IPCV-Coursework

Luca Gough

1 3D From Stereo

1.1 Introduction

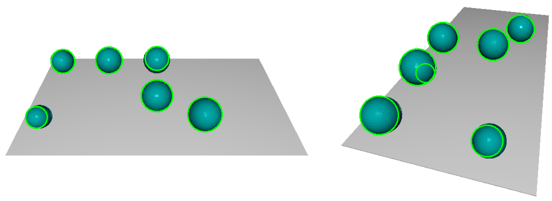
The aim of this section is to investigate and perform 3-D reconstruction using stereo images of a scene of spheres. 3-D reconstruction from stereo images in a calibrated system requires the translation and rotation between the two images as well as a set of corresponding points between the two images.
The first task is to obtain a correspondence between the centre of the spheres from each view, this is the process of matching each sphere centre from one view to another. This is achieved by locating the centres of each sphere in the image plane using Hough Transforms and finding matchings between these centre's by finding epipolar lines in the viewing image from points in the reference image. The identification of correspondence was the largest source of erroneous results.
Once this had been acquired, the coordinates of each of these corresponding pairs in 3-D space can be estimated by finding the closest point between the two epipolar lines generated from each of the two points. This process gives an estimate of the depth of the point relative to the viewing camera. Finally the radius of each sphere can be calculated using the distance of the sphere from the camera and the radius of the circle established with the Hough Transform.
This assignment aims to investigate the effects of noise on the poses of each camera simulating the fact that in a real life calibrated system the measured rotation and translation between each camera will have some error and how this will reduce the accuracy of the 3-D reconstruction. In addition to this different sizes and positions of spheres will affect the performance of the reconstruction. For example the main source of error was if two sphere's happened to lie in the same epipolar line in the viewing image as it made it impossible to differentiate between them when correspondence matching.

1.2 Methods

1.2.1 Correspondence Matching

In order to obtain corresponding points between the two viewpoints, the fact that all the objects in the scene have the same uniform shape could be used to find a set of sphere centre's in each scene. This is achieved by using OpenCV's Hough Circle function, this function takes a grayscale image and produces a list of circles in the image. The second parameter used for tuning this method is 'method' which specifies the algorithm used, in this case cv2.HOUGH_GRADIENT uses a standard Hough transform described in [2]. The 3rd parameter is the resolution of the image used in the transform, 1 meaning the accumulator is the same size as the input image. The next parameter is the minimum distance between circle centres, this value depends on how far away the camera is from the scene as circles will appear to have closer centres the further away the camera is. 'Param1' is edge detector threshold and 'Param2' is the accumulator threshold, lower values of param2 lead to detections of more 'imperfect' circles.

```
circles = cv2.HoughCircles(
    frame_gray,
    cv2.HOUGH_GRADIENT, 1, 30,
    param1=70, param2=35,
    minRadius=10,
    maxRadius=150)
```



(a) Reference image Hough Circles (b) Viewing image Hough Circles

Once the points of circles in both views had been acquired, the epipolar lines of the circle centres in the reference image had to be projected onto the viewing image. An epipolar line in 3D space is the line that passes through the focal point and a particular point in the image plane of a camera. The epipolar constraint states that a for point p_r viewed in the reference image, it's corresponding point p_l lies on the projection of p_r's epipolar line on the viewing image plane (Figure 2). This means that the centre's of the spheres in the viewing image will lie on the epipolar lines formed from their centre's in the reference image. This fact can be used to identify a correspondence between the centres.

