform each point in a image to its corresponding point in another image. Corresponding points are the images of a scene point in image planes of both the camera. Relative depth of points up to a certain scale can be calculated from disparities

using Triangulation(simple geometry).

2.3 3D Reprojection

Once we have the depth of each point in the image plane, we reproject each point in the image plane to the world planes according to their depths using our knowledge of Projective geometry and Homogeneous coordinate system. In this way we have a point cloud of a 3D coordinate system, this 3D points can be viewed in a 3D viewer.

3 DESCRIPTION

In this section we will give a step by step description of essential topics we learned to complete the project, we start with basic things required to understand the project which will be used here and there to describe the process.

3.1 Projective Geometry

To understand the importance of projective geometry we need to know what are perspective projections. If we see Figure(1) and try to guess, it is like viewing a same object from different angles and positions. It is as same as taking pictures of same scene from different angles and positions.
Likewise image plane of two camera catches perspective views of the same scene. The geometry of objects is strongly distorted by perspective image projection[2]. If we see Figure(2) parallel lines of a railroad track do not remain parallel in perspective view.
This is the time where we should know about projective plane. A projective plane is just a generalization of Euclidean plane where notion of distance is discarded since we have seen under perspective view formation even the ratio of the distance does not remain constant. In projective plane all line pairs meet at unique points, the case when two lines are actually parallel concept of ideal point is introduced. The projective plane comprise of ideal points and the plane at which scene points are projected, as shown in figure(3), the plane π and ideal points denoted by ideal rays. To see how things work together we should look on mathematical model for a Projective plane. The model will explain the two following axioms of Projective Geometry:

- Two distinct points determine a unique line.
- Two distinct lines determine a unique point.

3.2 Mathematical Model

- Point: A point is represented by a ray emanating from origin.
- Line: A line between two points is defined as a plane passing through rays representing the points.

The first axiom holds since two rays always define a plane passing through origin, which defines a line according to the model.
Now if we take any two such planes made by the rays passing through origin, they always cut each other at origin, and their intersection will be a ray from origin, which defines a point according to the model. From above observations and looking at Fig. 1 and Fig. 3, we can say that only following geometric properties of objects are preserved under projective geometry:

- Collinearity.
- Concurrence or intersection.
- Tangency.

In a nutshell image plane of both the camera can be thought of perspective projection of scene points. The relationship between such projections are projective transformations which allows us to jump from one image to another. This concepts can be mathematically represented with the knowledge of Homogeneous coordinate system. For more detailed information about Projective geometry have a look on following resources:

- Projective Geometry by Tom Davis.
- Appendix-Projective Geometry for Machine Vision by Joseph L. Mundy and Andrew Zisserman.

3.3 Homogeneous Coordinate

In the model just described above each point on the projective plane is represented in Homogeneous coordinate $(x, y, z)^t$. Any scalar multiple

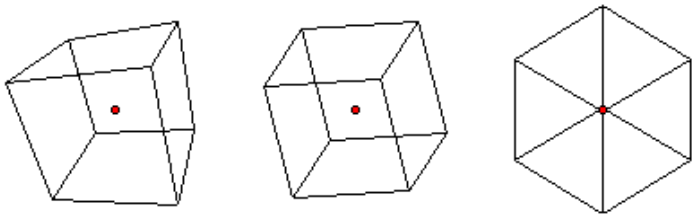


Fig. 1. Image Courtesy: http://whistleralley.com

Fig. 2. Image Courtesy: http://www.123rf.com

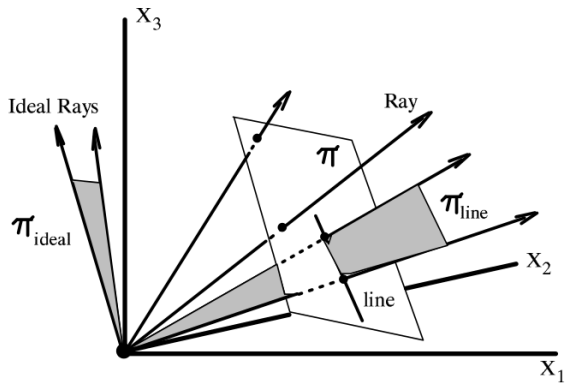


Fig. 3. 3D Ray Space Model

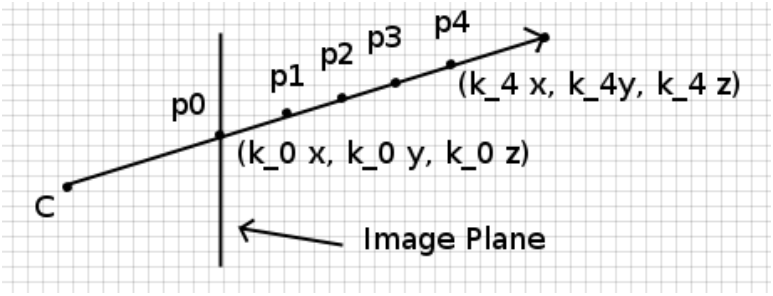


Fig. 4. 3D Ray crossing image plane

of $(x, y, z)^t$ viz. $(\lambda x, \lambda y, \lambda z)^t$ and $(x, y, z)^t$ are equivalent since both of them are represented by the same ray going through origin. Every point in 3D space can be represented in Homogeneous coordinate as $(x, y, z, z')^t$. When the third coordinate is zero, it is actually representing a point at infinity in Cartesian coordinate so in Homogeneous coordinate we have flexibility to represent point at infinity which is more convenient in calculations. When camera takes

