IPCV Part II

Introduction

I produced a stereo vision system that reconstructs spheres within three-dimensional space given two images of these spheres, taken from calibrated camera viewpoints. A circular Hough transform is applied to each image to determine the location of the circles in the image, which represent the spheres projected onto the image plane. Taking one of the images as the reference view, the epipolar lines for each of the circles is computed and visualised within the other image. The epipolar constraint equation is then used to calculate circle correspondences between the two images. Using this information, the spheres are reconstructed in 3D space and visualised next to the true spheres.

Method

Method – Hough Transform

Given two images, the circular Hough transform is applied to each one to detect all the circles within the image. These circles correspond to the spheres. Instead of passing in the grayscale image, I chose to pass in the inverted red colour channel instead, as this led to a clearer contrast between the circles and the background. I apply Gaussian blur to the image to make it smoother, as suggested by the documentation for OpenCV's Hough circle method [1]. The Hough transform then detects the circles within the image.

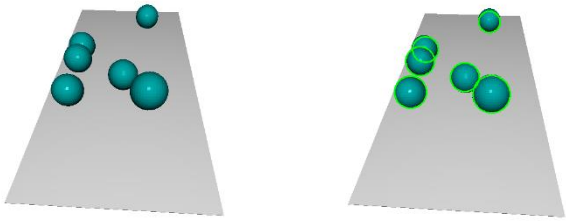


Figure 1 – Visualisation of circular Hough transform.

Method – Epipolar Lines

For a point within a reference image, the epipolar line is the line within the other image which represents all possible locations for the same point within that image.

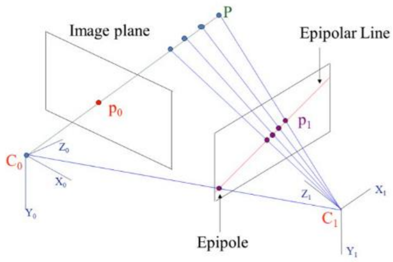


Figure 2 - Example of epipolar line for point p_0. Figure taken from [2].

For a calibrated stereo system, the epipolar line can be calculated by computing the essential and fundamental matrices and applying the latter to points within the reference image. For this task, there are two cameras C_0 and C_1 that produce images $image_0$ and $image_1$. Both cameras have the same focal length, f. As the cameras are calibrated, we know their exact positions, which are given by $H_0^{w \to c}$ and $H_1^{w \to c}$. These matrices represent the homogeneous transformation that will take a point in world coordinates to the coordinate space system of C_0 and C_1 respectively. Note that taking the inverse of these matrices defines the homogenous transformation from camera coordinates to world coordinates, $H_0^{c\to w} = (H_0^{w\to c})^{-1}$ and $H_1^{c\to w} = (H_1^{w\to c})^{-1}$.

For the epipolar line calculation, I have chosen to use $image_0$ as the reference image. Using the calibrated viewpoints, we can calculate the relative pose between the two cameras $H^{C_1\to C_0} = H_0^{w\to c}\, H_1^{c\to w}$. As $H^{C_1\to C_0}$ is homogenous, the matrix stores both the translation and rotation between the two cameras. We use this to extract T, the translation vector, and R, the rotation matrix. Thus, the essential matrix is defined as $E = RS$, where $S = \begin{pmatrix} 0 & -T_2 & T_1 \\ T_2 & 0 & -T_0 \\ -T_1 & T_0 & 0 \end{pmatrix}$. In order to convert pixel coordinates to image coordinates, we define a matrix $M = \begin{pmatrix} s_x & 0 & \frac{(-\hat{o}_x\, s_x)}{f} \\ 0 & s_y & \frac{(-\hat{o}_y\, s_y)}{f} \\ 0 & 0 & 1 \end{pmatrix}$ where $s_x = 1$, $s_y = 1$ and $(\hat{o}_x, \hat{o}_y)$ is the centre of the image in pixel coordinates. The fundamental matrix is then calculated as $F = M^T E\, M$.

Using the fundamental matrix, we calculate $u = Fp_0$ where $p_0 = [x\ y\ f]$ where x and y are the circle centres detected by the Hough transform. I then use $p_1 u = 0$ to draw the epipolar line.

