Bayesian Self-Calibration of a Moving Camera

Gang Qian and Rama Chellappa*

Center for Automation Research
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742-3275, USA
Phone: (301) 405-3656
Fax: (301) 314-9115
{gqian,rama}@cfar.umd.edu
Abstract

In this paper, a Bayesian self-calibration approach using sequential importance sampling (SIS) is proposed. Given a set of feature correspondences tracked through an image sequence, the joint posterior distributions of both camera extrinsic and intrinsic parameters as well as the scene structure are approximated by a set of samples and their corresponding weights. The critical motion sequences are explicitly considered in the design of the algorithm. The probability of the existence of the critical motion sequence is inferred from the sample and weight set obtained from the SIS procedure. No initial guess for the calibration parameters is required. The proposed approach has been extensively tested on both synthetic and real image sequences and satisfactory performance has been observed.

Key words: structure from motion, self-calibration, sequential Monte Carlo methods, video analysis

* Corresponding author
1 Introduction

Automatic retrieval of the intrinsic parameters of a (relatively) moving camera from an observed image sequence has been of great interest to researchers in computer vision since the early 1990s. Subsequent to the pioneering work on camera self-calibration reported by Maybank and Faugeras [1,2], numerous algorithms have been proposed to calibrate cameras with constant (see [3] for a review) or varying intrinsic parameters [4–7].

Although significant efforts have been made to solve the self-calibration problem, several challenges still remain: sensitivity to observation noise, initialization of the algorithms, and processing of critical motion sequences (CMS) [8–11]. The first two challenges are common difficulties arising in nonlinear problems such as camera self-calibration. In some situations, these two factors interact with each other and make the problem more complex. For example, large observation noise might create more local minima (or maxima) and can easily trap iterative optimization methods such as Levenberg-Marquardt and steepest-descent algorithms in local minima, preventing them from converging to the global optimal solution. Hence a good initial guess of the calibration parameters is needed. For recursive algorithms such as the extended Kalman filter [12], a good initial guess of camera parameters is also very crucial for the filter to converge to the true values.

In addition to the difficulties due to noise sensitivity and initialization, the existence of CMSs makes camera self-calibration even more difficult in practice. CMSs are sequences of camera motions that result in inherent ambiguities in camera self-calibration and therefore ambiguities in uncalibrated Euclidean
reconstruction [8]. Any practical self-calibration method must take into account the presence of CMSs. Previous research [9] has shown that ambiguous Euclidean reconstructions from a CMS are conjugated and hypothesis verification can be used to detect and determine the type of CMS. Nevertheless, in the presence of noise, some camera motion sequences which are not CMSs can also result in ambiguous Euclidean reconstructions. Moreover, it has been recently reported in [13] that camera motion sequences “close” to CMSs in the sense of producing ambiguous Euclidean reconstructions can be far away from any type of CMSs in the motion sequence space in the sense of $L_2$ norm. If hypothesis verification is applied to this kind of sequence, it will be classified as one type of CMS; consequently, the true solution, which is actually outside of CMSs, will be lost. Therefore, hypothesis verification is not effective in these circumstances.

In this paper, we focus on the main problem of self-calibration: estimation of the field of view (FOV) with all the other intrinsic parameters known. We will recursively solve the unknown focal length, camera extrinsic motion and scene structure. Recursive estimation of camera motion, structure using the extended Kalman filter was discussed in [14]. The work was later extended in [12], where constant focal length was added as an unknown variable, and improvements on the representation of 3D structures were made to obtain a more stable computational framework and more accurate estimates. However, due to the limitation of extended Kalman filter, a good initial guess is needed. Moreover, estimation and tracking of varying focal length was not considered in [12]. In this paper, we have developed a recursive self-calibration algorithm, capable of processing CMS and yielding reasonable calibration estimates without any specific requirements on initialization. Our algorithms can
also estimate and track the varying FOV throughout the image sequence. The new approach is developed based on the sequential importance sampling (SIS) technique. The SIS procedure has been recently introduced by [15] to estimate the state parameters of a non-linear/non-Gaussian dynamic system. In SIS, the joint posterior distribution of the state parameters given the observations is approximated by a set of samples and their related weights. The SIS procedure has been used for solving the structure from motion (SfM) problem. An SIS-based SfM algorithm has been developed in [16] and it was shown to be robust to feature tracking errors and to be able to handle motion/structure ambiguities. However, in that case, all the intrinsic parameters of the camera are assumed to be given. Here, we use the SIS method to attack the camera self-calibration problem because of its capability of solving problems involving non-linear systems. To reduce the dimensionality of the solution space and make the problem solvable using the Monte Carlo method, we decouple the estimation of the camera motion and scene structure. The camera motion (rotation and translation direction angels and focal length) is first estimated using the epipolar constraint. Then, the translation magnitude and the scene structure are computed from the motion parameters using triangulation.

2 Theoretical Background

2.1 Self-Calibration of a Moving Camera

In many practical situations, the calibration parameters of the camera used to capture the sequences is not available, i.e. the intrinsic parameters of the camera such as the field of view (or the focal length relative to the film size), the
position of the principal point, skew factor and lens distortion are not known beforehand. To reconstruct accurate 3D Euclidean structure and motion, these intrinsic parameters have to be found.

Assume that a perspective projection camera model is considered and that the lens distortion can be ignored or is known. The following calibration matrix is of interest:

\[ A = \begin{bmatrix}
    f k_u & f k_v \cot \theta & u_0 \\
    0 & \frac{f k_v}{\sin \theta} & v_0 \\
    0 & 0 & 1
\end{bmatrix} \]  

(1)

where \( f \) is the focal length of the camera in world coordinate units. \( k_u \) and \( k_v \) are the lengths in pixels of the unit length of the world coordinate system in the vertical and horizontal directions, respectively. \( u_0 \) and \( v_0 \) are the pixel coordinates of the principal point in the image plane. \( \theta \) is the angle between the vertical and horizontal axes in the image plane. Usually it is very close to \( \pi/2 \). Note that by writing the calibration matrix in the above form, we have moved the image plane to the front of the lens and have aligned the coordinate axes in the image plane with those in the world coordinate system.

Assume that a 3D point has projection \( m \) in the image plane. Let \([X, Y, Z]^T\) be its world coordinates and \([u, v]^T\) be the pixel coordinates of its projection. We know that the homogeneous coordinates of any vector \( \mathbf{v} = [v_1, v_2, \ldots, v_n]^T \) is \([v_1, v_2, \ldots, v_n, 1]^T\). Hence in the homogeneous coordinate system, we have the associated representation of the point as \( \mathbf{w} = [X, Y, Z, 1]^T \) and \( \mathbf{m} = [u, v, 1]^T \)
and they are related by a $3 \times 4$ projection matrix $\mathbf{P}$ as

$$\lambda \mathbf{m} = \mathbf{Pw} \quad (2)$$

where $\lambda$ is called projective depth and does not play any role in the location of $m$ in the image plane. Hence (2) is often rewritten as

$$\mathbf{m} \approx \mathbf{Pw} \quad (3)$$

by ignoring $\lambda$, where the symbol $\approx$ means that the two quantities are equal up to a scale factor. The projection matrix $\mathbf{P}$ can be decomposed as

$$\mathbf{P} = \mathbf{A}[\mathbf{R} - \mathbf{t}] \quad (4)$$

where $\mathbf{A}$ is the calibration matrix and $(\mathbf{R}, \mathbf{t})$ is the displacement of the camera, containing both rotation and translation.

The problem of self-calibration is to estimate the calibration matrix $\mathbf{A}$ purely from an observed image sequence without any knowledge or control of the motion of the camera. We will focus on the estimation of an unknown constant or varying FOV when all the other intrinsic parameters are given. We also assume that the camera moves continuously and takes many camera positions.

