CENTRO DE INVESTIGACIONES EN ÓPTICA, A.C.

Metaheuristic Approaches for 3D-reconstruction-related Problems

A DOCTORAL THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF DOCTOR OF PHILOSOPHY OF CENTRO DE INVESTIGACIONES EN ÓPTICA

Alan López Martínez

Under the guidance of
Dr. Francisco J. Cuevas de la Rosa

June 2019
León, Guanajuato, México

Versión definitiva
Incluye cambios sugeridos por revisores

Final version
Includes changes suggested by reviewers

Abstract

In the computer vision field, methods used for 3D reconstruction can be applied also for image understanding, pose estimation, visual tracking, robot navigation, camera calibration, visual measurements, among others. Due to the importance of its applications, this thesis covers three different problems that are closely related with the sparse 3D reconstruction pipeline. Thus, the work presented in this dissertation covers the following problems: i) Estimating geometric relations between two different views of the same scene; ii) Detecting image vanishing points; and iii) Extracting circular markers from digital images. Since these problems can be visualized as modeling estimations commonly formalized as optimization problems, traditional optimization techniques are generally used. These are based on the gradient or the Hessian of the cost function such as Gauss-Newton, Levenberg-Marquardt or Barzilai-Borwein methods. However, when a considerable number of unwanted abnormal data is present, these methods might fail. Other solution methods relies on accumulator space techniques like the Hough Transform (HT), while others employ a heuristic approach such as the Random Sample Consensus algorithm (RANSAC). However, HT-like solutions are slow, whereas RANSAC-like methods are not optimal. To propose a different solution technique, in this work we explore the utilization of metaheuristics, such as evolutionary and swarm-based algorithms. Therefore, the solutions presented in this dissertation require less computational cost in comparison with HT methods and perform better than RANSAC-based solutions. Under the proposed mechanism, new candidate solutions are iteratively built by considering the quality of models that have been generated by previous candidate solutions, rather than relying over a pure random selection as it is the case with classic RANSAC. Further, our solutions explore the search space optimally requiring less computational cost than HT methods, and at the same time having the capability of escape local optima differently from traditional optimization methods. As a result, our metaheuristc-based algorithms present a nice balance between accuracy and computational time. To validate the efficacy of the proposed approaches, several tests and a comparison with other techniques were carried out.

Publications

Accepted

- **López, A.**, and Cuevas, F. J. (2018). Automatic multi-circle detection on images using the teaching learning based optimisation algorithm. *IET Computer Vision*, 12(8), 1188-1199.
- **López, A.**, and Cuevas, F. J. (2018). Automatic circle detection on images using the teaching learning based optimization algorithm and gradient analysis. *Applied Inteligence*, 49(5), 2001-2016.

In review

- **López, A.**, and Cuevas, F. J. Multiple View Relations Using the Teaching and Learning Based Optimization Algorithm. Submitted to *Pattern Analysis and Applications*.
- **López, A.**, and Cuevas, F. J. Vanishing point detection using the teaching learning based optimisation algorithm. Submitted to *IET Image Processing*.

Acknowledgements

I would like to thank Dr. F. J. Cuevas De la Rosa for his guidance. Similarly, I would like to express my gratitude to my thesis committee members, Dr. A. Dávila, Dr. M. de la Torre and Dr. A. Muñoz, for their insightful discussions about this work. Also, I thank Dr. M. S. Hernández, and Dr. H. Sánchez for their help and suggestions.

I would also like to show my gratitude to Consejo Nacional de Ciencia y Tecnología (CONACYT) for the funding received towards my PhD; that was determinant in the development and continuity of my project. Similarly, I greatly thank Centro de Investigaciones en Óptica authorities, professors, technicians, and administrative personnel for all their help.

Finally, last but by no means least, I would like to thank all of my colleagues and friends for their good company and intellectual generosity.

Dedication

To my family and friends for their constant support and encouragement.

