Assignment 1: AutoCalib

Alex Chiluisa
Used on late day

I. Camera Calibration

This paper reports on the process to perform camera calibration following the approach described by Zhang [1]. First, I obtained the real-world (M) points (x,y) corresponding to the checkerboard image set. Removing the edges of the checkerboard, we have a final board of 5x8 with 6x9 points with a width of 21.5 mm. Then, by using the cv2.findChessboardCorners function [2], I found the image coordinates of the checkerboard and ensure the order between m and M matches for all the images.

A. Approximate Camera Intrinsic Matrix

Following the notation and equations described in section 2 in [1], I found the camera intrinsic matrix given by K in eq. 1. (Zhang's paper uses A as the camera intrinsic matrix, this report uses K)

$$K = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 0 \end{bmatrix} \tag{1}$$

Where $(u0, v0)$ are the coordinates of the principal point, α and β are the scale factors in image u and v axes, and γ is the parameter describing the skewness of the two image axes.

Using the pinhole model, we define the relationship between the point correspondences m and M as shown in eq. 2

$$s\tilde{m} = K[R \quad t]\tilde{M} \tag{2}$$

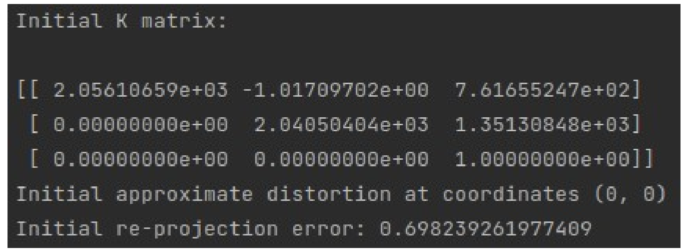
Where $\tilde{m}$, and $\tilde{M}$ are the augmented vector, refer to section 2.1 in [1]

Following the approach in section 3.1 in [1], I estimated the parameters on matrix K, by computing matrix V_i. To accomplish this task, I calculated the homography matrix H between the point correspondences m and M for n images, being n the number of images in the image calibration set. Having a total of 13 images, the size of V is given by $2nx6$, in this case, $V_{[26x6]}$. Therefore, we can solve the homogeneous equations in b:

$$Vb = 0 \tag{3}$$

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0 \tag{4}$$

A. Chiluisa is with the Department of Robotics Engineering, Worcester Polytechnic Institute, Worcester, MA 01609, USA (e-mail: ajchiluisa@wpi.edu)



```
Initial K matrix:

[[ 2.05610659e+03 -1.01709702e+00  7.61655247e+02]
 [ 0.00000000e+00  2.04050404e+03  1.35130848e+03]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Initial approximate distortion at coordinates (0, 0)
Initial re-projection error: 0.698239261977409
```

Fig. 1. Initial results of camera calibration

Where $b = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]$ and can be computed by using the singular value decomposition (SVD) [3]. Then using the appendix B in [1], we can find the parameters of the intrinsic matrix K.

B. Approximate Extrinsics

Here, I find the rotation and translation of the camera following the process in section 3.1 in [1]. Using the inverse matrix of K and homography of each images, I re-project the coordinates of M into every image plane given by the image coordinates $\hat{m}$. To accomplish the projection of M, I defined the homogenous coordinates of $M_{[1x4]}$, then find the product X' between the extrinsic matrix Rt and the homogenous coordinate M. subsequently, I found the norlmalized coordinates x, y and the radius of distorsion $r = \sqrt{(x^2 + y^2)}$. Then, I compute the pixel image coordinate U' and find U. Finally, I compute $\hat{m} = [\hat{u}, \ \hat{v}]$ by using eq. 12 and eq. 13 in [1]. Where k_1 and k_2 are the radial distortion coefficients. Initially, I assumed $k = (k_1, k_2)^T = (0, 0)^T$.

