Universitè de Bourgogne

Master Thesis

Synchronization of Projective Frames

Author:
Vishnu Veilu Muthu

Supervisor:
Prof. Andrea Fusiello
Dr. Federica Arrigoni

A Thesis Submitted for the Degree of
Master of Science in Computer Vision (MsCV)
in the

Universitè de Bourgogne

Carried out in the

Università degli Studi di Udine

· June 2017 ·

Abstract

Synchronization of projective frames is a method of integrating group of projective reconstructed matrices in such a way that they differ from the true reconstruction by a single global projective transformation. In theory, many standard methods are available for the projective reconstruction but most method suffers from common drawbacks which requires multiple iterative process and may not converge or only converge to a local minimum. To avoid such problems, we use a technique called “Global Synchronization”. Synchronization is a method of obtaining a unique solution of transformations, where the process is done by arranging the transformation between each camera of different views in a global network and solving them using graph modeling. The essence of the synchronization problem is thus modeled by a graph, where the cameras are associated to the nodes, and the transformation to the edges. This method of solving problem has appeared under various forms in different settings, such as sensor network localization, formation control in the control system and robotics communities, structure from motion in computer vision, and graph drawings in the discrete mathematics community. Thus our method produces global results also we do not have to initialize any values prior to the process and our method can also handle real world difficult cases like missing data in a unified manner. The underlying thesis demonstrate the performance of the new algorithms on random image sequences in synthetic data (both by adding image noise and missing data).

“Don’t take rest after your first victory because if you fail in second, more lips are waiting to say that your first victory was just luck.”

- A.P.J Abdul Kalam

Contents

List of Figures

List of Algorithms

Acknowledgments

First and foremost, I want to thank my supervisors, Prof. Andrea Fusiello and Dr. Federica Arrigoni for giving me an opportunity to work under their guidance and providing continual support throughout my thesis. Whenever I needed support, suggestion or anything their door was always open for help. Their patience, motivation and enthusiasm deserve special gratitude.

A very special gratitude goes out to Prof. David Fofi and Dr. Cédric Demonceaux for guidance and support throughout my stay. I would like to thank all the coordinators of VIBOT and MsCV program Dr. Yohan Fougerolle, Dr. Desire Sidibe, Dr. Oliver Morel, Dr. Robert Martì and Dr. Daniel Clark, for making this course possible. I would also like to pay my gratitude to all the administrative staff including Herma, Mathilde, Margaux and Aina who helped me at every step of this master course.

A special thanks to Joan Massich, Cansen Jiang, Dp Paudel, Richa Agarwal, Shrinivasan, Sepehr, Ak and Devesh who were always willing to help during my stay. They also provided me undying encouragement throughout my stay for which I am eternally grateful. I also want to thank all the professors and teachers in University of Burgundy and University of Girona who ensured that we not only had a great learning experience but also a memorable time during the program.

I also wish to acknowledge the love and support of my friends especially Yannik, Khatib, Bastien, Raabid, Gourab, Yogesh, Jose, Songyou, Kaisar, Èric, Kibrom, Benjamin,... and every Vibot and MsCV students who helped me during the masters program. They have enriched my life in the past years with fun, excitement and the time spent together shall never be forgotten. I am grateful to my parents, younger brother and relatives who have provided me through moral and emotional support in my life. Many Thanks!

Author:
Vishnu Veilu Muthu

Chapter 1

Introduction

Synchronization of projective frames is a method of integrating group of projective reconstructed matrices in such a way that they differ from the true reconstruction by a single global projective transformation. Projective reconstruction refers to the computation of the structure of a scene from images taken with uncalibrated cameras, resulting in a scene structure, and camera motion that differ from the true geometry by an unknown 3D projective transformation. In theory, many standard methods are available for the projective reconstruction but most method suffers from common drawbacks which requires multiple iterative process and may not converge or only converge to a local minimum. To handle such problems, we use a optimal solution called "Global Synchronization".

In the field of science, with the pieces of information from single block or unit, there is always a need for a better way to communicate with other units. This rule also applies with many new technology like internet, cloud,etc. Such idea is utilized in this thesis work. Synchronization is a method of obtaining a unique solution of transformations, where the process is done by arranging the transformation between each camera of different views in a global network and solving them using graph modeling. The essence of the synchronization problem is thus modeled by a graph, where the cameras are associated to the nodes, and the transformation to the edges.

In general, monocular cameras can measure line of sight to the other cameras but are not able to easily determine the distances and the position which leads to localization problem. As such, this problem has appeared under various forms in different settings, such as sensor network localization and formation control in the control system and robotics communities, structure from motion in computer vision, and graph drawings in the discrete mathematics community. Thus our method produces global results also we do not have to initialize any values prior to the process and our method can also handle real world difficult cases like missing data in a unified manner. The algorithm uses a simple feature extraction for images and uses for the reconstruction process, thus we can increase the time cost. The applications of the work are in the analysis of dynamical scenes where cameras are deployed in a static configuration or are mounted on robots.

1.1 Objectives

This master thesis is a step ahead in a direction to develop a method to find reconstruction of projective frames. This work aims to provide an overview and development on the performance of synchronization in complex matrices. The final goal of the thesis work is to survey the heterogeneous literature, study the conditions under which a projective reconstruction is made and analyze synchronization problem. This can be achieved by:

- Researching the state of the art literature on using projective reconstruction and projective factorization.
- Developing an analytical form to embed the synchronization problem.
- Implementing the method and validate the results in simulations.
- Experimenting the method on different sets of image data.

1.2 Document Organization

This thesis is organized into different chapters starting with a brief introduction about synchronization of projective frames. Chapter 2 provides a detailed discussion on the background about projective reconstruction. Then comes the description of the synchronization and related work to the research in Chapter 3. The algorithm is discussed in detail in Chapter 4. Chapter 5 of the thesis highlights the setup, experiments and the results. Finally conclusion and future work are addressed in Chapter 6.

Chapter 2

Projective Reconstruction

In this chapter, we consider the problem of projective reconstruction. Projective reconstruction is a method of computing the structure of a scene from images or corresponding points taken with uncalibrated cameras. This results in a scene structure and camera motion that may differ from the true geometry by an unknown 3D projective transformation. The task of reconstruction is to determine the unknown quantities of configuration of the corresponding 3D points and the locations of the cameras that projects the images. This gives the general idea behind the projective reconstruction and the problem of estimating the projective transformation matrix. The steps of recovery of the transformation between projective matrices are described in detail and different types of transformations are shown in the Appendix A.

2.1 Projective reconstruction

Projective geometry is a study of non-metrical geometric properties that are invariant with respect to projective transformations. Consider n stationary 3D points distributed in space as $X_j = [x_j, y_j, z_j]^T \in \mathbb{R}^3$ where $j = 1, ..., n$. Under m projective cameras P_i where $i = 1, ..., m$, the 3D point X_j is projected onto image points $x_{ij} = [u_{ij}, v_{ij}]^T \in \mathbb{R}^2$. The vector x_{ij} represents the image coordinates of the j-th 3D point seen in the i-th image. It is generally not possible that every points will be seen in every image frame, so only a subset of all possible x_{ij} are given. The projective reconstruction is to determine the camera projection matrices P_i and the 3D point locations X_j such that the projection of the j-th point in the i-th image is the measured x_{ij}. Assuming a pinhole (projective) camera model, this relationship is expressed as,

$$\mathbf{x_{ij}} \simeq \mathbf{P_i}\,\mathbf{X_j}, \tag{2.1}$$

where P_i is a 3×4 matrix of rank 3, X_j and x_{ij} are expressed in homogeneous coordinates, and the equality is intended to hold only up to an unknown scale factor λ_{ij}. Therefore, the projection equation is give by,

$$\lambda_{\mathbf{ij}}\,\mathbf{x_{ij}} = \mathbf{P_i}\,\mathbf{X_j}. \tag{2.2}$$

From the above equation, the projective matrix can be split to intrinsic and extrinsic parameters of the camera which is given by,

$$\mathbf{P} = \mathbf{K}\,[\mathbf{R}|\mathbf{t}], \tag{2.3}$$

where K is the intrinsic parameters which encodes the transformation in the image plane from the normalized camera coordinates to pixel coordinates. t and R are the extrinsic parameters which describes the position and angular attitude of the camera with respect to an external world coordinate system.