### 2.2 Critical Motion Sequences

In previous research on solving the camera self-calibration problem, it has been observed that not all camera motion sequences lead to unique camera intrinsic parameters and 3D Euclidean scene reconstruction. Camera motion sequences that produce ambiguous calibrations are called critical motion sequences.
Identification of CMSs with various assumptions on calibration has been systematically investigated in the literature. In [7], Sturm listed all CMSs when the calibration parameters are constant. The CMSs for known calibration parameters, except for a varying FOV, can be found in [11]. Kahl also discussed CMSs when some intrinsic parameters can vary [10]. Other references on identification of CMSs can be found in [17-19]. Recall that we assume that the camera moves continuously and takes many camera positions. According to [10,11], there are three types of CMSs for varying FOV that contain many camera positions:

- arbitrary translation with arbitrary rotation only about the optical axis
- translation along an ellipse or a hyperbola with the optical axis tangent to the ellipse or hyperbola
- translation along the optical axis with arbitrary rotation about the camera centers (at most two)

When the FOV is constant, only the first type of motion in the above CMS list is critical when many camera positions are present [8,11]. When the camera motion is not continuous, there are more critical motion sequences existing for the self-calibration of a camera with only an unknown FOV. Analyzing CMSs related to discontinuous camera motion is beyond the scope this paper.

We mainly deal with self-calibration ambiguities caused by the first kind of CMSs, since this kind of motion sequences are frequently encountered in practice. Because camera motion includes translation and rotation, the assumption of continuous camera motion implies that the rotation of the camera is also continuous if the camera rotates. Since the third type of CMSs contain at most two rotations about the camera centers, it rarely happens to a continuously
rotating camera. Although the second type of CMSs is not explicitly considered in this paper, we have shown by experiments that it is possible to remove the self-calibration ambiguities caused by this type of CMSs if we assume that the 3D scene is rigid and non-planar.

To handle the first type of CMSs, we need to find out the transformations between true and false Euclidean reconstructions. Since the false Euclidean reconstruction is actually a projective reconstruction, it is different from the true Euclidean reconstruction by a projective transformation, $\mathbf{T}_\Phi$ [20].

Assume that a false Euclidean reconstruction has been found. Let $\Delta f$ be the ratio of focal lengths of true and false Euclidean reconstructions in the initial time instant, i.e. $\Delta f = \frac{f_c}{f_\Phi}$ where $f_c$ and $f_\Phi$ are the true and false focal lengths, respectively. Let $\mathbf{t}_\Phi$ and $\mathbf{t}_c$ be the translation vectors associated with false and true reconstructions, respectively. Let $(\alpha_c, \beta_c, \rho_c)$ and $(\alpha_\Phi, \beta_\Phi, \rho_\Phi)$ be the translation direction angles associated with $\mathbf{t}_c$ and $\mathbf{t}_\Phi$. The translation vector is given by $\rho \cdot (\sin(\alpha) \cos(\beta), \sin(\alpha) \sin(\beta), \cos(\alpha))^T$. At any time instant, the true and false projection matrices are related by

$$
\mathbf{P}_c = \mathbf{P}_\Phi \mathbf{T}_\Phi^{-1} = A_\Phi [\mathbf{R} - \mathbf{Rt}_\Phi] \begin{bmatrix}
\Lambda^{-1} & \mathbf{0}_3 \\
\mathbf{0}_3^T & \kappa^{-1}
\end{bmatrix} = A_\Phi [\mathbf{R}\Lambda^{-1} - \kappa^{-1}\mathbf{Rt}_\Phi] \tag{5}
$$

where

$$
\Lambda = \begin{bmatrix}
\Delta f & 0 & 0 \\
0 & \Delta f & 0 \\
0 & 0 & 1
\end{bmatrix} \tag{6}
$$
$$\mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$  \hspace{1cm} (7)$$

and \( \theta \) is the rotation angle about the optical axis. Hence \( \mathbf{R} \Lambda^{-1} = \Lambda^{-1} \mathbf{R} \) and (5) can be written as

$$\mathbf{P}_e = A_{\phi} [\Lambda^{-1} \mathbf{R} - \kappa^{-1} \mathbf{R} t_{\phi}] = A_{\phi} \Lambda^{-1} [\mathbf{R} - \mathbf{R} (\kappa^{-1} \mathbf{R}^T \Lambda \mathbf{R} t_{\phi})]$$ \hspace{1cm} (8)

As a result,

$$t_e = \kappa^{-1} R^T \Lambda \mathbf{R} t_{\phi} = \kappa^{-1} \Lambda t_{\phi}$$ \hspace{1cm} (9)

After some straightforward algebra, we have

$$\rho_e = \rho_{\phi} \sqrt{\Delta f^{-2} \sin \alpha_{\phi}^2 + \cos \alpha_{\phi}^2}$$ \hspace{1cm} (10)$$

$$\alpha_e = \arccos \frac{\Delta f \cos \alpha_{\phi}}{\sqrt{(\Delta f \cos \alpha_{\phi})^2 + \sin^2 \alpha_{\phi}}}$$ \hspace{1cm} (11)$$

$$\beta_e = \beta_{\phi}$$ \hspace{1cm} (12)$$

Furthermore, the resulting rotation angles are same for different ambiguous reconstructions. If the focal length is free to vary, the relationship between true and false focal lengths at any time instant is given by

$$f_e = \Delta f \cdot f_{\phi}$$ \hspace{1cm} (13)$$

Regarding the transformations among the 3D structures, we have the following
relationship:

\[
\mathbf{w}_e = \mathbf{T}_\phi \mathbf{w}_\phi = \begin{bmatrix} \Lambda & \mathbf{0}_3 \\ \mathbf{0}_3^T & \kappa \end{bmatrix} \begin{bmatrix} \mathbf{s}_\phi \\ 1 \end{bmatrix} = \begin{bmatrix} \Lambda \mathbf{s}_\phi \\ \kappa \end{bmatrix}
\]  \hfill (14)

Hence

\[
\mathbf{s}_e = \kappa^{-1} \Lambda \mathbf{s}_\phi
\]  \hfill (15)

When a monocular camera is used, all the length parameters such as translation vectors and depths can be recovered only up to a scale factor. Once a normalization length basis is chosen, \( \kappa \) can be cancelled from the normalized values of the translation vectors and depths. This is why \( \kappa \), the last diagonal element of \( \mathbf{T}_\phi \), does not play any role in this case. Moreover, since the third diagonal element of \( \Lambda \) is 1, it can be seen that the normalized scene depths in the false Euclidean reconstructions are equal to the depth values in the true reconstructions.

The transformations of motion and scene structure reconstructions reveal the relationships among different reconstructions. It will be employed in the design of a novel self-calibration algorithm in the next section.

2.3 Sequential Importance Sampling

The SIS method has been recently proposed for approximating the posterior distribution of the state parameters of a dynamic system [15]. Usually, the state space model of a dynamic system is described by observation and state equations. Denote the measurement by \( \mathbf{y}_t \) and the state parameter by \( \mathbf{x}_t \).
The observation equation essentially provides $f_t(y_t|x_t)$, the conditional distribution of the observation given the state. Similarly, the state equation gives $q_t(x_{t+1}|x_t)$, the Markov transition distribution from time $t$ to time $t+1$. Let $X_t = \{x_i\}_{i=1}^t$ and $Y_t = \{y_i\}_{i=1}^t$. Samples drawn from $\pi_t(X_t) = P(X_t|Y_t)$, the posterior distribution of the states given all the available observations up to $t$, are needed to compute the ensemble statistics such as mean or modes. However, to directly draw samples from a complex, high-dimensional distribution is very difficult in practice. An alternative way to approximate the posterior distribution is by a set of samples called \textit{properly weighted samples} and their corresponding weights [15].

\textbf{Definition [15]} A random variable $X$ drawn from a distribution $g$ is said to be \textbf{properly weighted} by a weighting function $w(X)$ with respect to the distribution $\pi$ if for any integrable function $h$,

$$E_g h(X)w(X) = E_\pi h(X).$$

A set of random draws and weights $(x^{(j)}, w^{(j)}), j = 1, 2, \ldots$, is said to be properly weighted with respect to $\pi$ if

$$\lim_{m \to \infty} \frac{\sum_{j=1}^m h(x^{(j)})w^{(j)}}{\sum_{j=1}^m w^{(j)}} = E_\pi h(X)$$

for any integrable function $h$.