a photo, it projects all 3D scene points on 2D image plane, so camera is actually doing a mapping of 3D scene points on to the image plane. This mapping can be best understood via simple geometry. But before we go on it and elaborate it any more let us have a look on relationship between Homogeneous coordinate and Cartesian coordinate. At a first glance it may seem that why do we need that? We need this because all the computations will be performed in the form of matrix multiplications in Homogeneous coordinates while we interpret image matrix as a Cartesian plane having origin at top left corner. Actually if we look at the Homogeneous coordinate, it is just the extension of Cartesian coordinate with additional third term.

We have already seen that $(x, y, z)^t$ or $K(x, y, z)^t$ are equivalent since both of them are represented by the same ray. Now we make a statement and say that $(x, y, 1)^t$ in Homogeneous coordinate represent $(x, y)^t$ in Cartesian coordinate, so $(x, y, z)^t$ will be equivalent to $(x/z, y/z)^t$ in Cartesian coordinate. From above definition we can say that all equivalent points in the Homogeneous coordinate represent same Cartesian coordinate.

Now the question is why do we normalize the third coordinate to 1 to get Cartesian coordinate? The answer is very intuitive. Look at Figure(4), then the vertical bar is the image plane(a projective plane). We want to represent this projective plane as Cartesian plane. It doesn't matter whether a point is at position $P_1, P_2, P_3 or P_4$, it will be projected at the same point on the image plane. Let us say point at projective plane is (K_0x, K_0y, K_0z) so the equivalent Cartesian point $(x/z, y/z)$, now since $P_1 and P_0$ are on the same ray they should also get converted to same Cartesian point since their image is being formed at the same projective point (K_0x, K_0y, K_0z). Which is only true when we normalize the third coordinate to 1. There is no other way we can get the same point at least from addition, multiplication or the subtraction by the third term with first and second term. You can verify.

3.4 Advantages of Using Homogeneous Coordinate System:

1) Translation type of transformation can be easily represented by matrix multiplication.
2) Uniform scaling is possible.
3) Because all transformation can be represented as matrix multiplication, transformations can be combined to a single transformation matrix.
4) Point at infinity can be represented using finite parameters.
5) Prospective transformation can be easily captured.

Now we are at the situation where we can introduce basic pinhole camera model.

4 Pinhole Camera Model

A pinhole camera is a black box as shown Figure(5)[3]. The image is formed at the inside wall opposite to the hole. For mathematical convenience we consider the image is being formed at f distance away from the camera toward scene shown in Figure(6)[3]. It is just flipped version of the original image. If we consider camera to be at origin then the virtual image(considered for math. convenience) and original image point have the same coordinates accept in the original one they are all negative. And the proof of the above sentence can be given in a single line-"Any line in 3D space passing through origin, if passes through a point (x, y, z) then it also passes through a point $(-x, -y, -z)$." Now look at Figure(7) using similar triangle property

$$x/X = f/Z$$

$$x = fX/Z$$

similarly,

$$y = fY/Z$$

The above equations are according to the world coordinate. If we see at image matrix the origin is usually the top left corner, so we need each and every (x, y) according to the matrix, for that we have to add something in each term. Consider the principle point on Z

axis in Figure(8)[4]. The coordinate are $(0, 0, f)$. Now if we shift origin to the top left corner, the new coordinate would become (δ_x, δ_y) in the plane. So in our model it would become

$$x = fX/Z + \delta_x$$

$$y = fY/Z + \delta_y$$

x and y are image plane coordinate, according to the convention of the origin at top left corner. Where δ_x and δ_y are respectively half of the height and width of the image plane. This x and y are in mm, cm or meter, whatever unit we prefer but in image representation in a computer we only have pixel coordinates, so we have to convert this x and y into x,y represent pixel coordinate for that we need information about size of camera sensor (i.e. width and height) and total no. of pixel in each dimension. If we know this, we can find out width and height of an individual pixel. So let us say these are w and h so if multiply by w and h we get

$$x = w(fX/Z + \delta_x)$$

$$y = w(fY/Z + \delta_y)$$

$$x = \alpha_x X/Z + P_x$$

$$y = \alpha_y Y/Z + P_y$$

$$Zx = \alpha_x X + P_x$$

$$Zy = \alpha_y Y + P_y$$

$$Z\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha_x & 0 & P_x \\ 0 & \alpha_y & P_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

The above relationship can else be written in the following manner:

$$\delta m = KM$$

where:

- m - represent pixel coordinate
- δ - represent depth of corresponding scene point M
- K - Calibration Matrix

We also introduce a skew term γ which moderates the projected position x as a function of height Y in the world. This parameter has no clear physical interpretation but can help explain the projection of points into the image in practice[3].

$$K = \begin{pmatrix} \alpha_x & \gamma & P_x \\ 0 & \alpha_y & P_y \\ 0 & 0 & 1 \end{pmatrix}$$

In practice we may not have camera center aligned with the origin of world coordinate system. In such a case we may have to perform some translation and rotation on world coordinate such that, they come under camera centered origin. We will learn more about rotation and translation in the process of Rectification, time being we say we have a translation matrix C and a rotation matrix R. The matrix R indicates orientation of camera centered reference frame with respect to the world frame. Since we know inverse of rotation matrix is as same as its transpose, we may multiply world coordinate with transpose of R to represent them in to camera centered reference frame. So we have following relation:

$$\delta m = KR^T(M - C) \tag{1}$$

eq^n(1) represent the full pinhole camera model, without noise in image.
At this moment of time we should appreciate the importance of Homogeneous coordinate system by looking back at eq^n(1)

$$Z\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha_x & 0 & P_x \\ 0 & \alpha_y & P_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{pmatrix} \left(\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} \delta_x \\ \delta_y \\ \delta_z \end{pmatrix} \right)$$

$$\begin{pmatrix} Zx \\ Zy \\ Z \end{pmatrix} = \begin{pmatrix} \alpha_x & 0 & P_x \\ 0 & \alpha_y & P_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} w_{11} & w_{12} & w_{13} & \delta_x \\ w_{21} & w_{22} & w_{23} & \delta_y \\ w_{31} & w_{32} & w_{33} & \delta_z \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Look at $(Zx, Zy, Z)^t$ it is Homogeneous representation of $(x, y)^t$ and $(X, Y, Z)^t$, it is normalized Homogeneous representation of $(X, Y, Z)^t$. Here because of Homogeneous coordinate system the face of eq^n(1) get change into AX = b form of matrix multiplication.

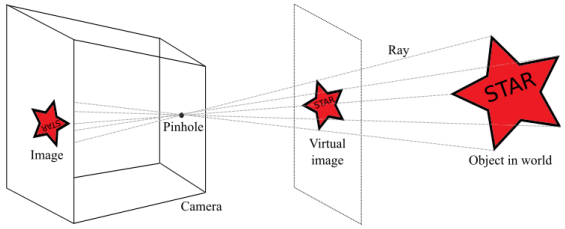


Fig. 5. Pinhole Camera

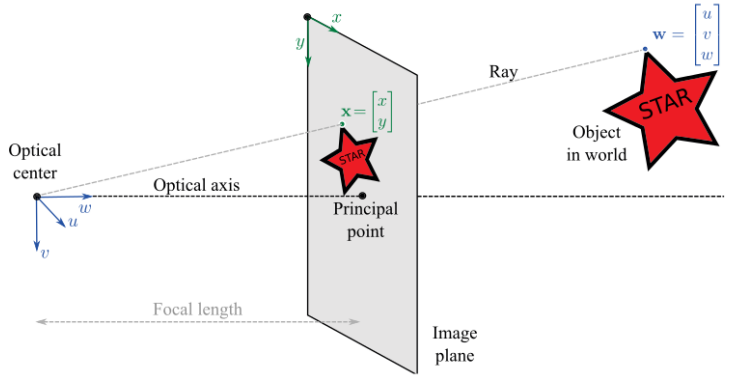


Fig. 6. Camera Model Terminology

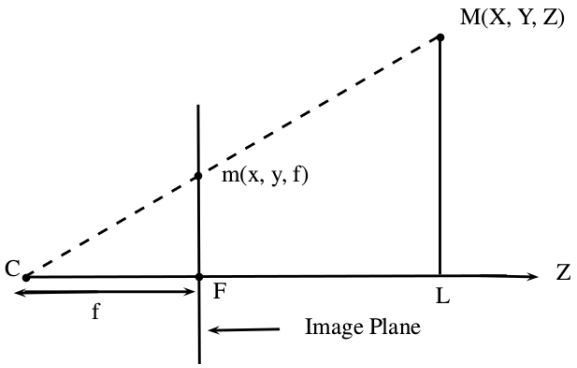


Fig. 7. Symmetric Triangle Property

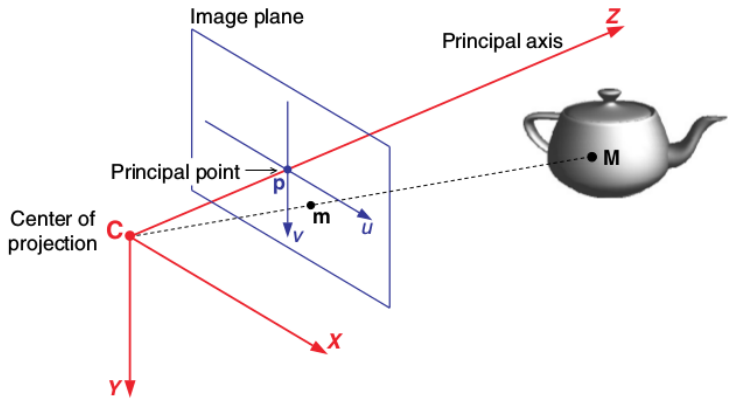


Fig. 8. Illustrated Camera Model

5 GEOMETRICAL TRANSFORMATIONS

Transformation	Preserves
Translation	Orientation
Rotaion (Rigid)	lengths
Projection	Straight line