To understand projective geometry in detail, we start with epipolar geometry. The epipolar geometry describes the geometric relationship between two or more perspective views of the same 3-D scene. Let us now consider corresponding image points x_i and x'_j of different camera matrices P_i and P_j that must lie on particular image lines, which can be computed without information on the calibration of the cameras. This implies that, given a point in one image, one can search the corresponding point in the other image along a line. A 3D point and the camera projection centers define a plane that is called epipolar plane and the line connecting the camera projection centers is called the baseline. The baseline intersects each image plane in a point called epipole. The two conjugate points x_i and x'_j, follows the two rays back-projected from image points and they will intersect at the 3D scene point X. Thus, by using the baseline and epipole the reconstruction of 3D point X can be found by solving,

$$\mathbf{e_r} = \zeta_\mathbf{i}\,\mathbf{x_i} - \zeta_\mathbf{j}\,\mathbf{x'_j}, \tag{2.4}$$

where ζ_i and ζ_j are the unknown depth of X. We can rearrange 2.4 as,

$$\zeta_\mathbf{j} = \frac{(\mathbf{e_r} \times \mathbf{x_i}).(\mathbf{x_i} \times \mathbf{x'_j})}{\left\|\mathbf{x_i} \times \mathbf{x'_j}\right\|^\mathbf{2}}. \tag{2.5}$$

Thus, using the depth values the 3D point can be reconstructed using:

$$\mathbf{X} = \begin{Bmatrix} -\,P_{1:3}^{-1}\,P_4 \\ 1 \end{Bmatrix} + \zeta_\mathbf{j} \begin{Bmatrix} P_{1:3}^{-1}\,x'_j \\ 0 \end{Bmatrix}. \tag{2.6}$$

The above description gives a detailed view of scene structure reconstruction and now we can see about camera motion reconstruction. When intrinsic parameters are known, the the normalized camera coordinates will be $x \leftarrow K^{-1}\,x$ and the relation between the corresponding image points will be given by,

$$\mathbf{x_i^t}\,\mathbf{E}\,\mathbf{x'_j} = \mathbf{0}, \tag{2.7}$$

where $E = [t]_\times\,R$ is the essential matrix. When both intrinsic and extrinsic parameters are known, then relation between the corresponding image points will be given by,

$$\mathbf{x_i^t}\,\mathbf{F}\,\mathbf{x'_j} = \mathbf{0}, \tag{2.8}$$

where $F = [e_r]_\times K_r R K_l^{-1}$ is the fundamental matrix matrix. Thus epipolar geometry can be described in several ways, depending on the amount of the prior knowledge about the stereo system. We can identify three general cases.

- If both intrinsic and extrinsic camera parameters are known, we can describe the epipolar geometry in terms of the projection matrices and can solve the reconstruction problem unambiguously.
- If only the intrinsic parameters are known, we can describe the epipolar geometry in the essential matrix and solve the reconstruction problem up to an unknown scale factor by estimating the extrinsic parameters.
- If neither intrinsic nor extrinsic parameters are known the epipolar geometry is described by the fundamental matrix and we can still solve the reconstruction problem but only up to an unknown, global projective transformation of the world.

Thus if a set of image corresponding points are given without the position, angular attitude and calibration of the cameras then this situation is referred to as weak calibration, and the epipolar geometry is described by the fundamental matrix and the scene may be reconstructed up to a projective ambiguity.

If three images of a scene are available, and point correspondences are known across all three views, then the above linear equations can be extended to three images and is called trifocal tensor. In case of n views, one has many more images of a scene and the projective reconstruction of n views can be done by factorization methods. The solution to the reconstruction problem may only be determined up to an unknown projective transformation T, applied both to points and cameras. A projective transformation for the model of 3D space containing world points is given by,

$$\mathbf{X} \mapsto \mathbf{T}\,\mathbf{X}, \tag{2.9}$$

where T is a non-singular 4×4 matrix representing a mapping between homogeneous coordinates. From this relationship, it is seen that the determination of camera matrices P_i and points X_j cannot be unique, given only corresponding image coordinates x_{ij} . Consider

$$\begin{aligned} \mathbf{x_{ij}} &= \mathbf{P_i}\,\mathbf{X_j} \\ &= (\mathbf{P_i}\,\mathbf{T^{-1}})(\mathbf{T}\,\mathbf{X_j}) \\ &= \mathbf{P_i'}\,\mathbf{X_j'}. \end{aligned} \tag{2.10}$$

In this relationship, new points $X'_j = T\,X_j$ are defined in terms of points X_j , and similarly new camera matrices $P'_i = P_i\,T^{-1}$ in terms of the camera matrices P_i. Since both (P_i, X_j) and (P'_i, X'_j) give rise to the same projected image coordinates x_{ij} , there is no way to choose between these two solutions to the reconstruction problem. In fact, there exists a complete family of solutions to the problem, corresponding to all possible choices of the matrix T. All such solutions are related to each other by the application of a projective transformation, and are hence called projectively equivalent. The above analysis does not rule out the possibility that other solutions to this reconstruction problem exist, not related to a particular obtained solution by any projective transformation. The standard factorization algorithms are Tomasi-Kanade factorization [30] and Sturm-Triggs [22] methods. The link between projective depth estimation and projective reconstruction of cameras and points was noted by Sturm and Triggs (1996), whereby it is shown that given the true projective depths, camera matrices and points can be found from the factorization of the data matrix weighted by the depths. Let us consider the set of projective depths λ_{ip} packed into a matrices, $3m \times n$ rescaled measurement matrix W that has rank at most 4. The matrix W can be factorized into a $3m \times 4$ matrix of projection multiplying a $4 \times n$ matrix of points, and this factorization corresponds to a valid projective reconstruction of n views using Singular Value Decomposition. This is given by the equation,

$$\begin{aligned} \mathbf{W} \equiv & \begin{bmatrix} \lambda_{11}\, x_{11} & \lambda_{12}\, x_{12} & \dots & \lambda_{1n}\, x_{1n} \\ \lambda_{21}\, x_{21} & \lambda_{22}\, x_{22} & \dots & \lambda_{2n}\, x_{2n} \\ \dots & & & \dots \\ \lambda_{m1}\, x_{m1} & \lambda_{m2}\, x_{m2} & \dots & \lambda_{mn}\, x_{mn} \end{bmatrix} \\ = & \begin{bmatrix} P_1 \\ P_2 \\ \dots \\ P_m \end{bmatrix} \begin{bmatrix} X_1 & X_2 & ... & X_n \end{bmatrix}. \end{aligned} \tag{2.11}$$

2.2 Collineation

The transformation between projectively equivalent matrix is computer as collineation. The projective reconstruction is done by chaining partial reconstructions that are obtained using a 6-points procedure described in [14] or Sturm-Triggs iteration described in [22]. After the process of reconstruction, triplets of projection matrices are obtained, which are related to the correct (Euclidean) one by a collineation of the 3D space. Thus the true transformation can be computed by a method called synchronization, which will be discussed in the later chapters. The collineation between n sets of reconstructed matrices can be obtained only if they are projectively equivalent matrices. Let us consider two sets of projective frames to understand the process. A reference projective frame is fixed, that is the one associated to the first triple $\{P_a, P_b, P_c\}$; subsequent triples with an overlap of two like $\{P'_a, P'_b, P'_d\}$ can be brought to the same frame by computing the proper collineation T. So let P_a and P_b be the same camera in two different projective frames, i.e., P'_a and P'_b represents the same camera in two different triplets. They are related by an unknown collineation T:

$$\mathbf{P_i}\,\mathbf{T} \simeq \mathbf{P'_i}. \tag{2.12}$$

In the next step, the elements of the matrices are reshaped to vector column-wise arrangement. This is denoted by introducing the *vec* operator. Thus the equation can be rewritten as,

$$\mathbf{vec}(\mathbf{P_i}\,\mathbf{T}) \simeq \mathbf{vec}(\mathbf{P'_i}). \tag{2.13}$$