Suppose $\{X_t^{(j)}\}_{j=1}^m$ is a set of random samples properly weighted by the set of weights $\{w_t^{(j)}\}_{j=1}^m$ with respect to $\pi_t$ and let $g_{t+1}$ be a trial distribution. Then the recursive SIS procedure to obtain the samples and weights properly weighting $\pi_{t+1}$ is as follows.
SIS steps: for $j = 1, \ldots, m,$

(A) Draw $X_{t+1} = x_{t+1}^{(j)}$ from $g_{t+1}(x_{t+1}|X_t^{(j)})$. Attach $x_{t+1}^{(j)}$ to form $X_{t+1}^{(j)} = (X_t^{(j)}, x_{t+1}^{(j)}$).

(B) Compute the “incremental weight” $u_{t+1}$ by

$$u_{t+1}^{(j)} = \frac{\pi_t(x_{t+1}^{(j)})}{\pi_t(X_t^{(j)}) g_{t+1}(x_{t+1}^{(j)}|X_t^{(j)})},$$

and let $w_{t+1}^{(j)} = u_{t+1}^{(j)} w_{t}^{(j)}$.

It can be shown [15] that $\{X_{t+1}^{(j)}, w_{t+1}^{(j)}\}_{j=1}^m$ is properly weighted with respect to $\pi_{t+1}$. Hence, the above SIS steps can be recursively applied to get the properly weighted set for any future time instant when the corresponding observations are available. The choice of the trial distribution $g_{t+1}$ is very crucial in the SIS procedure since it directly affects the efficiency of the proposed SIS method. In our approach, we select the Markovian transition probability density function as this trial distribution, i.e.

$$g_{t+1}(x_{t+1}|X_t) = q_{t+1}(x_{t+1}|x_t).$$

It can be shown that in this case $u_{t+1} \propto f(y_{t+1}|x_{t+1})$, which is the conditional probability density function of the observations at $t + 1$ given the state sample $x_{t+1}$ and it is also known as the likelihood function of $x_{t+1}$ since the observations are fixed.

Resampling

In SIS, an additional resampling step [15] often follows the sample weight evaluation after drawing new samples for the current state. Assume that the
sample set $\mathcal{S}_t = \{A^j_t\}_{j=1}^N$ is properly weighted by $\{w^j_t\}_{j=1}^N$. Resampling includes the following two steps [15].

(A) Draw a new sample set $\mathcal{S}_t'$ from $\mathcal{S}_t$ according to the weights $w^{(j)}_t$.

(B) Assign equal weights to all samples in $\mathcal{S}_t'$.

A major benefit of resampling is to statistically reduce bad samples (with small weights) and encourage good samples so that good samples will produce enough number of offspring to describe the distribution of future states. Since resampling will reduce the size of distinct samples, it might be harmful to do resampling when the variation of sample weights is small, i.e. when the samples are more or less equally weighted (important). Resampling is for a better empirical distribution of future states and it does not improve the estimation of current state since it introduces extra Monte Carlo variations in current samples. It is suggested to perform state estimation before resampling [15].

Sample Efficiency

SMC methods are based on importance sampling. The efficiency of an SMC method can be measured by comparing it with direct sampling from the target distribution. Quantitatively, it can be represented by the effective sample size (ESS), which is the size of the samples needed to be drawn from the target distribution to have the equivalent estimation accuracy using the SMC algorithm. Fixing the number of samples used in SMC, a large ESS indicates high efficiency. Although ESS depends on the statistics (functions of the states) to be estimated, it can be approximately computed [21] from sample weights $w$
using

\[
ESS = \frac{m}{1 + m \cdot var(w)}
\]  \hspace{1cm} (16)

where \( m \) is the number of samples. Over-resampling could greatly deteriorate the efficiency of an SMC algorithm and result in the \textit{sample impoverishment} problem, in which case all samples collapse to only a few points in the state space. To avoid over-resampling and sample impoverishment, resampling is not performed in every recursion. Instead, only when the ESS is below a certain threshold, resampling will be revoked.

3 \textbf{Bayesian Self-Calibration Using Sequential Importance Sampling}

In this section, we design a camera self-calibration algorithm assuming that the camera has an unknown constant or varying focal length or equivalently FOV with all the other parameters given. Our goal is to find an algorithm that does not have any specific requirement for initialization and is able to detect and handle the CMSs. The performance of the algorithm should degrade gracefully as the noise level in the observations increases. SIS is used as the main computational framework because of its capability for solving problems involving non-linear systems.

3.1 \textit{System Parameterization}

In the self-calibration problem, we are mainly interested in finding the intrinsic camera calibration parameters. However, due to the camera projection model, the computation of the calibration parameters (here the unknown focal
length) is coupled with that of camera extrinsic motion and scene structure parameters. Since both intrinsic and extrinsic parameters describe the characteristics of the camera, in many cases, they are viewed as unified camera motion parameters. When the camera translation magnitude is not considered, the estimation of the camera motion parameters can be decoupled from that of the structure parameters, by using geometric constraints on rigid body motion such as the epipolar constraint. The scene structure and the translation magnitude can then be easily recovered using triangulation from the motion estimates. This decoupling process in the estimation of motion and structure is also supported by research results from the psychophysics and neuroscience, which have shown that the perception cortical of spatial and object vision of human brains are separated [22] and in a later stage of visual processing, the perceived information from both spatial and object cortical is combined. The spatial and object information in neuroscience correspond to motion and structure, respectively. Moreover, decoupling reduces the dimensionality of the original high-dimensional state space (including motion and structure, the dimension of the latter increasing linearly with the number of feature points) to six or seven. This reduction in dimensionality is extremely preferred by sampling-based computational framework such as the SIS.

*Camera motion parameters* 

Before discussing the parameterization of camera motion, we introduce two 3D Euclidean coordinate systems used in our work. One coordinate system is attached to the camera and uses the center of projection of the camera as its origin. It is denoted by $C$. The $Z$ axis of $C$ is along the optical axis of the camera, with the positive half-axis in the camera looking direction. The $X$-$Y$ plane of $C$ is perpendicular to the $Z$ axis with the $X$ and $Y$ axes parallel to the
Fig. 1. Motion parameters of a moving camera with unknown FOV

borders of the image plane. Also, the $X$-$Y$-$Z$ axes of $C$ satisfy the right-hand rule. The other coordinate system is a world inertial frame, denoted by $I$. $I$ is fixed on the ground. The coordinate axes of $I$ are configured in such a way that initially, $I$ and $C$ coincide. When the camera moves, $C$ travels with the camera and $I$ stays at the initial position. It is worthy to mention that in our approach, the camera motion to be estimated is not between two successive time instants, instead, it is the \textit{global} overall motion of the camera in the world system $I$. By computing the camera motion in $I$, the translation can be accumulated over time and the scene structure can be reliably estimated.

Since the focal length (or FOV) is unknown, $\gamma$ is used to represent the unknown focal length (or FOV). It is noted that the ranges of focal length and FOV are $[0, \infty]$ and $[0, \pi]$, respectively. Since a sampling based procedure is to be used, a naturally bounded variable is preferred. Hence, instead of focal length, the vertical FOV of the camera is to be estimated in the algorithm. Based on the above discussion, the state vector describing both extrinsic (motion) and intrinsic (FOV) parameters could be defined as

$$x = (\psi_x, \psi_y, \psi_z, \alpha, \beta, \gamma)$$

(17)

Here $(\psi_x, \psi_y, \psi_z)$ are the rotation angles of the camera about the coordinate
axes of the inertial frame \( I \) and \( (\alpha, \beta) \) are the elevation and azimuth angles of the camera translation direction, measured in the world system \( I \). The unit vector in the translation direction is given by

\[
\mathbf{T}(\alpha, \beta) = (\sin(\alpha) \cos(\beta), \sin(\alpha) \sin(\beta), \cos(\alpha))^T
\]  

(18)

\( \gamma \) is the FOV of the camera. For simplicity, we still call \( \mathbf{x} \), the motion parameter: remember that the FOV is now included in \( \mathbf{x} \). If the FOV is free to change, one more component is added to the motion parameters.

\[
\mathbf{x} = (\psi_x, \psi_y, \psi_z, \alpha, \beta, \gamma_0, \gamma)
\]  

(19)

where \( \gamma_0 \) represents the FOV of the camera at the initial time instant and \( \gamma \) denotes the FOV at other time instants.

