Pedro Andrade Coelho Vieira

APE: AUTOMATIC POSE ESTIMATION

B.Sc. Dissertation

Federal University of Pernambuco
secgrad@cin.ufpe.br
www.cin.ufpe.br/~secgrad

RECIFE
December, 2017

Federal University of Pernambuco
Center of Informatics
B.Sc. in Computer Engineering

Pedro Andrade Coelho Vieira

APE: AUTOMATIC POSE ESTIMATION

A B.Sc. Dissertation presented to the Center of Informatics of Federal University of Pernambuco in partial fulfillment of the requirements for the degree of Bachelor in Computer Engineering.

Advisor: *Judith Kelner*

RECIFE
December, 2017

Pedro Andrade Coelho Vieira
APE: Automatic Pose Estimation/ Pedro Andrade Coelho Vieira. – RECIFE, December, 2017-
51 p. : il. (algumas color.) ; 30 cm.

Advisor Judith Kelner

B.Sc. Dissertation – Universidade Federal de Pernambuco, December, 2017.

1. Palavra-chave1. 2. Palavra-chave2. I. Orientador. II. Universidade xxx. III. Faculdade de xxx. IV. Título

CDU 02:141:005.7

To those that imagine things that never were and say "why not?"

Acknowledgements

Place here some acknowledgements.

Wir müssen wissen - wir werden wissen!

—DAVID HILBERT

Resumo

Operações marítimas como instalações de dutos submarinos de gás e petróleo são bastante comuns na indústria petrolífera. Nessas operações, os dutos precisam ser continuamente monitorados para evitar que haja excesso de tensão neles, o que resultaria em danos ao equipamento. O monitoramento pode ser realizado utilizando visão computacional, onde o duto é reconstruído em 3D usando um arranjo estéreo para que sua geometria possa ser estimada.

O processo de reconstrução 3D consiste em detectar pontos do duto nas imagens das câmeras e estimar os pontos 3D. Para tal, é preciso conhecer os parâmetros intrínsecos e extrínsecos de calibração do arranjo estéreo, que são valores usados na modelagem de cada câmera e a relação delas no arranjo. Os parâmetros intrínsecos modelam como pontos no mundo real são projetados para o sistema de coordenadas das imagens de cada câmera. Os parâmetros extrínsecos descrevem uma translação e rotação relativas entre as câmeras, e por isso são também chamados de pose relativa.

Os parâmetros intrínsecos dependem de cada câmera apenas, portanto não são alterados ao longo do tempo. Por outro lado, os extrínsecos são diferentes para todo arranjo estéreo que é montado antes das operações. Além disso, a pose relativa pode ser alterada durante as instalações, quando há colisões das câmeras com outros equipamentos. Atualmente, os parâmetros extrínsecos são estimados antes de cada operação e, portanto, não podem ser alterados durante as filmagens, em casos de alterações nos valores.

Essa monografia apresenta uma nova solução para a estimativa automática da pose relativa entre câmeras em um arranjo estéreo. Nossa solução é executada durante as operações usando imagens dos dutos a serem instalados. A implementação é dividida em duas etapas: uma etapa de *matching*, que busca correspondências entre pontos das imagens de cada câmera; e uma etapa de estimativa e otimização de pose, que computa os parâmetros extrínsecos a partir das correspondências da etapa anterior.

Realizamos testes em laboratório para avaliar a acurácia de nossa solução e a comparamos com uma técnica do estado da arte desenvolvida para este cenário. Os testes fizeram um ajuste de curva com pontos reconstruídos usando parâmetros extrínsecos estimados por nossa proposta e compararam a curva estimada com a original construída em laboratório. Os resultados mostraram que os pontos reconstruídos a partir de nossa proposta obtiveram, por ponto, um erro abaixo de 1 centímetro quando comparados com os pontos do gabarito obtido em laboratório, o que é considerado suficiente para a aplicação em questão. Nossa solução também obteve resultados mais acurados do que a técnica do estado da arte.

Palavras-chave: visão computacional; visão estéreo; correspondência de pontos; pose relativa; duto.

Abstract

Offshore operations such as the installation of oil and gas ducts are a frequent task in the oil extraction industry. During these operations, the duct requires continuous monitoring to avoid excessive tension along the line, which might cause damage to the equipment. To enable an efficient monitoring, the duct is virtually reconstructed with a stereo computer vision application and its geometry is then estimated.

The 3D reconstruction process consists of detecting points from the duct in the cameras' images to estimate the 3D points. This requires previous knowledge of the intrinsic and extrinsic parameters of the cameras, which are values used to model each camera and their relation in the stereo rig. The intrinsics model how the world points are transformed into the camera's image coordinates. The extrinsic parameters describe the relative orientation and position between the cameras, that is why they are also called relative pose.

The intrinsics only depend on each camera, therefore they do not usually change with time. On the other hand, the extrinsic parameters are different for every time the stereo rig is set before the operations. Besides, the relative pose can also be modified during the installations if the cameras collide with other equipments. The current method for estimating the relative pose between cameras in this application is executed before the operation, which means that there is no way to rectify the pose parameters if the stereo rig's configurations is changed during the operation.

This dissertation presents a novel method for automatically estimating the relative pose between cameras in a stereo rig. This solution is executed during the computer vision application using images from the duct to be installed. Our approach is divided in two stages: a matching stage, which aims to estimate correspondences between the camera's images; and pose estimation, which computes and optimizes a relative orientation and translation between them using the matches from the previous stage.

We performed laboratory experiments to evaluate the accuracy of our proposal and compared it with a state of art algorithm developed for the targeted scenario. The tests consisted of reconstructing 3D using poses computed with our solution and fitting a curve on them. The estimated curve was then compared with the original one in the laboratory. The results showed that the 3D reconstructed points based on our proposal had an average difference below 1 centimeter when compared to the original 3D points from the laboratory, which attends the standards of the targeted application. Our proposal performed more accurately than the state of art algorithm.

Keywords: computer vision; stereo vision; point matching; relative pose; duct.

List of Figures

List of Tables

List of Acronyms

DLT	Direct Linear Transformation 22
DP	Dynamic Programming 33
FLANN	Fast Approximate Nearest Neighbor Search 26
FOV	Field of View
MLESAC	Maximum-Likelihood Estimation by Random Sampling Consensus
PCA	Principal Component Analysis 27
PnP	Perspective-n-View 27
RANSAC	Random Sample Consensus 23
ROV	Remotely Operated Vehicle 13
SIFT	Scale Invariant Feature Transform 26
SURF	Speeded-Up Robust Features 27
SVD	Singular Value Decomposition 32

Contents

1

Introduction

Computer vision is a topic of interest in computer science. It aims to simulate or even overcome the human vision capabilities by extracting information from images within a specific context. One of the main applications of computer vision is 3D reconstruction, which uses 2D features from one or more images to obtain 3D information on the scene. The reconstruction is commonly achieved by using a stereo rig, which, in this work, is considered to be a pair of cameras with a common field of view. The cameras are separated by a relative translation and rotation between them.

Stereo 3D reconstruction consists of using the data captured on the cameras' images along with some camera model to solve the inverse of the projection that generated the cameras' images and estimate the real 3D structure of a scene (HARTLEY; ZISSERMAN, 2004). The usual pipeline for 3D reconstruction consists of: initial image processing, detection of features, matching and triangulation. Both the matching and triangulation steps of this pipeline require some input parameters to represent the camera model and the cameras' relative position and orientation. We call these inputs *calibration parameters* and they are divided into two groups: Intrinsics, which represent the camera model; and Extrinsic, which is also called Relative Pose and consists of a rotation and a translation of one camera relative to the other. With the calibration parameters, it is possible to represent both cameras mathematically and, thus, to solve several computer vision problems.

The possibility of reconstructing 3D scenes from camera images in computer vision allowed us to explore scenarios where it is impossible or too dangerous for humans to go, which is the case of underwater offshore oil and gas extraction. Overseas operations are challenging since they are complex and require reaching the extraction point, which is usually in deep underwater environments and can surpass a thousand meters in depth. In this work, we focus on deep underwater installations of oil and gas ducts, which we will also call pipes. Those are flexible and might get mishandled during the process, which could ruin the whole operation. Thus, it is common to use a Remotely Operated Vehicle (ROV) equipped with a stereo rig to assist most of the operation and avoid wrong duct configurations. An example of ROV is depicted in Figure 1.1.

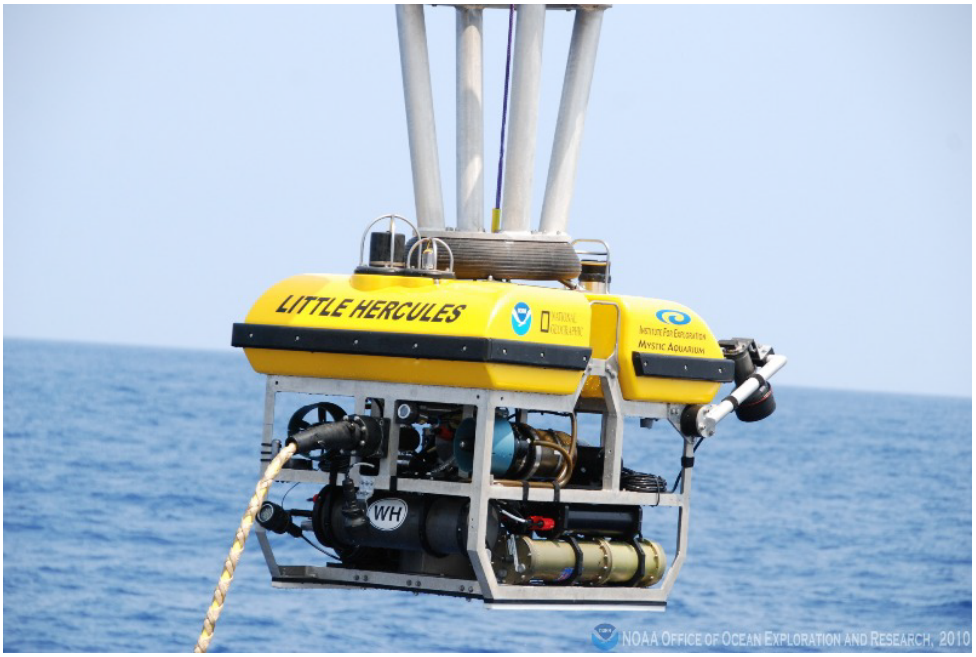


Figure 1.1: An example of a remotely operated vehicle for underwater environments.

For acquiring the calibration parameters of the ROV's cameras, it is common to have a separated calibration stage before the operation, where a predetermined pattern is presented to the stereo rig, and a calibration algorithm is used to estimate those parameters. While the cameras' intrinsics depend on each camera and do not vary in successive uses, the extrinsic parameters change each time the cameras are placed on the ROV and can also alter with pressure or collisions during the operations. In these situations, the 3D reconstructed scene might become unreliable and this results in a high cost and loss of time. Moreover, the current calibration process is a separated stage of the operation and, therefore, takes time and human effort. A method that executes the calibration process automatically and without further human intervention could be an alternative for minimizing the operation time and the human error.

One way to solve this problem is using an automatic calibration algorithm that can estimate the relative pose with images from the scenario of an application instead of a predetermined pattern. Before an offshore oil or gas extraction begins, all the equipment and the ducts that are going to be used in the procedure must submerge and be set where the operation will take place. In this period, the cameras are already recording and this would be the ideal moment for calibrating them since it wouldn't retard the operation. Besides, this algorithm could execute in parallel with the main application and check the extrinsic parameters periodically in case of variations in the cameras' relative pose.

In Section 1.1 we detail the main objective of this work. Section 1.2 details the problem's scenario and other aspects of the cameras, ducts and illumination that are relevant to our solution. In Section 1.3 we present this document's structure and its chapters' content.

1.1 Objective

This work has the objective of proposing a novel solution for estimating the relative pose between a pair of cameras in a stereo rig. The presented solution focus on offshore deep underwater oil and gas extraction operations, and must calibrate the extrinsic parameters with only images of the ducts used in such activities.

Since we consider that there is no previous pose calibration, this solution must include a matching stage that is able to estimate matches between images of both cameras without pose knowledge. Both the matching stage and the pose estimation stage must be robust to noise and outliers because of the low illumination and suspended particles present in the water. Additionally, our solution must have a similar accuracy to a stereo calibration algorithm that is used in this scenario (SANTOS et al., 2015).

1.2 Scenario

During the installation procedure of oil and gas ducts, the duct and the ROV usually are less than 30 meters above the seabed, which has a depth between 1000 and 4000 meters. The ROV keeps a distance between 5 and 15 meters from the flexible pipe and it performs slow movements to capture duct images from different angles.

The two cameras are firmly attached to the ROV with approximately parallel optical axes and 80 centimeters of horizontal distance between them. This setup was chosen to enable reliable 3D reconstruction with points at more than 5 meters away from the stereo rig. The current calibration of the cameras happens before the ROV's submersion and takes into account the water's refractive index.