By knowing m and $\hat{m}$, the projection error can be computed to estimated the accuracy the accuracy of the intrinsic and extrinsic parameters. The results of the initial estimation are shown in Fig. 1. The re-projection error is 0.6982

C. Non-linear Geometric Error Minimization

In this section, I optimized the intrinsic ($u0, v0$, α, β, and γ), extrinsic parameters (Rt), and distortion coefficients (k_1, k_2). By using the scipy.optimize function and a LM optimization, the parameters described above were optimized. An error vector was computed using a L2 norm as is described in [4], [5]. The results of the optimization parameters are shown in Fig. 2. The re-projection error, after optimization is 0.6887

II. Conclusion

Once the images have been calibrated and optimized, the re-projected points are displayed over the calibrated images

```
Optimized K matrix:

[[ 2.05610337e+03 -1.01765705e+00  7.61661379e+02]
 [ 0.00000000e+00  2.04049450e+03  1.35132308e+03]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Optimized distortion coordinates (0.013995657616332162, -0.10372669390051421)
Optimized re-projection error:  0.680786383137751
```

Fig. 2. Optimized results of camera calibration

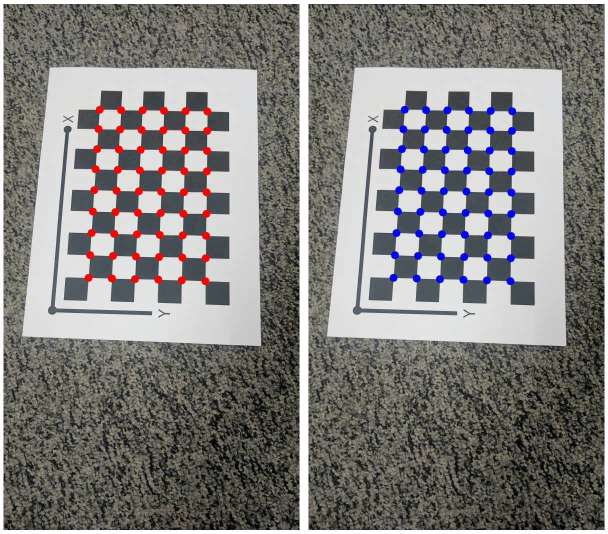


Fig. 3. Left: Original Images with detected points (red dots), right: calibrated images with re-projected points (blue dots)

(blue dots) as well as the detected points (red dots) in the original images as is presented on Fig. 3 - 15

References

[1] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
[2] https://www.geeksforgeeks.org/camera-calibration-with-python-opencv/.
[3] https://www.geeksforgeeks.org/singular-value-decomposition-svd/.
[4] https://opencv24-python-tutorials.readthedocs.io/en/stable/py_tutorials/py_calib3d/py_calibration/py_calibration.html.
[5] https://github.com/h-gokul/AutoCalib/.

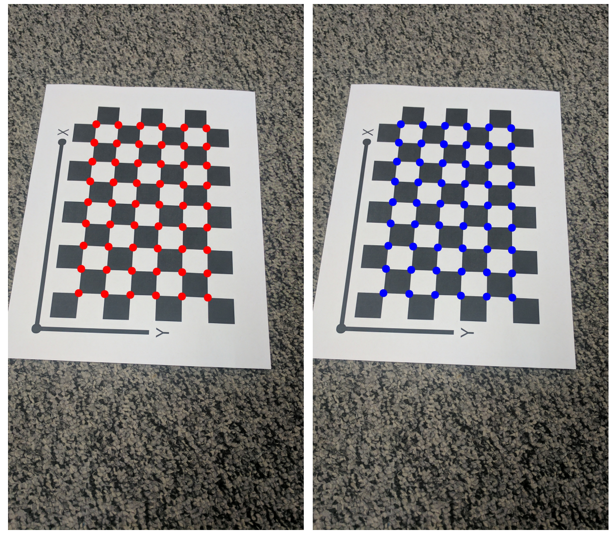


Fig. 4. Left: Original Images with detected points (red dots), right: calibrated images with re-projected points (blue dots)

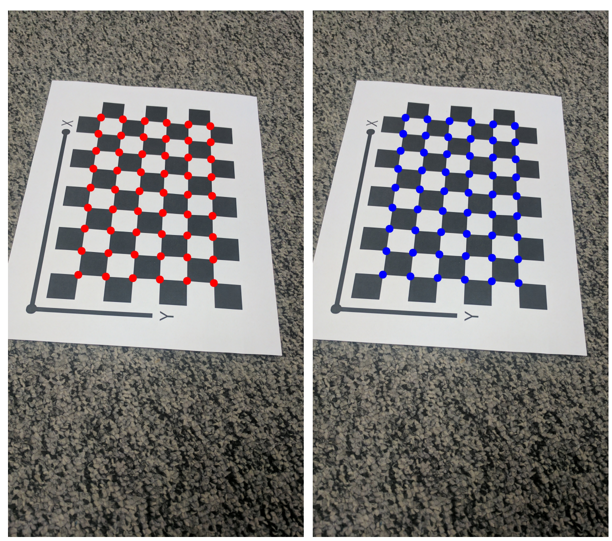


Fig. 5. Left: Original Images with detected points (red dots), right: calibrated images with re-projected points (blue dots)