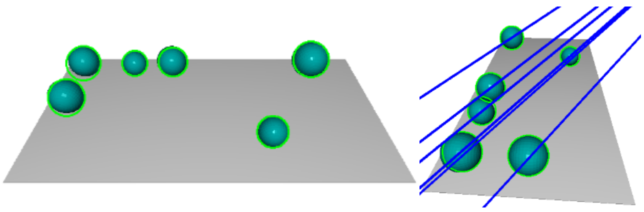


Figure 3 – Epipolar lines from left image visualised in right image.

Method – Circle Correspondence

I utilised the epipolar constraint equation ${p_1}^T F p_0 = 0$ to match the circles between the two images. Due to factors such as error in circle centre detection, it is unlikely that the constraint will be satisfied. Hence, I compute the correspondence that the minimises the sum of the absolute values of the epipolar constraints for all pairs in the correspondence. This is calculated by reducing the problem to the linear sum assignment problem [3], where the cost matrix is the epipolar constraint for each possible circle pair.

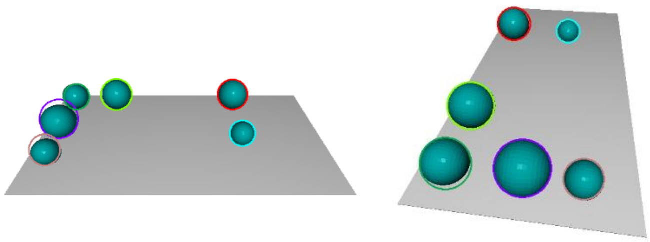


Figure 4 – Circle correspondences visualised.

Method – Sphere Centre Reconstruction

To reconstruct sphere centres within 3D space, I use the circle centre pairs (p_0, p_1) to compute the sphere centre within the 3D world coordinates P_{world}. To do this, I solve $ap_0 - bR^T p_1 - T -$

$c(p_0 \times R^T p_1) = 0$ for a, b and c. The reconstructed point in the camera coordinate system of $C_0$is $P_{C_0} = \frac{(ap_0 + bR^T p_1 + T)}{2}$. We convert this into world coordinates by applying the homogenous transformation $H_0^{c \to w}$. These points are displayed in a 3D visualisation where the true centres are displayed in red and reconstructed centres are displayed in blue. The error is computed by calculating the mean distance between the true centres and the reconstructed centres.

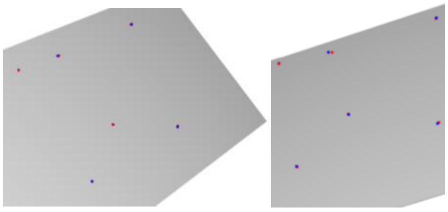


Figure 5 – Sphere centre reconstructions.

Method – Sphere Reconstruction

In order to estimate the radius, I project a point on the image circle circumference into the camera coordinates of both cameras and then into the corresponding world coordinates. The distance between this and the reconstructed centre is the estimated radius. The average of both estimates is used as the final estimate. In order to project the point on the circle circumference from image coordinates to camera coordinates, I use the perspective projection equation $P = \frac{Zp}{f}$, where Z is the depth of the circle centre in camera coordinates.

In order to visualise the radii, I combine the radii and centres estimates to draw estimated spheres and compare those to the ground truth spheres. Both sets of spheres are rendered within a 3D visualisation using red and blue point clouds. Point clouds are used instead of solid spheres as solid spheres would result in large spheres completely occluding smaller ones. The radius estimation error is calculated as the mean difference between the true radius and the estimated radius.

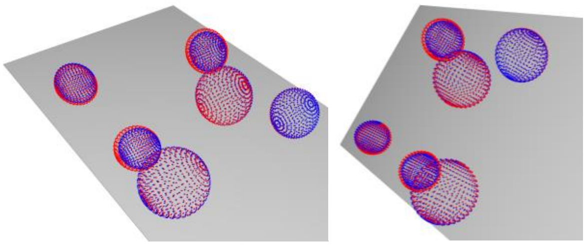


Figure 6 – Sphere reconstructions. (Reconstructed spheres in blue, ground truth in red).

Experiments and Results

Experiments and Results – Hough Transform

The Hough transform using the following parameters:

$$minDist = 15, gradThresh = 30, minRad = 5, maxRad = 50, houghThresh = 18$$

I selected these parameters by running simulations using the default configuration and manually tuning the parameters. Accurate circle detection is critical as all subsequent parts of the sphere reconstruction process rely on these estimates.

