CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

Departamento de Computación

Solución simultánea de varios subproblemas de Visión por Computadora

Que presenta

Heriberto Cruz Hernández

Para obtener el grado de

Doctor en Ciencias en Computación

Director de la tesis:

Dr. Luis Gerardo de la Fraga

Ciudad de México **Enero, 2019**

CENTRO DE INVESTIGACIÓN Y DE
ESTUDIOS AVANZADOS DEL INSTITUTO
POLITÉCNICO NACIONAL

Campus Zacatenco

Computer Science Department

Joined solution of several Computer Vision subproblems

A dissertation presented by

Heriberto Cruz Hernández

as the fulfillment of the requirement for the degree of:

Ph.D. in Computer Science

Advisor:

Dr. Luis Gerardo de la Fraga

Mexico City **January, 2019**

Resumen

El problema de reconstrucción 3D consiste en obtener el modelo 3D de una escena a partir de sus proyecciones en múltiples imágenes. En la literatura, este problema se ha tratado con la composición en otras cinco subtareas: la extracción de características, la búsqueda de correspondencias, la calibración de la cámara, la estimación de la pose y la triangulación. Estas subtareas se han estudiado como problemas aislados, de tal manera que el resultado de la reconstrucción 3D depende de qué tan bien se resuelve cada una de ellas. En este trabajo, estudiamos la solución simultánea de los subproblemas de reconstrucción 3D. Nuestro objetivo es explotar la información de subproblemas no contiguos suponiendo que la información utilizada para resolver una subtarea puede usarse para resolver otras subtareas. En el primer estudio, analizamos la auto calibración simultanea de la cámara, la estimación de la pose y la recuperación del modelo como un problema de optimización no lineal. Resolvemos los tres subproblemas utilizando como entrada las correspondencias de puntos en tres imágenes. Analizamos las condiciones mínimas necesarias para resolver simultáneamente los tres sub-problemas, propusimos un método para incluir restricciones de paralelismo y la ortogonalidad de líneas sobre el modelo. Logramos reconstruir modelos con conjuntos de datos sintéticos, pero también reales. En el segundo estudio, investigamos la extracción de características y la búsqueda de correspondencias de manera simultánea. Manejamos el problema compuesto a través del estudio Tipo de Orden, una característica invariable combinatoria del campo de geometría computacional. Mostramos la estabilidad del tipo de orden durante el proceso de generación de imágenes y como con el Tipo de Orden, es posible realizar una identificación automática de conjuntos de puntos en $\mathbb{R}^2$. Además, proponemos un método para clasificar los Tipos de Orden con respecto a su robustez al ruido y un método para identificar los tipos de orden que son adecuados para realizar la coincidencia de puntos. A partir de nuestro estudio, proponemos un nuevo tipo de marcadores fiduciales que nos sirven para realizar reconstrucción 3D y para realizar realidad aumentada. Los nuevos marcadores basados en el Tipo de Orden nos permiten resolver la extracción de características y la correspondencia de las mismas en distintas imágenes. La salida de este enfoque la usamos como entrada para aplicar nuestro otro enfoque donde ya obtenemos la reconstrucción 3D. Finalmente unimos los dos enfoques antes mencionados para resolver el problema completo de reconstrucción. La reconstrucción la realizamos usando los enfoques simultáneos, por lo que la reconstrucción se realiza en dos etapas. Primero la extracción y correspondencia de características de manera simultánea y finalmente en la segunda etapa, la solución simultánea a la auto calibración, estimación de poses y obtención del modelo. Con los resultados de esta tesis, contribuimos con un nuevo tipo de marcadores, un método nuevo y flexible para realizar la reconstrucción 3D y también con un nuevo enfoque para manejar la coincidencia de características, no solo para la reconstrucción 3D sino también para otras aplicaciones de Visión por Computadora.

Abstract

The 3D reconstruction problem consists in obtaining the 3D model of a scene from its projections in multiple images. In literature, the task has been treated as the composition of other sub-tasks, i.e., the feature extraction, feature matching, camera calibration, pose estimation, and triangulation. The 3D reconstruction sub-tasks are studied as isolated problems in such a manner that the result of the 3D reconstruction depends on how well the pipeline of tasks is solved. In this work, we study the simultaneous solution of the 3D reconstruction sub-problems. We aim to exploit the information of non-contiguous sub-problems by assuming that the information used to solve a sub-task can be used for solving other sub-tasks. In the first study, we study the simultaneous camera self-calibration, pose estimation and model retrieving as a non-linear optimization problem. We solve the three sub-problems using as input the point correspondences in three images. We analyzed the minimal necessary conditions to solve the problem, we proposed a method to include physical constraints for lines parallelism and orthogonality to solve the camera self-calibration, and we are able to reconstruct models with synthetic but also real datasets.

In the second study, we researched the joined features extraction and the features matching of points on a plane. We handled the composed problem through the study of Order Type, a combinatorial invariant feature from the computational geometry field. We show the order type stability during the image generation process, and how with Order Type, it is possible to perform automatic identification of point sets in $\mathbb{R}^2$. Furthermore, we propose a method to rank the Order Types regarding robustness to noise and a method to identify those Order Types that are suitable to perform point matching. From our study, we propose a new kind of fiducial markers. These new markers allow us to solve the features extraction and matching through multiple images. The two problems are solved simultaneously, using the new markers based in order type to automatically solve the point matching, and using this result perform the camera self-calibration the poses estimation and the model retrieval on three images.

Finally we joined the two proposed approaches to solve the complete 3D reconstruction problem. We perform the reconstruction in two stages. In the first, we perform the simultaneos features extraction and matching. In the second stage, we solve simultaneously the self-calibration, the pose estimation and the model retrieval.

With the results of this thesis we contribute with a new kind of fiducial markers, a new and flexible method to perform the 3D reconstrucion, and also a new approach to handle the feature matching, not only for 3D reconstruction but also for other Computer Vision applications.

Agradecimientos

A Dios, por la vida y por darme la oportunidad de cumplir mis metas. Sé que son muchos los obstáculos para poder estudiar un doctorado y también sé que sin la ayuda de Dios yo no estaría aquí.

A mi esposa Cinthya, por apoyarme a lograr este sueño, aunque a priori sabíamos que iba a traer consigo algunas dificultades. Muchas gracias amor. Busco el momento de aprovechar los conocimientos y las habilidades que adquirí para traer bienestar a nuestra familia.

A mis padres, Margarita y Pablo, por su apoyo incondicional a lo largo de mi vida en todos los aspectos y en especialmente a mi madre; porque sin tu apoyo esto no hubiera sido posible.

A mis hijas, Alexandra y Helena, por recordarme en todo momento la importancia de la alegría y por ser siempre mis motivaciones para la vida.

A mi asesor, el Dr. Luis Gerardo de la Fraga, por aceptarme como su alumno de maestría y de doctorado. Por su paciencia y por su dedicación como director de tesis. Sé que aún tengo mucho que aprender, pero siempre tendré en mente todo lo que me ha enseñado y las experiencias que pude tener bajo su dirección para buscar la eficiencia y la productividad. Espero algún día poder pagarle algo de todo el tiempo y esfuerzo que ha invertido en mí. Mil gracias.

A los doctores Sonia G. Mendoza Chapa, Adriana Lara López, Juan Manuel Ibarra Zannatha, Cuauhtémoc Mancillas López, y Carlos Artemio Coello Coello por formar parte del comité de evaluación de mi trabajo de tesis.

A Sofi, por el trato más amable que uno puede esperar. Por todo el apoyo moral, la confianza y por hacerme sentir que siempre hay alguien a quien puedo acudir desde mi ingreso al Cinvestav. Gracias Sofi.

A mis compañeros y amigos de la sala de doctorado por dejarme compartir con ustedes la aventura del doctorado. Por todos los consejos, por los conocimientos compartidos y por su apoyo.

A todos los profesores del departamento, por sus valiosas enseñanzas y por ser excelentes modelos a seguir.

Al CONACyT por el apoyo económico que recibí siempre de manera puntual. Sin duda, esa beca fue un factor decisivo para que pudiera llevar a cabo mis estudios de maestría y doctorado. Muchísimas gracias.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In Computer Vision, the 3D reconstruction problem is the process of retrieving the 3D structure of a scene using only its projections from distinct viewpoints. Projections, in this case, are bidimensional images taken by a camera. The solution to this problem has several applications in science and industry. Robots perform 3D reconstruction to know the structure of their environment [2, 3]. The design and video games industries obtain the 3D models of real objects in order to ease the generation of virtual scenes [4]. Medical imaging benefits from the 3D reconstruction by allowing the physicians to convert images taken from inside the human body in manipulable 3D models [5, 6]. In geosciences, aerial images are converted to 3D models to ease the study and analysis of terrains; it allows to obtains measurements like volumes and distances from the reconstructed models [7, 8].

The 3D structure is composed of the orientations and locations of the objects present in the scene; these involve the model in front of the camera but also the camera in itself. The task starts by receiving as input the set of images of a scene. The problem can be seen as the composition of other subproblems or steps in the following pipeline [9, 10]:

1) **Feature extraction:** detection and extraction of salient features in the images. Commonly, these features are points at corners, since they are elements that can be detected in different images taken from distinct viewpoints.

2) **Point matching:** detection of the same features in multiple images.

3) **Camera calibration:** estimation of the internal camera parameters. These internal parameters define how the original scene is projected to the images.

4) **Pose estimation:** estimation of the orientations and locations of the cameras and objects in a global coordinate system.

5) **Model estimation:** estimation or reconstruction of the original objects in the scene.

In literature, the subproblems above are integrated into a sequential pipeline, such that the output of each step is used as input of the next. In most of the existing literature works, the subproblems are seen as isolated tasks that require specific information to be solved [10, 11, 12, 13]. The information used in each step is often considered only for a specific subproblem, and once the problem is solved, that information is supposed to be depreciable for the other subproblems. This relationship among subproblems leads the final solution to be affected by errors accumulation, which is one of the most critical issues in the field [14, 15, 16]. In order to overcome this issue, some works [10, 11, 12, 13] propose to refine the solutions periodically or after a particula event, e.g., after the model estimation. The solutions refinement is performed by local optimization algorithms, i.e., non-linear least squares which use the current solution and take it to a better one by the minimization of a cost function. Due to the local nature of the refinements, when the current solution is not close enough to the global optimum, it could happen that refinement results are not consistent with the ground truth solution.

In this work, we study the solution to the 3D reconstruction subproblems by exploiting the information available in non-contiguous subproblems in the 3D reconstruction pipeline to help solve others separately or simultaneously. In our work we studied the simultaneous solution of the camera calibration, pose estimation, and model retrieval. We also studied mechanisms to take the information among several 3D reconstruction subproblems; in this sense, we propose a projective invariant feature for points on $\mathbb{R}^2$ which allow us to relate multiple subproblems. We propose to use this projective invariant as a descriptor for Computer Vision with many potential applications for automatic identification, textureless point matching and pose estimation. We investigate the use of the proposed descriptor in the context of the 3D reconstruction, but we also studied the descriptor itself. We studied alternative representations for de descriptor with the aim to enhance its use for the pose estimation problem, and we propose a method to optimize set of points on the plane in such a manner that its associated descriptor becomes more robust to noise. Our aim with this projective invariant is to work as a tool to communicate different non-contiguous 3D reconstruction pipeline subproblems.

1.1 Our work

The work presented in this thesis is developed in different directions. We have proposed three main studies that have as objective to exploit the information available in some of the 3D reconstructions subproblems to solve other non-contiguous steps of the general 3D reconstruction pipeline. In the following paragraphs, we summarize the three developed approaches. For each approach, we highlight the subproblems that are involved/connected, the information that is exploited and the general idea of them.

Self-calibration, pose estimation, and model reconstruction using three images. In the first study, we propose to solve simultaneously three of the most critical subproblems of the general pipeline: the camera self-calibration, the pose estimation, and the model estimation. We set up the task as a single objective optimization problem. Given as input only matched features in three views we find the camera intrinsic parameters, the poses for three images, and the reconstructed model simultaneously. An illustration of the involved steps is shown in Fig. 1.1. These parameters, as well as the location of the features on the images, are enough information to obtain a reconstruction of the 3D model. The idea is to apply a global search algorithm to find the decision variables vector that minimizes the reprojection error (see Sec. 2.9) of the three images. With this approach, we were able to reconstruct synthetic and real-world datasets. Additionally, we studied the minimal conditions to solve the problem, i.e., we studied the kind of movement that the images require (rotation and translation), the minimal number of images features required, and the stop criteria for the Differential Evolution algorithm, which is the heuristic we used to solve the optimization problem. Details of this approach and a complementary study about the inclusion of scene restrictions for lines parallelism and orthogonality in the problem statement are presented in the Chapter 3.

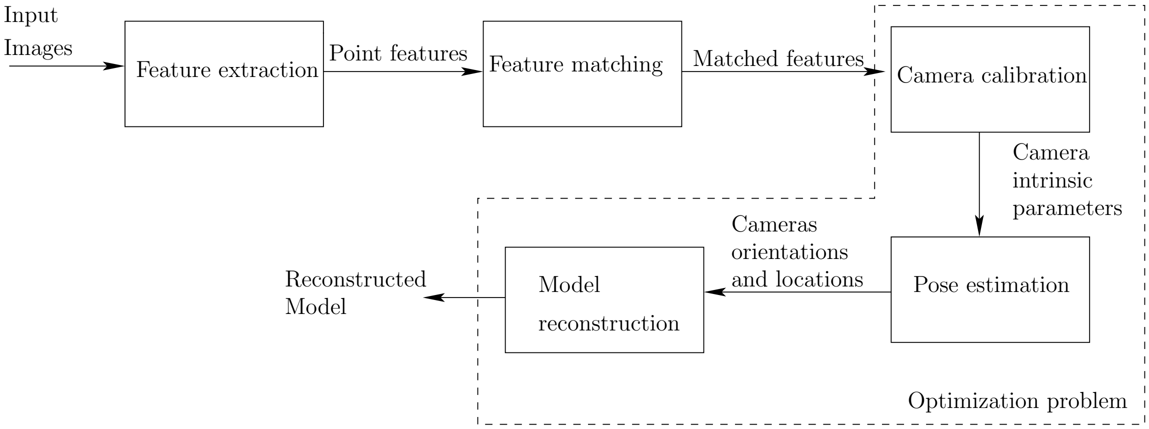


Figure 1.1: First study and the involved subproblems.

Identification of a point set based on order type With our second study, we aim to search for a geometrical feature that maintains unchanged through different steps of the 3D reconstruction general flow, i.e., an invariant, specifically a projective invariant. We explored the use of Order Type (OT) [1], a combinatorial invariant for points in $\mathbb{R}^n$, with $n = 2$ for the case of points on the plane, to propose it as a projective invariant. We studied the stability of OT during the image generation process (see Sec.2.1) and we found OT as a very suitable feature to perform the identification of a set of points in $\mathbb{R}^2$ automatically.

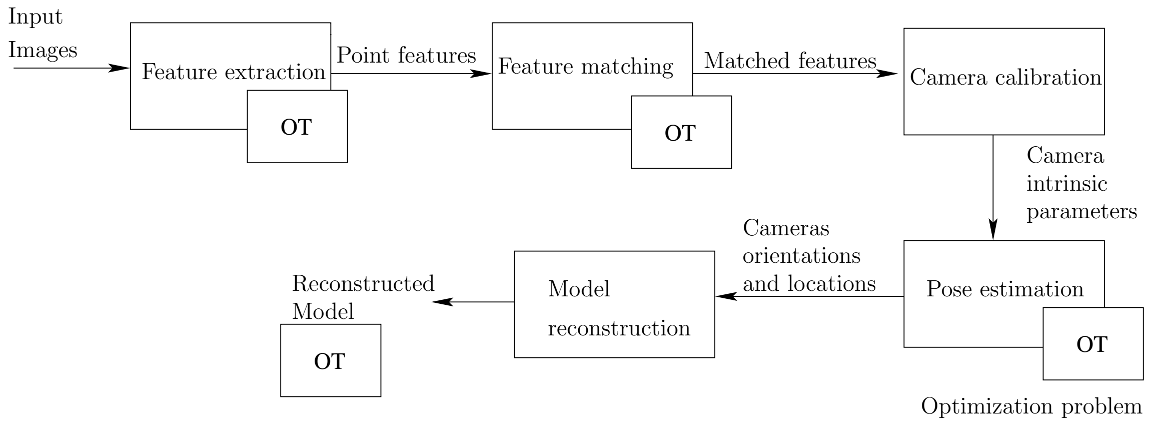


Figure 1.2: Second study scope. The OT and its invariance through subproblems of 3D reconstruction.

The idea with this study is to develop the mechanism that will allow us to communicate different 3D reconstruction subproblems. We wish to use the OT as a tool that will allow us:

- to assign a unique identification ID to a point set in $\mathbb{R}^2$,
- to identify the point sets among different images,
- to solve the feature matching subproblem.

As part of this study, we proposed a visual fiducial tag based in OT. First, we proposed the fiducial tag for automatic identification; we showed that a competitive number of different tags could be obtained using OT as the base, and additionally we analyze the OT invariance in the presence of noise. From this study we found that there are some OTs more robust to noise, and we propose a method to rank all possible OTs according to noise robustness. The scope of this study is shown in Fig. 1.2, for the case of planes, OTs can be automatically identified (it is kept unchanged) through the feature extraction, the point matching, and it has a relationship with the image generation process, the pose estimation, and the reconstructed model. Details of this study are presented in Chapter 4.

Point set matching with Order Type. In this third study, we analyze the OT for solving the point matching subproblem in Computer Vision. We aim to evaluate how suitable is OT for point set matching purposes. We analyze all the existing OTs in $\mathbb{R}^2$ for point sets with cardinality up to eight points, and we found that not all OTs can be used to solve the matching problem. We propose a method to identify those OTs that are suitable for this purpose, and we also provide the number of OTs suitable for this purpose. From this study, we found that a high percentage (90%) of all existing OTs can be used to solve the point matching subproblem. With the obtained results we complemented the proposed visual fiducial tag of our second study to add it the capability to solve the pose of the tag. To test our approach, we implemented an augmented reality application which entirely estimates the orientation of our fiducial tags from a single image. The subproblems that we relate through OT are the image generation process, the feature extraction, the point matching, the pose estimation, and the model estimation.

The idea is to use the OT for identification and also for point matching purposes. With the OT identification capability, we can characterize the point sets present in input images to build a dictionary of point sets, and this characterization can be used later to solve the matching subproblem. Since OT is based in the point set structure and not in textures, OT could also relate the model, which will allow us to identify and match point sets through images but also the model. We observe a good potential of OT for the 3D reconstruction since OT is not only defined for point sets in $\mathbb{R}^2$, it could also be used to solve the 3D the registration problem, which is the task of aligning two 3D point sets. With this study, we confirm that OT is suitable for point set matching and we confirm that OT has good potential for helping us to solve the 3D reconstruction. Details of this study are presented in Chapter 4.

1.2 Objectives

In the following, we state the objectives of this thesis.

1.3 General Objective

To design methods os strategies for solving simultaneously various subproblem in the 3D reconstruction problem by exploting the information of contiguous subproblems but also no contiguous.

Particular objectives

- Propose new approaches to exploit the information available in different 3D reconstruction subproblems.

- Propose methodologies for integrating knowledge in the 3D reconstruction process.

- Introduce new projective invariant features useful for solving the 3D reconstruction problem.

- Propose mechanisms to simplify 3D reconstruction subproblems based on partial prior knowledge.

1.4 Contributions

As a result of this thesis work, the following articles have been published:

JCR Journals

1. Heriberto Cruz-Hernández and Luis Gerardo de la Fraga. A fiducial tag invariant to rotation, translation, and perspective transformations. Pattern Recognition, 81:213 223, 2018 [17]. DOI: 10.1016/j.patcog.2018.03.024.

Other Journals

1. Heriberto Cruz Hernández and Luis Gerardo de la Fraga. Order type dataset analysis for fiducial markers. Data in Brief, 2018 [18]. DOI: 10.1016/j.dib.2018.08.126

In proceedings of international conferences

1. Heriberto Cruz Hernández and Luis Gerardo de la Fraga. A method to optimize the Order Type Maximal Perturbation through multiple single point displacements. In Numerical and Evolutionary Optimization – NEO 2018 (In preparation).

2. Luis Gerardo de la Fraga and Heriberto Cruz Hernández. Point set matching with order type. In José Francisco Martínez-Trinidad, Jesús Ariel Carrasco-Ochoa, José Arturo Olvera-López, and Sudeep Sarkar, editors, Mexican Conference on Pattern Recognition, pages 229–237, Springer Cham, 2018. [19]

3. Heriberto Cruz Hernández and Luis Gerardo de la Fraga. Fitting multiple ellipses with PEARL and a multi-objective genetic algorithm. In Leonardo Trujillo, Oliver Schütze, Yazmin Maldonado, and Paul Valle, editors, Numerical and Evolutionary Optimization – NEO 2017, pages 89–107, Springer Cham, 2018. [20].

4. Heriberto Cruz Hernández and Luis Gerardo de la Fraga. A Multi-objective Robust Ellipse Fitting Algorithm. In Maldonado, Yazmin and Trujillo, Leonardo and Schütze, Oliver and Riccardi, Annalisa and Vasile, Massimiliano, editors, NEO 2016: Results of the Numerical and Evolutionary Optimization Workshop NEO 2016, pages 141–158. Springer [21].

1.5 Structure of the document

Chapter 2: Presents the subproblems involved in the general 3D reconstruction problem; these are the problems we address and we aim to solve in a joined way.

Chapter 3: Presents the details for our first proposal that solves simultaneously the camera self-calibration, pose estimation and model reconstruction as an optimization problem.

Chapter 4: Describes our work for the development of automatic identification and matching for point sets based in Order Type for Computer Vision. This is the feature that we pretend to use to communicate various subproblems in the 3D reconstruction approach.

Chapter 5: In this Chapter, we add conceptually the solutions of the two proposals in Chapters 4 and 5 in a single frame. Points are detected automatically in an Order Type based fiducial marker, and these points are used to calculate the intrinsic camera parameters, the pose of three images, as well as the reconstructed model.

Chapter 6: Provides a summary of the presented work, the conclusions, and possible paths for future work.

Chapter 2

Computer Vision Subproblems

In this chapter, we look into the concepts that serve as the base for this thesis work. We aim to handle the 3D reconstruction problem, which can be seen as a pipeline of subproblems. Here we review each of these involved subproblems, their importance in the complete pipeline, the more popular approaches to solve them, and their main issues. In the 3D reconstruction problem, we want to retrieve the 3D scene from 2D images from distinct viewpoints, the images are the primary input for the problem, more specifically, we receive as input a sequence of images. Images are generated from the original scene with a digital camera, pictures are the result of a projection of the scene to the image plane situated in the camera projection plain, in the 3D reconstruction problem we aim to solve the inverse problem; thus, it becomes mandatory first to understand the image generation process and its related concepts.

2.1 Digital image generation process

In this section, we describe the projective transformation involved in the image generation process. In the physical sense, an image is generated through a series of physical and even chemical phenomenon that occur inside the camera. In Computer Vision, this transformation is modeled through the pinhole camera model, which is a fundamental concept for the development of this thesis. Assuming that the camera is fixed at the origin of a coordinate system, we can summarize the process in the following steps:

1. **Scene rotation and translation:** the scene is rotated and translated in front of the camera, this allows setting the point of view before the camera shot.

2. **3D points projection to the image plane:** the scene features, i.e., points, lines, planes, and 3D models are projected into the image. As s result of this projection, metric features as angles, distances, and parallelism are distorted by a projective transformation (they can not be retrieved back from a single image).

3. **Distortion:** The result of the projection in step two is altered by a radial distortion which is introduced by the convex nature of physical lenses. These effects are more perceptible in wide angle cameras.

4. **Pixel conversion:** due to the finite nature of the sensors in digital cameras, the result of the 3rd step is discretized to a limited image resolution. As result, we obtain pixels with integer coordinate positions.

The above transformations are modeled through the so-called pinhole camera model [22]. Given a point in a scene $\boldsymbol{P}_i = [x_i, y_i, z_i]^{\mathrm{T}}$, a reference point (the camera center) $\boldsymbol{o} = [0, 0, 0]^T$, and the image plane Π as shown in Fig. 2.1. A point on the image $\boldsymbol{p}_i = [u_i, v_i]^{\mathrm{T}}$ is the point where a line passing through $\boldsymbol{P}_i$ and $\boldsymbol{o}$ intersects with the image plane Π.

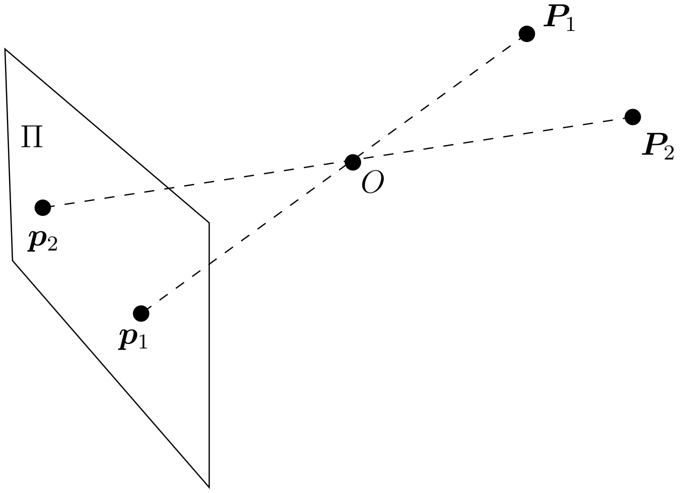


Figure 2.1: Projection of two scene points $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ to the image plane Π to obtain the image points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$.

The pinhole camera model explains the relationship between the points in the scene and their projection on an image. This relationship involves the internal characteristics of the camera, as well as the relative movement (rotation and translation) between the camera and the scene. The pinhole camera model is presented in Eq. (2.1).

$$\lambda \boldsymbol{p} = M\boldsymbol{P}, \tag{2.1}$$

where $M \in \mathbb{R}^{3\times 4}$ is known as the camera matrix, $\boldsymbol{P} = [x, y, z, 1]^{\mathrm{T}}$ and $\boldsymbol{p} = [u, v, 1]^{\mathrm{T}}$ are the scene point and its image projection, respectively, represented in homogeneous coordinates, and $\lambda \in \mathbb{R}$ is a scale factor. M can be decomposed as $M = K[R|\boldsymbol{t}]$. $K \in \mathbb{R}^{3\times 3}$ holds the internal/intrinsic characteristics of the camera, and it is defined as:

$$K = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.2}$$

where f_x and f_y define the focal distance in the scale of the x and y axis respectively, i.e., the distance between the camera center O and the image plane Π. When $f_x = f_y$, we say we have squared pixels. s is the skew, which is not zero when the x and y axes are not perpendicular. $[u_0, v_0]^{\mathrm{T}}$ is the principal point, which is the point where a line perpendicular to Π passes through the camera center O and intesects Π.

$[R|\boldsymbol{t}]$ is an augmented matrix that holds the external/extrinsic information about the camera, i.e., the pose, the orientation and relative location of the scene and the camera. The matrix $R \in \mathbb{R}^{3\times 3}$ is a rotation matrix which depends only of three parameters. A possible parametrization of R is: $R(\theta_1, \theta_2, \theta_3) = R_z(\theta_3)R_y(\theta_2)R_z(\theta_1)$ where R_y and R_z are rotation matrices around the principal z and y axes defined as:

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

and

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}.$$

The $\boldsymbol{t} = [t_x, t_y, t_z]^{\mathrm{T}}$ vector holds the information about the relative location or displacement of the scene with respect to the camera center.

In the following sections, we review the subproblems involved in the 3D reconstruction pipeline for this thesis work. First, we describe the feature extraction (Sec. 2.2), then we introduce feature matching (Sec. 2.3), then we look the camera calibration (Sec 2.4) and its variants (Sec 2.5), later the pose estimation (Sec. 2.6), and finally the triangulation (Sec. 2.7).

2.2 Feature extraction

In the context of 3D reconstruction, this subproblem consists in identifying the image salient features that are easily detectable through different images in the input sequence. It is expectable that these features correspond to scene salient characteristics seen from distinct viewpoints in the images. If one scene feature is seen from at least two images, with enough information, then it can be triangulated to obtain a 3D model point, which is one of the targets of our original problem. This is the reason why the feature extraction subproblem is the first and of crucial importance for the 3D reconstruction pipeline.

In the literature, it has been shown that the most convenient feature, due to its simplicity and easy detection, are the salient points, also known in the literature as corners. The salient points have no dimension and no area; thus they are invariant to rotation and scale. Even more, the projection of a point in the 3D scene results in a point on the image what is very convenient for 3D reconstruction purposes.

For the points feature extraction we find extensive literature. In [23] Lucas and Kanade propose to detect as interest points the center of those image regions where a correlation matrix A computed for a point and its neighborhood has maximal uncertainty. In [24] Harris and Stephens proposed an enhanced method which selects as features those elements with the maximum values for the value: $r = \det(A) - \alpha trace(A)^2 = \lambda_1 \lambda 2 - \alpha(\lambda_1 + \lambda_2)$, with $\alpha = 0.06$. The Harris and Stephens method is commonly known as the Harris corner detector, its results are illustrated in Fig. 2.2. Other more recent variations of this approach are the proposed by Triggs in [25] and Brown et al. in [26].

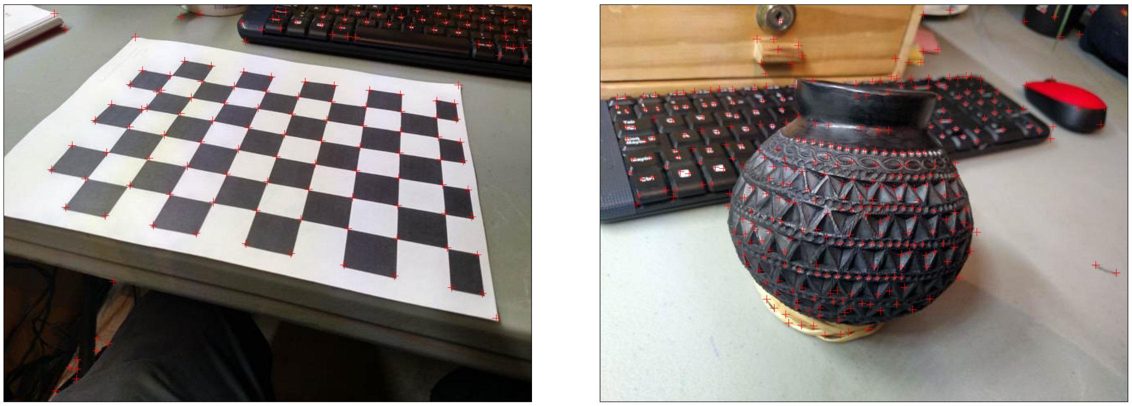


Figure 2.2: An Illustrative example of the features detection. The results are obtained through the Harris corner detector. The features are located at the intersection of two salient image lines.

Although the use of point is a common approach, the features could be any geometric artifact as lines, circles, or ellipses. In these alternative approaches, the task is to identify multiple instances, the most salient, of the model feature. For the case of lines, circles, and ellipses, the identification is performed by discretizing the parameters model space. As the result of the discretization, we obtain the parameters for a significant but finite number of models, a model for each combination of parameters, which is known as the Hough space [27]. After the discretization, each pixel in the image $\boldsymbol{p}$ is analyzed to identify those models in the Hough space that passes through $\boldsymbol{p}$. The Hough space is a finite list of possible models and the count of pixels $\boldsymbol{p}$ associated to each model. The models in Hough space with the higher pixel count are those that can be considered as present in the image. In Fig. 2.3, we show an instance of an image with points and the detected features through the Hough transform using lines as the model, additionally we show the corresponding Hough space.

The methods mentioned above only compute the locations of the features. Another approach is to compute the point as well as a descriptor (a vector that summaries the characteristics of the point neighborhood texture) [27]. State of the art in this other approach is the SIFT method [28] and its variations [29, 30]. The SIFT approach constructs a descriptor by analyzing the neighborhood of a corner. It defines a neighborhood of 16×16 pixels around an interest point, and it estimates the gradient for

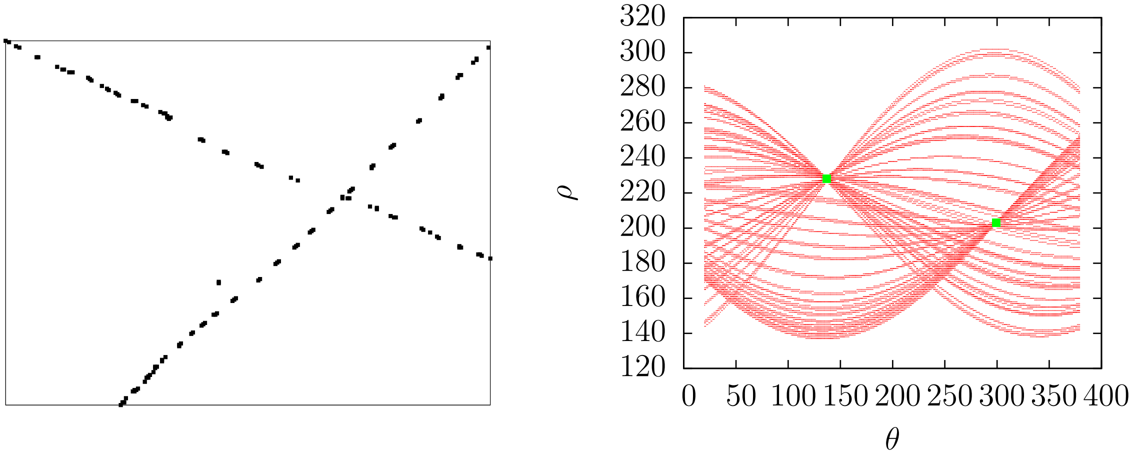


Figure 2.3: An illustrative examples for the Hough line detections. In left, image points in black. In right, the lines Hough space and the salient parameters as squares.

each pixel in the neighborhood. SIFT subdivides the 16×16 windows into quadrants that later are combined to obtain 128 values from the original 256 gradient values. This 128 values can be considered as the descriptor, but vector normalization can be performed to obtain higher stability. As mentioned earlier, this approach focus in the textures. The region descriptor is said to be invariant to scale, rotation and affine transformations. In the Fig. 2.5 we show an instance of the features detected with SIFT, which correspond to the salient image regions based in their local texture.

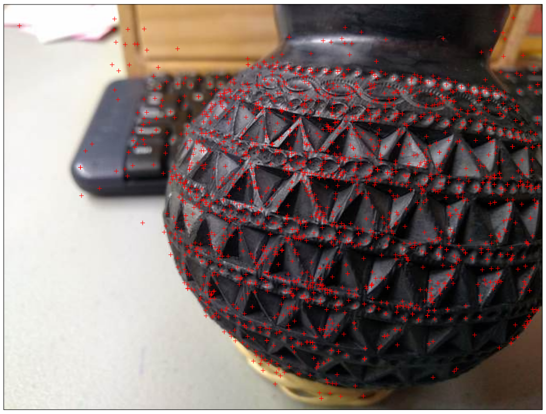


Figure 2.4: An illustrative example of the SIFT features. Here the features are the salient texture image regions.

2.3 Feature matching

The feature matching is the problem of identifying the same scene feature through different images of the input sequence. In the simplest case, the problem task is to determine the matching features of a pair of images. In the 3D reconstruction, this task is seen as a subproblem. Its solution is of crucial importance since it is the base for all the following sub-problems (camera calibration, pose estimation, and triangulation). As mentioned early, to retrieve a model point from the images it must be seen from at least two different viewpoints. The result of the feature matching can be considered as the location of a potential model point seen in the images. Although the feature matching is necessary to retrieve the model points, it is just the beginning for solving the complete problem. In Fig. 2.5 we show an instance of the matching obtained through the SIFT features. In the figure, we show a Mexican craft seen from two different viewpoints and the matched features.

The task is commonly solved in two steps: in the first one a preliminary matching is performed. This first matching is commonly done by using the SIFT [28] features with their associated descriptors. Given two images I_a and I_b and their detected features $\boldsymbol{p}_{ia}$ and $\boldsymbol{p}_{ib}$ with their SIFT descriptors. Each point $\boldsymbol{p}_{ia}$ is preliminarily matched with the point $\boldsymbol{p}_{ib}$ with the most similar descriptor vector. The result of this match is very rough, such that the match could not be one to one and even when it is one to one, the matching could be incorrect. To overcome this a validation is performed as second step. The validation consists in identifying, from the set of preliminary matches those that seem to be correct. The validation is performed through the robust estimation of the fundamental matrix, the fundamental matrix $F \in \mathbb{R}^3 \times 3$ is a projective transformation that holds

$$\boldsymbol{p}^{\mathrm{T}} F \boldsymbol{p} = 0. \tag{2.3}$$

With the robust estimation of F, those points that hold Eq. (2.3) (the inliers) are the final matched points. The robust estimation of F is solved using the random sample consensus algorithm (RANSAC) [31].

In the state of the art methods, the SIFT approach is the preferred to solve the matching; however, this approach mandatorily requires of textures in the scene.