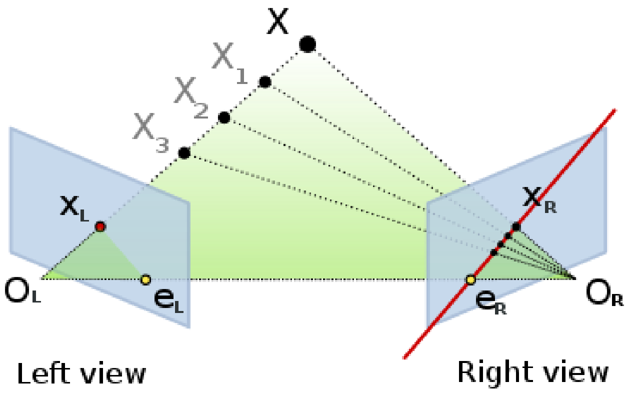
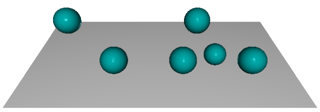
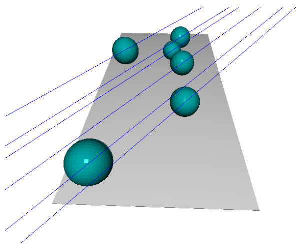


Figure 2: Epipolar Constraint



(a) Reference image



(b) Viewing image with epipolar lines

An approximate correspondence is found by finding pairs of epipolar lines and circle centres in the viewing image that have the smallest separation distance (Figure 4). This approach has the issue that if two circles lie on the same epipolar line, it's impossible to identify their matching circle in the reference image. This problem could be resolved by using two different reference images as the epipolar lines should intersect at the matching point.

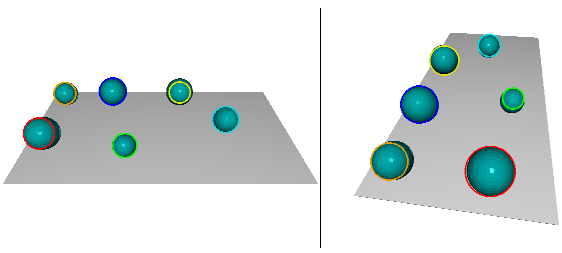


Figure 4: Correspondence Matching

1.2.2 3-D Reconstruction

The 3D position of each sphere can be approximated by finding the closest point between the two epipolar lines cast from the corresponding pair of circle centres. Since these lines will always be slightly skew from each other, we can assume the closest point on each line is ap_l and bp_r where a and b are some scalars of the the points in the image plane. Since we know the vector of the closest point to each line is perpendicular to each line we can assume that:

$$ap_l = bR^T p_r + T + c(p_l \times R^T p_r)$$

Or in other words, in viewing camera coordinates, the point on the viewing camera's epipolar line that is closest to the point on the reference camera's epipolar line is: that point plus some scaling of the perpendicular vector between the two. Since we know the rotation and translation between the two cameras we can rearrange this equation to:

$$T = ap_l - bR^T p_r - c(p_l \times R^T p_r)$$

$$H = \begin{bmatrix} p_{lx} & (R^T p_r)_x & (p_l \times R^T p_r)_x \\ p_{ly} & (R^T p_r)_y & (p_l \times R^T p_r)_y \\ p_{lz} & (R^T p_r)_z & (p_l \times R^T p_r)_z \end{bmatrix}$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = H^{-1} T$$

Solving for a, b and c we can use those scalars to take an average of the closest point on each line.

$$P = \frac{ap_l + bR^T p_r + T}{2}$$

This point is an estimate of the 3D-position of the sphere, showing that we can extract depth information from two images by knowing corresponding points and the transform between the two viewpoints.

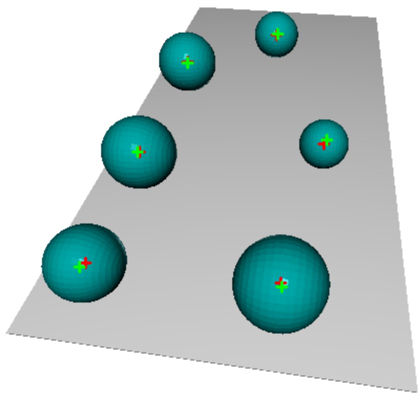


Figure 5: Red = Ground Truth, Green = Predicted Locations

Finally we can estimate the radius of each sphere by using the law of similar triangles (Figure 6), we know the radius of the Hough circles which is the projection of each sphere onto the image plane and now we know the distance of the centre of each sphere from the image plane.