The highest forms of understanding we can achieve are laughter and human compassion.

Richard Feynman

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 3D-reconstruction-related Problems

Sparse 3D reconstruction based on multiple view geometry is a popular research topic in computer vision. This is due to the more information and wide scope that is provided from the real-scene-view of multiple views. However, to perform sparse 3D reconstruction, combination of different approaches can be used. Actually, more than one pipeline for 3D reconstruction might exist. 3D-reconstruction-related problems are feature detection and matching, fundamental matrix estimation, camera self-calibration, camera pose estimation, along with others.

This dissertation describes the implementation of a set of algorithms that are related to the 3D reconstruction problem. These algorithms involve i) Feature detection and matching along with fundamental and homography matrix estimation; ii) Vanishing point estimation; and iii) Circular marker extraction. The present thesis analyses and compares the merits and drawbacks of the methods. It also focuses on metaheuristic-based approaches, such as Differential Evolution (DE) and the more recent Teaching Learning Based Optimization (TLBO) algorithm, among others.

1.2 Motivation and Objectives

The main objective of this thesis is to explore the utilization of metaheuristics in 3D-reconstruction-related problems: i) Epipolar geometry estimation; ii) Vanishing points detection; and iii) Circular marker extraction that can be applied for camera pose (position and orientation) estimation. To this purpose, different objectives have been set:

- To explore a novel application of the TLBO algorithm to different tasks related to image processing problems, and to sparse 3D reconstruction.
- To improve the search process in comparison with heuristic approaches.
- To implement an accurate approach for estimating the fundamental and homography matrices.
- To implement an accurate approach for vanishing point estimation.
- To implement an accurate approach for circular marker extraction.

1.3 Statement of Originality

I hereby declare i) that the work presented in this thesis have been carried out by me under the supervision of my advisor, Dr. Cuevas De la Rosa; and ii) that the work is original and has not been submitted in another thesis or dissertation to get a degree, similar or equal to the one intended to obtain here, or to any other university or institution.

1.4 Thesis Outline

The contents of this dissertation are as follows. Chapter 2, describes the minimum background theory needed to understand the present work. Chapter 3 depicts our proposed approach for the estimation of multiple view relations. Chapter 4, on its part, explains the proposed method

for vanishing point estimation. Further, Chapter 5 describes the work done in circular marker extraction. Finally, Chapter 6 states the conclusions and future work.

Chapter 2

Preliminaries: Background Theory

The work presented in this dissertation focuses in solving 3D-recostruction-related problems treated as combinatorial optimization problems. Thus, in this chapter, useful notation and terminology for that matter are formulated. Also, we will describe metaheuristic approaches for global optimization. We will present a brief review of concepts from projective geometry, which forms the basis for mathematical representation of multiple view geometry. Large segments of this chapter borrow liberally from [1].

The rest of the chapter follows this organization: Section 2.1 depicts the epipolar geometry; while Section 2.2 describes the concept of a vanishing point, and analyzes different methods for its estimation. Section 2.3, on its part, explains the problem of circular marker extraction. Finally, Section 2.4 presents different metaheuristics approaches, and discusses the need of this type of techniques for the presented work.

2.1 Epipolar Geometry

2.1.1 Introduction

Estimating geometric relations between images is to find a transformation to associate images of the same scene, but taken at different viewpoints [1]. The estimation of the epipolar geometry,

which is the intrinsic projective geometry between two views, can be found from different experimental setups. For instance, when a moving camera captures a static scene or a static camera views a moving object, or even when multiple cameras capture the same scene from different viewpoints. Mathematical expressions of the epipolar geometry are the fundamental matrix and the homography.

2.1.2 The Fundamental Matrix and Homography

In this work, we tackle the problem of estimating geometric relations from point correspondences. These relations are given by the Fundamental matrix $\boldsymbol{F}$, and the Homography $\boldsymbol{H}$. In this section, we first describe the operations to compute point correspondences, and then we show how to compute the fundamental matrix and the homography given by the epipolar restriction, using these point correspondences.