2.4 Camera calibration

In this section, we review the camera calibration problem, which aim is to estimate the intrinsic and extrinsic parameters of the camera that was used to generate the input images. In our 3D reconstruction problem, the solution to this subproblem is crucial. The intrinsic camera parameters define how the scene is projected into the image; thus this information is essential for making possible the model retrieval. The camera parameters are estimated by using a calibration pattern [32]. A calibration pattern is an object whose exact locations of its features (corners or lines) are known in advance with high precision. This process is illustrated in Fig. 2.6. The calibration pattern

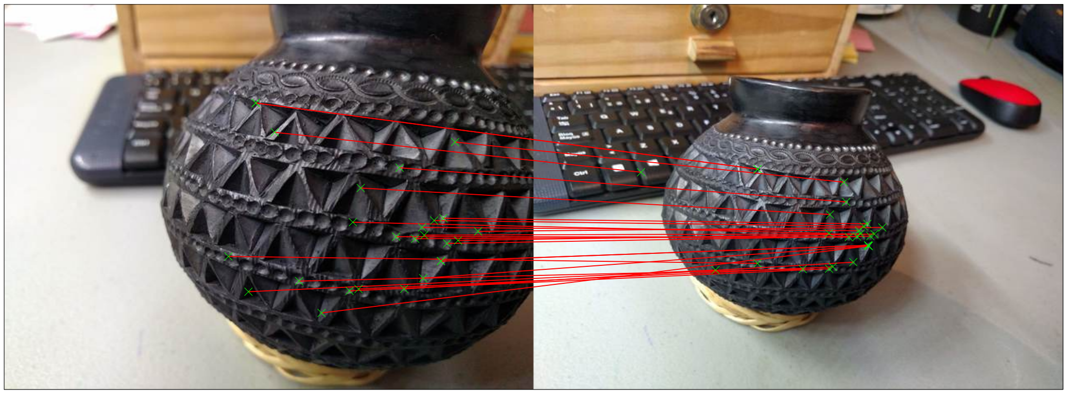


Figure 2.5: An illustrative example of the feature matching subproblem. Two images of the same scene from distinct viewpoints and the putative matched features with SIFT represented by lines.

is used to generate images with the camera we wish to calibrate. The images are analyzed to locate the calibration pattern features on the image. The pattern, as well as the image points, allow estimating the camera matrix M which is composed of the camera parameters. The estimation of M can be obtained through the Direct Linear Transformation (DTL) [22]; which consists in solving a linear system of equations for the eleven unknown M entries, in Eq. (2.1).

Some of the most popular calibration patterns are the chess boards, but there are also works that use other patterns such as ellipses [22] and 3D objects [33] to solve the calibration sub-problem. Some of the most popular calibration methods are the proposed by Tsai[32] and Zhang [34]. These methods use planar calibration patterns. The calibration methods first estimate the camera matrices by solving a linear least squares equations system for later refine them with a non-linear least squares method which also allows estimating the radial lens distortion coefficients.

One of the most important features of camera calibration is that it is performed offline. Camera calibration offers high-quality results at the cost of estimating first the camera parameters before images can be used for retrieving 3D information from them. When no calibration patterns are available, when camera parameters change dynamically on time, or when input images are generated with different and unknown cameras, the camera calibration is not a suitable tool, but the camera self-calibration is the alternative.

In the literature, we find that the camera calibration methods are very accurate. In this thesis, we aim to perform the 3D reconstruction using a not calibrated camera. Thus, the camera calibration is not an alternative; however, we use the camera calibration as a tool to obtain estimations, that due to its high precision, we use as ground truth. One of our contributions consists in estimating the intrinsic camera parameters, it is, in fact, a self-calibration method, which is reviewed in the following section.

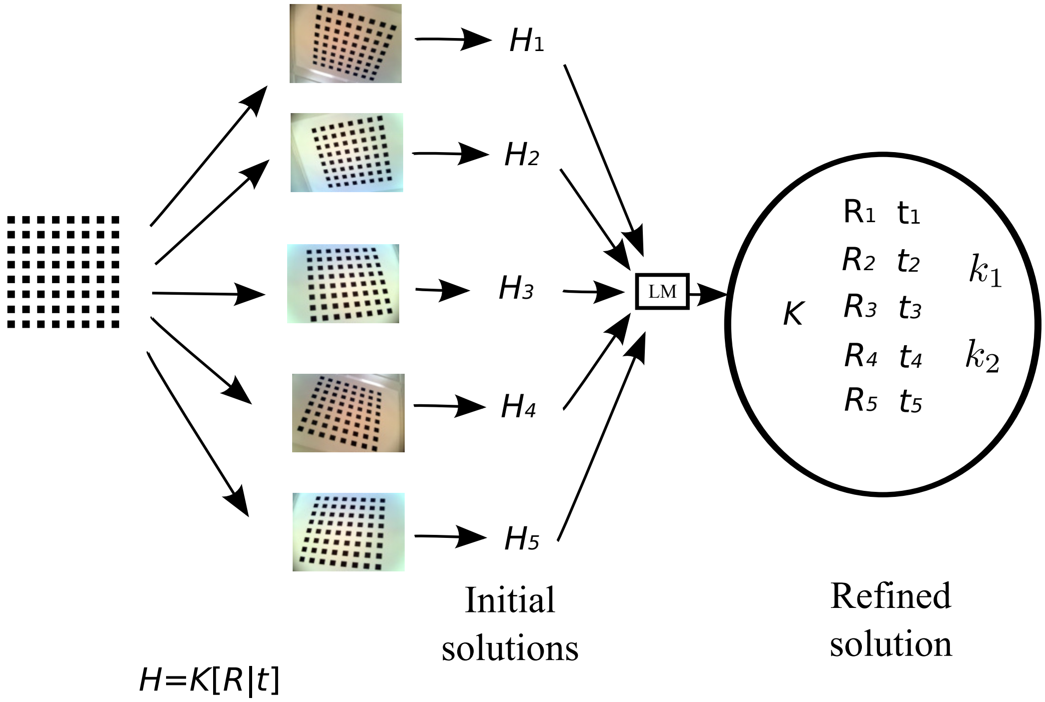


Figure 2.6: An illustration of the camera calibration. The chessboard as the calibration pattern and the input pictures used to obtain the camera intrinsic (K) and extrinsic parameters ($R, \boldsymbol{t}$).

2.5 Camera self-calibration

In the following, we present the camera self-calibration, which is the task of obtaining the intrinsic camera parameters without using any calibration pattern, since one of our contributions is a method to solve this problem. This central feature allows obtaining the internal camera parameters, i.e., focal distance, and pixel shape using only features (points) on images. The task cannot be solved in a general form, and some restrictions are used to regularize the problem, for instance, in Eq. (2.2), squared pixels are assumed $f_x = f_y$, the principal point is assumed to be in the center of the image, and the skew is assumed to be zero $s = 0$. With the mentioned constraints, the unique parameter to estimate is the focal distance $f = f_x = f_y$. The solutions obtained with these methods are less accurate than the obtained with calibration methods, but they are an alternative when no information for calibration is available. In practice, the solutions obtained through camera self-calibration are later combined with more information of the scene as more images or more model points, which are refined together through non-linear least squares to obtained a refined version. Some of the most popular self-calibration methods in literature are briefly described in the following paragraphs.

In [35], Hartley and in [36, 37] Pollefeys et al. propose to solve the camera calibration by first solving a projective reconstruction (assuming K as the identity matrix) for later upgrading the projective reconstruction to an Euclidean version by using more than three images.

In [38], Lei et al. propose to solve the calibration by using the Kruppa equations [22]. Authors estimate all unknown scalar factors by non-linear least squares. Once scalar factors are obtained, they compute the intrinsic parameters from the resulting non-linear constraints.

In [39], Triggs solves the problem through the use of the absolute quadric [22]. He first proposes to build a projective reconstruction for later rectify the reconstruction to a Euclidean one and obtaining the upgraded K matrix.

An essential characteristic of the self-calibration is that it is not a requirement to simultaneously estimate the camera extrinsic parameters. The methods based in the absolute conic, and Kruppa equations often solve only the intrinsic parameters, the camera poses can be estimated later with the estimated K. In the methods based in reconstruction upgrading, authors often solve first the camera poses and the camera intrinsic parameters are estimated in a second step.

The self-calibration sub-problem can be performed in an online manner. In practice the methods based on the Kruppa equations are the most popular; however, these methods are not accurate, and the Kruppa equations are difficult to solve. Although the camera self-calibration can be performed online, it only estimates the intrinsic camera parameters which are again just a piece to retrieve the complete scene which also must include an estimation of the model and the camera poses.

2.6 Pose estimation

Also known as extrinsic camera calibration, this task consists in estimating the extrinsic parameters of the camera associated to an image, i.e., the rotation matrix R which defines the orientation associated to the cameras as well as the translation vector $\boldsymbol{t}$ associated to its location. When the camera is not calibrated, but a calibration pattern is available, both the intrinsic and extrinsic parameters can be obtained simultaneously by the calibration methods presented in Sec. 2.4. The idea is to compute the camera matrix M for later decompose it. In general terms, the M has 11 degrees of freedom, and it can be estimated by using 6 3D/2D point matches. When the camera is already calibrated, the problem can be solved for 3,4, and 5 points. For these cases, the problem is named the Perspective n Points problem (PnP), with $n = 3, 4, 5$.

The pose estimation if another essential piece for the model retrieval. The camera pose defines how the scene was rotated and translated in the moment of the camera shot. If we count with a calibrated camera, and also we could with the pose associated with at least two images, the model points can be estimated through triangulation. In the literature, we find that it is very common to use restricted scenes, in such a manner that calibration patterns or information from other sensors are available [40]. The precision of the pose has a direct influence on the model points estimation.

When the camera pose is estimated using a planar object in the scene, the problem is known as pose estimation from homography decomposition. The homography decomposition requires a calibration pattern and the image correspondences in one image of at least four model points. One of the most popular methods for homography decomposition is the proposed by Zhang in his camera calibration method [34]. If the model points are not on a plane the problem is known as the Perspective n Points Problem (PnP), and it can be solved with at least three points, but more than one solution is obtained.

2.6.1 PnP

The minimal problem is about obtaining the parameters of the rotation matrix as well a the three parameters of the translation vector of the camera matrix from three points. This minimal problem is known as the perspective-3-point-problem (P3P). The problem is illustrated in Fig. 2.7. Since the three points on image $\boldsymbol{p}_a$, $\boldsymbol{p}_b$, and $\boldsymbol{p}_c$ define a triangle with known lengths and the same for the 3D points $\boldsymbol{P}_a$, $\boldsymbol{P}_b$, and $\boldsymbol{P}_c$, then, the angles formed by the rays passing through the camera center and each pair of 3D/2D points are known $\angle \boldsymbol{P}_a \boldsymbol{o} \boldsymbol{P}_b$, $\angle \boldsymbol{P}_a \boldsymbol{o} \boldsymbol{P}_c$, and $\angle \boldsymbol{P}_b \boldsymbol{o} \boldsymbol{P}_c$. The problem is to estimate the lengths of the line segments l_a,l_b, and l_c that connect each 2D $\boldsymbol{p}_i$ with its correspondent 3D point $\boldsymbol{P}_i$. Once the three lengths are solved, we can compute the rotation matrix and the translation vector from them. The solution to this problem is a polynomial equation system [41]. This P3P problem has four possible solutions. From these possible solutions, we must choose one, in practice the solution chosen is the one in which all 3D points lie in front of the cameras. For the P4P and the P5P problems there also exist ambiguities and multiple solutions, sixteen for the P4P and two for the P5P. For this reason, when more than five points are available, it is simpler to obtain the pose by decomposing the camera matrix M which can be easily decomposed when the camera is already calibrated as $[R|\boldsymbol{t}] = K^{-1}M = K^{-1}K[R|\boldsymbol{t}]$.

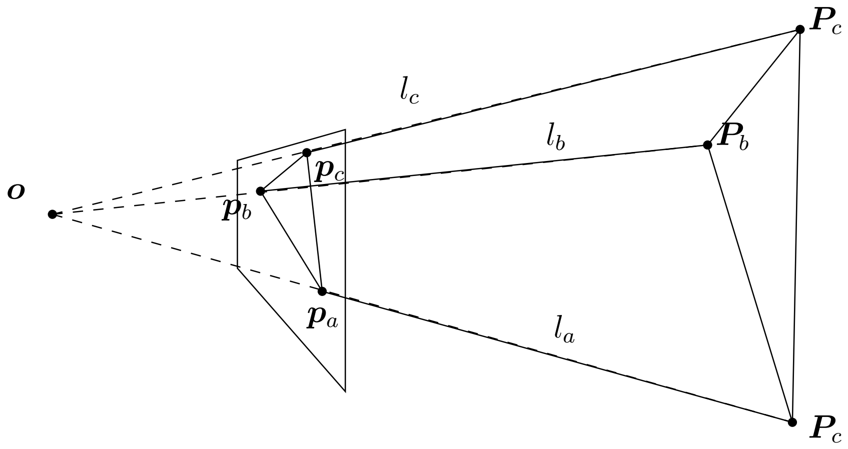


Figure 2.7: The P3P problem. The problem is to estimate the lengths of the line segments l_a,l_b, and l_c.

When the number of points is four, and when those four points are coplanar, the pose can be estimated through homography decomposition. In the last decade, the pose estimation from planar targets has been intensely studied [42, 43]. It has been found that the problem with four points has local minima [42]. These local minima are related to the rotation parameters. A more in-depth review of this fast is presented in Appendix. A. In the literature, we find that the best method to identify the best pose, even when there exist local, is to evaluate the reprojection error of all the found solutions. In practice, the correct solution has a lower reprojection error, what can be seen as the global minimum for the problem. In this thesis work, we implicitly exploit this fact, and we propose a method that can find the global minimum for the pose estimation.

2.7 Triangulation

The triangulation is the estimation of 3D points from image point correspondences and cameras as camera matrices M_i. The subproblem can be seen as finding the intersection of at least two rays passing through the camera centers and the feature correspondences for the same model point. The problem in concept is simple, and it has a unique solution that can be computed directly in the absence of noise. The problem becomes more complicated with noise. In this case, the intersection of rays is no longer valid; in fact, two rays could no intersect in any point in the space. To handle this problem the most acceptable alternative is to find the point that minimizes an error function. Different error functions can be minimized, for instance, we have the object space error, the angular error, and the reprojection error; a more extended review of these different errors is presented in Sec. 2.9. In the literature is well known that the minimization of the reprojection error allows obtaining the optimal solution even when the camera is not wholly calibrated. The triangulation through the mimization of the reprojection error requires comonly of iterative methods. In our thesis work, we need to perform multiple triangulation of points as fast as possible. The simplest but fastest approach is to find the intersection of rays through the Direct Linear Transformation algorithm (DLT) [22]. The DLT algorithm cannot be directly solved since due to the presence of noise the rays could not intersect in space and the number of solutions become infinite. Even more, we propose to solve the triangulation using the information from at least three images. The DLT algorithm allows performing triangulation utilizing the point matches in more than two images. The problem becomes overdetermined equations system that we solve through singular value decomposition for numerical stability. The triangulation through DLT is only valid when the camera is calibrated, however in this thesis work we show that the DLT is enough to allow us to solve the self-calibration problem.

In a generalized way, the triangulation problem through DLT can be seen as finding the space point $\boldsymbol{P}$ which minimizes the reprojection error of it projection to images and the image measurements, as illustrated in Fig. 2.8. Points in 3D space have three components (three degrees of freedom) $\boldsymbol{P} = [x, y, z]^{\mathrm{T}}$, and its projections on the image have two $\boldsymbol{p} = [u, v]$, thus, to triangulate a point $\boldsymbol{P}$, at least two images are required to solve the problem. It consists in using the equations for a same 3D point seen in two images, .i.e., $\lambda_1 \boldsymbol{p}_1 = M_1 \boldsymbol{P}$ and $\lambda_2 \boldsymbol{p}_2 = M_2 \boldsymbol{P}$; which leads to the solution to the overdetermined least squares problem. There exist two variations of this method the first assumes that a 3D point is expressed in homogeneous coordinates, i.e., $\boldsymbol{P} = [\lambda x, \lambda y, \lambda z, \lambda]$. This assumption, leads to obtain (for two images) a linear equation system with four equations and x, y, z, λ as unknowns of the form $A\boldsymbol{x} = 0$, and it is known as the homogeneous triangulation method [22]. The second, known as the inhomogeneous triangulation method assumes a fixed value $\lambda = 1$ and it leads to solve a linear equation system of the form $A\boldsymbol{x} = \boldsymbol{b}$ which can be solved through the normal equations. Both methods are very similar, but the homogeneous method considers that points can be at the infinity, i.e., the imaginary intersection point of two parallel lines, this occurs when very small values for λ are obtained in the solution. These linear methods are very simple but they are very easy to extend when more than two images are available.

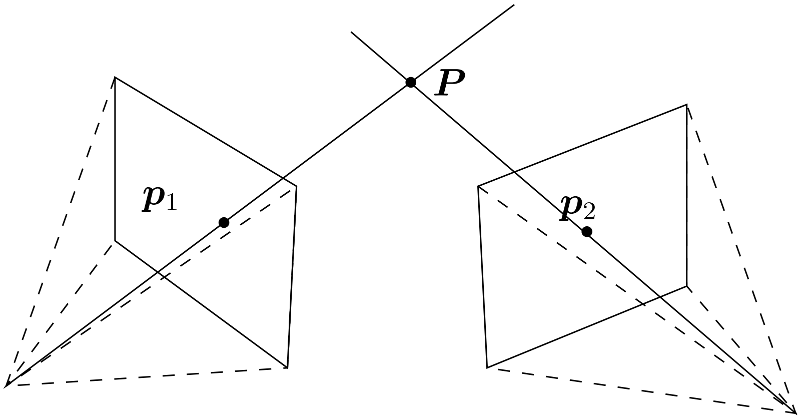


Figure 2.8: Triangulation of the 3D point $\boldsymbol{P}$ from its image correspondences $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$. Ideally $\boldsymbol{P}$ is the intersection of the two rays through the camera centers and the image points.

2.8 Bundle adjustment using the Levenverg - Marquardt algorithm

In the 3D reconstruction context, the Bundle Adjustment is the refinement of a current solution. Commonly, the improvement is performed through the minimization of a cost function based on an error measurement. The methods for the minimization are of local nature; this means that the algorithms require of an initial solution which is refined until reaching a local optimum of the cost function. Ideally, BA needs an initial solution that is close enough to the global minimum, but this can not always be guaranteed. In general terms, we can only consider BA to take an initial solution to the closest local minimum in the search space. The bundle adjustment can refine a different number of parameters; parameters to refine will depend on the problem solution to improve. In the literature, we find that BA could be used to refine every 3D reconstruction subproblems solutions, but also the complete reconstruction. The application of BA each time a subproblem is solved ensures better solutions however its application results expensive in practice. In this thesis work, we consider the refinement of the complete obtained reconstruction.

We consider BA as the non-linear refinement of the camera intrinsic and extrinsic parameters. The refinement can also involve the reconstructed point cloud and other parameters as lens radial distortion. The refinement is stated as an optimization problem where the reprojection error must be minimized, such that:

$$e = \sum_{j}^{m} \sum_{i}^{n} w_j(||\pi(M_j, \boldsymbol{P}_i) - \boldsymbol{p}_{ij}||^2),$$

where m is the number of images, n is the number of points, w_{ij} is the weight that reduces the contribution of outliers to the reprojection error cost function for the point $\boldsymbol{P}_i$ in the $j-$th image, π is the projection of the 3D point $\boldsymbol{P}_i$ into the image plane associated with the camera matrix M_j, and $\boldsymbol{p}_{ij}$ is the correspondent image point, e is known as the reprojection error.

The algorithm commonly used to solve the optimization problem is Levenverg-Marquardt [22, 44]. The method minimizes the sum of squares of 0linear functions: $s(\boldsymbol{x}) = 0.5 \sum_{h}^{k} f(\boldsymbol{x})^2$.

The method obtains a better solution in each iteration using the Jacobian matrix of $s(\boldsymbol{x})$. The complexity for BA is $O(n^3)$, where n here is the number of cameras [45, 46], but there are also some methods that solve the problem approximately [47] and reduce the complexity to O(n) or parallel methods [48].

2.9 Error measurement

In Computer Vision it is desirable to have an indicator that helps to evaluate the final result of a 3D reconstruction. One of these indicators would be the norm of the errors of the points in the three-dimensional space of the reconstruction and the ground truth model: $\sum |\boldsymbol{P}_i - \hat{\boldsymbol{P}}_i|$, where $\hat{\boldsymbol{P}}_i$ would be the ground truth model points. Unfortunately, this measure is in most cases impractical because most of 3D reconstruction approaches not always obtain a reconstruction point for each ground truth point, and for the case of non-metric reconstructions where the model is obtained up to an unknown scale factor, the error would also be scaled.

Instead of measuring the model distances in 3D space for measuring the quality of the reconstruction, it is preferred to use indicators that are invariant to scale, i.e., the reprojection error, the object space error, and the angular error. In the following subsections, we describe these kinds of errors.

2.9.1 Angular error

With a calibrated camera, i.e., the focal distance is known, we can obtain the direction of the ray passing through the camera center $\boldsymbol{o}$ and an image observation $\boldsymbol{p}$, denoted as $\boldsymbol{r}_{\boldsymbol{o},\boldsymbol{p}}$. Given the 3D point $\boldsymbol{P}$ correspondent to the $\boldsymbol{p}$ image point, a ray $\boldsymbol{r}_{\boldsymbol{o},\boldsymbol{P}}$ can also be obtained. The angular error [49] is the angle between the rays $\boldsymbol{r}_{\boldsymbol{o},\boldsymbol{p}}$ and $\boldsymbol{r}_{\boldsymbol{o},\boldsymbol{P}}$, i.e., $\theta = \angle(\boldsymbol{r}_{\boldsymbol{o},\boldsymbol{p}}, \boldsymbol{r}_{\boldsymbol{o},\boldsymbol{P}})$, which is illustrated in Fig. 2.9.

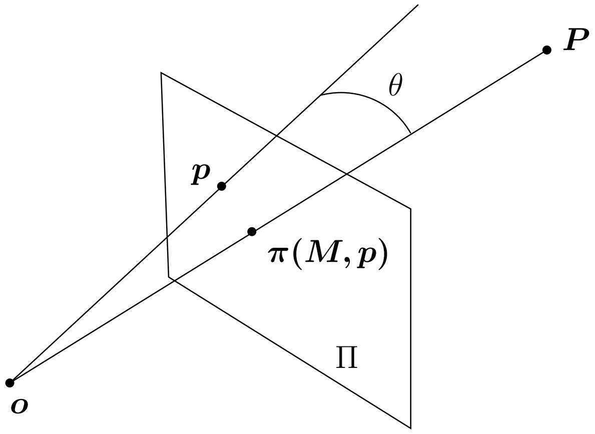


Figure 2.9: Given a 3D point $\boldsymbol{P}$, its image measurement $\boldsymbol{p}$, and the reprojection of $\boldsymbol{P}$ into Π, $\pi(M, \boldsymbol{P})$, the angular error is the angle θ formed between the line though $\boldsymbol{o}$ and $\boldsymbol{P}$ and the line through $\boldsymbol{o}$ and $\boldsymbol{P}$.

2.9.2 Object space error

Given a calibrated camera, and given the ray passing through the camera center $\boldsymbol{o}$ and an image observation $\boldsymbol{p}$, denoted as $\boldsymbol{r}_{\boldsymbol{o},\boldsymbol{p}}$. The object space error [50] is the orthogonal distance from 3D reconstructed point $\boldsymbol{P}$ to the ray $\boldsymbol{r}_{\boldsymbol{o},\boldsymbol{p}}$, which is illustrated in Fig. 2.10.

2.9.3 Reprojection error

It is defined as the norm of the difference in the image space. Given a ground truth image point $\hat{\boldsymbol{p}}$ and its approximation $\boldsymbol{p}$, the error can be measured as $||\boldsymbol{p} - \hat{\boldsymbol{p}}||$, and it is illustrated in Fig. 2.11. This is valid for image points, in order to evaluate a reconstructed model, we can use the reprojection of an estimated 3D point $\boldsymbol{P}$. Since our input are the images of the scene to reconstruct we could measure how similar are the ground truth images and the generated back from the reconstructed model. With this consideration, the reprojection error is defined as:

$$||\hat{\boldsymbol{p}} - \pi(\boldsymbol{P}, M)||,$$

where the $\pi(\boldsymbol{P}, M)$ is the projection of the 3D point $\boldsymbol{P}$ by the camera matrix M to the image space.

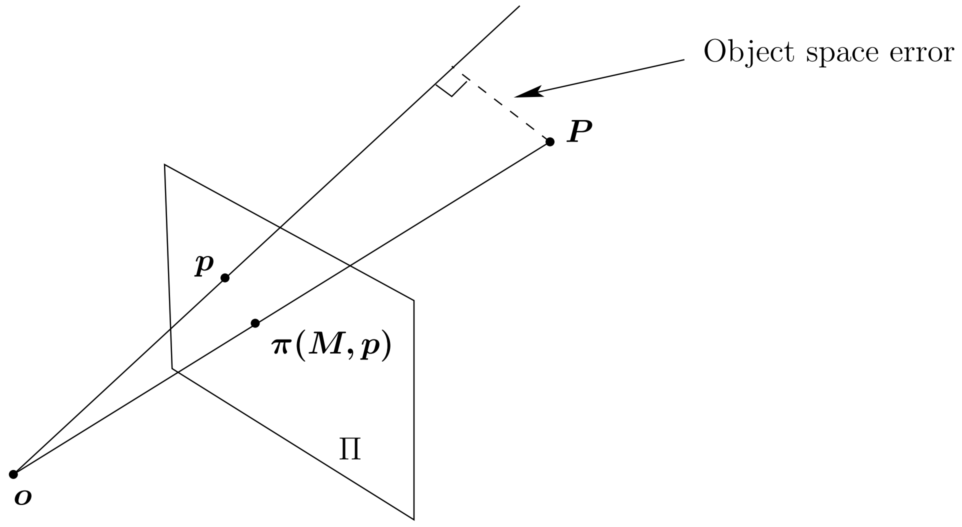


Figure 2.10: Given a 3D point $\boldsymbol{P}$, its image measurement $\boldsymbol{p}$, the object space error is the orthogonal distance between the 3D point $\boldsymbol{P}$ and the line through $\boldsymbol{o}$ and $\boldsymbol{p}$.

The reprojection error is one of the most used indicators in SFM. It allows to evaluate 3D reconstructions, and its minimization is used to estimate homographies, projective transformations, and it is also used to refine solutions by Bundle Adjustment.

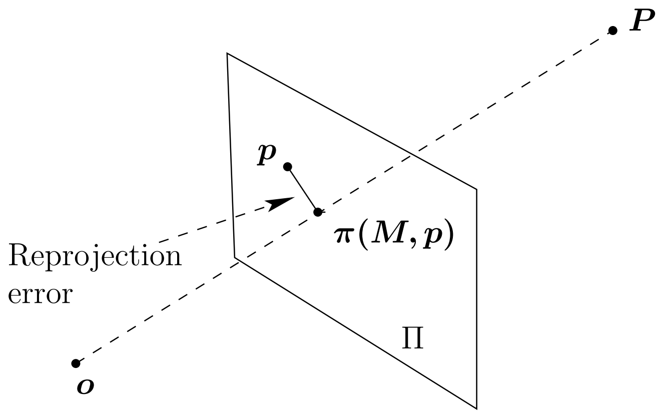


Figure 2.11: Given a 3D point $\boldsymbol{P}$, its image measurement $\boldsymbol{p}$, and the reprojection of $\boldsymbol{P}$ into Π, $\pi(M, \boldsymbol{P})$, the reprojection error is the distance between $\boldsymbol{p}$ and $\pi(M, \boldsymbol{P})$.

In this thesis work, we focus on minimizing this error measure. This error is also known as the geometrical error, and it is a valid error to minimize even when utilizing not calibrated cameras, which is our scenario since we propose to solve the self-calibration problem as well as other subproblems simultaneously.

2.10 3D reconstruction

The 3D reconstruction problem has been widely studied and many works have been developed. In the following, we present a brief description of the most relevant works that are part of the state of the art in 3D reconstruction.

The 3D reconstruction works can be first divided by the information that is available to solve the problem. There are two main categories, the **incremental** and the **global** approaches; both approaches consider static scenes.

In the incremental approach, the idea is to generate an initial reconstruction using just a few number of images (commonly two) for later start processing new images, one in each iteration. The processing of each image involves the estimation of its camera orientation and location. Each new image contributes with new matched features that allow performing the triangulation of new reconstruction points (those that are visible in at least two images). Some of the most relevant characteristics of these works are: they perform bundle adjustment after a new image is processed, they allow to perform the reconstruction even when the images are gradually given, they can work with a non-calibrated camera. In this category, we find the following works.

In [10], Schönberger and Frahm propose a SFM approach that enhances many of the sub-problems in the full pipeline, it provides a next view selection method, a robust triangulation method, a geometric verification of matched points, and also a strategy to perform the bundle adjustment, triangulation and outlier filtering.

In [11], Zheng and Wu present a SFM approach that includes a novel pose estimation method that solves the exact pose estimation using the already computed poses and only image points, i.e., do not require the 2D/3D matching.

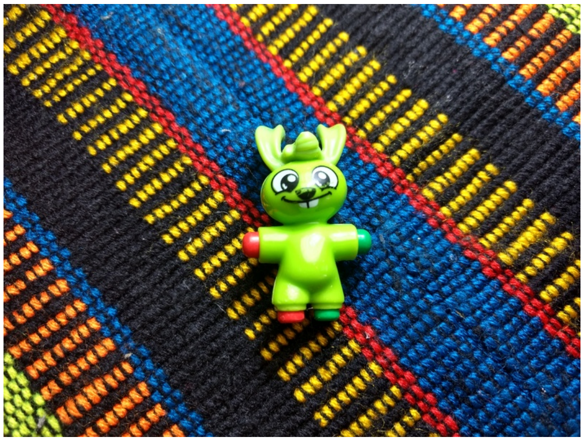
In [12], Wu presents an SFM approach that tries to solve all the problem in O(n) time. The work focus in the feature matching, which is one of the bottlenecks for SFM and also it uses an accelerated BA with O(n) time complexity that uses preconditioned conjugate gradient [51], additionally, in order to increase the number of triangulated points it proposes to periodically try the points re-triangulation also considering those points that were not possible to triangulate in previous intents.

In [13], Moulon et al., propose to solve the SFM problem by estimating robustly the feature matching and camera pose estimation sub-problems using RANSAC and the so-called the *a contrario* methodology. The used methodology allows to avoid setting fixed thresholds for the inlier counting step of RANSAC; it chooses a threshold automatically by exploiting the fact that outliers must have an high deviation.

The global approach assumes more information. It assumes that all images to be processed are given at the beginning. The camera is assumed to be calibrated and also it assumes the relative orientation between each pair of images to be known. The aim of this approach is to estimate the orientations and locations of all the cameras simultaneously in the global coordinate system. Some of the main characteristics of this approach are: images can be given unordered, it disseminates the residual errors among all the cameras external parameters and the final reconstruction. Many authors propose to perform bundle adjustment to reduce errors in the input relative poses and also in the final reconstruction. In this category we find the following works:

In [52], Cui and Tan propose a global SFM that first generates depth images from each pair of related images. Since every pair of related images allows to obtain a local reconstruction, all the camera poses could be theoretically integrated into a global coordinate system from the 3D/3D local models registration. The paper proposes to avoid the 3D models registration due to the poor quality that can be obtained using only two images. Instead, the authors propose to estimate depth-maps for later solve the camera poses.

In [53], Crandall et al. propose a global SFM approach. It uses the focal distance value present in the attached EXIF information of the input images, and it also requires the relative camera pose between images pairs. This EXIF information is metadata that is incorporated into images taken with modern cameras of smartphones; an example can be seen in Fig. 2.12 The paper first states the problem as a discrete optimization problem to initialize the parameters, which aim is to avoid local minima and later the problem is handled in a continuous mode.



Attribute	Value
Focal Length	4.7 mm
Pixel X Dimension	4608
Pixel Y Dimension	3456
X-Resolution	72
Y-Resolution	72
Resolution Unit	Inch
Date and Time	2017:11:24 11:45:38
YCbCr Positioning	Centered
Exposure Time	1/30 sec.
F-Number	f/2.0
ISO Speed Ratings	100
Exif Version	2.2
Components Configura	Y Cb Cr -
Shutter Speed	4.90 EV (1/29 sec.)
Aperture	2.00 EV (f/2.0)
Saturation	Low saturation
Sharpness	Soft

Figure 2.12: Instance of a picture and its associated EXIF metadata. Note that the focal distance is present in this example.

Our work focus on exploiting the information of one sub-problem to solve others, thus our contributions could be used for both, the incremental and the global approach.

Chapter 3

Self-Calibration, Pose estimation, and Model reconstruction using Three images

In this chapter, my advisor and me research about the joined solution to three of the core subproblems in the 3D reconstruction pipeline. Here we assume that the features extraction and the features matching based in Order Type is previously solved. In the Chapter 4 we detail how these sub-problems are solved simultaneously through the Order Type concept. Then, our inputs for this study is a sequence of images, the features locations, an associated descriptor for each of the images and the information about the matches. This information is the list of images and, per each image, a sublist of the visible image features in it.

For our approach, we aim to solve the camera self-calibration, the pose estimation, and the model reconstruction, three of the 3D reconstruction core sub-problems. In the literature it has been shown that the self-calibration problem can be solved using at least three images; then we propose our approach to work with this minimum number of images. In a sequence of images taken with the same camera, it is expectable that the difference between adjacent images is small; if the difference is little, then it is an acceptable hypothesis to expect to find a sufficient number of matched features in the three images.

The three addressed sub-problems in this approach are intimately related. Each of the sub-problems could be solved if the two remaining sub-problems are previously solved. The camera can be calibrated knowing the image poses and knowing the 3D model in the scene; another case could be that an image pose can be solved if the camera is calibrated and we count with the model in the scene, and so on.

Based on the previously mentioned observations, in this chapter, my advisor and me argue that we can simultaneously solve the three core sub-problems to obtain a method that simplifies the 3D reconstruction problem with competitive results.

First, we need to mention that in the existing literature each sub-problem is solved separately. For each of the subproblems, the current methods deal with their limitations and complications. For instance, in the self-calibration case, the Kruppa equations suffer from numerical instabilities, what makes them difficult to solve, and multiple solutions could be obtained. For the pose estimation, in the most popular methods [54], it happens that the problem is stated as the solution of an equations system of high order polynomials; that due to their nature have multiple solutions, up to 25 or 40 candidate solutions. For the model reconstruction, it is necessary first to have a precalibrated camera and to have a good quality estimation of the poses. The model is obtained through triangulation. There exist triangulation methods that consider that the camera could be not calibrated, in which case we will obtain a projective reconstruction. Those methods allow obtaining points that are affected by the same projective transformations of the camera. In such a manner that the triangulated points can be later transformed to obtain the metric version of the reconstruction, however, these methods exist only for two images which limits the accuracy of the obtained models even when more input images are available.

To avoid all the mentioned issues, we formulate the equations to calibrate the camera, find the pose of three images, and to reconstruct the scene (in 2D or 3D), all these three problems at the same time. The generated problem is a non-linear because there are involve sinus and cosines in the rotation angles. Therefore, the problem is treated as a non-linear optimization problem, and solved directly using the Differential Evolution (DE) algorithm. A non-linear problem is normaly solved using the result of a linear problem as a seed for a non-linear optimization problem. Examples of non-linear algorithms are the Newton, Gauss-Newton or Levenberg-Marquard algorithms. As in the formulated problem it is not known how to solve it linearly, the heuristic DE is used instead.

We study the proposed approach when the input feature correspondences are distributed in an arbitrary position, but additionally, my advisor and me propose a method to incorporate constraints of parallelism and perpendicularity for planar models. The inclusion of these constraints is not applicable to the general case, but when the input data allows its application, this can be seen as a method to exploit the information previously known about the scene to improve the results of the reconstruction.

In real datasets it is common to find that the input data is not always ideal; the data could be contaminated with atypical data, to deal with this fact, we propose a method to deal with outliers.

The reconstruction refinement is a way to ensure the best results, but an abuse of it results expensive, thus to address this issue, we also propose a mechanism to reduce the number of times the total solution is refined with the bundle adjustment algorithm.

To evaluate our proposal we validate it for the minimal conditions (motion type, and the number of required point correspondences) in which our approach can work, and we tested our approach with 2D and 3D synthetic and real datasets.

This chapter is structured as follows, in the Sec. 3.1, my advisor and me present the details of the proposed approach for solving the camera self-calibration, pose estimation and the model reconstruction simultaneously as an optimization problem. In the Sec. 3.2, we present the experimental validation with synthetic and real datasets. In Sec. 3.3 we present the discussion, and in Sec. 3.3 we present the remarks for this study.

3.1 Proposed approach

First, we need a strategy to join the three sub-problems. One method to evaluate the overall result of reconstruction is to measure the reprojection error. In the literature, the reprojection error is known to be a good measure to assess the reconstruction quality. This error is only computed using distances of points in the images space; thus it does not need more information than image points measurements and their corresponding reprojection using the reconstructed model. With this fact, my advisor and me propose to state the simultaneous problem as an optimization problem in which we minimize this reprojection error. Each of the three involved sub-problems requires different parameters to be optimized, for the camera self-calibration we need to solve five parameters, the corresponding to the intrinsic camera parameters, but this can be simplified if we consider square pixels ($f_x = f_y$ in Eq. 2.2 in page 10), the principal point at the center of the image, and zero skewness ($o = 0$ in the same Eq. 2.2), thus we need to solve only the focal distance. Since we also aim to address the pose estimation, for the three images, my advisor and me need to solve 18 pose parameters; because the pose for a single image consists of six parameters, three rotation angles, and three translation components. We need to solve the reconstruction in a fixed coordinate system; otherwise, there would exist infinite solutions to the problem, one answer for each possible rotation, translation and scale transformation. To solve this, we fix one of the pose angles and we delimit the search space with box constraints. This strategy allows us to set the solution in a unique coordinate system, but at the same time, it will enable us to reduce one of the decision variables. Then, our approach for the simultaneous self-calibration, pose estimation and model recontruction is stated as looking for the decision vector:

$$\boldsymbol{w}_1 = [f, \theta_2^1, \theta_3^1, \boldsymbol{\theta}^2, \boldsymbol{\theta}^3, \boldsymbol{t}^1, \boldsymbol{t}^2, \boldsymbol{t}^3]^{\mathrm{T}},$$

where, $\boldsymbol{w}_1 \in \mathbb{R}^{18}$, f is the focal distance of the camera, $\boldsymbol{\theta}^j = [\theta_1^j, \theta_2^j, \theta_3^j]^{\mathrm{T}}$ and $\boldsymbol{t}^j = [t_1, t_3, t_3]^{\mathrm{T}}$ corresponds to the three Euler angles and the translation vector associated to the $j-$th image, respectively. The vector $\boldsymbol{w}_1$ encodes all the required parameters to define the 3D pose and a calibration matrix for the three images. The value of θ_1^1 is fixed to zero, thus, is not included in $\boldsymbol{w}_1$. Here we say we are using the rigidity of the viewing scene as a fundamental constraint.

