Ambiguities in Camera Self-Calibration

Jun-Cheng Chen
University of Maryland College Park
{pullpull}@umiacs.umd.edu

1. Introduction

Structure from motion (SfM) is the problem of computing the 3D scene and camera parameters from a video or collection of images. SfM can be further classified as calibrated and un-calibrated. In calibrated SfM, the internal camera parameters are known. This is a much easier problem than the un-calibrated case, where these parameters are unknown. Solving for the internal camera parameters are known as the camera self/auto calibration problem.

However, there are sequences/videos that their internal parameters cannot be uniquely determined. (i.e. that is, there are many different settings of internal parameters that give rise to the same video.) In the following sections, we are going to discuss and prove that three cases of motions, (1) pure translation, (2) single rotation, and (3) single rotation about X/Y/Z-axis and translation, are CMS, and the necessary and sufficient conditions of a sequence not being a CMS.

2. Critical Motion Sequence

For the proof, we assume we are given *m* cameras and *n* points. Camera parameters are represented as $f, (t_x^j, t_y^j, t_z^j), j = 1,2,\ldots,m$, 3-D points as $(X_i, Y_i, Z_i), i = 1,2,...,n$, and image of *i*-th point in *j*-th camera as (u_i^j, v_i^j).

2.1. Motions which are CMS

(1) Pure translation

For the pure translation, we assume the image formation equations as

$$u_i^j = \frac{f(X_i - t_x^j)}{(Z_i - t_z^j)} \quad v_i^j = \frac{f(Y_i - t_y^j)}{(Z_i - t_z^j)}$$

Let $f' = sf;\ Z_i' = sZ_i;\ t_z^j{}' = st_z^j$. We can adjust *s* to have different parameter configurations but still result in the same (u_i^j, v_i^j). Hence, we show pure translation is CMS.

(2) In-plane rotation and single rotation about X or Y-axis

Consider the in-plane rotation case, given the rotation matrix and equations as

$$\mathbf{R}^{\mathbf{j}} = \begin{bmatrix} r_{11}^j & r_{12}^j & 0 \\ r_{21}^j & r_{22}^j & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$u_i^j = \frac{f(r_{11}^j X_i + r_{12}^j Y_i)}{Z_i} \quad v_i^j = \frac{f(r_{21}^j X_i + r_{22}^j Y_i)}{Z_i}$$

We can have the same conclusion as in pure translation case by letting $f' = sf;\ Z_i' = sZ_i$.

Next, let's consider other single rotations. For the case of **rotation along X-axis, g**iven the rotation matrix and the corresponding projection

$$R^j = \begin{bmatrix} 1 & 0 & 0 \\ 0 & r_{22}^j & r_{23}^j \\ 0 & r_{32}^j & r_{33}^j \end{bmatrix}$$

$$u_i^j = \frac{f_x(X_i)}{r_{32}^j Y_i + r_{33}^j Z_i} \qquad v_i^j = \frac{f_y(r_{22}^j Y_i + r_{23}^j Z_i)}{r_{32}^j Y_i + r_{33}^j Z_i}$$

Similarly, we can let $f_x' = sf_x;\ X_i' = X_i / s$ that will give the same (u_i^j, v_i^j). In addition, it is also true for the case of **rotation about Y-axis**. Thus, all of three single rotations are CMS.

However, rotation about *X* or *Y*-axis is less critical as compared to that of in-plane rotation and translation case in the sense that such sequences are not critical if we know the aspect ratio between f_x and f_y.

(3) Single rotation about X/Y/Z-axis and translation

In this case, the projection equations can be written as

$$u_i^j = \frac{f_x(X_i - t_x^j)}{r_{32}^j Y_i + r_{33}^j Z_i - t_y^j}$$

$$v_i^j = \frac{f_y(r_{22}^j Y_i + r_{23}^j Z_i - t_y^j)}{r_{32}^j Y_i + r_{33}^j Z_i - t_z^j}$$

Let $f_x' = sf_x;\quad t_x^j{}' = t_x^j / s;\quad X_i' = X_i / s$, we will have infinite ambiguous solutions of internal parameters simply by adjusting *s* as in previous cases. In fact, translation plays no role in reducing calibration ambiguities (proof later).