Let us consider a and b two vectors of $\mathbb{R}^n$ be $vec(P'_i)$ and $vec(P_i T)$. Their equality up to a scale can be written as: $rank[a;b] = 1$. This is to say that all minors of [a; b] are zero. There are n(n-1)/2 of such order-two minors, and they can be obtained by multiplication of b by a suitable $n(n-1)/2 \times n$ matrix that contains the entries of a. Let us call this matrix $[a]_\times$ in analogy to the $\mathbb{R}^3$ case, where equality up to a scale reduces to $a \times b = 0$. Since by construction, a belongs to the null-space of $[a]_\times$, its rank is at most n - 1. Hence $a \simeq b$ gives rise to the linear system of n(n-1)/2 equations $[a]_\times b = 0$ where only n-1 of them are independent. The matrix $[a]_\times$ is composed by n-1 blocks arranged by rows. The the i^{th} block B_i has (n - i) rows and n columns (i = 1,...,n - 1):

$$\mathbf{B_i} = \begin{bmatrix} 0_{1\times(i-1)} & -a_{i+1} & a_i & 0 & 0 & \dots & 0 \\ 0_{1\times(i-1)} & -a_{i+2} & 0 & a_i & 0 & \dots & 0 \\ 0_{1\times(i-1)} & -a_{i+3} & 0 & 0 & a_i & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & & \dots \\ 0_{1\times(i-1)} & -a_n & 0 & 0 & 0 & \dots & a_i \end{bmatrix} \tag{2.14}$$

$$[\mathbf{a}]_\times = \begin{bmatrix} B_1 \\ B_2 \\ B_{n-1} \end{bmatrix} \tag{2.15}$$

Thus equality of two vectors $vec(P'_i)$ and $vec(P_i T)$ of $\mathbb{R}^n$ up to a scale can be written as,

$$[\mathbf{vec}(\mathbf{P'_i})]_\times\, \mathbf{vec}(\mathbf{P_i}\,\mathbf{T}) = \mathbf{0}. \tag{2.16}$$

Using the properties of the Kronecker product, the unknown vec(T) can be separated from the previous equation as,

$$[\mathbf{vec}(\mathbf{P'_i})]_\times\, (\mathbf{I_{4\times 4}} \otimes \mathbf{P_i})\, \mathbf{vec}(\mathbf{T}) = \mathbf{0}. \tag{2.17}$$

Since the coefficient matrix has rank at most 11, at least two camera matrices are needed to stack-up the 15 equations required to compute the 4 x 4 matrix T up to scale. As said earlier, this is the reason why the projective reconstruction needs triples of cameras with an overlap of two matrices.

Algorithm 1: Transformation

1 **Objective**

Given set of projective reconstructed matrices $\{P_a^i, P_b^i$ and $P_c^i\}$ and $\{P_a^j, P_b^j$ and $P_d^j\}$, if two of the pairs of the sets are projectively equal then their transformation can be computes as T.

2 **Algorithm**

(i) The projectively equal matrices are converted to vector by reshaping $vec(P_i')$ and $vec(P_i\,T)$ from the equation, $\mathbf{vec(P_i\,T)} \simeq \mathbf{vec(P_i')}$.

(ii) Let $a = vec(P_i')$ and $b = vec(P_i\,T)$, then $[a]_\times$ is computed by,

$$\mathbf{B_i} = \begin{bmatrix} 0_{1\times(i-1)} & -a_{i+1} & a_i & 0 & 0 & \dots & 0 \\ 0_{1\times(i-1)} & -a_{i+2} & 0 & a_i & 0 & \dots & 0 \\ 0_{1\times(i-1)} & -a_{i+3} & 0 & 0 & a_i & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & & \dots \\ 0_{1\times(i-1)} & -a_n & 0 & 0 & 0 & \dots & a_i \end{bmatrix} ; \;\; [\mathbf{a}]_\times = \begin{bmatrix} B_1 \\ B_2 \\ B_{n-1} \end{bmatrix}.$$

(iii) The transformation T is made into null space by rearranging the equation as, $[\mathbf{vec(P_i')}]_\times\,(\mathbf{I_{4\times 4}} \otimes \mathbf{P_i})\,\mathbf{vec(T)} = \mathbf{0}$.

(iv) Single value decomposition is used and the eigenvector with smallest eigenvalue is taken as $vec(T)$.

(v) Reshaping the $vec(T)$ into a 4×4 matrix will give the final transformation T.

Chapter 3

Synchronization

In this chapter, we consider the method of synchronization, related work which is previously done in the literature and finally the extended work done in this thesis. This chapter also gives a brief explanation about graph theory and relates the idea of graph theory to synchronization method.

3.1 Synchronization

This chapter deals with the problem of global {cameras|images |sensors |frames} network {orientation|motion}. The network is made of cameras (or images, in Photogrammetry) and cameras are sensors, of course; in abstract what matters is that a 3D reference frame (3 orthogonal axis with an origin) is attached to the sensor|camera|image that represent its position and angular attitude. These two properties are collectively referred to as "orientation" in Photogrammetry or "motion" in Computer Vision, if one consider discrete samples of a moving camera. Mathematically orientation|motion is described by elements of the group of direct isometries, SE(3) (a.k.a. special Euclidean group). Therefore the goal is to recover location and/or attitude of a bunch of cameras|images|sensors organized in network. The links of this network (edges of a graph) are relative measures of one cameras|images|sensors with respect to (some of) the others. The "global" adjective means that we are interested in solutions that consider all the measurements at once, as opposed to "incremental" approaches that grow a solution by adding pieces iteratively (such as resection-intersection in the context of structure-from-motion).

3.1.1 Related Work

This thesis work extends the work of Malapelle et al [19] (computing parallax maps from monocular and uncalibrated video sequences) and Arrigoni et al [3] (estimating camera motion in the context of structure-from-motion). Synchronization is a topic that connects with many field like computer vision, sensor networks, automatic controls, robotics, graph theory (parallel rigidity), topography/surveying. There are two different type of branch address this problem.

The threading methods treat the problem of finding an initialization of points as a searching algorithm and generally determine a reference image from the set of images. The coordinate frame of reference image acts as the projective frame. More complex threading methods are based on finding paths in the topology graph. The corner stone of all threading methods is the cost they impose on edges and paths: Kang et al. accumulate the residual error [16], Marzotto et al. impose constraints on the ratio of overlapping area and residual error [21] for Structure from Motion (SfM).

Batch approaches find a global solution by linearly approximating the motion model. For instance, Govindu uses such an approach for SfM in order to build a linear system by quaternions [12] or Lie algebra [13]. Trying to solve a similar SfM problem, Sturm [29] using homographies as input to compute the initial rotation and translation for nonlinear optimization, then factorization method, followed by averaging known rotations.

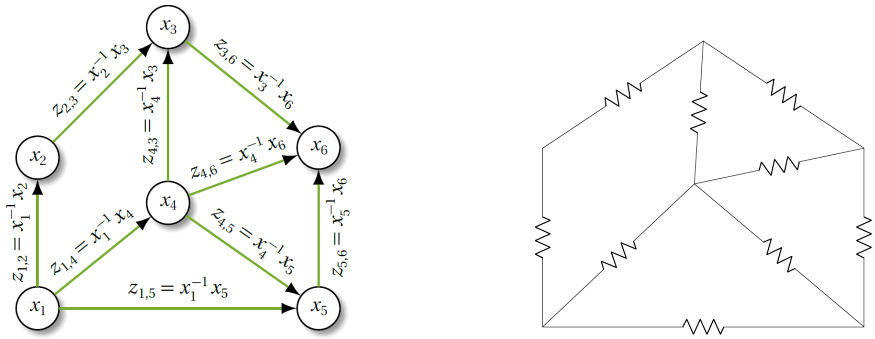
Several instances of synchronization have been studied in the literature:

- When considering $\Sigma = \mathbb{Z}_2$, we have sign synchronization [7].
- When considering $\Sigma = R$, we have time synchronization [11, 17] from which the term "synchronization" was originated.
- When considering $\Sigma = R^d$, we have state / translation synchronization [25, 32].
- When considering $\Sigma = SO(d)$, we have rotation synchronization [1,5,6,10,15,18,20,27,28] and also known as "rotation averaging", which give rise to the Group synchronization problem.
- When considering $\Sigma = SE(d)$, we have rigid-motion synchronization [2, 4, 13, 24, 31, 33] and also known as "motion averaging", which can be seen as a stage of structure-from-motion that starting from the epipolar graph to find globally consistent orientations or motions for the cameras or images.
- When considering $\Sigma = SL(d)$, we have homography synchronization [26].
- When considering $\Sigma = \mathbb{S}_d$, we have permutation synchronization [23].