In our approach, each evolutionary algorithm individual has associated a proposal for the solution vector $\boldsymbol{w}_1$ that defines a reconstruction and a calibration matrix, thus it has associated a reprojection error $g_1(\boldsymbol{w}_1)$. We use this reprojection error $g_1(\boldsymbol{w}_1)$ as the fitness function for the evolutionary algorithm.

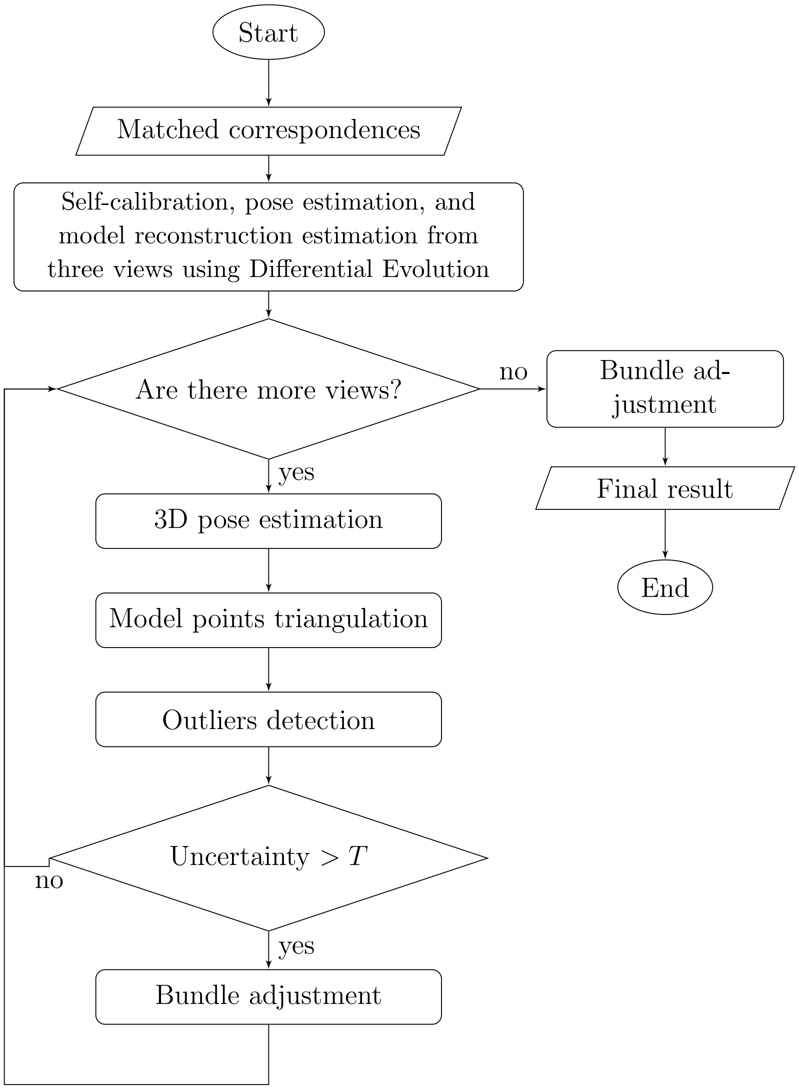


Figure 3.1: General flow for our 3D reconstruction approach.

To evaluate $g_1(\boldsymbol{w}_1)$ we need to set all the encoded parameters as part of the putative reconstruction, i.e., the camera parameters and the camera poses for the three images and the reconstructed model. The decision vector $\boldsymbol{w}_1$ does not include the reconstructed model points per se, but it contains all enough information to compute a reconstruction of the model by triangulation. The specific steps to evaluate the fitness function $g_1(\boldsymbol{w}_1)$ are the following:

1. Define the calibration matrix K using $f \in \boldsymbol{w}_1$
2. Define the three projection matrices: M_1, M_2, and M_3 as $M_j = K[R_j|\boldsymbol{t^j}]$ with $R_j = R_z(\theta_3^j)R_y(\theta_2^j)R_z(\theta_1^j)$, $\theta_1^1 = 0$.
3. Triangulate the model points $\boldsymbol{P}_i$ using the correspondences in the three images and the projection matrices in step 2 using the DLT algorithm.

4. Normalize the obtained cloud point $\{\boldsymbol{P}_i\}$: center the computed cloud point $\{\boldsymbol{P}_i\}$ to its centroid $\bar{\boldsymbol{P}}$, and scale it to ρ/σ, where σ is standard deviation of all points coordinates x, y and $z \in \{\boldsymbol{P}_i\}$, i.e., $\boldsymbol{P}_i = \rho/\sigma(\boldsymbol{P}_i - \bar{\boldsymbol{P}})$.

5. Compute the reprojected points $\lambda\hat{\boldsymbol{p}}_{ij} = M_j\boldsymbol{P}_i$ for each point in the step 4.

6. Return

$$g_1(\boldsymbol{w}_1) = \frac{1}{l}\sum_{i=1}^{l}\sum_{j=1}^{m}||\boldsymbol{p}_{ij} - \hat{\boldsymbol{p}}_{ij}||^2, \tag{3.1}$$

In the procedure above, in the step 4, we set the scale factor to $\rho\sigma$ to make the problem to be defined in a fixed scale. The value for ρ is set arbitrary and it can be modified, it depends on application, and it controls the model size with respect to global optimization algorithm box constraints. If $0 < \rho < 1$ the reconstructed model and translation vectors $\boldsymbol{t}$ will be obtained in smaller scale, $\rho > 1.0$ will increase the reconstruction size. For this paper my advisor and me set $\rho = 0.01$, but we also consider the algorithm reset to auto-adjust the model scale in Sec. 3.1.4.

Model reconstruction For our approach we use the DLT algorithm for the triangulation. In literature we find that this method is one of the simplest but it is only suitable and valid for Euclidean or metric reconstructions, i.e. with a calibrated camera, in the case of projective or affine reconstructions its accuracy is not so good [55]. Although the method seeks for Euclidean reconstructions, the evolutionary process of the DE starts by obtaining projective reconstructions that are later being evolved until converging to the Euclidean version. Taking this process in consideration, our selection for triangulation does not affect the final results. At the beginning of the DE, the DLT triangulation allows us to obtain triangulation approximations that are enough to guide the evolutionary process to the Euclidean reconstruction, as long as the solutions are closer to the final result, the DLT triangulation is more accurate and my advisor and me obtain good results. Since we only use the final result, the use of DLT is valid for the proposal, even when we solve the self-calibration sub-problem, and additionally it eases the use of the information of multiple images (more than two), which helps our method to deal with noise.

In [56], the authors study in deep the triangulation problem from three views. They propose a method to obtain the optimal solution through the extraction of the roots of a set of multivariate polynomial equations. In their approach they construct a 47 × 47 matrix for the equations system, which has 47 roots counting the real and complex solutions; even more, for ensuring the solution stability, authors use a high precision linear algebra library [57], which in practice results very costly, up to 30 seconds per triangulation. A crucial fact is that, even with all considerations to ensure the optimal, the authors report that the linear solution followed by refinement through the BA finds the same global minimum in all their experiments. This fact serves as a support of using this triangulation in our approach since we need to solve the triangulation of points from its correspondences in at least three views with the better but fastest results.

3.1.1 Solution to the joined problem through Differential Evolution

We propose to use the DE's classical version (rand/1/bin), which is good for real parameters optimization [58] and has shown good results in many engineering problems [59].

DE is an evolutionary optimization algorithm originally proposed by Kenneth Price and Raíner Stor [59] for global optimization in continuous spaces. It is a population based optimizer where each individual represents a solution to a cost function $g(\cdot)$ to be minimized or maximized (we consider the minimization case).

The DE rand/1/bin version is based in a recombination operator that uses three randomly selected parents $\boldsymbol{w}^{r1}$, $\boldsymbol{w}^{r2}$, and $\boldsymbol{w}^{r3}$ from the population W to generate new trial individuals. These trial individuals can be incorporated to W depending on its fitness. The parameters for the algorithm are: the search space Q (called box constraints), population size N, crossover probability $R \in [0, 1]$, difference weight $F \in [0, 2]$. DE evolves the population W in a certain number of iterations, in this work, my advisor and me repeat the algorithm until an automatic stop criteria is reached. As stop criteria we use the proposed in [60]: algorithms stops if $s < g(\boldsymbol{w}_{\text{worst}}) - g(\boldsymbol{w}_{\text{best}})$. The complete DE algorithm is shown in Alg. 1;

The DE performs the search of the decision vector $\boldsymbol{w}_1$ that minimizes the fitness function $g(\cdot)$. For our purposes, we need the solution $\boldsymbol{w}_1$ to be inside a delimited subspace $Q \in \mathbb{R}^{18}$. Through the evolutionary process, in the generation of new trial individuals $\boldsymbol{v}$, it can happen that some of its elements $\boldsymbol{v}_i$ get out of the feasible search space Q. There exist many approaches to handle this situation [61, 58, 62, 63], but in this paper, we use the reinitialization approach [58], which consists in generating a new value inside Q for all the components outside Q. This handling is performed in the lines nine and ten of the Alg. 1. In the algorithm, the new value is generated randomly with a uniform distribution between $[Q^i_{\text{min}}, Q^i_{\text{max}}]$.

The box constraints and DE configuration parameters used in this work are shown in Tab. 3.1.

The proposed approach in this section works for points on arbitrary distribution. The model to reconstruct can be a 2D or a 3D model. There are certain cases where some information is available for the model to reconstruct. In this thesis, we want to make possible to exploit information previously known about the model. For this, we study the inclusions of two types of constraints about lines parallelism and perpendicularity. To include the additional information, we propose to use the same algorithm, but we propose to use a modified version of the triangulation method. The details for this complementary approach is explained in the Sec. 3.1.2.

```
Require: $Q$,$N$, $F$, $r$, $s$.
Ensure: A solution vector $\boldsymbol{w}$
Initialize randomly the population $W = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_N\}$ in the search space $Q$.
do
  for each $\boldsymbol{w} \in W$ do
    Select randomly three different individuals from $W$, $\boldsymbol{w}^{r1}$, $\boldsymbol{w}^{r2}$, and $\boldsymbol{w}^{r3}$.
    Select a random number $i_{\text{rand}} \in [1, n]$.
    for $i = 1$ to $n$ do
      if $U(0,1) < r$ or $i = i_{\text{rand}}$ then
        $\mathbf{v}_i \leftarrow \boldsymbol{w}_i^{r3} + F(\boldsymbol{w}_i^{r1} - \boldsymbol{w}_i^{r2})$
        if $\mathbf{v}_i \notin Q$ then
          $\mathbf{v}_i \leftarrow U(Q_{\text{min}}^i, Q_{\text{max}}^i)$
        end if
      else
        $\mathbf{v}_i \leftarrow \boldsymbol{w}_i$
      end if
    end for
    if $g(\boldsymbol{v}) < g(\boldsymbol{w})$ then
      $\boldsymbol{w} \leftarrow \mathbf{v}$
    end if
  end for
  Find the worst solution $\boldsymbol{w}_{\text{worst}}$
  Find the best solution $\boldsymbol{w}_{\text{best}}$
while $(s < g(\boldsymbol{w}_{\text{worst}})g(\boldsymbol{w}_{\text{best}}))$
```

Algorithm 1: DE (rand/1/bin) version.

Table 3.1: DE Box constraints and parameters for the self-calibration problem.

Parameter	2D Case	3D Case
f	[100,5000]	[100,5000]
θ_1	[-180^o,180^o]	[-180^o,180^o]
θ_2	[-90^o,90^o]	[-180^o,180^o]
θ_2	[-90^o,90^o]	[-180^o,180^o]
θ_3	[-180^o,180^o]	[-180^o,180^o]
t_1, t_2	[-100,100]	[-100,100]
t_3	[10,1000]	[10,1000]
Population size	50	50
Crossover probability	0.7	0.7
Differential constant	0.9	0.9
Stop criteria s	0.001	0.001

3.1.2 Self-calibration, pose estimation, and model reconstruction exploiting prior knowledge about the model

In the camera self-calibration problem, we assume not to count the model in the scene. If we would know, then we would be addressing with the camera calibration problem. However, for some cases, we could know partial information of the model in the scene. For instance, for the case of autonomous cars with a camera on board, we know that the road is formed by two parallel lines, the corresponding to the limits of the road. Even more, in the human-made environments, it is prevalent to find structures where parallel lines intersect. For that kind of cases, we propose to exploit the information to enhance our general approach. We propose to use the general approach explained in Sec. 3.1, but using the partial knowledge of angles between lines to constraint the location of the reconstructed points. Specifically, we propose to introduce these constraints as new expressions to perform the triangulation step.

As the strategy to include the information in our joined sub-problems approach, we propose a modified method for the triangulation of model points. We introduce two new triangulation expressions; one for the case when the model points are on parallel lines and the other when the points lie on perpendicular lines, both cases on the plane. The proposed expressions are the result of imposing the location of the points to reconstruct to the desired lines structure.

Expressions derivation

Considering a model with points $\{\boldsymbol{P}_i\}$ on a plane and constrained coordinates x_c, y_c, i.e., $\boldsymbol{P} = [x_c, y_c, 0, 1]^{\mathrm{T}}$.

Assuming $x_c = a$, $y_c = b$, $a \neq b$, and a homography $H \in \mathbb{R}^{3\times 3}$ with elements h_{ij}. We propose to modify the triangulation by DLT to constraint the model for the two considered cases.

Parallelism between lines: Considering two parallel lines on the model, $y_c = a$ and $y_c = b$, the computation of points model points in each line are:
For model points on the line $y_c = a$, $\boldsymbol{P}_{ia}$:

$$\boldsymbol{P}_{ia} = \left[\frac{h_{12}a + h_{13} - u_i(h_{32}a + h_{33})}{(h_{31}u_i - h_{11})}, a, 1\right]^{\mathrm{T}},$$

for model points on the parallel line $y_c = b$, $\boldsymbol{P}_{ib}$:

$$\boldsymbol{P}_{ib} = \left[\frac{h_{12}b + h_{13} - u_i(h_{32}b + h_{33})}{(h_{31}u_i - h_{11})}, a, 1\right]^{\mathrm{T}}.$$

Perpendicularity between lines: Considering points on two perpendicular lines, $x_c = a$ and $y_c = b$. The model points on the line $x_c = a$, $\boldsymbol{P}_{ia}$ correspond to:

$$\boldsymbol{P}_{ia} = \left[a, \frac{h_{11}a + h_{13} - (h_{31}a + h_{33})u}{(h_{32}u - h_{12})}, 1\right]^{\mathrm{T}},$$

the model points on the perpendicular line with $y_c = b$, $\boldsymbol{P}_{ib}$ are:

$$\boldsymbol{P}_{ib} = \left[\frac{h_{12}b + h_{13} - u_i(h_{32}b + h_{33})}{(h_{31}u_i - h_{11})}, b, 1 \right]^{\mathrm{T}}.$$

Our proposal of model constraints simplifies the triangulation step in such way that the estimation of the points can be performed from only one view. In the DLT from three images, we solve an overdetermined linear equations system. The solution involves the use of a numerical method, the SVD for best numerical stability. With our approach, we can estimate model points using only one an image and even more, in a closed-form way.

In our general approach for self-calibration, pose estimation, and model reconstruction of planar models with arbitrary points distribution of the Sec. 3.1 we find 18 parameters in the decision vector $\boldsymbol{w}_1$. These 18 parameters correspond to the focal distance and the 17 pose parameters assuming $\theta_1^1 = 0$, which fix the orientation of the reconstruction. For the case of line constraints, the orientation is given by the assumed values of the lines used for the constraints, thus for the constraints proposed in this subsection, the decision vector $\boldsymbol{w}_1$ has 19 parameters, i.e., the focal distance and the 18 parameters associated with the poses of the three images.

In this section, we present our approach for three core sub-problems, the self-calibration, the pose estimation, and the model reconstruction. In our approach, we solve the three sub-problems using the matched features in three of the images in the input sequence. This approach allows us to obtain an initial solution to the three sub-problems. If the input sequence has more than three images, that means that we need to process the additional images. As mentioned early, the pose estimation can be performed if we count with a calibrated camera, and also the model observed in the image. For the additional images we have image point matched features, some of these matched features correspond to the model obtained in the joined approach. Thus with all the results from the joined problem we can: estimate the pose for each of the remaining images in the sequence in the incremental approach, i.e., one by one. The new images will contain the featured matched location for new model points; then we can obtain new model points that contribute to the reconstructed model. Additionally, each additional image provides information about the already reconstructed model points. We use this information to refine the solution in such a manner that we can compute the fines function using more than three images or even we can re-triangulate model points all the available images. The details for the pose estimation are described in the Sec. 3.1.3.

3.1.3 3D pose estimation for more views

In this section, we describe our approach for solving the pose estimation sub-problem. Once we count with an initial reconstruction and the result of the camera self-calibration, we aim to address the pose for the remaining images of the input sequence. A camera pose is defined by six parameters, the six camera extrinsic parameters, three rotation angles, and a translation vector. The pose for a single image defines the model position and orientation at the moment of the camera shot. Since we aim to incrementally add new points to the initial reconstruction obtained in the joined sub-problems approach, then we first require to solve the pose for new images. In the literature, we find many strategies to solve the problem. A review of the traditional methods is presented in section 1. In practice, it was seen that the pose estimation implementations, e.g., those used in augmented reality, suffer from jumps in the image that interrupt the fluid movements of the virtual objects in the augmented scene.

In the last decade, it has been found that the pose estimation problem, at least for the case of planes, has local minima. Nowadays, the visual pose estimation jumps are associated with the problem multimodality and new works have been emerged to address the problem. The state of the art methods have found that there exist one local minimum and one global minimum to the problem; these local minima are present on the rotation parameters. To address the multimodality most of the methods find the two solutions, in such a manner that one of both is chosen as correct, generally the one with the less reprojection error. In appendix A, we investigate the pose multimodality, and we confirmed the existence of the two local minima.

In this section, we handle the pose estimation problem for only one image, but in our joined approach, we address the problem using three images; if the multimodal exits for a single image, then the problem will also exist for three images.

To handle this issue we use the global optimization heuristic. This approach allows us to handle the multimodality for the pose estimation.

Given the estimation for the calibration matrix K and a set of the reconstructed model points $\{\boldsymbol{P}_i\}$. We propose to add new views to the scene and incrementally add points to the initial reconstructed model. For this, we estimate the 3D pose of the additional views using the $\{\boldsymbol{p}_{ij}\}$ points visible on it, and using DE again. For the problem we fix the reconstructed point cloud $\{\boldsymbol{P}_i\}$ and K to look for the solution vector $\boldsymbol{w}_2 = [\boldsymbol{\theta}^j, \boldsymbol{t}^j]^{\mathrm{T}}$ that best minimizes the reprojection error of the model points visible in the new view, i.e., $g_2 = \frac{1}{l}\sum_{i=1}^{l} ||\boldsymbol{p}_{ij} - \hat{\boldsymbol{p}}_{ij}||^2$. Here the decision vector $\boldsymbol{w}_2 \in \mathbf{R}^6$ encodes only the camera extrinsic parameters for the $j-$th image. The box constraints and DE parameters for this problem are the same from Tab. 3.1 except for $t_1, t_2 \in [-300, 300]$. The reason to use bigger box constraints for the translation vectors is to allow the model to have greater distances from the camera center. Since the pose estimation problem is much easier than the self-calibration problem, the population size is set to 30 and the stop criteria is set to 0.01.

As it occurs in the simultaneous self-calibration, pose estimation and model reconstruction, the optimal value for $\boldsymbol{w}_2$ might be outside the defined search space Q. We detect this issue when at least one of the translation components of the pose is very near its corresponding box constraints. When that occurs, it is necessary to set a new reconstruction scale and restart the DE to obtain a new valid solution. In the Sec. 3.1.4 we detail the scale adjustment for the joined sub-problem but also for the pose estimation case.

Estimation of the camera distortion coefficients

When more than five images have been processed and added to reconstruction, for the 2D case, it is possible to apply bundle adjustment using the complete camera model in (2.2) to obtain f_x, f_y, o, u_0, v_0 and also to compute the camera lens distortion coefficients.

In the literature, the camera calibration involves the estimation of the camera intrinsic parameters, the extrinsic ones and in some cases, if the data allows it, the estimate of the radial distortion coefficients. In the Zhang calibration method, the distortion coefficients are solved in the refinement step. The solution obtained using a calibration pattern and the correspondences in three images through a linear method is refined through non-linear least squares method, BA. In the Zhang's method the five camera intrinsic parameters, the camera extrinsic parameters for each of the three camera poses, and two distortion coefficients are refined. BA requires an initial value to perform the refinement. We also apply the same approach with the difference that we use the values of $\boldsymbol{w}_1$ that obtained with our joined approach to obtain a refined version.

We initialize f_x and f_y with the value $f \in \boldsymbol{w}_1$, we also initialize the skewness as $o = 0.0$ and the principal point, $[u_0, v_0]$ as the center of the image. We initialize the two distortion coefficients k_1 and k_2 with zero, and we set as initial values the seventeen pose parameters $[\theta_2^1, \theta_3^1, \boldsymbol{t}^1, \boldsymbol{\theta}^2, \boldsymbol{t}_2, \boldsymbol{\theta}^3, \boldsymbol{t}_3] \in \boldsymbol{w}_1$.

To evaluate the associated reprojection error, we do not count with a calibration pattern. Instead, we triangulate the model to compute the reprojection error as we do in the Sec. 3.1. This is a main difference with the Zhangs method since we are performing the refinement of the simultaneous, self-calibration, pose estimation and model reconstruction through the BA.

The correct improvement of all the camera parameters, as well as the distortion coefficients, depends in some cases in the input data. In practice a good number of image correspondences are needed, and even more, they must be well distributed in the three images. For this reason, to perform the refinement, all the feature correspondences in the three images and its reprojection error are used in the refinement through BA.

3.1.4 Scale adjustment

In this step, we aim to allow the reconstructed model and the poses to estimate to be inside the DE search space Q and the defined box constraints. In our joined subproblems approach, we solve the camera self-calibration, the pose estimation, and the model reconstruction. Since we solve the camera self-calibration, this means that we do not use any calibration pattern, then it also means that we do not priorly know the model to reconstruct, its form nor its size. To perform the search through the DE we define box constraints that delimit the search space. Depending on the model size and the scene structure it could happen that the defined box constraints are not well suited to allow DE to find the optimal solution, i.e., the optimal solution can be outside the defined box constraints. We need to handle the issue for two different parts of our approach, first we need to handle the scale adjustment at the self-calibration, and then we need to handle it in the estimation of single images pose estimation.

In our three joined sub-problem approach we set an initial scale through the parameter ρ. The value of ρ, in each model triangulation, scales the model to a fixed scale. We propose an initial value $\rho = 0.01$, but if the self-calibration fails. We need to adjust the scale to a smaller size, and we require to restart the evolutionary algorithm. Depending on the value of ρ, the proposed approach implicitly will also scale the translation vectors associated with the estimation of the three involved images.

For the pose estimation case, the same issue can happen. To adjust the scale, we can not just change a scale parameter. In the pose estimation we use the fixed reconstructed model, then to scale the scene, we need to scale all the reconstructed model,all reconstructed points components must be scaled, but also all the translation vectors in the poses for all the already processed images. The resultant model should be obtained in a smaller scale. This model and translation vectors reduction will allow the fixed box constraints to handle bigger scenes whithout affecting the reconstructed scene structure. Once the scene scale is changed, the evolutionary can be restarted to a new solution. With this approach, a failure, in the estimation of $\boldsymbol{w}_1$ or $\boldsymbol{w}_2$ can be detected if some of the $\boldsymbol{w}_1$ or $\boldsymbol{w}_2$ components are on the box constraints limits.

3.1.5 Outliers detection

In our approach, the primary source of outliers is the wrong feature matching of features in images. They can lead to lousy triangulation results, in consequence, they can affect the self-calibration step and also the 3D pose estimation problem. To deal with outliers, we propose to delete them from the reconstructed model in a simple manner. After estimating the 3D pose for a new view and computing the new model points, we delete those model points that seem to be outliers by assuming a normal distribution of the reprojection errors e_i associated to each point $\boldsymbol{P}_i$. We compute the mean $\overline{e}$ and standard deviation σ_e of the reprojection error for the points $\{\boldsymbol{P}_i\}$, and we delete from the model those points that satisfy: $\boldsymbol{P}_i | e_i > \overline{e} + 2.5\sigma_e$.

3.1.6 Evaluating the reconstruction quality to reduce refinements

As mentioned early, the reconstruction refinement through BA is one method to ensure the best possible results. In the literature, we find that some approaches propose to perform a refinement each time a sub-problem in the 3D reconstruction is solved. The use of BA is of such an order that in some works the refinement is performed for each model triangulated. The use of BA is expensive; thus it can be beneficial to reduce its application. In this section, we propose a method to determine when refinement is convenient. In the incremental 3D reconstruction approach, the reconstruction error increases with each processed image as the result of the errors accumulation. Then, we propose to monitor the solution quality in such a manner that when the quality is degraded under a fixed threshold, we perform a refinement.

As an indicator, we evaluate the uncertainty of the reconstructed models and the estimated poses. We indeed do not have with this information, but we can use the reprojection error statistics to propagate the error to the 3D elements of the reconstruction as proposed in [64].

Considering the reprojection of a point $\boldsymbol{P}_i$ as the function: $\boldsymbol{F}_{ij}(\boldsymbol{w}_2, \boldsymbol{P}_i) = [\hat{u}_{ij}, \hat{v}_{ij}]^T = [g3(\boldsymbol{w}_2, \boldsymbol{P}_i), g_4(\boldsymbol{w}_2, \boldsymbol{P}_i)]^T$. We use the Jacobian matrix of F_{ij}: $J_{ij} = \frac{\partial F_{ij}}{\partial P_j} \in \mathbb{R}^{2n \times 3}$, where n is the number of processed images in which P_i is visible, to propagate the uncertainty of the reprojection error to the model. For this we perform the following procedure:

1. Compute the covariance of the reprojection errors for each point $\boldsymbol{P}_i$ as: $\sigma_0^2 = \mathbf{e}^T \cdot \mathbf{e}$, where $\mathbf{e} = [u_1 - \hat{u}_1, v_1 - \hat{v}_1, u_2 - \hat{u}_2, v_2 - \hat{v}_2, \ldots, u_n - \hat{u}_n, v_n - \hat{v}_n]^{\mathrm{T}}$.

2. Compute covariance matrix $\mathrm{Cov} = (J^{\mathrm{T}} \cdot J)^{-1}$.

3. Compute uncertainty of each $\boldsymbol{P}_i$ as: $\sigma_{P_j}^2 = \mathrm{Cov}_{jj}\sigma_0^2/(2n-3)$.

With the uncertainty for n points in the reconstructed model we propose the total reconstruction uncertainty as:

$$\frac{\sum_{i=1}^{m}(\sigma_{x_i} + \sigma_{y_i} + \sigma_{z_i})}{3n}. \tag{3.2}$$

As a particular case, for 2D scene models, the denominator in expression (3.2) becomes $2n$, and we omit σ_{z_i} in the numerator. This change is because in planar models we fix the value of the z coordinate to zero; thus it has no uncertainty.

Reconstruction points and camera positions are given at an unknown scale with respect to the real world. This scale also affects the uncertainty in the system. For this reason, the uncertainties in the system must always be computed to a fixed scale factor. This scale factor can be set arbitrary or calculated if we have a distance reference for the real world.

3.2 Experimental validation

To validate our approach, we perform six experiments in distinct scenarios. The first four experiments are related to bidimensional models which include experiments with synthetic and real data, and also the use of constraints on the model to solve the self-calibration problem. These experiments are described in section 3.2.1. The last two experiments are related to 3D models. The fifth experiment is the reconstruction of a scene from a sequence of five images of a model composed with cuboids, and as sixth, we apply our approach to the Oxford dinosaur dataset with 36 images and more than 4600 point correspondences. These two experiments are described in section 3.2.3.

3.2.1 Experiments with bidimensional datasets

The aim of this first experiment is to find the minimum number of point correspondences to successfully solve the camera self-calibration, pose estimation, and reconstruction from input sequences of 2D models with our approach. To estimate this number. We generate a planar model with 40 uniformly distributed random points on a plane with size 60×90 mm, and we generated three views with distinct viewpoints considering a general motion (with both, rotation and translation transformations). The selected 3D poses is a configuration that allows that most of the model points are visible in the three views (See Fig.3.2). To obtain the views, we set an image size of 640×480 pixels, $f = 1300$, zero skewness and the principal point at the center of the image. Details for the poses of the generated views are shown in Tab. 3.2. An illustration of the generated model and views are shown in Fig. 3.2.

Table 3.2: Details for the synthetic generated views.

	1st image	2nd image	3th image
θ_1	$0.0°$	$-135.93°$	$142.67°$
θ_2	$82.5°$	$84.79°$	$83.652°$
θ_3	$89.95°$	$89.95°$	$89.95°$
t_1	-53.00 mm	-52.36 mm	-29.20 mm
t_2	2.82 mm	-2.07 mm	5.08 mm
t_3	-138.78 mm	-146.36 mm	-194.91 mm

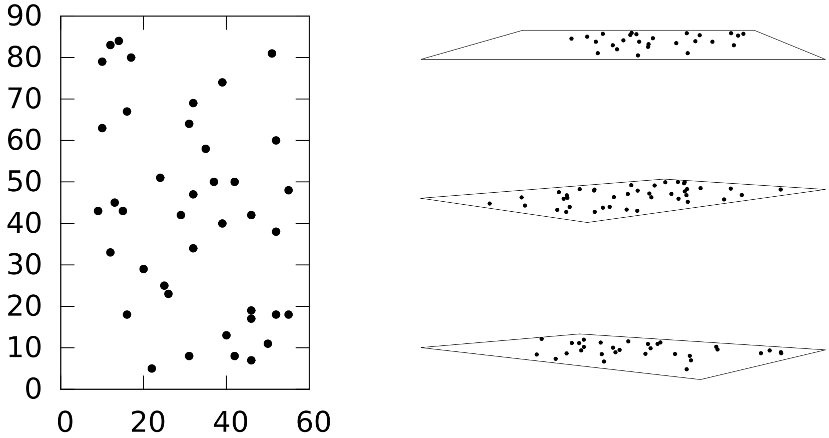


Figure 3.2: Model and three different views used to determine the minimal number of point for self-calibration with 2D models.

We performed the self-calibration using the correspondences visible in the three images, and we varied the number of correspondences used to solve the problem from 3 to 10, for each execution we selected the points correspondences to be used randomly, and we used the DE settings in Tab. 3.1. For each configuration, we performed 100 executions and we counted the successful executions. We consider an execution as successful if it allows getting the ground truth focal distance with a relative error of less than 1%. Results for the percentage of success and the median of the cost function evaluations for the 100 executions are shown in Fig. 3.3.

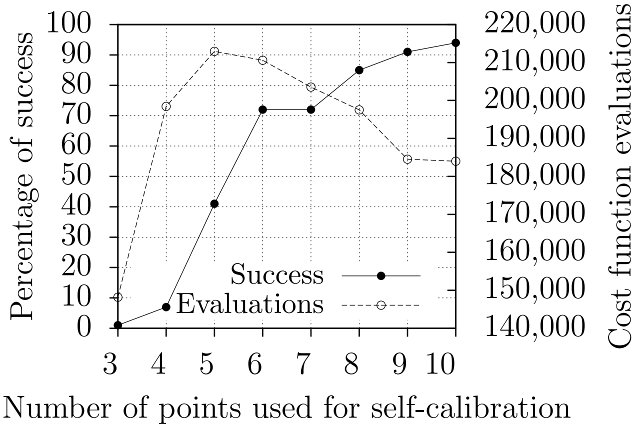


Figure 3.3: Results for camera self-calibration with planar models. Percentage of success and the median of cost function evaluations for distinct number of point correspondences.

3.2.2 3D pose estimation with planar models

This experiment aims to test our approach with the 3D pose estimation problem. We aim to determine the minimum number of points required to solve the problem and also to estimate the related cost.

We use the same dataset of the self-calibration experiment in Sec. 3.2.1, with the difference that in this pose experiment we use the ground truth planar model, the image points in the three images, and the original intrinsic camera parameters used to generate the images.

We apply the proposed approach for 3D pose estimation in Sec. 3.1.3 fixing the planar model and the intrinsic camera parameters leaving the 3D pose as unknown. We use a population size of 30, a stop criteria $s = 0.01$ and the box constraints specified in the Tab. 3.1, in page 33. For each image, we vary the number of points used to estimate the pose from 3 to 10, and for each number we perform 100 independent executions, choosing the points for each execution randomly.

In Fig. 3.4 we show the percentage of success and the median of the cost function evaluations obtained for this experiment. We consider an execution as successful if the obtained solution has a relative error of less than 3% for the six pose parameters.

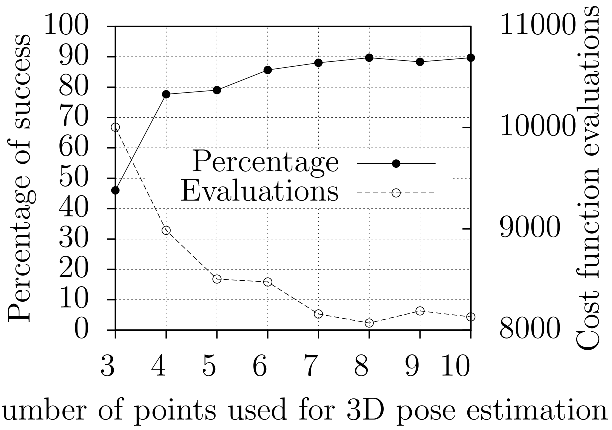


Figure 3.4: Results for 3D pose estimation with planar models. Percentage of succes and the median of cost function evaluations for distinct number of points used for the estimation.

Self-calibration in critical motion configurations

In this experiment, we tested our self-calibration approach in terms of the critical motion sequences. We use the same planar model and the same conditions, but we fixed the number of correspondences to eight, and we varied the poses of the three views. We generate images with pure translation, with pure rotation and use the views with general motion. For each configuration, we performed 100 executions and again we considered as successful cases those with 1% of relative error of the ground truth focal distance. Results for this experiment are shown in Tab. 3.3.

Table 3.3: Percentage of success of our proposed method with three different motion types.

Motion type	Percentage of success
Pure translation	2%
Pure rotation	17%
General motion	85%

Self-calibration with triangulation constraints

In this experiment, we validate our proposal for model constraints in the self-calibration problem. We consider three kinds of constraints: model points on the plane (plane constraint), model points on the plane and parallel lines (parallelism constraint) and model points on the plane and perpendicular lines (perpendicularity constraint).

To test our approach in real conditions, we use the Zhang's dataset [34], which consists of five views of a chessboard. The dataset contains the original model with 256 points and the correspondences of each model point in the five images, but for our experiments, we only use the correspondences data.

We used eight correspondences in three images to perform the self-calibration. For the planar constraint, we selected randomly eight correspondences on the plane and for the lines constraints we selected the eight feature correspondences lying on lines that satisfy the angle constraints.

For the parallelism and perpendicularity constraints, we used the proposed triangulation expressions in Sec. 3.1.2 to define two lines with values $y = 1$ and $y = -1$, for the case of parallelism, and $x = 0$ and $y = 0$ for the case of perpendicularity.

We obtain the reconstruction from the five images using our complete approach. We first perform the self-calibration using three randomly selected images and the specified model constraint, then we estimate the 3D pose for the two remaining views using four points of the reconstructed model, and finally, we perform BA to obtain a refined version of the reconstructed scene.

For each constraint, we perform 100 independent executions. In Tab. 3.4 we show the median of the relative error of the focal distance obtained in the self-calibration step and the median of cost function evaluations. In Tab. 3.5 we show the final results of applying BA to estimate the refined version of the intrinsic parameters and the RMS of the reconstructed model. The final result is the same for the three considered constraints in all cases and executions.

Table 3.4: Results for the proposed triangulation constraints. Median of 100 independent executions.

Constraint	f relative error	g_1 evaluations
Plane	7.09	153 975
Parallelism	4.16	183 200
Perpendicularity	12.25	198 125

Table 3.5: Results for the Zhang's dataset.

Parameter	Ground-truth	Proposed method
f_x	832.50	833.50
f_y	832.52	833.38
o	0.204494	0.314250
u_0 (pixels)	303.95	312.16
v_0 (pixels)	206.58	198.48
k_1	-0.22	-0.23
k_2	0.19	0.17
Reprojection RMS (pixels)	0.335	0.126
Reconstruction RMS (cms)	–	0.053

3.2.3 Reconstruction of 3D environments

In order to validate the proposed approach for the reconstruction of three-dimensional models, we perform two experiments with real data. We built a 3D model using cuboids to test our self-calibration with 3D models, and we also reconstruct the complete cuboids model from five images. Additionally, we reconstruct the Oxford dinosaur from its 36 views. Details of these experiments are described in Sec. 3.2.3 and Sec. 3.2.3 respectively.

Self-calibration with 3D models and reconstruction

For this experiment, we built a toy using cuboids, and we generate 5 views with a previously calibrated camera using the Zhang's calibration method. With the aim of estimating the minimal conditions to perform self-calibration with 3D models, we manually extract the correspondences in the five images, and we perform the self-calibration step varying the number of points used for the self-calibration from 3 to 10. We select arbitrary three images from the dataset and for each configuration, we perform 100 instances of the experiment selecting randomly the points correspondences used. The result of this experiment is shown in Fig. 3.5. We show the percentage of success considering a relative error of 5%, the median of the cost function evaluations required for a distinct number of point correspondences.