Using the default sphere generation configuration, I evaluated the transform's accuracy on 20 images. For 75% of the images, all circles were correctly detected. Overall, 92% of circles were accurately detected. The transform struggles most when spheres overlap, and when perspective distortion leads to sphere projections appearing elliptical in images. *Figure 7* demonstrates both of these cases, with the left image showing an ellipse being classified as two circles and the right image showing two closely grouped circles being detected as one.

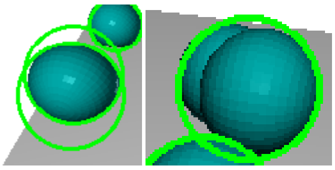


Figure 7 - Two incorrectly detected circles.

The $minDist$, $minRad$ and $maxRad$ parameters are given in pixels and thus are dependent on the sphere sizes and separations. For most reasonable values, my system works effectively. However, it struggles with large sphere radii and with small separation distances between sphere centres as these lead to more overlapping circles, which lead to more erroneous detections. The Hough transform also struggles with circles that are at the plane boundary, as there is a contrast between the plane colour and the background (which means gradient information is present).

Experiments and Results – Epipolar Lines

My epipolar line calculation is clearly correct, as when both sets of circles are correctly detected by the Hough transform, the epipolar lines pass through the centre of each circle within the viewing image (see *Figure 3*). Errors within this part only occur due to Hough circle detection errors.

Experiments and Results – Circle Correspondence

The circle correspondence is correct in the vast majority of cases (see *Figure 4*). It produces incorrect correspondences when two circles lie along the same epipolar line in the viewing image, as visualised in *Figure 8*. In this case, the purple and light green circles lie along the same epipolar line and therefore have the same epipolar constraint value. Thus, the correspondence is incorrectly calculated.

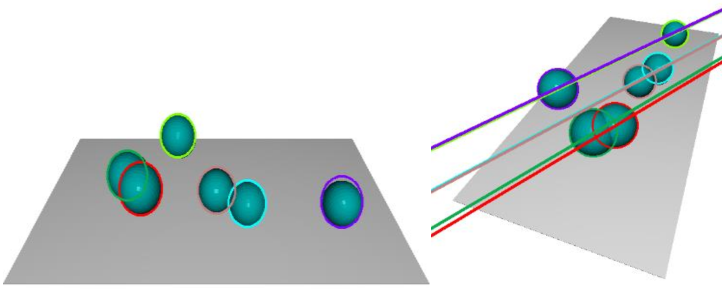


Figure 8 – Incorrect correspondence with epipolar lines visualised.

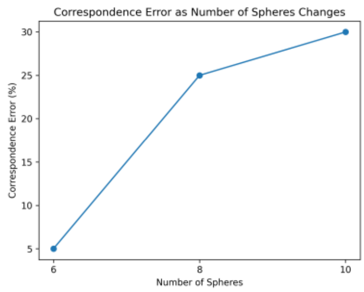


Figure 9 – Change in correspondence error with number of spheres.

Additionally, cases where the circle detection fails to detect all the circles or detects too many circles lead to incorrect classifications. I investigated the relationship between the number of spheres and the percentage of images for which the correspondence is incorrectly calculated. I did this by manually verifying the correspondence for 20 images for each of the sphere sizes. The result is shown in *Figure 9* and demonstrates that a greater number of spheres leads to a higher reconstruction error. This is because the chance that a sphere will lie on the epipolar line of another sphere in the viewing image increases with the number of spheres.

Experiments and Results – Sphere Centre Reconstruction

The sphere reconstructions tend to be accurate and produce a low error. Occasionally, an estimated centre will be very far from the true centre. This occurs due to erroneous circle pairing. As discussed in the previous section, this is usually because two circles lie on the same epipolar line, or because the Hough transform doesn't correctly detect all circles.