5.1 Translation

A point (x, y, z) can be translated to (x', y', z') by a translation vector (t_x, t_y, t_z) as-

$$\begin{pmatrix} x' & y' & z' \end{pmatrix} = \begin{pmatrix} x & y & z \end{pmatrix} + \begin{pmatrix} t_x & t_y & t_z \end{pmatrix}$$

Translation transformation can be represented by matrix multiplication in homogeneous coordinate system by multiplying with translation matrix (T_t) as-

$$T_t = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{pmatrix}$$

$$\begin{pmatrix} x'' & y'' & z'' & h \end{pmatrix} = \begin{pmatrix} x & y & z & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{pmatrix}$$

from the translated Homogeneous coordinate, actual translated coordinate can be calculated by dividing each dimension by last parameter (h) of homogeneous coordinate

$x' = x''/h$

$y' = y''/h$

$z' = z''/h$

NOTE: This is the advantage of using Homogeneous coordinate system, we can do the translation using matrix multiplication.

5.2 Scaling

For scaling a vector by s_x in X direction, s_y in Y direction and s_z in Z direction transformation matrix is -

$$T_s = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} x' & y' & z' & h \end{pmatrix} = \begin{pmatrix} x & y & z & 1 \end{pmatrix} \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

For uniform scaling in all directions -

$$\begin{pmatrix} x' & y' & z' & h \end{pmatrix} = \begin{pmatrix} x & y & z & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1/s \end{pmatrix}$$

5.3 Rotation

Rotation is a bit complex transformation, let us do it in steps. First we will rotate about one of our principle axis say Z-axis.

To rotate a point by angle θ about Z-axis

$$x' = xcos\theta - ysin\theta$$

$$y' = xsin\theta + ycos\theta$$

$$z' = z$$

$$T_z^\theta = \begin{pmatrix} cos\theta & sin\theta & 0 & 0 \\ -sin\theta & cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

similarly about X-axis and Y-axis

$$T_x^\theta = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta & sin\theta & 0 \\ 0 & -sin\theta & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_y^\theta = \begin{pmatrix} cos\theta & 0 & -sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ sin\theta & 0 & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

5.4 Reflection

1) First reflection about plane -

Reflection about xy plane, the transformation matrix is -

$$T_{xy} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2) Reflection about axis -
Reflection about X-axis, the transformation matrix is -

$$T_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

5.5 Combining Different Transformations

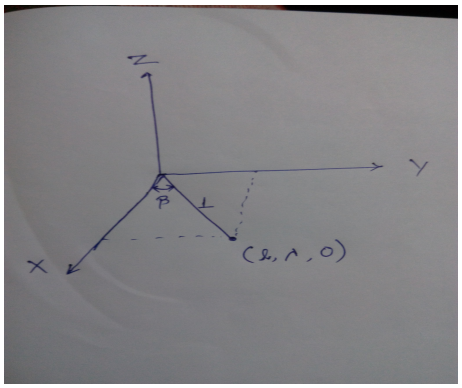
Let us use the combination of different transformation in an example.
EXAMPLE:- Rotation of a point by angle θ about an arbitrary line with direction cosine (l,m,n) and a point (a,b,c) through which it passes.
Solution: steps-
Step-1 : Translate origin to (a,b,c)

$$T_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -a & -b & -c & 1 \end{pmatrix}$$

Step-2 : Rotate the line such that it coincide with one of the principle axis. It involves two rotation $T2_{r1}$ and $T2_{r2}$.

$T2_{r1}$ = to rotate such that line lies in one of the principle plane say XY plane.

$T2_{r2}$ = to rotate in XY plane about Z-axis such that line coincide with X-axis.
Calculating $T2_{r1}$ -



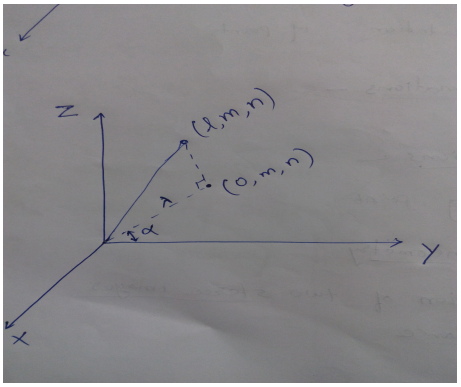
$\lambda = \sqrt{m^2 + n^2}$
$sin\alpha = n/\lambda$
$cos\alpha = m/\lambda$

now rotation by α about X-axis to get transformed line in XY plane.

$$T2_{r1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\alpha & -sin\alpha & 0 \\ 0 & sin\alpha & cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T2_{r1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & m/\lambda & -n/\lambda & 0 \\ 0 & n/\lambda & m/\lambda & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

after $T2_{r1}$ Transformation line will lie in XY plane, now again rotate the line by β about Z-axis to coincide with X-axis.