The stereo rig used in the installation of gas and oil ducts consists of two cameras Kongsberg OE15-101c with a 82 degree field of view when submerged. Both cameras of the stereo rig are resistant to water and high pressure because of the depth in which the operation occurs. Since there is no natural light at this depth, the ROV is equipped with an artificial light. However, the light attenuation is significant in this depth and, thus, the cameras are built for low illumination input with 10^{-2} to 10^{-5} LUX of light sensibility. These cameras are monochromatic and have wide-angle lenses, which means that there is an accentuated radial distortion in the images. Figure 1.2 shows the camera used in our scenario.

The ducts in this scenario are flexible and can assume various forms. Their diameter is between 20 and 50 centimeters, and is approximately constant. The pipe's length has a few kilometers, but only the last 10 to 30 meters are captured by the stereo rig. To facilitate the pipe's detection by a computer vision algorithm, it is painted with the regular pattern depicted in Figure 1.3. In this pattern, the white areas must have a length d similar to the pipe's diameter and the dark regions a length of $d/2$. The white areas of the pattern are detected as blobs, which we will also call vertebrae (PESSOA et al., 2015).

Figure 1.2: The model of the camera used in our scenario.

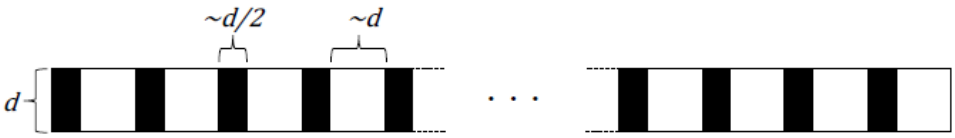


Figure 1.3: Painting pattern of the duct to assist its detection in the images.

The duct is suspended by its extremities, which makes it almost coplanar during the operation. Because of that and since the duct is detected by its vertebrae, we consider the duct a sequence of points from a planar curve in a 3D environment. This serves as a constraint to help detect, match and reconstruct the pipe, and also aids to estimate the relative pose in our proposal.

The cameras' images have a 640×480 pixel resolution, are monochromatic and slightly blurred because of the low illumination underwater scenario, as shown in Figure 1.4. Since the pipe is the only illuminated object in the scene, its vertebrae are brighter than the background, which enables the duct's detection as a set of blobs.

1.3 Document Structure

The next chapters are organized as the following: Chapter 2 provides an overview of some basic concepts related to this work, such as projective geometry and the algorithms used in our proposal. In Chapter 3, we review related research on point matching and relative pose estimation. Chapter 4 presents our proposal and details its implementation based on the two previous chapters. In Chapter 5, our proposal is tested using images from a controlled laboratory environment. Finally, in Chapter 6, we present our final considerations and suggest some future works.

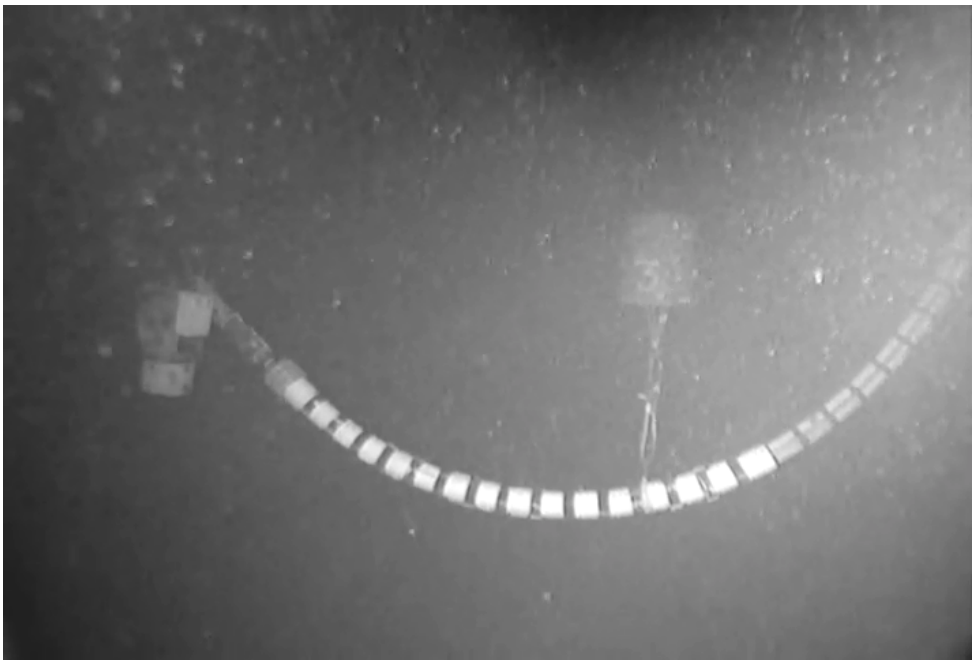


Figure 1.4: Example of a frame depicting the duct during the operation. It is possible to notice that the background is darker than the pipe's vertebrae and the image is slightly blurred.

2

Theoretical Foundation

In this chapter, we present the main theoretical foundation to help contextualize the methods described in this document. The mathematical notation used in this work is described in Section 2.1. We give a brief presentation of the projective geometry in Section 2.2. Section 2.3 describes the pinhole camera model, which will be used to model all cameras in this dissertation. In Section 2.4, we present the basics of epipolar geometry. Section 2.5 describes the stereo rectification process used by some works in this monograph. In Section 2.6, explains a basic triangulation algorithm used for the 3D reconstructions in our proposal and experiments. Section 2.7 presents the RANSAC, which is an random iterative estimator widely used in computer vision.

2.1 Notation

The notation used in this dissertation is similar to the one used by HARTLEY; ZISSERMAN (2004). Scalar variables are represented using English or Greek letters in upper or lower case as in A, a and α.

Vectors are represented with English or Greek letters in upper or lower case and bold as in $\mathbf{A}$, $\mathbf{a}$ and $\boldsymbol{\alpha}$. Elements in a vector are enumerated using brackets such as $\mathbf{v} = [v_1, v_2, v_3]^\mathsf{T}$, where $\mathbf{u}^\mathsf{T}$ is the transpose of a vector $\mathbf{u}$. By default, we consider that all vectors are column vectors, *i.e.* a matrix with just one column. Given two column vectors $\mathbf{u}$ and $\mathbf{v}$, their dot product and their cross product are $\mathbf{u}^\mathsf{T}\mathbf{v}$ and $\mathbf{u} \times \mathbf{v}$, respectively. When the vectors are two matching points $\mathbf{x}$ and $\mathbf{x}'$ from stereo rig's images, we use the symbol $\leftrightarrow$ to represent the match as in $\mathbf{x} \leftrightarrow \mathbf{x}'$.

Matrices with two or more columns are represented with English letters in upper case and sans serif font as in A and H. Since both camera projection matrices and pose matrices are represented as P, we opted to represent the first one with the script letter $\mathscr{P}$. The transpose of a vector A is represented as A^T. The multiplication between a matrix A and a vector $mathbfv$ are represented as $\mathsf{A}mathbfv$.

Functions are represented with English or Greek letters in upper or lower case followed by parenthesis with their input parameters.

2.2 Projective Geometry

The Euclidean geometry is a usual way of representing points in an n-dimensional space. However, it has some drawbacks when modeling the behavior of a projective camera, where points in the real world are projected onto the camera's image plane. One of the major disadvantages is that the Euclidean space is not invariant to projective transformations. In these situations, the projective geometry has the advantage of not having this issue and possesses some benefits such as the representation of points in homogeneous coordinates.

2.2.1 Homogeneous Coordinates

In homogeneous coordinates, both points and lines are represented with vectors. A point

$$\mathbf{x_{euclidean}} = [x_1, x_2, ..., x_n]^\mathsf{T} \tag{2.1}$$

in the Euclidean space $\mathbb{R}^n$ is represented as

$$\mathbf{x_{projective}} = [kx_1, kx_2, ..., kx_n, k]^\mathsf{T} \tag{2.2}$$

in the homogeneous coordinates in the projective space $\mathbb{P}^n$, where $k \neq 0$ and $k \in \mathbb{R}$. Because of that, one point in the Euclidean space is represented by infinite points in the projective space with different values of k. In a similar way, a point in homogeneous coordinates is mapped back to Euclidean space by dividing its members by k. In a 2D space ($\mathbb{P}^2$ space),

$$\mathbf{x_{projective}} = [x, y, k]^\mathsf{T} \tag{2.3}$$

is mapped to

$$\mathbf{x_{euclidean}} = [{}^{x}/_{k},\ {}^{y}/_{k}]^\mathsf{T}. \tag{2.4}$$

In a $\mathbb{P}^2$ projective space, a line

$$ax + by + ck = 0 \tag{2.5}$$

that contains a point $\mathbf{x} = [x, y, k]^\mathsf{T}$ is represented by the vector $\mathbf{l} = [a, b, c]^\mathsf{T}$. From equation 2.5, we have the relation

$$\mathbf{l}^\mathsf{T}\mathbf{x} = 0. \tag{2.6}$$

From this relation, the intersection between two lines $\mathbf{l}$ and $\mathbf{l}'$ is $\mathbf{l} \times \mathbf{l}'$ and the line that contains two points $\mathbf{x}$ and $\mathbf{x}'$ is $\mathbf{x} \times \mathbf{x}'$.

The projective geometry can also represent points at infinity using homogeneous coordinates. A point $\mathbf{x} = [x, y, k]^\mathsf{T}$ in the projective space is considered to be at infinity when $k = 0$. With equation 2.6, one can note that a line $\mathbf{l} = [0, 0, c]^\mathsf{T}$ contains all points at infinity. Each of

these lines is called line at infinity.

2.2.2 Projective Transformations

Projective transformations or homographies represent perspective views, *i.e.* the effect of viewing a scene from one single point. They can translate, reflect, scale, rotate and shear an image. Homographies are linear transformations in the projective space, but they are non-linear in the Euclidean space, and this happens because of the division by k when mapping from $\mathbb{P}^n$ to $\mathbb{R}^n$, which is a rational operation. Lines are transformed to lines with a projective transformation, but parallelism relation is not necessarily held.

In $\mathbb{P}^2$, a homography H is a 3×3 matrix that transforms a point $\mathbf{x}$ to $\mathbf{x}'$, up to scale. It is applied to a point by doing

$$\mathbf{x}' = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \mathbf{x} = \mathsf{H}\mathbf{x}, \tag{2.7}$$

where $\mathbf{x}$ and $\mathbf{x}'$ are in homogeneous coordinates.

Affine transformations or affinities are a subcategory of projective transformations. H_A is an affinity if $h_{31} = h_{32} = 0$ and $h_{33} = 1$, as in

$$\mathsf{H}_A = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.8}$$

Unlike homographies, affine transformations keep the parallelism between lines and the ratio between lengths and areas.

2.3 Pinhole Camera Model

A camera model describes the relation between 3D points in the scene and their respective 2D points in the camera's image. The pinhole model is widely used and is a simple projection of 3D points in the image plane as in

$$\mathbf{x} = \mathsf{K}\mathsf{R}[\mathsf{I} \mid -\mathbf{C}]\mathbf{X} = \mathscr{P}\mathbf{X}, \tag{2.9}$$

where $\mathbf{X}$ and $\mathbf{x}$ are the scene point and the projected point in homogeneous coordinates, I is the 3×3 identity and $\mathscr{P}$ is the camera's projection matrix, which encapsulates all information to project a scene point onto the camera's image.

R and $\mathbf{C}$ are the camera's extrinsic parameters, which are a rotation and a translation that define its orientation and position in the scene. The matrix $P = \mathsf{R}[\mathsf{I} \mid -\mathbf{C}] = [\mathsf{R} \mid \mathbf{t}]$ is also known as the camera's pose, where $\mathbf{C}$ represents the center of the camera in the scene's coordinate

system and $\mathbf{t} = -\mathsf{R}\mathbf{C}$ is the camera's center in its own coordinate system. When using two or more cameras, one camera is usually considered to have $\mathsf{R} = \mathsf{I}$ and $\mathbf{C} = [0,0,0]^\mathsf{T}$ (origin of the scene's coordinate system) and the other cameras have their positions defined by the relative pose between them and the origin camera.

The matrix K encapsulates the camera's intrinsic parameters as in

$$\mathsf{K} = \begin{bmatrix} f_x & 0 & x_{pp} \\ 0 & f_y & y_{pp} \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.10}$$

where f_x and f_y are the camera's focus in the x and y axis, and the point $[x_{pp}, y_{pp}]^\mathsf{T}$ is its principal point.

The focus of a camera is the distance between the camera's center and the projection plane. In most cases, the camera's pixels are not considered to be squares, then $f_x \neq f_y$. The principal point is the intersection between the optical axis (last row of R) and the image plane.

2.4 Epipolar Geometry

The epipolar geometry describes the relation between two views from the same scene. It depends on the cameras' intrinsic and extrinsic parameters. This relation is represented by the fundamental matrix F, which is a 3×3 matrix with rank 2.

