Exploring The Camera Self Calibration Problem

Johan Ospina
Princeton University
jospina@princeton.edu

Abstract

In traditional Structure From Motion pipelines starting from a good initial guess generally helps the algorithms involved converge faster. Historically systems tend to utilize well structured inputs from pre-calibrated cameras in order to achieve a valid metric reconstruction (offset by a similarity transform). Only until recent efforts to use crowd sourced images for 3D reconstruction were methods for extracting camera intrinsic parameters from scene properties truly needed in order to initialize a starting guess for non linear optimizations such as Bundle Adjustment. Using properties of Projective Geometry such as the Absolute Quadric and the plane at infinity, encapsulated well by the Absolute Dual Quadric, one can find the camera intrinsic parameters per view on well conditioned sequences.

1. Introduction

Traditionally, Structure From Motion pipelines take advantage of constant or known camera parameters as they capture the world around them. This works well as the camera parameters can be known *a priori* in algorithms such as Bundle Adjustment. In such cases a camera is usually calibrated ahead of time using a fabricated object that has known coordinates[4].

Recently, interest has shifted towards creating metric reconstructions from unordered and untagged public image sets. One such example is of "Building Rome In A Day"[1] which efficiently sifted through public data sets of tourist photos and efficiently built a large scale bundle adjustment problem from the feature correspondences present. However, in both of these approaches the camera intrinsics are known *a priori*, either by manual calibration or by using EXIF tags from the cameras, and are in turn used to guide the Bundle Adjustment algorithm. The main limitation for these approaches is that an image cannot be used if it does not have known camera parameters.

In contrast, Self-Calibration is an attempt to calibrate the camera(s) by finding intrinsic parameters that are consistent with the underlying projective geometry of a sequence of images. These algorithms make no or few assumptions about the camera that imaged the scene [8]. The method proposed initially by [12] and later refined in [13] outlines a framework to solve for the Absolute Conic, an entity invariant to any metric transformations in $\mathcal{P}^3$ space. Once this entity has been identified, it can be used to create a rectifying transformation on a reconstruction defined up to a projective ambiguity in order to obtain a metric reconstruction.

What this work found is that the simple linear method explained in these works give a good approximation of the ground truth camera intrinsic parameters when the camera configurations are not degenerate and image matches are within some margin of error. While these degenerate configurations are not rare or complex, they are only a problem if all camera poses lie in that group. Important members of these degenerate configurations include pure rotations, pure translations, and orbital motions. When these degenerate configurations are observed it leads to ambiguities that would need more information in order to be resolved. However, the main allure of this method is that per image varying camera intrinsic configurations can be recovered for as long as the Dual Image of the Absolute Quadric can be uniquely identified in the scene.

2. Previous Work

Most of the self-calibration algorithms are concerned with unknown but constant intrinsic camera parameters[8]. However, In [12], camera self-calibration in the case of varying intrinsic camera parameters was also studied. Regardless, all of these methods utilize the concept of the *Absolute Conic* (Ω). Ω is a particular conic found at the plane at infinity. This conic is invariant under any similarity transformations (Rotation, translation, and scale) so its relative position to a moving camera is constant. The key insight is that this conic can be used as a virtual calibration object that is present in every sequence of images. This concept is captured well by Figure 1. In [5], constraints on the Absolute Conic are imposed based on the epipolar geometry of the views and the famous Kruppa Equations can be used to find the camera intrinsics on sequences that only use constant camera parameters.

Later Luong and Faugeras [9] use the Kruppa equations to derive systems of polynomial equations in order to solve for the Absolute Conic. The main feature of this self-calibration technique is that it does not relate all the images in a single projective frame (which is hard to do in some cases and explored in this report), only pairwise epipolar calibration [8].