$cos\beta$ = l
$sin\beta$ = λ

$$T2_{r2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & l & \lambda & 0 \\ 0 & -\lambda & l & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

so, T_2 = $T2_{r1}$ $T2_{r2}$

Step-3 : After above two steps, we have transformed coordinate system such that our line became the X-axis. Our objective was to rotate point about the given line by angle θ, now has become rotation about X-axis. So third transform is T_3 is rotation matrix by angle θ about X-axis.

$$T_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta & sin\theta & 0 \\ 0 & -sin\theta & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

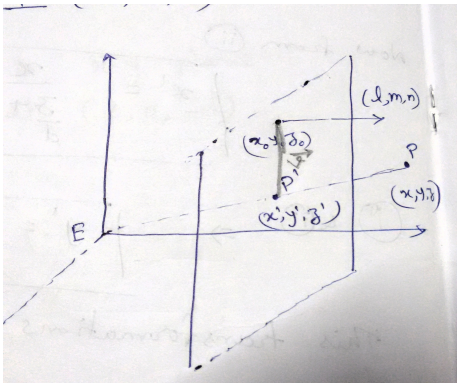
Step-4 : Inverse of step2 and step1 respectively to get back to original coordinate system again. i.e.
$T_4 = T2_{r2}^{-1} T2_{r1}^{-1} T_1^{-1}$
combine transformation is -
$T = T_1 T_2 T_3 T_4$

6 Projection

Projection in the context of our project is mapping a 3D coordinate on the camera's image plane. There are many techniques of projection like Orthogonal (parallel) projection and other, but we will discuss in detail about Prospective projection only because this is what a camera does while taking a image.

7 Perspective Projection

The two most characteristic features of perspective are that objects are smaller as their distance from the observer increases and that they are foreshortened, means an object's dimensions along the line of sight are shorter than its dimensions across the line of sight. This is how the depth information are associated with Prospective view of a 3D scene.
Special Property:- Two parallel lines after prospective projection becomes nonparallel and seems to meet at a point called vanishing point as we have already seen in Figure(2) of Railroad track. In case of human eyes all the rays from the 3D object are focused on eyes.
To get a Prospective Projection, all we required to know is location of eye and the direction of principle axis or the projecting plane. So now let us consider -
1) Eye at origin (0,0,0)
2) Projecting Plane passes through (x_0, y_0, z_0)
3) Perpendicular vector to the plane has direction cosine as (l,m,n).



Now we have to calculate the projective transformation of a scene point (x,y,z) on to the projecting plane at (x′,y′,z′).

$$\vec{V} = \begin{pmatrix} x' - x_0 & y' - y_0 & z' - z_0 \end{pmatrix}$$

$\vec{V}$ is the vector in the projective plane, so the dot product of $\vec{V}$ with the vector perpendicular to the plane is zero. i.e.

$$\vec{V}.(l, m, n) = 0$$

$$l(x'-x_0)+m(y'-y_0)+n(z'-z_0) = 0 \quad (2)$$

Equation of line EP in parametric form is -

$$\left.\begin{aligned} x' &= tx \\ y' &= ty \\ z' &= tz \end{aligned}\right| \qquad (3)$$

From eq.(1) and eq.(2)

$l(tx - x_0) + m(ty - y_0) + n(tz - z_0)$ = 0

$$t = \frac{lx_0 + my_0 + nz_0}{lx + my + nz}$$

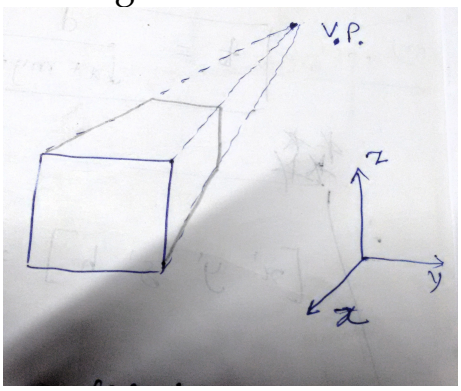
Lets say d = $lx_0 + my_0 + nz_0$, then d is the perpendicular distance of plane from the origin.
So the Projected point on the plane can be obtained by following matrix transformation under the prospective projection.

$$\begin{pmatrix} x' & y' & z' & h \end{pmatrix} = \begin{pmatrix} x & y & z & 1 \end{pmatrix} \begin{pmatrix} d & 0 & 0 & l \\ 0 & d & 0 & m \\ 0 & 0 & d & n \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

we can project any point in the space on the desired plane (Image plane) using above transformation.