Let $\mathbf{X}$ be a 3D point in the scene and let $\mathbf{x}$ and $\mathbf{x}'$ be its image in homogeneous coordinates in two different views, then the relation between $\mathbf{x}$ and $\mathbf{x}'$ is

$$\mathbf{x}'^\mathsf{T} \mathsf{F} \mathbf{x} = 0. \tag{2.11}$$

From equations 2.6 and 2.11, one can notice that F maps points in one view to lines in the second view. These lines are called epipolar lines and they all meet at the epipolar point or epipole, which is the image point of the other camera's center, as illustrated in Figure 2.1.

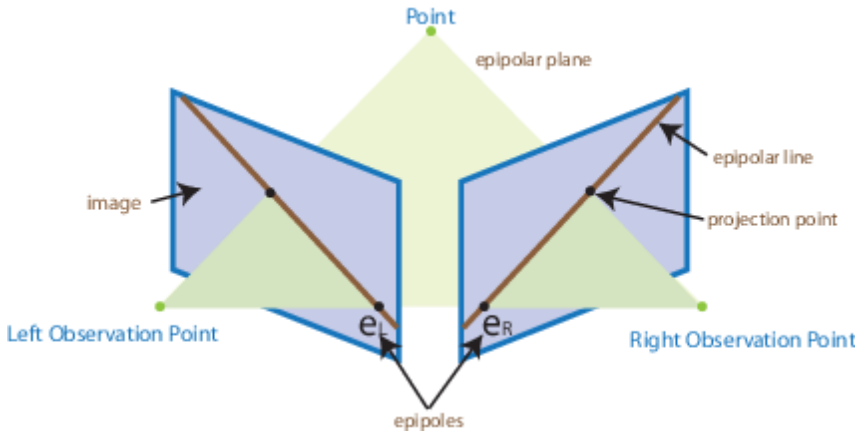


Figure 2.1: Epipolar plane, line and point in a stereo 3D reconstruction scene. (Source: https://www.mathworks.com/help/vision/ref/epipolar_inimage.png)

2.5 Stereo Rectification

Stereo rectification is widely used in stereo computer vision applications that require matching features from different frames or cameras. Two images are rectified if all matching points $\overline{\mathbf{x}} = [\overline{x}, \overline{y}, 1]$ and $\overline{\mathbf{x}}' = [\overline{x}', \overline{y}', 1]$ have the same y coordinate, *i.e.* $\overline{y} = \overline{y}'$ (HARTLEY; ZISSERMAN, 2004). The stereo rectification process consists in finding two projective transformations H and H' applied in two sets of points $\mathbf{x}$ and $\mathbf{x}'$ in homogeneous coordinates from a stereo rig that

$$\begin{aligned} \overline{\mathbf{x}} &= \mathsf{H}\mathbf{x} \\ \overline{\mathbf{x}}' &= \mathsf{H}'\mathbf{x}'. \end{aligned} \tag{2.12}$$

Once H and H' are applied, the fundamental matrix between the images must become

$$\mathsf{F} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}, \tag{2.13}$$

where $\overline{\mathbf{x}}^{\mathsf{T}} \mathsf{F} \overline{\mathbf{x}}' = \overline{y} - \overline{y}' = 0$.

Although H and H' are projective transformations, they may become affine transformations or similarities using constraints from the input sets, which happens in the scenario used in this dissertation.

2.6 Triangulation

Since matched points do not usually satisfy the epipolar constraint because of the image noise, there are various methods for minimizing the reconstruction error. In this dissertation, we use the linear triangulation method described in HARTLEY; ZISSERMAN (2004).

For a pair of matching points in homogeneous coordinates $\mathbf{x} = [x, y, 1]^{\mathsf{T}}$ and $\mathbf{x}' = [x', y', 1]^{\mathsf{T}}$ from cameras with projection matrices $\mathscr{P}$ and $\mathscr{P}'$, we seek to find a 3D point $\mathbf{X}$ that

$$\begin{aligned} \mathbf{x} &= \mathscr{P}\mathbf{X} \\ \mathbf{x}' &= \mathscr{P}'\mathbf{X}. \end{aligned} \tag{2.14}$$

These equations don't have an exact solution in most computer vision applications. Thus, we estimate a 3D point $\check{\mathbf{X}}$ that exactly satisfies the equations in 2.14 and minimizes the distance between its left and right image points $\check{\mathbf{x}}$ and $\check{\mathbf{x}}'$, and the detected points $\mathbf{x}$ and $\mathbf{x}'$.

One approach to this problem is to use a Direct Linear Transformation (DLT) to solve a redundant set of equations in the form $\mathsf{A}\mathbf{X} = \mathbf{0}$, where A is a four-dimensional square matrix, $\mathbf{X}$ is in homogeneous coordinates, and $\mathbf{0}$ is a 4D column vector filled with zeros. The matrix A is computed from the equations in 2.14.

To calculate A, we start by removing the homogeneous scale factor from $\mathbf{x}$ and $\mathbf{x}'$ using the algebraic property $\mathbf{a} \times \mathbf{a} = 0$. To satisfy equations 2.14, we should have

$$\begin{aligned} \mathbf{x} \times \mathscr{P}\mathbf{X} &= 0 \\ \mathbf{x}' \times \mathscr{P}'\mathbf{X} &= 0. \end{aligned} \tag{2.15}$$

Equation 2.15 gives three equations for each image point as in

$$\begin{aligned} x(\mathbf{p}_3^\mathsf{T}\mathbf{X}) - (\mathbf{p}_1^\mathsf{T}\mathbf{X}) &= 0 \\ y(\mathbf{p}_3^\mathsf{T}\mathbf{X}) - (\mathbf{p}_2^\mathsf{T}\mathbf{X}) &= 0 \\ x(\mathbf{p}_2^\mathsf{T}\mathbf{X}) - y(\mathbf{p}_1^\mathsf{T}\mathbf{X}) &= 0, \end{aligned} \tag{2.16}$$

where $\mathbf{p}_i^\mathsf{T}$ is the ith row of $\mathscr{P}$. An analogous set of equations to the equations 2.16 can be obtained from $\mathbf{x}'$ and $\mathscr{P}'$, and both sets are linear in the components of $\mathbf{X}$.

From the two sets of equations we obtain A as

$$\mathsf{A} = \begin{bmatrix} x\mathbf{p}_3^\mathsf{T} - \mathbf{p}_1^\mathsf{T} \\ y\mathbf{p}_3^\mathsf{T} - \mathbf{p}_2^\mathsf{T} \\ x'\mathbf{p}'^\mathsf{T}_3 - \mathbf{p}'^\mathsf{T}_1 \\ y'\mathbf{p}'^\mathsf{T}_3 - \mathbf{p}'^\mathsf{T}_2, \end{bmatrix} \tag{2.17}$$

where the two first equations from each set were used.

Once A was computed, we solve the problem $\mathsf{A}\mathbf{X} = \mathbf{0}$ using the DLT method. It calculates the solution as the singular vector that corresponds to the smallest singular value of o A. The DLT can also be used with variations such as using more than four equations or normalizing the data before solving the equation for $\mathbf{X}$.

2.7 RANSAC

The Random Sample Consensus (RANSAC) is an estimator proposed by FISCHLER; BOLLES (1981) that estimates the parameters of a model by sampling random sets of observed data and trying to fit them in the fitting model.

The RANSAC algorithm is composed of two steps. The first step selects a random sample and instantiates a model with it. Those usually contain minimal information to represent the fitting model. Thus, for each sample, the model's parameters are computed using only elements of this sample. In the second step, the algorithm counts the number of items from the entire dataset that fit in the instantiated model within a predefined error threshold, which are known as inliers. The features that exceed this error threshold are called outliers and are removed from the dataset once the algorithm is finished.

After a number N of iterations, the algorithm chooses the fitting model that has more inliers. The number of iterations N is determined by

$$N = \frac{\log(1-p)}{\log(1-(1-e)^s)}, \tag{2.18}$$

where e is the percentage of outliers in the dataset, s is the sample's size and p is the desired probability that at least one sample only contains inliers.

2.8 Discussion

This chapter presented the main theoretical foundations necessary for contextualizing and understanding the literature review and our proposal, which are presented in the next chapters. In Chapter 3, we use the concepts shown in this chapter to review some related work in point matching and relative pose estimation.

3

Related Work

In this chapter, we review related research on uncalibrated 3D reconstruction, by presenting relevant work on both point matching and relative pose estimation, which are two important topics in the field. In Section 3.1 we discuss relevant work of point matching in computer vision. We direct our attention to studies that aim to find correspondences between points in at least a pair of images, without previous knowledge of the extrinsic calibration of the camera rig. In Section 3.2 we discuss research on relative pose estimation, focusing on stereo rigs using sets of previously matched points.

3.1 Matching

There are several methods for finding correspondences between images and, according to ZITOVA; FLUSSER (2003), they are divided into two groups. The first group is the Area-based methods, which deal directly with an image pair and estimate the matches by measuring the correlation among windows within the images. The second one is the Feature-based methods, which depend on detecting distinct characteristics in a picture.

The area-based methods, also known as the correlation-like methods, estimate the correspondences without detecting salient objects within the image. They explore ways of measuring similarities between the images or windows of predefined size within the pictures. This is usually achieved by using the following approaches: the Fast Fourier Transform (BERTHILSSON, 1998), Hausdorff distance (HUTTENLOCHER; KLANDERMAN; RUCKLIDGE, 1993), and the sum of absolute differences (MÜHLMANN et al., 2002). Area-based methods are not adequate for point matching due to two reasons. First, part of those estimates a correlation between the images, and do not return any information on point matching. The other part makes use of disparity maps that are suitable for point matching, but lack in a robust mapping under noise. Besides, such methods are highly sensitive to illumination and contrast variations, which are typical characteristics in our scenario.

The feature-based methods estimate correspondences using either an object or a characteristic of the image. Those objects, or features, can be regions (WANG; ZHENG, 2008; MATAS et al., 2004; WEI; QUAN, 2004), edges (BAY; FERRARI; VAN GOOL, 2005; MEDIONI;

NEVATIA, 1985), points (ABDEL-HAKIM; FARAG, 2006; LOWE, 1999; BAY et al., 2008), or the spatial relations between points (BELONGIE; MALIK; PUZICHA, 2000). The common pipeline for these methods starts by detecting and selecting the features. SHI et al. (1994) proposed a criterion for choosing which features to use. Once the features are obtained, they must be described somehow to allow a comparison between them. The method for describing the features varies with de matching algorithm, but the description is normally stored in a container called descriptor. Descriptors contain the relevant information to identify features in the image and compare it with each other. With the features detected and described, the matches are estimated calculating the difference between the descriptors. The smaller the difference is, the more likely it is that a pair of descriptors in different images represent the same feature. One may not want to compare all possible pairs of descriptors since there may be several and this might slow the algorithm. Thus, there are several approaches to choose which descriptors to compare, such as *brute force*, *k-d trees* (BENTLEY, 1975), *best bin first* (BEIS; LOWE, 1997), and *Fast Approximate Nearest Neighbor Search (FLANN)* (MUJA; LOWE, 2009).

MATAS et al. (2004) explore image regions for estimating matches. Their algorithm searches *extremal regions*, which are areas in the image that can be detected with high repeatability and are assumed to possess some invariant and stable properties. One property is that these regions are closed under projective transformations of the images coordinates. In our scenario, the points from the pipe are approximately coplanar. Therefore, the author's results are not reliable to our context since transformations of coplanar points are affine.

The methods using edges as features consist of finding contours, shadows or textures on objects from the scene (BAY; FERRARI; VAN GOOL, 2005; MEDIONI; NEVATIA, 1985). In our research scenario, the objects are deep underwater, so the images lack in contrast, and there are little shadows and textures. Hence, such methods are not adequate to the study conditions.

Furthermore, there are methods using points on the image to find the correspondences. Usually, such methods make use of *keypoints*. These are distinguishing points in the picture that are recognizable after image transformations, like rotations, translations and illumination changes. Thus, each keypoint is represented in a descriptor, which has the required information to distinguish it in the image, such as intensities of pixels surrounding the keypoint (LOWE, 1999) or distance and relative position to other keypoints BELONGIE; MALIK; PUZICHA (2000). Finding images correspondences with keypoints can be interpreted as calculating the difference between all the descriptors and matching those of smallest difference. Techniques using points are suitable for our scenario allowing detection in deep underwater environments.

There are approaches in point-based methods that describe the keypoints with local information like the intensity of a set of pixels near the keypoint. A well-known method from this group is the Scale Invariant Feature Transform (SIFT) (ABDEL-HAKIM; FARAG, 2006), which uses a histogram of oriented gradients to describe the keypoints. There are variations of the SIFT method, such as: *CSIFT* (ABDEL-HAKIM; FARAG, 2006), that differs from the original SIFT by considering color information on the descriptor; *N-SIFT* (CHEUNG; HAMARNEH,

2007), which generalizes the SIFT to *N* dimensions; and the PCA-SIFT (KE; SUKTHANKAR, 2004), that uses Principal Component Analysis (PCA) on the descriptor in the matching process. Another common algorithm for describing keypoints is the Speeded-Up Robust Features (SURF) (BAY et al., 2008), which proposes to accelerate the detection and descprition procedure of SIFT by optimizing it and making it more invariant to image transformations. Although these methods apply to our scenario, they depend on features such as textures and high contrast regions, which are not present in deep-underwater environments.

