STRUCTURE FROM MOTION

by

Akash Chandra Shekar

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Computer Science

Charlotte

2018

ABSTRACT

AKASH CHANDRA SHEKAR. Structure From Motion. (Under the direction of DR. ANDREW WILLIS)

The Structure from Motion (SfM) systems use knowledge of a camera's image formation model and a time-sequence of images from the camera to estimate camera motion and 3D scene structure. Formulations of this problem decompose the problem into two sub-problems: (1) solving fore the camera trajectory and solving for the depth of 3D scene points observed by the camera in multiple images. Solutions are provided by finding correspondences between (x, y) locations in images take from different poses taken at distinct times in the camera trajectory. Traditional SfM find these correspondences by extracting and tracking features, e.g., corner points, match local regions within images. However, these approaches provide very sparse geometric descriptions of the scene structure and omit much of the geometric information in the image. Other , semi-dense, approaches attempt to find large collections of correspondences by solving for pixel-level correspondences which provide more geometric information about the scene. Unfortunately, unlike features, pixel values may are susceptible to viewpoint, illumination, and other image formation phenomena which requires semi-dense approaches to consider photometric correction in order to robustly determine correspondences.

ACKNOWLEDGEMENTS

If you decide to have a acknowledgements page, your acknowledgement text would go here.

The Acknowledgement page should be brief, simple, and free of sentimentality or trivia. It is customary to recognize the role of the advisor, the other members of the advisory committee, and only those organizations or individuals who actually aided in the project. Further, you should acknowledge any outside source of financial assistance, such as GASP grants, contracts, or fellowships.

DEDICATION

If you decide to have a dedication page, your dedication text would go here.

The Dedication page, if used, pays a special tribute to a person(s) who has given extraordinary encouragement or support to one's academic career.

INTRODUCTION

If you decide to have an introduction page, your introduction text would go here.

Depending on the discipline or the requirements of the student's advisory committee, an Introduction may be included as a preliminary page.

TABLE OF CONTENTS

LIST OF FIGURES viii

LIST OF TABLES ix

LIST OF ABBREVIATIONS x

CHAPTER 1: INTRODUCTION 1

CHAPTER 2: Methodology 7

2.1. Rigid Body Motion 7

2.2. Exponential Map 9

2.3. Camera model 11

2.4. Epipolar geometry 16

2.4.1. The eight-point linear algorithm 18

2.4.2. Structure Reconstruction 20

2.5. Non linear Optimization 21

REFERENCES 24

LIST OF FIGURES

FIGURE 2.1: Thin lens camera model 12

FIGURE 2.2: Pin-hole camera model 12

FIGURE 2.3: Epipolar geometry 17

LIST OF TABLES

LIST OF ABBREVIATIONS

ECE	An acronym for Electrical and Computer Engineering.

CHAPTER 1: INTRODUCTION

Structure from Motion (SfM) is the photogrammetric process of estimating the three-dimensional structure of a scene from a set of two-dimensional images, this is achieved by tracking the motion of the camera generating these images and correlating information within the time-sequence of images. SfM has many applications in the field of robotics including augmented reality, geoscience and ***[put more here]***. In robotics, SfM is applied as one approach for visual odometry, the process of estimating the egomotion of an robot using only the inputs of cameras attached to it, this is used highly in self driving car and unmanned aerial vehicle. In the field of augmented reality, SfM used to estimate the lost depths of the environment from images, this depth information enables to augment the virtual objects into the real world and even allows to interact with augmented objects. [depth map not defined, basic physical interaction vague].

The point in 3D can be estimated by triangulation of 2D projections from two or more images, i.e with the known camera orientation, we can back-project 2D points from images onto 3D environment and try to determine their intersection. In order to achieve that we need find a set of points in one image which can be identified as the same points in another image, this is called as image correspondence. Traditionally this was done by feature extraction and matching[1]. The feature extraction is the process of selecting the key points in the image which are unique and distinguishable and represent them in a efficient descriptor,which are later used for matching in the other image. There are many possible choices for features and descriptors, like SIFT, SURF, ORB etc. However in more recent implementations[2], instead of features, the image intensities are used directly for matching. Once the correspondence are established, the intersection between back-projected 3D points is estimated by minimize the re-projection error, which is done by imposing the epipolar constraint, which states that the image points and 3D points should be in a same plane. In order to address the noise in estimating the correspondence, a Gaussian model is assumed with zero mean, hence minimizing the re-projection error gives the maximum likelihood estimate of 3D point. On the other hand, with the known image correspondence and their 3D points, we can estimate the camera pose and orientation with the similar procedure. Where we try to solve for relative camera ordination which minimizes the re-projection error between known 3D and 2D point.