2.2. Which motions are not CMS?

We analyze the most general case and so we use the full model for the internal calibration given by the matrix K. To facilitate the derivation, here we assume the image formation equation in homogenous coordinate as

$$\tilde{\mathbf{x}}_i^j = \mathbf{K}[\mathbf{R}^j \ \ \mathbf{t}^j]\tilde{\mathbf{X}}_i,$$

$$where\ \tilde{\mathbf{x}}_i^j = \begin{bmatrix} \mathbf{u}_i^j \\ \mathbf{v}_\mathbf{i}^j \\ 1 \end{bmatrix}, \tilde{\mathbf{X}}_i = \begin{bmatrix} \mathbf{X}_i \\ 1 \end{bmatrix}$$

Then,

$$\tilde{\mathbf{x}}_i^j = \mathbf{KR}^j \mathbf{X_i} + \mathbf{Kt}^j$$

Our approach is to find if there exists more than one set of parameters that gave rise to the same video. We know a sequence is CMS if there is another solution, that is,

$$\tilde{\mathbf{x}}^\mathbf{j} = \mathbf{K'}[\mathbf{R}^{\mathbf{j}'} \quad \mathbf{t'}]\tilde{\mathbf{X}}_\mathbf{i}'$$

Then, assume ***K = I***,

$$\begin{aligned} \tilde{\mathbf{x}}_\mathbf{i}^\mathbf{j} &= \mathbf{R}^\mathbf{j}\mathbf{X_i} + \mathbf{t}^\mathbf{j} \\ &= \mathbf{AA}^{-1}\mathbf{R}^\mathbf{j}\mathbf{AA}^{-1}\mathbf{X_i} + \mathbf{AA}^{-1}\mathbf{t}^\mathbf{j} \\ &= \mathbf{A}[\mathbf{A}^{-1}\mathbf{R}^\mathbf{j}\mathbf{A} \quad \mathbf{A}^{-1}\mathbf{t}^\mathbf{j}]\begin{bmatrix} \mathbf{A}^{-1}\mathbf{Xi} \\ \mathbf{1} \end{bmatrix} \end{aligned}$$

We found that it is CMS if and only if $\boldsymbol{A^{-1}R^jA}$ is a rotation matrix. We take ***K=I***, $\boldsymbol{R^j}$ and $\boldsymbol{t^j}$ as the ground-truth parameters and then investigate if there exists other sets of parameters which give rise to the same video. We propose trying ***K=A*** as a guess. With this guess we obtain new $\boldsymbol{R^{j}}$**'**, $\boldsymbol{t^{j}}$**'** and $\boldsymbol{X_i}$**'**. To be a valid set of parameters, the only constraint that this new set should satisfy is that $\boldsymbol{R^{j}}$*'* be a rotation matrix. Hence rotation is the only parameter that determines whether or not a sequence is a CMS. Translation does not play any role. Finally, a sequence is a CMS if and only if $\boldsymbol{R^{j}}'$ is a rotation matrix.

If we assume $\boldsymbol{A^{-1}R^{j}A}$ to be a rotation matrix, then by a variable substitution $\boldsymbol{X=AA^{T}}$, we get $\boldsymbol{R^{j}XR^{jT}=X}$ where $\boldsymbol{X}$ is a symmetric matrix. A sequence is a CMS if and only if there exists a symmetric matrix $\boldsymbol{X}$, which is not the identity matrix, such that it satisfies the equation $\boldsymbol{R^{j}XR^{jT}=X}$ for all $\boldsymbol{R^{j}}$s (note that $\boldsymbol{R^{j}}$ is the rotation matrix corresponding to the j-th camera). For a particular j, the solution of the equation is given in terms of the eigen-vectors of $\boldsymbol{R^{j}}$. Since the desired $\boldsymbol{X}$ should satisfy the equation for all the cameras, it must be in the intersection of all the $\boldsymbol{X^{j}}$s (the individual solution for each j). Therefore, if **X=I** is the only solution then the sequence is not a CMS and otherwise it is.