3.2 Graph Theory

Let Σ be a group with unit element 1_Σ and $\vec{G} = (V, E)$ be a finite simple digraph, with $n = |V|$ vertices and $m = |E|$ edges. A Σ-labelled graph is a digraph with a labelling of its edge set by elements of Σ, that is a tuple $\Gamma = (V, E, z)$ where $z : E \to \Sigma$ is a labelling of the edges such that if $(u, v) \in E$ then $(v, u) \in E$ and $z(v, u) = z(u, v)^{-1}$. Hence, we may often consider G, the undirected version of $\vec{G}$. The cycle $v_1v_2, v_2v_3 \dots v_\ell v_1$ in Γ is null cycle if and only if,

$$\mathbf{z(v_1, v_2) \cdot z(v_2, v_3) \cdot \ldots \cdot z(v_\ell, v_1) := z(v_1v_2, v_2v_3, \ldots v_\ell v_1) = 1_\Sigma.} \tag{3.1}$$



(a) a (b) b

Figure 3.1: (a) Graph model and (b) Kirchhoff's voltage law similar to graph model

The model of the graph is shown in the (Fig. 3.1a). Let $\Gamma = (G, z)$ be a Σ-labelled graph for $G = (V, E)$ and $x : V \to \Sigma$ be a vertex labelling. Then x is a consistent labelling if and only if for each edge $e = (u, v) \in E$ we have

$$\mathbf{x(v) = x(u) \cdot z(e) \iff z(e) = x(u)^{-1} \cdot x(v).} \tag{3.2}$$

The consistency error of $\tilde{x}$ is defined as,

$$\epsilon(\tilde{\mathbf{x}}) = \sum_{\mathbf{(u,v) \in E}} \mathbf{f(\tilde{z}(u, v) \cdot z(u, v)^{-1}).} \tag{3.3}$$

where $\tilde{z}$ is the edge labeling induced by $\tilde{x}$: $\tilde{z}(u, v) = \tilde{x}(u)^{-1} \cdot \tilde{x}(v)$. The labelled graph Γ has a consistent labeling if and only if it contain only null cycles. In such case the graph is called balanced or synchronized. The structure of the consistent labeling graph is similar to the electrical analogy with the Kirchhoff's voltage law. The directed sum of the electrical potential differences (voltage) around any closed network is zero which is shown in the (Fig. 3.1b). The edges with outlying labels are solved using the graph theory from Marek Cygan [[9] and [8]].

3.3 Synchronization in $(\mathbb{C}^{d\times d})$

Synchronization method is extended to $\mathbb{C}^{d\times d}$ in the thesis work. Let us assume now that we are given a symmetric function $f : \Sigma \to \mathbb{C}^{d\times d}$ with a unique minimum at 1_Σ and $f(1_\Sigma) = 0$. Suppose that Σ is a group which admits a matrix representation through $d \times d$ matrices (i.e., Σ can be embedded in $\mathbb{C}^{d\times d}$), where the group operation reduces to matrix multiplication and $1_\Sigma = I_d$. Let $U \in \mathbb{C}^{dn\times d}$ be the vector block matrices of X_u containing the vertex labels and $Z \in \mathbb{C}^{dn\times dn}$ be the block matrix $T_{u,v}$ containing the edge labels. Thus two matrices U and Z respectively, which are matrices composed of $d \times d$ blocks as,

$$\mathbf{U} = \begin{bmatrix} X_1^{-1} \\ X_2^{-1} \\ \dots \\ X_n^{-1} \end{bmatrix}, \quad \mathbf{U^{-b}} = \begin{bmatrix} X_1, X_2, \dots X_n, \end{bmatrix}, \quad \mathbf{Z} = \begin{bmatrix} I & T_{1,2} & \dots & T_{1,n} \\ T_{2,1} & I & \dots & T_{2,n} \\ \dots & & & \dots \\ T_{n,1} & T_{n,2} & \dots & I \end{bmatrix} \tag{3.4}$$

For a complete graph, the consistency constraint rewrites

$$\mathbf{Z} = \mathbf{UU^{-b}}. \tag{3.5}$$

If the graph G is not complete, Z is not fully specified, or equivalently, it has zero entries in correspondence of missing edges, whereas UU^{-b} is fully specified, hence we shall write the constraint as

$$(\mathbf{Z} - \mathbf{UU^{-b}}) \circ (\mathbf{A} \otimes \mathbb{1}_{\mathbf{d\times d}}) = \mathbf{0}. \tag{3.6}$$

where A is the (0-1) adjacency matrix of G (with zero diagonal) and $\circ$ is the Hadamard product. The previous equation can be rewritten as

$$\mathbf{Z_A} = (\mathbf{UU^{-b}}) \circ (\mathbf{A} \otimes \mathbb{1}_{\mathbf{d\times d}}).. \tag{3.7}$$

were $Z_A = Z \circ A$ represent the matrix of the available measures with zero entries in correspondence of missing edges (and along the diagonal). The cost function of the synchronization is given by,

$$\epsilon(\mathbf{U}) = \|(\mathbf{Z} - \mathbf{UU^{-b}}) \circ (\mathbf{A} \otimes \mathbb{1}_{\mathbf{d\times d}})\|_\mathbf{F}^\mathbf{2}. \tag{3.8}$$

The Hadamard product with A mirrors the summation over the edges of E in the definition of the consistency error.

3.3.1 Noiseless case

Let us consider the "noiseless" case, i.e. $\epsilon = 0$, and let us start assuming that the graph is complete, Since $U \in \mathbb{C}^{dn\times d}$ then we have $U^{-b}U = nI$. The rank of U is d and the equation is given by,

$$\mathbf{Z} = \mathbf{UU^{-b}} \iff \underset{\wedge\ \mathrm{rank}(\mathbf{Z})=\mathbf{d}}{\mathbf{ZU} = \mathbf{nU}}. \tag{3.9}$$

the solution U is the eigenvectors of Z associated to eigenvalues n. Since Z have rank d, all the other eigenvalues are zero, so n is the largest eigenvalues of Z.

3.3.2 Missing edges case

In the case with missing edges, the graph is not complete and the adjacency matrix A plays an important role in solving the system,

$$\mathbf{Z_A} = (\mathbf{UU^{-b}}) \circ (\mathbf{A} \otimes \mathbb{1}_{\mathbf{d\times d}}). \tag{3.10}$$

The global symmetric matrix is given by,

$$\mathbf{Z_A} = \begin{bmatrix} I & A_{1,2}\,T_{1,2} & \dots & A_{1,n}\,T_{1,n} \\ A_{2,1}\,T_{2,1} & I & \dots & A_{2,n}\,T_{2,n} \\ \dots & & & \dots \\ A_{n,1}\,T_{n,1} & A_{n,2}\,T_{n,2} & \dots & I \end{bmatrix} \tag{3.11}$$

where,

$$\mathbf{A_{i,j}} = \begin{cases} 1, & \text{if } \mathrm{T_{i,j}} \text{ is known.} \\ 0, & \text{otherwise.} \end{cases} \tag{3.12}$$

The adjacency matrix A gets "inflated" by the Kronecker product with $\mathbb{1}_{d\times d}$ to match the block structure of the measures. we can rewrite the equation as,

$$\mathbf{Z_A} = (\mathbf{UU^{-b}}) \circ (\mathbf{A} \otimes \mathbb{1}_{\mathbf{d\times d}}) = \mathrm{blkdiag}(\mathbf{U})(\mathbf{A} \otimes \mathbb{1}_{\mathbf{d\times d}})\,\mathrm{blkdiag}(\mathbf{U^{-1}}). \tag{3.13}$$

which implies that

$$\mathbf{Z_A U} = \mathrm{blkdiag}(\mathbf{U})(\mathbf{A} \otimes \mathbb{1}_{\mathbf{d\times d}})\,\mathrm{blkdiag}(\mathbf{U^{-1}})\mathbf{U} = \mathrm{blkdiag}(\mathbf{A1})\mathbf{U} = (\mathbf{D} \otimes \mathbf{I})\mathbf{U}. \tag{3.14}$$

where $D = \mathrm{diag}(A\mathbf{1})$ is the degree matrix of the graph ($A\mathbf{1}$ is the sum of the rows of A). If $\epsilon = 0$, the solution U is the eigenvectors of $(D\otimes I_d)^{-1}Z_A$ associated to the largest d eigenvalues of $(D \otimes I_d)^{-1}Z_A$ which are equal to 1's.

