Triangulation

Sai Srinija Sakinala
University of South Florida
FL, USA
sakinala@mail.usf.edu

I. Introduction

There are different types of coordinates when a picture is captured from a camera. They are- world coordinates, camera coordinates and image coordinates. There are some applications where we might need to map one coordinate to another. For such applications, various techniques are being used. One of them is Triangulation, where world coordinates are estimated with pixel locations given. In this cases, various other factors should be considered too. Such parameters are categorized into two- intrinsic and extrinsic. To calculate these coordinates, Tsai algorithm is used in this project.

II. Algorithm

As discussed in the previous section, there are two types of parameters- extrinsic and intrinsic. Extrinsic parameters map camera coordinates with world coordinates. Intrinsic coordinates map pixel locations to camera coordinates. Tsai algorithm needs training to output these parameters. For the training purpose, we need to give atleast 6 correspondences of pixel locations and their respective 3D coordinates. These correspondences are taken for each view considered. These are passed to the algorithm to get parameters for each camera. Then the process proceeds as follows:

- It outputs matrices K and P. K is a 3x3 calibration matrix which is one of the parameters. P is a 3x4 projection matrix which is a product of all the parameters i.e.,

$$P = K[R|t] \tag{1}$$

- As K is already known, R and t are calculated from the projection matrix P. Now, we have all the parameters that are needed for the triangulation part.
- In the triangulation part, we aim to minimise residual as much as possible. Hence the least square over the residual can be used. The least square of the residual is reformulated as-

$$p = (\sum_{j=1}^{N}(I - v_j v_j^T))^{-1} \sum_{j=1}(I - v_j v_j^T)c_j \tag{2}$$

- In equation (2), p represents the 3D coordinate that we want to estimate, N is the number of cameras present, v_j is a normalised vector of pixels with respect to world coordinate and c_j is distance vector between origins of world coordinates and camera coordinates.
- In order to estimate p, we need to calculate v_j and c_j. These two can be estimated from the formulae-

$$v_j = R_j^T K_j^{-1} x_j \tag{3}$$

$$c_j = -R_j^T t_j \tag{4}$$

- With equations (2), (3) and (4), an estimate of 3D coordinates is obtained.

III. Implementation

In this project, two images of the same object are captured from two cameras in different angles giving two different views. From each image, minimum of 6 correspondences of tuples in the form (X,Y,Z,x,y) are taken. These coordinates are passed to Tsai algorithm[1]. This algorithm returns projection matrix and calibration matrix for each image. After obtaining these matrices, R and t matrices are found for each image. Using the equations (2), (3) and (4) for triangulation, p is found out in the end which correspond to the respective 3D world coordinates. Triangulation takes the input of pixel locations for which world coordinates are to be found. The inputs and outputs of the implementation are as follows-

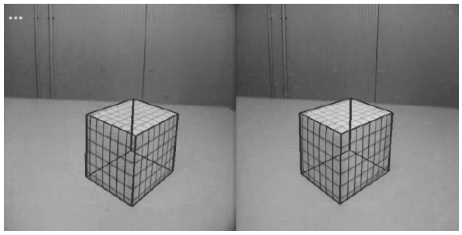

Fig. 1. Two pictures of same cuboid object taken with two cameras.

The images are passed to a line detector tool available online[2], to get accurate pixel locations and approximate 3D world coordinates.

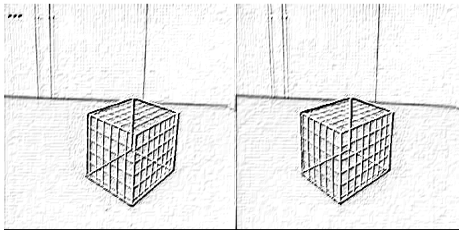

Fig. 2. Pictures after detecting lines.