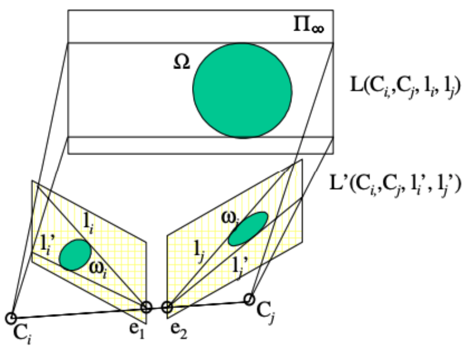

Figure 1. Example of the concept of the absolute conic embedded in the plane at infinity. Courtesy of [8]

3. Theory, Design, and Implementation

3.1. Pinhole Camera Model

A perspective camera is defined by the equation:

$$x \propto PX$$

where $\propto$ represents equality up to a non-zero scale factor. Both x and X are the 2-D and 3-D homogeneous vectors in $\mathcal{P}^2$ and $\mathcal{P}^3$ projective bases respectively. In both the Metric or Euclidean frames **P** can be factorized as follows:

$$\text{P = K[R|-Rc] where K = } \begin{bmatrix} \mathrm{f}_x & \mathrm{s} & \mathrm{c}_x \\ 0 & \mathrm{f}_y & \mathrm{c}_y \\ 0 & 0 & 1 \end{bmatrix}$$

Where f_x, f_y are the focal lengths on the x and y axes. Additionally s,c_x, c_y represent the skew in the pixel and x and y positions of the principal point on the image.

3.2. Epipolar Geometry

For a 3D point X that is imaged in two images, x and x'. A well known relationship exists relating x and x' to the epipolar geometry of this configuration. Point x' is bound to the line created in the image by the product of the fundamental matrix F and the corresponding point x such that the following relationship holds true:

$$x'^{\intercal} F x = 0$$

Additionally the Fundamental Matrix has rank 2 and the right and left null-space of F corresponds to the epipoles. One can obtain the Fundamental Matrix from two projection matrices **P** and **P'** as [13] shows:

$$F = (P'^{\intercal})^{\dagger} P^{\intercal} [e]_x$$

In the scope of this work, the ability to generate the Fundamental Matrix between two views is required. For this one can turn to the 8 point algorithm proposed by Hartley in [7]. Given a set of corresponding points on can rewrite the first Fundamental Matrix relation at the beginning of this section as:

$\begin{bmatrix} xx' & yx' & x' & xy' & yy' & y' & x & y & 1 \end{bmatrix} f = 0$

where $p_1 = [x, y, 1]$ and $p_2 = [x', y', 1]$ and f is a vector containing the elements of the Fundamental Matrix F. Stacking 8 of these equations from 8 point correspondences allows one to solve for the elements of F, though a preprocessing step where one normalizes the image correspondences so that they have a standard deviation of 1 is necessary for good results.

3.3. Homographies

A homography can be used to transfer image points that correspond to 3D points on a specific plane from one image to another. Such relationships can be denoted as $x' \propto HX$ where H is the homography that corresponds to that plane. There is an important relationship for these homographies and the Fundamental Matrix presented in [13]:

$$F \propto [e^{'}]_x H \text{ and } H = [e^{'}]_x F - e^{'} a^{\intercal}$$

where $[e^{'}]_x$ is a skew symmetric matrix that can be used to represent the cross product of the epipole and a is an arbitrary vector on a plane.

3.4. Image to Image Correspondences

For this method to work, at least 8 image to image correspondences of static objects must be available pair-wise between at least 3 views. This is a non trivial task that is still an active research area in computer vision. This pipeline utilizes two Computer Generated Imagery (CGI) data sets that provide ground truth optical flow between image sequences.

The first was the Flying Things data set [10] which provides ground truth optical flow between image sequences of various objects in a CGI environment. The second was the MPI Sintel [2] data set which is an open source CGI movie with ground truth optical flow defined for its scenes. For simplicity in the code logic, initial positions that would only follow static background elements of the scene were used. As will be shown, this proved complicated as the space of good sequences to use that had little to no moving parts was very limited. To that end two synthetic environments were created. The first was generates random cameras that image a static random 3D point cloud in a controlled environment as shown in Figure 2. The second environment was a static camera imaging a point point cloud that undergoes Rigid Body motion. While these environments are not creating images from the real world, the results acquired are applicable and show promise to these methods working on well conditioned real images.