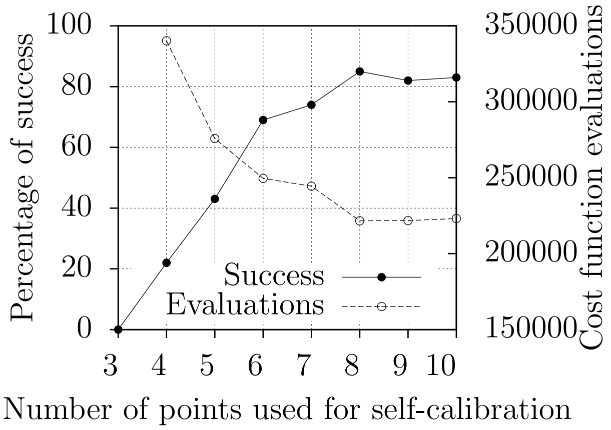


Figure 3.5: Results for camera self-calibration with 3D models.

The final result of the complete reconstruction, considering an uncertainty threshold of 1 mm are shown in Tab. 3.6. The used views and a reproduction computed from the solution are shown in figure 3.6. The superior, frontal and the lateral views from the reconstructed model are shown in figure 3.7.

Oxford dinosaur reconstruction

We use the 36 images from the Oxford's dinosaur available at [65]. We used 15 points for the camera self-calibration step. In order to set the scale, we assumed

Table 3.6: Camera parameters and reprojection RMS for the reconstructed cuboids model.

Parameter	Value	Ground truth
f_x	974.39	982.28
f_y	974.39	976.5
o	0.0	7.05
u_0	400	290.20
v_0	300	209.44
Reprojection RMS	1.66	
Uncertainty	0.22 mm	
Reconstruction points	45	

that real object height is 25 cm We used 0.5 mm as the uncertainty threshold. After applying the complete procedure, we obtain 4633 3D reconstruction points. The camera parameters, the reprojection RMS and reconstruction uncertainty are shown in Tab. 3.7. Fig. 3.8 shows one of the dinosaur's dataset image and a new view generated from the 3D reconstruction.

Table 3.7: Camera parameters results for Oxford dinosaur dataset.

Parameter	Value
f	2,788.92
o	0.0
u_0	360
v_0	288
Reprojection error RMS	0.22
Uncertainty	0.43 mm

3.3 Discussion of results

Our approach has four main advantages: 1) Features in images can be in any geometric configuration, 2) we integrate three of the main subproblems involved in the 3D reconstruction problem (self-calibration, 3D pose estimation, and model reconstruction), 3) we simplify the self-calibration step in such way that we avoid the use of traditional concepts as the fundamental matrix at the same time we simultaneously use the information of three images directly to solve the problem, 4) We directly find the solution through a non-linear optimization method which is solved by DE and this solution can be used to start BA.

We can solve the self-calibration, 3D pose estimation, and 3D reconstruction from three views, this is because if we consider the selected points on the views as the projection of the points in the model, and if we take the Fourier transform of this projection, then in the Fourier space, each projection will be an slide of the frequency

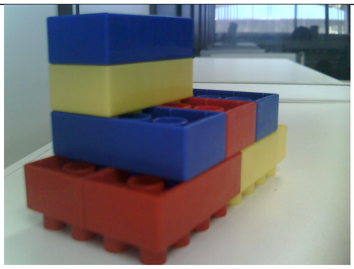
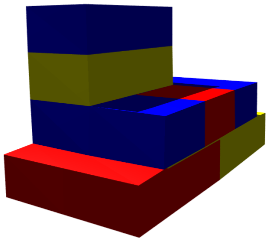
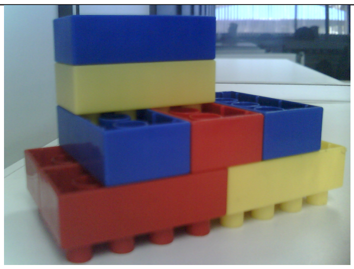
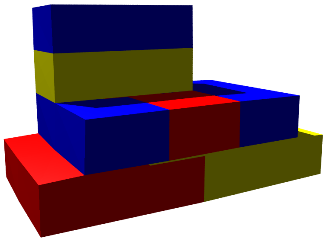
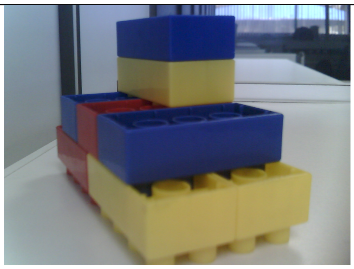
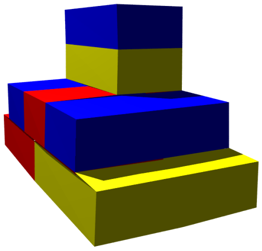
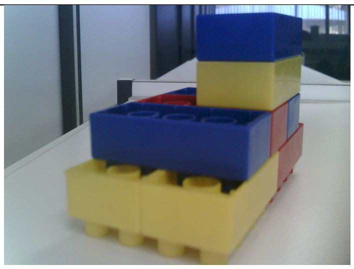
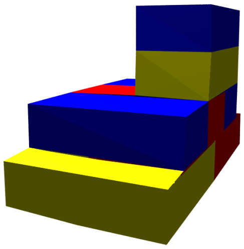
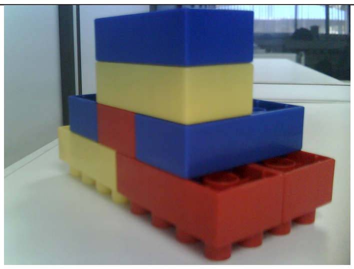
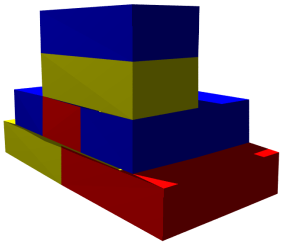
View	Image	Reproduction
1		
2		
3		
4		
5		

Figure 3.6: Results for the cuboids model reconstruction.

specter with the same orientation of the view. Using at least three projections with distinct orientations, then it is possible to fix the orientation of the three respectively slides in the Fourier space [66]. This is based is the so called Fourier Slice Theorem, which is much like Computed Tomography [67, 66].

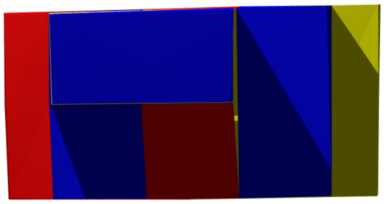
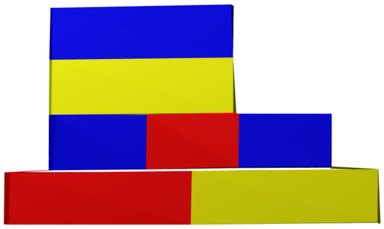
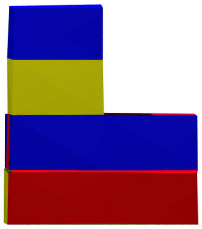
View	Image
Superior	
Frontal	
Lateral	

Figure 3.7: Lateral views for the reconstructed cuboids model.

We use the DLT algorithm for the triangulation. This method is suitable and valid for Euclidean (or metric) reconstructions, in the case of projective or affine reconstructions its accuracy is not so good [55]. Although our method seeks for Euclidean reconstructions, the evolutionary process of the DE starts by obtaining projective reconstructions that are later being evolved until converging to the Euclidean version. Taking this process in consideration, the selection of DLT algorithm does not affect the final result. At the beginning of the DE, the DLT triangulation allows us to obtain triangulation approximations that are enough to guide the evolutionary process to the Euclidean reconstruction, as long as the solutions are closer to the final result, the DLT triangulation is more accurate, and we can expect good results. Since we only use the final result, the use of DLT is valid for the proposal, and additionally it eases the use of the information of multiple images (more than

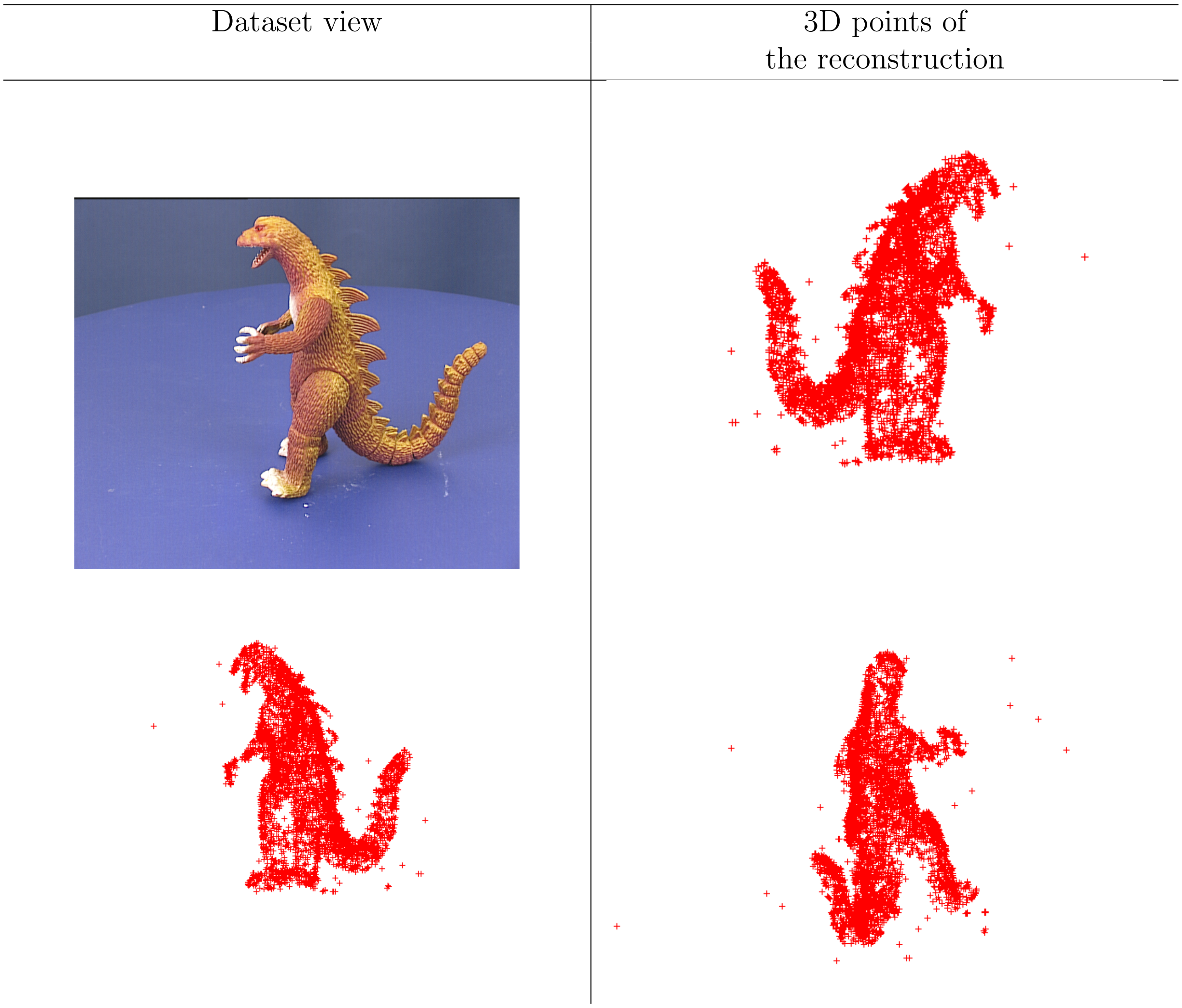


Figure 3.8: Left, instance of the images in the Oxford dinosaur dataset. Right, 3D points of the reconstructed model.

two), which helps our method to deal with noise.

From the experiments in Sec. 3.2 and Sec. 3.2.2 we observe that our approach is suitable to solve the self-calibration, pose estimation and reconstruction problems in synthetic and real datasets. In the results of Fig. 3.3 we observe that for self-calibration with planar models our approach can obtain a 72% of success with a relative error less than the 1% using only six correspondences in three images and also we observe that the number of required cost function evaluations reduces with the increase of used points. In the case of self-calibration, pose estimation and reconstruction of 3D models we observe similar results. In Fig. 3.5 we observe that we also require six points to perform self-calibration and we also observe that the number of cost function evaluation decreases as long as the number of used points is augmented.

We solve the 3D pose estimation problem using DE by minimizing the reprojection error of the obtained partial reconstruction. We solve the problem directly as a non-linear problem with DE. This strategy allows us to obtain the solution that best minimizes the reprojection error when the data contains outliers, or even when it seems to exist multiple solutions to the problem.

For the 3D pose estimation problem, in 3.4 we observe that our approach requires four points correspondences to solve the problem with a percentage of success higher to the 77% and we also observe a reduction of the cost function evaluations is the number of image points used is increased. In Fig. 3.4 we observe that with three points we obtain with our approach a 46% of success. This partially high percentage is due to the 3D pose estimation problem has many solutions when three points are used, and in some of the performed experiments, the pose is found as it is expected [22].

From the results in Sec. 3.2.2 we observe that our approach requires a general motion to have a good performance but the pure translation motion results more challenging than the pure rotation motion for our approach.

In the experiment in Sec. 3.2.2 we tested our approach with a real dataset and the proposed constraints for planar models. From the results in Tab. 3.4 we observe that the use of the proposed line constraints helps to reduce the relative error obtained in the self-calibration but increase the number of cost function evaluations required for self-calibration. These results are positive since it means that the inclusion of partial knowledge helps our approach to obtain better results and also shows that the improvement in the quality of the results come with an associated increment in the number of evaluations. This increment in the number of evaluations could be caused by the search of the additional angle in the decision vector, the search of an additional angle increases the size of the search space and this is reflected in the cost function evaluations required.

Our approach is suitable for the reconstruction of 3D models. In Fig. 3.6 we observe that our results are good for visualization purposes and also in Fig. 3.7 we observe that the angles in the reconstructed model are preserved. An important aspect to mention is that in our experiments with cuboids models we were not able to estimate the five parameters of the camera. This is because the input images do not have enough point correspondences distributed on all the images.

From the results of the reconstruction of the Oxford dinosaur experiment in Sec. 3.2.3 we observe that our approach can work in challenging conditions like camera variations and presence of outliers.

3.4 Complexity of the proposed algorithm

The algorithms used in Computer Vision in the last 30 years have been developed with the idea of obtaining first a solution by linear methods for later refine the initial solution with a non-linear algorithm [22].

The cost associated with a linear solution is the same associated with the inversion of a matrix. In practice, we do not perform matrix inversions, because the associated numerical instabilities; instead we use the QR decomposition (or the Singular Value Decomposition) since to do it that way is more stable numerically. Also, some of the linear problems can be solved through the eigendecomposition of the matrix. All those problems have an O(mn^2) complexity, where m is generally the number of point matches (or the number of equations), and n is the number of unknowns.

The non-linear algorithm finds the minimal global solution to a problem in which the target is to minimize the sum of squared errors. Typically, the most used non-linear algorithm is the Levenberg-Marquardt (LM). This algorithm can solve non-linear least squares problems, and it does make use of the second derivative of the target function.

In the literature, we find that the complexity of the non-linear algorithms is not mentioned. This fact is because, theoretically, the problem handled by the iterative solution of various non-linear problems. Instead, it is generally considered the number of iterations used by the algorithm to reach the desired precision of the answer.

The Newton method is the best non-linear method to minimize a cost function since will require the least number of iterations, but it needs to evaluate the first and the second derivative of the cost function. The Newton method is known to have a quadratic convergence, i.e., that the error in an iteration will be equal to the square of the error in the previous iteration.

The Newton method is not used for solving least square problems, instead it is used the Gauss-Newton (GN) method. This fact is because we can put aside the second derivative. The LM method interpolates between the GN and the gradient descent methods. LM presents linear convergence, but it is the best algorithm to solve the non-linear least squares problems.

In this work, we solve a non-linear problem through the Differential Evolution (ED) heuristic. The heuristics are methods that seek an approximated solution, but they can not ensure to find the global optimum.

The proposal of using ED aims to solve a non-linear problem directly. In other words, with that approach, we don't need to look first for a linear solution for later refine it with the non-linear method for least squares.

The convergence ratio for the heuristics is also linear, but they are definitely slower than the GN, in such a manner that the number of cost function evaluations can be of thousands, but they do not require to evaluate the derivative of the cost function, what is an advantage. In fact, the heuristics are recommended when we do not know the derivative of the cost function. In our proposal, we use the ED to find an approximate solution for later refine it through LM. In the performed experiments it is recommended to execute the ED at least five times, in at least one those executions we can find an enough good solution to start the Bundle Adjustment with LM.

3.5 The Newton method and its quadratic convergence

If $f(x)$ is a non-linear function, we can linearize it by using its expansion with the Taylor series:

$$f(x) = \sum_{i=0}^{\infty} \frac{(x-a)^i}{i!} f^{(i)}(a). \tag{3.3}$$

The expansion in Eq. (3.3) is performed in the neighborhood of a point a.

We require the function f; thus, we require to compute the first derivative f' and equal it to zero. If we expand the first derivative f' using the Eq. (3.3) to its two first terms (the concerning to a linear approximation), we have:

$$f'(x) \approx f'(a) + f''(a)(x-a). \tag{3.4}$$

Making $f'(x)$ equal to zero we obtain:

$$a_{n+1} = a_n - \frac{f'(a_n)}{f''(a_n)}. \tag{3.5}$$

The expression in Eq. (3.5) is an iterative expression that requires an initial point a_0 to start. This initial point a_0 is an initial solution to the problem f; but we have the contradiction that a_0 must be very near to the global optimum that we do not know priorly and we are looking for.

The expression in Eq. (3.5) is the Newton method, it requires the existence of the first f' and the second derivative f'', but even more, it requires that f' and f'' are sufficiently smooth at the neighborhood of a_0.

To demonstrate the quadratic convergence of the Newton method, we know that the minimum of the first derivative is at $f'(x) = 0$. Suppose that the root of $f'(x)$ is β, then using the Eq. (3.4) we have:

$$f'(\beta) = f'(a_n) + f''(a_n)(\beta - a_n)(\beta - a_n) + R_1,$$

where

$$R_1 = \frac{1}{2!} f'''(a_n)(\beta - a_n)^2.$$

Since β is a root of $f'(\beta)$, then:

$$\begin{aligned} 0 = f'(\beta) = f'(a_n) + f''(a_n)(\beta - a_n) + \frac{1}{2} f'''(a_n)(\beta - a_n)^2, \\ f'(a_n) + f''(a_n)(\beta - a_n) = -\frac{1}{2} f'''(a_n)(\beta - a_n)^2. \end{aligned}$$

By dividing the last expresion by $f''(a_n)$:

$$\begin{aligned}
\frac{f'(a_n)}{f''(a_n)} + (\beta - a_n) &= \frac{-f'''(a_n)}{2f''(a_n)}(\beta - a_n)^2, \\
-(a_{n+1} - a_n) + \beta - a_n &= \frac{-f'''(x_n)}{2f''(a_n)}(\beta - a_n)^2, \\
\beta - a_{n+1} &= \frac{-f'''(a_n)}{2f''(a_n)}(\beta - a_n)^2, \\
\epsilon_{n+1} &= \frac{|f'''(a_n)|}{2|f''(a_n)|}\epsilon_n^2.
\end{aligned}$$

The last expression shows that convergence for the Newton method is quadratic, we observe that the error in the $n-$th iteration is the square of the error in the iteration $n+1$, iff the following conditions are satisfied:

1. $f''(a_n) \neq 0$, for all $a_n \in I$, $I \in [\beta - r, \beta + r]$ for some $r \geq |\beta - a_0|$
2. f''' is continuous for each $\beta \in I$.
3. a_0 is sufficiently close to β.

3.6 The developed method

In the following table, we summarize the existing methods for performing a reconstruction.

Table 3.8: Possible methods for estimating a reconstruction.

Equation	Known information	The solved problem
$\lambda\boldsymbol{p} = K[R\|\boldsymbol{t}]\boldsymbol{P}$	$\boldsymbol{p}$ y $\boldsymbol{P}$	K, it is a calibration method
$\lambda\boldsymbol{p}_{ij} = K[R_i\|\boldsymbol{t}_i]\boldsymbol{P}$, for $i = \{1, 2\}$	$\boldsymbol{p}_1$, $\boldsymbol{p}_2$, $K = I$	$\boldsymbol{P}$, R_i, $\boldsymbol{t}_i$, projective reconstruction (deformated)
$\lambda\boldsymbol{p}_{1j} = K[R_1\|\boldsymbol{t}_1]\boldsymbol{P}$	$\boldsymbol{p}_{1j}$, $\boldsymbol{P}$	K, R_1, $\boldsymbol{t}_1$, self-calibration
$\lambda\boldsymbol{p}_{ij} = K[R_i\|\boldsymbol{t}_i]\boldsymbol{P}$	$\boldsymbol{p}_1$, $\boldsymbol{p}_2$ and $\boldsymbol{p}_3$	K, R_i, $\boldsymbol{t}_i$, $\boldsymbol{P}$ it is a non-linear problem

For the Tab. 3.8, $\boldsymbol{p}$ is a set of points on an image, $\boldsymbol{P}$ are points in 3D, R is a rotation matrix, $\boldsymbol{t}$ is a translation vector (R_i and $\boldsymbol{t}_i$ define the pose for the $i-$th image).

There exist many variants to the showed scheme [68], in the Tab. 3.8. The difference between a calibration method and one that performs self-calibration is that $\boldsymbol{P}$ is an exact model which represents an additional step for the calibration problem. In contrast, for the self-calibration, we require additional information about $\boldsymbol{P}$, for instance, that the points are on the plane or arranged in a grid [68].

Alternatively, we can suppose the model in the scene to be a cuboid [69]; or if we count with a picture of a plane viewed from the top, which would result in a very rough model (because we have the model at the resolution and size of the pixels in the image). In the two mentioned examples we can not recover the scale of the reconstruction, which is a characteristic of the self-calibration methods.

The proposed method used the rigidity of the scene implicitly as a constraint, i.e., the object(s) within the scene remains unchanged in the three pictures (images) that the method requires.

3.7 Study remarks

In this chapter, we presented a direct method to solve the camera self-calibration problem the 3D pose estimation and the 3D reconstruction simultaneously. We showed that the minimal conditions are at least six points to apply our approach for 2D models. We confirmed that our approach of constraining the model through the modification of the triangulation step can improve the results of the self-calibration step. We showed the applicability of our approach with synthetic data and also with real datasets for both, 2D and 3D models. Moreover, we showed that the final results of our complete proposal can obtain very similar results to those obtained with a calibration method like the Zhang's [34], which is known in the literature a very stable and accurate method for camera calibration, even though when we solve the problem without any calibration pattern.

In the next chapter, we research the joined solution of two other 3D reconstruction sub-problems, the feature extraction, and the feature matching. For that purpose, we exploit a combinatorial invariant from the Computational Geometry field that allows us to share information between non-contiguous sub-problems.

Chapter 4

Simultaneous Feature extraction and Feature Matching

In this chapter, we research about the joined solution of two sub-problems to start the 3D reconstruction pipeline, the feature extraction and the feature matching. In the 3D reconstruction pipeline, the feature extraction is the first sub-problem to solve; the aim in this sub-problem is to identify the salient image features, commonly the salient points, that could be observed through different images in the input image sequence. The extracted features are the input for the next sub-problem, the feature matching sub-problem. The features matching aims to identify the features in multiple images that correspond to a same 3D element of the scene, of course in the 3D reconstruction problem the model in the scene is unknown, but this can be undestood as the identification of the same feature observed from multiple viewpoints.

The extraction of salient points and their matching are intimately related sub-problems. In the traditional approaches, many features are detected in each of the images of the input sequence. The detection of features for every single image is performed without considering the same process for other images. In practice, it is common that many features are detected per image; insufficient points reduce the probability of finding a correct match but also many point features increase the probability for mismatched features or ambiguities, a feature detected in one image could be matched to more than one feature in another image.

We require a mechanism to encode the structure of a set of points. We are solving the 3D reconstruction; thus we are addressing the problem using multiple images. We expect that some features images in one image match the features in another image. Then, the aim is to find a concept of structure that maintains unchanged even when the feature points are affected by all the transformations involved in the image generation process. With those requirements, we research the use of Order Type (OT) as the mechanism to encode the structure of a set of points. Once the features (points) are extracted, their matching is performed using OT.

The OT is a combinatorial invariant from the Computational Geometry field, it encodes the structure a set of point C in terms of the triplets of points orientation.

The OT can be represented by the so-called $\lambda-$matrix that encodes the orientations for all the triplets in C. For each set of points C there is an OT that defines it, and it has an asociated $\lambda-$matrix. For this thesis, we see this $\lambda-$matrix as a descriptor for the set of points C.

In our original problem, we aim to reconstruct the 3D structure of a scene from images. For this specific chapter we aim to solve the feature extraction and the matching from a specific case: a new kind of fiducial markers. Then, as the first part of this study we analyze if the OT can be retrieved from images of those new fiducial markers, taken by a digital camera. When generating an image, a set of points on a scene is tranformed by rigid object transformations as rotations and translations but also by a perspective transformation, all introduced by the pinhole model in Eq. (2.1) in the page 10. Also, the obtained points are converted to pixels before obtaining an image. Then for this study we need to analyze how these transformations affect a set of points on the plane and its OT before using it for our purposes.

An OT exists for every set of points but the number of different OTs is finite, this means that every set of points belong to a certain class of OT. In the last decade the OT has been widely studied, and now a days we know the number of different classes for OT, even more, we count, from other researchers studies [1], with an instance for each OT for the set of points on the plane with carcardinality $|C| \leq 11$.

We analyze the effects of the image generation process for all OT instances of set of points with cardinality lesser or equal to eight points. Specifically we study the conditions in which we can retrieve the OT for the set of points on the plane. From this results we found that the OT behaves well when computed from images, then we propose it as a descriptor of points on the scene plane but also image plane.

The $\lambda-$matrix in its computation allows obtaining an invariant labeling of the points in C. We propose to exploit this fact for our purposes. We aim to use the invariant labeling for solving the features matching. In our study we found that although every set of points has an associated $\lambda-$matrix, not all the OTs allows to solve the point matching uniquely.

We called the new fiducial markers as Order Type Tags (OTTs). These tags aim to create an application to study them, and to validate the core proposals in more realistic scenarios, in which the features extraction and the features matching are solved in an automatic approach using both simulated ray-traced images as also real images. In this sense, we studied the proposed features for performing automatic tags identification, features matching, pose estimation, and within an augmented reality application.

This chapter is structured as follows: in the Sec. 4.1 we briefly explain the OT concept that works as the base for the proposed features this chapter, in the Sec. 4.2 present the idea of OTT and detail its computation. In the Sec. 4.3 we describe how the features matching is solved simultaneously by the OT. In the Sec. 4.5 we studied the OTT under the effects of the image generation process, we study all the processes and transformations that can affect the calculation of the $\lambda-$matrix. In the Sec. 4.6 we analyse the effects of noise and other perturbations that can affect the $\lambda-$matrix, we studied it in an analytic approach and we proposed a method to quantify the sensibility through the Maximal Perturbation concept. In the Sec. 4.7 we present how the λ-matrix relates the pose estimation sub-problem. In the Sec. 4.8 we present all the experimental validation. We analyze the stability of the OT to the image generation process, the feature matching throgh it, the noise sensibility; we performed the experiments using synthetical ray-traced images and also real images taken with a physical camera sensor. In the Sec. 4.9 we present the discussion of the results for this chapter and finally in the Sec. 4.10 we draw the remarks of the whole study.

4.1 Order Type

The OT describes a set of points in terms of its triplets orientations. The orientation of a triplet is correlated with the signed area as follows. Let $C = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \dots, \boldsymbol{p}_n\}$ be a set of n points in the two-dimensional Euclidean plane. For each point $\boldsymbol{p}_i$, its coordinates are $[x_i, y_i]^{\mathrm{T}}$. We say that the orientation of a triple $(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3) \in C$ is positive (denoted by $A(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3) > 0$) if the expression (4.1) is greater than zero, is negative (denoted by $A(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3) < 0$) if the expression (4.1) is negative, and is null (denoted by $A(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3) = 0$) if the expression (1) is equal to zero.

$$A(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3) = \det \begin{vmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & x_3 \\ 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{vmatrix}, \tag{4.1}$$

where $\det|\cdot|$ corresponds to the determinant of the matrix. $A(\cdot)$ represents the double of the area of the triangle formed by the three given points.

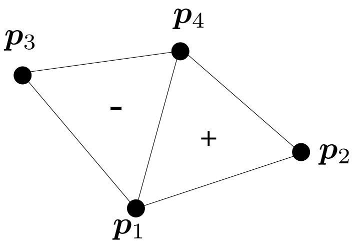


Figure 4.1: Orientation examples for two triplets. $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_4\}$ have a positive orientation since the area formed is positive. $\{\boldsymbol{p}_1, \boldsymbol{p}_3, \boldsymbol{p}_4\}$ have a negative orientation.

The OT of a point set $C = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_n\}$ is a function that asigns to each ordered triple i, j, k in $\{1, \dots, n\}$ the orientation (either clockwise or counter-clockwise) of the point triple $\boldsymbol{p}_i$, $\boldsymbol{p}_j$, $\boldsymbol{p}_k$ [1].

It is said that two sets of points C_1 and C_2 are combinatorially equivalent if they have the same OT. The OT is stored using an Order Type Representation (OTR).

The OTRs can be seen as data structures that quantify the triplets orientations. Many of them have been proposed [70] but one of the most compact is the λ-matrix.

The λ-matrix is an OTR originally proposed by Goodman and Pollack [71]. It is defined as a $n \times n$ matrix, for n points, in which for each entry $\lambda(i,j)$ determines the number of positive triples $\{\boldsymbol{p}_i, \boldsymbol{p}_j, \boldsymbol{p}_k\}$, for $k = \{1, 2, \ldots, n\}$, $k \neq i$, $k \neq j$, $i \neq j$. In another point of view, $\lambda(i,j)$ is the number of points within the set which are at the left of the oriented line that pass through points $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$. In the Fig. 4.2 we illustrate this concept; we show five points on a plane $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_4, \boldsymbol{p}_5$, a directed line is defined through the points $\boldsymbol{p}_1$ to the point $\boldsymbol{p}_2$, two points lay at the left side of the line, an only one at the right side. We are considering the points at the left, then the $\lambda-$matrix at its entry $\lambda(1,2)$ contains the number two indicating that there are two points at the left of the directed line.

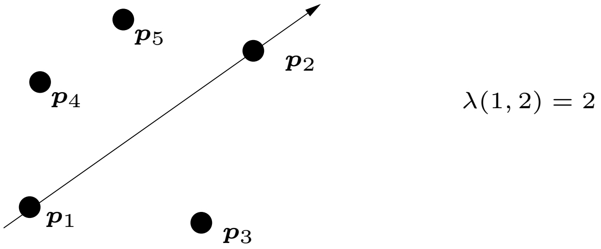


Figure 4.2: A λ-matrix entry. Considering a directed line that passes from $\boldsymbol{p}_1$ to $\boldsymbol{p}_2$, the λ-matrix entry value is the number of points that are in left side the line. This is, the number points $\boldsymbol{p}$ of a point set which satisfy the condition $A(\boldsymbol{p}_i, \boldsymbol{p}_j, \boldsymbol{p}) > 0$ (A computes the signed area of the triangle $\boldsymbol{p}_i, \boldsymbol{p}_j, \boldsymbol{p}$).

An important aspect to mention is that the λ-matrix depends on points labeling. Two different labellings of the same point set will correspond to two different λ-matrices. Although λ-matrix is sensible to point set labeling, the OT is not [71].

For this thesis, we require to obtain the same $\lambda-$matrix even when we do not count with labeled points. A naive form to handle this would be to try with each of the possible labellings, computing the associated $\lambda-$matrices and selecting the minimal matrix in the lexicographical order. This approach would not be viable since there exist $n!$ possible labellings. Instead, to avoid computing and ordering all those $\lambda-$matrices, we use the canonical ordering proposed in [71]. The canonical ordering is method to obtain a reduced number of labellings (order of points), in such a way that we can compute a minimal $\lambda-$matrix from that reduced number of labbellings possibilities. With the use of the canonical ordering we can compute a $\lambda-$matrix per each point on the convex hull set; thus, in the worst case, we will get n $\lambda-$matrices when all points in C compose the convex hull. Once the minimal lexicographical $\lambda-$matrix using the canonical ordering is obtained we will count with the set of points, the $\lambda-$matrix that describes the structure of it, and at the same time an invariant labeling.

Then, in the Sec. 4.2, we detail the method for computing the minimal $\lambda-$matrix for a set of points and the invariant labeling. In the Sec. 4.3 it is described how these elements are used to solve the point matching simultaneously.

4.2 Feature extraction

The feature extraction is one of the crucial aspects to solve in most of the pattern recognition problems and digital signals processing. They are also the case for the 3D reconstruction. We focus on the salient points on the images. In general terms, the features we will work with, are a set of points and the relationship between them, i.e., the structure that the points form.

The feature extraction is one of the crucial aspects to solve in most of the pattern recognition problems and digital signals processing. The reconstruction process described in this chapter is applied over the specific case of the new OTTs.

As it had been mentioned, we are working with points, then it is necessary to identify points on images of an OTT. To draw points over a imagen a geometrical object in the plane must be used to identify those points. Circles could be used but these have the problem that the centroid of an ellipse is not equal than its center [72, 73, 74] and a projected circle is an ellipse. Therefore triangles were choise as the geometrical object to identify, being the position of their vertices the points to identify on the images. The structure of OTTs and the image processing steps to identify the triangles vertices will be explained in Sec. 4.4, within the context of the new OTTs.

4.3 Feature matching

Each set of points has an associated $\lambda-$matrix. We have the hypothesis that this $\lambda-$matrix maintains unchanged even when all the transformations involved in the image generation process are applied. Then, the idea for the feature matching is to obtain the coordinates for a set of points C from an image with its associated descriptor, i.e., $\{C, \lambda(C)\}$, where $\lambda(C)$ is the $\lambda-$matrix for C. Then, given a set of points on another plane we can obtain the same features. We propose that the same features will be obtained for the same set of points if C is observed in two or more different images from different viewpoints, but also if it is computed directly from the model; these two cases are illustrated in the Fig. 4.3.

From the OT viewpoint, if two point sets C_1 and C_2 have the same λ-matrix and thus the same OT, C_1 and C_2 will be combinatorially equivalent.

In this chapter we say that two sets of points C_1 and C_2 correspond, i.e., they are the same set of points observed from different viewpoints, if their descriptors ($\lambda-$matrices) are equal.

The $\lambda-$matrix depends on the labeling, and there exist $n!$ possible labellings. To handle this, in [71] Goodman and Pollack propose to consider the minimal λ-matrix in the lexicographical order from those obtained with the canonical ordering. The canonical order is a method to label the points in a set of points on the plane; it

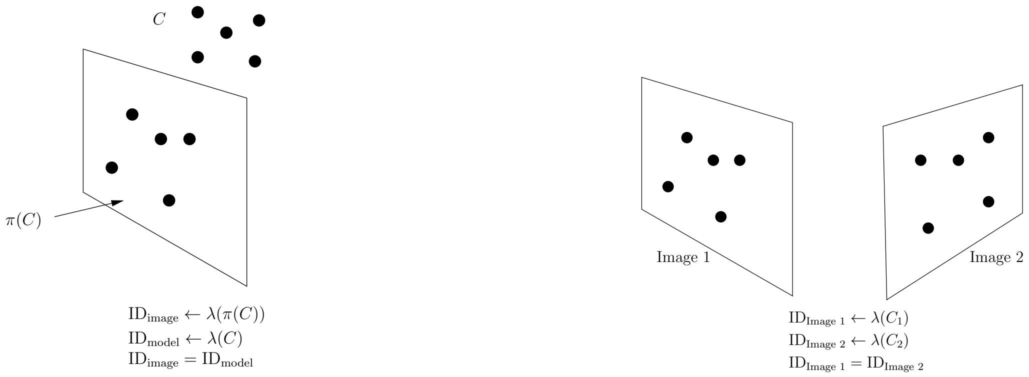

C
$\pi(C)$
$\text{ID}_{\text{image}} \leftarrow \lambda(\pi(C))$
$\text{ID}_{\text{model}} \leftarrow \lambda(C)$
$\text{ID}_{\text{image}} = \text{ID}_{\text{model}}$
Image 1
Image 2
$\text{ID}_{\text{Image 1}} \leftarrow \lambda(C_1)$
$\text{ID}_{\text{Image 2}} \leftarrow \lambda(C_2)$
$\text{ID}_{\text{Image 1}} = \text{ID}_{\text{Image 2}}$

Figure 4.3: Left: The same features computed from the model and and image. Right: the same features computed from two images with different viewpoints.

generates a reduced number of labellings, specifically, it proposes one labelling per each element on the convex hull. For a given set of points C, the canonical order first computes the convex hull of the input $\text{conv}(C)$, then each element on the convex hull is used as the reference (central point) for performing the circular ordering of the points in C. The circular ordering fix a point as the reference for the rest of the points, then the rest of the points are labeled in a counterclockwise order. The canonical order pseudocode is shown in Algorithm 2 and illustrated in Fig. 4.4.

```
Require: A point set $C$
Ensure: All canonical orderings of $C$
1: Compute convex hull $\text{conv}(C)$ of $C$.
2: Chose a point on $\text{conv}(C)$ and label it as the firsts element $\boldsymbol{p}_1$.
3: Sort $C$ circularly counter clockwise from the starting point $\boldsymbol{p}_1$. The remaining
   points will be labeled according to the circular order. If three point are collinear
   (their signed area is zero), the next point in the order will be the one with lesser
   Euclidean distance.
4: Go to step 2 choosing the next point in the convex hull as $\boldsymbol{p}_1$ to get other canonical
   ordering.
```

Algorithm 2: All canonical orderings computation

Since we can choose $m = |\text{conv}(C)|$ different initial points as $\boldsymbol{p}_1$, there will also be m labellings. One of the most famous algorithms for computing the convex hull is Graham's algorithm [75]. The algorithm requires first to order the points in the set for later identifying those that belong to the convex hull set. For the ordering step, Graham's algorithm performs a circular ordering. For this thesis, we take advantage of Graham's points sorting for computing the canonical orderings.