$$Z = \text{Depth},\ f = \text{Focal Length},\ r = \text{Circle Radius}$$

$$\frac{Zr}{f} = R = \text{ Sphere Radius}$$

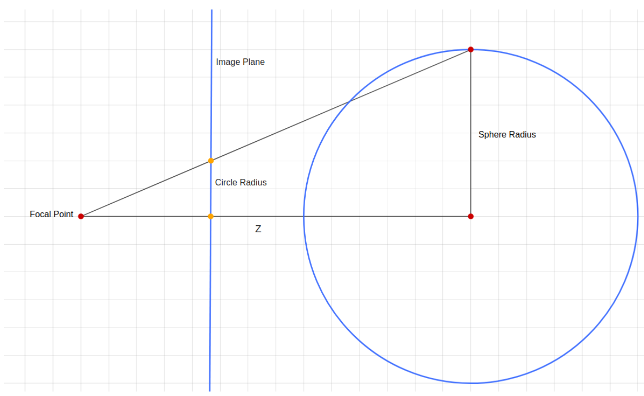
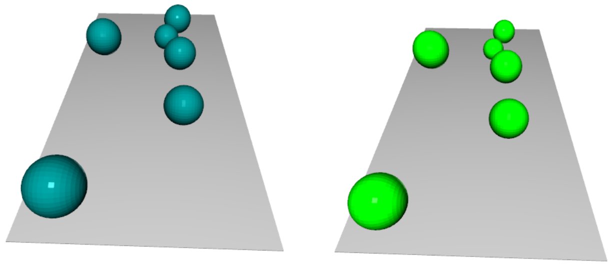


Figure 6: Similar Triangles

We can now use the centres and radii computed to display the sphere estimates alongside the ground truth in the 3D simulation.



(a) Ground Truth (b) Generated Spheres

In order to gain a metric of performance of the implementation, we can compute the average error in distance between the ground truth and the generated spheres. To achieve this, a correspondence between the predicted spheres and the ground truth is required. This can be done by matching the Hough circles used to generate the predicted spheres with the ground truth by finding the position of each real sphere in camera space and minimising the distance between pairs of circles and spheres.

1.3 Experimentation and Results

One of the main failure cases during testing of the implementation was when two sphere's we're colinear in their epipolar lines, meaning it was impossible to tell them apart in terms of their correspondence with the two sphere's in the other view. It is evident in figure 8 and 9 that the red circle has been misidentified in the second viewpoint.

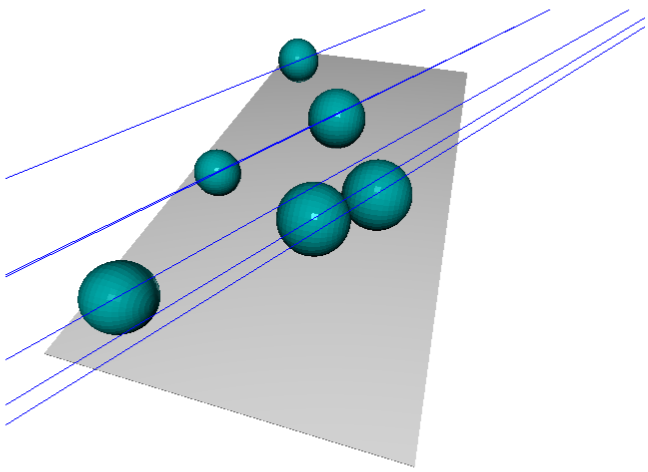


Figure 8: Colinear Epipolar lines

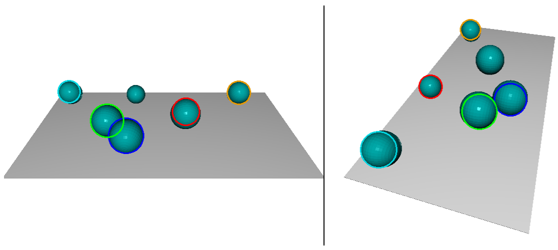


Figure 9: Resulting correspondence

This issue is then compounded when we try to estimate the sphere's position and radius. This is clear to see in figure 10 where one of the spheres appear behind the plane because the intersection of the epipolar lines in 3D between the centres of the two different spheres happens behind the plane.