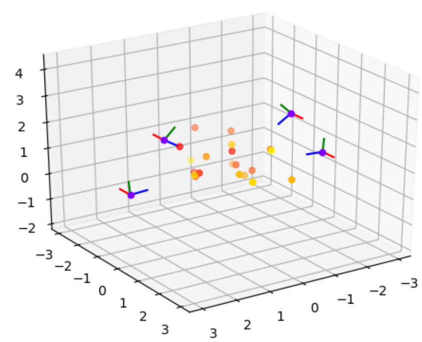

Figure 2. Example of the sample environment I create using parametric surfaces

3.5. Stratification of 3D geometry

Usually, the world is perceived as a Euclidean 3D space. In Computer Vision problems, it is sometimes not possible or necessary to understand the full Euclidean structure of 3D space. It might be interesting to deal with other types of projective geometries. One can think of these as different lenses that one can view the world through. The simplest is Projective, Followed by Affine, then Metric, and finally Euclidean structure [12]. Each one of these strata enforces stricter constraints on the geometry of a scene than the previous one with Euclidean being a perfect replication of a 3D environment. A visual example can be found in Figure 3

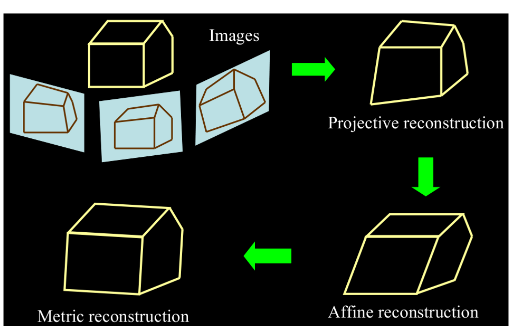

Figure 3. Example of the different stratification of 3D Geometry, curtesy of [3]

3.6. Generating A Projective Reconstruction

For an image sequence of n views, once can generate a projective reconstruction via the following steps. The first step is to calculate the Fundamental Matrix F between 2 views, in this work these are chosen to be the first 2. Once the F has been identified Pollefeys proposes in [13] that since camera movement is relative, one can choose the first camera to be the canonical camera and then choose the second camera such that its epipolar geometry aligns with the calculated F .

Therefore the first 2 out of n views can be found via the equations:

$$P_1 = \begin{bmatrix} I_{3x3} & | & 0_3 \end{bmatrix}$$
$$P_2 = \begin{bmatrix} [e_{12}]_x F_{12} + e_{12} a^\intercal & | & \sigma e_{12} \end{bmatrix}$$

These relations are derived from the ones presented in 3.3. The reader should also note that these equations are not completely determined by the epipolar geometry calculations laid out in previous sections. In order to use these relationships one needs to satisfy the remaining 4 degrees of freedom (a and σ). As noted by [13] a determines the position of the reference plane (plane at infinity in affine or metric reconstructions) and σ determines the global scale for the reconstruction. Since a specific projective reconstruction is not what is needed, one can simply set $\sigma = 1$ and $a = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\intercal$. Thus the relations can be simplified to:

$$P_1 = \begin{bmatrix} I_{3x3} & | & 0_3 \end{bmatrix}$$
$$P_2 = \begin{bmatrix} [e_{12}]_x F_{12} & | & e_{12} \end{bmatrix}$$

Once this initial structure is defined, one can triangulate a projective structure using the DLT algorithm as explained in [6]. For example every image point for P_1 and P_2 is of the form:

$$x_{1m} = P_1 X_m$$
$$x_{2m} = P_2 X_m$$

And a linear system can be defined of the form $AX = 0$ where an entry of A is:

$$A = \begin{bmatrix} up^{3\intercal} - p^{1\intercal} \\ vp^{3\intercal} - p^{2\intercal} \\ u'p'^{3\intercal} - p'^{1\intercal} \\ v'p'^{3\intercal} - p'^{2\intercal} \end{bmatrix}$$

Where $x_{im} = \begin{bmatrix} u & v \end{bmatrix}^\intercal$ and p^i is the i-th row of a projection matrix P. Once A is populated, one can *triangulate* the 3D points such that they agree with the camera matrices. The proof of this method as well as its intricacies in computation are left as an exercise to the reader, page 312 of [6] is an excellent resource and the code base for this project contains a working example.