Moreover, there are point-based methods that also make use of spatial relations between points as the descriptor for each point, which is the case of the Shape Context algorithm, proposed in BELONGIE; MALIK; PUZICHA (2000). The Shape Context algorithm describes each keypoint by the relative position and the distance from the keypoint to all the others. Therefore, it does not use any other information from the images, making it adequate for scenarios with low contrast and little texture information. This method, however, is not robust to noise, which is typical condition in our scenario.

3.2 Pose Estimation

Most of the stereo applications for 3D reconstruction in computer vision require knowledge of the relative pose between cameras. When the scene is known, finding the cameras' pose is a Perspective-n-View (PnP) problem, otherwise, the pose between cameras must be obtained using a two view approach. The relative pose is a transformation that relates the position and orientation of both cameras in a stereo rig. This transformation can be decomposed as a rotation and a translation from one camera to the other.

In general, the methods for solving this problem can be divided in two groups. The first uses no previous knowledge of the cameras' intrinsics. In this case, the pose can be obtained from the fundamental matrix (ZHANG, 1998; FITZGIBBON; ZISSERMAN, 1998; KUKELOVA et al., 2015). LONGUET-HIGGINS (1981) proposed the eight-point algorithm, a method for estimating the fundamental matrix from eight point matches in the images. HARTLEY (1997) improved it with the normalized eight-point algorithm and compared it with other methods.

The second group constrains the solution domain by using the intrinsics. This constraint makes it suitable to use the essential matrix instead of the fundamental matrix, since it takes the intrinsics into account and, therefore, has fewer degrees of freedom. Although the eight-point can also compute the essential, other methods do it needing fewer point matches. FAUGERAS; MAYBANK (1990) proposed one of the first methods for finding the relative pose with the essential matrix. This method estimated a camera's movement based on two different frames, which is a similar problem. The stereo rig's intrinsics is known in our scenario, so the essential matrix approach is better suited than the fundamental matrix one.

For previously calibrated cameras, five point matches are required to retrieve the essential matrix when no rotation and translation restrictions are considered. Thus, this is called the

five-point problem. NISTÉR (2004) proposed one of the commonly used methods for solving the five-point problem. It is a fast method and can be used within a RANSAC loop (FISCHLER; BOLLES, 1981) to improve accuracy and reduce error, while still maintaining a real time performance (NISTÉR, 2005). TORR; ZISSERMAN (2000) presented the MLESAC, another estimator suitable for this method, maximizing the likelyhood, instead of the number of inliers, to choose a final solution. Nistér's method and its variations are also widely used accompanied by bundle adjustment (TRIGGS et al., 1999).

STEWENIUS; ENGELS; NISTÉR (2006) presented a variation of Nistér's five-point algorithm, computing a Gröbner basis to obtain the relative pose. HEDBORG; FELSBERG (2013) proposed an iterative method based on Powell's Dog Leg algorithm and claimed it to be approximately twice as fast as Nistér's. BATRA; NABBE; HEBERT (2007) solved the five-point problem using nine quadratic equations with six variables instead of a tenth degree polinomial, as in Nister's algorithm, and obtained results more robust to noise. FATHIAN; GANS (2014) presented an alternative to the essential matrix by representing the pose problem in the quaternion space.

Some scenarios have other constraints aside from the intrinsics, which can be used to decrease the problem's degrees of freedom and, thus, the number of point matches needed for estimating the relative pose. FRAUNDORFER; TANSKANEN; POLLEFEYS (2010), KALANTARI et al. (2011) and SWEENEY; FLYNN; TURK (2014) proposed methods that required three points for computing the pose by having partial knowledge about the cameras orientation. OTSU; KUBOTA (2014) presented a method using two points by "exploiting a common reference direction between poses", as they mentioned. These methods are appropriate for our problem because there are restrictions to both the relative rotation and translation of the cameras.

(FREDRIKSSON et al., 2016) proposed an algorithm to estimate the relative pose with unkown point matches. Although this may seem a useful solution to our problem since it doesn't need a set of matches as input, this algorithm is not suitable for real-time application. MUSLEH et al. (2016) published a method for finding stereo pose in real time for automotive applications, where the pose varies while the vehicle is moving. Their dynamic approach for estimating relative pose is relevant for our work in situations that accidents change the pose. YANG; TANG; HE (2017) presented an accurate solution for scenarios with unknown coplanar points using at least four point matches. This work is of particular importance for our proposal because it deals with the similar constraint of having all the matched points in a same plane.

3.3 Discussion

This chapter covered relevant algorithms for matching and pose estimation. Even with several methods in the literature, our scenario has a series of constraints that are not dealt by a single algorithm. The repetitive pattern and the lack of illumination and textures limit the

number of reliable solutions for the matching stage. In the other hand, characteristics such as the coplanar constraint, the partial knowledge of the cameras' pose and the need to adapt to orientation changes are present in many works. However, all these variations are not present in a single work.

We propose a novel method using the constraints and variations cited above to have an optimal solution for our scenario. The next chapter describes our proposal for relative pose estimation in deep underwater environments.

4

Automatic Pose Estimation

In this chapter, we present our proposal for automatically estimating relative pose between cameras of a stereo rig in underwater environment. The objective of this proposal is to estimate the relative pose between cameras of a stereo rig, in real time applications, without using predetermined patterns. In order to attend these requirements, we calculate the rotation and translation of each camera in the scenario using detected points from the stereo rig's left and right images and some constraints from the scenario.

We opted to consider that the left camera has zero rotation and translation in the scene's coordinates, which means that the left camera pose is $\mathsf{R} = \mathsf{I}$ and $\mathbf{t} = \mathbf{0}$, where I is the three dimensional identity and $\mathsf{0}$ is a 3D vector filled with zeros. Because of that, the right camera pose is equal to the relative pose between the cameras.

As it was seen in the previous chapter, estimating the relative pose requires that the detected points in both images are matched. Our proposal focuses on scenarios where there are few information to match points using feature-based methods and, since none of the presented matching techniques had an acceptable performance in our scenario, we also propose a matching stage for this problem (FIGUEIREDO et al., 2016).

Therefore, the proposed method is divided into two separated stages, Matching and Pose Estimation. The Matching stage is executed repeatedly for a predefined number of input sets from different frames, which is described in Section 3.1. It starts with a RANSAC estimator to find the matches and uses them to perform stereo rectifications on the images. After that, the Pose Estimation stage, described in Section 3.2, estimates an initial pose using the output from the matching stage and optimize its result with a bundle adjustment (TRIGGS et al., 1999).

4.1 Matching Stage

In this first stage, our method receives two sets of detected points. In our scenario, each point is a vertebra from the flexible pipe detected by the algorithm proposed in PESSOA et al. (2015). We use a RANSAC estimator to find the rectification parameters that maximize the number of correspondences between points from one set to the ones from the other. Our first step in the RANSAC iteration is to use samples from the input to rectify the input set and enable

the matching process. The rectification process is described in Subsection 4.1.1. After the first step, we estimate the point matches using the y coordinate of the rectified points, as described in Subsection 4.1.2. Then the RANSAC evaluates each rectification by the number of inliers after matching the points and the absolute difference between a pair of points in a match.

4.1.1 Rectification

Since usual point matching algorithms require the relative pose of the stereo rig to find point correspondences, we opted to add a stereo rectification algorithm so that we could match image points. Our approach starts by sampling a pair of points in each camera image and considering that they form two matches. From that, we use the algorithm presented in CESAR et al. (2014) to estimate the rectifications parameters, which permits the later matching process.

As described in Section 2.5, the stereo rectification process consists in finding two projective transformations H and H' that equal the y coordinates of two points in a match. The rectified points $\overline{\mathbf{x}}$ and $\overline{\mathbf{x}}'$ are then obtained by applying these transformations in matching points $\mathbf{x} \leftrightarrow \mathbf{x}'$ from the left and right images, as shown in equation 2.12. Since the cameras in our scenario are considered to have projection planes approximately coplanar and similar projection parameters, the projective transformations are equivalent to a rotation for each point set, $R(\alpha)$ and $R(\beta)$, where $R(\theta)$ is a 2D clockwise rotation of angle θ (CESAR et al., 2014). Thus, in our case, the rectification matrices are

$$\begin{aligned} \mathsf{H} = R(\alpha) &= \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \mathsf{H}' = R(\beta) &= \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{4.1}$$

Using equation 4.1 in equation 2.12, the rectified points are

$$\begin{aligned} \overline{\mathbf{x}} = \begin{bmatrix} \overline{x} \\ \overline{y} \end{bmatrix} &= \mathsf{H}\mathbf{x} = \begin{bmatrix} x\cos\alpha - y\sin\alpha \\ x\sin\alpha + y\cos\alpha \end{bmatrix} \\ \overline{\mathbf{x}}' = \begin{bmatrix} \overline{x}' \\ \overline{y}' \end{bmatrix} &= \mathsf{H}'\mathbf{x}' = \begin{bmatrix} x'\cos\beta - y'\sin\beta \\ x'\sin\beta + y'\cos\beta \end{bmatrix}. \end{aligned} \tag{4.2}$$

Since the rectified point matches $\overline{\mathbf{x}} \leftrightarrow \overline{\mathbf{x}}'$ have $\overline{y} = \overline{y}'$, we can equal the y coordinates of $\mathsf{H}\mathbf{x}$ and $\mathsf{H}'\mathbf{x}'$, and obtain

$$x\sin\alpha + y\cos\alpha - x'\sin\beta - y'\cos\beta = 0. \tag{4.3}$$

Generalizing this equation to a set of n matches, we have the system

$$\begin{bmatrix} x_1 & y_1 & -x'_1 & -y'_1 \\ x_2 & y_2 & -x'_2 & -y'_2 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & -x'_n & -y'_n \end{bmatrix} \begin{bmatrix} \sin\alpha \\ \cos\alpha \\ \sin\beta \\ \cos\beta \end{bmatrix} = \mathsf{A}\dot{\mathbf{y}} = \mathbf{0} \tag{4.4}$$

where $\mathbf{0}$ is a column vector of zeros with n rows and the subscript in x_i and x'_i represents the ith element of the set of points from the left and right images, respectively.

The system in equation 4.4 has 2 degrees of freedom which are the two rectification angles α and β. The sine and cosine functions cause the solution to be non-linear and, therefore, a linearizion is recommended for a direct solution. A possible linearizion consists of replacing the non-linear vector $\dot{\mathbf{y}} = [\sin\alpha, \cos\alpha, \sin\beta, \cos\beta]^\mathsf{T}$ by a linear one $\mathbf{y} = [y_1, y_2, y_3, y_4]^\mathsf{T}$ with the constraints

$$y_1^2 + y_2^2 = 1 \tag{4.5}$$

$$y_3^2 + y_4^2 = 1. \tag{4.6}$$

For solving the new linear equation, one can use a Singular Value Decomposition (SVD) decomposition $\mathsf{A} = \mathsf{UDV}^\mathsf{T}$. With this approach, an approximate solution is obtained from the third and fourth columns of V, namely $\mathbf{u} = [u_1, u_2, u_3, u_4]^\mathsf{T}$ and $\mathbf{v} = [v_1, v_2, v_3, v_4]^\mathsf{T}$, which correspond to the smallest singular values. Since the problem has two variables, the solution must be in a two-dimensional space and must be a linear combination of $\mathbf{u}$ and $\mathbf{v}$ as in

$$\mathbf{y} = \gamma\mathbf{u} + \delta\mathbf{v}. \tag{4.7}$$

Replacing the equation 4.7 in 4.5 and 4.6, leads to

$$(\gamma u_1 + \delta v_1)^2 + (\gamma u_2 + \delta v_2)^2 = 1 \tag{4.8}$$

$$(\gamma u_3 + \delta v_3)^2 + (\gamma u_4 + \delta v_4)^2 = 1, \tag{4.9}$$

and summing up these resultant equations 4.8 and 4.9, we have

$$\mathbf{u}^\mathsf{T}\mathbf{u}\gamma^2 + \delta\mathbf{u}^\mathsf{T}\mathbf{v}\gamma\delta + \mathbf{v}^\mathsf{T}\mathbf{v}\delta^2 = \gamma^2 + \delta^2 = 2, \tag{4.10}$$

since $\mathbf{u}$ and $\mathbf{v}$ are from an orthonormal basis.

The equations 4.8 and 4.9 represent two ellipses, and the equation 4.10 describes a circumference. The possible solutions to γ and δ, and, therefore, to α and β are the intersections of the circumference with the two ellipses, which leads to two or four values. If there are four solutions, the one with $\gamma > \delta$ must be used (CESAR et al., 2014). Finally, we apply the transformations $\mathsf{H} = R(\alpha)$ and $\mathsf{H}' = R(\beta)$ to all the left and right input points from the sets, $\mathbf{x_i}$

and $\mathbf{x}'_\mathbf{i}$, which gives us the rectified sets points $\overline{\mathbf{x}_\mathbf{i}}$ and $\overline{\mathbf{x}_\mathbf{i}}'$, according to equation 2.12.