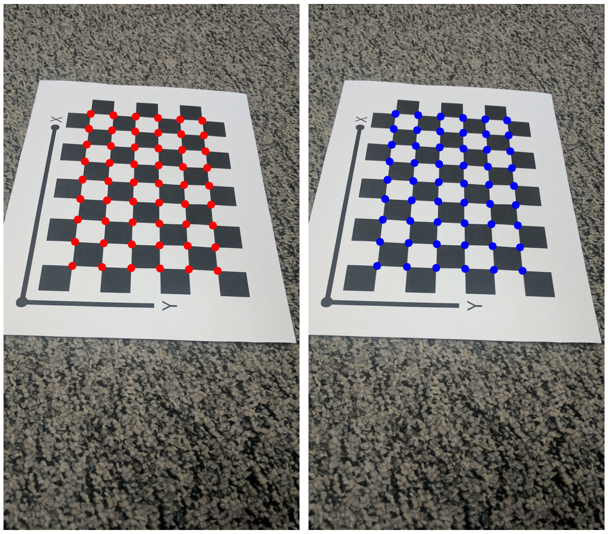


Fig. 6. Left: Original Images with detected points (red dots), right: calibrated images with re-projected points (blue dots)

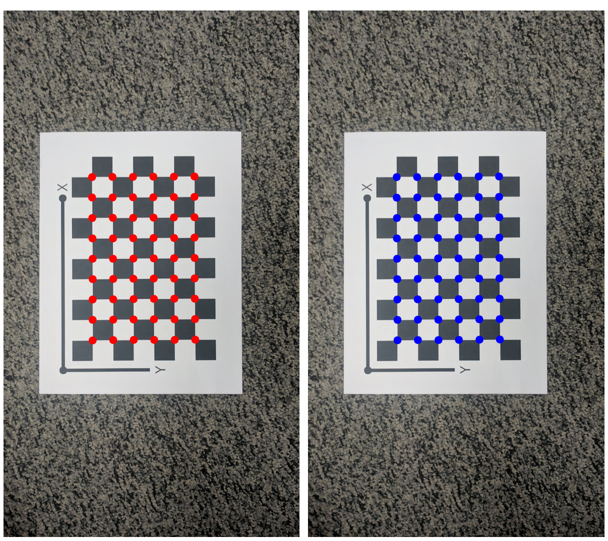


Fig. 7. Left: Original Images with detected points (red dots), right: calibrated images with re-projected points (blue dots)

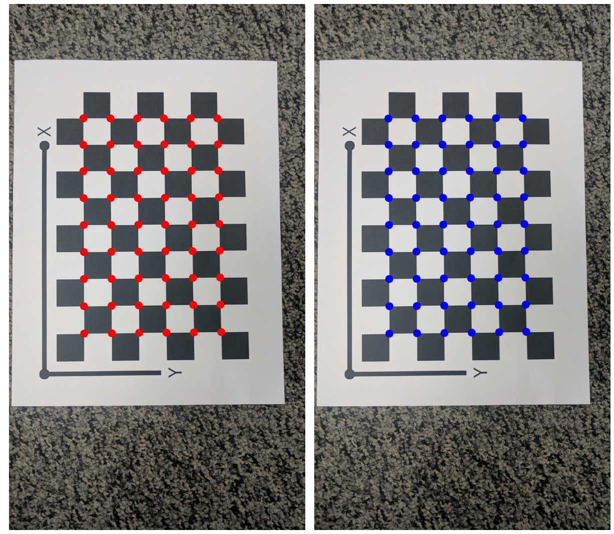


Fig. 8. Left: Original Images with detected points (red dots), right: calibrated images with re-projected points (blue dots)