We can determine if two sets of points, A and B, have the same OT by computing their minimal lexicographical $\lambda-$matrices. If $\lambda(\cdot)$ represents the function that computes the minimal lexicographical $\lambda-$matrix, then, if $\lambda(A) = \lambda(B)$ we can say that A

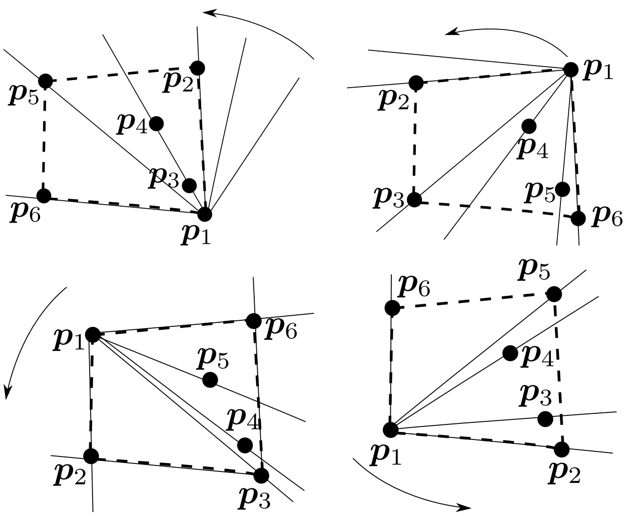


Figure 4.4: The four canonical orderings for $C = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_6\}$.

and B have the same structure, and we can consider them as the same set observed from different viewpoints.

Algorithm 3 computes the minimal lexigraphical $\lambda-$matrix. This algorithm computes only once the $\lambda-$matrix through the Algorithm 4, the other asociated matrices are premutations of the first one. Also is not necessary to calculate all the elements of the $\lambda-$matrix because $\lambda_{ji} = n - 2 - \lambda_{ij}$, therefore $\lambda-$matrix could have a triangular form.

In this section, we mention that we have the hypothesis that the proposed features, the points, the $\lambda-$matrix, and the associated labeling, maintain invariant to the transformations involved in the image generation. We believe this since if we have in a scene or a model points in one of the sides of a defined line, in the projection points an lines to an image with perspective, the points do not cross the lines either. Of course, this hypothesis must be validated, and for that reason, we study the case of a new type that fiducial markers with real image generation conditions (as limited resolution camera sensors, acquisition errors, and noise). We present an analysis of these aspects in Sec. 4.8.

4.4 A new kind of fiducial markers based in Order Type

Based on the proposal of using the OT as the features for the vision tasks, we propose a new kind of visual fiducial tags. The visual tags are elements that can be artificially included in a scene to ease computer vision tasks as automatic objects identification, camera calibration or pose estimation. In the literature, we can find many proposals for visual fiducials. Most of them are build as high contrast square models, but of course, some proposals also exploit the use of colors. The fiducials tags include

```
Require: a point set $C$
Ensure: The associated $\lambda$-matrix to $C$.
Read all the elements of $C$ in the list $L_p$.
Find the point with minimum $y$ coordinate in $L_p$.
If there are several points with that minimum $y$ coordinate value,
find in $L_p$ the one with minimun $x$ value.
This will be the reference point $\mathbf{p}_1$, but
will be identified with its index $min$, this is $\mathbf{p}_1 = L_p[min]$
Copy the list, $L_1[0] = min$; $j = 1$, $L_1[j] = i$ for $i \in \{1, \ldots, n-1\}$ and $i \neq min$
Sort circularly $L_1$
$M_0$ =computes_Lambda_matrix($L_p, L_1$)
$L_{\text{hull}} \leftarrow$ ConvexHull($L_p, L_1$)
$k = |L_{\text{hull}}|$. There are $k$ points in the convex hull.
$L_2 = L_1$, list $L_2$ copy of indexes in list $L_1$
flag = 0
for $(i = 1; i <= k; i{+}{+})\{$
    Copy the list, $L_2[0] = L_{\text{hull}}[i]$; $l = 1$, $L_1[l] = j$ for $j \in \{1, \ldots, n-1\}$ and
    $l \neq L_{\text{hull}}[i]$
    Sort circularly $L_2$
    for( $j = 0; j < n-1; j{+}{+}$ ) {
        for( $j2 = j+1; j2 < n; j2{+}{+}$ ) {
            $M_1[j][j2] = M_0[L_2[L_1[j]]][L_2[L_1[j2]]]$
            $M_1[j2][j] = M_0[L_2[L_1[j2]]][L_2[L_1[j]]]$
        }
    }
    if( $M_1 < M_0$ ) { // compareMatrices($M_0, M_1$)
        $M_0 = M_1$
        $L_2 = L_1$
    }
    else if ( $M_1 == M_0$ )
        flag = 1 // Matrices are equal! There is not a minimum $\lambda$-matrix
}
$M_0$ is the minimun lexicographical $\lambda$-matrix if flag is equal to 0.
```

Algorithm 3: Algorithm to calculate the minimal lexigraphical λ-matrix of a point set

information to the scene that it is easy to read using a camera sensor, and they are very used by robots to perform visual odometry and objects identification. Nowadays, the most common visual fiducial tags are the bases on binary patterns. These kinds of tags are given as a grid of squares in which each of the cells can have a binary tone, black or white. In this kind of tags, we find the ART tags, April tags, and others. One of the main characteristics that define them is that they are defined in a square model. When this kind of tags are observed with a camera, their edges, and in general

```
Require: The list of $n$ points $L_p$ and the list of indexes for a circular order
Ensure: The $\lambda-$matrix
    for $(i = 0; i < n - 1; i{+}{+})\{$
      for $(j = i + 1; j < n; j{+}{+})\{$
        counter= 0
        for $(p = 0; p < n; p{+}{+})$ {
          if $(p == i || p == j)$ {
            continue
          }
          if( $A(L_1[i], L_1[j], L_1[p]) > 0$ ) { // A calculates the signed area
            counter++
          }
          $M[i][j] =$ counter
          $M[j][i] = n - 2 -$ counter
        }
      }
    }
```

Algorithm 4: computes_Lambda_matrix(L_p, L_1)

their geometry is affected by the perspective effects. With the aim to make the tag readable, an image rectification must be solved first. The rectification is performed by the estimation of a homography between the square model and the corners of the tag observed on the image plane. Once the image is rectified, for each of the tags cells, again with a previously defined location is analyzed to obtain the assigned binary tone. The complete tag defines a binary string that must be processed for error correction to finally identify the tag from a dictionary of valid binary strings.

The other and less explored kind of tags are the projective invariant visual fiducial tags. This kind of tags are also used for automatic identification and pose estimation, but they distinguish from the binary in that they can be read and identified without the need of the image rectification. In the literature, we find that the authors propose the projective invariant tags to be a delimited area, which has not to be mandatory a square, as in the case of the binary tags, with some points or circles inside (usually four). In the projective invariant tags, the relationship among the contained points is the structure that identifies each of the different tags. Projective invariant tags have some advantages over the binary approaches. First, they can be identified using the points positions directly without any rectification or normalization, which provides a faster identification process. The form for the border of the tag is not limited to squares theoretically. Some of the disadvantages of the projective invariant tags are that most of them are based on the cross ratio concept. This concept, also known as the double ratio is a real number associated to a list of four collinear points which is computed from the ratio between the relative distances among the points. One of the disadvantages of the tags based in this concept is that, since it is a real number, the number of different tags that can be designed is unclear and most of the times it depends on the application, and it is also very sensitive to noise. Moreover, this kind of tags requires training or characterization to identify the obtained raged for the cross ratio that can be obtained from a set of different tags when detecting them with a camera.

In this section, we describe the proposed tags. We first describe the proposed structure for OTTs, i.e., its elements and their organization, and then we describe the process and sub-processes involved in their automatic identification.

The OTTs are composed of three main elements: the quiet area, the data area, and the tag points, an instance of them is shown in Fig. 4.5. The quiet area is a white region that contains all other elements of the tag; its purpose is to help the data area to be detected completely since it serves as a separator for data area and other objects in the scene seen in the image. The data area is a black square that simultaneously works as a finding pattern. Its purpose is to serve as the object that is easiest to identify (it is supposed to be the most prominent black object on the scene) at the same time it serves to delimit the image segment where the tag points are contained. Tag points constitute the point set C, and their arrangement defines the OT and the $\lambda-$matrix (ID). We propose to define the set of points as triangles vertices to ease their identification from the image. The use of triangles allows computing their vertices positions at sub-pixel precision.

For defining triangles in OTTs we take a point set from the database, and we manually define triangles by looking to use the least number of triangles but assuring that all points are used as the vertex of at least one triangle.

The process to detect and identify OTTs consists of three phases: the potential OTTs detection, the point set estimation, and the ID computation. These three phases are explained in the subsequent sub-sections.

Detection of potential Order Type tags

In this stage, we identify from an image those segments that could potentially contain a valid OTT. We first adequate image to make the potential Order Type Tags (POTTs) easy to detect in the image. We convert images in color to grayscale [76], and we apply Otsu's thresholding method for binarization. In the binary image, we look for the groups of black connected pixels.

We aim to reduce the false positives that can be caused by small pixel groups or noise, for this reason we only consider those objects that area is greater than a threshold. Not all objects detected in this phase correspond to the data area of an OTT, but the point set estimation phase allows to reject those do not have the OTT structure.

Features extraction from the Order Type tags

In this stage, we want to find all the points that compose the point set. First, we detect all the triangles that are present in the data area of the OTT. For this, using image processing thecniques, we apply morphological operators. First we apply the

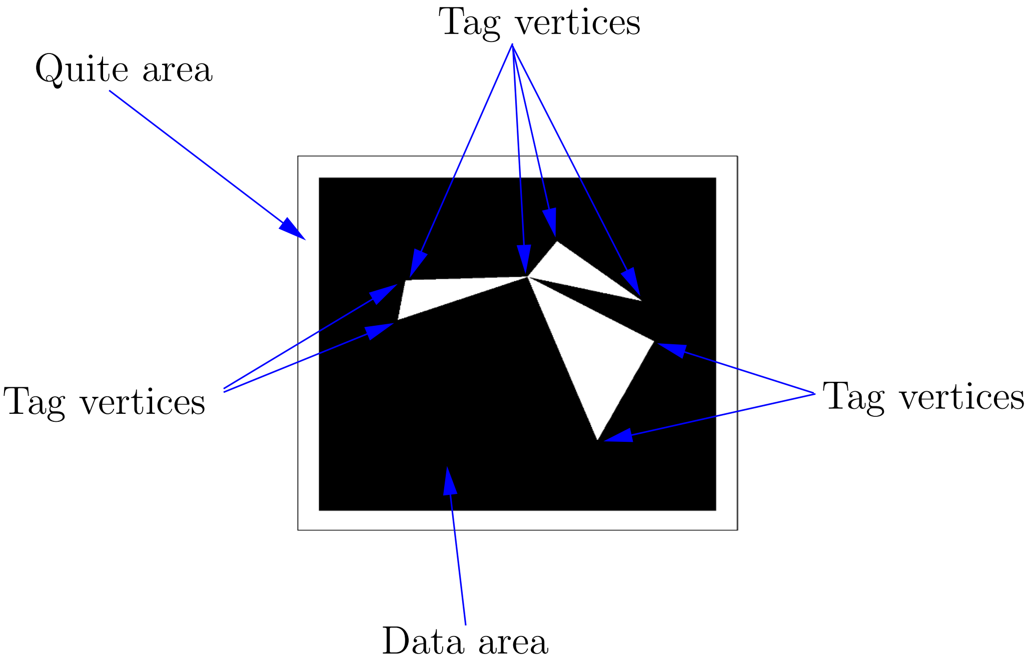
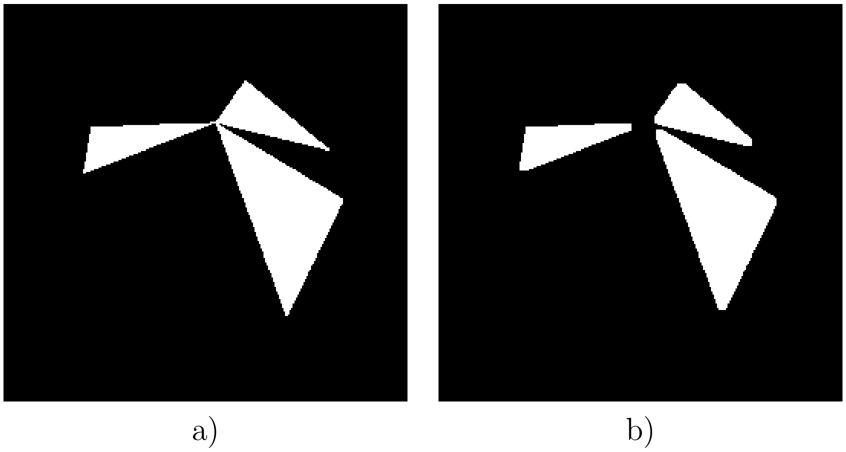


Figure 4.5: Instance of the proposed OTTs and its anatomy. The tag was constructed using the 3th point set with cardinality 7 of the database in [1]. Point set $C = \{[206, 159], [214, 127], [176, 49], [42, 144], [47, 175], [129, 178], [149, 206]\}$.

opening operator, what allows us to eliminate small groups of white pixels, that could be noise, but also it allows us to separate triangles in the OTT that share vertices. We show the effects of this operator in the Fig. 4.6. After this step, we consider the remaining objects as potential triangles.

Once we detect the triangles, we extract the vertices for each triangle. We perform the analysis of each triangle independently, i.e., we treat each POTT as a separate binary image. Before processing a triangle we need to return the triangles to their initial size, we perform this by applying the inverse morphological operator, the dilatation. With the dilatation we make the triangle to return to its initial size with the aim to estimate its vertices.

We consider as triangles those white pixel objects with the highest area inside the POTT. Then, for each of the detected triangles, we estimate their vertices. For this, we look for the three perimeter pixels that enclose the highest area. This approach,allows us to obtain a good estimation most of the times, however due to the effects of the morphological operators, the precision of the obtained vertices is not so accurate and we require to refine the estimation. Using this first vertices approximation, we segment the perimeter pixels in three chains, each chain corresponding to the pixels on a triangle's side, as shown in the Fig. 4.7. For each pixels chain, we apply linear regression using Principal Component Analysis algorithm, and we compute triangle vertices as the intersection of two lines as shown in figure 4.8. This strategy allows us to obtain the vertices at subpixel precision.



a) b)

Figure 4.6: a) A POTT before opening operator (Join triangles). b) A POTT after opening operator (Separated triangles).

When two or more triangles in the tag share a vertex, we can obtain multiple estimations for the same vertex; this is because we estimate the shared vertex position once for each triangle that shares it. To handle this, we require a method to obtain a unique estimation of the shared vertices, then, we cluster the estimated vertices fixing the point set cardinality as the number of clusters to find, in this sense, a cluster with more than one element will be represented by the centroid of the elements that compose it.

In this stage, we can detect if an OTT does not comply with the expected structure. We can detect this case when the number of estimated vertices differs to the point set cardinality, in that case, we reject the POTT and the process, and we do not require to compute the tag identification.

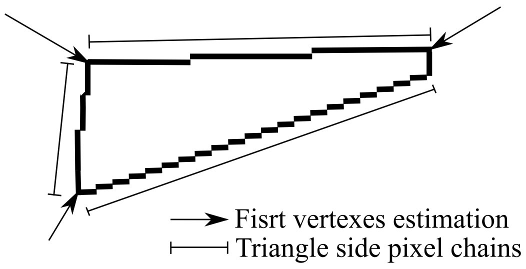


Figure 4.7: Triangle vertices first estimation and side pixel chains.

Automatic tags identification

In the visual tags, each tag counts with a predefined identification number or string. The sets of identification numbers compose a dicionary in such a way that every valid tag has an associated ID from the dictionary. For the OTTs, this ID is the $\lambda-$matrix. As mentioned in the Tab. 2 in the page 10, the number of different OTs that exist for

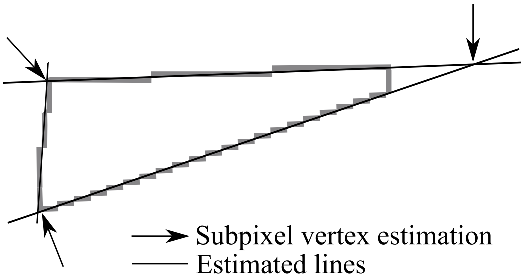


Figure 4.8: Estimated lines and subpixel vertices estimation.

the set of points with cardinality 7 is 135, and for those with eight points, it is 3 315. Then, we can build two dictionaries, depending on the number of points that we use as triangles vertices. Once we extract the vertices of the triangles for a POTT,i.e., the set of points C, we just need to compute the associated $\lambda-$matrix.

For this thesis, we use the proposed OTTs to validate the features extraction, and the simultaneous point matching using an automated approach and real images. We validate the proposed features in this chapter and the OTTs in the Sec. 4.8.4; there, we evaluate the OTTs in different conditions of tilt and distance. In this section, we designed and built the OTT shown in the Fig. 4.5 manually, but we followed a specific methodology; we describe this methodology in the appendix A.

In following sections we use the proposed visual fiducial tags to evaluate the proposed features in realistic conditions. As described in this section the automatic features extraction from the OTTs require of additional processing before computing the $\lambda-$matrix from the encoded set of points. These additional processing, morphological operators, vertices estimation, clustering, introduce additional sources of noise. It is interesing to test our proposal of feeatures in realistic scenarios. In the Sec. 4.8, we present the validation experiments using the OTTs.

4.5 The images generation process in the new fiducial markers

We are exploiting the OT in the 3D reconstruction problem; for this purpose, we need to correctly identify the OT from a set of points on the image or a reconstructed plane. For the case of a set of points on the image, we need to remember that, the points on the image are the result of the scene projection to the image plane. This projection comprises multiple linear and non-linear transformations. These transformations are 3D rotations introduced by the rotation matrix R in the Eq. (2.1) at the page 10, translations introduced by the vector $\boldsymbol{t}$ in the Eq. (2.1), the projective transformations associated to the perspective introduced by the K matrix in Eq. (2.1) and the pixels generation result of the limited resolution of the camera sensor. Along with the transformations involved in the image generation process, we also need to consider the effects of noise.

If we are able to estimate the OT from images correctly, then we can consider the OT as the tool for the proposed features in this chapter. First, we need to consider the previous studies about OT. The OT was initially proposed by the Goodman and Pollack in [71] in 1983. The authors introduced the general idea of the order and the $\lambda-$matrix, but at that moment it was unknown with precision the number of different OTs that exist for points on the plane. In [1], Aichholzer et al. in 2002 studied in deep the problem and found the number of different OTs for different pointsets cardinalities, even more, the authors also provide an instance, a set of points, for each OT, i.e., an OT instance.

In the Tab. 4.1 we show a review of the different existing OTs. In the table, we see that the number of OTs grows exponentially with the cardinality of the sets. For instance, for sets of points with eleven points there exists more than two billion OTs.

Table 4.1: Oswin Aichholzer *et al.* Order type database summary.

Set	$\lvert C^k \rvert$ = Number of OTs
C^3	1
C^4	2
C^5	3
C^6	16
C^7	135
C^8	3315
C^9	158 817
C^{10}	14 309 547
C^{11}	2 334 512 907

In [1], each OT instance is defined on a plane, and each of the compounding points is given as point positions on the plane as a pair of integer coordinates. Depending on the cardinality of the set, the integer coordinates are given with different precisions. For those sets of points with cardinality lesser than eight, each point is given with eight bits of precision. This can be interpreted as that every OT instance with cardinality lesser or equal than eight can be drawn in a square with 2^8 rows and 2^8 columns, i.e., 256×256. In the case of the OTs with nine, then and eleven points every point is given with 16 bits of precision, i.e., every OT can be drawn in a square of $65\,536 \times 65\,536$ dimensions. As early stated, the camera sensors have a limited resolution, for instance, a 36 Megapixels sensor allow to obtain images with a resolution of $6\,144 \times 4\,912$ pixels, which is still very far from the minimal resolution required to determine all the OTs from images taken with such a sensor. For this reason, and for practical purposes, in this thesis, we focus on the analysis using OTs with cardinality lesser or equal than eight, which means that theoretically could be acquired with a Half-size VGA (480×320) camera sensor or greater.

For validating the proposal of considering the OT as the features for this thesis we propose to test the effects of the image generation process with each OT instance exhaustively. We use the OT instances for validating our proposal since from the results in [1] we count with an instance for each OT, then we can test our proposals for each instance. The details of this experimental validation is described in the Sec. 4.8.

4.6 Order Type robustness to noise

As another mandatory analysis of the proposed features, we want to analyze the sensibility of OT to noise. Here we consider noise to the undesired displacements that a point in a set of points C could suffer. These displacements can be the product of image error acquisitions, as it is common to occur in real-world applications. It could also come from the pixels generation; which is the result of the finite nature of the camera sensor. Both cases are considered in the general analysis approach that we propose.

To analyze how the OT can be affected by noise, we need to observe how the OT is defined, and in which conditions it can be changed.

The OT is computed from the orientations of the triplets that compose a set of points C . The orientation of all the triplets, define the OT; thus the OT will no be affected, changed, as long as the triplets orientations are maintained. In the minimal case, the OT will change if at least one of the triplets change its orientation. With this idea, we can determine how sensible is an OT instance if we determine how difficult is to change the orientation of at least one of its triplets. A set of points is composed of multiple triplets, but we can evaluate the sensibility of the OT instance if considering the triplet that is more propense to change its orientation. In this direction, we define the concept of the Maximal Perturbation value. We define the Maximal Perturbation value of a set of points C as the maximal displacement that any point in C can have in any direction without changing the OT of C. This concept is illustrated in Fig. 4.9, but it is defined as the half of the distance between a point $\boldsymbol{p}_i$ to the closest line defined by two points $\boldsymbol{p}_j$ and $\boldsymbol{p}_k$, with $i \neq j \neq k$. The algorithm for computing the Maximal Perturbation value is defined in Alg. 5, where d corresponds to the Euclidean distance between a point $\boldsymbol{p}_c$ and the line $\overline{\boldsymbol{p}_a\boldsymbol{p}_b}$ through the points $\boldsymbol{p}_a$ and $\boldsymbol{p}_b$.

The MP is different for each set of points, and this is true even for different sets of points with the same OT. Then, we analyze the MP for each OT instance with cardinality lesser than eight. The details and the results of this experiment are shown in Sec. 4.8.2.

4.7 Computing the pose from the matched points

The pose camera pose estimation, also known as extrinsic camera calibration consists in estimating the relative movement between a scene and the camera. The pose is defined by a 3D rotation, which can be defined indistinctly as a rotation matrix, three Euler angles or quaternions, and a translation vector in $\mathbb{R}^3$. To perform the single pose estimation, we must count with a set reference. The reference establishes

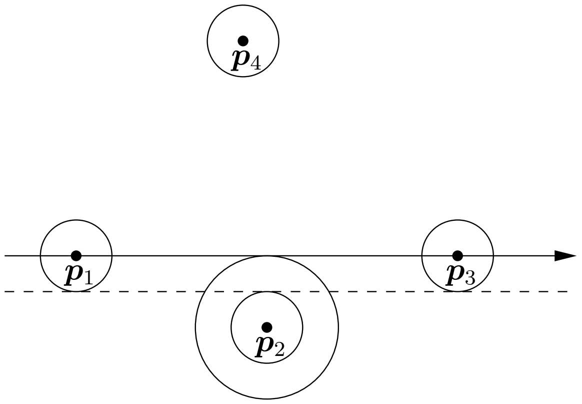


Figure 4.9: Four points on a plane, the bigger circle radius is the orthogonal distance from point $\boldsymbol{p}_2$ to the line through $\boldsymbol{p}_1$ and $\boldsymbol{p}_3$. $\lambda(1,3) = 1$ and points can be in any position inside the small circles without changing the OT. If both points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, or points $\boldsymbol{p}_2$ and $\boldsymbol{p}_3$ cross simultaneously the dashed line, the OT changes (it becomes a triangle with another point inside).

```
Require: A set of points $C_l^k = \{\boldsymbol{p_1}, \boldsymbol{p_2}, \boldsymbol{p_3}, \dots, \boldsymbol{p_k}\}$.
Ensure: $\mathrm{MP}(C_l^k)$
$d_{\min} = -\infty$
for all pair of points $\{\boldsymbol{p}_a, \boldsymbol{p}_b\} \subset C_l^k, a < b$ do
    for each point $\boldsymbol{p}_c \in C_l^k, a \neq b \neq c$ do
        Let $\boldsymbol{z} = \boldsymbol{p}_b - \boldsymbol{p}_a$, $\boldsymbol{w} = \boldsymbol{p}_c - \boldsymbol{p}_a$, and
        $d = ||\boldsymbol{z} - \boldsymbol{z} \cdot \frac{\boldsymbol{w}}{||\boldsymbol{w}||}||$
        if $d < d_{\min}$ then
            $d_{\min} = d$
        end if
    end for
end for
return $\mathrm{MP}(C_l^k) = d_{\min}/2$
```

Algorithm 5: Pseudocode for the maximal perturbation of a set of points $\mathrm{MP}(C_l^k)$.

the position of the objects that compose the scene in a general coordinate system, including the camera. In our case, this reference is a model.

To estimate the pose there exists multiple estimation methods. We can organize them in closed form or iterative methods. For the case of planes, Zhang and Faugeras propose homography decomposition methods, and a more recently we find the Infinitesimal Pose estimation, this later considers the existence of local minima in the error function details of this issue are detailed in the Apendice 1, where we analyze and reproduce the problem. These methods for pose estimation from planes can be solved using at least four points correspondences between the model and the image.

For the case of 3D objects we can apply matrix decomposition, but also the Perspective n Points (PnP) approach. In first we estimate a transformation matrix by the DLT algorithm for later perform the homography decomposition. The PnP approach involves the solution to a system of polynomial equations that lead to multiple solutions that have to be then discarded to find the correct.

In all the mentioned methods, to apply the techniques the search of correspondences must be first solved, then, this section we describe how the OT is used to directly address the matches between the features on an image and its associated model. As early mentioned the set of points with no three collinear points, has an associated OT. In a given scene with a model, a set of points, and the model seen in an image we want to be able to determine which of the points on the model corresponds to what point on the model.

To solve the problem, we use the invariant labeling involved in the $\lambda-$matrix computation. Once the points matching is performed, we can apply any of the existing methods for estimating the pose.

4.8 Experimental validation

In this section, we aim to validate the features extraction proposed in this thesis. As mentioned early in the Sec. 4, we are considering as features a set of points from an image accompanied by its associated $\lambda-$matrix and its invariant labeling. For the experiments in this section we build many synthetic scenes, i.e., a model composed of a set of points on the plane and a camera directed to the model to take a shoot and generate an image. We know apriori that the set of points on the model has an associated OT, then the task in this section is to verify if we can retrieve successfully the same model OT but from images with perspective and all the transformations involved in the image generation process. To ensure the proposal will work in a general way, we, fortunately, count with an instance for each OT from work in [1]; then we can perform our experiments for each OT instance; thus, our analyses are presented in an exhaustive but general approach.

We performed five experiments; first in Sec. 4.8.1, we generate synthetic images from multiple viewpoints (different rotation angles, tilt, and distances) using each OT instance directly as the model. This experiment aims to verify that our approach applies to all the existing OTs, but if not, we want to know the cases or the conditions in which it is not.

In the second experiment in Sec. 4.8.2, we analyzed the sensibility of OT to the noise, we analyze it through the analysis of the maximal perturbation, in this experiment we find that the OT instances support different level of noise. Retrieving the OT from images involves that we will obtain the features points with additive noise but also discretization errors associated with the pixels generation. For this reason, it is essential to analyze how sensible is the OT to noise. For this purpose, we computed the maximal perturbation for every OT instance that we used in the previous experiment. From this study, we find that every OT instance supports a different amount of noise.

In the third experiment in Sec. 4.8.3, analyzed the direct feature matching using the extracted features. The features based on OT allows solving the point matching directly, this is possible when there exists a unique minimal $\lambda-$matrix. In this experiment, we study this aspect, and we found that not all the OT instances are suitable for determining the matching between the features; we quantified this aspect, and we found that most of the OT instances allow the matching directly.

As the fourth experiment in Sec. 4.8.4, we test the features in a more realistic scenario. We generate ray-traced images using a proposal of visual fiducial tags that are based on OT. This test aims to verify how the features can be extracted in realistic conditions in an automatic extraction approach. We performed the experiments in different tilt conditions and also in different distances between the tags and the camera.

As a final experiment in Sec. 4.8.7, we test the automatic features approach. The proposed features allow to perform the identification of a plane automatically and to solve the feature matching directly. The solution to these both aspects allows solving the pose estimation between a model in the scene and an image, which is enough information for calibrating a camera and implementing augmented reality. We show an implementation that uses the proposed features to perform augmented reality.

4.8.1 $\lambda-$matrix extraction from synthetic images

For this experiment, we aim to verify that we can extract the features, i.e., the set of points and its associated $\lambda-$matrix from a set of points that are projected into images. To validate it, in a synthetical approach, we simulate the image generation process, which in essence involves the simultaneous effects of:

- the 3D rotations and translations of the camera (or the objects),
- the projective non linear transformation caused by the perspective introduced by the camera (related to the focal distance) and its obliquity,
- and the pixel generation created by the finite resolution of the camera (discretization errors).

Since we count with an instance of each OT, then we can perform our analysis in the exhaustive approach, this fact ensures that we can test the proposed method with all the OTs and in many of the possible conditions, scenes, that can occur when extracting the features from images. For this experiment we use the OT instances with cardinality seven and eight, this is because these groups are represented by 1 byte per point coordinate, this is, it is necessary a resolution of 256 × 256 pxels that can be recognized with a typical low-cost camera sensor of 640 × 480 pxels, and also because these groups have enoght different OT instances,, i.e., 135 for C^7 and 3 315 for C^8.

$\lambda-$matrix extraction in different rotations and tilt conditions

For each OT instance in C^7 and C^8 we generate multiple images, each scene corresponds to the set of points as the model and the camera looking at the center of the model, this is illustrated in Fig. 4.10. For each image, we use two different rotation angles. We are considering to kinds of rotation, in the first hand we consider a rotation angle around the z axis, which we parametrize by θ_1, and in the other hand, we use a tilt angle, around the y axis, which we denote by θ_2.

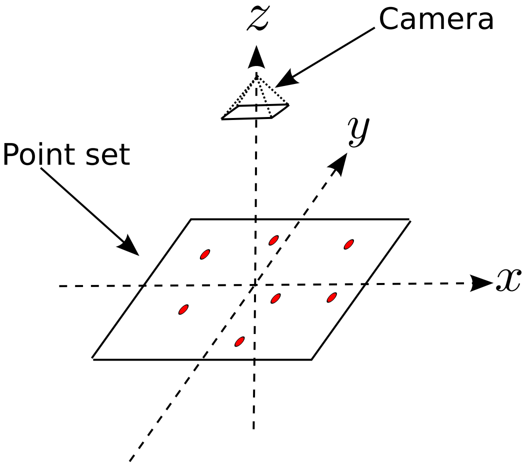


Figure 4.10: Camera plane and plane position on the 3D coordinate system.

The experiment consists of the following; we compute the $\lambda-$matrix for each OT instance C_l^k from the model without any transformation. We compute this $\lambda-$matrix using the Alg. 2 in the page 60, and we consider it as the ground truth $\hat{\lambda}$. Then, each time we generate an image of C_l^k we extract the features, the $\lambda-$matrix from the transformed points on the image $\tilde{\lambda}$, and we verify the correct $\lambda-$matrix retrieval. We can verify when the both computed $\lambda-$matrices, the one from the model and the one from the image are equal, this is, $\hat{\lambda} = \tilde{\lambda}$, which means that we can correctly retrieve the OT with the given OT instance and scene conditions. In other words it means that we can succesfully identify the OT class for the OT instance, and also means that we obtained correctly the invariant labeling to solve the point matching direclty.

The specific parameters we used to generate the scenes and to generate the images are the following. For the image generation we used the pinhole camera model in Eq. (2.1) in Pag. 10. For the pixels generation we truncated the transformed points $\boldsymbol{p}$ to the closest integer value, i.e., $\boldsymbol{p} = [\lfloor u + 0.5 \rfloor, \lfloor v + 0.5 \rfloor, 1]^T$, where $\lfloor \cdot \rfloor$ is the floor operator.

For the scene we use a camera with the following characteristics:

- Image resolution: 640 × 480 pixels.
- Focal distance: $f_x = f_y = 1000$.
- Principal point: Center of image at $(u_0 = 320, v_0 = 240)$.

- Obliquity: $o = 0$.
- Distance to the tag 1 000 a.u.
- We assumed each point set on the plane $z = 0$.
- Each point set was translated to place its centroid to be in the origin of the coordinate system.

From this experiment we generated a total of 3 240 images for each OT instance C_l^k by rotating the plane around z axis from $\theta_1 = 0°$ to $\theta_1 = 359°$ in steps of 10°, and changing the tilt by rotating the model arount the y axis from $\theta_2 = 0°$ (without tilt) to $\theta_2 = 89°$ (high tilt) in steps of 1°. We fixed the third rotation angle to $\theta_3 = 0°$.

The results of this experiment are shown in the Tab. 4.2; in the table, we show the number of analyzed OT instances for C^7 and C^8, the number of analyzed images, the number of successful cases, the number of failed instances, and for summary the percentage of success. Here, a failed case indicates that the ground truth $\lambda-$matrix and the one computed from the points of the analyzed image are different.. This matrix difference indicates that the OT could not be retrieved, thus we could not identify the OT instance, nor solving the feature matching. The percentage of successful cases shows that in most of the rotation conditions we can extract the features correctly and we can correctly identify the OT. We observe that the percentage of success is higher for the images generated with C^7 as the model. We associate this behavior with the maximal perturbation values, which should be higher for the instances in C^7 than for those in C^8 which makes them more robust to noise. However, even for the images generated with C^8 we obtained a 93.98% of success, which represents most of the cases.

Table 4.2: Results for simulations of OT recovering from images with rotation.

	7 points	8 points
Number of analyzed point sets	135	3 315
Images analyzed	437 400	10 740 600
Successful recovered $\lambda-$matrices	427 437	10 093 612
Failed recovered $\lambda-$matrices	9 963	646 988
Percent of success	97.72%	93.98%

With the aim to better understand the cases in which the features extraction fails; in Fig. 4.11 we show the histogram of failed cases. In the figure, we plot the tilt angle, θ_2, and the number of failures for C^7 and C^8. In the graph, we see that the errors occur in high tilt conditions, i.e., when $\theta_2 > 65°$. In the figure, we again appreciate that the OT instances in C^7 support more tilt, in such a way that the failures occur for $\theta_2 > 80°$.

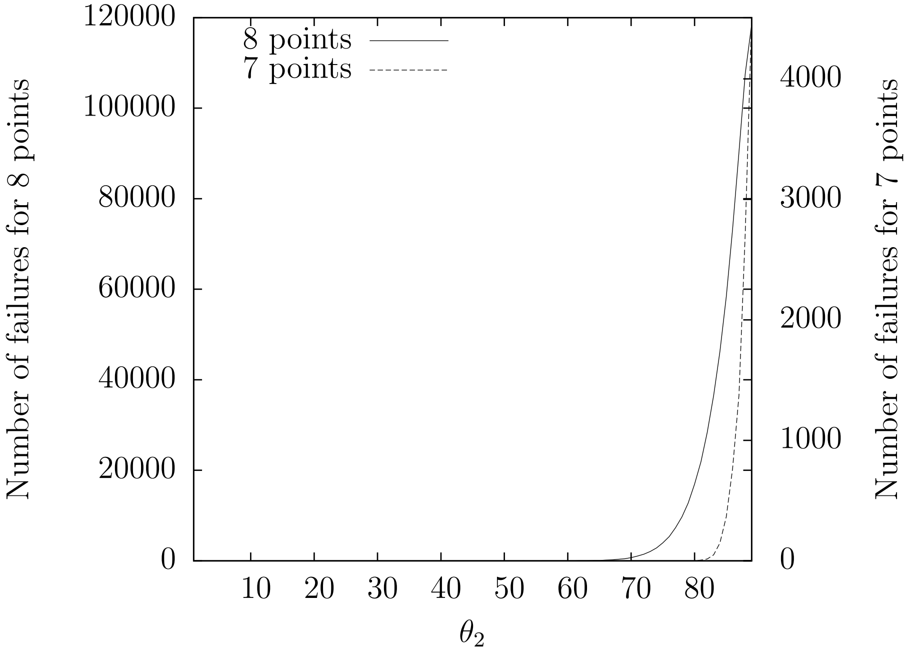


Figure 4.11: Histograms of failures for rotation.

$\lambda-$matrix extraction in different distance conditions

In the previous experiment, we changed the rotation and tilt conditions of the generated images, in a fixed distance. Then to complement the experiment, we analyzed the effects of the distance in the features extraction. As similar as in the rotation and tilt experiment, we generated for each OT instance in C^7 and C^8 multiple images in different conditions, now we fix a rotation configuration $\theta_2 = 0$, and we maintained the intrinsic camera parameters, but we vary the distance from $d = 500$ to $d = 10\,000$ in steps of 10 unities. The aim here is again to verify if the $\lambda-$matrix computed from the points on the image is the same to the ground truth. We show the results of this experiment in table 10. In the table, we observe that the OT instances in C^7 allow again to obtain the better results, with 97.85% of successful cases. In the images, a greater distance corresponds to a decrease of the area occupied by the set of points on the image. In a smaller area, the distance of between the points that compose the OT instance is smaller; if the distances are smaller, then the maximal perturbation value must also be reduced, what serves as the explanation for this behavior. In the Fig. 4.12 we show the histogram of failures for this distance experiment. We observe that errors increase as long as the distance d increases, in such a way that for the OT instances in C^7, 100% of failures occur at $d > 5580$; and for those in C^8, the 99.95% of failures happened at $d > 2440$.

Table 4.3: Results for simulations of OT recovering varying distance.