Once the Projective Frame is initialized with the 3D points from the previous method, one can pose a simple perspective-n-point problem for each of the remaining views where the camera matrix P_n is estimated using another formulation of the DLT algorithm. This time a linear system can be defined of the form $AP = 0$ where an entry of A is:

$$A = \begin{bmatrix} 0^\intercal & -w_i X_i^\intercal & y_i X_i^\intercal \\ w_i X_i^\intercal & 0^\intercal & x_i X_i^\intercal \end{bmatrix}$$

Once A is populated with every 2D-3D correspondence between the camera and the known 3D points, one can solve for the camera matrix P_n. This process can be seen in 5

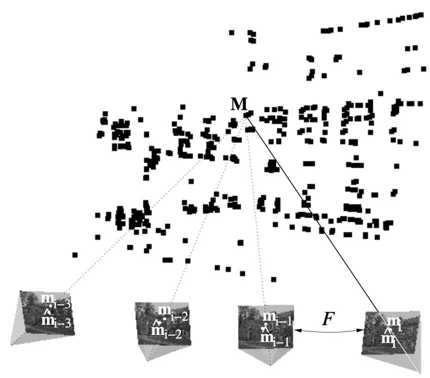

Figure 4. Visual Representation of the methods discuss thus far, First the initial structure is created from the epipolar geometry of the first and second views. Then each successive camera matrix is solved for using the methods described previously

The resulting camera matrices retrieved from this process will differ by a Projective Transformation and thus will not make much visual sense, however one can use metrics like reprojection error in order to verify correctness of the Projective Reconstruction.

3.7. Pollefeys' Method

Once a Projective Reconstruction of a scene is available, one can use the methods proposed by Pollefeys in [12] and later refined in [13]. As mentioned previously in Figure1, one of the most important concepts in Self Calibration is the *Absolute Conic* and its projection onto images. The simplest way to represent the Absolute Conic is with the Absolute Dual Quadric (Ω^*). In the Metric case [13] shows that Ω^* is an invariant entity which is equal to $diag(1, 1, 1, 0)$. For a Projective Reconstruction Ω^* can be represented as a 4x4 rank 3 symmetric positive definite matrix. The argument presented is that a transformation that converts Ω^* in a Projective Frame to $diag(1, 1, 1, 0)$, will convert the scene from a Projective Reconstruction to a Metric Reconstruction.

The projection of the Dual Absolute Quadric in the image is described by the equation:

$$\omega^* \propto P\Omega^* P^\intercal$$

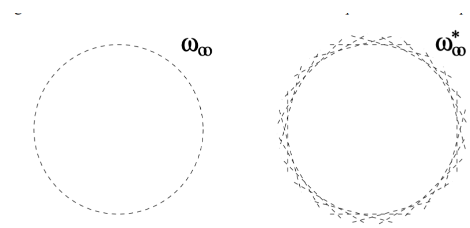

Figure 5. An example of the Image of the Absolute Conic ω and the Dual Image of the Absolute Conic ω^*

Additionally in the Euclidean coordinate frame [13] makes note that it is easy to verify that:

$$\omega^* \propto KK^\intercal$$

These result of these two relationships is that one can directly define constraints on Ω^* given some knowledge of the camera intrinsics.

$$\omega^* \propto KK^\intercal = \begin{bmatrix} f_x^2 + s^2 + u^2 & sf_y + c_x & c_x \\ sf_y + c_x c_y & f_y^2 + c_y^2 & c_y \\ c_x & c_y & 1 \end{bmatrix}$$