Image matching

The feature correspondence task aims to find the pixel coordinates in two different images I and I' that refer eventually to the same point in the world. The image matching process consists of three main operations: i) Feature detection; ii) Feature description; and iii) Feature matching.

- *i) Feature detection.* In the detection operation, we must find stable matching primitives. Choosing special points when matching images and perform a local analysis on these ones instead of looking at the image as a whole has the advantage of reducing the computational cost. There are many feature points detectors, some of these are: Harris Corners [3], Scale Invariant Feature Transform (SIFT) [4], Speeded Up Robust Features (SURF) [5], Features from Accelerated Segment Test (FAST) [6], Binary Robust Independent Elementary Features (BRIEF) [7], Oriented FAST and Rotated BRIEF (ORB) [8] and others. In this work, the SURF method has been employed since its complexity is $O(log(N))$, and the method is invariant to illumination, scale, and rotation. Also, the SURF detector has demonstrated to be effective for both, high and low resolution images [9].

- *ii) Feature description.* In the second operation, feature description, the previously detected features in I and I' are described with a compress structure. The descriptors of the image features can be computed with some algorithms including SIFT, SURF, BRIEF along with others. In this work, the SURF descriptor is employed. The SURF algorithm computes a 64-element descriptor vector to characterize each feature point. When the SURF detector and descriptor are applied to the images I and I', two sets of feature points described by its own vector are obtained, namely $E = \{e_1, e_2, \cdots, e_M\}$ and $E' = \{e'_1, e'_2, \cdots, e'_{M'}\}$, respectively.

- *iii) Feature matching.* In the final operation, feature matching, the descriptor of each feature within the first set is compared with all other descriptors in the second set using some distance calculation. In this work, the Euclidean distance is used to compare descriptor vectors from the first image with descriptors of the second image to build pairs of corresponding points; the match is selected as the one with the shortest distance. After these three operations, N point matches $\boldsymbol{x_i} \leftrightarrow \boldsymbol{x'_i}$ are found, and a set $U = \{\boldsymbol{x_1} \leftrightarrow \boldsymbol{x'_1}, \cdots, \boldsymbol{x_N} \leftrightarrow \boldsymbol{x'_N}\}$ with matches is generated. For this process, an erroneous estimation of matched points may emerge on different sections of the images. This is due to the fact that the process does not discriminate with complete certainity one point from another.

The noisy dataset U obtained with the above process is the input of our algorithm to compute the geometric relations given by the fundamental matrix and the homography as described in Chapter 3. We know describe these geometric relations that our method estimates considering the epipolar geometry.

Fundamental Matrix

Let there be a set of N matched points $\boldsymbol{x_i} \leftrightarrow \boldsymbol{x'_i}$ between two images I and I'. The 2D image positions of these points are denoted in homogeneous coordinates as $\boldsymbol{x_i} = (x_i, y_i, 1)^T$ in the I image, and $\boldsymbol{x'_i} = (x'_i, y'_i, 1)^T$ in the I' image. These positions are related by the epipolar geometry as follows:

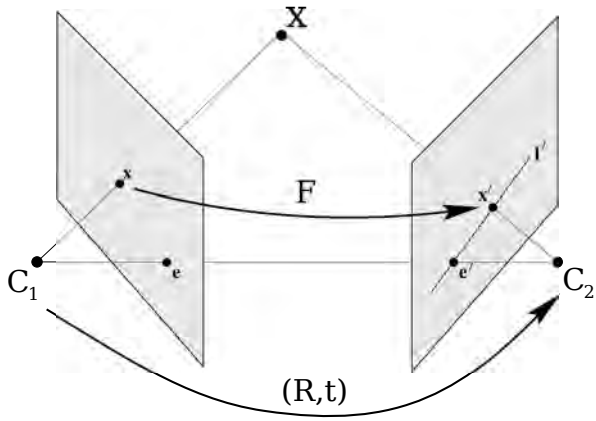

Figure 2.1: Point correspondence geometry: Fundamental matrix. The two cameras are indicated by their centres $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$ and image planes. The camera centres, 3-space point $\boldsymbol{X}$, and its images $\boldsymbol{x}$ and $\boldsymbol{x}'$ lie in a common plane. An image point $\boldsymbol{x}$ back-projects to a ray in 3-space defined by the first camera centre, C_1, and $\boldsymbol{x}$. This ray is imaged as a line $\boldsymbol{l}'$ in the second view. The 3-space point $\boldsymbol{X}$ which projects to $\boldsymbol{x}$ must lie on this ray, so the image of $\boldsymbol{X}$ in the second view must lie on $\boldsymbol{l}'$ [1].

$$\boldsymbol{x_i'}^T \boldsymbol{F} \boldsymbol{x_i} = 0, \tag{2.1}$$

where $\boldsymbol{F}$ is the fundamental matrix, and can be computed with a set of eight good matches as described in [1,10]. The epipolar geometry represents the intrinsic geometry between two-views. It is independent of the scene structure and only depends on the camera's internal parameters and relative localization between the cameras $(\boldsymbol{R}, \boldsymbol{t})$. The fundamental matrix $\boldsymbol{F} \in \mathbb{R}^{3\times3}$ is the algebraic representation of this intrinsic geometry, called epipolar geometry.

The epipolar geometry can be used to validate the match $\boldsymbol{x_i} \leftrightarrow \boldsymbol{x_i'}$, since it constrains the position of the points $\boldsymbol{x_i}$ and $\boldsymbol{x_i'}$. As shown in Figure 2.1, the epipolar line at the point $\boldsymbol{x_i'}$, in the second image, is the intersection of the epipolar plane passing through the optical centers $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$ and the point $\boldsymbol{X_i}$ within the plane of the second image. If the matrix $\boldsymbol{F}$ and the point $\boldsymbol{x_i}$ in the first image are known, the epipolar line in the second image where the point $\boldsymbol{x_i'}$ is restricted to be, is given by $\boldsymbol{F}\boldsymbol{x_i}$. Similarly, $\boldsymbol{F}^T\boldsymbol{x_i'}$ specifies the epipolar line in the first image that corresponds to the point $\boldsymbol{x_i'}$ in the second image. This epipolar constraint that restrict the positions of $\boldsymbol{x_i}$ and $\boldsymbol{x_i'}$ is used in our proposed method to evaluate candidate solutions.

Given a matrix $\boldsymbol{F}$ computed from noisy point correspondences, the quality of the estimated

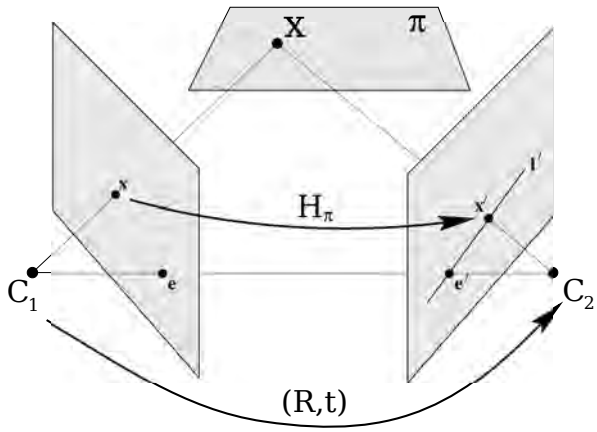

Figure 2.2: Epipolar geometry: The Homography case. A point x in one image is transferred, differently from the fundamental-matrix case, via the plane π to a matching point x' in the second image. The epipolar line through x' is obtained by joining x' to the epipole e'.