8 Vanishing Point

A point at which receding parallel lines viewed in perspective appear to converge.



Properties of Vanishing Point: Lines parallel to the projecting plane remains parallel after projection.
Vanishing points lie along X,Y or Z direction are called **Principle Vanishing** Points.

8.1 How to find vanishing point for a Projection :

Lets consider the family of lines with direction cosines (p, q, r) are projected on a plane. A point A_0 (a, b, c) lies on one of the line, then -
$x = pt + a$

$y = qt + b$
$z = rt + c$
for different values of a,b and c we will have different lines with direction (p,q,r).
$A = Dt + A_0$, D=(p, q, r)

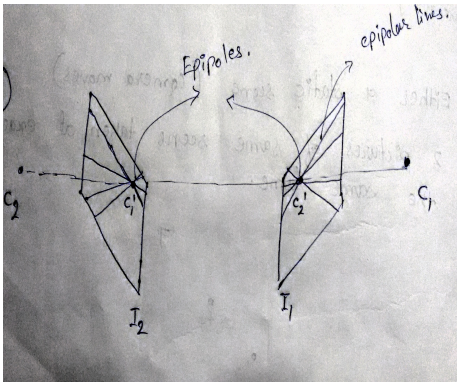
$limt_{t\to\infty}\vec{A'}$ = $\vec{A_\infty}$ T

T is the projection transformation matrix.
A_∞ = is a point at infinity in the direction D.

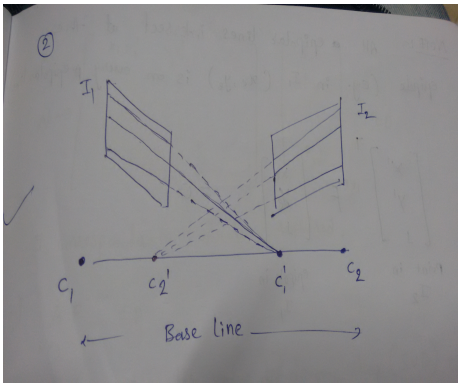
9 Epipolar Geometry :

'Epipolar geometry for two stereo images will be explained in this section. Let us consider two image planes I_1 and I_2 as shown below.
Physical Interpretation of epipolar lines : If we find the family of planes through base-line and rotate them, then the lines where the plane cuts these two image planes are the conjugate pair of epipolar lines.

9.1 Special cases



case 1 epipolar lines are always slanted because they converge to a point of image of other camera, i.e. if the camera projection lies on image plane of other camera then we can see the point of convergence of epipolar lines.



case 2 if Projection of one camera do not lies on image plane of other camera, then all the epipolar lines of one image plane seems to converge toward projection of other camera in the same plane.

10 Rectification of Stereo images

Image rectification is a transformation process used to project two-or-more images onto a common image plane. There are many strategies for transforming images to the common plane.

10.1 Need of Rectification

The pair of stereo images have corresponding matching point on the conjugate epipolar lines. Now we saw epipolar lines are slanted in the previous section, for searching corresponding point in stereo image pair we make these epipolar lines horizontal so that we can linearly search for correspondence. The transforming two images on a common plane makes the epipolar lines horizontal , This transformations is called **Rectification**.

10.2 How to Rectify images :

To rectify the stereo pair of images, we just have to reproject both images on a common plane by using the Prospective projection transformation(in section 5). To have the epipolar lines parallel and horizontal the common plane of projection must be parallel to the line joining to cameras.

11 CORRESPONDENCE

We are starting with a most general case where both of the camera are not aligned with the world frame, we have following eq^n for both the cameras :

$$\rho_1 m_1 = R_1^T K_1 (M - C_1) \tag{4}$$

$$\rho_2 m_2 = R_2^T K_2 (M - C_2) \tag{5}$$

from eq^n(4)

$$M = C_1 + \rho_1 R_1 K_1^{-1} m_1$$

putting it into eq^n(5)

$$\rho_2 m_2 = \rho_1 K_2 R_2^T R_1 K_1^{-1} m_1 + K_2 R_2^T (C_1 - C_2) \tag{6}$$

If we compare $K_2 R_2^T (C_1 - C_2)$ with eq^n(5) we can write -

$$\rho_{e_2} e_2 = K_2 R_2^T (C_1 - C_2) \tag{7}$$

where e_2 is the Epipole in second image, which is image of first camera center in the second image. To simplify the notations we will say

$$A = K_2 R_2^T R_1 K_1^{-1}$$

Now eq^n(6) becomes :

$$\rho_2 m_2 = A m_1 + \rho_{e_2} e_2 \tag{8}$$

The matrix A is referred to as infinite Homography. A is an invertible 3x3 matrix which, for every point m_1 in the first image, gives Homogeneous coordinate Am_1 for vanishing point in the second new of the projecting ray of m_1 in the first camera[4]. As shown in Figure(9)