If *a priori* knowledge about the camera parameters exists, such as skew (s) is 0 and the principal points (c_x, c_y) are 0 then this relationship can be simplified to:

$$\omega^* \propto KK^\intercal = \begin{bmatrix} f_x^2 & 0 & 0 \\ 0 & f_y^2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

These assumptions are common in practice and there exists a simple normalization step proposed in [13] that will leave the focal lengths at unity and set the principal points to zero. Though it is important to mention that this normalization will not work if an image has been cropped or altered before consumption.

$$P_N = K_N^{-1} P \text{ with } K_N = \begin{bmatrix} w+h & 0 & \frac{w}{2} \\ 0 & w+h & \frac{h}{2} \\ 0 & 0 & 1 \end{bmatrix}$$

If these conditions are met, some interesting independent linear relationships can be found:

$$\begin{aligned} P_1 \Omega^* P_2^\intercal &= 0 \\ P_1 \Omega^* P_3^\intercal &= 0 \\ P_2 \Omega^* P_3^\intercal &= 0 \end{aligned}$$

Additionally, if it is known that $f_x = f_y$ one more constraint can be posed:

$$P_1\Omega^* P_1^\intercal - P_2\Omega^* P_2^\intercal = 0$$

Since Ω^* is a symmetric $4x4$ matrix it can be represented by 10 parameters:

$$\Omega^* = \begin{bmatrix} a & b & c & d \\ b & e & f & g \\ c & f & h & i \\ d & g & i & j \end{bmatrix}$$

Depending on the *a priori* knowledge one might have about K the Dual Absolute Quadric can be found with only 3 views, if the focal lengths are equal. If one considers only the assumption that there is no skew present, then the Quadric can be solved with constraints from 4 different views.

Something to note is that this method can fail for some common configurations if all views lie within a specific degenerate configuration (pure rotations for example). A more rigorous analysis can be found here [11] but some key configurations are shown in 6

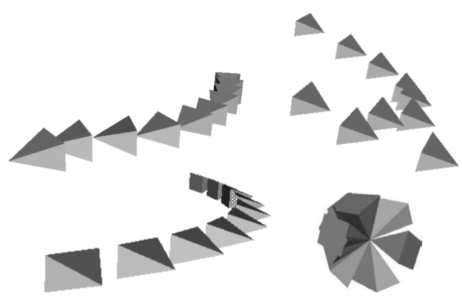

Figure 6. Examples of common motion sequences that would yield ambiguous results that cannot be used for Self Calibration. These include pure translation, pure rotation, orbital motions, and planar motions

Once Ω^*, and by extension ω, is known. The camera intrinsic matrix K can be found through Choleski Decomposition of $\omega = KK^\intercal$. Additionally, if a non refined reconstruction is desired (i.e a starting point for Bundle Adjustment) then an rectifying homography can be computed in order to straighten a projective reconstruction and functionally calibrate the world. In [15] it is shown that this transformation has the form: $T = E\Lambda^{\frac{1}{2}}$ where E are the eigen vectors of Ω^* and Λ is a diagonal matrix of the eigen values of Ω^* with the smallest value set to 1. With this rectifying transformation the Projective Reconstruction can be upgraded to a Metric one.

$$X_{metric} = T^{-1} X_{projective}$$
$$P_{metric} = P_{projective} T$$

With this Metric Reconstruction at hand, one can use methods like RQ decomposition to extract K in this setting as well.

It is important to note that this initial reconstruction should be further refined through non-linear refinement methods such as Levenberg-Marquardt (i.e Bundle Adjustment) to obtain better results, this work currently does no refinement on this initial reconstruction, though it is important to note that with ground truth data one can get good results as well.

4. Quantitative and Qualitative Results

The most important parameter this work wanted to explore was the accuracy of focal length retrieval using this self calibration method. However, the following results loosely apply towards other parameters such as s, c_x, and c_y. The reason these are not explored in the same depth as focal distance is due in part to the normalization proposed in 3.7 which implies that the principal point lies in the center of the image and is a highly likely scenario in most digital cameras. Additionally assuming that skew is zero is also a very prevalent and highly likely in practice.