However, SfM is a chicken and egg problem with both depth and camera orientation unknown.This again tries to minimize the re-projection error between the observed 2D image points and predicted (re-projected) 2D image point,which is expressed as the sum of squares of a large number of nonlinear, real-valued functions. Thus, the minimization is achieved using nonlinear least-squares algorithms like Guass-Newton or Levenberg–Marquardt. This process is also called as Bundle adjustment, as it is analogous to the process of optimally adjusting bundles of light rays originating from each 3D feature and converging them on each camera's optical center.

Over the years many real time approaches have be suggested to tackle this problem, and these approaches are broadly classified as Direct vs. Indirect, Dense vs. Sparse and Incremental vs. Global methods.

Direct vs. Indirect: All the SfM approaches have a underlaying probabilistic model in order to address the noise in measurements and camera representation, which uses the Maximum Likelihood approach to maximize the probability of obtaining the actual measurements, this differentiation of SfM is based on how the probabilistic models are defined. The direct method as the name specifies, directly uses the sensor values (pixel values) – light received from a certain direction over a certain time period – as as input for probabilistic model, hence this method tries to minimize the photometric error. This is analogous to the process of image alignment, whose goal is to minimize the sum of squared error between two images, the template T and the image I warped back onto the coordinate frame of the template.

$$\sum_x [I(\omega(x,p) - T(x)]^2 \tag{1.1}$$

1.1 is minimized with respect p, which in case of SFM are the camera orientation, this minimization is a process of non linear optimization.

On the other hand, the indirect method uses the intermediate representation of sensor measurements as input for probabilistic model, the sensor values can be either represented as sparse set feature key-points, this need a efficient feature detectors like Scale Invariant Feature Transform (SIFT) to find the valid the correspondence between images, and the methods like Random Sample Consensus algorithm (RANSAC) are applied in order to avoid the outliers. An alternative to using key features, is to find the dense regularized optical flow, which tries to match the selected points from one image to another, assuming that both images are part of a sequence and relatively close to one another. Both this representation tries to minimize the geometric errors.

Dense vs. Sparse: As the name suggests the sparse methods uses only selected set of independent points, where as dense methods attempt to use and reconstruct all pixels in the 2D image domain. On the same line there is a intermediate approaches (semi-dense) tries to use the largely connected and well-constrained subset of points if not all of them. The choice of this method affects the addition of geometry prior, in the sparse formulation, since there is no notion of neighborhood, and geometry parameters (key-point positions) are conditionally independent. Dense (or semi-dense) approaches on the other hand exploit the connectedness of the used image region to formulate a geometry prior, typically favoring smoothness.

Incremental vs. Global:Incremental SfM[3] begins by first estimating the 3D structure and camera poses of just two cameras based on their relative pose. Then additional cameras are added on incrementally and 3D structure is refined as new parts of the scene are observed. On the other hand global SfM[4] considers the entire problem at once. Here the objective is to estimate the global camera poses and 3D structure by removing outliers and by applying an averaging scheme [cite ???]. [why is this discussed? are their results different? is one of interest to you?] [cited eariler][isn't the LSD a incremental method]

LSD SLAM [2] is a semi-dense, direct method [you have not defined sparse methods, dense methods, direct methods, or indirect methods do this in a review of approaches above – similar to that from Direct Sparse Odometry article (https://arxiv.org/pdf/1607.02565.pdf) from TUM] [have done changes] optimizes the geometry directly on the image intensities, which enables using all information in the image.In addition to higher accuracy and robustness in particular in environments with little key-points, this provides substantially more information about the geometry of the environment.Images are aligned by Gauss -Newton minimization of the photometric error.

$$E(\xi) = \sum_i (I_{ref}(P_i) - I(\omega(p_i, D_{ref}(p_i), \xi)))^2$$

Where D_{ref} is the estimated depth of reference frame,$\xi \in se(3)$ is Lie-algebra representation of rigid body motion and ω is the affine wrap function. The above error function gives the maximum-likelihood estimator for ξ assuming i.i.d. Gaussian residuals. $\delta\xi^{(n)}$ is computed for each iteration by solving for the minimum of Gauss-Newton second-order approximation of E: [no description of image warping needed here, briefly discuss image warping for direct methods as a methods for image pixel matching see Baker2004 http://www.ncorr.com/download/publications/bakerunify.pdf]

$$\delta\xi^{(n)} = -(J^T J)^{-1} J^T r(\xi^{(n)})$$

with

$$J = \frac{\partial r(\epsilon \circ \delta\xi^{(n)})}{\partial \epsilon}$$

The new estimate is then obtained by multiplication with the computed update

$$\xi^{(n+1)} = \delta\xi^{(n)} \circ \xi^{(n)}$$

The overall system is composed of three major components, tracking, depth map estimation and map optimization.