*Camera kinematics and projection models*

Given the above motion parameterization, a dynamic system including both camera kinematics and projection models, is used.

\[
\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{n}_x
\]

(20)

\[
\mathbf{y}_t = \text{Proj}(\mathbf{x}_t, \mathbf{P}) + \mathbf{n}_y
\]

(21)

where \( \mathbf{x}_t \) is the state vector and \( \mathbf{y}_t \) is the observation at time \( t \). \( \mathbf{n}_x \) denotes the dynamic noise in the camera motion, describing the time-varying property of the camera motion. If no prior knowledge about motion is available, a random walk will be a suitable alternative for modeling the camera motion. \( \text{Proj}(\cdot) \) denotes the perspective projection function, describing the projection of feature points with structure \( \mathbf{P} \) in world system \( I \) onto the image plane after camera motion \( \mathbf{x}_t \). It can be interpreted as follows. Suppose the 3D position
of a point $p$ in the world system $I$ is $P = (X, Y, Z)^T$ and its 3D position in current camera-centered system $C$ is $P_t = (X_t, Y_t, Z_t)^T$. Then the projection of $p$ onto the image plane after camera motion $m_t$ is

\[
\begin{align*}
    u &= f \frac{X_t}{Z_t} \\
    v &= f \frac{Y_t}{Z_t}
\end{align*}
\]  

(22)

(23)

where $f$ is the focal length of the camera. If the camera motion parameter at time $t$ is $x_t = (\Psi, \alpha, \beta)$, then

\[
P_t = R(\Psi)(P - \rho T(\alpha, \beta)) \]

(24)

where camera translation vector $T$ is given by (18) and $\rho$ is the translation magnitude. $\Psi = (\psi_x, \psi_y, \psi_z)$ denotes the camera rotational angles and the rotation matrix $R(\Psi)$ can be computed by

\[
R(\Psi) = \begin{pmatrix}
    n_1^2 + (1 - n_1^2) \eta & n_1 n_2 (1 - \eta) + n_3 \zeta & n_1 n_3 (1 - \eta) - n_2 \zeta \\
    n_1 n_2 (1 - \eta) - n_3 \zeta & n_2^2 + (1 - n_2^2) \eta & n_2 n_3 (1 - \eta) + n_1 \zeta \\
    n_1 n_3 (1 - \eta) + n_2 \zeta & n_2 n_3 (1 - \eta) - n_1 \zeta & n_3^2 + (1 - n_3^2) \eta
\end{pmatrix}
\]  

(25)

where $n = (n_1, n_2, n_3)^T = \frac{\Psi}{|\Psi|}$ is the direction cosine vector, $\zeta = \sin |\Psi|$, and $\eta = \cos |\Psi|$.

3.2 Weight Computation of Camera Motion Samples

Based on the above camera kinematics and projection models, we present an SIS method for finding an approximation to the posterior distribution of
the motion parameters. As mentioned above, the trial distribution in the SIS procedure used in our approach is chosen as \( g_{t+1}(x_{t+1}|X_t) = q_{t+1}(x_{t+1}|x_t) \). Therefore, during the SIS step (A), we will draw samples from the distribution of \( x_t + n_x \). The "incremental weight" \( u_{t+1} \) in this case is proportional to the likelihood function of the observation given the motion parameters, \( u_{t+1} \propto f(y_{t+1}|x_{t+1}) \).

To derive the likelihood function, consider the case when only one point is observed. Assume that at the initial time instant, a point \( p \) is projected to \( (u_0, v_0) \) in the first image plane. At time \( t \) after camera motion \( x_t \), feature tracking results indicate that \( p \) is at \( (u_t, v_t) \) in the image plane at current camera position. Due to the feature tracking noise, \( (u_t, v_t) \) is a noisy measurement of the true projection of \( p \). Suppose that the distribution of feature tracking noise is normal with zero mean and covariance matrix given by

\[
\begin{bmatrix}
\sigma_u & 0 \\
0 & \sigma_v
\end{bmatrix}
\]

When the camera only rotates, the likelihood function can be computed directly as

\[
f((u_t, v_t)|x_t) = \frac{1}{2\pi\sigma_u\sigma_v}\exp\left\{-\frac{(u_t - u'(u_0, v_0, \Psi_t))^2}{2\sigma_u^2} + \frac{(v - v'(u_0, v_0, \Psi_t))^2}{2\sigma_v^2}\right\}
\]

(26)

where \( (u', v') \) is the reprojected position of point \( p \) in current image plane after
camera rotation. It can be computed using \((u_0, v_0)\) and camera rotation angles \(\Psi_i\) according to (23). In this case, the scene structure \(P\) is not involved in the computation of \((u', v')\).

When the camera motion includes non-zero translation, the computation of the likelihood \(f((u_t, v_t)|x_t)\) becomes more difficult. (26) can not be used directly since \(P\) will be needed to compute \((u', v')\) when \(T(\alpha, \beta)\) is non-zero. However \(P\) is not represented in the state vector. To overcome this difficulty, in our approach, we applied conditional expectation over the structure to derive the equation for the likelihood function:

\[
    f((u_t, v_t)|x_t) = \int_{P_t} f(y_t|x_t, P)p(P)dP
\]

(27)

where \([P_t, P_u]\) denotes the range of the feature depths such that all the features can be observed at both camera positions with positive feature depth values. However, the computation of the integral in (27) requires knowledge of the prior distributions of the scene structure \(P\) and the translation magnitudes, which are unknown. The epipolar constraint [23] is used here to resolve this difficulty. Recall that the epipolar constraint says that the perspective projections of a 3D point on the two image planes taken from different viewpoints lie on their corresponding epipolar lines, which are the intersections of the two image planes with the epipolar plane containing the 3D point and the two centers of projection (COP) of the camera. Given the image position of a point in one view and the camera motion parameters between the two views, the epipolar line related to this point in the other view can be easily determined. Note that

\[
    E_P \{f(y_t|x_t, P)\} = E_{P_t}\{f(y_t|x_t, P_t)\}
\]
where \( \mathcal{P}_t \) represents the image positions of the feature points at time \( t \). The prior distribution of \( \mathcal{P}_t \) is not available. We assume that \( \mathcal{P}_t \) is uniformly distributed on the pixel sites on the corresponding epipolar line segments. Since no prior knowledge about the ratios between feature depth values and camera translation magnitude is given, the whole epipolar line is under consideration. However only the epipolar line segment that agrees with the positive-depth constraint is used to evaluate the above expectation.