$$\begin{aligned} (\mathbf{D} \otimes \mathbf{I_{d\times d}})^{-1}\mathbf{Z_A} &= (\mathbf{D} \otimes \mathbf{I_{d\times d}})^{-1}(\mathbf{Z} \circ (\mathbf{A} \otimes \mathbb{1}_{\mathbf{d\times d}})) \\ &= (\mathbf{D} \otimes \mathbf{I_{d\times d}})^{-1}\,\mathrm{blkdiag}(\mathbf{U})(\mathbf{A} \otimes \mathbb{1}_{\mathbf{d\times d}})\,\mathrm{blkdiag}(\mathbf{U^{-1}}) \\ &= \mathrm{blkdiag}(\mathbf{U})(\mathbf{D} \otimes \mathbf{I_{d\times d}})^{-1}(\mathbf{A} \otimes \mathbb{1}_{\mathbf{d\times d}})\,\mathrm{blkdiag}(\mathbf{U^{-1}}) \\ &= \mathrm{blkdiag}(\mathbf{U})[(\mathbf{D^{-1}A}) \otimes \mathbb{1}_{\mathbf{d\times d}}]\,\mathrm{blkdiag}(\mathbf{U^{-1}}) \end{aligned} \tag{3.15}$$

Hence $(D \otimes I_d)^{-1}Z_A$ and $(D^{-1}A) \otimes \mathbb{1}_{d\times d}$ are similar, i.e., they have the same eigenvalues. $(D^{-1}A)$ is the transition matrix of the graph which has 1's as the largest eigenvalues. Thus if the graph is connected, $(D^{-1}A) \otimes \mathbb{1}_{d\times d}$ also has 1's as the largest eigenvalues.

3.3.3 Noisy case

When noise is present, the minimum consistency error is not 0 and the eigen solutions are the eigenvectors corresponding to λ_1,λ_2,...,λ_d as the largest eigenvalues of d. But due to noise $(D \otimes I_d)^{-1} Z_A$ and $(D-1A) \otimes \mathbb{1}_{d\times d}$ are not similar and so $(D^{-1}A) \otimes \mathbb{1}_{d\times d}$ has $\lambda < 1$ with multiple d as the largest eigenvalues.

Algorithm 2: Synchronization

1 **Objective**

Given the projective transformation between all the possible combination T_{ij} which is constructed to the global matrix Z, the decomposition with give the true transformation U.

2 **Algorithm**

(i) For each projective transformation T_{ij}, the matrices are arranged in the global symmetric matrix $Z \in \mathbb{C}^{dn\times dn}$.

$$Z = \begin{bmatrix} I & T_{1,2} & \dots & T_{1,n} \\ T_{2,1} & I & \dots & T_{2,n} \\ \dots & & & \dots \\ T_{n,1} & T_{n,2} & \dots & I \end{bmatrix}$$

(ii) The adjacency matrix A is computed as,

$$A_{i,j} = \begin{cases} 1, & \text{if } \mathrm{T_{i,j}} \text{ is known.} \\ 0, & \text{otherwise.} \end{cases}$$

(iii) The matrix Z and A are converted to sparse and checked for the complete connection of network.

(iv) The global symmetric matrix Z is updated by adding the information of A as,

$$Z_A = \begin{bmatrix} I & A_{1,2}\,T_{1,2} & \dots & A_{1,n}\,T_{1,n} \\ A_{2,1}\,T_{2,1} & I & \dots & A_{2,n}\,T_{2,n} \\ \dots & & & \dots \\ A_{n,1}\,T_{n,1} & A_{n,2}\,T_{n,2} & \dots & I \end{bmatrix}$$

(v) $(D \otimes I_{d\times d})^{-1}$ is computed where D is the sum of rows of A

(vi) Eigen decomposition for sparse matrix $(D \otimes I_{d\times d})^{-1} Z_A$ is computed and eigen values with maximum of 4 are taken as U.

Chapter 4

Algorithm

In this chapter, the development of the synchronization of projective frames algorithm is described in detail. The theoretical and practical relation between projective reconstruction, transformation and synchronization are composed in pseudo-code and flow chart. Finally we represent the overall procedure for the proposed algorithm to be implemented. Later, the work is motivated with the idea of application for static and dynamic cameras.

4.1 Algorithm structure

4.1.1 Methodology

A set of projective reconstructions between triples of cameras are computed from the correspondence points as from Chapter 2. Each partial reconstruction is determined up to an unknown projective transformation. The main goal of the work is to compute such unknown projective transformations to bring all the partial reconstruction into the same reference system. Let $P_s = 3 \times 4$ be the projective matrix of camera s in a world reference system where we have $s = 1, 2, .., n$. Then, the projective matrix of camera s in partial reconstruction i is given by $P_s^i = 3 \times 4$. The relation between P_s^i and P_s is given by the equation,

$$\mathbf{P_s^i} = \mathbf{P_s T_i}, \tag{4.1}$$

where, $T_i = 4 \times 4$ is the unknown projective transformation that maps the world reference system into partial reconstruction i. From the different combinations of partial reconstruction we can obtain the relation between transformation T as,

$$\mathbf{P_s^i} = \mathbf{P_s T_i}; \mathbf{P_s^j} = \mathbf{P_s T_j} \Rightarrow \mathbf{P_s^j} = \mathbf{P_s^i T_i^{-1} T_j}, \tag{4.2}$$

and,

$$\mathbf{T_{ij}} = \mathbf{T_i^{-1} T_j}, \tag{4.3}$$

here the projective transformation that maps reconstruction i into reconstruction j is given by T_{ij}. T_{ij} can be computed from the equality of 2 vectors up to scale, assuming that reconstructions i and j share 2 cameras, this is discussed in detail in the previous Chapter 2. The matrix of projective transformation is inverted when the direction is reversed.

$$\mathbf{T_{ij}} = \mathbf{T_{ji}^{-1}}. \tag{4.4}$$

As discussed in the chapter 3 synchronization the graph $G = (V, E)$ is constructed. Each vertex corresponds to a partial reconstruction of 3 views and it is labeled with an known transformation T_i. Each edges corresponds to an overlap of 2 cameras between 2 partial reconstructions, and it is labeled with a unknown transformation T_{ij}. The goal is to estimate vertex labeling, given on edge labeling to estimate T_i given from a redundant set of T_{ij}. It is solved by eigenvalue decomposition method and this method is known as synchronization. Once T_i is known, several projective matrix of the same camera can be computed by,

$$\mathbf{P_s} = \mathbf{P_s^i T_i^{-1}} \, \textbf{and} \, \mathbf{P_s} = \mathbf{P_s^j T_j^{-1}}. \tag{4.5}$$

From many P_s matrices, a final single P_s matrix is obtained by a simple matrix arithmetic mean and this method is known as single averaging.

4.1.2 Flow chart

The algorithm is designed with different stages to fit the system. The input is given by the set of images and output is obtained by the real projective matrices of the images. The flow chart is shown in the (Fig. 4.1)

- In the stage one (Pre-processing), for each images 100 feature points are extracted with strong corner using Harris corners method. All the sets of feature points are combined into group of all combination of triplets which is easy for doing the projective reconstruction. Point matching and RANSAC(algorithm detailed in Appendix B) algorithm are used to compute the correspondence points of the triplets.

- In the stage two (reconstruction), the projective reconstruction is done from 3 views that are obtained using a 6-points procedure described in [14] or Sturm-Triggs iteration described in [22].