2.3. Necessary and Sufficient Condition of a sequence not being a CMS

The eigen-vectors corresponding to two rotations with independent rotation axes are different. This implies that the intersection of $\boldsymbol{X^{j}}$s is the identity matrix and this implies that this is not a CMS. If the rotations $\boldsymbol{R^{j}}$s share a common axis of rotation, then they also share the eigen-vectors and so there exist a symmetric matrix $\boldsymbol{X}$, other than identity, which satisfy the equations $\boldsymbol{R^{jT}XR^{j}=X}$ for all j. And hence this is a CMS.

According to the above analysis, we can conclude the necessary and sufficient condition for a sequence not being a CMS is least two independent rotation axes [3], and one axis of rotation is a CMS. The same necessary and sufficient condition holds for the planar scene. For example, Checker-board pattern is a planar scene and calibration is successful if there are enough rotations.

2.4. How to detect CMS and to mitigate the problem?

As one approach, we propose to run a SfM algorithm such as EKF to obtain a solution. We can then analyze this solution to see if it is one of the CMSs. If so we can conclude that the original motion is also CMS. This is based on the fact that if a motion is a CMS, then SfM algorithms like EKF will provide one of the solutions from amongst the many possible. Once we detect that the motion is a CMS, we know which parameters we can trust and which we cannot.

To mitigate the problem, there are two ways that firstly, we can add initial motion with rotation along 2 different axes when capturing the video. In addition, we can also include the GPS information which can fix the camera translation and increase the chance of resolving the ambiguity.

3. CMS Verification

For proof of concept, we use the auto-calibrator implementation proposed in [1]. The reason why to use online parameter estimation algorithm, EKF, is that video is a sequential data, so we can use the previous state to estimate the next one. Moreover, EKF can handle certain extent of

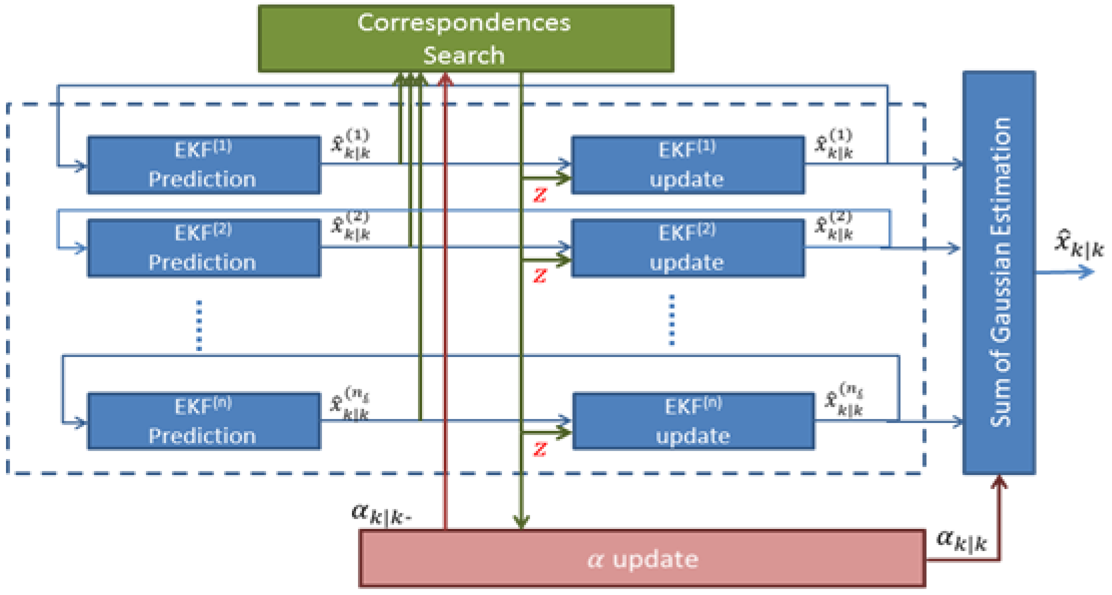

Figure 1. The concept of sum of Gaussian Framework is to run a set of filters simultaneously, and each filter is assumed to be normally distributed. The combination weight for each filter can be decided based on the difference (i.e. it is re-projection error in our case.) between measurement and the estimate, $h(x_k)$. The final state estimate is the weighted average of the state estimates of each filter. In addition, the weighted aggregation of predicted estimate for the feature points can also be used to define the search region in the following video frame which is useful for tracking.

non-linearity.