We assume that the feature point positions in the first image frame are exact. Let \( l \) denote the epipolar line segment of \( p \) at \( t \) and let \((x_1, y_1), (x_2, y_2)\) be the two terminal points of \( l \). The locations of the two points are easy to find given the camera motion \( \mathbf{x}_t \) and the image position of \( p \) in the first frame using the positive-depth constraint. Without loss of generality, let us assume that \(|y_2 - y_1| \geq |x_2 - x_1|\) and \( y_2 > y_1 \). Let \( k = \frac{x_2 - x_1}{y_2 - y_1} \) be the slope of the epipolar line and \( x_t(y) = x_1 + k(y - y_1) \) be the function of the epipolar line \( l \). The likelihood function of the motion parameter given a single point observation is given by

\[
\begin{align*}
f((u, v) \mid \mathbf{x}_t) & = \frac{1}{d} \int_{y_1}^{y_2} \frac{1}{2\pi\sigma_u\sigma_v} \exp \left\{ - \frac{(u - x_t(y))^2}{2\sigma_u^2} - \frac{(v - y)^2}{2\sigma_v^2} \right\} \, dy \\
& = \frac{\lambda}{2a} \exp \left\{ \frac{b^2}{2a^2c^2} (erf(f_1) - erf(f_2)) \right\}
\end{align*}
\] (28) (29)
where

\[
\begin{align*}
    f_i &= \frac{y_i a^2 + b}{\sqrt{2 \pi d}}, i = 1, 2 \\
    \lambda &= \frac{1}{\sqrt{2 \pi d}} \exp \left\{ -\left( \frac{(y_2 - y_1)^2 + (x_2 - x_1)^2}{2 \sigma_u^2} + \frac{\sigma_v^2}{2} \right) \right\} \\
    d &= \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}, \text{ the length of the epipolar line segment} \\
    c &= \sigma_u \sigma_v \\
    b &= k(x_1 - u - ky_1) \sigma_v^2 - v \sigma_u^2 \\
    a &= \sqrt{\sigma_v^2 k^2 + \sigma_u^2}
\end{align*}
\]

and \( \text{erf}(\cdot) \) is the standard normal error function.

Given 2D trajectories of a set of feature points, multiple likelihood functions can be applied in different situations. If all the points are “good” features in the sense that they are not on moving objects nor mismatched, then \( f(y_i | x_t) \) is obtained by multiplying the individual likelihood functions of the feature points.

\[
f(y_i | x_t) = \prod_{i=1}^{M} f(y_i^{(i)} | x_t) = \prod_{i=1}^{M} f((u_i^{(i)}, v_i^{(i)}) | x_t)
\]

where \( M \) is the total number of the observed feature points. In some cases, some of the points are known to be not “good” beforehand. If the number of “bad” points is less than half of the total number of tracked feature points, the following equation can be used to compute the likelihood function for the observations:

\[
f(y_i | x_t) = \prod_{i \in \mathcal{G}} f(y_i^{(i)} | x_t)
\]
where $\mathcal{G} = \{ i : f(y_t^{(i)} | x_t) \geq \text{median}(\{f(y_t^{(i)} | x_t)\}_{i=1}^M) \}$.

3.3 Processing Critical and Pseudo-critical Motion Sequences

If a motion sequence is critical, the self-calibration algorithm should be able to detect the presence of CMSs. CMS detection is modeled as a hypothesis testing problem and the posterior probability of the non-criticalness of the motion sequence is estimated. In the hypothesis testing problem, a binary variable $I_c$ is introduced to indicate the presence of a CMS:

$$I_c = \begin{cases} 1, \text{the motion sequence is critical} \\ 0, \text{the motion sequence is not critical} \end{cases}$$

This hypothesis testing problem can be naturally embedded in the SIS procedure. In the current case of interest, only one class of CMS exists: motion sequences that do not contain any rotation about an axis parallel to the image plane. Therefore, if the motion sequence is critical, motion samples with rotation only about the optical axis are enough to interpret the trajectories of the feature points. On the other hand, if the motion sequence is not critical, motion samples with rotation about axes parallel to the image plane have to be used to interpret the feature trajectories. Hence, two sets of samples are involved in the SIS procedure. The set of samples with only rotation about the optical axis is denoted by $\mathcal{X}_C$ since it can explain the feature trajectories in the image plane introduced by the CMS. The other set of samples is denoted by $\mathcal{X}_C'$ because it will be used to explain the feature trajectories caused by general motion sequences other than CMSs. In the initialization stage of SIS, samples are generated in these two sets. Because no knowledge of the criticalness of
the motion sequence is available at the beginning, equal numbers of samples are used in the two sets. The weights of the samples can be computed directly using (29).

During the camera movement, the criticalness of the motion sequence can change. A critical motion sequence up to time \( t \) can become non-critical at time \( t + 1 \) if rotation about axes parallel to the image plane is present at time \( t + 1 \). However, a non-critical motion sequence can never become critical. If the indicator \( I_C \) is viewed as the state of a dynamic system, this dynamic system can be characterized by a Markov chain. If the probability that a critical motion sequence becomes non-critical at time \( t \) is \( P_{C \rightarrow \mathcal{G}}(t) \), the state transition probabilities of the Markov chain are:

\[
\begin{align*}
P(I_C(t + 1) = 0|I_C(t) = 1) &= P_{C \rightarrow \mathcal{G}}(t) \\
P(I_C(t + 1) = 1|I_C(t) = 1) &= 1 - P_{C \rightarrow \mathcal{G}}(t) \\
P(I_C(t + 1) = 0|I_C(t) = 0) &= 1 \\
P(I_C(t + 1) = 1|I_C(t) = 0) &= 0
\end{align*}
\] (34)

To take this fact into account in SIS, when drawing new samples for time \( t + 1 \) from samples for time \( t \), the samples in \( \mathcal{X}_C \) need to be transferred to \( \mathcal{X}_G \) with probability \( P_{C \rightarrow \mathcal{G}}(t) \). This can be done by adding rotation components about axes parallel to the image plane, which can be drawn from a trial distribution. \( P_{C \rightarrow \mathcal{G}}(t) \) is unknown and no knowledge about it is available. Intuitively, 0.5 could be a good value for \( P_{C \rightarrow \mathcal{G}}(t) \) for all \( t \) since it gives the maximum uncertainty to the occurrence of the transformation of the motion sequence from critical to non-critical.
In the SIS procedure, the sample-weight set describes the posterior distribution of the motion parameters:

\[
P(\mathcal{X}_t|\mathcal{Y}_i) = \sum_{I_C} P(\mathcal{X}_t, I_C|\mathcal{Y}_i) \\
= P(\mathcal{X}_t|I_C = 1, \mathcal{Y}_i) P(I_C = 1|\mathcal{Y}_i) + P(\mathcal{X}_t|I_C = 0, \mathcal{Y}_i) P(I_C = 0|\mathcal{Y}_i)
\]

The samples in \( \mathcal{X}_C \) are properly weighted by their corresponding weights with respect to \( P(\mathcal{X}_t|I_C = 0, \mathcal{Y}_i) \), the posterior distribution of the motion parameters conditional on the motion sequence is not critical. The posterior probability of the presence of the critical motion sequence, \( \pi_t(I_C = 1) = P(I_C = 1|\mathcal{Y}_i) \), can be obtained using the following theorem.

**Proposition 1** Assume that \( \{\mathcal{X}_C, \mathcal{X}_G\} \) is properly weighted by \( \{\mathcal{W}_C, \mathcal{W}_G\} \) with respect to \( P(\mathcal{X}_t|\mathcal{Y}_i) \). \( \mathcal{X}_C \) is the sample set related to the hypothesis that a critical motion sequence is present and \( \mathcal{W}_C \) is the associated weight set. Then \( \pi_t(I_C = 1) \), the posterior probability of criticalness of the given motion sequence, is given by

\[
\pi_t(I_C = 1) = \lim_{m \to \infty} \frac{\sum_{w_c \in \mathcal{W}_C} w_c}{\sum_{w_c \in \mathcal{W}_C} w_c + \sum_{w_g \in \mathcal{W}_G} w_g}
\]

where \( m \) is the number of samples.

The proof of the theorem is very straightforward.