The tracking component continuously estimates the rigid body pose with respect to the current key-frame, using the pose of the previous frame as initialization.LSD slam tracks new frame by minimizing the variance-normalized photometric error

$$E_p(\xi_{ji}) = \sum_{p \in \Omega_{D_i}} \left\| \frac{r_p^2(p, \xi_{ji})}{\sigma^2_{r_p(p,\xi_{ji})}} \right\|_\delta$$

with

$$r_p(p, \xi_{ji}) := I_i(p) - I_j(\omega(p, D_i(p), \xi_{ji}))$$

$$\sigma^2_{r_p(p,\xi_{ji})} := 2\sigma_I^2 + \left(\frac{\partial r_p(p, \xi_{ji})}{\partial D_i(p)} \right)^2 V_i(p)$$

where $\|.\|$ is the Huber norm

$$\left\| r^2 \right\|_\delta := \begin{cases} \frac{r^2}{2\delta} if \; \|r\| \leq \delta \quad , \|r\| - \frac{\delta}{2} \; otherwise \end{cases}$$

applied to the normalized residual. The depth map estimation component uses tracked frames to either refine or replace the current key-frame. Depth is refined by filtering over many per-pixel, small-baseline stereo comparisons coupled with interleaved spatial regularization. If the camera has moved too far, a new key-frame is initialized by projecting points from existing, close-by key-frames into it. Each key-frame is scaled such that its mean inverse depth is one, which enables more small-baseline stereo comparisons. For every new key-frame added, the possibility of loop closure is checked by performing the reciprocal tracking check. The map optimization component is responsible for the updating depth map into global map, it detect loop closure and scale drift by estimating similarity transform (sim(3)) to close by existing key-frames.The global map is represented as a pose graph consisting of key-frames as vertices's with 3D similarity transforms as edges, elegantly incorporating changing scale of the environment and allowing to detect and correct accumulated drift. Each key-frame consists of a camera image, an inverse depth map and variance of the inverse depth.Edges between key-frames contain their relative alignment as similarity transform, as well as corresponding covariance matrix. The map, consisting of a set of key-frames and tracked sim(3)-constraints, is continuously optimized in the background using pose graph optimization. The error function that is minimized is defined by (W defining the world frame)

$$E(\xi_{W1}....\xi_{Wn}) \coloneqq \sum_{(\xi_{ji},\Sigma_{ji})\in\varepsilon} (\xi_{ji} \circ \xi_{Wi}^{-1} \circ \xi_{Wj})\Sigma_{ji}^{-1}(\xi_{ji} \circ \xi_{Wi}^{-1} \circ \xi_{Wj})$$

CHAPTER 2: Methodology

Structure from motions (SfM) is the process of triangulating the three-dimensional structure from two-dimensional images, along with estimating the motion of camera (visual odometer), hence it is some time called as Visual SLAM.

2.1 Rigid Body Motion

One of the goals of SfM it to estimate the Camera trajectory, which is a rigid body motion. We need an efficient model to represent and compute this rigid body motion. The camera position is represented by a 3D Vector in an Euclidean space, this camera position is chosen to represent the world frame and specify the translation and rotation of the scene relative to that frame. The rigid body motion itself is composed of a rotation and translation.

Traditionally, rotation is represented by a 3×3 special orthogonal matrix called rotational matrix. Special Orthogonal matrix SO(3) are the matrix which satisfy $R^T R = RR^T = I$ and have a determinant of +1.

$$SO(3) = \{R \in \mathbb{R}^{3\times3} \mid R^T R = I, det(R) = +1\}$$

The rotation transformation of the coordinates $X_c$of a point p relative to frame C to its coordinates X_w relative to frame W is

$$X_w = R_{wc} X_c$$

Also, because the rotational matrix are orthogonal, we have$R^{-1} = R^T$, on this line the inverse transformation of coordinates are

$$X_c = R_{wc}^{-1} X_w = R_{wc}^T X_w$$

The continuous rotation of a camera is described as a trajectory $R(t) : t \to SO(3)$ in the space $SO(3)$.When the starting time is not t = 0, the relative motion between time t_2 and time t_1 will be denoted as $R(t_2, t_1)$. The composition law of the rotation group implies