As discussed above, a training set with tuples of points i.e., pixel locations and their respective 3D coordinates are passed to Tsai algorithm. In this case, the bottom corner in the front is taken as the origin. The training sets for the images and their respective parameter files are as follows- After passing these

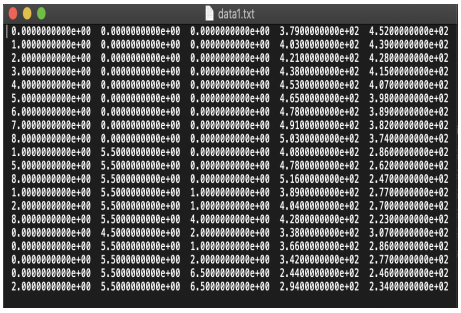

Fig. 3. Training set for the first image.

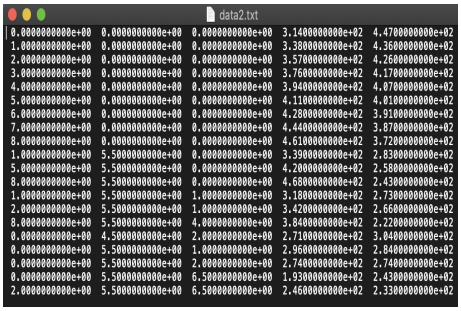

Fig. 4. Training set for the second image.

```
Projection matrix P :
[[ 3.24399358e+01 -4.76606993e+00 -1.28747870e+01  3.84275990e+02]
 [ 3.16754137e+00 -3.31839634e+01  1.25473557e-01  4.51737811e+02]
 [ 3.49941829e-02 -1.52747040e-02  2.93090345e-02  1.00000000e+00]]

Residues :
182.0469240964868

Calibration matrix K :
[[-637.49713161  -24.68219815  358.51845532]
 [  -0.         -638.49839154  268.19909952]
 [  -0.           -0.            1.        ]]
```

Fig. 5. Parameters obtained for the first image.

```
Projection matrix P :
[[ 3.06027664e+01 -4.21990626e+00 -1.01861502e+01  3.15644797e+02]
 [ 1.29623114e+00 -3.27141130e+01  1.57058126e+00  4.47884116e+02]
 [ 2.76878913e-02 -1.38137604e-02  3.41206479e-02  1.00000000e+00]]

Residues :
218.60681317502622

Calibration matrix K :
[[-655.23906456  -13.70826183  263.03052709]
 [  -0.         -664.27837776  255.17029602]
 [  -0.           -0.            1.        ]]
```

Fig. 6. Parameters obtained for the second image.

parameters to the triangulation method, the 3D coordinates of the respective pixel locations are obtained as follows:

```
3D cordinates for

[[401]
 [441]
 [  1]]

 and
[[335]
 [438]
 [  1]]
is
[[ 1.4683934 ]
 [ 0.44534424]
 [-0.44801559]]
```

Fig. 7. 3D coordinates for respective pixel locations.

IV. Conclusion

The triangulation algorithm works fine when the world coordinates are unknown, but parameters and pixel locations are known. However,this procedure of reducing the residual to a minimal value is not as effective as other procedures which solve with techniques like non-linear least squares or SVD. The results of can be made more accurate by using more views rather than considering just two, for which, time efficiency increases, but accuracy improves.

V. Issues

There are few issues with the implementation. They are-

- The training set needs to be composed for all the images that are considered which is a hectic task and time consuming.
- The triangulation procedure depends on the parameters outputted by the Tsai algorithm, depending on its efficiency.
- The accuracy is directly proportional to the number of images considered. As only two images are used for training, accuracy is expected to be low.

VI. References

[1]Tsai Camera calibration by Alex, https://github.com/alexprz/tsai-camera-calibration.

[2]Online Edge Detection by Pinetools, https://pinetools.com/image-edge-detection.

[3]Computer Vision: Algorithms and Applications by Richard Szeliski, http://szeliski .org/Book/drafts/SzeliskiBook-20100903-draft.pdf.