### 3.4 The Self-Calibration Algorithm

Based on the discussion in the last section, a Bayesian camera self-calibration algorithm using SIS can be designed. Before the algorithm is presented, one more issue needs to be addressed.
Samples in $\mathcal{X}_c$ are to be transferred to $\mathcal{X}_g$ with a certain probability. Let the sample-weight pairs before this transfer be $\mathcal{S}_c = (\mathcal{X}_c, \mathcal{W}_c)$ and $\mathcal{S}_g = (\mathcal{X}_g, \mathcal{W}_g)$. The transfer is done by reducing the weights of the samples in $\mathcal{S}_c$ to half of the original values such that the number of samples belonging to $\mathcal{S}_c$ after resampling has been decreased by half. Then, the samples of $\mathcal{S}_c$ are put into $\mathcal{S}_g$ with the remaining half weights. Therefore, the new sample-weight sets after the transfer step become

$$\hat{\mathcal{S}}_c = (\mathcal{X}_c, \frac{\mathcal{W}_c}{2}), \quad \hat{\mathcal{S}}_g = (\{\mathcal{X}_c, \mathcal{X}_g\}, \{\frac{\mathcal{W}_c}{2}, \mathcal{W}_g\})$$  \quad (36)

After the transfer of samples, resampling is done to the samples in $\hat{\mathcal{S}}_c$ to prepare samples for the next time instant. For samples in $\hat{\mathcal{S}}_g$, two procedures will be utilized. Resampling is applied to the samples from $\mathcal{S}_g$. A crucial procedure, called uniforming, is applied to the samples transferred from $\mathcal{S}_c$ to $\hat{\mathcal{S}}_g$. As we mentioned in the last section, due to the fact that only a finite number of samples are used to describe the posterior distribution of the parameters, the empirical distribution of the FOV is not uniform. Uniforming is used to explore the fact that the posterior distribution of the FOV should be uniform in $[0, \pi]$ if the motion sequence is critical no matter how the FOV is distributed according to the empirical samples and weights. Let $\mathcal{X}_{c \rightarrow g} = \{x^{(j)}\}_{j=1}^k$ be the samples transferred with weights $\mathcal{W}_{c \rightarrow g} = \{\frac{w^{(j)}}{2}\}_{j=1}^k$ in $\hat{\mathcal{S}}_g$. These sample-weight pairs are denoted by $\mathcal{S}_{c \rightarrow g} = (\mathcal{X}_{c \rightarrow g}, \mathcal{W}_{c \rightarrow g}) = (\mathcal{X}_c, \frac{\mathcal{W}_c}{2})$. Assume that $m$ samples are used in the SIS procedure. Uniforming applied to $\mathcal{S}_{c \rightarrow g}$ can be done in the following way:

**Uniforming Step**

For $j = 1, \cdots, k,$
(A) Uniformly draw \( \Gamma = \{\gamma^{(i)}\}_{i=1}^{n_j} \), samples of FOV from \([0, \pi]\) where

\[
n_j = \frac{m w^{(j)}}{2(\sum_{w_c \in W_c} w_c + \sum_{w_g \in W_g} w_g)}
\]

For \( i = 1, \ldots, n_j \)

(B) Compute the associated focal length in units of the height of the imaging film: \( f^{(i)} = (2 \tan \frac{\gamma^{(i)}}{2})^{-1} \). Motion sample \( x^{(j)} \) is used as a seed to produce more samples. Let the focal length associated with the FOV in \( x^{(j)} \) be \( f^{(0)} \). By using the transformation among ambiguous motion estimates derived in section 2, the camera motion parameters related to the current focal length \( f^{(i)} \) can be found directly using (12). A new motion sample can be formed as

\[
x^{(i)}_j = (0, 0, \Psi_z, \alpha^{(i)}_j, \beta, \gamma^{(i)})
\]

(37)

where \( \Psi_z \) and \( \beta \) are the corresponding components in \( x^{(j)} \). Hence, \( X^{(j)}_d = \{x^{(i)}_j\}_{i=1}^{n_j} \) are the new samples obtained from seed \( x^{(j)} \).

If the focal length is free to vary, the associated sample value of \( \gamma_0 \) needs to be changed properly. The new samples can be written as

\[
x^{(i)}_j = (0, 0, \Psi_z, \alpha^{(i)}_j, \beta, \gamma^{(i)}_0, \gamma^{(i)})
\]

(38)

where

\[
\gamma^{(i)}_0 = 2 \arctan \frac{f^{(0)}_0 f^{(i)}}{f^{(0)} + f^{(i)}} \quad f^{(0)}_0 = \left(2 \tan \frac{\gamma^{(0)}}{2}\right)^{-1}
\]

(39)

and \( \gamma^{(0)}_0 \) is the value of \( \gamma_0 \) (FOV at the initial time instant) in the seed sample \( x^{(i)} \).
(C) $\{X_{ij}^{(k)}\}_{j=1}^{k}$ contains the samples by uniforming the sample-weight pair $\mathcal{S}_{C \rightarrow \varphi}$.

Based on the above discussion, the SIS procedure for Bayesian self-calibration proceeds as follows.

**Bayesian Camera Self-Calibration Using SIS**

1. **Initialization.** Draw samples of the motion parameters $\{x_{0j}^{(j)}\}_{j=1}^{m}$ from the initial distribution $\pi_0$. $\pi_0$ describes the distribution of the motion parameters $x_0$ before the camera moves. The absence of camera motion does not imply that $x_0 = 0$. Although the rotation angle vector $\psi$ and the translational vector are zero, the translational angles can be uniformly distributed. Hence, in $\{x_{0j}^{(j)}\}$, the components of the rotation angles are all set to zero and the samples of $\alpha$, $\beta$ and $\gamma$ (and $\gamma_0$ if the focal length is free to change) are drawn from the uniform distribution in $[0, \pi]$, $[0, 2\pi]$ and $[0, \pi]$, respectively. Since all the samples are drawn from the exact posterior distributions, equal weights are assigned to these samples. Since at the moment, rotation angles are all zeros, all the current samples belong to $\mathcal{X}_C$ and $\mathcal{X}_\varphi$ contains no samples.

   For $t = 1, \cdots, \tau$:

2. **Sample transfer.** Two pairs of sample-weight sets are available: $(\mathcal{X}_C, \mathcal{W}_C)$ and $(\mathcal{X}_\varphi, \mathcal{W}_\varphi)$. Transfer all samples in $\mathcal{X}_C$ to $\mathcal{X}_\varphi$, and assign half weight to each sample. Denote the sample-weight pair transferred from $\mathcal{S}_C$ to $\mathcal{S}_\varphi$ by $\mathcal{S}_{C \rightarrow \varphi} = (\mathcal{X}_C, \frac{\mathcal{W}_C}{2})$. The new sample-weight pairs after sample transfer are $\hat{\mathcal{S}}_C = (\mathcal{X}_C, \frac{\mathcal{W}_C}{2})$ and $\hat{\mathcal{S}}_\varphi = \{\mathcal{S}_{C \rightarrow \varphi}, \mathcal{S}_\varphi\} = (\{\mathcal{X}_C, \mathcal{X}_\varphi\}, \{\frac{\mathcal{W}_C}{2}, \mathcal{W}_\varphi\})$.

3. **Resampling and uniforming.** Resample the samples in $\hat{\mathcal{S}}_C$ according to their associated weights. $\hat{\mathcal{X}}_C$ is used to represent the set containing...
the resulting samples. For samples in $\tilde{\mathcal{S}}_C$, uniforming and resampling are applied to samples belonging to different sets. Uniforming is performed on the samples in $\mathcal{S}_{C-\varphi}$. The samples originally in $\mathcal{X}_C$ are then resampled. The sample set produced by these two procedures is denoted by $\hat{\mathcal{X}}_C$. Since resampling and uniforming have been executed, all the samples in $\hat{\mathcal{X}}_C$ and $\hat{\mathcal{X}}_C$ have equal weights. Let $\{\hat{x}^{(j)}_{i-1}\}_{j=1}^m$ denote the current samples.

4) **Sample generation.**

For $j = 1, \ldots, m$:

Draw $x_i^{(j)}$ from the distributions of $\hat{x}^{j}_{i-1} + n_x$. The following distributions can be used for the dynamic noise variables in the translation direction angles. $n_\kappa \sim U(-\delta_\kappa, \delta_\kappa), \kappa \in \{\alpha, \beta\}$ where $\delta_\alpha$ and $\delta_\beta$ can be chosen as positive numbers. The distributions of the dynamic noise variables in the rotation angles depend on $\hat{x}^{j}_{i-1}$. If $\hat{x}^{j}_{i-1}$ is in $\hat{\mathcal{X}}_C$, disturbances are only added to the $Z$ component of the rotation angles with $n_{\psi_z} \sim \mathcal{N}(0, \sigma_z)$. Otherwise, dynamic disturbances can be added to all three components of the rotation angles and the associated distributions can be $n_{\psi_t} \sim \mathcal{N}(0, \sigma_t), t \in \{x, y, z\}$ where $\delta_x$, $\delta_y$ and $\delta_z$ can also be chosen as some small positive numbers.