$$R(t_{2,}t_0) = R(t_{2,}t_1) \times R(t_{1,}t_0), \vee t_0 < t_1 < t_2 \in R$$

On the other hand the translation is represented by a $T \in R^3$,1×3 vector which adds the translation values in each dimension.With this, the complete rigid body motion is represented by

$$X_w = R_{wc} X_c + T_{wc} \tag{2.1}$$

However, the above equation is not linear but affine. We may convert this to linear by using homogeneous coordinates, where we append 1 for 1×3 vector and make it a 1×4 vector,

$$\bar{X} = \begin{bmatrix} X \\ 1 \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{bmatrix} \in \mathbb{R}^4$$

With this new notation for point, we can rewrite the transformation from equation 6 as following

$$\bar{X}_w = \begin{bmatrix} X_w \\ 1 \end{bmatrix} = \begin{bmatrix} R_{wc} & T_{wc} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ 1 \end{bmatrix} = \bar{g}_{wc} \bar{X}_c \tag{2.2}$$

where the 4×4 matrix $\bar{g}_{wc} \in R^{4\times4}$is called the homogeneous representation of the rigid-body motion.

The set of all possible configurations of a rigid body can then be described by the space of rigid-body motions or special Euclidean transformations called SE(3)

$$SE(3) = \{\bar{g} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \mid R \in SO(3), T \in R^3\} \subset \mathbb{R}^{4\times4}$$

Finally, one other dimension we could consider is the scale, this can represented by a Similarity transformation $Sim(3)$, which is the composition of a rotation, translation and a uniform rescaling,

$$Sim(3) = \{\bar{g} = \begin{bmatrix} R & T \\ 0 & s^{-1} \end{bmatrix} \mid R \in SO(3), T \in R^3, s \in R\} \subset \mathbb{R}^{4\times4}$$

2.2 Exponential Map

The special orthogonal group in three dimensions can be represented by a 3×3 rotation matrix $R \in SO(3)$, which must satisfy the constraint $R^T R = I$, this implies that the $SO(3)$ transformations leaves the quantity $x^2 + y^2 + z^2$ invariant. The group $SO(3)$ has 9 parameters, but the invariance of the length produces six independent conditions, leaving three free parameters, Hence, the dimension of the space of rotation matrices $SO(3)$ should be only three, and six parameters out of the nine are in fact redundant.We can use this to have better representation of Rigid body motion.

The $SO(3)$, $SE(3)$ and $SIM(3)$are categorized under the special group called the Lie group, which represents the smooth differentiable manifolds. Every Lie group has a tangent space at identity called the Lie algebra, which is a vector space used to study the infinitesimal transformations.

For rotation group $SO(3)$ its lie algebra $so(3)$ is represented by

$$so(3) \doteq \left\{ \hat{\omega} \in \mathbb{R}^{3\times 3} \mid \omega \in \mathbb{R}^3 \right\}$$

where $\hat{\omega}$is a skew symmetric matrix for the vector ω

The map from the space $so(3)$ to $SO(3)$ is called the exponential map.

$$exp : so(3) \rightarrow SO(3); \hat{\omega} \mapsto e^{\hat{\omega}}$$

$$R(t) = e^{\hat{\omega}t} \tag{2.3}$$

The equation 2.3 represents a rotation around the axis $\omega \in \mathbb{R}^3$ by an angle of t radians. And the inverse mapping is obtained by logarithm of SO(3)

$$log : SO(3) \rightarrow so(3); log(R) \mapsto \hat{\omega}$$

We can extend this to full rigid body motion which also involves the translation along with the rotation, for rotation group $SE(3)$ its lie algebra $se(3)$ is represented by .

$$se(3) \doteq \left\{ \hat{\xi} = \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix} \mid \hat{\omega} \in so(3), v \in \mathbb{R}^3 \right\} \subset \mathbb{R}^{4\times 4}$$

with $v(t) = \dot{T}(t) - \hat{\omega}(t)T(t) \in \mathbb{R}^3$

Similarly, the exponential map from the space $se(3)$ to $SE(3)$is given by

$$exp : se(3) \rightarrow SE(3); \hat{\xi} \mapsto e^{\hat{\xi}}$$

and as before inverse to the exponential map is defined by logarithm

$$log : SE(3) \rightarrow se(3); log(g) \mapsto \hat{\xi}$$

[THIS IS TOO SIMILAR to source work elsewhere, bordering on plagiarism – REMOVE!, you should not be going into this much detail for the differential properties of rotations as this is not something you are comfortable with, write down the most important relationships, i.e., the differential of the rotation and why you need it for SfM, you will want to define sim3 transforms and how they are distinct from se3 transforms then leave this topic] [I wrote this, because I had done some study on exponential map between lie group and lie algebra and vis versa, I was hoping to keep this, but let me know your thoughts] [exponential map of SIM(3) is way too complicated and I am working on it, I will incorporate it going forward.]