fundamental matrix is evaluated using the epipolar lines. This is done by considering the distance between the points and the epipolar lines to which they must belong. Considering the notation $\boldsymbol{F}\boldsymbol{x_i} = (\beta_1, \beta_2, \beta_3)^T$, the distance $d(\boldsymbol{x'_i}, \boldsymbol{F}\boldsymbol{x_i})$ between the point $\boldsymbol{x'_i}$ and the line $\boldsymbol{F}\boldsymbol{x_i}$ can be computed as follows:

$$d(\boldsymbol{x'_i}, \boldsymbol{F}\boldsymbol{x_i}) = \frac{\boldsymbol{x'_i}\boldsymbol{F}\boldsymbol{x_i}}{\sqrt{\beta_1^2 + \beta_2^2}}. \tag{2.2}$$

Likewise, denoting $\boldsymbol{F}^T\boldsymbol{x'_i} = (\beta'_1, \beta'_2, \beta'_3)^T$, the other corresponding distance can be calculated as:

$$d(\boldsymbol{x_i}, \boldsymbol{F}^T\boldsymbol{x'_i}) = \frac{(\boldsymbol{x'_i})^T\boldsymbol{F}\boldsymbol{x_i}}{\sqrt{(\beta'_1)^2 + (\beta'_2)^2}}. \tag{2.3}$$

To evaluate $\boldsymbol{F}$, a mismatch error EF_i^2 produced by the i-correspondence $\boldsymbol{x_i} \leftrightarrow \boldsymbol{x'_i}$ is defined by the sum of squared distances from the points to their corresponding epipolar lines as follows:

$$EF_i^2 = [d(\boldsymbol{x'_i}, \boldsymbol{F}\boldsymbol{x_i})]^2 + [d(\boldsymbol{x_i}, \boldsymbol{F}^T\boldsymbol{x'_i})]^2. \tag{2.4}$$

Homography

If the match points are said to be in a plane π, a homography can be computed. As shown in Figure 2.2, two perspective images can be geometrically linked through a plane π of the scene by a homography $\boldsymbol{H} \in \mathbb{R}^{3\times3}$. The homography $\boldsymbol{H}$ is a projective transformation, and it relates matching or corresponding points belonging to the plane π that is projected into two images by $\boldsymbol{x'_i} = \boldsymbol{H}\boldsymbol{x_i}$ or $\boldsymbol{x_i} = \boldsymbol{H}^{-1}\boldsymbol{x'_i}$. To find the homography given by two different views, a linear system from a set of four different corresponding points (matches) can be generated, and then solved [1].

The estimated homography can be evaluated to obtain an accuracy measure. This quality measure can be computed by considering the distance (usually the Euclidean distance) between the position of the point estimated with the $\boldsymbol{H}$ matrix and the actual position of the projection of the observed 3D point. Thereby, the i-correspondence $\boldsymbol{x_i} \leftrightarrow \boldsymbol{x'_i}$ will produce a mismatch error EH_i^2 that is defined as the sum of squared distances from the points to their estimated positions with the following equation

$$EH_i^2 = [d(\boldsymbol{x'_i}, \boldsymbol{H}\boldsymbol{x_i})]^2 + [d(\boldsymbol{x_i}, \boldsymbol{H}^{-1}\boldsymbol{x'_i})]^2. \tag{2.5}$$

Having described the epipolar geometry, and how a candidate fundamental matrix $\boldsymbol{F}$ and homography $\boldsymbol{H}$ can be evaluated, we now depict work related with the applications of the epipolar geometry, and also different methods for its estimation.

2.1.3 Epipolar Geometry: Applications and Related Work

The importance of estimating geometric relations becomes evident considering that it is a necessary task for many computer vision applications. For instance, geometric relations are needed for stitching together series of images to generate a panorama image [11–13]. Also, super-resolution approaches for multiple images are applied in overlapped regions that are estimated conforming to the calculated geometry [14, 15]. Other applications are related to the

calibration of camera networks, where each camera pose and focal length are computed with their associated correspondences [16–18]; to medical images, where the localization and tracking of objects is needed [19, 20]; to the control of robots, where homographies are used [21–23]; to the removal of camera movements, when studying the motion of an object in a video [24]; among other uses [25–27].