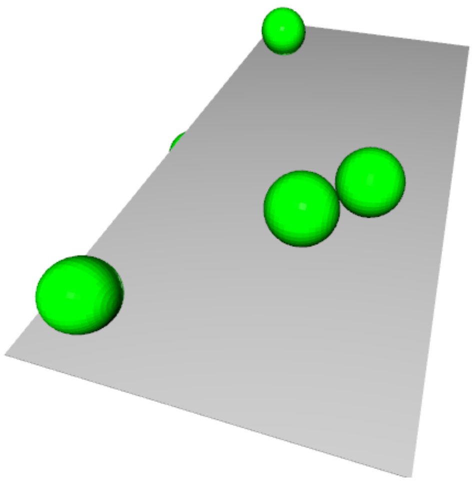


Figure 10: Resulting generated spheres

Another failure mode for the implementation was if in either of the views one of the spheres was partially occluded by another as the Hough Transform would fail to detect it, minimising this error came down to tuning the Hough functions parameters. However it could not be optimised for every possible case as making the detector more sensitive would lead to more false positives. A second issue with the Hough transform is that if areas of the spheres we're in shadow then it would find a circle at the terminator of the shadow rather than the edge of the sphere, this lead to a small offset in the computed centre of each sphere.
In order to simulate error in the measurements of the rotation and translation between two cameras in a real-world calibrated setup, random noise was added to each camera's position and orientation. The original translation and rotation we're then passed to the 3-D reconstruction routines. For determining the effects of different amounts of noise on the measured reconstruction error, the noise was sampled from a range with bounds that we're incrementally increased but the size of range remained constant (Ensuring that at each increment a minimum amount of noise is always added). For each increment 20 iterations we're run and the error was averaged across each of them. The results of this for translation noise can be seen in figure 11. It's clear to see a positive correlation between translation noise and the error in the computed position of the sphere, this is because as the noise increases the true intersection point of the epipolar lines moves further away from the predicted point. For both the rotation noise and translation noise experiments, the initial location and rotation of each camera is kept constant across all the iterations, as well as the locations of all the spheres.

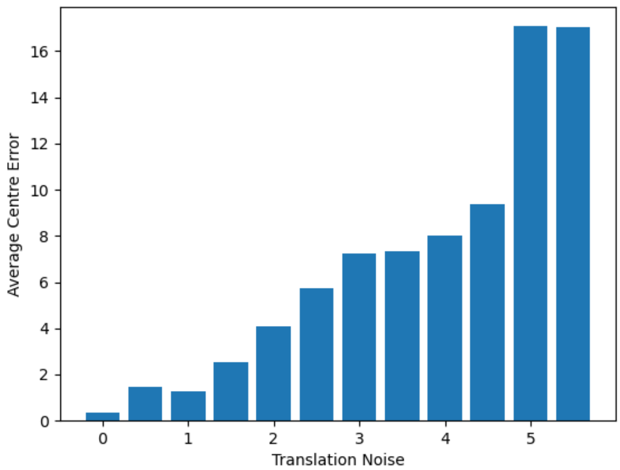


Figure 11: Average error given varying levels of camera translation noise (20 iterations per increment)

The same trend can be observed when the same process is applied to rotational noise. This is clear to see in figure 12. However in both of these plots there outliers, part of this is due to the fact that the noise is randomly sampled in a range so there will be variation between each iteration and but the main source of error is where certain configurations of camera orientations that produce co-linear epipolar lines. It is evident that small changes in position lead to much more error in the centre positions compared to small changes in rotation.