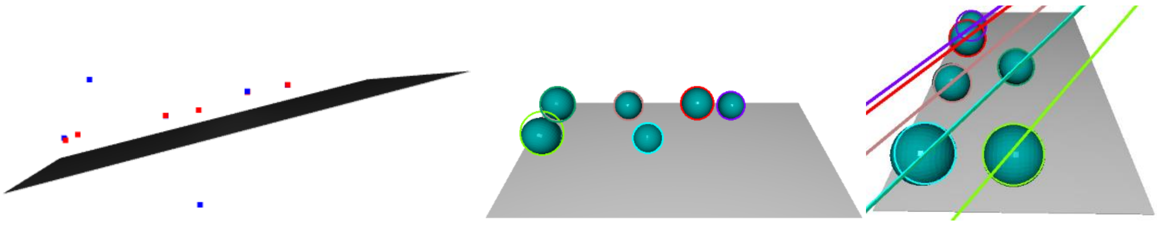
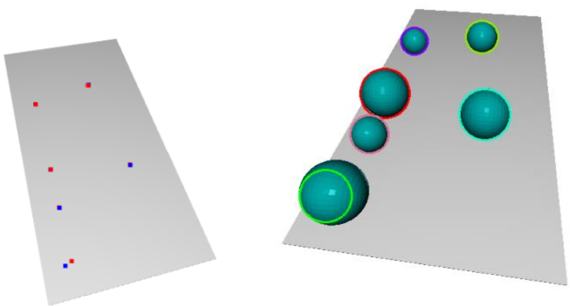


Figure 10 – Incorrect sphere centres caused by faulty correspondence.

Figure 10 demonstrates 4 out of 6 circle centres being correctly reconstructed whilst the other two centres are significantly above and below the true points. This is due to the incorrect correspondences caused by the light blue and turquoise epipolar lines being the same.

I measured the mean distance between the sphere centres and the reconstructed centres across a number of different images. I also tracked the number of accurately reconstructed circle centres (defined as being reconstructed within 10% of the radius of the true sphere). A mean distance of 0.26 and a median distance of 0.11 are achieved. The average number of sphere centres correctly reconstructed is 5 out of 6. This indicates that the reconstruction algorithm performs well. The mean distance is broadly broken down into three categories. $d < 0.1$ normally represents perfect reconstructions. $0.1 \leq d < 1$ occurs when the majority of points are correctly reconstructed (5 or 4 out of 6). The additional error in these cases is normally due to Hough circles being broadly correct but the actual centre being slightly off. *Figure 11* shows an example of this.



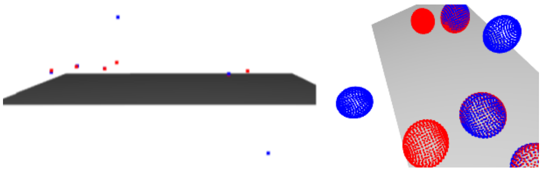
$d \geq 1$ usually occurs when the correspondences are incorrectly calculated, as shown in *Figure 10*. 50% of cases fall into the first category, 40% fall into the second category and 10% fall within this final category.

Figure 11 – Bottom left circle not accurately bound by Hough detection and hence a modest reconstruction error occurs.

<u>Experiments and Results – Sphere Reconstruction</u>

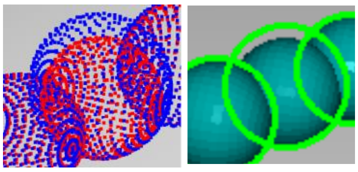
To evaluate the radius estimation calculation, I recorded the mean radius error and the number of correct radius estimates (defined as an estimate within 10% of the true value) over a range of test images. Across 20 samples, my system achieves a mean radius error of 0.23 and correctly calculates 2 out of 6 radii on average. Visual inspection confirms that most radius estimates are correct within a modest error margin and look similar to *Figure 6*. The two main causes of error occur when the correspondence is calculated incorrectly and when the circle detection is inaccurate.

When the correspondence is calculated incorrectly, the spheres don't match and thus the radius estimate is the mean between two different radii. However, as spheres are generated from within a random range $[radMin, radMax]$, the error is a function of this range. Typically, the error is around



0.5 in these cases which is large but not as large as expected, as spheres have an expected radius of $\frac{radMin+rad\ Max}{2}$. An example of this is shown in *Figure 12*.

Figure 12 – Correspondence matching error leading to inaccurate radius estimation.



Most radius estimation errors occur due to the Hough circle transform inaccurately calculating the radius. *Figure 13* shows a case where the radius size is overestimated, leading to a corresponding overestimate of the sphere radius. This shows that the sphere reconstruction approach works but is susceptible to errors in previous steps.

Figure 13 – Hough circle detection incorrectly estimating radius, leading to a radius estimate error.

Experiments and Results – Failure Modes as Sphere Properties Vary

My system performs well across a reasonable range of possible parameters. My system struggles with large circles that exceed the *maxRad* parameter from the Hough transform, as it is unable to detect these, as highlighted in *Figure 14*. Similarly, spheres that are smaller than *minRad* are also not detected, as shown in *Figure 15*. Both cases contribute to reconstruction error.