	7 points	8 points
Number of analyzed point sets	135	3 315
Images analyzed	128 250	3 149 250
Successful recovered IDs	125 498	2 232 380
Failed recovered IDs	2 752	916 870
Percent of success	97.85%	70.89%

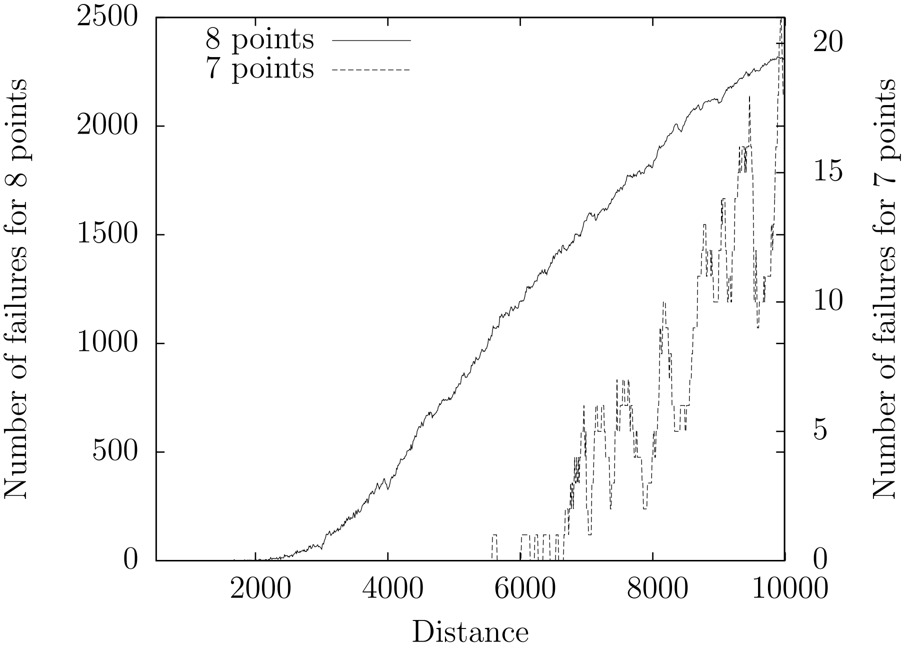


Figure 4.12: Histograms of failures for distance. Top: results for seven points. Bottom: results for eight points.

4.8.2 Robustness of $\lambda-$matrix to additive noise in point positions

In this section, we evaluate in a quantitative but general approach, how sensible are the OT instances to noise. Every set of points with no three or more colinear points belongs to a particular class of OT. The OT depends on the orientation of the triplets conforming every set of points, what implicitly makes OT robust to noise, i.e., the OT will not change with small perturbations in the positions of the set of points confirming the set. This section aims to evaluate the sensibility of the OT instances. To perform this evaluation, we compute the MP value for each OT instance provided in [1] for those set of points with cardinality greater or equal to five but lesser than eight. We perform the computation using the original positions for every set of points without applying any transformation. This approach allows us to understand how each of the existing OTs behaves in the presence of noise.

We computed the MP for each set of points in C^k with cardinality $5 < k < 7$. For summarizing purposes, we define different levels of noise value v, from 0.5 to 9 unities in steps of 0.5. Based in this values, we define the sets $D^k = \{C^k_l | \mathrm{MP}(C^k_l) \leq v\}$, which correspond to the OT instances for which the associated $MP(C^k_l)$ is lesser than a given v value.

In Tab. 4.4 we show the values for v and the number of OT instances that support the v level of noise, i.e., $|D^k|$. As can be seen in Tab. 4.4, for $v = 6.0$ and eight points there is possible to get only 15 distinct OTs, i.e., if it is allowed noise in each point position of at most 6a.u., then it is possible to get only 15 OTs. For five points it is possible to get three different OTs even with higher values of $v = 9.0$. From the results in the table we observe that as long as we increase the noise, for higher values of v, the number of Ots that supported is decreased.

Table 4.4: Number of OTs for different noise allowed in point positions. Bold numbers indicate the maximum number of existing OTs for $k = \{8, 7, 6, 5\}$.

v	$\lvert D^8\rvert$	$\lvert D^7\rvert$	$\lvert D^6\rvert$	$\lvert D^5\rvert$
0.5	**3315**	**135**	**16**	**3**
1.0	3296	**135**	**16**	**3**
1.5	1240	**135**	**16**	**3**
2.0	642	**135**	**16**	**3**
2.5	371	**135**	**16**	**3**
3.0	231	86	**16**	**3**
3.5	135	60	**16**	**3**
4.0	83	47	**16**	**3**
4.5	56	32	**16**	**3**
5.0	37	26	**16**	**3**
5.5	26	18	**16**	**3**
6.0	15	15	**16**	**3**
6.5	10	8	**16**	**3**
7.0	5	7	12	**3**
7.5	4	4	10	**3**
8.0	3	3	8	**3**
9.0	3	3	6	**3**

The results in the Tab. 4.4 confirms that every OT instance has a different Maximal Perturbation value. The proposed MP concept allows us to rank the OT instance according to the OT instances, in such a way that we can give preference to use those OT instances with higher MP, which are more robust to noise.

4.8.3 Order Types suitable for point matching

In this experiment, we analyze the OT database in [71] for C^5, C^6, C^7, and C^8 for checking which instances are suitable for point matching. For each C_i^k instance, we perform the Alg. 2 for each OT instance C_l^k and we count the canonical orderings with a $\lambda-$matrix equal to the minimal lexicographical. We denote as E_i^k the instances with a unique minimal $\lambda-$matrix that are suitable for point matching through OT. The uniqueness of the minimal $\lambda-$matrix ensures that there also exists a single invariant labeling that allows us to solve the point matching without ambiguities. In Tab. 4.5 we show the count for each E^k set with $k = 5, \ldots, 8$, to contrast the results we also show the number of all the existing OT instances and the percentage of the OT instances that are suitable for the direct feature matching.

Table 4.5: The number of OTs suitable for point matching.

k	$\lvert E^K \rvert$	$\lvert C^k \rvert$	Percentage of suitable instances
5	2	3	66.66%
6	11	15	73.33%
7	131	135	97.77%
8	3303	3315	99.63%

From the results in Tab. 4.5 we observe that most of the OT instances allow solving the feature matching directly using the invariant labeling. The percentage of the OT instances with a unique minimal $\lambda-$matrix increases with the cardinality of the sets. In the case of $k = 5$ we see that only two of the three OT instances comply with the $\lambda-$matrix uniqueness condition. The set that does not is that in which all the points are on the convex hull. For this case, from the computation of the canonical ordering, we obtain five labelings, but the associated lambda matrix for each of them is the same, they all are equal, then we can not have a unique labeling that identifies each point, and we can not solve the feature matching without ambiguities. The same case occurs for the other values of k with all the points on the convex hull, but there are other cases like these are shown in the Fig. 4.13.

From the Fig 4.13 we observe that those OT instances that present symmetry in some of the composing triplets.

4.8.4 Automatic $\lambda-$matrix extraction from visual fiducial tags

In this section, we test the $\lambda-$matrix extraction in more realistic conditions. We aim to test the correct features extraction, the $\lambda-$matrix computation and the correct matching using ray-traced images but also real images taken with a physical camera sensor. For this subsection, we perform three experiments, for all of them we use the visual fiducial tags that we proposed, the OTTs. The each OTT encode a set of points as the vertices of a set of triangles. The tag and the vertices are auto identifiable elements, in such a way that we can estimate the position for the set of points at

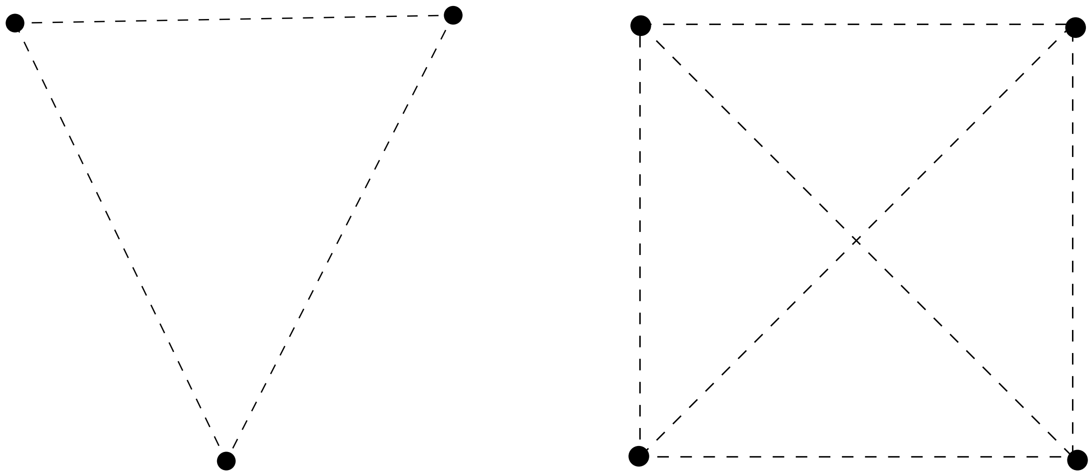


Figure 4.13: Instances of set of points that have not a unique minmal $\lambda-$matrix and that are not suitable for solving the feature matching direclty.

sub-pixel precision. More details about the proposed OTTs were presented in the Sec. 4.4, in page 61.

In the Sec. 4.8.1 we confirmed that the OT can be retrieved even when an OT instance is transformed by the image generation process and all the involved transformations. However, along with the transformations and inherent noise in the image acquisition, there could be other aspects that can affect the extraction of features, and in the context of this thesis, also the features matching process. In the features extraction from the OTTs we perform additional operations to obtain the features, i.e., the morphological operators, and vertices estimation (consult the Sec. 4.4, in page 61). These both additional operations represent different sources of noise that are inherent in the use of OTTs; thus it is crucial to evaluate the features performance in the proposed visual fiducial tags. As the first experiment, we built scenes of an OTT instance in front of the camera; ; we generated 179 images with the OTT visible in different tilt angles at a fixed distance. In the second experiment, we fixed the tag in front of the camera without tilt, and we generated 171 images by increasing the distance from the tag to the camera. The details of these both experiments are detailed in the following subsections.

4.8.5 Order Type Tags rotation and tilt test

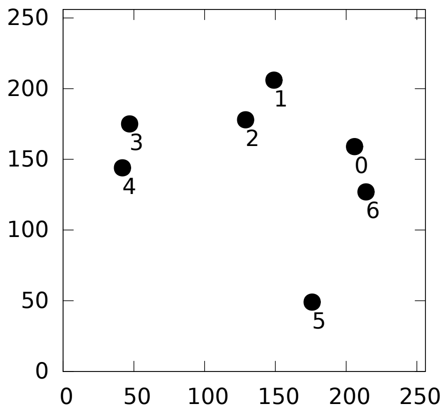
For testing our proposal of the OTT design with we implemented the OTT decoder as described in the Sec. 4.4. We arbitrarily used the tag shown in Fig. 4.5 to generate artificial scenes with the help of the ray-tracer program POV-Ray [77]. For scene generation we considered the specifications in Table 4.6. In the generated scenes we fix the intrinsic parameters of the camera, and we vary a tilt angle from -89 to 89 with changes of one degree. In the experiment, we priorly know that the high tilt conditions all the set of points encoder by the tag will tend to become all colinear

Table 4.6: Experiment conditions for OTT rotation experiment.

Algorithms parameters	
Area threshold	3% of the image width
Scene characteristics	
Tag size	8.26772 × 8.26772 Inch (21 × 21 cm.)
tag-camera distance	50 cm.
Tilt angles	From $-89°$ to $89°$ in $1°$ steps
Camera characteristics	
Focal distance f	700 pixels
Obliquity o	0
Image size	640×480 pixels

and the λ matrix will not be able to be computed. In this experiment, we want to know the maximal and minimal tilt values in which the tag can be detected correctly. In total we generated 179 ray-traced images, some instances of them are shown in figure 4.15. For the design of the OTT we used an arbitrary OT instance. The base OT instance is shown in the Tab. 4.14, in there we also present the associated $\lambda-$matrix. For designing the OTT from the OT instance, we performed the OTTs design methodology proposed in the Appendix A.

Point set distribution



Associated λ-matrix

$$\begin{bmatrix} - & 5 & 4 & 3 & 2 & 1 & 0 \\ 0 & - & 2 & 5 & 4 & 3 & 1 \\ 1 & 3 & - & 4 & 3 & 2 & 2 \\ 2 & 0 & 1 & - & 5 & 4 & 3 \\ 3 & 1 & 2 & 0 & - & 5 & 4 \\ 4 & 2 & 3 & 1 & 0 & - & 5 \\ 5 & 4 & 3 & 2 & 1 & 0 & - \end{bmatrix}$$

Figure 4.14: Set of points used to generate the test for OTT and its associated $\lambda-$matrix. The set of point is $\{(206, 159), (149, 206), (129, 178), (47, 175), (42, 144), (176, 49), (214, 127)\}$.

From the 179 images, 165 images were detected successfully and decoded. For the complementary 14 images, our implementation did not detect the marker on the image thus the ID could not be decoded. These 14 images were those image with the highest rotation, seven images from $\theta_2 = [89°, 83°]$ and seven images for $\theta_2 = [-82°, -89°]$.

The percent of the success of detected correct IDs for this experiment was 92.17%. These results suggest that the proposed automatic detection preserves the features in high tilt conditions.

4.8.6 Order Type Tags distance test

For this experiment, we used the same OTT along the same camera intrinsic characteristics used in the previous experiment in Sec. 4.8.6. We fixed $\theta_2 = 0°$ and we varied the distance from OTT to the camera from 30cm. to 200cm. in steps of 1cm. For this experiment a total of 171 images were generated, some of the generated images are shown in figure 4.16.

From the 171 processed images, 91 of them were successfully recovered. We detected the OTT in all 171 images but point set estimation was not correct in all cases thus ID was also affected. We observed that 100% of failures happened with $d \geq 62$cm.

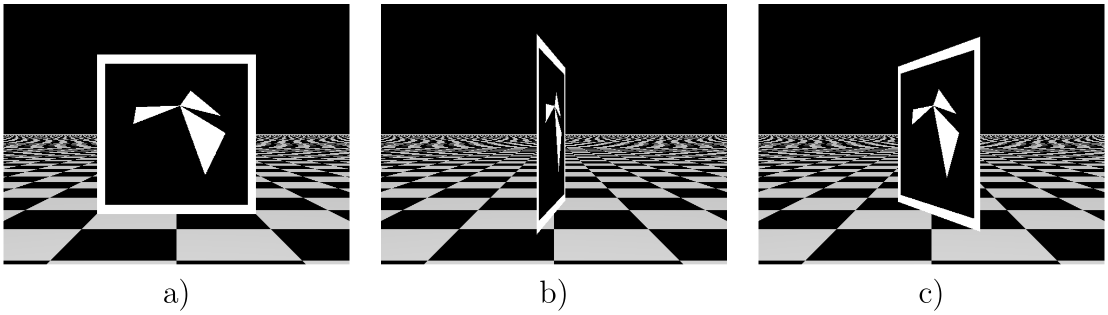


Figure 4.15: Instances of the ray-traced images for the rotation experiment. a) Tag with 0° grades of rotation. b) Tag with 80° grades of rotation. c) Tag with $-60°$ grades of rotation..

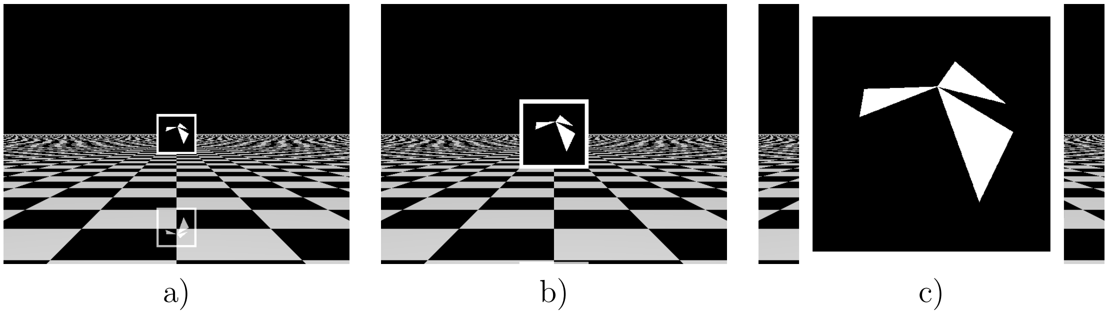


Figure 4.16: Instances of the ray-traced images for the distance experiment. a) Tag to 2000 cm of distance. b) Tag to 115 cm of distance. c) Tag to 30 cm of distance.

4.8.7 Augmented reality application

With the aim to show the applicability of the study in this chapter, we implemented an augmented reality application. Using the proposed OTT, we apply the automatic extraction of the points, the $\lambda-$matrix computation and the point matching. To generate the augmented reality, we use the model used to create the marker and the extracted features to perform the camera calibration and the camera pose estimation. The extracted point, the known model, and the point matching allows us to apply the Zhang's camera calibration method, which also allows us to estimate the camera pose from a homography decomposition method. In our implementation, we are using the most straightforward approach that enables us to obtain fast image processing, and we are not performing any parameters refinement through BA. In Fig. 4.17 we show the result of this experiment, we show six images of the tag obtained with a camera. Each image shows virtual object over it: the detected vertices as green dots, tag axes: x (red), y (green), and three virtual objects (three cubes rotating in their vertical axis), one placed at the center of the tag and two other at two opposite corners.

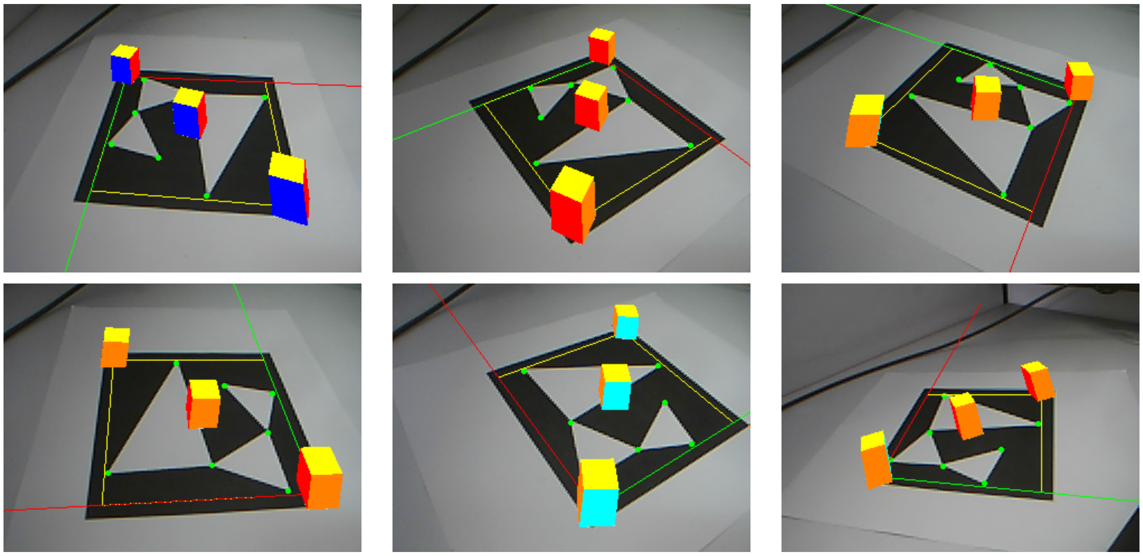


Figure 4.17: OTT used for Augmented Reality Application. Six images rotating the marker. The pose is fully obtained (check out how axis lines rotates with the marker).

4.9 Discussion of results

In this chapter, we proposed to solve two of the 3D reconstruction sub-problems simultaneously, the features extraction and the features matching. For this, we offer features that are defined as a set of points but also the structure among the points of the set. For concept behind this structure, we used the OT. The OT studies the set of points concerning the orientation of its triplets of points. Depending on distribution and organization of the points, the OT can be different. The number and the characteristics of the OT are well known, in such a way that we can say that each set of points with no collinear points has an associated OT. The number of different OTs depends on the cardinality of the set of points, and this means that each set of points belonging to a particular class of OT. For our purposes we want the OT to be invariant to the image generation process. We studied this aspect, and we found that if we identify the OT from a model, a set of points, without any transformation applied, and we can obtain or determine the same OT for the same model projected on an image with perspective, noise, and pixels generation. For the scope of this thesis, these results validate our proposal for the features. The computation of the OT involves the calculation for an invariant labeling. This labeling is associated with a minimal $\lambda-$matrix that describes a set of point. From the viewpoint of the OT, this invariant labeling allows solving the features matching, in such a way that if we have two set of points with that belong to the same OT class, then the labeling for each point en both sets will indicate which point of each set match. From our studies and experiments we find and validate that this also applies for the case of projected images. Then, in the context of the 3D reconstruction, we say that the features based on the OT allow us to identify planes, where these planes can be images taken from different viewpoints of a planar scene but also the same relationship will hold between the points on an image and the model. These results change the 3D reconstruction pipeline, because now we, instead of computing the features to later

In the traditional approach, a set of features is extracted from the images in the input sequence. The features are extracted independently, later pairs of images are analyzed to try to find matches between the elements. The problem is solved for pairs of images, and the results from this partial matching need to be consolidated when more than two images are available, which complicates the problem. With our proposes features the 3D reconstruction pipeline changes. Given the input sequence of images, we extract the features from each image. From this step, we obtain the points, the associated $\lambda-$matrix that describes and identifies the set of points and the associated invariant labeling on the image. Then for the matching we first have to determine the images that have the set of points with the same $\lambda-$matrix, this can be understood as the task to identify the same set of points observed from different viewpoints. Then for those corresponded identified images, the invariant labeling directly indicates the match among the points in different images. We have to mention that this analysis implicitly consolidates the matching process for all the images in the input sequence. Since the proposed features can be estimated for sets of points on the image but also in the model, we apply the same concept for solving the pose estimation, which is the estimation of the relative motion among the camera and the model for a specific image, which apart for the feature and features matching, also related the pose estimation sub-problem with the same concept.

To validate our proposal we proposed the OTTs, which are visual fiducial tags that encode a set of points as the vertices of a set of triangles. The OTTs use the $\lambda-$matrix as an identification number or ID. With these tags, we could implement automatic identification and augmented reality applications that served us to evaluate the proposed features for realistic conditions.

We found a drawback for the proposed features; we showed that we could identify two sets of points if they have the same $\lambda-$matrix. This verification cannot be performed if the cardinalities of the two sets are not equal. This represents a new problem, that is interesting to study, but for now is out of the scope of this thesis.

We also studied the effects of the noise for the proposed features, we introduced a method to quantify the sensibility of a set of points to noise through the Maximal Perturbation concept. The Maximal perturbation depends on the coordinates of each element in a set of points. This fact means that it will not be invariant to the image generation process. However, if we count with the OT instances as models of a scene, the concept permits to rank the OT instances according to their robustness to noise. This last study shows the path to investigations that can be handled as future work.

4.10 Study remarks

In this chapter, we presented a new kind of features for the 3D reconstruction problem. The proposal solves the feature matching, and the features matching simultaneously, but also it relates the pose estimation sub-problem simultaneously. We showed that the features could be applied to images with different viewpoints but also between images and the model. We also demonstrated the application using visual fiducial which shows the applicability of the proposal for automatic identification and augmented reality applications.

In the next chapter, we present a joined approach using the approach shown in Chapter 4 and the given in this chapter. The idea in that chapter is to integrate both strategies in the complete 3D reconstruction pipeline, in such a way that the reconstruction of a scene is performed with the simultaneous solution to many of the involved sub-problems.

Chapter 5

A Complete Scheme using both Approaches

In this chapter, we research about a joined approach using the two proposals presented in the chapters 4 and 5, i.e., the simultaneous feature extraction and matching, as well as the camera self-calibration, pose estimation and model reconstruction. As early mentioned in the Chapter 2, the 3D reconstruction, in the traditional approach, involves the solution of a sequence of multiple sub-problems: the features extraction, the features matching,the camera calibration, the pose estimation, and finally our main aim, the retrieval of the 3D model. In the previous chapters, we proposed and studied two approaches that solved some of the 3D reconstruction pipeline sub-problems simultaneously. In the development of this thesis, we first examined, in Chapter 3 (in page 27), the simultaneous solution to three of the five sub-problems simultaneously, the self-calibration, the pose estimation, and the 3D retrieval. A prerequisite for the approach was to count with the features identified in at least three images of the images input sequence and the match between the features, then in order to apply the three sub-problems approach, we needed to solve first, the first two sub-problems. To handle that, we later studied in Chapter 4 (in page 55), the simultaneous solution to the two remaining sub-problems, the feature extraction, and matching.

In this Chapter, we study the complete solution to the 3D reconstruction problem by applying the two proposed approaches consecutively. First, the feature extraction and matching, later using the the self-calibration, the pose estimation, and the 3D model retrieval. This new joined approach represents a fully automated 3D reconstruction approach that applies all the research obtained in the previous chapters, but that also represents a new approach or flow to solve the 3D reconstruction.

Here we aim to show the direct application of the use of OT in the 3D reconstruction pipeline. Using an OTT instance, as proposed in the Sec. 4.4 in page 61, and three images of it from distinct viewpoints, the aim is first to solve the features extraction and matching for the three images. Then we apply our self-calibration, pose estimation and 3D model retrieval. At this step, we can obtain the model reconstruction.

5.1 Experimental validation

The flow of the complete squeme is shown in Fig. 5.1. The three input images will be simulated pictures of a OTT. In Fig. 5.2 it is ilustrated the squeme: The points in the three images are matched using OT, and for this step is only necessary to perform two matches, between images 1 and 2, and between images 3 and 1 (or between images 3 and 2). Once the point in three input images are matched, then it is possible to apply the method to obtain the simultaneous self-calibration, the three poses estimation and the reconstruction (as it is shown at the right of Fig. 5.2).

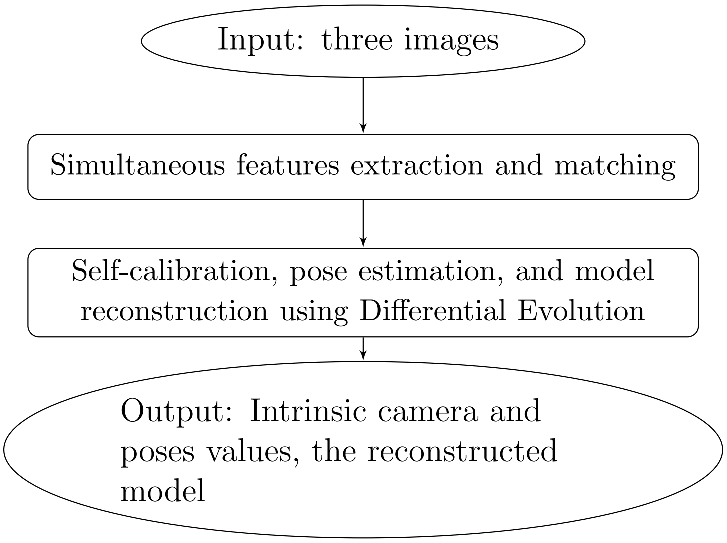

Input: three images
Simultaneous features extraction and matching
Self-calibration, pose estimation, and model reconstruction using Differential Evolution
Output: Intrinsic camera and poses values, the reconstructed model

Figure 5.1: 3D reconstruction flow using the two proposed approaches in this thesis.

The points for the used OTT are ilustrated in Fig. 5.3. The associated minimal lexicographical λ-matrix for these points is given algo in the Fig. 5.3 at the left. The point values are shown in Tab. 5.1. The designed OTT is shown in Fig. 5.4. Note that the used points for this OTT correspond to an instance of C^7, but they are arranged to cover the area of a square fiducial marker, then the points have not the same values that the corresponding instance with the same λ-matrix in the database in [1].

Table 5.1: Characteristics for the chosen OT instance.

Number of points	7
Points locations	$\{[0,0],[254,0],[127,40],[215,127],[254,254],[40,127],[0,254]\}$
Maximal perturbation	15.075619
Triangle vertices	$\{[127,40],[215,127],[40,127]\}$

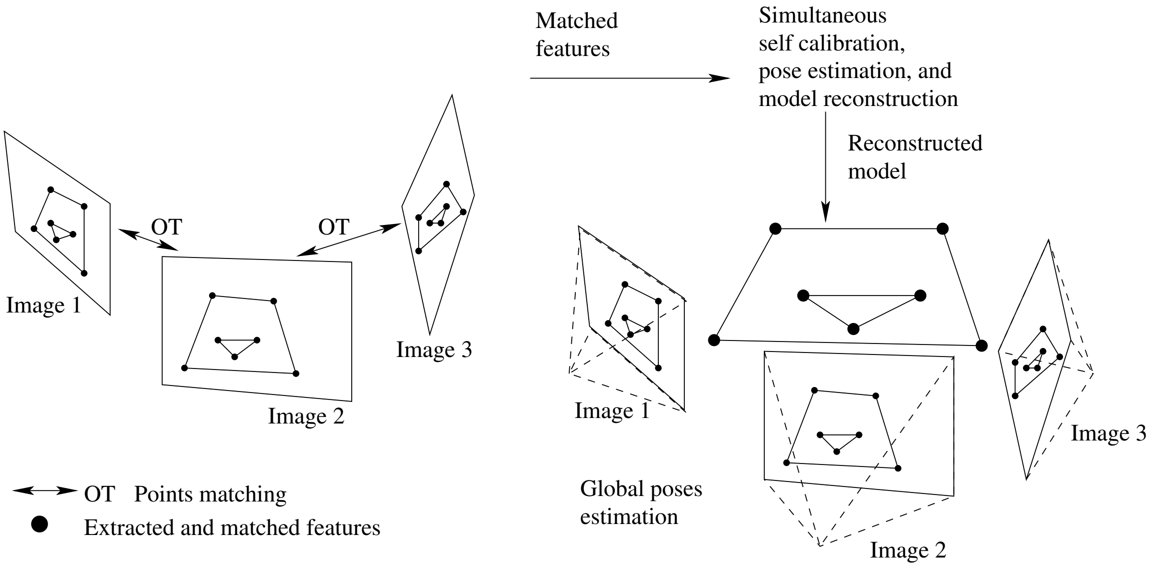
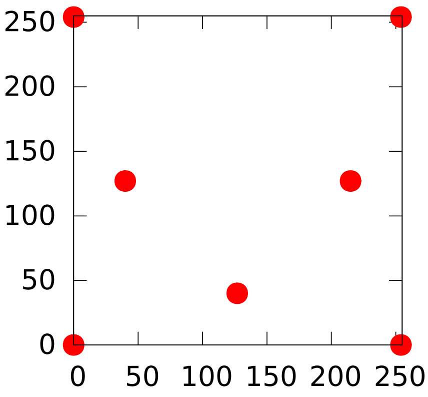


Figure 5.2: The idea for automatic features extraction and their matching followed by simultaneous self-calibration, pose estimation and model reconstruction.



$$\lambda = \begin{bmatrix} - & 5 & 4 & 3 & 2 & 1 & 0 \\ 0 & - & 2 & 5 & 4 & 3 & 1 \\ 1 & 3 & - & 4 & 3 & 2 & 2 \\ 2 & 0 & 1 & - & 5 & 4 & 3 \\ 3 & 1 & 2 & 0 & - & 5 & 4 \\ 4 & 2 & 3 & 1 & 0 & - & 5 \\ 5 & 4 & 3 & 2 & 1 & 0 & - \end{bmatrix}$$

Figure 5.3: Left) Base OT instance for the experiment, seven points. Right) The associated $\lambda-$matrix.

We aim to perform the joined approach with many and different poses for the images. For the experiment, one hundred triplets of images were generated. Each image is generated from the OTT showed in Fig. 5.4 taken at different rotation angles. The used camera parameters were: size of the image 640 × 480 pixels, constant focal distance of 1000, principal point at the center of the image, and axis obliquity equal to zero. The following scenario was used: OTT is located on the plane xy and is translated $[-127, -127, 0]^{\mathrm{T}}$ (the center of the coordinate systems is in the middle of

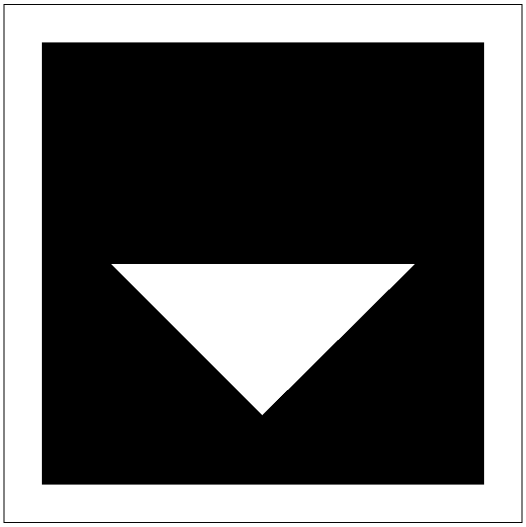


Figure 5.4: The generated OTT using the base OT instance.

the OTT). The camera is localized at point $\mathbf{c}$ which is calculated as:

$$\mathbf{c} = \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} = \begin{bmatrix} r\cos(\phi)\cos(\theta) \\ r\cos(\phi)\sin(\theta) \\ r\sin(\phi) \end{bmatrix}, \tag{5.1}$$

with $r = 800$. Eq. (5.1) calculates points over a sphere of radius equal to 800, as it is shown in Fig. 5.5. The camera is viewing to the center of the coordinate system and the direction of the up vector is $[0, 0, 1]^{\mathrm{T}}$. With these three vectors, camera position, viewing point, and up vector, it is possible to calculate a matrix R, where R^{T} is the orientation of the camera with respect to the world coordinate system. The values for the two angles ϕ and θ are randomly chosen within the ranges shown in Tab. 5.2. We set a unique condition, the angular difference between the three θ angles must be greater than 10°. The images were generated using a Povray script [77]. Some instances of the generated triples of images are shown in Fig. 5.6. These images were processed with a program written in C which calculates the black square vertices (see the image of the maker in Fig.5.4) and the white triangle vertices with a similar process described in Sec. 4.4, in page 61: the perimeter of the square and the triangle are obtained, then a first estimation of the vertices is obtained, then the pixels of each edge are extracted, and for each point set a line is fitted using PCA algorithm; the refined vertices positions are the intersection points of those lines. In this way, the vertices positions are obtained at subpixel precision.

In Tab. 5.2 are also show the parameters values used by DE, which have the same values used in Chapter 3.

Table 5.2: Experiment scenes conditions.

	Values or ranges
ϕ	20° to 70°
θ	0° to 359°
f	1 000
u_0	320
v_0	240
s	0.0
Camera-Tag distance	800
Population size	50
Crossover probability	0.9
Differential constant	0.7
Stop criterion	0.001
Maximal cost function evaluations	1 000 000

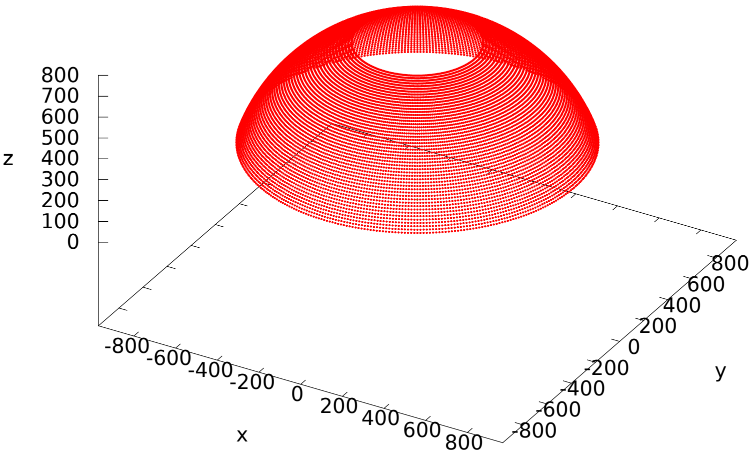


Figure 5.5: Possible positions for the camera. Every point on the surface is represents a possible location of the camera looking to the origin.

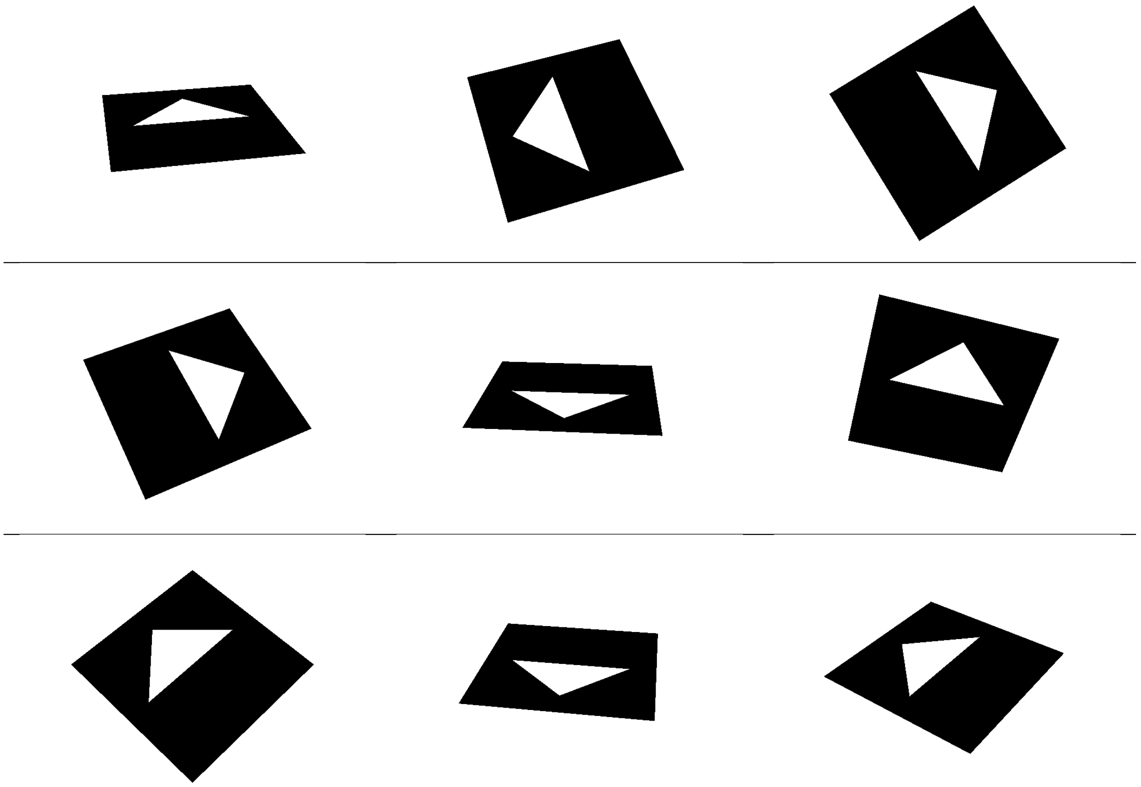


Figure 5.6: Three instances of the generated sets of three images for the experiment. One set per row.