4.1.2 Matcher

Once all points are rectified based on the sample from the rectification stage, we use a matching algorithm to validate our initial guess of matches. This algorithm estimates the matches based on the y coordinate, since rectified matches $\overline{\mathbf{x}} = [\overline{x}, \overline{y}]^\mathsf{T} \leftrightarrow \overline{\mathbf{x}}' = [\overline{x}', \overline{y}']^\mathsf{T}$ should have $\overline{y} = \overline{y}'$. The matching algorithm used in our proposal is described below.

Our matching algorithm consists of three main steps: *Initialization*, where an initial set of matches is obtained based on the y coordinate, *Transformation Estimation*, which estimates an affine transformation with the correspondences from the first step, and *Final Estimation*, where a new set of matches is obtained using the transformation from the last step. This algorithm estimates the correspondences based uniquely on the points' coordinates and on the cameras' intrinsics, which makes it suitable for noisy scenarios or when the detected points are difficult to be distinguished from each other.

The Initialization step uses the y coordinates of the points to acquire an initial set of matches. Since corresponding points have similar y values, the similarity metric for considering two points $\mathbf{x} = [x, y]^\mathsf{T}$ and $\overline{\mathbf{x}}' = [\overline{x}', \overline{y}']^\mathsf{T}$ a match is the disparity $\delta_y = d(\overline{y}, \overline{y}')$, where $d(a, b)$ is the Euclidean distance between a and b. The matching is done with a sequence alignment using a Dynamic Programming (DP) approach. The replacing cost is the disparity δ_y between a pair of points and the cost to create a gap in the matching sequence is the maximum disparity allowed τ, which was defined empirically as 2 pixels. This first step has an ambiguous output, since more than one point might have similar y coordinates, as shown in Figure 4.1. The following steps help mitigate this problem.

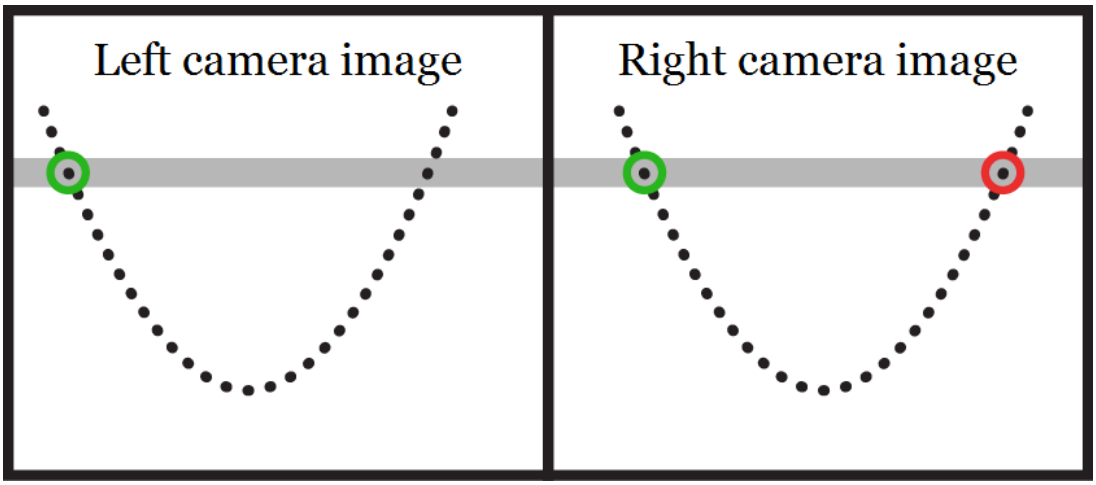


Figure 4.1: Ambiguous situation in the first step of the algorithm caused by only using the y coordinate for matching points. Here, the green point in the left image could be mistakenly matched with any other point in the gray area, such as the red one in the right image. (Adapted from FIGUEIREDO et al. (2016))

The Transformation Estimation step aims to obtain a transformation H that minimizes

the Euclidean distance between all pairs of corresponding points. The first constraint of H is that it must be a homography because the original 3D points from the pipe are approximately coplanar (HARTLEY; ZISSERMAN, 2004). Besides, with the input set already rectified, the transformation does not change the y value of the points, but only the x coordinate. Therefore, H must be an affinity such as

$$\mathsf{H} = \begin{bmatrix} a & b & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.11}$$

where a, b and $c \in \mathbb{R}$.

Another RANSAC estimator is used in this step to filter wrong matches from the Initialization step when calculating H. In each iteration, a transformation is computed using a sample of three points. Then the RANSAC calculates the distance

$$\delta = d(\mathsf{H}\overline{\mathbf{x}} - \overline{\mathbf{x}}') \tag{4.12}$$

for each pair $\overline{\mathbf{x}} \leftrightarrow \overline{\mathbf{x}}'$ from the first step, where d is the euclidean distance function. The hypothesis is evaluated by the number of points that have

$$\delta < \tau, \tag{4.13}$$

where τ is the disparity threshold defined in the first step. The number of iterations N for this RANSAC estimator was defined by the equation 2.18, where the expected percentage of outliers e is approximately 30%, the sample's size is 3, and the chosen probability that at least one sample only contains inliers is 99%. Therefore, the number of iterations is $N = 15$.

The Final Estimation step of the matching algorithm repeats the Initialization, but the points from one set are now transformed by H so that the x coordinates of corresponding points obey the equation 4.13. Another DP table is used here to re-estimate the point matches. This sequence alignment only differs from the first one by the similarity metric, which is now the Euclidean distance δ from equation 4.12 instead of δ_y. The affine transformation allows us to use the points' x coordinate to solve the ambiguous situations from the first step, as depicted in Figure 4.2. At the end of this step, a new set of correspondences is obtained maximizing the number of inliers for the current rectification parameters.

The number of inliers evaluates the rectification parameters hypothesis after matching the points using the same threshold τ. In the case of more than one hypothesis with the same number of inlier matches, we choose the one that minimizes the absolute difference between corresponding points $\overline{\mathbf{x}} \leftrightarrow \overline{\mathbf{x}}'$.

After a hypothesis is chosen, the original points from the current frame, $\mathbf{x}$ and $\mathbf{x}'$, and the respective set of matches are stored for later pose estimation. With data from a predefined number of frames, our algorithm starts the Pose Estimation Stage, described in the following

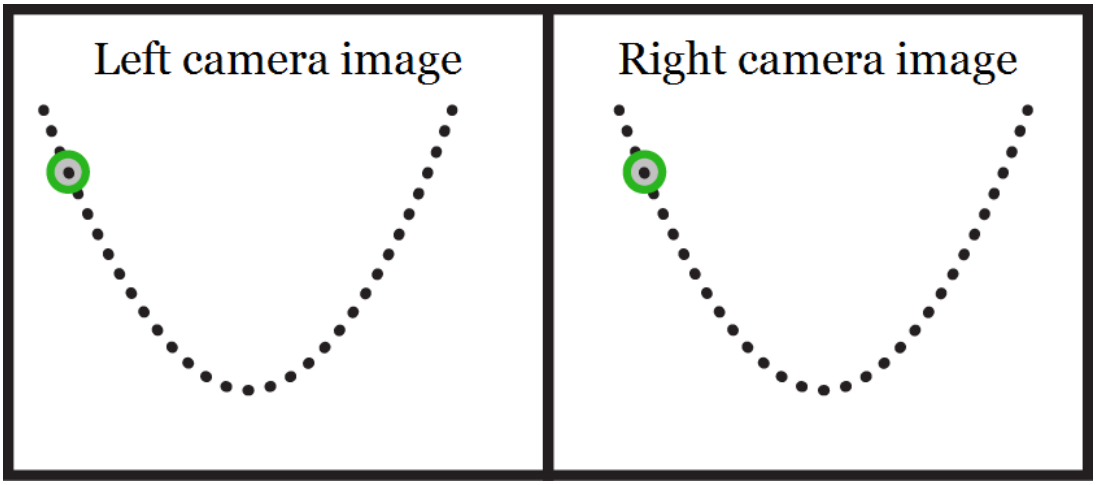


Figure 4.2: Comparing points with the euclidean distance instead of just the y coordinate reduces the ambiguous region (gray) to a circle of radius τ. (Adapted from FIGUEIREDO et al. (2016))

section.

4.2 Pose Estimation Stage

The Pose Estimation stage of our solution uses the cameras' intrinsics and the data collected in the previous stage to estimate the relative orientation R and position $\mathbf{t}$ of the stereo rig. We also use as input for this stage the stereo rig's approximate baseline, which is the distance between the center of both cameras. This stage starts by doing an initial estimation of the relative pose, which is described in Subsection 4.2.1. We use this initial estimation to triangulate the matching points and obtain reconstructed 3D points. Finally, a bundle adjustment problem is mounted to minimize the reprojection error between the image detected and predicted points. This optimization problem is described in Subsection 4.2.2.

4.2.1 Initial Estimation

For our initial estimation, we use Nistér's five-point algorithm (NISTÉR, 2004) to obtain the essential matrix E of our stereo rig, which solves the equation

$$[\mathbf{x};\ 1]^{\mathsf{T}}\mathsf{K}^{\mathsf{T}}\mathsf{E}\mathsf{K}'[\mathbf{x}';\ 1] = 0, \tag{4.14}$$

where $\mathbf{x}$ and $\mathbf{x}'$ are corresponding points in the left and right images, K and K' are their cameras' respective intrinsics. We used an implementation of Nistér's algorithm that considers both intrinsics to be

$$\mathsf{K} = \mathsf{K}' = \begin{bmatrix} f_x & 0 & x_{pp} \\ 0 & f_y & y_{pp} \\ 0 & 0 & 1 \end{bmatrix} \tag{4.15}$$

with $f_x = f_y$ being the cameras' focal lengths, and $[x_{pp};\ y_{pp}]^\mathsf{T}$ their principal point. In our scenario, neither the cameras' intrinsics are equal, nor the focal lengths' values f_x and f_y are the same. As in NISTÉR (2004), we normalize both point sets with the inverse of their respective intrinsics to obtain new sets of points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ with the equations

$$\begin{aligned} \hat{\mathbf{x}} &= \mathsf{K}^{-1}\mathbf{x} \\ \hat{\mathbf{x}}' &= \mathsf{K}'^{-1}\mathbf{x}'. \end{aligned} \tag{4.16}$$

With the normalized points, we can solve equation 4.14 using $\mathsf{K} = \mathsf{K}'$ with $f_x = f_y = 1$ and $x_{pp} = y_{pp} = 0$, which is the three-dimensional identity matrix.

Once the essential matrix E is recovered, the relative pose $\mathsf{P} = [\mathsf{R} \mid \mathbf{t}]$ is obtained by doing a singular value decomposition of E. We can write the SVD of E as

$$\mathsf{E} \sim \mathsf{U}\,diag(1,1,0)\mathsf{V}^\mathsf{T}, \tag{4.17}$$

where U and V are orthonormal matrices with determinant greater than zero, $diag(1,1,0)$ is a 3D diagonal matrix of rank 2 and $\sim$ represents a similarity up to scale. From that decomposition, let $\mathbf{t_u}$ be the last column of U, $\mathsf{R_a} = \mathsf{UDV}^\mathsf{T}$ and $\mathsf{R_b} = \mathsf{UD}^\mathsf{T}\mathsf{V}^\mathsf{T}$, where

$$\mathsf{D} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.18}$$

There are up to four solutions to the relative pose problem, which are $\mathsf{P}_1 = [\mathsf{R_a} \mid \mathbf{t_u}]$, $\mathsf{P}_2 = [\mathsf{R_a} \mid -\mathbf{t_u}]$, $\mathsf{P}_3 = [\mathsf{R_b} \mid \mathbf{t_u}]$ and $\mathsf{P}_4 = [\mathsf{R_b} \mid -\mathbf{t_u}]$ (HARTLEY; ZISSERMAN, 2004). One of those results is the stereo rig's correct configuration, and another is a twisted pair where one of the cameras is rotated 180 degrees, and the other two are their reflections. To choose one of the solutions, one has to triangulate one of the points correspondence with each pose. The configuration where the triangulated point has positive depth is considered the stereo rig's true configuration.

The translation $\mathbf{t_u}$ has unit length, thus it only represents the direction of the translation and not its true scale. This is solved by using the stereo rig's approximate baseline to scale $\mathbf{t_u}$. Once the pose estimation is up to scale, we triangulate every match from the matching stage with the linear triangulation method presented in Section 2.6. The triangulated points are used in the pose optimization process described in the next section.