eq^n(8) algebraically expresses the geometrical observation that, for a given point m_1, the corresponding point m_2 in the second image lies on a line l_2 through the Epipole e_2 and vanishing point Am_1 of the projection ray of m_1 in the first camera. At this point, we introduce a representation for a vector $a = (a_1, a_2, a_3)^t \in R^3$ let $[a]_\times$ denote following skew symmetric matrix :

$$[a]_\times = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}$$

This representation helps us understanding following relationship. Let e_1 vector $b = (b_1, b_2, b_3)^t \in R^3$ then

$$a \times b = [a]_\times b$$

from eq^n(8) we can say that vectors Am_1, e_2 and m_2 are linearly dependent, putting together in a different way we can say :

$$|m_2 e_2 A m_1| = 0$$

or

$$m_2^T (e_2 \times A m_1) = 0$$

$$m_2^T [e_2]_\times A m_1 = 0$$

$$m_2^T F m_1 = 0 \tag{9}$$

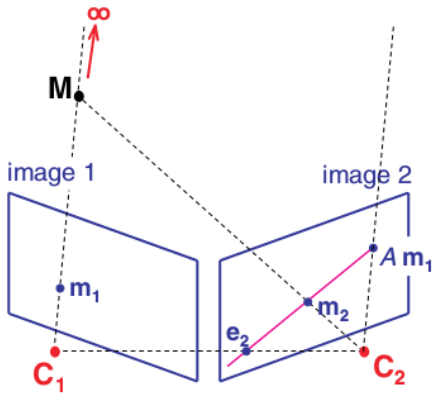


Fig. 9. Illustrated View Formation

$$\boxed{F = [e_2]_\times A}$$

We will exploit the relationship of eq^n(9) in finding out corresponding point in second image of a given point m_1 in the first image. Now the challenge is to find out fundamental matrix. To find out fundamental we need at least 8 point - correspondence once we have eight point and their corresponding points we can use Hartley's algorithm to find out fundamental matrix. Or the other better way would be applying SIFT (Scale Invariant Feature Transform) algorithm to find out feature descriptor in both images and then using SAD(Sum of Absolute Diff. method) we can find out corresponding feature points in images, then on the set of these feature points we apply RANSAC which gives us values of parameter involved in statistical model exempted from the effect of outliers(bad matches) using these parameter we can use maximum likelihood estimate to find out best matches for give points once we have some matches we can again use Hartley's algorithm to find our fundamental matrix.

12 3D Reprojection

Using fundamental matrix, we can find out corresponding point of each point in one image in the second image. This information would help us in finding 3D coordinate of a scene point of which images are our corresponding in both images. To keep this simple, we present the model of a stereo rig. in which two cameras with same internal parameter are place as shown in Figure(10) . The projection eq^n for stereo rig. are :

$$\rho_1 \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = K \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

and

$$\rho_2 \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = K \begin{pmatrix} X - b \\ Y \\ Z \end{pmatrix}$$

where

$$K = \begin{pmatrix} \alpha_x & S & P_x \\ 0 & \alpha_y & P_y \\ 0 & 0 & 1 \end{pmatrix}$$

for the above eq^n -

$$x_1 = \alpha_x X/Z + SY/Z + P_x$$

$$y_1 = \alpha_y Y/Z + P_y$$

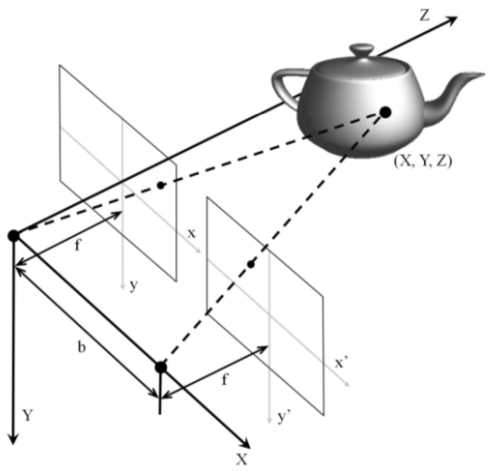


Fig. 10. A Stereo Rig

$$x_2 = \alpha_x(X - b)/Z + SY/Z + P_x$$

$$x_1 = \alpha_y Y/Z + P_x$$

After rectification in particular we have $y_1 = y_2$ and $x_1 = x_2 + \alpha_x b/Z$ Now solving above eq^n we can easily find out X, Y and Z, given K.

13 Results

We have used open source library function of OpenCV to find out fundamental matrix and then 3D scene points of corresponding points in the images. We used point cloud and point cloud visualizer, we have created 3D view of the scene. Here is the snapshot of our result.