5.2 Results

The process showed in Fig. 5.1 was executed 100 times. In the Tab. 5.3 are shown the mean, the standard deviation, the median, the min, and the max value obtained, for three measurements for the one hundred executions: the focal distance relative error, the root mean square (RMS) of the error computed using the ground truth model and the reconstructed, the RMS of the reprojection error, and the number of fitness function evaluations. Remember that the fitness is the reprojection error.

Table 5.3: Errors obtained from the experiment considering the 100 executions.

	f relative error (%)	Model RMS	Number of function evaluations
mean	3.4501	2.5758	139 070
std.	8.3593	5.9388	29 777
median	0.7227	1.1243	131 150
min	0.0080	0.7806	102 850
max	43.489	49.982	287 700

For all the one hundred cases and the three hundred images, the matching was estimated correctly. This excellent behavior is because two reasons: 1) the high

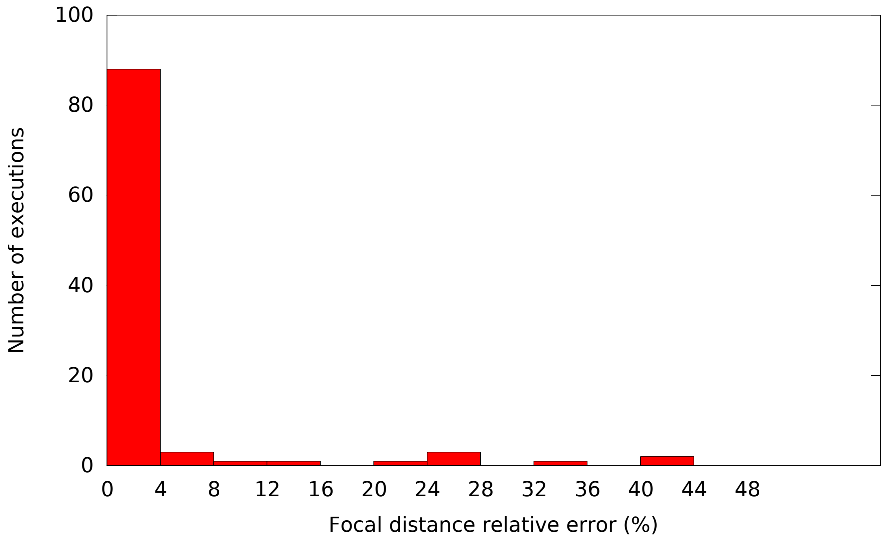


Figure 5.7: Relative error for the focal distance, 88 executions present an error below the 4%.

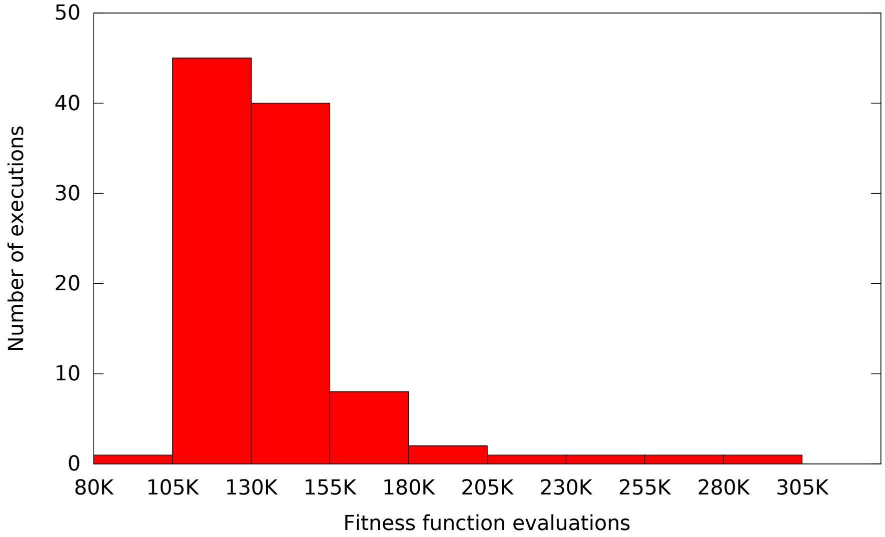


Figure 5.8: Function evaluations distribution. 90 executions require less than $180,000$ cost function evaluations.

maximal perturbation value of 15.08 shown in Table 5.1 that absorbs all posible noise in the point positions, and 2) the processing of the images uses ideal images generated by simulation. Concerning the precision of the reconstruction, we observe, in the Tab. 5.3, that the relative error of the obtained solution has a median of 1.12 unities. The median of the relative error for the focal distance shows a value of 0.72% of error which is an indicator of good reconstructions. In the table, we also observe that the mean value of the relative error for the focal distance is little higher. This value is

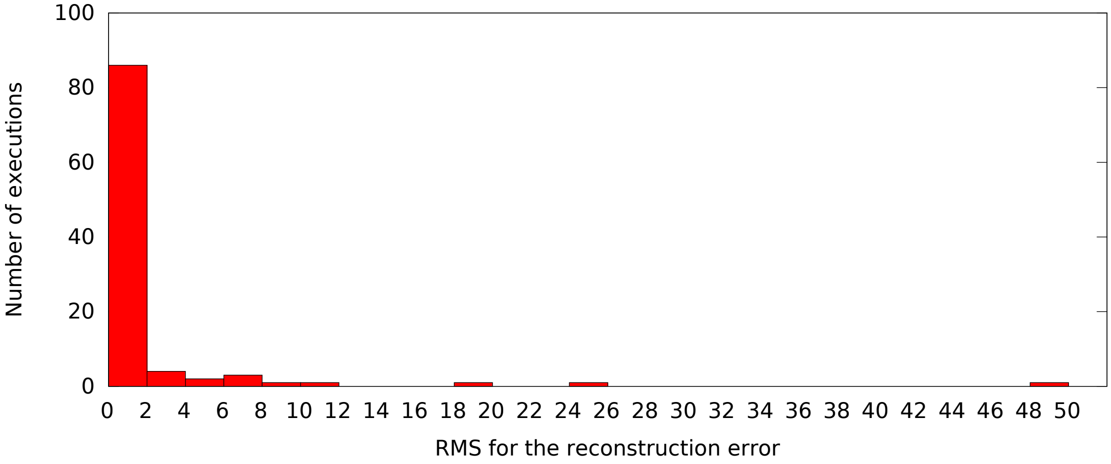


Figure 5.9: Reconstruction RMS distribution. 86 executions obtained an RMS error below 2 model unities.

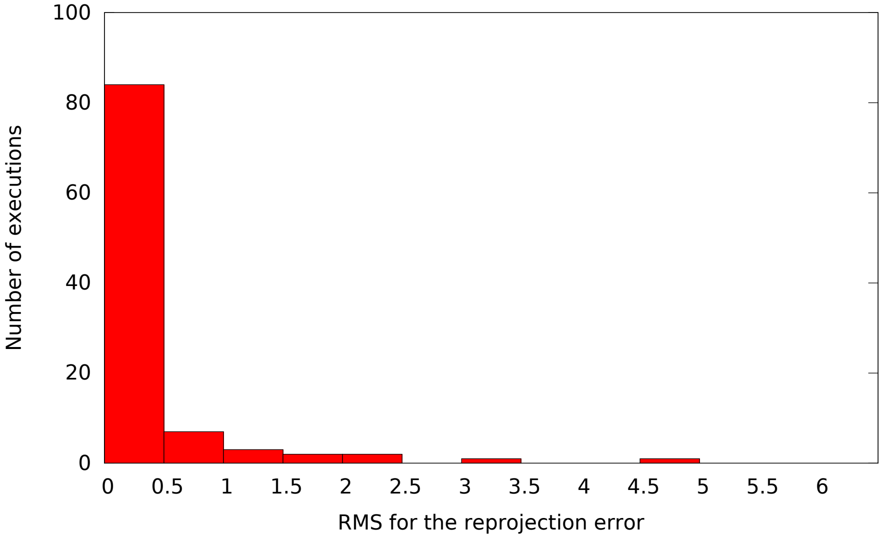


Figure 5.10: Reprojection error distribution. 84 executions obtained an RMS error below 0.5 pixels.

the result of the error obtained in the atypical cases, as denoted by the max value. With the aim to provide a better perspective of the obtained results we analyzed the focal distance through a histogram, we show the result of this analysis in Fig. 5.7. In the figure, we show in the horizontal axis different values for relative error for the focal distance, and in the vertical axis, we show the number of the experiment instances that obtained an error within the specified range. In the figure, we observe that 88 of the 100 executions obtained a relative error of the focal distance in the range $[0\%, 4\%]$.

As complementary graphs in Fig. 5.8, also given as a histogram, we observe that 45 executions solved the problem with $[105\,000, 130\,000]$ fitness function evaluations, and 40 of the 100 executions instances are in the range $[130, 000, 150\,000]$. Concerning the precision of the reconstruction, in Fig. 5.9 we observe that 86 of the 100 of the executions were solved with an RMS of reconstruction error below two unities, and finally in the Fig. 5.10 we observe that 84 of the 100 executions instances ended with an RMS of the reprojection error below 0.5 pixels, this last graph, in contrast to the shown in 5.9, shows how sensible is the final result to little differences of the values for the cost function: although the reprojection error is small, the associated reconstruction error could be greater.

5.3 Discussion

In this section, we showed the joined approach of the features extraction and the simultaneous self-calibration, pose estimation, and model retrieval. With the features extraction and simultaneous matching approach, we solve the matching among images. This is, the points on one image are matched to the points on another image or to multiple images with the same associated OT. We must mention that this is indeed possible because the images are planes, and even more, the model observed in the images is also a plane. In chapter 4 we only considered the OT for planes. A plane model projected to an image result in a plane also, formally, the model plane is projected to image by a non-linear transformation of the projective geometry, the homography. In different conditions, in which we would want to perform the features matching between images generated by a non-planar model the correct matching cannot be ensured. We showed in chapter 4 that the OT maintains unchanged under the transformations of the image generation process, but we showed that only using planar models. Then, in the case with 3D models, there will be some cases that will allow obtaining correct features matching but it is not guaranteed. This new problem is another of the existing problems that are out of the scope of this thesis but that we believe to have a solution in the study of strategies to use the OT in its 2D version for arbitrary models. In Sec. 5.3.2 we give some of the analysis and ideas we have obtained in a direction to solve this interesting problem. In the results presented in the graphs, we observe that we successfully were able to reconstruct scenes in arbitrary conditions applying the two proposed approaches. The experiment shows that we could correctly perform the feature extraction as well as the camera self-calibration, the pose estimation, and the model reconstruction, all that in an automated and in such a way that some sub-problems were solved simultaneously with the two proposed approaches. In the experiments, we observed some instances that had bad results, for instance, the one in which we obtained a relative error of 43% of the focal distance. This case is presented when the condition of the images do not represent enough difference between images (the images do not contribute with enough information) to solve the problem.

5.3.1 Complexity of the feature matching process

In the literature, the feature extraction and feature matching are performed in the following way. First, for each image in the input sequence, a set of points is extracted. This process is independent per each image; this means that there is not any dependency between the points extracted from an image and the ones extracted from another image, thus it is possible to perform the point extraction in parallel, i.e., to process several images at the same time. Second, the feature matching must be performed. The aim is to identify the points in an image that correspond to the points in another. In a more meaningful sense, the aim is to find features of the 3D model but observed from different viewpoints, of course, without knowing the model. For this feature matching, commonly, the process is performed using pairs of images. The problem is reduced to find the features that correspond in pairs of images. For instance, when this matching is performed, the features in an image A and image B with the most similar SIFT descriptors can be candidates to be matched features, but this must also be validated through the epipolar constraints.

Here, we want to highlight some aspects. First, the matching is performed using image pairs, this means that each pair of images needs to be processed; which also means that the number of possible pairs to process has a quadratic complexity.

The second highlight is in the sense of the triangulation of the 3D model points: it is beneficial to perform the triangulation using all the available correspondences in different viewpoints. Thus, the processed images pairs require additional processing, in order to identify those features that are visible in more than two images, three, four, or more. Then to find the same feature in multiple images can result in a very expensive task.

In our approach, we propose the feature matching in such a way that we can perform the feature extraction and the feature matching of images separately. This independence means we could in parallel extract the features for each of the images. Later we can again in parallel find the matches. We just need to take a pair of images, to extract their points, and to verify if their associated λ-matrix coincide. If the λ-matrices coincide, then we have to associate the features in the images pair as matches using the invariant labeling. Then we could take the third image; this third image just needs to be compared against any of the previously processed images, for instance, the first one. The process can continue and can be applied to more than two images, in this approach, the matches for a new image can be directly computed by using the new image and one of the already processed. Thus the complexity of feature matching for our proposed idea is linear with respect to the number of images.

5.3.2 Feature matching for an arbitrary scene without using OTTs

We already mentioned the future work about solving the features matching when the original model is not a plane. In this section, we aim to save the revision about the problem and also the seed ideas for solving it. First, we must mention that the problem consists in identifying, through the OT, and if the features in one image match the features in another image. First, we must mention that the OT concept applied in this thesis is valid for sets of points on the plane; this is because we know that the orientation of the triplets maintains under the image generation process. We can say that if all the points observed in an image A are all observed in other image B with a different viewpoint, then there is a high possibility that we could detect that they correspond to the same OT. In the case of 3D models, we could say that the model points have different depth w.r.t to the image plane. This difference of depth makes that points are affected differently by the effects of 3D rotations and also translations. As a result of the different transformations, it can happen that some points that are visible from one image easily become not visible from another image, because of the effects of occlusion. If that happens, the set of points can mismatch in their cardinalities, making the OT matching to have so many solutions, this case is illustrated in Fig. 5.11. In the following, we will refer to this problem as the occlusion issue.

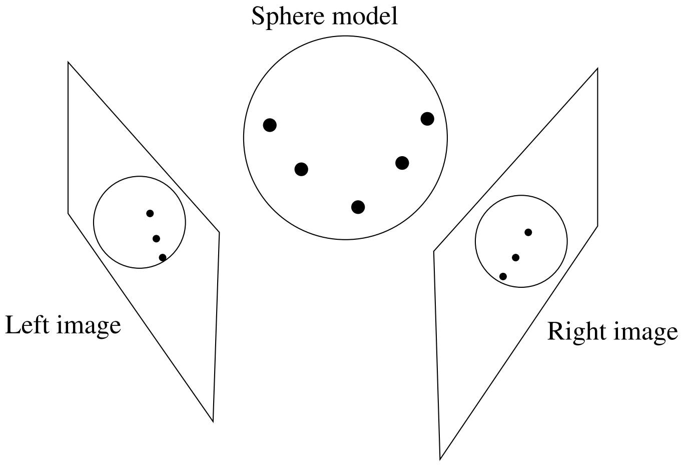


Figure 5.11: Illustration for occlusion issue when handling 3D models. A 3D model in the center with five 3D features. Two images, one at left and one at right. Due the the 3D position of the features both images show only some of the features, not all.

Another problem that will be very propense to occur is about the features associated OT change. Model points in 3D are projected into the image plane. The projection and the location of the features on images depend on the 3D rotation, the perspective effect, defined by the focal distance and the distances. In the end, our insumes are only features on images. Imagine a 3D object observed in an image A and the same object observed in image B from another viewpoint at a different 3D rotation angle. If we consider only the extracted features from A, we will obtain that A has an associated OT λ_A, analogy, B will also have an associated OT λ_B, we will desire that the both obtained $\lambda-$matrix will be the same since both images were generated with the same 3D model, however the effects of 3D rotations on the locations in images can severely affect the OT as is illustrated in figure 5.12. In the figure, we observe a triangular, quadrangular pyramid in two images with different viewpoints. In the first image we observe that if we consider all the five features of the model, we obtain the features arranged all in the convex hull set. In the second image, we observe the same projected model but it is very notorious a different features arrangement. We find four points on the convex hull and the remaining inside the hull. In the following, we will refer to this issue as OT local order disruption.

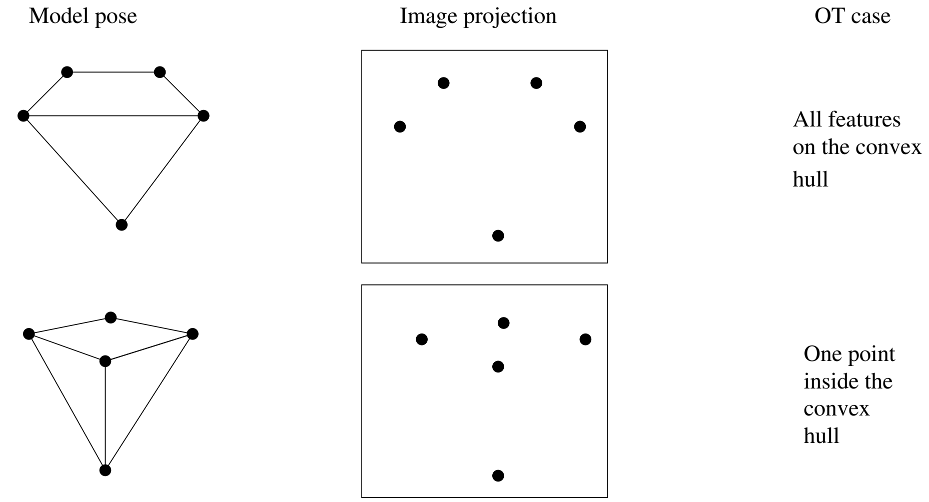

Model pose
Image projection
OT case
All features on the convex hull
One point inside the convex hull

Figure 5.12: Illustration for the OT disruption. In the left column we show the same model with different poses. In the central colum we show the features as they would be obtained in images. In the right column we show the OT case detected in the images. The OT in images changes with 3D rotations.

Some basis for solving the both detected issues, the occlusion, and the local order disruption, can be in the direction of considering only planar sections of the 3D model. An illustrative 3D model instance for this case is shown in fig 5.13.

A second approach can be more flexible, we could think about manifolds, i.e., if we assume that the 3D model is not planar, we could suppose that locally we could handle some sections of it as if they were. With this idea, this approach can be seen as a variation of the presented in the previous paragraph. Since the OT support some movement of their points without changing the internal orientations (as the maximal perturbation defines it), this idea can be an exciting strategy to study. This case is illustrated in the Fig. 5.14.

The to mentioned ideas, we see necessary first to detect planar and delimited sections, or regions that behave as planar, of images that enclose a region where a plane section of the 3D model is visible, in such a manner that it is possible to apply the direct plane to plane OT detection, features extraction and matching can be performed. These ideas are presented as future work. Thus they are out of the scope of this thesis, but they could represent improvements to the presented work, but also come with new and own issues.

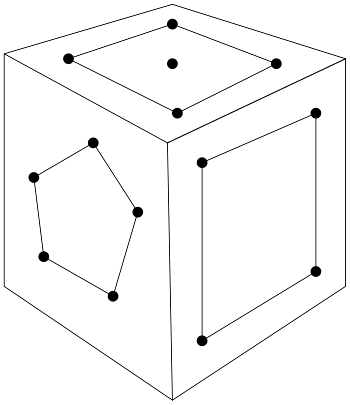


Figure 5.13: A 3D model composed by planes. Each plane on the model has an associated OT.

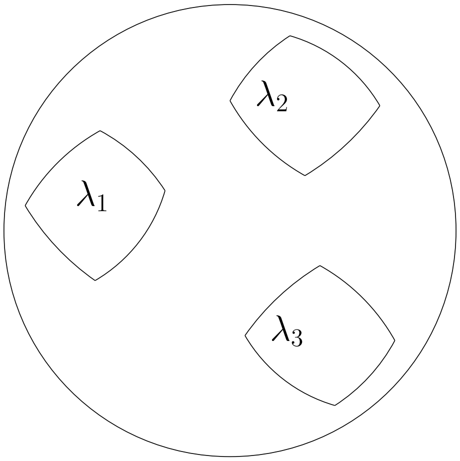


Figure 5.14: A 3D model with no planes. The model can be studied considering plane manifolds. In the figure three $\mathbb{R}^2$ manifolds are shown. These manifolds could be handled as if they were planes due the their local planarity.

5.4 Study remarks

In this Chapter we explored a joined approach, to perform a complete 3D reconstruction applying the simultaneous solution to various of the sub-problems involved in the 3D reconstruction. This joined approach involved the solution to the five sub-problems in two phases. In the first phase, we solved the features extraction and the features matching from input sequences of three images. The matched features obtained in this phase allows us to apply the other approach, in which we solve the self-calibration, the pose estimation, and the model retrieval simultaneously. These both approaches are detailed in the Chapters 3 and 4, respectively. With the application of the two approaches, we show the real applicability of the proposed methods to the 3D reconstruction problem. The more noteworthy result in this Chapter is the validation of the joined proposed approaches in Chapters 3 and 4, and also the validation of a new way of solving the reconstruction problem. With our proposals we show that the 3D reconstruction problem can take advantage of multiple pipeline sub-problems to propose new ways to handle the problems but also we show that the inclusion or exploitation of information that is present in multiple sub-problems can help to view the main problem differently, and thus allowing us to explore new methods for the solutions.

In the presented experiments of this Chapter we showed the 3D reconstruction process using the proposed OTTs. The experiments performed in this Chapter were performed with a specific tag but the method is general, and it will work with other tags with arbitrary points distributions. The contribution of this Thesis is to show that we can, in fact, obtain the scene reconstruction, including the model, using only the information of the input sequence of images. Another advantage of the proposed approaches is that the images and the reconstructed models can be identified through the associated $\lambda-$matrix, this is interesting since, it means that we are not only solving the reconstruction but also, simultaneously we are identifying in an automated approach the model in the scene.

Chapter 6

Conclusions

In this thesis, we studied the 3D reconstruction problem. As mentioned in the introduction on page 10, the 3D reconstruction is composed of mainly five sub-problems, i.e., the features extraction, the features matching, the estimation of the camera intrinsics, the pose estimation and the model retrieval. In the literature, these five subproblems have been solved in an isolated way. This isolated approach leads to a pipeline where the solution (output) obtained from each of the sub-problems is the input for another sub-problem. In this sense, we can say that as well as the solutions are transferred between sub-problems, the errors or misestimations are also transferred. The errors obtained in one sub-problem are propagated to the next sub-problem, and in the end, all the errors in the pipeline are accumulated in such a way that the final reconstruction is affected. The errors in each sub-problem must be minimized to try to obtain good solutions (scene reconstructions) at the end. In this traditional approach, in general, we could say that other authors have focused on developing the best methods to solve each of the five sub-problems separately. This single sub-problems approach has been exploited as much as possible, however, even considering the most popular methods we can find that they still have some associated issues that represent open problems.

In this thesis, we aimed to explore the solution to simultaneous 3D reconstruction sub-problems. We wanted to research the relationship between the 3D reconstruction sub-problems to search new solution approaches that exploit the found relationships. We hypothesized that the sub-problems could be simultaneously handled since the sub-problems are related, and even more, the information among different sub-problems can contribute for solving the involved sub-problems in the simultaneous approaches. With this idea, we focused the research on two methods. In the first one, the presented in Chapter 3 we studied the simultaneous self-calibration, the pose estimation, and the model reconstruction. We studied the three contiguous sub-problems, which involves the three core elements of the reconstruction that we aim to obtain, this is, the intrinsic camera parameters, the extrinsic parameters related to the images and the estimation of the model, these three elements constitute all the elements of a scene reconstruction. In this approach, we receive as input only the matched features in three images, but if more images are given, an incremental reconstruction approach can be applied. We proposed to solve the three sub-problems simultaneously from three images as a single optimization problem. We assume that the scene observed in the three images maintains unchanged, and we exploit the rigidity of the scene. The cost function to minimize is the reprojection error. The formulated optimization problem is non-linear, for this reason, we proposed to solve it through a heuristic, more specifically, using the Differential Evolution (DE). The usual form to solve a non-linear problem is to find a way to estimate an initial solution, possibly given by a linear method, and then refine that initial solution using the Levenberg-Marquardt algorithm. The State of the Art methods to obtain a 3D reconstruction from Structure from Motion [9, 10] use extra information presented in the metadata of the photographs taken with modern cameras and freely available on the Internet. This information can be used to estimate an initial value for the focus of the camera. Only very recently [78] a method that uses only the point in the scene has been proposed. In [78] it is proposed a new method using the fundamental matrices to give an initial value for the focus of the camera, but still with respect to other image taken with a calibrated camera. It is clear that once the bundle adjustment –the non-linear algorithm as Levenberg-Marquardt– is applied to the same data –even with little different initial estimates– the solution of [78] and the proposed in this thesis must give the same global solution. This is a hypothesis, although seems valid and clear, that must be confirmed in the future work.

In the second approach, the presented in Chapter 4, we studied the simultaneous solution to the features extraction and the features matching sub-problems. For that purpose, we studied the OT as the base for the simultaneous solution. The OT is a concept from the Computational Geometry field that describes the sets of points regarding the orientations (signed area) of the composing triplets of points. Each set of points on the plane has an associated OT that identifies it, and its computation involves the estimation of the associated invariant labeling of the points which also identifies each of the points on the set. In this thesis, we studied those properties from the Computer Vision viewpoint. We first found that the OT is a concept that we can obtain and use from images generated with a camera sensor since the OT maintains unchanged to all the transformations involved in the image generation process, which involves 3D rotations, 3D translations, perspective and pixels generation. From that study, we proposed a new kind of fiducial markers, the Order Type Tags (OTTs), that serve to apply the OT concept to the 3D reconstruction problem but also for contribute to the field with new fiducial tags for auto-identification and pose estimation for augmented reality. The OTTs encode a set of points, the set of points encode a specific OT, which we exploit to solve auto-identification, and also the OT has the associated invariant labeling, of the points that we exploit to solve the matching. The invariant labeling of points within a set of them is the feature that allows us to solve the feature matching. This is because if two different set of points with same $\lambda-$matrix, after computation of the $\lambda-$matrices, each point on each set ends with an assigned label, then the points in one set of points with the same label in the other set of points are matched. We also found that not all the existing OTs can be used to solve the matching problem, but most of them can. In this few cases, there is an ambiguity in the points labeling when does not exists a unique minimal lexicographical $\lambda-$matrix. To solve this ambiguity, more information is needed, and we propose to add triangles, then it is possible to create an edges matrix (we build it in a similar way as the $\lambda-$matrix does) which involves the edges in the added triangles composing the OTTs. With this additional matrix we solve the ambiguity issue; thus we can use all the existing OTs for matching purposes. Additionally, we studied the robustness of OT to the effects of noise coming from the estimation of the location of the points from the images taken with a camera. We found and proposed to quantify the robustness for the sets of points through the Maximal Perturbation (MP) value. The MP indicates the displacement that the points in a set can suffer without changing the original associated OT. With this concept, we were able to found the most robust OT instances in such a way that now, we have a method to estimate the MP for any set of points and we can choose the more robust set of points for our Computer Vision applications.

As the final part of the thesis, the presented in Chapter 5, we studied a joined approach. The aim in that method was to join the two previously proposed approaches to solve the 3D reconstruction problem completely. In this joined method we showed that the correcly solution in two stages of the 3D reconstruction. We showed that the simultaneous features extraction and features work correctly and allow us to obtain the correct inputs for the next stage, the simultaneous solution to the camera self-calibration the pose estimation, and the reconstructed model. With this joined approach we showed that we could perform the complete reconstruction of scenes containing a single instance of the proposed OTTs. The indifference with the methods in the SoA., we performed the features extraction, from the OTTs without the use of textures, we obtained the features at subpixel precision with the exploitation of the OTT design and we simultaneously and correctly solved the features matching in all our experiments. The extracted features and matched features from three images. Allows us to apply the second phase, the application of the approach presented in Chapter 4 , to obtain the reconstruction of the ground truth scene as well as the camera intrinsic parameters. In this study, we found that in most of the cases we can obtain solutions with a relative error below five percent. Results that depend on the difference of rotation between images (the test scenes are generated randomly with a minimal difference of 10 degrees) but that can be improved if we use images with a higher angular difference, which can be interpreted as more useful information from images. From this study, we found that the work proposed in this thesis is applicable to perform the complete scene reconstruction using the proposed OTTs but also it allows us to find new directions to the presented research. This new direction is the application of the results in this thesis in the direction of reconstructing scenes using the OT features without the need of the OTTs. This new problem is an open problem and represents an exciting and new approach to contribute to the field.

6.1 Future work

Still, there are a lot of ideas that must be verified that result from this work:

1. Compare the solution of the proposed method here and the obtained with the initial estimate of the focal length provided in [78].

2. Verify if the optimization problem is convex. From our studies and our research in Chapter 3 and Appendix A, we found that the problem to estimate the pose from a single planar view has two minima. But we still do not if the general problem presented in Sec. 3.1, in page 29, is convex o have several local minima. If the problem is convex, then a heuristic must no be used because there are very efficient algorithms to solve a convex problem with linear complexity.

3. Propose other non-linear problems that could be solved using evolutionary algorithms.

4. Generalize the use of Order Type (OT) for any reconstruction from planes. To reach this point also the followings points should be addresed.

5. Investigate how a λ-matrix obtained from less o more points can be used for matching.

6. Could be possible to use robust statistics to detect an OT with noisy point positions or in presence of dozens of points?

7. To build an open software package to detect a set of Order Type Tags (OTTs).

8. To maximize the Maximal Perturbation values of a set of points. This can produce a set of OTTs with maximal noise resistant.

9. Use the created OTT to solve other problems in the pipeline of Structure of Motion.

10. The notion of OT is kept in higher dimensions. How OT can help us in 3D?

Appendices

Appendix A

Design of Order Type Tags instances

In this section we detail how to use Order Type tags for automatic identification, point matching, and pose estimation for augmented reality applications. We explain how we perform the selection of OT instances, the tags design from the chosen OTs for their use in an augmented reality application.

As mentioned in Chapter 4, in page 55, OT instances are sets of points on the plane with a certain configuration that allows to characterize them and even more, for most of instances, to uniquely identify each of their points from images taken with a camera. We denote each set of points as C_l^k, where k is the set of points cardinality and l indicates the instance index. As mentioned in Chapter 4 the number of different OT instances have been studied by Oswin Aichholzer *et al.* in [1]. They provide the number of different OT instances but also they give a set of points instance for each OT. Here we explain the OT instances provided in [1] to select and to generate the proposed OTTs.

We studied in Chapter 4, that all OT instances are useful to perform automatic identification but not all of them are useful for automatic point matching. We also observed that OT instances are robust to positional noise. This means that the OT maintains unchanged even when the points that conform it are contaminated with noise. Not all OT instances support the same amount of noise, some of them are more robust to noise than others. We measure each set of point robustness through the Maximal Perturbation coefficient $\mathrm{MP}(C_l^k)$, which indicates the displacement in pixels that can be added to any of the points to C_l^k without changing its OT.

For an extended introduction of OTs, their origin and their design details, see the Chapter 4.

OTT is a proposal of this thesis for the application of OT and their instances. They are fiducial tags that allow to perform automatic identification, automatic point matching, and pose estimation for augmented reality applications. The concept of fiducial tags in Computer Vision, is a finite set of well known markers that are easy to detected with a camera, to identify (to distinguish which of the possible markers is present in the scene), and to compute the pose for it (the relative translation and orientation between the tag and the camera).

Depending on the application, if the number of distinct required tags is well known, it is desirable to use those tags that give the best performance. In OTTs, this would be the tags that are more robust to noise, due to the noise is one of the factors that can lead to tag identification errors.

In the following we describe the methodology we use to generate n OTTs for augmented reality applications.

1. Define the n number of required different OTTs. The lowest the value for n is, the more robust to noise the tags will be.
2. Choose the value for k (the cardinality for the set of points) using the Tab. A.1, such that $|E^k|$ has the minimum value but greater or equal than n. The set of points that conform E^k correspond to the OT instances with cardinality k that are suitable for point matching and thus for pose estimation.
3. Compute the maximal perturbation for each set of points in E^k, i.e., MP(E^k) and sort in descendant order the elements E^k accordingly to MP(E^k). The n first elements E^k after the sort are those set of points that are more robust to noise and are the ones that must be chosen.
4. Use each of the the n set of points from E^k with the biggest MP(E^k_l) obtained in the step 3 to build the OTTs as described in Sec. 4.6.

Table A.1: The number of OTs suitable for point matching and pose estimation.

Set	$\|E^k\|$ =Number of OTs
E^5	2
E^6	11
E^7	13
E^8	3303

As described in Sec. 4.4, the OTTs are a black square with triangles inside. Each triangle's vertex correspond to a point of the point set C^k_l or E^k_l. Since in this section we focus on those OT instances that allow point matching and pose estimation, given a set of points E^k_l, the aim is to find a set of triangles that use all points that conform E^k_l with the following rules:

- Each point in E^k_l must be used as a vertex of at least one triangle.
- Two different triangles can share at most one vertex.
- The number of triangles must be minimized.
- The area of all the triangles must be maximized.

In this work we manually tried to satisfy the mentioned rules, but the automatic Order Type Tags generation is an open problem that could be worked as future work.

In order to illustrate the process to generate an Order Type Tag from an Order Type Instance, we explain the process for constructing the tag for C_3^7, which is the set of points used to illustrate the proposal in Sec. 4.4.

Given the set of points $C_3^7 = \{[206,159],[214,127],[176,49],[42,144],[47,175],[129,178],[149,206]\}$ we first compute the Delunay triangulation and we plot it as shown in Fig. A.1.

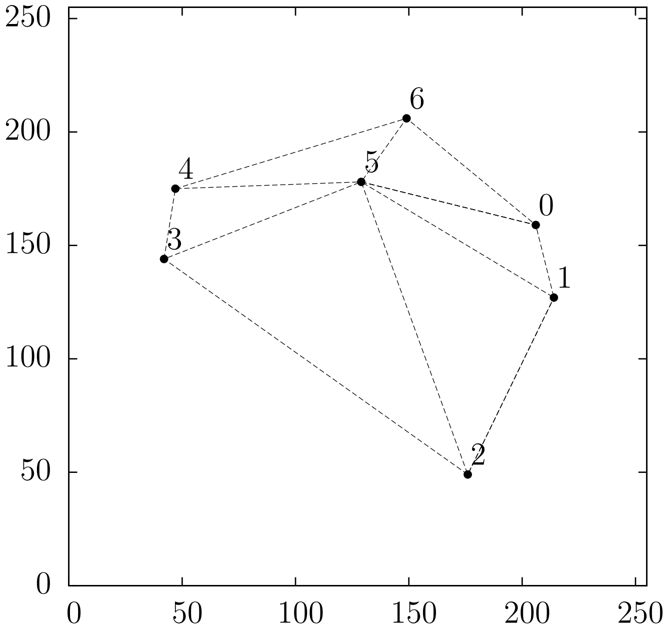


Figure A.1: Order Type instance C_3^7 and its Delunay triangulation.

As second step we visually identify the triangle with the highest area, in this case the formed by the points 5, 3, 2. If we choose 5, 3, 2, the rest of the points can be used by the triangles 4, 5, 6 and 5,1,0 since two different triangles can only share one vertex. This selection would give us a very big triangle and the other two with small area. Another alternative is to choose the triangle 5, 2, 1, if we choose this one, we can use the rest of the points with the triangles, $5, 6, 0$ and $7, 5, 9$. For this example we use the second option, the one that includes 5, 2, 1, since it seems to give triangles with more similar areas.

Once the triangles are identified we generate the tag by generating a black filled square and by putting the selected triangles inside in white. We used gnuplot to generate the tag, the code used is shown in Fig. A.2.

The obtained tag is the shown in Fig. A.3 and it can be printed used a conventional laser printer for using it for augmented reality applications.

As mentioned the process is manual, we use the Delunay triangulation since it tends to give equilateral triangles, since all minimum internal angles are maximized,

```
set terminal png size 1050,1050 #Size for the output image
set output ’tag.png’        #Name for the output image
set xrange [-15:270]        #We consider a square 15 unities more than the max,
set yrange [-15:270]        #this avoids triangles be sufficient far from the border.
unset xtics                 #Hide x axis ticks
unset ytics                 #Hide x axis ticks
unset border                #Hide graph border
unset key                   #Hide graph key
set size ratio -1           #Set squared aspect ratio
#Draws black square.
set object 1 polygon from -15,-15 to 270,-15  to 270,270 to -15,270 to -15,-15
set object 1 fc rgb ’black’ fillstyle solid 1.0 border lt -1
set lmargin  4;
set rmargin  4;
set tmargin  4;
set bmargin  4;
#We define and draw first triangle
set object 2 polygon from 129.0,178.0 to 176.0,49.0 to 214.0,127.0
set object 2 fc rgb ’white’ fillstyle solid 1.0 border lc rgb ’white’
#We define and draw second triangle
set object 3 polygon from 129.0,178.0 to 206.0,159.0 to 149.0,206.0
set object 3 fc rgb ’white’ fillstyle solid 1.0 border lc rgb ’white’
#We define and draw third triangle
set object 4 polygon from 129.0,178.0 to 47.0,175.0 to 42.0,144.0
set object 4 fc rgb ’white’ fillstyle solid 1.0 border lc rgb ’white’
plot ’point.dat’ lc rgb’black’ ps 0.001 #We force gnuplot to draw a point 0,0
```

Figure A.2: Code for generating an Order Type tag from the definition of triangles.