2.3 Camera model

The 2D image is formed by capturing the light energy (irradiance) for every pixel, this process can be mathematically represented by thin lens camera model, which describes the relationship between the three-dimensional coordinators to its projection onto the image plane. The thin lens model is represented by a optical axis and a perpendicular plane called the focal plane.The thin lens itself is characterized by its focal length and diameter, the focal length is the distance from optic center to where all the ray intersect the optic axis, while the point of intersection itself is called the focus of the lens. One of the important properties to consider is that the rays entering the lens through optic center are undeflected while the rest of the rays are. With this model the irradiance on the image plane is obtained by the integration of all the energy emitted from region of space contained in the cone determined by the geometry of the lens.

Using similar triangles, from above figure, we obtain the following fundamental equation of the thin lens

$$\frac{1}{Z} + \frac{1}{z} = \frac{1}{f}$$

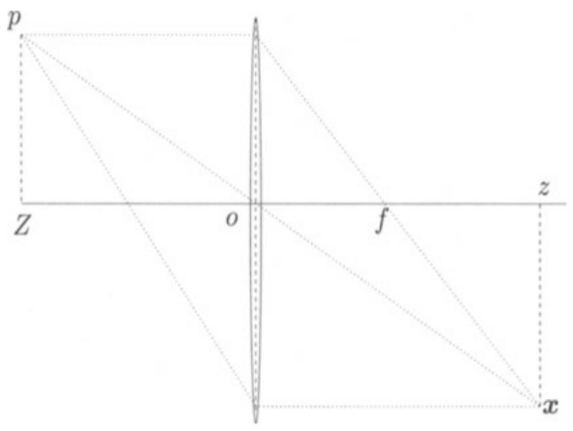

Figure 2.1: Thin lens camera model

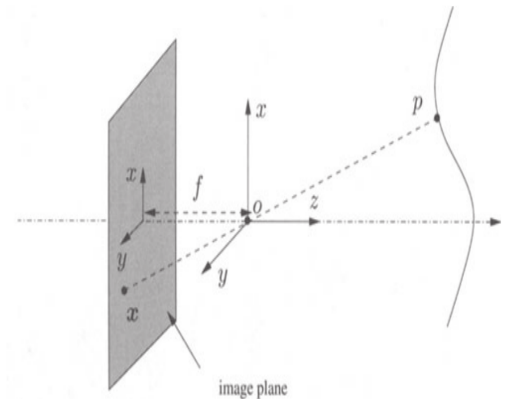

Figure 2.2: Pin-hole camera model

For the simplification of calculation, we consider a Ideal camera model called the pinhole camera model, here the aperture of a thin lens is assumed to decreased to zero, all rays are forced to go through the optical center o, and therefore they remain undeflected.Consequently,as the aperture of the cone decreases to zero, the only points that contribute to the irradiance at the image point $x = [x, y]$ are on a line through the center 0 of the lens. If a point p has coordinates $X = [X, Y, Z]$ relative to a reference frame centered at the optical center 0, with its z-axis being the optical axis (of the lens), then it is immediate to see from similar triangles in Figure that the coordinates of p and its image x are related by the so-called ideal perspective projection.

$$x = -f\frac{X}{Z} \tag{2.4}$$

$$y = -f\frac{Y}{Z} \tag{2.5}$$

[write projection as a matrix transform similar to ROS][have it represented below]This mapping of 3D point to 2D is called projection and is represented by π

$$\pi : R^3 \to R^2$$

This is also written as $x = \pi(X)$.

The negative sign in the 2.4 and 2.5 makes the object appear upside down on the image plan, we can handle this by moving the image plane to front of optic center to $\{z = -f\}$ this will make $(x, y) \to (-x, -y)$.There for the equation 2 and 3 changes to

$$x = f\frac{X}{Z}$$

$$y = f\frac{Y}{Z}$$

This can be represented in matrix form as

$$x = \begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z}\begin{bmatrix} X \\ Y \end{bmatrix}$$

In homogeneous coordinates, this relationship can be modified as

$$Z\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

The above equation can be decomposed into

$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

with

$$K_f = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \in R^{3\times 3}, \Pi_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 4}$$

The matrix $\Pi_0$is a standard projection matrix.