In the next section, we briefly review EKF and then describe how Javier's EKF calibrator works and how to incorporate GPS information.

3.1. Extended Kalman Filter

In the formulation of Kalman filter, a dynamical system can be modeled using state and measurement equations. (i.e. Both equations include noise which is normally distributed.)

$$
\begin{aligned}
x_{k+1} &= f_k(x_k) + w_k \\
z_k &= h_k(x_k) + v_k
\end{aligned}
$$

where $f_k(\cdot)$, $h_k(\cdot)$ are the state transition and measurement functions respectively, and $w_k \sim N(0, Q_k)$, and $v_k \sim N(0, R_k)$, and x_k is the state vector, and z_k is the measurement from the video.

The exteneded Kalman filter (EKF) is the nonlinear version of Kalman filter, where $f_k(\cdot)$, $h_k(\cdot)$ are assumed nonlinear. However, EKF does linearize these two functions about the current estimate, $\hat{x}_{k|k}$, using first-order Taylor expansion. Then, we can still follow standard Kalman filter paradigm to compute the new estimates.

The state and measurement equation for EKF are

$$x_{k+1} = F_k x_k + u_k + (f(\hat{x}_{k|k}) - F_k \hat{x}_{k|k-1}))$$
$$z_k = H_k x_k + v_k + (h_k(\hat{x}_{k|k-1}) - H_k \hat{x}_{k|k-1}))$$

where $F_k = \frac{\partial f}{\partial x}\Big|_{\hat{x}_{k|k}}$ and $H_k = \frac{\partial h}{\partial x}\Big|_{\hat{x}_{k|k-1}}$.

Besides, last terms for both equations are known values.

.3.2. Javier's Auto-calibrator

The system framework and explanations of the auto-calibrator are shown in Figure 1. More details about the implementation can be found in [1][2] for interested readers.

The state vector in Javier's work consists of two parts. One is camera parameter, including internal and external parameters, and the other is 3D coordinate of each feature point. (For this work, it assumes $f_x = f_y$).

$$x = (x_{cam},\ x_{map}),$$

where $x_{map} = (y_1,\ y_2,\ \dots, y_{n_f}),\quad y_i = (X_i,\ Y_i,\ Z_i)$, the coordinate of ith feature point.

For x_{cam}, the internal parameters include (1) focal length, (2) principal point, and (3) two distortion parameters.

$$x_{cam} = (x_{int},\ x_{ext})$$
$$x_{int} = (f,\ C_x,\ C_y, \kappa_1,\ \kappa_2)$$

The external parameters include (1) camera position, (2) camera orientation (represented in quaternion), (3) movement velocity, and (4) angular velocity.

$$x_{ext} = (r_{cam},\ q_{cam},\ v_{cam},\ \omega_{cam})$$

In this work, it assumes a single camera is used and is without changing focal length, so the internal parameters will be the same for the whole video sequence. Thus, we can simply set the corresponding part of F_k for the internal parameters, x_{int}, as identity matrix for the part of internal parameter and noise in state equation to zero (i.e. or close to zero).

For external part, x_{ext}, we have to compute the Jacobian matrix based on the constant velocity model as shown in the following matrix.

$$\begin{bmatrix} r_{cam}^{k+1} \\ q_{cam}^{k+1} \\ v_{cam}^{k+1} \\ \omega_{cam}^{k+1} \end{bmatrix} = \begin{bmatrix} r_{cam}^{k} + v_{cam}^{k} \times \Delta t \\ q_{cam}^{k} \times q(\omega_{cam}^{k} \times \Delta t) \\ v_{cam}^{k} \\ \omega_{cam}^{k} \end{bmatrix}$$

Similarly, the 3D coordinate of feature points will also be the same during the entire video and we set F_k for them as identity matrix,
For H_k, we compute the H_k using pin-hole camera model.