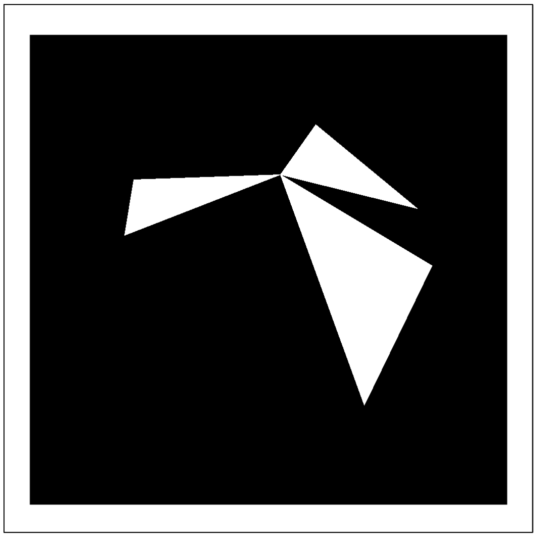


Figure A.3: Instance of the proposed Order Type Tags. The tag was constructed using the 3-th point set with cardinality 7 of the database in [1]. Point set $C = \{[206, 159], [214, 127], [176, 49], [42, 144], [47, 175], [129, 178], [149, 206]\}$.

and also to help to visualize the possible triangles. As mentioned, the automatic OTT generation is left for future work.

Appendix B

The pose ambiguity problem

In this section, we describe the problem of having multiple minima in the plane-based pose estimation problem. As described in Chapter 3, in page 27, the 3D reconstruction problem involves different sub-problems, including the pose estimation. Thus, if the pose estimation problem has multiple local minimums, then also the 3D reconstruction problem does.

In the early years, the Computer Vision community thought that the pose estimation from noisy measurements was a problem with a single minimum. Multiple approaches were developed over this idea like homography decomposition methods [34, 79] that minimized an algebraic error and PnP methods [54, 80, 81] that minimized a geometric error. The latter methods work by finding an initial solution for later refine it by minimizing a cost function through local and iterative optimization techniques. Although the methods were theoretically correct, in practice, the augmented reality and pose estimation implementations showed some visual discontinuities (jumps) when showing virtual objects for augmented reality applications or when tracking objects pose with a camera. Recently, the problem was studied by Schweighofer and Pinz in [42]. Authors showed that the origin of the jumps was the existence of more than a single local optimum in the pose estimation, specifically in the rotation parameters. Experimentally, authors found that in general exist only two local minimums for the pose parameters.

From this point, new methods that expect to find two local minimums have been proposed [42, 43]. The main idea of the new approaches is to find multiple initializations for later iteratively refine each of the initial solutions to a local optimum, although in [43] a closed-form method that directly tries to reach the global minimum is proposed. As the final solution, the one with the lowest cost function value is selected, which ideally would be the global minimum.

In order to clarify the ambiguity problem, in the rest of the section, we reproduce the results reported in [42] about the multi-modality of the pose estimation problem.

For a given plane in 3D scene and its projection on image space, there is a correct pose associated, the ground truth. This pose depends on the six parameters (three rotation angles and the translation vector with three elements) as well as the intrinsic camera parameters. If we assume to know priorly all the pose parameters, except one of them (a rotation angle), then the task is to estimate the value for that unknown angle.

To observe the two local optimums we set a synthetic scene. We use a planar pattern (four points forming a square with two units per side and centered at the centroid). We generated 6 images using constant intrinsic and extrinsic camera parameters, except for the distance from the camera to the tag (the t_3 parameter). We set the camera intrinsic and extrinsic parameters as shown in Tab. B.1, and we vary t_3 from 5.5 to 10.5 in increments of one unit to obtain the six images.

Table B.1: Scene intrinsic and extrinsic parameters for the performed experiments.

Intrinsic		Extrinsic	
f_x	1000	θ_1	0°
f_y	1000	θ_2	50°
s	0	θ_3	0°
u_0	320	t_1	0
v_0	240	t_2	0
Image size	640 × 480		

For each image we assume we don't know the real value of θ_2 and we look for it in exhaustively form. We try each possible value of θ_2 from $-100°$ to $-100°$ in changes of one degree. For each value of θ_2 we project the square model to image to compute the squared reprojection error of the four points. The result of this experiment is shown in Fig. B.1.

The results in Fig. B.1 show that squared reprojection error at the ground truth value, $\theta_2 = 50°$, is close to zero (1.9698), despite that it there exits other local optimum with a higher reprojection error. The value for the second local optima of θ_2 for each of the six images is shown in Tab. B.2. The other local optimum has a higher reprojection error than the ground truth, but it is reduced as long as t_3 increases, this is because at farther distances, the reprojection of the model is shown in a smaller area of the image, thus errors are smaller too. In Fig. B.2 we show the reprojection of the ground truth (red) and the incorrect local optimum (dotted green) for $t_3 = 5.5$. As shown in Fig. B.2, the other local optimum comes from the fact that corresponding corners of the image and the reprojected square seem to be near at the negative value of θ_2 but far from the ground truth solution.

As complementary experiment we repeated the experiment to search not only for the θ_2 parameter but also for the θ_1. We set the distance from camera to the model to $t_3 = 5.5$ and we maintained the same parameters in Tab. B.1, except for θ_1 and θ_2.

Table B.2: Local optima for each value of t.

t	5.5	6.5	7.5	8.5	9.5	10.5
Local optimum	-45	-47	-49	-50	-48	-50
Squared reprojection error	14 728	7 517.3	4 222.9	2 556.0	1 618.8	1 090.0

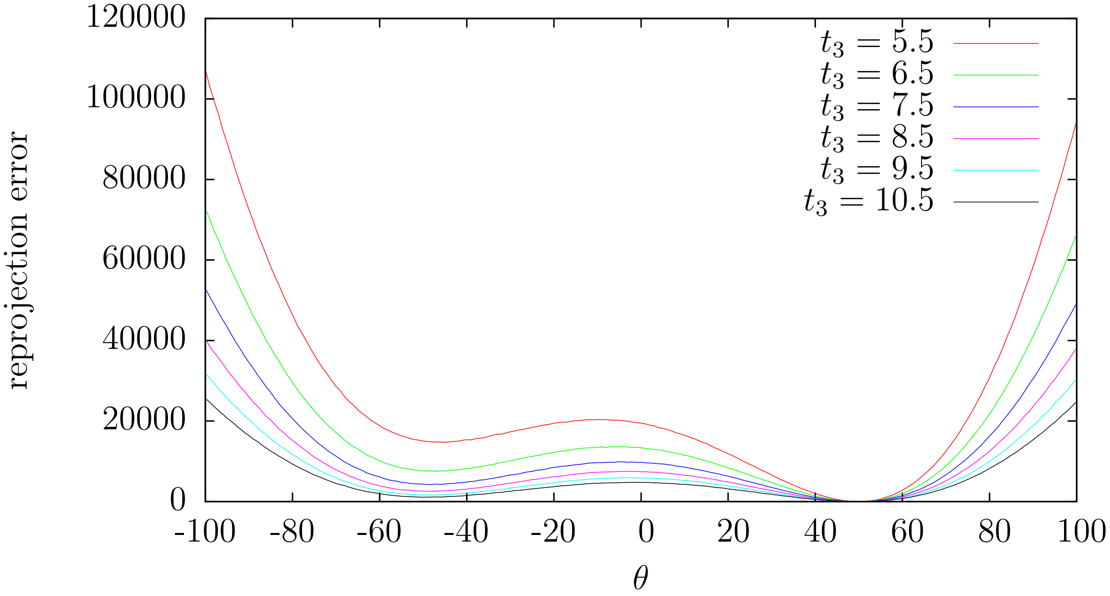


Figure B.1: Squared reprojection error associated to the different values of θ_2 for different values of t_3.

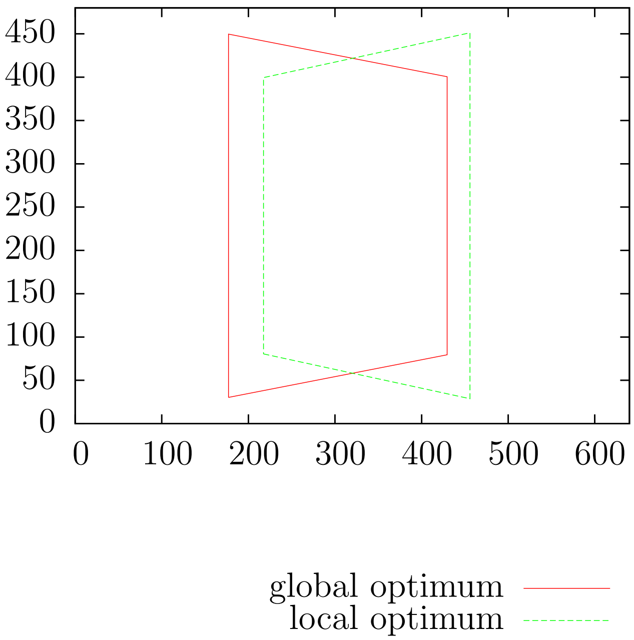


Figure B.2: Global optimum at $\theta_2 = 50°$, the local optimum at $\theta_2 = -47°$.

Similarly to the previous experiment we varied θ_1 and θ_2 in the interval $[-100, 100]$ in steps of one degree and we compute the squared reprojection error for each pair of values. The result of this experiment is shown in Fig. B.3. In the figure two local optimums are also observable, one at the ground truth $\{\theta_1 = 0°, \theta_2 = 50°\}$ where the squared reprojection error is 4.8805, and the other at $\{\theta_1 = -1°, \theta_2 = -45°\}$ with squared reprojection error of 1465.1. Visually, the comparative between incorrect local optimum and the ground truth is very similar to the shown in Fig. B.2 since the result obtained in the both performed experiments is differs in at most two degrees.

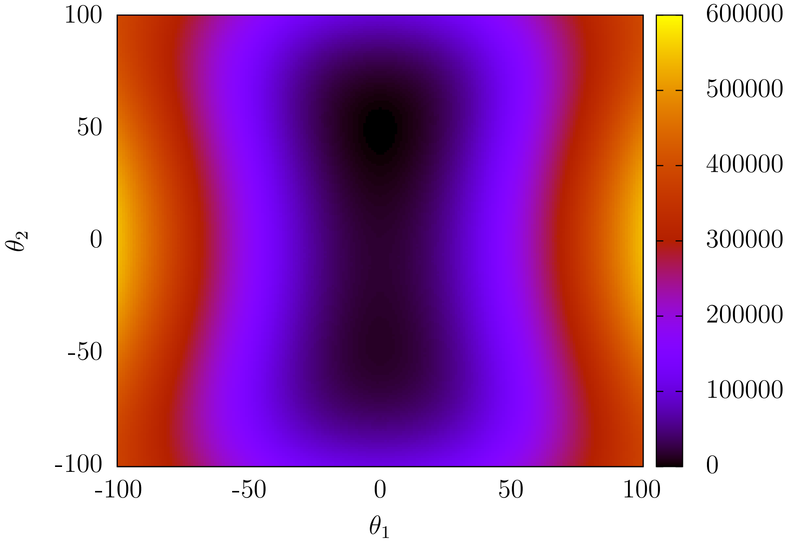


Figure B.3: Visualization of the two local optimums. Global optimum at $\{\theta_1 = 0°, \theta_2 = 50°\}$, the local optimum at $\{\theta_1 = 1°, \theta_2 = -45°\}$.

As shown in the results of the performed experiments, there exist two local optima for the solution to the pose estimation problem from planes. The obtained results coincide with the reported in [42]. This have some implications for the problem we are dealing with for several reasons. First, the pose estimation is one of the problems we solve simultaneously through Differential evolution. If the pose estimation problem have multiple minima, thus the 3D reconstruction also does. Second, in our approach of Chapter 3 we require to optimize simultaneously the pose for three images. If there exist two local optima for each image, then it is possible that the search space could have at least six local optima, just considering the rotation parameters of the image poses. Third, the nature of the problem serves as justification for using Differential Evolution as optimization technique, even thought, knowing the structure of the problem can be very useful for developing more efficient methods to solve the simultaneous self-calibration, pose estimation and model reconstruction as future work.

B.1 Pose precision

In this section we study the impact of using the OT instances point sets to compute the camera pose instead of using the four points arranged in a square as most of other tags do [82, 83, 84]. In literature, the methods to solve the pose estimation problem, PnP, homography decomposition (HD) and others [42, 43], all of them allow to estimate the pose from planar tags with at least four points. In augmented reality it is very common to find tags, fiducials, or camera calibration patterns that are given as planar squared objects. For this reason, most of applications and existing works, usually estimate the pose by using the four corners of these objects. When the problem is treated as a least-squares problem, as it occurs in the HD methods, it is expectable that increasing the number of points to solve the problem (assuming no outliers) we can obtain more accurate estimations. In this thesis we propose to use OT instances to perform automatic identification, point matching and also for camera pose estimation. As mentioned in Sec. 4, we propose to use point sets given on a plane with cardinality greater or equal than five. For this reason we present an analysis about using OT instances as model instead of the four corner points of a squared model. For our analysis we quantify the precision, specially when using images taken from long distances from camera to the target, since the precision tends to be less accurate as the distance increases. We analyze the effect of image noise in the generated experiments, and also the effect of the points distribution over the pose precision.

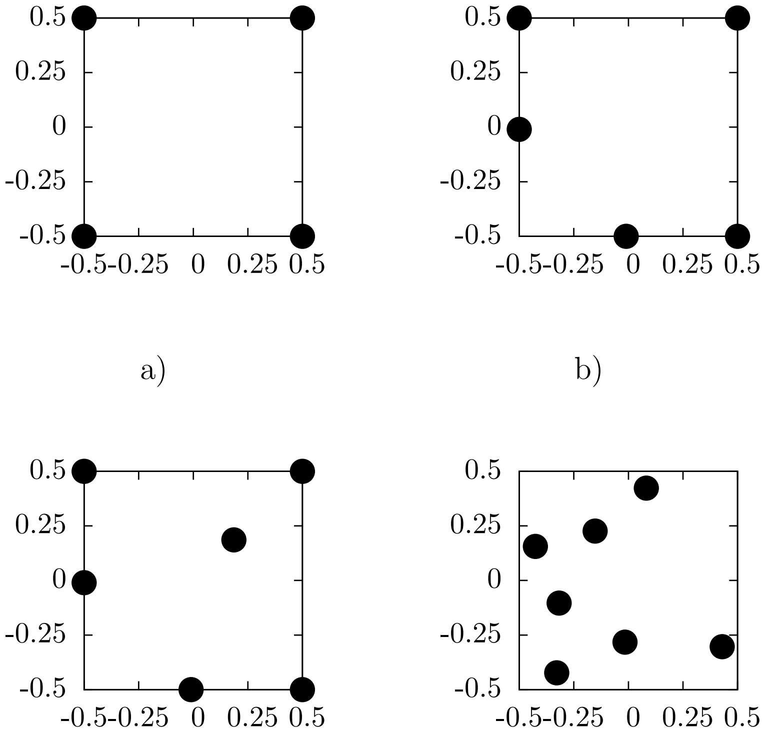
For our experiments we use two state of the art algorithms for pose estimation. The first (RPP) is the proposed in [42], which is an iterative pose estimation method that already considers the existence of two local optima, in such way that the global optimum is guaranteed. The other method (IPPE) is the proposed in [43], this is a non-iterative method that also considers the local optima.

For our analysis we performed four experiments with the base camera and scene configurations shown in Tab. B.3.

Table B.3: Base scene for the performed pose precision analysis.

Tag size (Model size)	1×1 a.u.
Rotation parameters	$\theta_1 = 0°$, $\theta_2 = -50°$, $\theta_3 = 90°$, $R = R_z(\theta_3)R_y(\theta_2)R_z(\theta_1)$
Translation parameters	$\boldsymbol{t} = [0, 0, 9]^{\mathrm{T}}$
Camera parameters:	$f = 700$, $o = 0.0$, $u_0 = 340$, $v_0 = 240$
Images size	640×480 pixels
Noise	Isotropic normal distribution with mean $[0, 0]$, and $\sigma = 1$

We compare the pose precision in the same scene but changing the number of points used to estimate the pose as well as their distribution. The sets of points and the distributions used for the experiment are shown in Fig. B.4.



a) b)

c) d)

Figure B.4: Set of points user for the pose precision experiments. Different number of points and different distributions.

For each set of points, we fixed the scene configuration of Tab. B.3, except the θ_1 rotation angle. We varied θ_1 from 0° to 180° in steps of 1° and we repeated the experiment 30 times. In some experiment instances, due to the added noise in the image coordinates, and due to the considerable distance from the camera to the targe, the correct pose is not estimated correctly. For each experiment instance, if the angular error with respect to the ground truth pose is less than five degrees, the pose estimation is considered as a successful pose estimation.

The global results of all the performed experiments is shown in Tab. B.4. We show the success percentage for the RPP and the IPPE methods. Additionally, we consider the methods coincidence percentage, i.e., if the RPP and the IPPE methods, for the same experiment instance return the same estimation with an angular error less than five degrees, then the solutions are considered as coincidence. If the coincidence is additionally succesfull (less than five degrees with respect to the ground truth pose) then, we detone the estimation as a successul coincidence.

Table B.4: Global results for the performed results.

	RPP	IPPE	Coincidence %	Success coincidence %
Set of points a)				
Success %	99.226	99.226	100	99.2265
Angular error mean	1.6858°	1.6816°		
Set of points b)				
Success %	97.7348	97.5322	99.3554	97.3112
Angular error	1.7357	1.7342		
Set of points c)				
Success %	98.5635	98.1952	99.1529	97.9558
Angular error	1.6949	1.6923		
Set of points d)				
Success %	88.1400	86.1142	93.3333	83.7937
Angular error	2.2465	2.2367		

A complementary perspective of the results is given in Fig. B.5 for the four, five, six, and seven points respectively. In the graphs we show the success percentage for the RPP, the IPPE methods, as well as the success coincidence percentage for each value of θ_1.

B.2 Results

In the results in Tab. B.4, we observe that the success percentage obtained with the RPP and the IPPE methods is very similar, and in three of the four cases, the angular error is bellow 1.8 degrees with success percentages above the 95%. This means that both methods, in most of the times, return the same pose estimation, as shown in the coincidence column with coincidence above 99% for three of the four set of points. Both methods also return good estimations, as shown in the success coincidence column. In three of the four cases the success 97% of the estimations were successful, except for the case **d)** with the 83.79%.

A similar behavior is observed in the Fig. B.5. In the plots corresponding to the set of points a), b), and c) we observe that both RPP and IPPE have very similar success percentages for all values of θ_1.

The greater success percentage, as well as the lowest angular error is obtained with the set of points **a)**, the corresponding to the squared model with four points at the corners. In this sense, the second best results are obtained with the set of points **c)**. The third best result is obtained with the set of points **b)** and the worst with the **d)**.

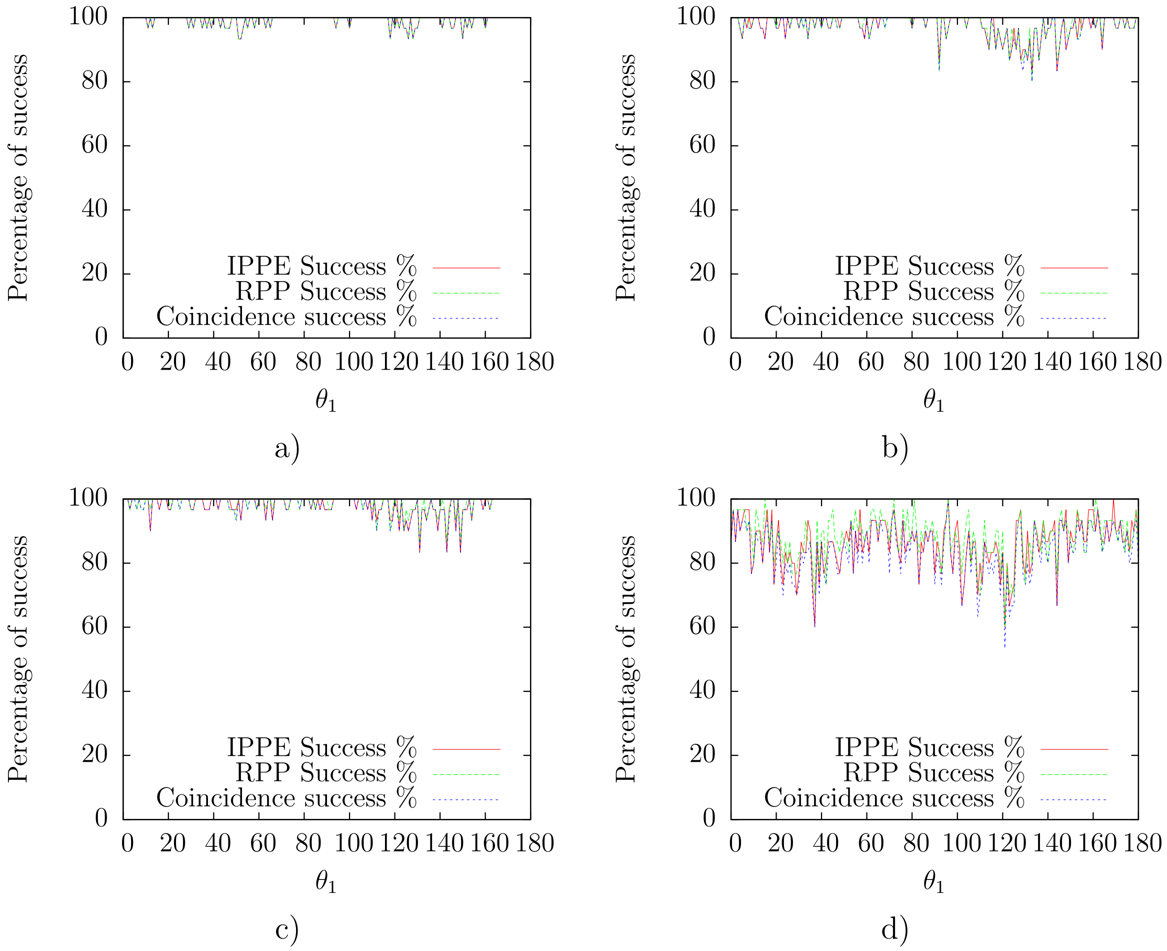


Figure B.5: Results for the pose precision. a) Results using the point set with four points. b) Results using the point set with five points. b) Results using the point set with six points. b) Results using the point set with seven points.

B.3 Discussion

We associate the results with the points distribution, to explain this hypothesis, lets analyze the following case. Let a model to be conformed by seven points, the first four ($\boldsymbol{P}_1$, $\boldsymbol{P}_2$, $\boldsymbol{P}_3$, and $\boldsymbol{P}_4$) at the corners of a square conforming the convex hull and the last three ($\boldsymbol{P}_5$, $\boldsymbol{P}_6$, and $\boldsymbol{P}_7$) inside the convex hull. If at least three points in the model define a small area in the model, then this three points will also tend to generate a small area in the image. If points in image define small areas, a pixel of noise will produce a greater estimation error, than the error that could be obtained when using model points with larger distances/areas (See Fig. B.6).

Since the pose is estimated using the complete set of points, the ones at the convex hull (externals) and the ones inside the convex hull (internals), and since that in the estimation all the points contribute equally in the pose estimation. If the number

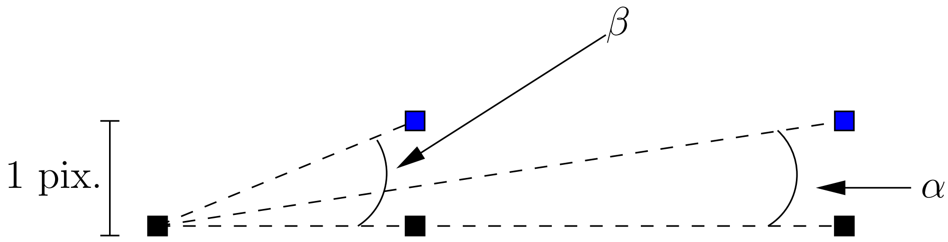


Figure B.6: Noise effect on points with distinct distances. Three points in left, center and right. The two left points are displaced by noise to the blue positions. The error angles is lower as distance increases, $\alpha < \beta$. Greater angle errors, are propagated in greater pose estimation errors.

of internal points is greater than the externals, the pose estimation will be solved in greater proportion with the information from the internal points, what will produce greater estimation errors. This hypothesis matches with the results in Tab. B.4, and we could expect to have better pose estimations if using the four corner points, as well as additional points on the squared border.

The point identification and the points matching capabilities of using OT instances are features that are not possible to obtain by using only four corners of a squared model. The fact that many of OT instances have multiple internal points, along with the analysis performed in this section, allow us to infer that using only OT instances points to estimate the pose will come with an associated error. To reduce this effect, it could be useful to design a tag that also exploits the corners of a squared model border to estimate the camera pose when using OTTs.

Bibliography

[1] Oswin Aichholzer, Franz Aurenhammer, and Hannes Krasser. Enumerating order types for small point sets with applications. *Order*, 19(3):265–281, Sep 2002.

[2] Sejong Heo, Jaehyuck Cha, and Chan Gook Park. Monocular visual inertial navigation for mobile robots using uncertainty based triangulation. *IFAC-PapersOnLine*, 50(1):2217 – 2222, 2017. 20th IFAC World Congress.

[3] Nathan Piasco, Dsir Sidib, Cdric Demonceaux, and Valrie Gouet-Brunet. A survey on visual-based localization: On the benefit of heterogeneous data. *Pattern Recognition*, 74(Supplement C):90 – 109, 2018.

[4] Ali Hosseininaveh Ahmadabadian, R. Yazdan, A. Karami, M. Moradi, and F. Ghorbani. Clustering and selecting vantage images in a low-cost system for 3d reconstruction of texture-less objects. *Measurement*, 99(Supplement C):185 – 191, 2017.

[5] Marine Fau, Raphal Cornette, and Alexandra Houssaye. Photogrammetry for 3d digitizing bones of mounted skeletons: Potential and limits. *Comptes Rendus Palevol*, 15(8):968 – 977, 2016.

[6] Hein A.M. Daanen and Agnes Psikuta. 10 - 3d body scanning. In Rajkishore Nayak and Rajiv Padhye, editors, *Automation in Garment Manufacturing*, The Textile Institute Book Series, pages 237 – 252. Woodhead Publishing, 2018.

[7] A. Riquelme, M. Cano, R. Toms, and A. Abelln. Identification of rock slope discontinuity sets from laser scanner and photogrammetric point clouds: A comparative analysis. *Procedia Engineering*, 191(Supplement C):838 – 845, 2017. ISRM European Rock Mechanics Symposium EUROCK 2017.

[8] Matilde Balaguer-Puig, ngel Marqus-Mateu, Jos Luis Lerma, and Sara Ibez-Asensio. Estimation of small-scale soil erosion in laboratory experiments with structure from motion photogrammetry. *Geomorphology*, 295(Supplement C):285 – 296, 2017.

[9] Mingwei Cao, Li Cao, Wei Jia, Yujie Li, Zhihan Lv, Liping Zheng, and Xiaoping Liu. Evaluation of local features for structure from motion. *Multimedia Tools and Applications*, 77(9):10979–10993, May 2018.

[10] J. L. Schnberger and J. M. Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, June 2016.

[11] Enliang Zheng and Changchang Wu. Structure from motion using structure-less resection. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2075–2083, 2015.

[12] C. Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision - 3DV 2013*, pages 127–134, June 2013.

[13] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Adaptive structure from motion with a contrario model estimation. In *Proceedings of the 11th Asian Conference on Computer Vision - Volume Part IV*, ACCV'12, pages 257–270, Berlin, Heidelberg, 2013. Springer-Verlag.

[14] Leonardo Gomes, Olga Regina Pereira Bellon, and Luciano Silva. 3d reconstruction methods for digital preservation of cultural heritage: A survey. *Pattern Recognition Letters*, 50:3 – 14, 2014. Depth Image Analysis.

[15] Habib Fathi, Fei Dai, and Manolis Lourakis. Automated as-built 3d reconstruction of civil infrastructure using computer vision: Achievements, opportunities, and challenges. *Advanced Engineering Informatics*, 29(2):149 – 161, 2015. Infrastructure Computer Vision.

[16] Sylvain Jay, Gilles Rabatel, Xavier Hadoux, Daniel Moura, and Nathalie Gorretta. In-field crop row phenotyping from 3d modeling performed using structure from motion. *Computers and Electronics in Agriculture*, 110:70 – 77, 2015.

[17] Heriberto Cruz Hernández and Luis Gerardo de la Fraga. A fiducial tag invariant to rotation, translation, and perspective transformations. *Pattern Recognition*, 81:213 – 223, 2018.

[18] Heriberto Cruz Hernández and Luis Gerardo de la Fraga. Order type dataset analysis for fiducial markers. *Data in Brief*, 2018.

[19] Luis Gerardo de la Fraga and Heriberto Cruz Hernandez. Point set matching with order type. In José Francisco Martínez-Trinidad, Jesús Ariel Carrasco-Ochoa, José Arturo Olvera-López, and Sudeep Sarkar, editors, *Mexican Conference on Pattern Recognition*, pages 229–237, Cham, 2018. Springer International Publishing.

[20] Heriberto Cruz Hernández and Luis Gerardo de la Fraga. Fitting multiple ellipses with pearl and a multi-objective genetic algorithm. In Leonardo Trujillo, Oliver Schütze, Yazmin Maldonado, and Paul Valle, editors, *Numerical and Evolutionary Optimization – NEO 2017*, pages 89–107, Cham, 2019. Springer International Publishing.

[21] Heriberto Cruz Hernández and Luis Gerardo de la Fraga. A multi-objective robust ellipse fitting algorithm. In Yazmin Maldonado, Leonardo Trujillo, Oliver Schütze, Annalisa Riccardi, and Massimiliano Vasile, editors, *NEO 2016: Results of the Numerical and Evolutionary Optimization Workshop NEO 2016 and the NEO Cities 2016 Workshop*, pages 141–158, Cham, 2018. Springer International Publishing.

[22] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, 2003.

[23] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[24] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proceedings of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[25] Bill Triggs. *Detecting Keypoints with Stable Position, Orientation, and Scale under Illumination Changes*, pages 100–113. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[26] M. Brown, R. Szeliski, and S. Winder. Multi-image matching using multi-scale oriented patches. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 510–517 vol. 1, June 2005.

[27] R. Szeliski. *Computer Vision: Algorithms and Applications*. Texts in Computer Science. Springer, Washington, USA, 1 st edition, 2010.

[28] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov 2004.

[29] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346 – 359, 2008. Similarity Matching in Computer Vision and Multimedia.

[30] Yan Ke and R. Sukthankar. Pca-sift: a more distinctive representation for local image descriptors. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–506–II–513 Vol.2, June 2004.

[31] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis an automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[32] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344, August 1987.

[33] Junhua Sun, Xu Chen, Zheng Gong, Zhen Liu, and Yuntao Zhao. Accurate camera calibration with distortion models using sphere images. *Optics & Laser Technology*, 65(Supplement C):83 – 87, 2015.

[34] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, March 2000.

[35] R.I. Hartley. Euclidean reconstruction from uncalibrated views. In *Proceedings of the Second Joint European - US Workshop on Applications of Invariance in Computer Vision*, pages 237–256, London, UK, UK, 1994. Springer-Verlag.

[36] M. Pollefeys, L. Van Gool, and A. Oosterlinck. The modulus constraint: a new constraint self-calibration. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 1, pages 349–353 vol.1, Aug 1996.

[37] Marc Pollefeys and Luc Van Gool. *Self-calibration from the absolute conic on the plane at infinity*, pages 175–182. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

[38] Cheng Lei, Fuchao Wu, Zhanyi Hu, and H. T. Tsui. A new approach to solving kruppa equations for camera self-calibration. In *Object recognition supported by user interaction for service robots*, volume 2, pages 308–311 vol.2, 2002.

[39] B. Triggs. Autocalibration and the absolute quadric. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 609–614, Jun 1997.

[40] Juan Li, Juan A. Besada, Ana M. Bernardos, Paula Tarro, and Jos R. Casar. A novel system for object pose estimation using fused vision and inertial data. *Information Fusion*, 33:15 – 28, 2017.

[41] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, Aug 2003.

[42] G. Schweighofer and A. Pinz. Robust pose estimation from a planar target. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2024–2030, Dec 2006.

[43] Toby Collins and Adrien Bartoli. Infinitesimal plane-based pose estimation. *International Journal of Computer Vision*, 109(3):252–286, Sep 2014.

[44] B. Triggs, P.F. McLauchlan, R.I. Hartley, and A.W. Fitzgibbon. *Bundle Adjustment — A Modern Synthesis*, pages 298–372. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[45] Timothy A. Davis and William W. Hager. Dynamic supernodes in sparse cholesky update/downdate and triangular solves. *ACM Trans. Math. Softw.*, 35(4):27:1–27:23, February 2009.

[46] Manolis I. A. Lourakis and Antonis A. Argyros. Sba: A software package for generic sparse bundle adjustment. *ACM Trans. Math. Softw.*, 36(1):2:1–2:30, March 2009.

[47] Sameer Agarwal, Noah Snavely, Steven M. Seitz, and Richard Szeliski. *Bundle Adjustment in the Large*, pages 29–42. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[48] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz. Multicore bundle adjustment. In *CVPR 2011*, pages 3057–3064, June 2011.

[49] Juho Kannala, Sami S. Brandt, and Janne Heikkilä. *Self-calibration of Central Cameras from Point Correspondences by Minimizing Angular Error*, pages 109–122. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[50] C. P. Lu, G. D. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, Jun 2000.

[51] Martin Byröd and Kalle Åström. *Conjugate Gradient Bundle Adjustment*, pages 114–127. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[52] Z. Cui and P. Tan. Global structure-from-motion by similarity averaging. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 864–872, Dec 2015.

[53] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *CVPR 2011*, pages 3001–3008, June 2011.

[54] S. Li, C. Xu, and M. Xie. A robust O solution to the perspective-n-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1444–1450, July 2012.

[55] R.I. Hartley and P. Sturm. Triangulation. *Computer vision and image understanding*, 68(2):146–157, 1997.

[56] H. Stewenius, F. Schaffalitzky, and D. Nister. How hard is 3-view triangulation really? In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 686–693 Vol. 1, Oct 2005.

[57] Laurent Fousse, Guillaume Hanrot, Vincent Lefèvre, Patrick Pélissier, and Paul Zimmermann. Mpfr: A multiple-precision binary floating-point library with correct rounding. *ACM Trans. Math. Softw.*, 33(2), June 2007.

[58] K. Price, R.M. Storn, and J.A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Natural Computing Series. Springer Berlin Heidelberg, 2006.

[59] Rainer Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359, December 1997. 10.1023/A:1008202821328.

[60] K. Zielinski and R Laur. *Stopping Criteria for Differential Evolution in Constrained Single-Objective Optimization*, pages 111–138. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[61] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, 10(6):646–657, Dec 2006.

[62] J. Ronkkonen, S. Kukkonen, and K. V. Price. Real-parameter optimization with differential evolution. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 506–513 Vol.1, Sept 2005.

[63] Jarosłlaw Arabas, Adam Szczepankiewicz, and Tomasz Wroniak. Experimental comparison of methods to handle boundary constraints in differential evolution. In Robert Schaefer, Carlos Cotta, Joanna Kołodziej, and Günter Rudolph, editors, *Parallel Problem Solving from Nature, PPSN XI*, pages 411–420, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[64] S.J. Ahn, W. Rauh, and H.J. Warnecke. Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola. *Pattern Recognition*, 34(12):2283–2303, January 2001.

[65] Oxford University. Visual geometry group datasets, 2015. http://www.robots.ox.ac.uk/~vgg/data/data-mview.html.

[66] R. M. Mersereau and A. V. Oppenheim. Digital reconstruction of multidimensional signals from their projections. *Proceedings of the IEEE*, 62(10):1319–1338, Oct 1974.

[67] A. Punjani, M. A. Brubaker, and D. J. Fleet. Building proteins in a day: Efficient 3d molecular structure estimation with electron cryomicroscopy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):706–718, April 2017.

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

[69] Luis Gerardo de la Fraga and Oliver Schütze. Direct calibration by fitting of cuboids to a single image using differential evolution. *International Journal of Computer Vision*, 81(2):119–127, Feb 2009.

[70] Greg Aloupis, John Iacono, Stefan Langerman, Özgür Özkan, and Stefanie Wuhrer. The complexity of order type isomorphism. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 405–415. SIAM, 2014.

[71] Jacob E. Goodman and Richard Pollack. Multidimensional sorting. *SIAM Journal on Computing*, 12(3):484–507, 1983.

[72] Jun-Sik Kim and Ho-Won Kim. A camera calibration method using concentric circles for vision applications. In *Proceedings of the 5th Asian Conference on Computer Vision*, 2001.

[73] E. R. Davies. *Computer Vision, Fifth Edition: Principles, Algorithms, Applications, Learning.* Academic Press, Inc., Orlando, FL, USA, 5th edition, 2017.

[74] J-S. Kim, P. Gurdjos, and IS. Kweon. Geometric and algebraic constraints of projected concentric circles and their applications to camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):637–642, April 2005.

[75] R. Johnsonbaugh. *Discrete Mathematics.* Pearson/Prentice Hall, 2009.

[76] G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library.* O'Reilly Media, 2008.

[77] Persistence of Vision (TM) Raytracer., http://www.povray.org, accessed = 2018-12-04.

[78] J.H. Brito. Autocalibration for structure from motion. *Computer Vision and Image Understanding*, 157:240–254, 2017.

[79] P. Sturm. Algorithms for plane-based pose estimation. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, volume 1, pages 706–711 vol.1, June 2000.

[80] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate O(n) solution to the pnp problem. *International Journal of Computer Vision*, 81(2):155, Jul 2008.

[81] J. A. Hesch and S. I. Roumeliotis. A direct least-squares (dls) method for pnp. In *2011 International Conference on Computer Vision*, pages 383–390, Nov 2011.

[82] P. D. Virulkar and A. N. Bhute. Comparative study: Location based mobile advertisement publishing system. In *Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on*, pages 570–574, Feb 2015.

[83] Hiroko Kato, Keng T. Tan, and Douglas Chai. *Barcodes for Mobile Devices*. Cambridge University Press, New York, NY, USA, 1st edition, 2010.

[84] Edwin Olson. AprilTag: A robust and flexible visual fiducial system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3400–3407. IEEE, May 2011.