Increasing the separation between sphere centres improves the performance as there is less overlap between spheres. This leads to more accurate circle detection, and thus reduces reconstruction error. Conversely, when the sphere separation is small, spheres are more likely to overlap, leading to misclassification by the Hough detector, as illustrated in *Figure 16*.

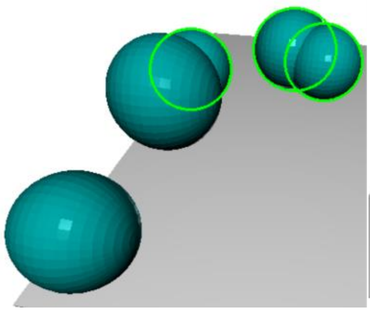


Figure 14 - Large spheres not detected.

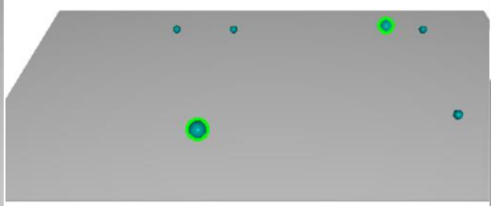


Figure 15 – Small spheres not detected.

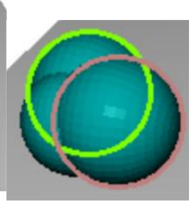


Figure 16 – Close sphere centres.

Experiments and Results – Noise Analysis

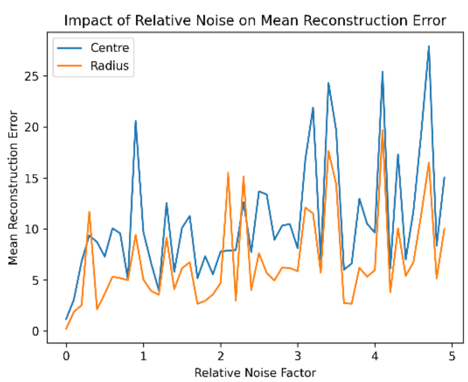


Figure 17 - Reconstruction error as noise varies.

I evaluated the impact of noisy relative pose by adding Gaussian noise to T and R (which represent the relative pose between cameras). *Figure 17* shows the results of this. As noise level increases, so does the reconstruction error. Noise is added to pose by adding values sampled from $\sim N(0, noise_{fact})$. Even with small amounts of noise, no circles are accurately reconstructed. This is because the essential matrix calculation is dependent on the relative pose, and more noise leads to a greater error with the essential matrix calculation. Thus, the epipolar lines are incorrect, which leads to an invalid correspondence and hence erroneous reconstructions.

Conclusion

I have produced a robust and effective stereo vision system. It is able to accurately reconstruct spheres across a wide range of possible sphere configurations. For ideal configurations where spheres are well-separated and not occluded in both views, the system produces near perfect reconstruction. The biggest flaw in the system is the Hough circle detection. Whilst the parameters have been tuned for strong performance, it often fails to correctly estimate all circle centres and radii. I would look to make use of colour information more effectively to filter all non-circle objects from the image, allowing for more accurate detections. To address flaws that occur due to scale issues, I could provider looser $minDist$, $minRad$ and $maxRad$ bounds to the Hough transform, however this would slightly harm performance on reasonably sized circles. I could also use an elliptical Hough transform to avoid issues relating to perspective distortion.

Currently, the system selects one image as the reference view to reconstruct sphere centres. To improve accuracy, I could try selecting both images as the reference view and taking the average of the centre estimates.

In conclusion, I have produced a strong sphere reconstruction system that performs effectively.

References

[1] *OpenCV Hough Circle Documentation*. OpenCV. https://docs.opencv.org/4.x/dd/d1a/group__imgproc__feature.html#ga47849c3be0d0406ad3ca45db65a25d2d (accessed on 06/12/2023).

[2] *EGGN 512 - Lecture 23-1 Epipolar and Essential*. Professor William Hoff, Colorado School of Mines. https://www.youtube.com/watch?v=Opy8xMGCDrE (accessed on 06/12/2023).

[3] *SciPy v1.11.4 Manual - scipy.optimize.linear_sum_assignment*. SciPy. https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linear_sum_assignment.html#scipy-optimize-linear-sum-assignment (accessed on 06/12/2023).