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} \cong \begin{bmatrix} f & 0 & C_x \\ 0 & f & C_y \\ 0 & 0 & 1 \end{bmatrix} [R|-Rt] \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix},$$
$$where\ R \in R^{3\times3}\ and\ t \in R^{3\times1}$$

3.3. Experimental results

We test five sequences, (1) MITRE sequence with single rotation along Y-axis, (2)(3) office scene I and office scene II both with at least two independent axis rotations, (4) office scene III with pure translation, and (5) checker board with pure in-plane rotation. All videos are in the resolution 320x240. The experimental results and explanation can be found in Table I.

Focal Length	MITRE Sequence (320×240)	Javier's Sequence 1 (320×240)	Javier's Sequence 2 (320×240)	Pure Translation (320×212)	In-plane Rotation (320×212)
					
Ground Truth	0.613148	0.6065625	0.6153125	0.77238	1.028125
Estimated Results	0.603495 (105 frames)	0.61113 (104 frames)	0.604219 (164 frames)	0.6375 (521 frames)	0.680449 (369 frames)

Table 1. For the experimental results of the first three sequences, we show comparable results to the ground truth in PTAM coordinate [8] (i.e. the conversion is done by dividing the estimate focal length in pixels with the max(width, height)) using ~100 frames because they are non-CMS based on the analysis discussed in Section 2. (In Javier's implementation, because both focal lengths, f_x and f_y, are assumed equal, even though the video only have one rotation motion along single axis, it is not CMS as described in Section 2.1.) For the last two CMS sequence (pure translation and in-plane rotation), there exists infinite possible solutions satisfied the equations. For these sequences, the estimated results are far from the ground truth.

4. Extension: Javier's EKF with GPS

To perform coordinate transformation (i.e. given a point in the A coordinate system, we want to get its corresponding coordinate in B), we need to know rotation, translation, and the scale between these two coordinate systems.

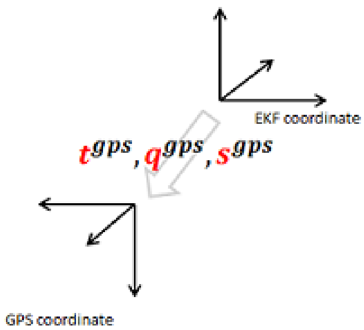

Figure 2. Coordinate transformation requires three parameters for alignment, translation, rotation, and scale.

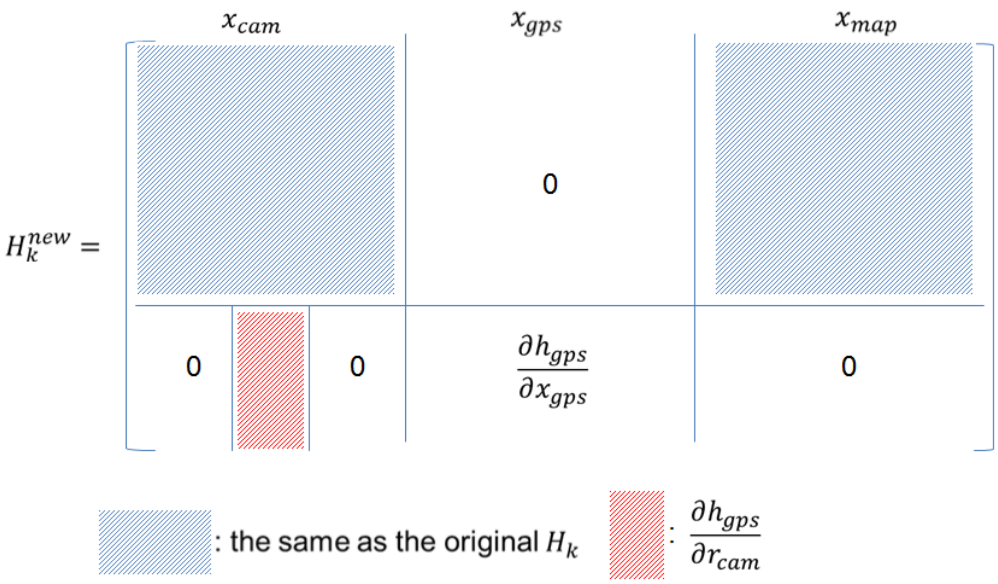

Figure 3. it shows the layout of new H_k. The blue shaded area is the Jacobian matrix of image projection equation with respect to $x = (x_{cam}, x_{map})$. For the block in-between two blue areas, because GPS measurement does not appear in image projection equation, its Jacobian is zero. For the GPS part, given the transformation equation, h_{gps} (i.e. it transforms the camera location in EKF coordinate to GPS coordinate), we take its Jacobian and put into the new H_k. Because the camera position, r_{cam}, gets involved in the h_{gps}, its Jacobian is not zero with respect to other state parameters and shown in red.