- In the stage three (transformation), the transformation of the projective matrices is computed with 2 overlapping matrices. The transformation matrix is normalized my dividing with the 4th root of the determent to make the determent 1, this forces the vertices of the graph for a closed solution. Since the projective transformation have both complex and negative eigen values, the normalized transformation matrix is complex. Then these matrices are arranges in the global matrix as Z.

- In the stage four (synchronization), the sparse eigen decomposition is computed with the highest 4 eigen values are taken as U. To remove the complex values, U is divided by the first block $X1$ with 4×4 matrix. By this way, the vector in the complex plane is arranged in the same direction with different scaling. Now the scale is removed by using least square method and the complex terms are removed.
- In the stage five (single averaging), the final computed transformation is multiplied with each sets of reconstructed matrices, an arithmetic mean is made to get the real projective matrix.

4.2 Applications

4.2.1 Dynamic camera

An approach based in dynamic motion of camera is analyzed here. Using Structure-From-Motion method, with images or videos as input data advanced sensor fusion techniques can be done with enhanced knowledge of orientation and tie-points computing epipolar or trifocal geometry. More robust methods results in relative rotations and translations computation by solving a motion synchronization problem can be studied by braking into rotation synchronization or translation synchronization or solved both in one step. The global methods can be seen as an effective and efficient way of computing path planning in robotics and 3D reconstruction using the projective reconstructed matrices.

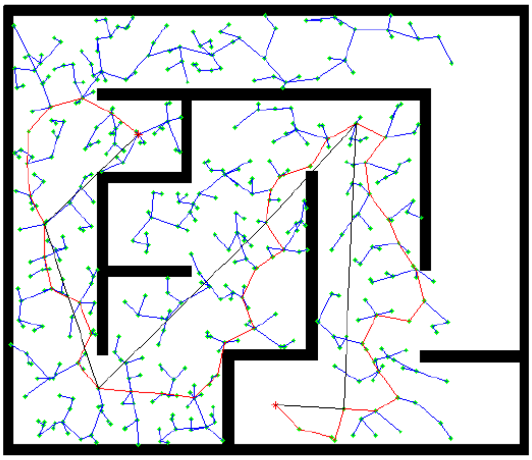


Figure 4.2: Path planning in robotics

4.2.2 Static camera

An approach based in static camera is analyzed here. Many real time application based on surveillance camera for security purposes. Multiple eyed devices which is similar to the eyes of

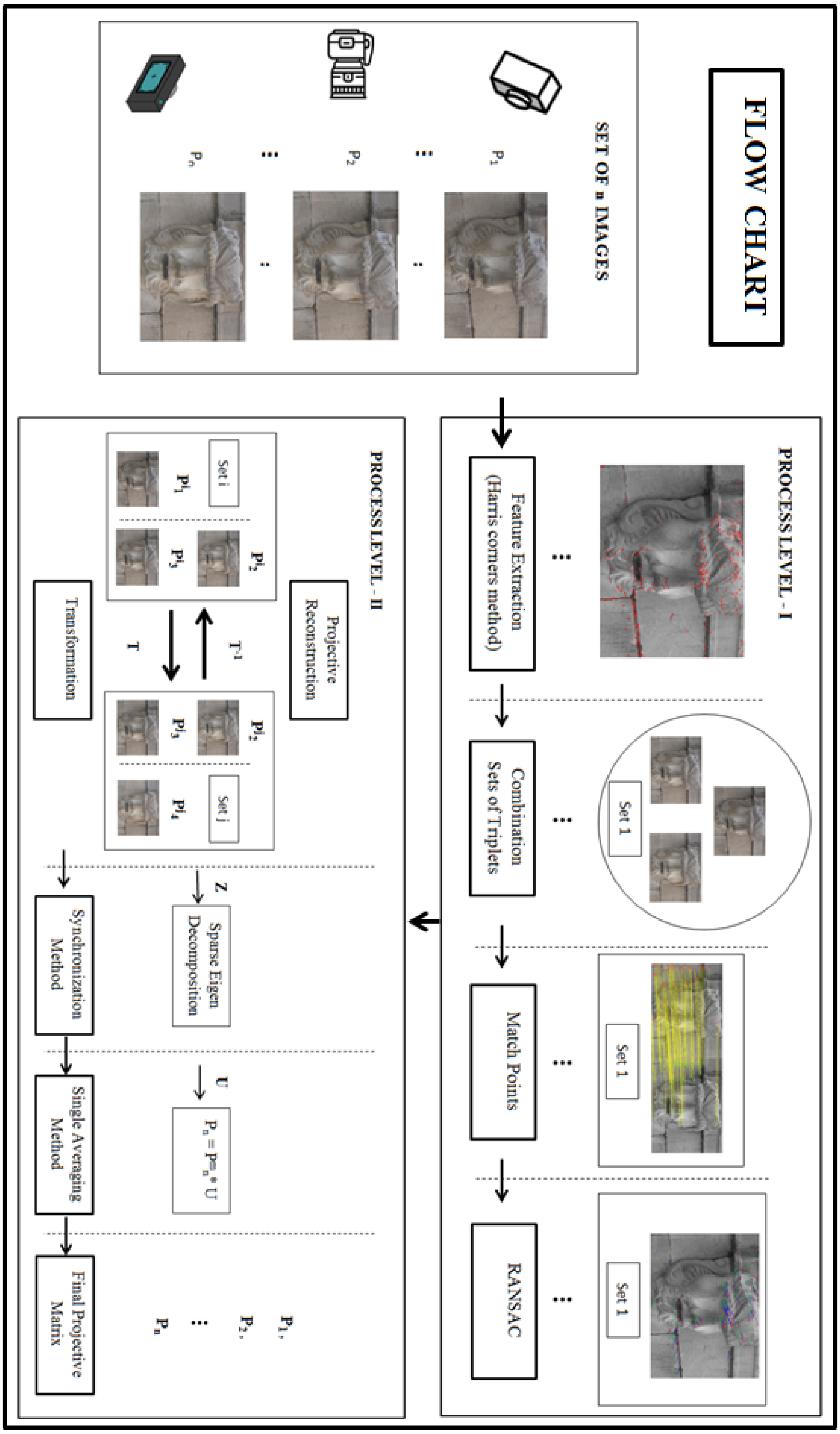


Figure 4.1: Flow chart diagram

house flies, has the best application to the method. Devices like 360 degree cameras,etc can be easily synchronized. Virtual reality techniques can be done with the application like video games and live streaming system.

Figure 4.3: Eyes of house flies

Algorithm 3: Synchronization of Projective Frames

1 **Objective**

Given set of n uncalibrated images with both internal and external parameters of the camera are unknown. The goal is to compute the global synchronized projective matrices $\hat{P}_1$,$\hat{P}_2$...$\hat{P}_n$ of all images n.

2 **Algorithm**

(i) For each images, feature points are extracted using Harris corners method.
(ii) All the sets of feature points are combined into group of all combination of triplets. $\binom{n}{k} = {}^nC_k = \frac{n!}{k!(n-k)!}$
(iii) For each set of triplet, pairs of points set are used to get the correspondences of 2D points $\{ x_i \leftrightarrow x_i' \leftrightarrow x_i'' \}$ using matching point function and RANSAC fitting fundamental matrix function.
(iv) A threshold value is set for i in complete sets $\{ x_i \leftrightarrow x_i' \leftrightarrow x_i'' \}$ for selecting sets of triplets as to reconstruct projective matrices we need atleast 6 points.
(v) From the selected triplets, projective reconstruction is made as P_1^j,P_2^j and P_3^j, where j is the number is selected triplets.
(vi) The Transformation function is computed for the reconstructed matrices set as nodes with T for two overlapping condition, I identity for same matrices and 0 zeros for one overlapping condition. Thus the global matrix Z is computed.
(vii) The synchronization method is used to compute U.
(Viii) Compute the projective matrices after transformation. $\hat{P}_n = P_n^m * U$
(ix) Single averaging method is used to get the final Projective matrices $\hat{P}_1$,$\hat{P}_2$...$\hat{P}_n$

Chapter 5

Experiments & Results

In this chapter, we begin by describing the experimental setup for the proposed method. Then the results of applying the method on the synthetic data is presented. For a unique setup, we present a method to determine the change in the error to the ground truth value to understand the system based on the noisy environment.