With the rigid body transformation from 2.2, we can represent the overall geometric model for an ideal camera as below

$$\lambda\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}\begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix} \tag{2.6}$$

Where λ is the unknown scale factor.

However, the above equation represents the ideal model where the retinal frame centered at the optical center with one axis aligned with the optical axis.But in practice, this does not true and the origin of the image coordinate frame typically in the upper-left corner of the image.we need to address this relationship between the retinal plane coordinate frame and the pixel array in our camera model. This can be represented by a special matrix as following

$$K_s = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3\times 3}$$

with these new parameters the we can represent the interstice parameters of camera as following

$$K = K_s K_f = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} fs_x & s_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

where

- o_x: x-coordinate of the principal point in pixels,
- o_y: y-coordinate of the principal point in pixels,
- $fs_x = \alpha_x$: size of unit length in horizontal pixels,
- $fs_y = \alpha_y$: size of unit length in vertical pixels,
- $\frac{\alpha_x}{\alpha_y}$: aspect ratio σ,
- fs_θ: skew of the pixel, often close to zero.

Now, the ideal camera model 2.6can be updated as

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix} \tag{2.7}$$

In the matrix notation,

$$\lambda x = K\Pi_0 g X_0 \tag{2.8}$$

To summarize, 2.8 represents the projection of three-dimensional coordinates X_0 by camera with orientation (extrinsic parameters)g and intrinsic parameters K, onto two-dimensional coordinate x with known scale λ

In addition to above linear distortion, if a camera has a wide field of view, there will be a significant distortion along radial directions called radial distortion. Such a distortion can be models by

$$x = x_d(1 + a_1 r^2 + a_2 r^4)$$

$$y = y_d(1 + a_1 r^2 + a_2 r^4)$$

where (x_d, y_d) are the coordinates of the distorted points, $a_1, a_2$are the coefficients of radial distortion and r is the radius of the radial distortion.

2.4 Epipolar geometry

Consider two images of the same point p from two camera position with relative pose (R, T), where $R \in SO(3)$ is the relative orientation and $T \in \mathbb{R}^3$is the relative position,then if $X_1, X_2 \in \mathbb{R}^3$ are the 3-D coordinates of a point p relative to the two camera frames, by the rigid-body transformation we have

$$X_2 = RX_1 + T$$

Now,let $x_1, x_2 \in \mathbb{R}^3$be the homogeneous coordinates of the projection of the same point p in the two image planes with respective unknown scales of $\lambda_1$and λ_2.

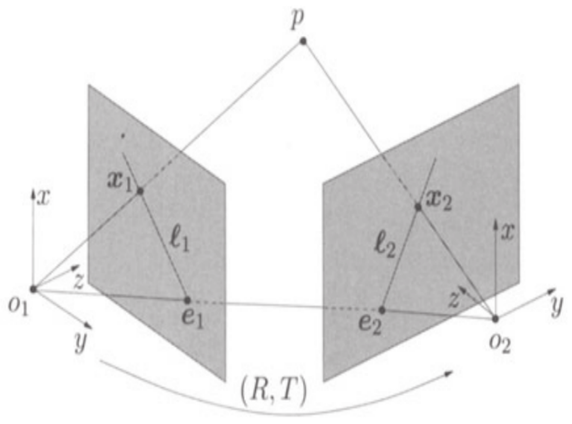

Figure 2.3: Epipolar geometry

$$\lambda_2 x_2 = R\lambda_1 x_1 + T$$

By multiplying both the side by $\hat{T}$

$$\lambda_2 \hat{T} x_2 = \hat{T} R \lambda_1 x_1$$

By multiplying both the side by x_2

$$0 = x_2^T \hat{T} R \lambda_1 x_1 \tag{2.9}$$

This is the epipolar constraint and the matrix $E = \hat{T}R$ is called the essential matrix.It encodes the relative pose between the two cameras. Geometrically, it impose that the The vector connecting the first camera center o_1 and the point p, the vector connecting o_2 and p, and the vector connecting the two optical centers o_1 and o_2 clearly form a triangle. Therefore, the three vectors lie on the same plane.Hence the their triple product which measures the volume of the parallelepiped is zero.

2.4.1 The eight-point linear algorithm

With epipolar constrain between two images, we should be able to retrieve the relative pose of the cameras. The eight-point linear algorithm is a simple closed-form algorithm, it consists of two steps: First a matrix E is recovered from a number of epipolar constraints; then relative translation and orientation are extracted from E.