As can be seen in Figures 7 and 8 when the Absolute Quadric can be found, the method works with a low relative error. For this analysis error is defined as:

$$error = \frac{|f_c - f_{gt}|}{f_{gt}}$$

where f_c is the computed focal length and f_{gt} is its ground truth value.

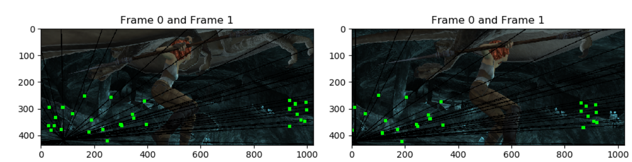

Figure 7. Example Fundamental Matrix Calculation for the Sintel Sequence [2]. For this sequence, my implementation achieved an error of 0.612% for the f_x value and an error of 0.5077% for the f_y value. This one used 4 consecutive frames

However, it is important to note that not all configurations are well suited for this method. Given that these optical flow data sets have non static components present, it is difficult to discern whether a sequence is degenerate or if there is an issue with the image to image correspondences. Therefore only a rough qualitative analysis can be shown for these data sets. Indeed, while some sequences worked for both data sets, others did not.

In order to quantify how well this method works, 2 simple synthetic environments were created. The first was a

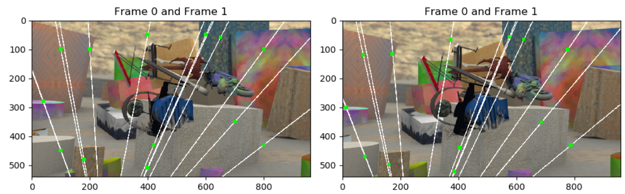

Figure 8. Example Fundamental Matrix Calculation for a sequence in the FlyingThings dataset [10]. For this sequence, my implementation achieved an error of 0.0338% for the f_x value and an error of 0.0303% for the f_y value. This one used 5 consecutive frames

static camera imaging a random point cloud at various poses and translations (see 11), since motion is relative this would be the same as a camera imaging a static scene at different locations. The point cloud poses were generated using the matrix exponential, and a random translation was drawn from a Gaussian distribution. By laying out the scene this way, there was a low probability of landing at a critical configuration as explained in 6. In order to measure robustness to noise, I carried out an experiment where I generated 100 trials for 4 and later 10 views with Gaussian noise at varying standard deviations (now on denoted σ), a visual can be found in Figures 12 and 10.

The second experiment I ran was for a known configuration that could be modified using a parametric surface definition. While the configuration shown in 9 is a degenerate one at first glance, it was modified slightly by perturbing the camera poses a small amount via the Matrix Exponential so the principal axes of the cameras wouldn't all intersect at the same point. This changed the parametric dome configuration from an orbital one to a more general, non-degenerate one.

An interesting result from these two experiments is that this method is effective at deriving the camera intrinsic parameters for general configurations of poses. However, in a fully unstructured case the experiments show that this system is more sensitive to noise. Indeed in 12, error rises higher than 10% at a σ of 0.2 and steadily grows as the σ values trend towards 1. Interestingly, in the structured environment the method proves to be more robust to the noise it receives as can be seen in 10. This is likely due to the fact that the environment is a lot more structured than in the previous experiment.