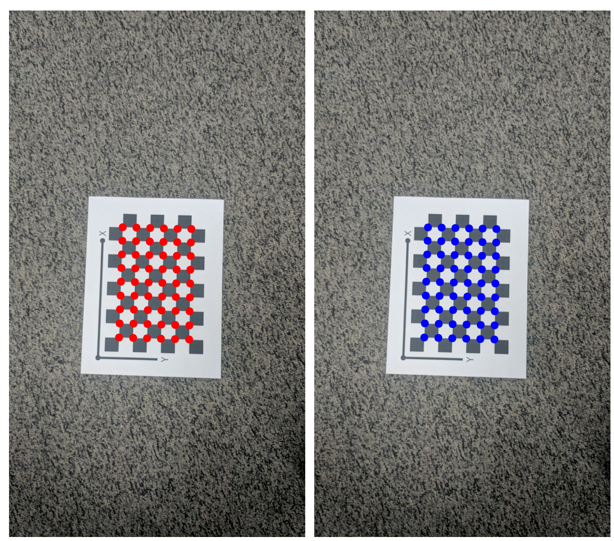


Fig. 9. Left: Original Images with detected points (red dots), right: calibrated images with re-projected points (blue dots)

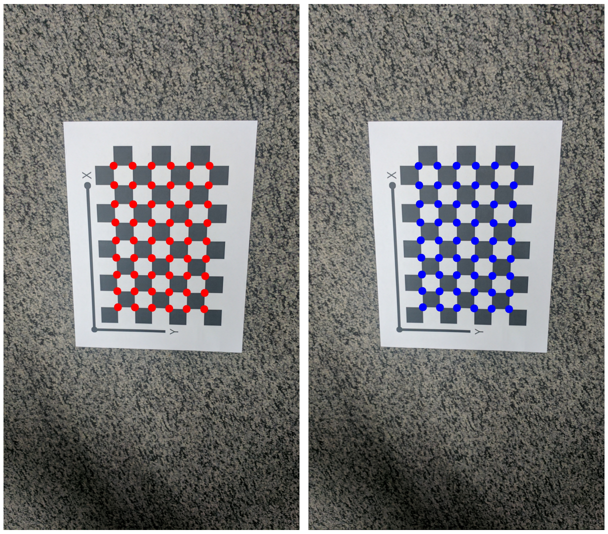


Fig. 10. Left: Original Images with detected points (red dots), right: calibrated images with re-projected points (blue dots)

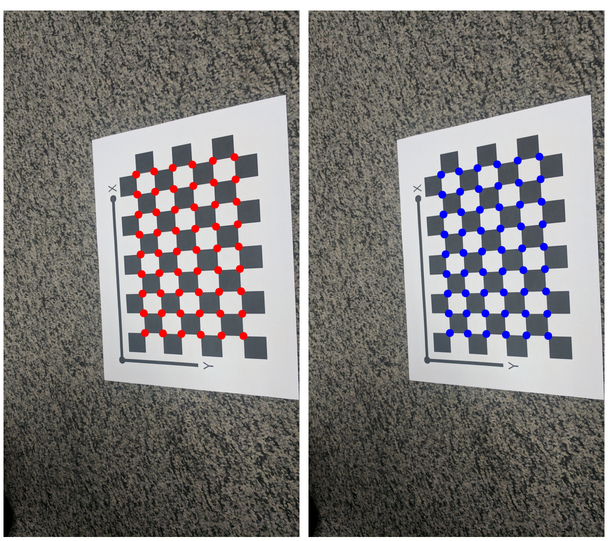


Fig. 11. Left: Original Images with detected points (red dots), right: calibrated images with re-projected points (blue dots)

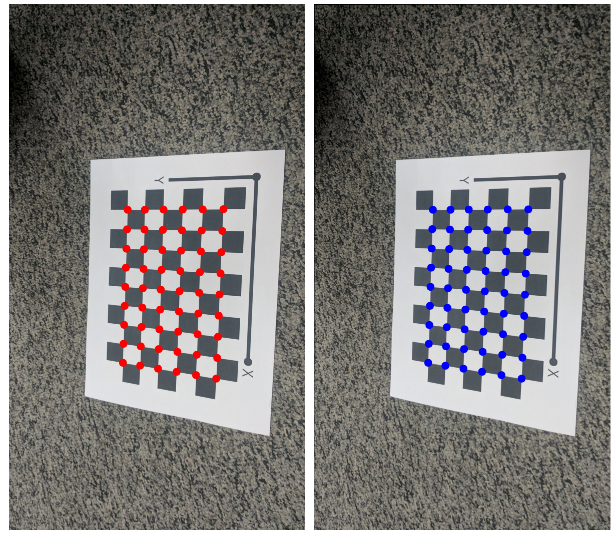


Fig. 12. Left: Original Images with detected points (red dots), right: calibrated images with re-projected points (blue dots)