4.2.2 Pose Optimization

After obtaining the initial pose estimation, we create a bundle adjustment problem using all matches from the previous stage, the estimated pose, the reconstructed 3D points, the cameras' intrinsics and the stereo rig's baseline. Now let $\mathscr{P}$ be the projection matrix of the left camera and $\mathscr{P}'$ the one of the right camera, then we have

$$\begin{aligned} \mathscr{P} &= \mathsf{K}[\mathsf{I} \mid \mathbf{0}] \\ \mathscr{P}' &= \mathsf{K}'[\mathsf{R} \mid \mathbf{t}], \end{aligned} \tag{4.19}$$

where K and K' are the left and right cameras' intrinsics, I is the 3D identity matrix, $\mathbf{0}$ is a three-dimensional column vector filled with zeros and $\mathsf{P} = [\mathsf{R} \mid \mathbf{t}]$ is the previously estimated pose. The problem we aim to solve here is

$$\underset{\mathsf{R},\mathbf{t},\mathbf{X}}{arg\,min} \sum_{i=0}^{n} \sum_{j=0}^{m} (||\mathbf{x}_{i,j} - \mathscr{P}\mathbf{X}_{i,j}|| + ||\mathbf{x'}_{i,j} - \mathscr{P}'\mathbf{X}_{i,j}||), \tag{4.20}$$

where n is the predefined number of frames used in the matching stage, m is the number of points matched in a frame i, $\mathbf{x}_{i,j}$ and $\mathbf{x}'_{i,j}$ are the jth matched point of the ith frame and $\mathbf{X}_{i,j}$ is the 3D point reconstructed with $\mathbf{x}_{i,j}$ and $\mathbf{x}'_{i,j}$. We use the Levenberg-Marquardt optimization algorithm (MARQUARDT, 1963) to solve this problem since it is a frequently used method for solving non-linear least squares problems. Once the rotation R and translation $\mathbf{t}$ values are optimized, our solution outputs the pose $\mathsf{P} = [\mathsf{R} \mid \mathbf{t}]$ and the pose estimation algorithm is finished.

4.3 Discussion

This chapter described our proposed solution for point matching and pose estimation. In the next chapter we present ou methodology of experimentation and validate our proposal by executing some tests in a controlled laboratory environment.

5

Experiments

This chapter presents the experiments for the proposed solution and has the objective of evaluating it in a controlled real-world scenario. Here, we describe the methodology of experimentation and present the results from the comparison of our solution with another calibration algorithm.

Section 5.1 describes the hardware and software setup that were used for testing the algorithm. In Section 5.2 we detail the methodology used to validate our proposal. Section 5.3 shows the results of the comparison between our solution and the current calibration algorithm in our application. In Section 5.4, we discuss the results shown in the previous section.

5.1 Implementation

The solution was implemented in C++ and compiled with Microsoft Visual C++. We used the OpenCV library (BRADSKI, 2000) for basic image processing and computer vision operations, and Eigen library (GUENNEBAUD et al., 2010) for matrices and linear algebra routines. The algorithm was implemented single threaded and we did not focus on code optimizations. The hardware used for testing is a desktop with processor Intel i7-5820K 3.30GHz and 16 GB of RAM.

The stereo rig used to record the test videos consists of two monochromatic cameras Kongsberg OE15-101c with an 82 degree FOV when submerged and separated by approximately 80 centimeters.

5.2 Methodology

Although the proposed solution can match and estimate the relative pose with any planar set of points, we decided to validate it by simulating the real-world scenario's conditions. During the offshore operations, the curve formed by the duct's points is usually similar to a catenary, which is a simple curve to model and to recreate in a laboratory.

The parametric equations for a catenary curve ς in the plane $z = 0$ and centered in the y axis are

$$\varsigma(s) = \begin{bmatrix} a\,\text{arcsinh}(s/a) \\ \sqrt{a^2+s^2} \\ 0 \end{bmatrix}, \tag{5.1}$$

where s is the length in an arbitrary section of the catenary and a is a scale parameter that is equal to the catenary's minimum radius of curvature. In a catenary of total length ℓ suspended by two points $\mathbf{p_1} = [x_1, y_1]^\mathsf{T}$ and $\mathbf{p_2} = [x_2, y_2]^\mathsf{T}$, the parameter a is obtained by solving the equation

$$\sqrt{\ell^2 + v^2} = 2a\sinh(h/2a), \tag{5.2}$$

where $h = |x_1 - x_2|$ is the horizontal disparity between $\mathbf{p_1}$ and $\mathbf{p_2}$, and $v = |y_1 - y_2|$ is the vertical disparity.

Thus, to test our proposal, we recorded different videos of an ordinary catenary in a laboratory and calibrated the pose parameters with each of them. Additionally, we also compared our solution to the calibration algorithm by pattern that is performed before submerging the cameras (SANTOS et al., 2015). This pattern calibration algorithm represents our goal for pose accuracy. We used the estimated poses to reconstruct the points from a validation video and compared their accuracies.

The catenary was replicated using a rope with $\ell = 4.470 \pm 1e$-3 meters of length suspended by two points on a planar wall. The disparity parameters were measured as $h = 2.347 \pm 2e$-3m and $v = 0.035 \pm 1e$-3m. The ground truth obtained with equation 5.1 for the minimum radius of curvature was $0.562 \pm 9e$-4 meters. The ducts vertebrae were simulated using black squares placed at regular intervals of 15 centimeters over the rope, as in Figure 5.1.

The calibration algorithm by pattern estimates both the cameras' intrinsics and the relative pose. It computes the intrinsic and extrinsic parameters by detecting a double pattern as shown in Figure 5.2. Since we only aim to compare the extrinsic parameters, we used the intrinsics from the pattern calibration as input to our solution so that the only difference in the 3D reconstruction is the input poses.

The next subsections describe the test cases and the evaluation methods that we used for our proposal.

5.2.1 Test Cases

Since the ROV has limited movement capabilities, the tests also aimed to verify if different camera movements impact in our proposal's results. Thus, we used four test cases with three types of camera translations: *heaving*, which is the linear vertical movement (up and down); *surging*, in which the camera does a linear longitudinal movement (front and back); *swaying*, which the camera performs a linear lateral movement (left and right); and *mixed movements*, which is a combination of the first three test cases. The stereo rig was moved slowly in the videos, with amplitude of about 3m and was rotated only to center the curve when the cameras

Figure 5.1: An example of a frame from the laboratory videos. The catenary was represented with a sequence of black squares to be detected by the same algorithm that detects ducts in the real application. (Source: The author)

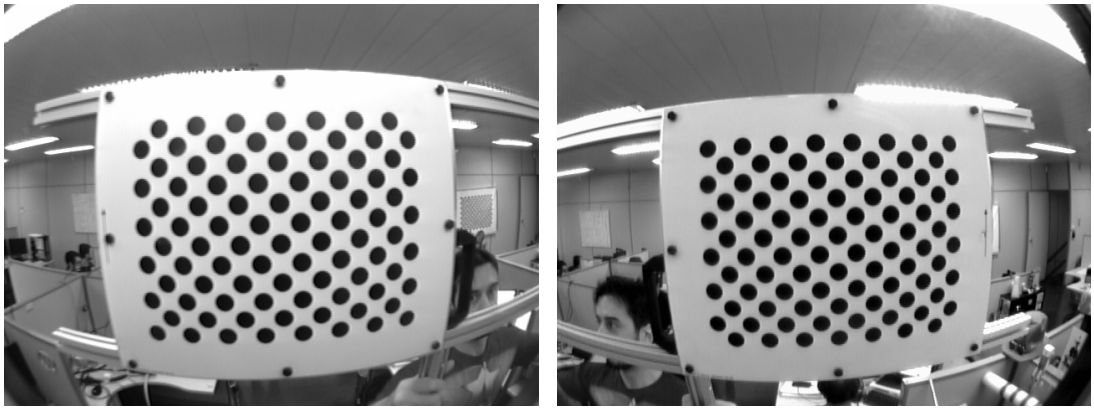
were in the extremities of the movements.

The Heaving test case was recorded with the cameras at 2 meters from the catenary and the amplitude of the vertical movement was approximately 1.5 meters. There was a slight lateral movement when the cameras were at the bottom position to avoid occlusion by an object from the scene. Figure 5.3 depicts the left camera video at the up (5.3a) and down (5.3b) extreme positions.

The Surging test case was recorded with the cameras centered with the catenary' centroid and the longitudinal movement had a length of approximately 3 meters starting at 1.5 meters from the curve. Figure 5.4 presents the left camera video at the front (5.4a) and back (5.4b) extreme positions.

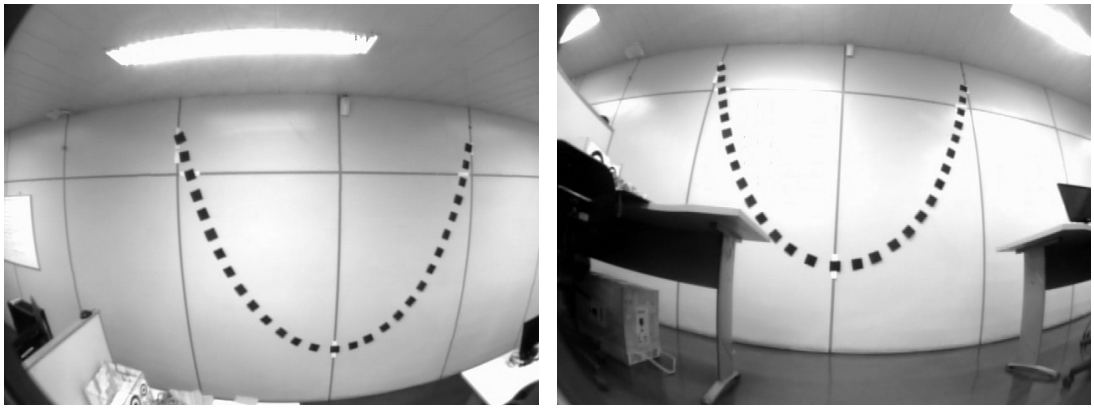
The Swaying test case was recorded with the cameras at 2 meters from the catenary, the lateral movement had a width of approximately 4 meters and was centered with the curve's centroid. Figure 5.5 illustrates the left camera video at the left (5.5a) and right (5.5b) extreme positions.

The Mixed Movements test cases is a mix of all the first three test cases in one video. The heaving and surging movements were repeated at approximately 1.5, 3.0 and 4.5 meters from the curve and were centered with the catenary's centroid. We use this test case to compare our proposal with the calibration by pattern, which is a more realistic representation of the movements that the stereo rig would perform in a real application scenario.



(a) **(b)**

Figure 5.2: Left (a) and right (b) frames from a calibration procedure using the pattern calibration algorithm. (Source: The author)



(a) Cameras in extreme up position. **(b)** Cameras in extreme down position.

Figure 5.3: Left camera frames from the Heaving test case video. (Source: The author)

5.2.2 Evaluation

We used two metrics to evaluate the 3D reconstructions: *Curve Fitting Error*, which is the Euclidean difference between the reconstructed points and the fitted curve; and *Minimum Radius of Curvature Error*, which is the difference between the parameter a of the reconstructed curve and the ground truth obtained with equation 5.2. Besides, we also compared the accuracy of the poses' estimated curves, which is the percentage of frames from each test case that estimated the parameter a with less than 3% and 1% of error.

When reconstructing points, the relative pose has a major influence in the points' precision, and thus evaluating the reconstruction error gives us a hint on whether the relative pose is accurate. Since obtaining a ground truth to the points' position in every frame is impracticable, we opted to compare point sets reconstructed using different poses with a catenary curve in which each set is fitted. The curve fitting error is the RMS of the difference between reconstructed points and the points in the fitted curve in which each point should be.

Comparing the minimum radius of curvature allows us to evaluate each frame's set of 3D points as a whole instead of just independent points, which are more susceptible to noise and

(a) Cameras in extreme front position. **(b)** Cameras in extreme back position.

Figure 5.4: Left camera frames from the Surging test case video. (Source: The author)

(a) Cameras in extreme left position. **(b)** Cameras in extreme right position.

Figure 5.5: Left camera frames from the Swaying test case video. (Source: The author)

outliers. To compute the minimum radius of curvature a from each frame, we used the sets of 3D reconstructed points from the frame to fit a catenary. The catenary parameters were estimated per frame optimizing

$$\underset{\mathsf{R},\mathbf{t},a,s_i}{arg\,min} \sum_{i=0}^{n} ||\varsigma(a,s_i) - \mathsf{R}\mathbf{X}_i + \mathbf{t}||, \tag{5.3}$$

where $\varsigma(a,s_i)$ is the equation 5.1 with a as a parameter, s_i represents the length of the catenary at the point $\mathbf{X}_i$ given a total of n points, and R and $\mathbf{t}$ are a rotation and a translation that determine the relative position and orientation of the curve from the cameras. Equation 5.3 was optimized using the Levenberg-Marquardt non-linear optimization algorithm (MARQUARDT, 1963).

5.3 Results

Having described our experiments' procedure, we now present the comparisons' results. Since our scenario is susceptible to outliers, we prefer to use the median, rather than the mean, as a measure of central tendency. We now show the results of the curve fitting error and the minimum radius of curvature error for each test case.