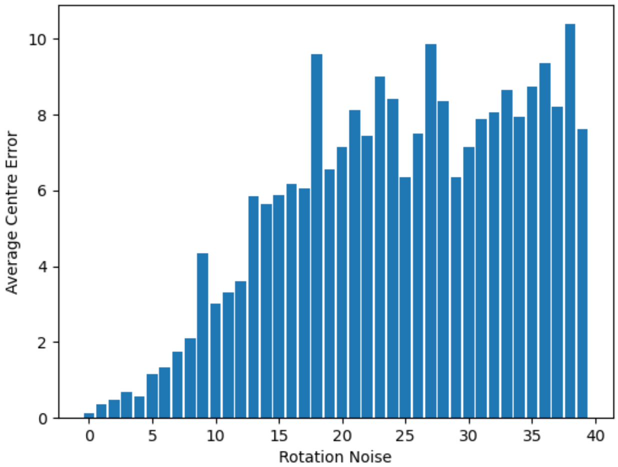
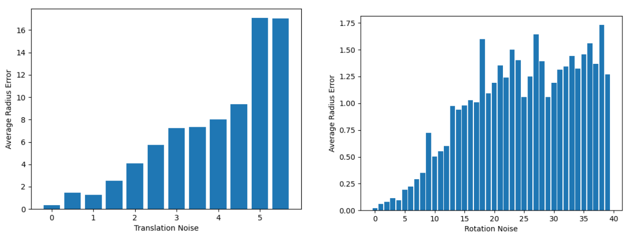


Figure 12: Average error given varying levels of camera rotation noise (20 iterations per increment)

The error in the calculated radius served as another performance metric for the 3-D reconstruction (figure 13a and 13b). This measure directly mirrors the error in centre position because the radius is computed using the centre position and so an error in the depth of the centre will lead to an error in the calculated radius. However the magnitude of the errors in radius was much less than the magnitude of the errors in centre position. In order to reduce these errors two or more reference cameras could be used, minimising the error in the measurement of position and rotation between the reference camera and the viewing camera.



(a) Translation noise, radius error

(b) Rotation noise, radius error

Figure 13

1.4 Conclusion

In summary, by far the largest problem with the approach was with the correspondence matching as it leads to the intersection of inaccurate epipolar lines and therefore inaccurate 3-D reconstruction. The affects of this could be mitigated by implementing a threshold on the distance between the closest points on the two lines. This prevents erroneous results from being added to the scene however it doesn't fix the correspondence issue. Since the scene is very uniform where the only objects in the scene are the same shape and colour, correlation-based correspondence matching (Scanning images with sub-windows [5]) probably won't be very effective. A better approach to correspondence matching could be to use a feature based method such as SIFT (Scale Invariant Feature Transform), transforming the image into a set of feature vectors that are invariant with respect to scaling, rotation and translation [3]. If a reliable method of correspondence matching is used then the biggest intractable source of error is object occlusions in the viewing image.
Throughout the experimentation process it became clear that having accurate values for the positions and rotations of each camera is very important however if we have a robust method for identifying correspondences then we can use those corresponding points to estimate the fundamental matrix (Matrix that contains the transformation information between the cameras) using singular value decomposition[4] [1]. If this can be done accurately then it would eliminate the requirement for the setup to be calibrated.

References

[1] R.I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.

[2] John Illingworth HK Yuen, John Princen and Josef Kittler. Comparative study of hough transform methods for circle finding. image and vision computing. pages 71—-77, 1990.

[3] David Lowe. Object recognition from local scale-invariant features. 1999.

[4] M. Pilu. A direct method for stereo correspondence based on singular value decomposition. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 261–266, 1997.

[5] Sungchul Kang Yoon Keun Kwak Sukjune Yoon, Sung-Kee Park. Fast correlation-based stereo matching with the reduction of systematic errors. 2004.