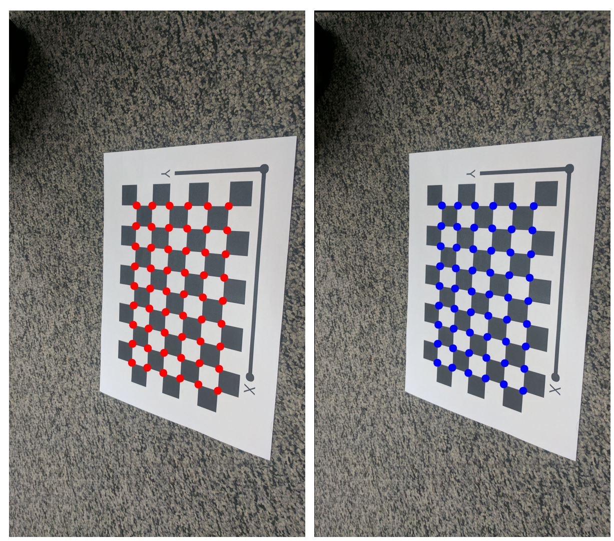


Fig. 13. Left: Original Images with detected points (red dots), right: calibrated images with re-projected points (blue dots)

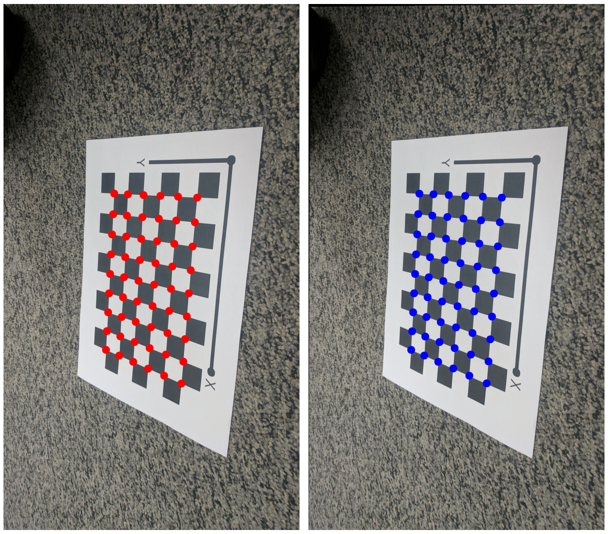


Fig. 14. Left: Original Images with detected points (red dots), right: calibrated images with re-projected points (blue dots)

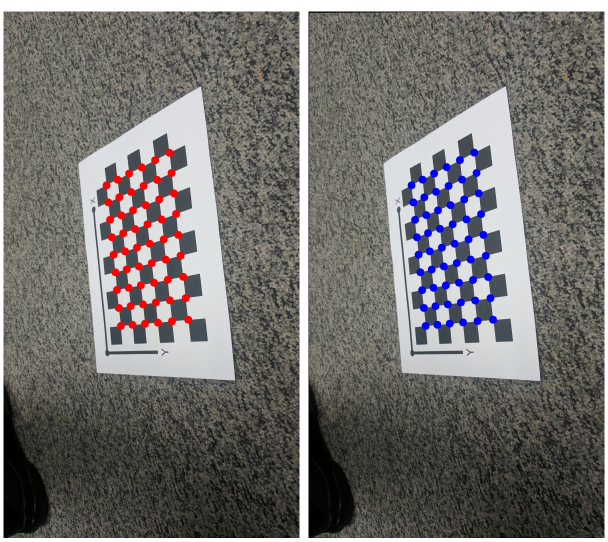


Fig. 15. Left: Original Images with detected points (red dots), right: calibrated images with re-projected points (blue dots)