To obtain the results, we used a validation video with 1896 frames and few movements with the stereo rig to minimize the error, since exaggerate motion would affect the detection and reconstruction of the points. The results obtained for both the curve fitting error and the minimum radius of curvature error are shown in Table 5.1, where "Radius Error < 3%" and "Radius Error < 1%" are the percentage of frames which had a minimum radius of curvature with an error smaller than 3% and 1% of the ground truth, respectively. Figures 5.6, 5.7, 5.8 and 5.9 show the minimum radius of curvature error per frame in the validation video compared to the results from the pose obtained via calibration by pattern. These graphics depict the stability of the tested poses, therefore the interpretation of their different behavior in specific parts of the videos is not in the scope of this work.

Table 5.1: Results from the comparisons of the absolute curve fitting error (Fitting Error) and the absolute minimum radius of curvature error (Radius Error).

	Heaving		**Surging**		**Swaying**		**Mixed**		**Pattern**	
	$\bar{e}$	σ	$\bar{e}$	σ	$\bar{e}$	σ	$\bar{e}$	σ	$\bar{e}$	σ
Fitting Error (mm)	4.53	1.22	5.65	1.06	5.43	1.08	2.82	1.51	5.23	1.28
Radius Error (mm)	6.78	2.96	9.37	2.71	8.89	2.85	3.83	2.87	8.86	2.98
Confidence Interval (mm)	±0.18		±0.16		±0.17		±0.17		±0.18	
Radius Error < 3% (%)	100.00		98.78		99.42		100.00		100.00	
Radius Error < 1% (%)	35.23		5.50		10.88		68.87		20.61	

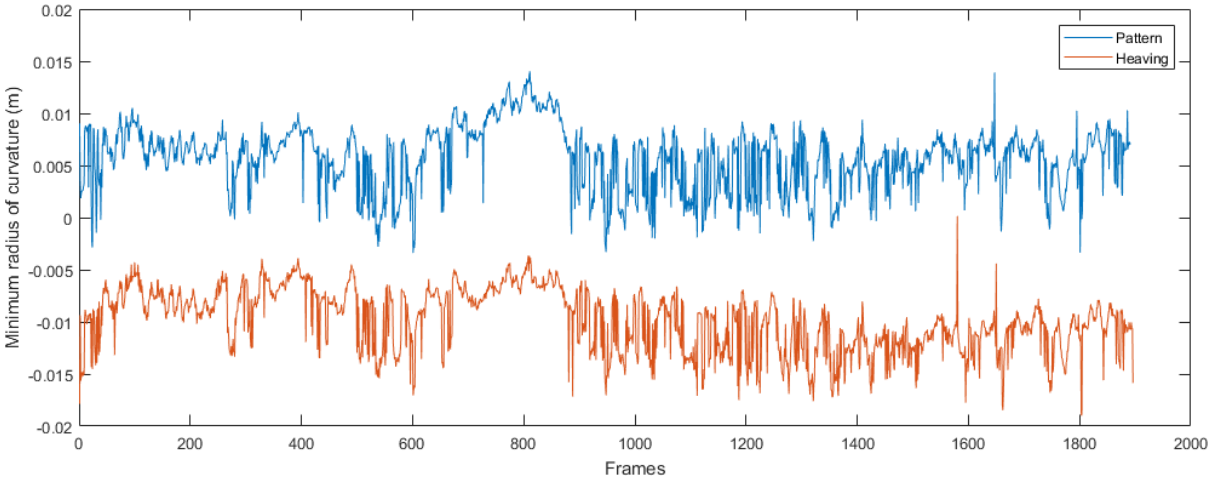


Figure 5.6: Minimum radius of curvature error in the *Heaving* case test. (Source: The author)

5.4 Discussion

As shown in Table 5.1, the Mixed Movements test case showed an overall smaller error on both metrics than the three test cases composed of only one linear movement. This result was expected because of the additional information from different perspectives that the first one has when compared to the other three. We can also see that behavior comparing Figure 5.9 with

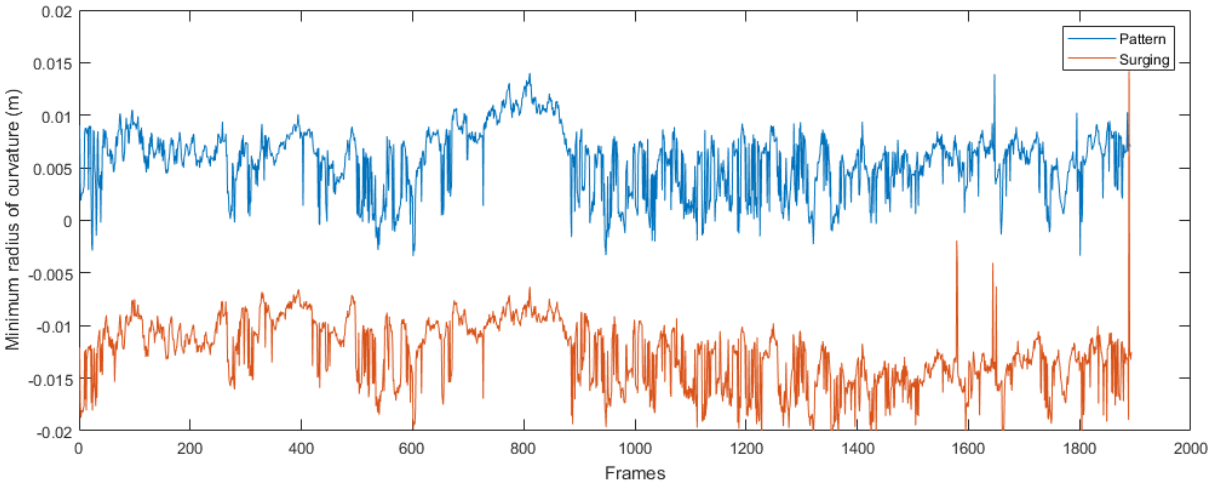


Figure 5.7: Minimum radius of curvature error in the *Surging* case test. (Source: The author)

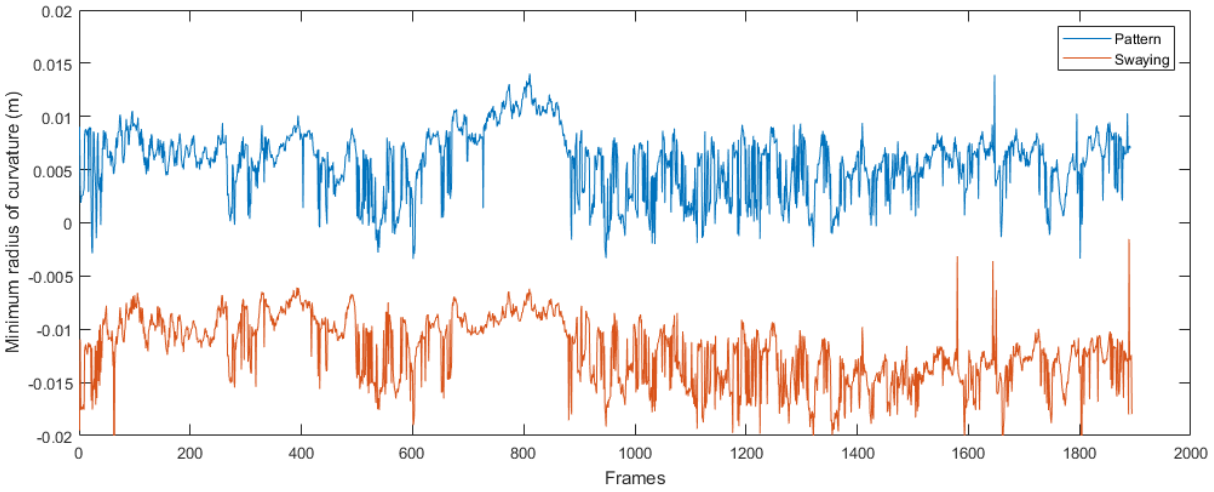


Figure 5.8: Minimum radius of curvature error in the *Swaying* case test. (Source: The author)

Figures 4.7, 4.8 and 4.9, where the error curve is closer to zero in the Mixed Movements test case.

Between the three single movement test cases, the Heaving video had the best results with smaller errors and more accuracy. Its minimum radii of curvature had errors below 1% of the ground truth in three times more frames than the other two cases. The Surging and Swaying test cases had worse results with an approximate error of one centimeter.

In the Surging case, this might have happened because the longitudinal movement doesn't provide a significant difference in the images' perspective, which means that the frames have image points with similar configurations and this tends to reduce the optimization's accuracy.

The Swaying case's results also had a disadvantage, which was that the obliquity of the curve's main plane was higher in the extreme positions. This culminates in a greater noise when detecting the vertebrae and thus reduces the pose accuracy.

When analyzing the results' accuracy, all cases had approximately 100% of the errors below 3% of the ground truth. However, only the Mixed Movements test case had more than

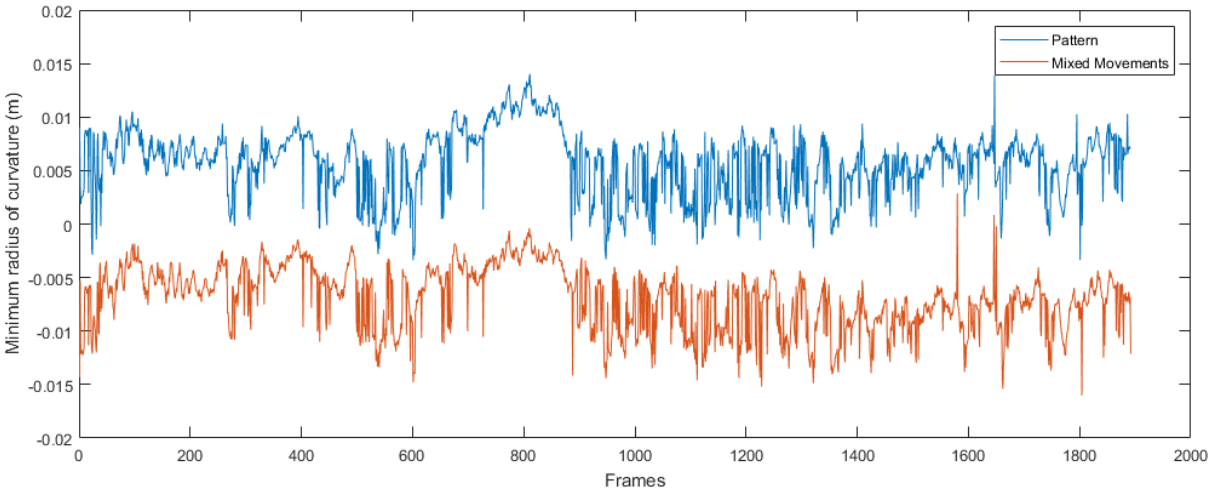


Figure 5.9: Minimum radius of curvature error in the *Mixed Movements* case test. (Source: The author)

50% of the results below 1% of error, followed by the Heaving pose and the Pattern pose with 35.26% and 20.61%, respectively.

As shown in Figure 5.9, while the pose from our solution tended to underestimate the radius of curvature, the pose calibrated by pattern overestimated it with an error similar to the Swaying test case. Using a confidence interval with 99% of reliability (see Table 5.1), our solution's pose had a difference greater than 4 millimeters to the pattern calibrated pose. Although our pose had better results, the curve fitting error and the minimum radius of curvature error of both poses were smaller than one centimeter. Such results are similar to the ones in SANTOS et al. (2015) and is within the "range of accuracy required for the installation operations", as mentioned by the authors.

The pattern calibration algorithm was expected to have a better accuracy, since it deals with more information and constraints to estimate the relative pose. One possible reason for the tests' outcome is that the validation video is similar to the calibration videos, which means that the poses are optimized specifically for those frames. This happened because there wasn't much noise in the test videos, however there is significantly more noise in deep underwater environments, which disfavors our solution.

Even without code optimizations, the presented solution has a low computational cost and the pose optimization converges in a few iterations. The matching stage can perform in real-time and the calibration algorithm can be executed in parallel with some other real-time vision applications.

In the next chapter, we discuss the achievements of our work and present our final considerations about it. We also suggest future works to improve our solution.

6

Conclusions

In this work, we presented a solution for automatic estimation of a stereo rig's relative pose in deep underwater installations of oil and gas ducts. This solutions is divided into two stages: a matching algorithm to find correlated points in a stereo image pair without having the stereo rig's extrinsic parameters; and a pose estimation algorithm that computes an initial pose and optimizes it. The proposed algorithm accomplished the objective of computing the relative pose between cameras in a stereo rig using images from the operations. We found it to be just as accurate as the state of art algorithm used in the targeted scenario (SANTOS et al., 2015).

Our proposal was tested in a controlled real-world scenario and we compared it to the pattern based calibration algorithm. From the tests and all the research, we highlight the following conclusions:

- The matching stage of the presented solution executed at approximately 19 frames per second, which is enough to obtain the required amount of information to compute an accurate relative pose in less than 5 min. With parallelization and some code optimizations, it might execute faster.