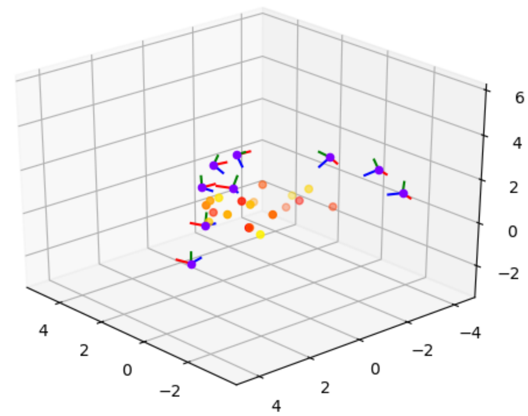

Figure 9. Structured Environment aiming to be more realistic than the static camera environment

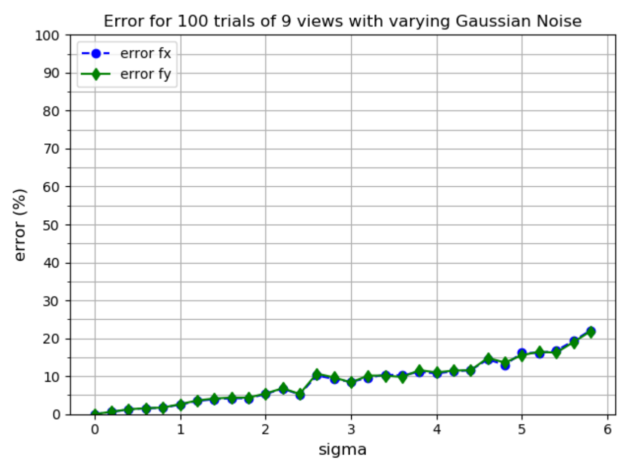

Figure 10. Average Error of the Structured Environment Scenario for 100 trials

5. Strengths and Weaknesses

The framework for self calibration presented in [12] is a robust method for finding the Absolute Conic and using it as a virtual calibration object in the scene to upgrade Projective Reconstructions to metric. It benefits from the fact that only a Projective Reconstruction of a non-degenerate configuration is needed compared to previous works that require strict epipolar constraints on the images. As was shown in the previous section, this method is robust when noise in image matches is low, though this does not hold when noise increases.

There are many points of failure in for this method. The very first one is found when generating image correspondences, if good matches cannot be found then the error will