Therefore, to incorporate the GPS information into EKF, we have to put those parameters, translation, rotation, and scale into the state vector.

$$x_{new} = (x_{cam}\, ,\ x_{gps}\, ,\ x_{map}),$$

where $x_{gps} = (t_{gps},\ q_{gps},\ s_{gps})$ is the translation from EKF to GPS coordinates. $q_{gps} \in R^4$ is the quaternion from EKF to GPS coordinates. $s_{gps} \in R^1$ is the scale.

Because the registration parameters should be always the same for the two coordinate systems, we can follow the same way that Javier [1] did for the internal parameters and set the F_k of GPS part, x_{gps}, as identity matrix.

For new H_k, we have to compute the Jacobian matrix using the new measurement equation shown as follows and combine it with the old one.

$$z_{gps} = \begin{bmatrix} X_{gps} \\ Y_{gps} \\ Z_{gps} \end{bmatrix} \in R^3$$

$$z_{gps} = h_{gps}(x_{gps}) = s_{gps}\, R(q_{gps})(r_{cam} - t_{gps}) + n_{gps},$$

where $n_{gps} \sim N(0,\ G_k)$.

In Figure 3, it shows the layout of the resulting H_k.

To test the effectiveness of GPS information, we run the calibrator at different starting points at MITRE sequence (the same sequence as shown in Table 1) and fix the number of observations to 50 frames.

MITRE sequence: Ground truth focal length 0.613148		
Starting frame	With GPS	Without GPS
200^{th} frame	0.583242	0.521362
300^{th} frame	0.560840	0.543391
400^{th} frame	0.591406	0.537540

Table 2. it shows the results of EKF auto-calibrator [1] with state vector augmented with GPS information.

From Table 2, we can find that with GPS information, the calibration results converge faster to ground truth. The results also show GPS information helps mitigate the ambiguities on translation and absolute coordinate parameters.

5. Conclusions

To sum up, we prove the necessary and sufficient conditions for a sequence not being a CMS that it should have at least two independent axes of rotations. The rotations can help resolve the ambiguities of the solutions for the camera focal length. In addition, CMS can be detected through analyzing the reconstructed camera trajectory and comparing with the cases discussed in the report. Finally, the problem can be mitigated with initial rotations when capturing the video.

Moreover, we also show how to incorporate GPS information into the auto-calibrator proposed in [1]. To do that, we need to augment the original state vector, x_k, with transformation parameters from EKF coordinate to GPS coordinate and compute new state transition and measurement matrices, F_k and H_k.

6. References

[1]. J. Civera et. al. “Camera Self-Calibration for Sequential Bayesian Structure From Motion”, ICRA, 2009.

[2]. J. Civera et. al. “Inverse depth parametrization for monocular SLAM”, IEEE Transactions on Robotics, 2008.

[3]. Y. Ma, R. Vidal, J. Kosecka, and S. Sastry, “Camera Self-Calibration: Geometry and Algorithm”, UC Berkeley Memorandum No. UCB/ERL, 1999.

[4] Brian D. O. Anderson, John B. Moore, Optimal Filtering, Chap 8.

[5] G. Welch and G. Bishop, “An Introduction to the Kalman Filter”, Technical report 95-041 Department of Computer Science UNC, 2006

[6] Extended Kalman filter, wikipedia, http://en.wikipedia.org/wiki/Extended_Kalman_filter.

[7] M.G. Giebner, “Tightly-Coupled Image-Aided Inertial Navigation System via a Kalman Filter,” Ph.D. thesis of Air Force Institute of Technology, Air University, 2003.

[8] G. Klein and D. Murray, “Parallel Tracking and Mapping for Small AR Workspaces”, ISMAR'07, Nara, 2007.