- The experiments at the laboratory showed that the relative pose estimated with our solution was sufficiently accurate according to the target application standards. In our test cases, 99.5% of the results had a minimum radius of curvature error inferior to 3% of the measured ground truth. When analyzing the mixed movements test case alone, 68.87% of the frames had an error of less than 6 millimeters.

  These results were similar to the ones obtained with the pattern calibration algorithm, which was the standard for the offshore operations, as stated by its authors. However, the camera images from the laboratory have significantly less noise than the underwater scenario. This means that further tests are necessary to validate our proposal for usage in real-world operations.

- When comparing the three test cases with unidirectional translations, the heaving case had the best results with an average curvature error of less than 7 millimeters and 35% of the frame errors bellow 1% of the ground truth. Although these three

cases had approximately two times the error of the mixed movements test case, they are all within the required accuracy to the offshore operations. This means that our proposal might still be accurate if the ROV movements are limited when executing the calibration.

This dissertation contributes with the study of computer vision applications for underwater duct installation operations by proposing a solution for point matching and relative pose estimation. Part of the matching stage was published in the paper "A dynamic-programming approach for point matching in calibrated stereo arrangements" at the Brazilian workshop WVC in 2016 (FIGUEIREDO et al., 2016).

6.1 Future Work

Even though we have accomplished our objectives with the presented solution, there are some aspects which could be improved.

Our proposal was tested in a scenario with little noise and had better results than the pattern calibration in some cases. However, these tests do not represent accurately the difference between the two algorithms, since the target application works exclusively in deep underwater scenarios. For a complete evaluation of our proposal, it should be tested on underwater scenarios, both controlled and in offshore operations. These scenarios have irregular illuminations and noise because of floating particles, which tends to worsen the results from our solution.

The current implementation matches points from a predetermined number of frames before estimating the relative pose. To improve our proposal's accuracy, the estimation could be executed every frame using the pose from the previous execution as input. This approach is supposed to converge fast so that the relative pose would be constantly updated and refined.

Another possible improvement is to add the coplanar restriction when optimizing the relative pose. In the current implementation, every point is optimized independently, but the duct vertebrae have a coplanar constraint, which could be considered when executing the Levenberg-Marquadt algorithm at the end of the pose estimation stage. This would reduce the reconstruction error perpendicular to the duct's main dominant plane.

At last, other sensors could be used to aid the relative pose optimization, such as a sonar. With it, one could detect the dominant plane and use it as an additional constraint when estimating and optimizing the pose, as in the previous paragraph. Using other sensors could help reduce the error of reconstructing the points and estimating the pose, since it would not only depend on the images from the cameras.

References

ABDEL-HAKIM, A. E.; FARAG, A. A. CSIFT: a SIFT descriptor with color invariant characteristics. In: COMPUTER VISION AND PATTERN RECOGNITION, 2006 IEEE COMPUTER SOCIETY CONFERENCE ON. **Anais...** [S.l.: s.n.], 2006. v.2, p.1978–1983.

BATRA, D.; NABBE, B.; HEBERT, M. An alternative formulation for five point relative pose problem. In: MOTION AND VIDEO COMPUTING, 2007. WMVC'07. IEEE WORKSHOP ON. **Anais...** [S.l.: s.n.], 2007. p.21–21.

BAY, H. et al. Speeded-up robust features (SURF). **Computer vision and image understanding**, [S.l.], v.110, n.3, p.346–359, 2008.

BAY, H.; FERRARI, V.; VAN GOOL, L. Wide-baseline stereo matching with line segments. In: COMPUTER VISION AND PATTERN RECOGNITION, 2005. CVPR 2005. IEEE COMPUTER SOCIETY CONFERENCE ON. **Anais...** [S.l.: s.n.], 2005. v.1, p.329–336.

BEIS, J. S.; LOWE, D. G. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In: COMPUTER VISION AND PATTERN RECOGNITION, 1997. PROCEEDINGS., 1997 IEEE COMPUTER SOCIETY CONFERENCE ON. **Anais...** [S.l.: s.n.], 1997. p.1000–1006.

BELONGIE, S.; MALIK, J.; PUZICHA, J. Shape context: a new descriptor for shape matching and object recognition. In: NIPS. **Anais...** [S.l.: s.n.], 2000. v.2, p.3.

BENTLEY, J. L. Multidimensional binary search trees used for associative searching. **Communications of the ACM**, [S.l.], v.18, n.9, p.509–517, 1975.

BERTHILSSON, R. Affine correlation. In: PATTERN RECOGNITION, 1998. PROCEEDINGS. FOURTEENTH INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 1998. v.2, p.1458–1460.

BRADSKI, G. OpenCV. **Dr. Dobb's Journal of Software Tools**, [S.l.], 2000.

CESAR, V. et al. Uncalibrated image rectification for coplanar stereo cameras. In: COMPUTER VISION THEORY AND APPLICATIONS (VISAPP), 2014 INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2014. v.3, p.648–655.

CHEUNG, W.; HAMARNEH, G. N-sift: N-dimensional scale invariant feature transform for matching medical images. In: BIOMEDICAL IMAGING: FROM NANO TO MACRO, 2007. ISBI 2007. 4TH IEEE INTERNATIONAL SYMPOSIUM ON. **Anais...** [S.l.: s.n.], 2007. p.720–723.

FATHIAN, K.; GANS, N. R. A new approach for solving the Five-Point Relative Pose Problem for vision-based estimation and control. In: AMERICAN CONTROL CONFERENCE (ACC), 2014. **Anais...** [S.l.: s.n.], 2014. p.103–109.

FAUGERAS, O. D.; MAYBANK, S. Motion from point matches: multiplicity of solutions. **International Journal of Computer Vision**, [S.l.], v.4, n.3, p.225–246, 1990.

FIGUEIREDO, R. F. et al. A dynamic-programming approach for point matching in calibrated stereo arrangements. **WVC 2016**, [S.l.], 2016.

FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. **Communications of the ACM**, [S.l.], v.24, n.6, p.381–395, 1981.

FITZGIBBON, A. W.; ZISSERMAN, A. Automatic camera recovery for closed or open image sequences. In: EUROPEAN CONFERENCE ON COMPUTER VISION. **Anais...** [S.l.: s.n.], 1998. p.311–326.

FRAUNDORFER, F.; TANSKANEN, P.; POLLEFEYS, M. A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles. **Computer Vision–ECCV 2010**, [S.l.], p.269–282, 2010.

FREDRIKSSON, J. et al. Optimal Relative Pose with Unknown Correspondences. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** [S.l.: s.n.], 2016. p.1728–1736.

GUENNEBAUD, G. et al. **Eigen v3**. 2010.

HARTLEY, R. I. In defense of the eight-point algorithm. **IEEE Transactions on pattern analysis and machine intelligence**, [S.l.], v.19, n.6, p.580–593, 1997.

HARTLEY, R. I.; ZISSERMAN, A. **Multiple View Geometry in Computer Vision**. 2.ed. [S.l.]: Cambridge University Press, ISBN: 0521540518, 2004.

HEDBORG, J.; FELSBERG, M. Fast iterative five point relative pose estimation. In: ROBOT VISION (WORV), 2013 IEEE WORKSHOP ON. **Anais...** [S.l.: s.n.], 2013. p.60–67.

HUTTENLOCHER, D. P.; KLANDERMAN, G. A.; RUCKLIDGE, W. J. Comparing images using the Hausdorff distance. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, [S.l.], v.15, n.9, p.850–863, 1993.

KALANTARI, M. et al. A new solution to the relative orientation problem using only 3 points and the vertical direction. **Journal of Mathematical Imaging and Vision**, [S.l.], v.39, n.3, p.259–268, 2011.

KE, Y.; SUKTHANKAR, R. PCA-SIFT: a more distinctive representation for local image descriptors. In: COMPUTER VISION AND PATTERN RECOGNITION, 2004. CVPR 2004. PROCEEDINGS OF THE 2004 IEEE COMPUTER SOCIETY CONFERENCE ON. **Anais...** [S.l.: s.n.], 2004. v.2, p.II–506.

KUKELOVA, Z. et al. Efficient solution to the epipolar geometry for radially distorted cameras. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION. **Proceedings...** [S.l.: s.n.], 2015. p.2309–2317.

LONGUET-HIGGINS, H. C. A computer algorithm for reconstructing a scene from two projections. **Nature**, [S.l.], v.293, n.5828, p.133–135, 1981.

LOWE, D. G. Object recognition from local scale-invariant features. In: COMPUTER VISION, 1999. THE PROCEEDINGS OF THE SEVENTH IEEE INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 1999. v.2, p.1150–1157.

MARQUARDT, D. W. An algorithm for least-squares estimation of nonlinear parameters. **Journal of the society for Industrial and Applied Mathematics**, [S.l.], v.11, n.2, p.431–441, 1963.

MATAS, J. et al. Robust wide-baseline stereo from maximally stable extremal regions. **Image and vision computing**, [S.l.], v.22, n.10, p.761–767, 2004.

MEDIONI, G.; NEVATIA, R. Segment-based stereo matching. **Computer Vision, Graphics, and Image Processing**, [S.l.], v.31, n.1, p.2–18, 1985.

MÜHLMANN, K. et al. Calculating dense disparity maps from color stereo images, an efficient implementation. **International Journal of Computer Vision**, [S.l.], v.47, n.1-3, p.79–88, 2002.

MUJA, M.; LOWE, D. G. Fast approximate nearest neighbors with automatic algorithm configuration. **VISAPP (1)**, [S.l.], v.2, n.331-340, p.2, 2009.

MUSLEH, B. et al. Pose self-calibration of stereo vision systems for autonomous vehicle applications. **Sensors**, [S.l.], v.16, n.9, p.1492, 2016.

NISTÉR, D. An efficient solution to the five-point relative pose problem. **IEEE transactions on pattern analysis and machine intelligence**, [S.l.], v.26, n.6, p.756–770, 2004.

NISTÉR, D. Preemptive RANSAC for live structure and motion estimation. **Machine Vision and Applications**, [S.l.], v.16, n.5, p.321–329, 2005.

OTSU, K.; KUBOTA, T. A two-point algorithm for stereo visual odometry in open outdoor environments. In: ROBOTICS AND AUTOMATION (ICRA), 2014 IEEE INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2014. p.1042–1047.

PESSOA, S. et al. A Segmentation Technique for Flexible Pipes in Deep Underwater Environments. In: BRITISH MACHINE VISION CONFERENCE (BMVC). **Proceedings...** BMVA Press, 2015. p.135.1–135.12.

SANTOS, I. et al. Real Time Radius of Curvature Measurement During DVC Operations Based on Flexible Pipe 3D Reconstruction. In: OTC BRASIL. **Anais...** [S.l.: s.n.], 2015.

SHI, J. et al. Good features to track. In: COMPUTER VISION AND PATTERN RECOGNITION, 1994. PROCEEDINGS CVPR'94., 1994 IEEE COMPUTER SOCIETY CONFERENCE ON. **Anais...** [S.l.: s.n.], 1994. p.593–600.

STEWENIUS, H.; ENGELS, C.; NISTÉR, D. Recent developments on direct relative orientation. **ISPRS Journal of Photogrammetry and Remote Sensing**, [S.l.], v.60, n.4, p.284–294, 2006.

SWEENEY, C.; FLYNN, J.; TURK, M. Solving for relative pose with a partially known rotation is a quadratic eigenvalue problem. In: D VISION (3DV), 2014 2ND INTERNATIONAL CONFERENCE ON, 3. **Anais...** [S.l.: s.n.], 2014. v.1, p.483–490.

TORR, P. H.; ZISSERMAN, A. MLESAC: a new robust estimator with application to estimating image geometry. **Computer Vision and Image Understanding**, [S.l.], v.78, n.1, p.138–156, 2000.

TRIGGS, B. et al. Bundle adjustment—a modern synthesis. In: INTERNATIONAL WORKSHOP ON VISION ALGORITHMS. **Anais...** [S.l.: s.n.], 1999. p.298–372.

WANG, Z.-F.; ZHENG, Z.-G. A region based stereo matching algorithm using cooperative optimization. In: COMPUTER VISION AND PATTERN RECOGNITION, 2008. CVPR 2008. IEEE CONFERENCE ON. **Anais...** [S.l.: s.n.], 2008. p.1–8.

WEI, Y.; QUAN, L. Region-based progressive stereo matching. In: COMPUTER VISION AND PATTERN RECOGNITION, 2004. CVPR 2004. PROCEEDINGS OF THE 2004 IEEE COMPUTER SOCIETY CONFERENCE ON. **Anais...** [S.l.: s.n.], 2004. v.1, p.I–106.

YANG, Z.; TANG, L.; HE, L. A New Analytical Method for Relative Camera Pose Estimation Using Unknown Coplanar Points. **Journal of Mathematical Imaging and Vision**, [S.l.], p.1–17, 2017.

ZHANG, Z. Determining the epipolar geometry and its uncertainty: a review. **International journal of computer vision**, [S.l.], v.27, n.2, p.161–195, 1998.

ZITOVA, B.; FLUSSER, J. Image registration methods: a survey. **Image and vision computing**, [S.l.], v.21, n.11, p.977–1000, 2003.