The entries of E are denoted by

$$E = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \in \mathbb{R}^{3\times3}$$

The matrix E is reshaped into vector $E \in \mathbb{R}^9$

$$E = [e_{11}, e_{21}, e_{31}, e_{12}, e_{22}, e_{32}, e_{13}, e_{23}, e_{33}]^T$$

and for $x_1 = [x_{1,}y_1, z_1]^T \in \mathbb{R}^3$and $x_2 = [x_{2,}y_2, z_2]^T \in \mathbb{R}^3$ define

$$a = [x_1x_2, x_1y_2, x_1z_2, y_1x_2, y_1y_2, y_1z_2, z_1x_2, z_1y_2, z_1z_2] \in \mathbb{R}^9$$

With these new notations, we can rewrite the 2.9 as below

$$a^T E = 0$$

this representation emphasizes the linear dependence of the epipolar constraint on the elements of the essential matrix.

Now,with a set of corresponding image points $(x_1^j, x_2^j), j = 1, 2, ..., n$ we can define a matrix $\chi \in \mathbb{R}^{n\times9}$ associated with these measurements to be

$$\chi = \left[a^1, a^2, ..., a^n\right]^T$$

In the absence of noise, the vector E satisfies

$$\chi E = 0$$

In order to obtain the unique solution, the rank of the matrix $\chi \in \mathbb{R}^{n\times 9}$ needs to be exactly eight. This should be the case when we have $n \geq 8$ "ideal" corresponding points, hence the name The eight-point linear algorithm.

Because of errors in correspondences, we try to find the E that minimizes the least-squares error function $\|\chi E\|^2$.We do this by choosing eigenvector of $\chi^T\chi$ that corresponds to its smallest eigenvalue.

Once we have E, we need to extract the pose ($R \in SO(3)$ and $T \in \mathbb{R}^3$) from E, we know that [5]a nonzero matrix $E \in \mathbb{R}^3$ is an essential matrix if and only if E has a singular value decomposition $(SVD) E = U\varSigma V^T$with

$$\varSigma = diag\{\sigma, \sigma, 0\}$$

for some $\sigma > 0$ and $U, V, \in SO(3)$

With this we can obtain two relative pose

$$(\hat{T}_1, R_1) = (UR_Z(+\frac{\pi}{2})\Sigma U^T, UR_Z^T(+\frac{\pi}{2})V^T)$$

$$(\hat{T}_2, R_2) = (UR_Z(-\frac{\pi}{2})\Sigma U^T, UR_Z^T(-\frac{\pi}{2})V^T)$$

Among these one with that gives the meaningful (positive) depth are selected as the valid pose.

With $R_Z(\pm\frac{\pi}{2}) = \begin{bmatrix} 0 & \pm 1 & 0 \\ \pm 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

2.4.2 Structure Reconstruction

One remaining thing to find is the position of points in three-dimension by recovering their depths relative to each camera frame.With the estimated pose (Translation is T is defined up to the scale γ) and point correspondence, we have

$$\lambda_2^j x_2^j = \lambda_1^j R x_1^j + \gamma T$$

for $j = 1, 2, 3..., n$

where, $\lambda_1$and $\lambda_2$are depths with respect to the first and second camera frames, respectively. One of the depths is redundant, if λ_1 is known, we can estimate λ_2 as a function of (R, T).Hence we can eliminate, say, λ_2 from the above equation by multiplying both sides by $\hat{x_2}$

$$0 = \lambda_1^j x_2^j R x_1^j + \gamma x_2^j T$$

This is represented as linear equations

$$M^j \bar{\lambda^j} = \left[\hat{x_2^j} R x_1^j, \hat{x_2^j} T\right] \begin{bmatrix} \lambda_1^j \\ \gamma \end{bmatrix} = 0$$

where $M^j = \left[\hat{x_2^j} R x_1^j, \hat{x_2^j} T\right] \in \mathbb{R}^{3\times2}$and $\bar{\lambda^j} = \begin{bmatrix} \lambda_1^j \\ \gamma \end{bmatrix} \in \mathbb{R}^2$ for $j = 1, 2, 3..., n$

since all n equations above share the same γ; we define a vector $\vec{\lambda} = [\lambda_1^1, \lambda_1^2, ..., \lambda_1^2, \gamma]$ and a matrix $M \in \mathbb{R}^{3n\times(n+1)}$as

$$M = \begin{bmatrix} \hat{x}_2^1 R x_1^1 & 0 & 0 & 0 & 0 & \hat{x}_2^1 T \\ 0 & \hat{x}_2^2 R x_1^2 & 0 & 0 & 0 & \hat{x}_2^2 T \\ 0 & 0 & \ddots & 0 & 0 & \vdots \\ 0 & 0 & 0 & \hat{x}_2^{n-1} R x_1^{n-1} & 0 & \hat{x}_2^{n-1} T \\ 0 & 0 & 0 & 0 & \hat{x}_2^n R x_1^n & \hat{x}_2^n T \end{bmatrix}$$

Then the equation

$$M\overrightarrow{\lambda} = 0$$

determines all the unknown depths up to a single universal scale. The linear least-squares estimate of $\overrightarrow{\lambda}$ is simply the eigenvector of $M^T M$s that corresponds to its smallest eigenvalue.