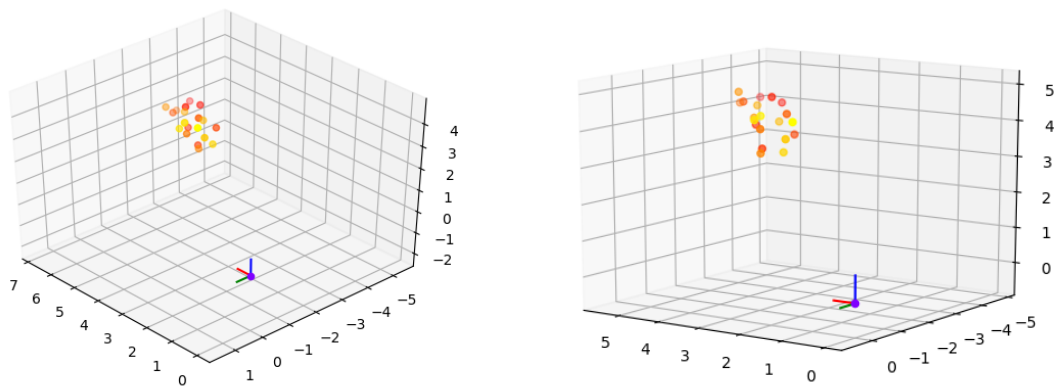

Figure 11. Static Camera Environment

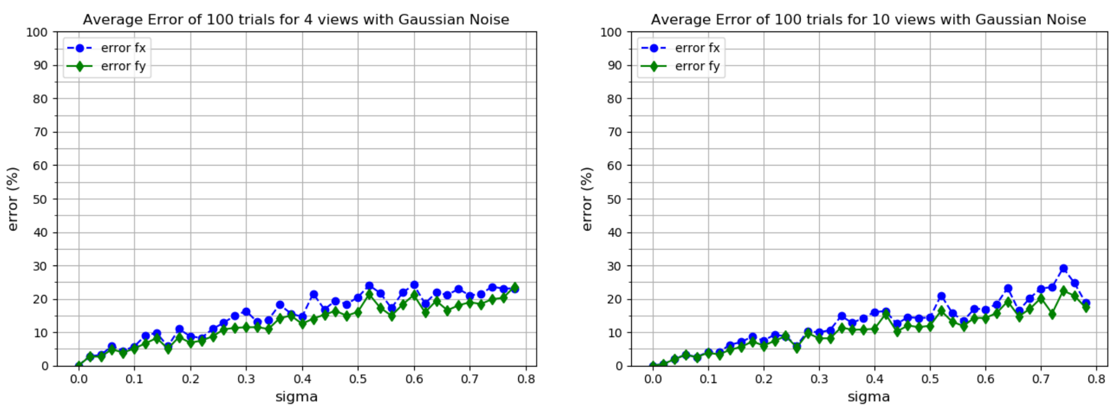

Figure 12. Average Error measured for 100 trials with different levels of Gaussian noise (bottom axis)

be too great to self calibrate. The next point of failure comes when generating the Fundamental Matrix between the two initial views, if any degeneracies are present that lead to an ambiguous solution then a Projective Reconstruction is not possible. Such configurations are thoroughly explained in [14] but a common degeneracy is due to images of co-planar points. The next point of failure is when deriving the camera matrices for new views, if not enough points are imaged in a view, it will fail to provide a solution. At which point the number of usable views will decrease. If we have less than 3 views then self calibration will not be possible. The last point of failure comes in the form of degenerate geometrical configurations, if the cameras that imaged the scene lie completely in any degenerate configuration groups, then self calibration will not be possible.

6. Conclusion

This is an interesting and arguably robust method for self calibration and a good initial starting point for non-linear optimization methods such as Bundle Adjustment. This method is useful when reconstructing environments imaged by cameras whose intrinsics cannot be found through other means. Special care needs to be taken, however, to ensure that no degeneracies that could cause this pipeline to fail exist. For some workflows, it can make more sense to use cameras calibrated ahead of time through some of the more controlled methods that were mentioned at the start of this report. In the event that pre-calibrated cameras are not able to be used, this is a good initial position to start from in a SfM problem and can help such methods converge faster and more accurately.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building rome in a day. *Commun. ACM*, 54(10):105–112, Oct. 2011.

[2] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Com-*

puter Vision (ECCV), Part IV, LNCS 7577, pages 611–625. Springer-Verlag, Oct. 2012.

[3] Manmohan Chandraker, Sameer Agarwal, David Kriegman, and Serge Belongie. Globally optimal algorithms for stratified autocalibration. *International Journal of Computer Vision*, 90:236–254, 11 2010.

[4] M. Devy, V. Garric, and J. J. Orteu. Camera calibration from multiple views of a 2d object, using a global nonlinear minimization method. In *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97*, volume 3, pages 1583–1589 vol.3, 1997.

[5] Gallego, Guillermo, Mueggler, Elias, Sturm, and Peter. Translation of "zur ermittlung eines objektes aus zwei perspektiven mit innerer orientierung" by erwin kruppa (1913), Dec 2017.

[6] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

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

[8] E. E. Hemayed. A survey of camera self-calibration. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, 2003.*, pages 351–357, 2003.

[9] Quang-Tuan Luong and Olivier D. Faugeras. *Self-Calibration of a Stereo Rig from Unknown Camera Motions and Point Correspondences*, pages 195–229. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[10] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. arXiv:1512.02134.

[11] Marc Pollefeys. Self-calibration and metric 3d reconstruction from uncalibrated image sequences. 01 1999.

[12] Marc Pollefeys, Reinhard Koch, and Luc Van Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 32:7–25, 08 1999.

[13] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59:207–232, 09 2004.

[14] P.H.S Torr, A Zisserman, and S.J Maybank. Robust detection of degenerate configurations while estimating the fundamental matrix. *Comput. Vis. Image Underst.*, 71(3):312–333, Sept. 1998.

[15] B. Triggs. Autocalibration and the absolute quadric. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, CVPR '97, page 609, USA, 1997. IEEE Computer Society.