5.1 Setup

An implementation of the proposed methods in Matlab served for the experiments. We generated 100 normally distributed 3D points around the camera center. From camera setup, we take a random radius of a sphere and the camera position is made to slide on the surface with different internal parameters. A small change in the distance between the adjacent image is make to make random translation. Then from the camera setup, we also randomly adjust rotations to simulate different views for the camera. Thus our setup is made similar to a real camera mounted on a tripod which has all the 3 degree of freedom for rotation and translation. From those projected views and with the help of 3D points, we generate the image points for each views. There is a percentage of missing points randomly selected. Later, this setup is experimented by adding image noise for understanding the algorithm better.

5.2 Error Methods

Calculating the error is one of the important in the process. The difference between the ground truth and the obtained value determines the error. We use three methods to compute the error values,

5.2.1 Algebraic error

The algebraic error is the simplest method to determine the error in the probabilistic scale. The method uses a simple Frobenius norm between the ground truth and the value obtained.

The error is also called as residual error as it shows the minimized cost function. But this error method does not give further information, it is used simply to determine the range of the algorithmic difference.

5.2.2 Transformation error

Transformation error is used to determine the error between the sets of reprojected matrices before and after the synchronization process with the ground truth set. This method also uses norm between T and T' (before and after the synchronization) which is calculated as described in the chapter 2.

5.2.3 Reprojection error

In this method, the error involves estimating a correction for each correspondences of two images. The process is done by comparing the difference between the real 2D points and computed 2D points using the reconstructed projective matrices and transformed 3D points. Then the difference between these correspondences points will give the error in the pixel range.

5.3 Synthetic experiments

We ran experiments on different levels of image noise $[10^{-3} : 10]$ added to the image in pixel range. Each experiment is repeated with 100 random runs with different setup for the data collection. We used a setup composed of 20 different cameras for each setup. After collecting data from different samples a simple mean shows the rate of change in reprojection error with respect to noise. Here the experiment is tested by using all points in the visible range and partial visible range. The graph (Fig. 5.1) shows the progressive error of full visible points after 100 iteration of random samples. The "full visible points graph" uses a full measurement matrix (all points are seen in all cameras) and thus includes also Sturm's method [22]. Clearly we can see that the synchronization method cannot compete with Sturm's with a full matrix. The next step is to compare with partial visible points. The "partial visible points graph" (Fig. 5.2) refers to a more realistic situation where only some points are visible in each image. In this case Sturm's cannot be used and the benefits of synchronization can be appreciated over Minimum spanning tree (A MST is a subset of the edges of a connected, edge-weighted undirected graph (not necessarily connected) that connects all the vertices together, without any cycles and with the minimum possible total edge weight and so it is a spanning tree whose sum of edge weights is as small as possible) for moderate image noise. More (future) work is needed to understand and fix the behaviour with large noise.

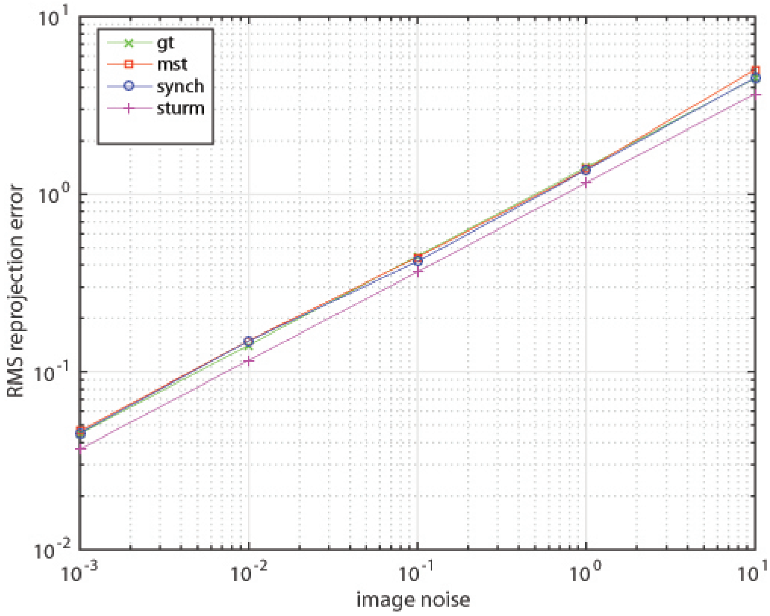


Figure 5.1: Full visible points

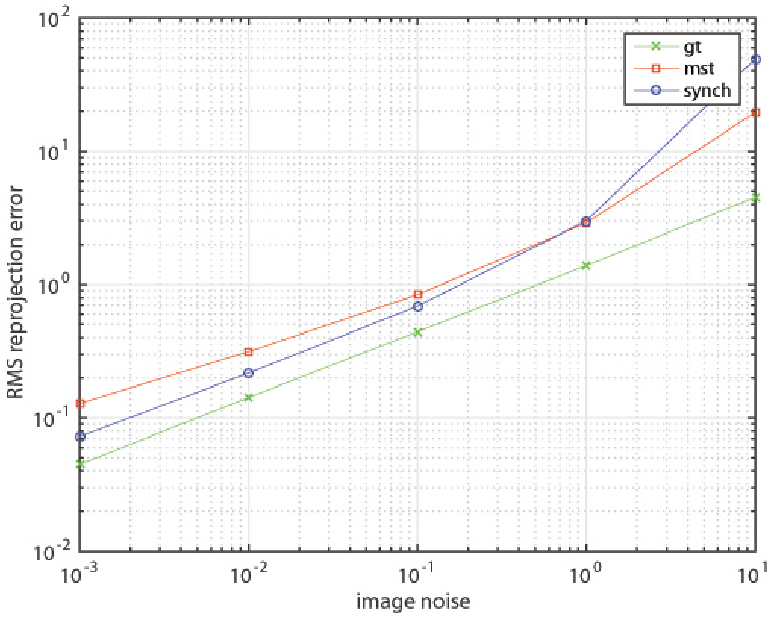


Figure 5.2: Partial visible points

Chapter 6

Conclusions & Future Work

In this chapter, we conclusions about the thesis with the obtain results, give an overview of the research work and the limitation of the methods. Along, with a discussion about possible future work.

6.1 Conclusion

This thesis research is about the extended work for the synchronization problem, which has been utilized for the projective reconstruction. Our method serves as a common base that can be used for projective reconstruction problem with global interface. The work also gives a description about computing the projective reconstruction, transformation and synchronization in complex form with the working of the algorithm using synthetic data. Both the theoretical and practical ideas discussed shows the robustness of the application with the algorithm.

The main feature of the algorithm is that the method is easier to implement on different hardware and software environment with less power as the proposed analytical methods intended to find a solution without an initialization and multiple iterative process. We tested this method with its performance to the maximum limits of behavior under the influence of noise and missing data.

The limitation occur when there is not enough of triplets and if the graph is not completely connected then there is a drift in the synchronization process. We also have the limitation in the process of single averaging as some time one or more projective matrices of similar group fall far away from the original reconstructed matrix . Even with these limitations, from the experimental results showed that this approach is working well and output with satisfactory results.

6.2 Future Work

The future work can be extended based on the theoretical and practical approach. The further work are listed as follows,

- Based on the focus on the reconcilement of fundamental or essential matrices between pair of views and the working on the viewing graph.
- Explore the analogy between computing epipolar scales and the reconcilement of essential matrices, and try to extend it to fundamental matrices.
- In the case of essential matrices the scales are needed to bring the reconcilement into a synchronization problem, so the varying scale can be determined.
- The method based on tracking the 3D points from the final projective matrix, this way the relation between the image points and the projective matrices can be analyzed.
- An alternative idea of using the iterative methods, in the synchronization and the single averaging method.
- This algorithm can be improved and implemented in the real time applications for the devices that are embedded in automobile systems for tracking and navigation.