5) **Weight computation and re-sampling**

- Compute the weights of the samples, $\{w_i^{(j)}\}$ using (29), (31) or (32), depending on the property of the feature point correspondences. Notice that in this case, the computation of the positions of terminal points of the epipolar line $l$ involves not only the extrinsic parameters of the camera motion, but also $\gamma$, the field of view. The resulting samples and their corresponding weights $(\mathcal{X}_i^{(j)}, w_i^{(j)})$ are properly weighted with respect to $\pi_i(\mathcal{X}_i)$. 

30
• Compute the current ESS using (16). When the ESS is lower than a threshold, \((m/3)\) in our implementation, resample the above samples to amplify important motion samples so that good samples could be found for the motion parameters for the next time instant.

3.5 Inference of Translation Magnitude and Depth Distributions

By using the above SIS procedure for camera motion, the posterior distributions of the camera extrinsic and intrinsic parameters can be approximately described by motion samples and their corresponding weights. The posterior distributions of translation magnitudes and depths of feature points can be obtained using triangulation. In [16], two algorithms have been presented to find the posterior distributions of feature depths when the camera is fully calibrated. Both algorithms can be applied here to obtain \(p(z|\mathcal{Y}_t)\), since nothing has changed except that the unknown FOV \(\gamma\) is included in the motion vector. During the computation, the given constant FOV used in the case of calibrated camera needs to be replaced by FOV samples, from the SIS procedure for camera motion. Note that these algorithms find out the posterior of depth, using all the feature correspondences up to the current time instant. If the posterior distributions using only the instantaneous observation are of interest, a much simpler algorithm will be sufficient. Let \(\mathbf{y}_t\) be the feature correspondences at time \(t\). We are looking for approximations to \(p(\mathbf{x}_t, \rho_t, z|\mathbf{y}_t)\), where \(\mathbf{x}_t\) is camera motion parameters defined as (17) or (19), and \(\rho_t\) is the magnitude of the camera translation, \(z\) the depth of feature points. To remove the global scaling ambiguity, we normalize all the length variables according
to the depth value of one of the feature points. Using the Bayes’ rule, we have

\[ p(x_t, \rho_t, z | y_t) = p(y_t | x_t, \rho_t, z)p(x_t)p(\rho_t)p(z) \] (40)

If we can draw samples of \((x_t, \rho_t, z)\) and compute their likelihood as weights, the joint motion and depth distribution can then be described. However, direct sample from this space is infeasible. Because of the large number of feature points, the dimensionality of the sample space is fairly large and effectively drawing samples from high-dimensional sample space is extremely difficult. To overcome this problem, we utilize the knowledge of the posterior distribution of the motion parameters.

\[ p(x_t) = \frac{p(x_t | y_t)}{p(y_t | x_t)} \] (41)

From the SIS procedure for camera motion, samples of the motion posterior have been obtained. Although these motion samples are with respect to \(p(x_t | y_t)\), their locations in the motion subspace still indicate the peaks of \(p(x_t | y_t)\). Due to the resampling step, one peak of \(p(x_t | y_t)\) might have multiple copies. Samples far apart enough are used as seeds for drawing new motion samples. For each new motion sample \(x^{(j)}\), the related sample of \(\rho^{(j)}\) and \(z^{(j)}\) can be determined using triangulation by solving for the maximum likelihood as

\[ (\rho^{(j)}, z^{(j)}) = \arg \max_{\rho, z} p(y_t | \rho, z, x^{(j)}) \] (42)

The resulting likelihood is then used as the weight for the joint sample \((x^{(j)}, \rho^{(j)}, z^{(j)})\).

To summarize, we have the following sampling procedure for computing the joint motion and structure posterior.
Random Sampling for Joint Motion and Structure Posterior

1) **Motion sample seed generation.** Let \( \{x_{m}^{(j)}\}_{j=1}^{m} \) be the motion sample set obtained using the SIS procedure for motion distribution and it is w.r.t. \( p(x_{i}|y_{i}) \). Put the first sample \( x_{m}^{(1)} \) into the seed set \( S \). For each of the remaining motion sample \( x_{m}^{(j)}, j = 2, \ldots, m \)

(a) Compute the minimum distance from \( x_{m}^{(j)} \) to the seed set.

\[
d_{\text{min}} = \min_{k} \{d_{k} : d_{k} = |x_{m}^{(j)} - x_{s}^{(k)}|, x_{s}^{(k)} \in S\}
\] (43)

(b) Create seeds If \( d_{\text{min}} \) is larger than a prechosen threshold, this motion sample is accepted as a seed, because it is far away from any existing seeds.

For each seed \( x_{s}^{(k)}, k = 1, \ldots, K \)

2) **Motion and structure sample generation.**

Draw \( n \) motion samples around this seed. Let the new motion sample be \( x^{(i)} \) and it can be obtained as

\[
x^{(i)} = x_{s}^{(k)} + n_{x}
\] (44)

where \( n_{x} \) represents kinematics noise variables and it can be chosen to be Gaussian.

For each newly drawn motion sample \( x^{(i)} \), compute the maximum likelihood estimate of the translation magnitude and depth using (42).

3) **Sample weight evaluation.** Compute the likelihood function \( p(y_{i}|x^{(i)}, \rho^{(i)}, z^{(i)}) \) and assign this likelihood function to the weight related with this joint motion/structure sample i.e.

\[
w_{j}^{(i)} = p(y_{i}|x^{(i)}, \rho^{(i)}, z^{(i)})
\] (45)
(4) **Sample collection.** When all the motion sample seeds have been used, collect all the joint samples and weights. These samples and weights provide an approximation to the joint motion and structure posterior distribution using the current feature correspondences, $p(x_t, \rho_t, z|y_t)$.

4 Experimental Results and Performance Analysis

By using the proposed algorithm for Bayesian self-calibration, constant or varying FOV can be recovered and furthermore, the motion of the camera and the scene structure can be reconstructed.

4.1 *Constant Field of View*

Two experimental results using synthetic image sequences are presented first. The synthetic feature trajectories are corrupted by additive white Gaussian noise (AWGN).

*A Case Study*

In the first experiment, the standard deviation (STD) of the AWGN is 0.5 pixel. During the camera motion, rotation about the X axis is present, hence the motion sequence is not critical. The posterior distributions of motion parameters are shown in Figure 3. In each plot, the distributions of the motion parameter at different time instants are listed from the top of the figure to the bottom. Bold solid lines indicate true values of the motion parameters. In Figure 3, plots (a,b,c) are the distributions of the rotational angles $\psi_x, \psi_y$ and $\psi_z$, respectively. They are in the range $[-\pi, \pi]$. The three plots in the
Fig. 3. Camera motion estimation results in the case study. (a)-(g) are the posterior distributions of the camera motion. (h) shows the effective sample sizes during the SIS procedure for camera motion and (i) is the posterior probability of the non-criticalness of the camera motion sequence.

The second row shows the distributions of the translational angles $\alpha$ and $\beta$ and the translation magnitudes. $\alpha$ is in $[0, \pi]$ and $\beta$ in $[0, 2\pi]$. We can see that the resulting posterior distributions of motion parameters have peaks very close to the ground-truths. Figure 3(g) shows posterior distributions of the field of view. The effective sample sizes at different time instants are shown in Figure 3 (h).

Figure 3 (i) shows the probability of the non-criticalness of the motion sequence. The horizontal axis is the time axis and the corresponding value on the vertical axis indicates the probability of the non-criticalness of the motion sequence up to that time. Since this probability converges to one, we know
Fig. 4. Feature point depth estimates in the case study. The empirical mean of the depth samples are shown as dotted lines with squares and the values of ground-truth are marked by circles.

that this motion sequence is not critical.