2.5 Non linear Optimization

In practice, because of the noise in image correspondence and other errors we cannot measure the actual coordinates but only their noisy versions, say

$$\widetilde{x}_1^j = x_1^j + \omega_1^j$$

$$\widetilde{x}_2^j = x_2^j + \omega_2^j$$

where $x_1^j$and $x_2^j$are the ideal image coordinates and $\omega_1^j = \left[\omega_{11}^j, \omega_{12}^j, 0\right]^T$and $\omega_2^j = \left[\omega_{21}^j, \omega_{22}^j, 0\right]^T$are localization errors (called residuals) in the correspondence.Therefore, we need a way to optimize the parameters (x, R, T) that minimize this errors.

One of the minimalistic approach to optimality is to minimize the squared 2-norm of residuals, if we choose the first camera frame as the reference

$$\phi(x,R,T,\lambda) = \sum_{j=1}^{n} \left\|\omega_1^j\right\| + \left\|\omega_2^j\right\|^2 = \sum_{j=1}^{n} \left\|\widetilde{x}_1^j - x_1^j\right\|^2 + \left\|\widetilde{x}_2^j - \pi(R\lambda_1^j x_1^j + T)\right\|^2$$

The above error is often called the "re-projection error", since x_1^j and $x_2^j$are the recovered 3-D points projected back onto the image planes. This process of minimizing the above expression for the unknowns (R, T, x_1, λ) is known as bundle adjustment[6].

One of the many ways to minimize this squared error is the Gauss-Newton method. Gauss-Newton is a iterative method for finding the value of the variables which minimizes the sum of squares,it starts with the initial guess and this method does not need the second derivatives (Hessian matrix) of the of function, which is often expensive and sometimes not possible to compute, instead the Hessian is approximated with the Jacobian matrix of the function.

For the least-square function of form

$$\min_{x} \sum_{i} r_i(x)^2$$

Gauss-Newton method iteratively solves

$$x_{t+1} = x_t - H^{-1}g$$

with gradient g

$$g_j = 2\sum_{i} r_i \frac{\partial r_i}{\partial x_i}$$

and the Hessian H

$$H_{jk} = 2\sum_{i} r_i \left(\frac{\partial r_i}{\partial x_j}\frac{\partial r_i}{\partial x_k} + r_i \frac{\partial^2 r_i}{\partial x_i \partial x_k}\right)$$

by dropping the second order term we can approximate the Hessian matrix with Jacobian matrix

$$H_{jk} = 2\sum_i J_{ij}J_{ik}$$

with

$$J_{ij} = \frac{\partial r_i}{\partial x_j}$$

The modification to Gauss-Newton method is The Levenberg-Marquardt algorithm,

$$x_{t+1} = x_t - (H + \lambda I)^{-1} g$$

this modification is a hybrid between the Newton method ($\lambda = 0$) and a gradient descent step size $1/\lambda$ for $\lambda \longrightarrow \infty$

REFERENCES

[1] G. Klein and D. Murray, “Parallel tracking and mapping for small AR workspaces,” in *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR’07)*, (Nara, Japan), November 2007.

[2] J. Engel, T. Schöps, and D. Cremers, “Lsd-slam: Large-scale direct monocular slam,” pp. 834–849, 2014.

[3] N. Snavely, S. M. Seitz, and R. Szeliski, “Photo tourism: Exploring photo collections in 3d,” *ACM Trans. Graph.*, vol. 25, pp. 835–846, July 2006.

[4] K. Wilson and N. Snavely, *Robust Global Translations with 1DSfM*, pp. 61–75. Cham: Springer International Publishing, 2014.

[5] T. S. Huang and O. D. Faugeras, “Some properties of the e matrix in two-view motion estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 1310–1312, Dec 1989.

[6] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle adjustment - a modern synthesis,” in *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV ’99, (London, UK, UK), pp. 298–372, Springer-Verlag, 2000.