14 Conclusion

The problem of stereo vision is broad and we do not claim that this report gives a complete solution at all. Even we would prefer to call it mere introduction to the problem. To find out the different aspects of problem we have gone through extensive reading of books, research paper, articles, appendices and web surfing. Definitely we have not included all the things we have learned and read but not understood, to keep thing simple and in a flow. To just give a hit to above statement we came to know so many things like Belief propagation, Graph cut algorithm, Birch field measurements, error optimization such as RANSAC, MLE(Maximum Likelihood Estimation), Singular Value Decomposition, Sum of squared distance method, various transformation models like Euclidean, Similarity, Affine and projective etc. winner takes all approach, Diagonalization of a matrix and many which we don't remember even.
The only thing we can say is this report gives exposure to the problem for a naive learner.

15 Future Work

In our case we have presented a simple case of a stereo rig., although in practice we might use a single camera and then take pictures from different positions

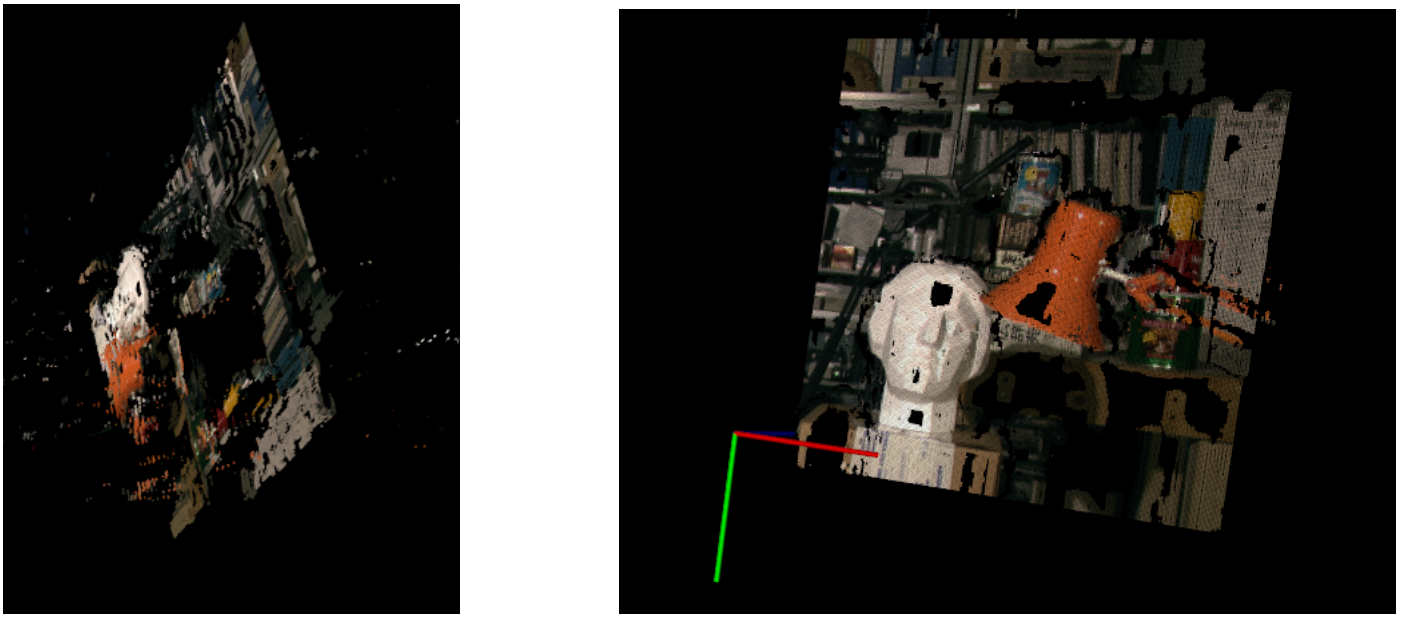


Fig. 11. Results
In the first figure objects at different depths(relative) are shown.

and angles, generating 3D structure from such input is more erroneous because the only information we have is in the form of images, so to find out faithful 3D structure from such inputs is a challenge. We would like to presure this challenge. This would includes multiple images of the same scene or might be a video and then build the 3D structure of the scene from image only information.

16 Acknowledgement

At the beginning we were completely unknown about how would we tackle the problem. We are thankful to Professor Gautam Dutta for mentorship, valuable suggestions and guidance. We are also grateful to Professor Asim Banerjee and Pratik Shah for giving greater exposure to the problem.

References

[1] Brown, Duane C. (May 1966). *"Decentering distortion of lenses"*
[2] Joseph L. Mundy and Andrew Zisserman. *"Appendix-Projective Geometry for Machine Vision"*
[3] Simon J. D. Prince *"Computer Vision : Models, Learning And Inference"*
[4] Theo Moons, Luc Van Gool, and Maarten Vergauwen *"3D Reconstruction from Multiple Images"*
[5] $https://en.wikipedia.org/wiki/Homogeneous_coordinates$
[6] Anup Chawla IIT Delhi *"Computer Aided Design Course (video lectures)"*
[7] Rich Radke *"CVFX Lecture 14 and 15"*
[8] Andrea Fusiello *"Elements of Geometric Computer Vision"*
[9] Ashutosh Saxena, Sung H. Chung, Andrew Y. Ng *"3D Depth Reconstruction from a single Still Image"*