Figure 4 gives the results of depth estimation. The dotted line shows the minimum mean square error (MMSE) depth estimates and the solid line shows the true depth values. Since the estimates are very close to the true values, it is difficult to distinguish one from the other.

A Critical Motion Sequence

Critical motion sequences were also generated to test the proposed algorithm. One example is included here. In this example, the virtual camera only translates without any rotation. In Figure 5, plots (a)-(e) show the posterior distributions of the camera rotation and translation direction angles; plot (f) displays the distribution of FOV and (g) gives the MMSE estimates (thick line) of the translation magnitudes at different time instants together with the standard deviations (thin lines) of the estimates. Since this sequence is critical with respect to unknown focal length, no unique solutions can be found for the elevation angle $\alpha$, FOV and translation magnitudes. Therefore, large number of peaks exist in the distributions of $\alpha$ and FOV and the estimates of
Fig. 5. Camera motion estimation results when the camera motion is critical. (a)-(f) are the posterior distributions of the camera motion. (g) shows the MMSE estimates of the translation magnitudes (thick curve) and the standard deviation of the estimates (thin lines). Since the camera motion is critical, the uncertainties in the translation magnitude estimates are large. The true values are shown by the straight line. (h) shows the effective sample sizes during the SIS procedure for camera motion. (i) is the posterior probability of the non-criticalness of the camera motion sequence.

translation magnitudes bear large uncertainties. Figure 5 (i) shows the probability of non-criticalness of this motion sequence. It can be seen that the probability of non-criticalness of the camera stays below 0.4 for this sequence, indicating that this motion sequence is critical.

The MMSE estimates of normalize depth are shown in Figure 6. It can be seen that the estimates are very close to the true depth values. This is because that the depth values of the feature points can be determined correctly even when
Fig. 6. Feature point depth estimates when the camera motion is critical. The empirical means of the depth samples are shown as dotted lines with squares and the values of ground-truth are marked by circles. Although the camera motion is critical with respect to an unknown FOV, the feature depth can be uniquely determined. We can see the MMSE estimates of the feature depth are very close to the true values.

the motion sequence is critical.

4.2 Freely Varying Field of View

Now let us look at examples when the FOV of the camera can freely vary.

Circular Camera Motion

In this experiment, a virtual camera moves along a circle and at the same time the field of view of the camera is enlarged. Figure 7 (a) - (g) show posterior distributions of the motion parameters and (h) shows the MMSE estimates of the translation magnitudes. Figure 7 (i) shows the probability of non-criticalness of the motion sequence. It can be seen that this probability initially starts with a relatively low value. At the beginning of the sequence, the out-of-plane rotation is small. Due to observation noise in feature correspondences, the sequence looks like a critical motion sequence. Along with the increase in the
Fig. 7. Camera motion estimation results in the case of circular motion with varying FOV. (a)-(g) are the posterior distributions of the camera motion parameters. (h) shows the MMSE estimates of the translation magnitudes (thick curve) and the standard deviation of the estimates (thin lines). The straight line shows the true values. (i) is the posterior probability of the non-criticalness of the camera motion sequence.

rotation angle about the X axis, the probability of non-criticalness of the sequence approaches to one eventually. The MMSE estimates of depths are shown in Figure 8.

Elliptical Camera Motion

We also tested the proposed algorithm using an image sequence produced by a virtual camera moving along an ellipse with the optical axis of the camera tangent to the ellipse. Recall that this type of motion sequence was found critical [10,11], when the FOV varies. The feature points are spread randomly
Fig. 8. Feature point depth estimates in the case of circular motion with varying FOV. The empirical mean of the depth samples are shown as dotted lines with squares and the values of ground-truth are marked by circles.

in the 3D space and they are not on a plane. The feature correspondences are corrupted by AWGN with standard deviation of one pixel. The motion and depth estimation results are shown in Figures 9 and 10, respectively. We can see that these results are quite close to the true values. Therefore, it has been experimentally shown that it is possible to remove the calibration ambiguity introduced by the motion along an ellipse, which is the second type of CMS mentioned in Section 2.2.

4.3 Bayesian Calibration of Outdoor Surveillance Cameras

In this experiment, an image sequence was collected using an outdoor Philips ENVC120W camera. The vertical FOV of the camera ranges from 2.55 to 44.31 degrees. The camera was manually panned to the left during video capture, with FOV of about of 44.31 degrees.

Figure 11 shows the first and last frames in the image sequence, on which the feature point locations marked by squares are superimposed. According to the configuration of the coordinate system, camera motion contains positive
Fig. 9. Camera motion estimation results in the case of elliptical motion with varying FOV. (a)-(g) are the posterior distributions of the camera motion parameters. (h) shows the MMSE estimates of the translation magnitudes (thick curve) and the standard deviation of the estimates (thin lines). The straight line shows the true values. (i) is the posterior probability of the non-criticalness of the camera motion sequence.

Fig. 10. Feature point depth estimates in the case of elliptical motion with varying FOV. The empirical mean of the depth samples are shown as dotted lines with squares and the values of ground-truth are marked by circles.
Fig. 11. Two frames from an outdoor sequence. The left image is the first frame of the image sequence, superimposed by features (squares) and the right image is the last frame of the sequence.

rotation about the X axis. The camera motion estimation results are shown in Figure 12. We can see that the results show that the motion sequence is not critical, with positive X rotation and the distribution of FOV is around the ground-truth.

4.4 3D Face Modeling Using Uncalibrated Camera

In this example, a face sequence containing 17 frames was captured using SunCamera II, which is an adjustable CCD color camera. The vertical FOV of SunCamera II is 33 degrees (0.576 radian).

Figure 13 shows the first frame of the image sequences, on which the feature points (squares) and their trajectories throughout the entire sequence are superimposed. By using the proposed Bayesian self-calibration algorithm, the FOV of the camera is accurately estimated and a 3D face model is reconstructed. Figure 14 shows the motion estimation results. The MMSE estimate of the vertical FOV at the last time instant is 0.5804 radian and it is very close to the ground-truth. The reconstructed face model is shown in Figure 15. The figure at the upper-left corner is the texture map of the face and the
Fig. 12. Camera motion estimation results for an outdoor sequence. (a)-(f) are the posterior distributions of the camera motion. (g) shows the MMSE estimates of the translation magnitudes (thick curve) and the standard deviation of the estimates (thin lines). Since the camera motion is critical, the uncertainties in the translation magnitude estimates are large. The true values are shown by the straight line. (h) shows the effective sample sizes generated during the SIS procedure for camera motion. (i) is the posterior probability of the non-criticalness of the camera motion sequence.

remaining figures show the 3D face model viewed from different angles.

5 Conclusions

In this paper, we have presented an algorithm for camera self-calibration using SIS. Our efforts have concentrated on the main problem of self-calibration: estimation of the FOV with all the other intrinsic parameters known, where
Fig. 13. The first frame of the face sequence with features (squares) and their trajectories (curves).

Fig. 14. Camera motion estimation results using the face sequence. (a)-(g) are the posterior distributions of the camera motion. (h) shows the effective sample sizes during the SIS procedure for camera motion and (i) is the posterior probability of the non-criticalness of the camera motion sequence.

the unknown FOV can be either constant or varying throughout the image sequence. The proposed algorithm is capable of processing the CMSs and quasi-CMSs and it does not have any specific requirements for initialization.
Fig. 15. The intensity texture map and reconstructed 3D model of the face sequence. The proposed method has also been tested extensively and satisfactory experimental results have been obtained. A future research direction could be the extension of the algorithm to self-calibration with more unknown camera intrinsic parameters such as the position of the principle point and the aspect ratio.
6 Acknowledgments

The work presented in this paper is partially supported by the U.S. Army Research Laboratory (ARL) Collaborative Technology Alliance contract DAAD19-01-2-0008.

References