Appendix A

Transformation

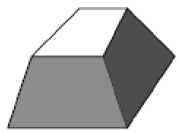
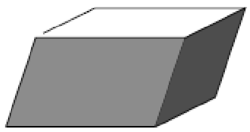
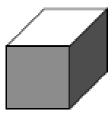
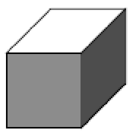
Group	Matrix	Distortion	Invariant properties
Projective 15 dof	$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^\mathsf{T} & v \end{bmatrix}$		Intersection and tangency of surfaces in contact. Sign of Gaussian curvature.
Affine 12 dof	$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$		Parallelism of planes, volume ratios, centroids. The plane at infinity, π_∞.
Similarity 7 dof	$\begin{bmatrix} s\mathtt{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$		The absolute conic, Ω_∞.
Euclidean 6 dof	$\begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$		Volume.

Figure A.1: Projective Transformations (Image from [14])

Appendix B

RANSAC

Algorithm 4: RANSAC Algorithm

Input : data - a set of observations
model - a model that can be fitted to data
n - the minimum number of data required to fit the model
k - the number of iterations performed by the algorithm
t - a threshold value for determining when a datum fits a model
d - the number of close data values required to assert that a model fits well to data

Output: best_model - model parameters which best fit the data (or nil if no good model is found)
best_consensus_set - data points from which this model has been estimated
best_error - the error of this model relative to the data

```
iterations := 0 ;  best_model := nil;  best_consensus_set := nil;  best_error := infinity
while iterations <k do
    maybe_inliers := n randomly selected values from data
    maybe_model := model parameters fitted to maybe_inliers
    consensus_set := maybe_inliers
    for every point in data not in maybe_inliers do
        if point fits maybe_model with an error smaller than t then
            add point to consensus_set
    if the number of elements in consensus_set is >d then
        (this implies that we may have found a good model, now test how good it is)
        this_model := model parameters fitted to all points in consensus_set
        this_error := a measure of how well this_model fits these points
        if this_error <best_error then
            (we have found a model which is better than any of the previous ones, keep it
             until a better one is found)
            best_model := this_model ; best_consensus_set := consensus_set ;
            best_error := this_error
        increment iterations
    return best_model, best_consensus_set, best_error
```

Bibliography

[1] M. Arie-Nachimson, S. Z. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, and R. Basri. Global motion estimation from point matches. *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on. IEEE, 2012.*, 2012. [Cited on page 10.]

[2] F. Arrigoni, A. Fusiello, and B. Rossi. Spectral synchronization of multiple views in se(3). *SIAM Journal on Imaging Sciences*, 2016. [Cited on page 10.]

[3] Federica Arrigoni, Andrea Fusiello, and Beatrice Rossi. Camera motion from group synchronization. *3D Vision (3DV), 2016 Fourth International Conference*, 2016. [Cited on page 9.]

[4] F. Bernard, J. Thunberg, P. Gemmar, F. Hertel, A. Husch, and J. Goncalves. A solution for multi-alignment by transformation synchronisation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*, 2015. [Cited on page 10.]

[5] Sharp G. C., Lee S. W., and Wehe D. K. Multiview registration of 3d scenes by minimizing error between coordinate frames. In *eccv*, pages 587–597, 2002. [Cited on page 10.]

[6] A. Chatterjee and V. M. Govindu. Efficient and robust large-scale rotation averaging. In *iccv*, 2013. [Cited on page 10.]

[7] M. Cucuringu. Synchronization over z2 and community detection in signed multiplex networks with constraints. *Journal of Complex Networks*, page 469506, 2015. [Cited on page 10.]

[8] Marek Cygan, Fedor V. Fomin, ukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Micha Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer International Publishing, ISBN:9783319212746, 2015. [Cited on page 11.]

[9] Marek Cygan, Stefan Kratsch, Marcin Pilipczuk, and Magnus Wahlstrm. Clique cover and graph separation: New incompressibility results. *ACM Transactions on Computation Theory (TOCT)*, 2014. [Cited on page 11.]

[10] Arrigoni F., Magri L., Rossi B., Fragneto P., and Fusiello A. Robust absolute rotation estimation via low-rank and sparse matrix decomposition. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 491–498, 2014. [Cited on page 10.]

[11] A. Giridhar and P.R. Kumar. Distributed clock synchronization over wireless networks: Algorithms and analysis. *Proceedings of the IEEE Conference on Decision and Control*, pages 4915–4920, 2006. [Cited on page 10.]

[12] Govindu. Combining two-view constraints for motion estimation. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE*, 2001. [Cited on page 10.]

[13] V. M. Govindu. Lie-algebraic averaging for globally consistent motion estimation. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. Vol. 1. IEEE*, pages 684–691, 2004. [Cited on page 10.]

[14] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. [Cited on pages iii, 6, 16, and 25.]

[15] R.I. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *International Journal of Computer Vision*, 2013. [Cited on page 10.]

[16] E.-Y. Kang, I. Cohen, and G. Medioni. A graph-based global registration for 2d mosaics. *ICPR*, 2000. [Cited on page 10.]

[17] Richard Karp, Jeremy Elson, Deborah Estrin, and Scott Shenker. Optimal and global time synchronization in sensor nets. Technical report, Center for Embedded Networked Sensing: University of California, Los Angeles, 2003. [Cited on page 10.]

[18] Wang L. and Singer A. Exact and stable recovery of rotations for robust synchronization. *Information and Inference: a Journal of the IMA*, 2(2):145–193, 2013. [Cited on page 10.]

[19] Francesco Malapelle, Andrea Fusiello, Beatrice Rossi, Emiliano Piccinelli, and Pasqualina Fragneto. Uncalibrated dynamic stereo using parallax. *Image and Signal Processing and Analysis (ISPA), 2013 8th International Symposium*, 2013. [Cited on page 9.]

[20] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. IEEEr*, 2007. [Cited on page 10.]

[21] R. Marzotto, A. Fusiello, and V. Murino. High resolution video mosaicing with global alignment. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004*, 2004. [Cited on page 10.]

[22] Sturm P and Triggs B. factorization based algorithm for multi-image projective structure and motion. *ECCV, Springer-Verlag*, page 709720, 1996. [Cited on pages 6, 16, and 21.]

[23] Deepti Pachauri, Risi Kondor, and Vikas Singh. Solving the multi-way matching problem by permutation synchronization. *Advances in Neural Information Processing Systems 26*, pages 1860–1868, 2013. [Cited on page 10.]

[24] D. M. Rosen, C. DuHadway, and J. J. Leonard. A convex relaxation for approximate global optimization in simultaneous localization and mapping. In *icra*, pages 5822 – 5829, 2015. [Cited on page 10.]

[25] W.J. Russel, D.J. Klein, and J.P. Hespanha. Optimal estimation on the graph cycle space. *IEEE Transactions on Signal Processing*, 59(6):2834 – 2846, 2011. [Cited on page 10.]

[26] P. Schroeder, A. Bartoli, P. Georgel, and N. Navab. Closed-form solutions to multiple-view homography estimation. *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 650–657, Jan 2011. [Cited on page 10.]

[27] A. Singer. Angular synchronization by eigenvectors and semidefinite programming. *Applied and Computational Harmonic Analysis*, 30(1):20 – 36, 2011. [Cited on page 10.]

[28] A. Singer and Y. Shkolnisky. Three-dimensional structure determination from common lines in cryo-em by eigenvectors and semidefinite programming. *SIAM Journal on Imaging Sciences*, 4(2):543 – 572, 2011. [Cited on page 10.]

[29] P. Sturm. Algorithms for plane-based pose estimation. *Computer Vision and Pattern Recognition, 2000. CVPR 2000*, 2000. [Cited on page 10.]

[30] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization approach. *International Journal of Computer Vision*, page 137154, 1992. [Cited on page 6.]

[31] A. Torsello, E. Rodola, and A. Albarelli. Multiview registration via graph diffusion of dual quaternions. *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE*, pages 2441 – 2448, 2011. [Cited on page 10.]

[32] R. Tron, L. Carlone, F. Dellaert, and K. Daniilidis. Rigid components identification and rigidity enforcement in bearing-only localization using the graph cycle basis. In *IEEE American Control Conference*, 2015. [Cited on page 10.]

[33] R. Tron and R. Vidal. Distributed 3-D localization of camera sensor networks from 2-D image measurements. *IEEE Transactions on Automatic Control*, 59(12):3325–3340, 2014. [Cited on page 10.]