Most of the solutions for the estimation of geometrical relations are very complex and sometimes inefficient, since classical optimization techniques are used. For instance, conventional approaches to geometrical solutions can be found in [28–32]. Despite their popularity, these conventional methods present the great weakness of being very sensitive to the initial solution. If the initialization is far from the optimum, then it is difficult to converge to an optimal solution. Thus, those classical optimization methods also have more chance of being entrapped in a local minimum.

To avoid the drawbacks of the aforementioned methods, yet preserving accuracy, heuristic approaches have been proposed. The most popular of such approaches is the RANdom SAmple Consensus (RANSAC) [33]. In the RANSAC algorithm, a minimum number of samples of experimental data are randomly taken. Then, for each sample, a model is proposed and evaluated according to a distance error to determine how well each model fits the data. This process is repeated until a number of iterations are completed, and the model with the lower error (maximum number of inliers) is taken.

As shown in this subsection, the information contained in the fundamental matrix or homography is related to the camera pose. Thus, as explained in the introduction of the section, this information can be used for 3D reconstruction. Another important concept that might arise in the pipeline of 3D reconstruction is the vanishing point. Finding the vanishing point can be used for camera calibration, therefore estimation of vanishing points is sometimes included in the 3D reconstruction pipeline. We now describe in the next subsection the vanishing point.

2.2 Vanishing Points

2.2.1 Introduction

A Vanishing Point (VP) is defined in the literature as the convergence point of projected 3D parallel lines in an image plane. A VP can be located either at infinity or at some finite distance. For example, for a set of parallel lines in the 3D world, their corresponding VP lies at infinity when the 3D lines are projected onto a set of parallel lines in the 2D image plane. Conversely, when the 2D projections of the 3D parallel lines are *nonparallel* 2D lines, the VP lies at a certain finite distance. Different methods for VP extraction have been proposed; the next subsection presents techniques for VP detection.

2.2.2 Vanishing Point Estimators

State of the art methods for VP estimation can be categorized into three classifications: Algorithms aiming to estimate a single VP, like the work of [34]; three orthogonal VP as in [35]; or any possible non-orthogonal VP as done in [36]. While some methods require knowledge of camera calibration, others operate in a non-calibrated setting. Other traditional methods on VP detection use a Hough transform of the line segments on the Gaussian sphere [37–39], but usually they are low. Thus, more recent VP detection algorithms use Random Sample Consensus (RANSAC) algorithms or accumulator space techniques. For instance, in the work of [36, 40] VP hypotheses are generated by computing the intersections of line segments and then selecting the most probable ones using image-based consistency metrics and RANSAC. Other algorithms also rely on a RANSAC-like approach [40]. Vanishing points have different applications as explained in the next subsection.

2.2.3 Vanishing Points: Applications and Related Work

A VP can be used as an invariant feature for inferring the 3D structure of a real scene. Thus, VP estimation is extensively used in 3D reconstruction [1], augmented reality applications [41], camera calibration algorithms [42], robotic navigation methods [43, 44], visual measurement [45,46], image understanding [47], among other applications. Since these are relevant problems, developing an efficient VP detector has been extensively investigated, and different solutions have been offered. We now explain in more detailed different methods for VP detection.

Ideally, estimation of vanishing points can be done by finding the intersection of a set of imaged parallel scene lines [1]. However, due to measurement noise, the imaged line segments will generally not intersect in a unique point. Therefore, VP detection can be visualized as a clustering problem, where line sets are to be organized according to a common VP.

Usually, VP detection strategies proceed by first computing line clustering, and then performing VP estimation. For line clustering, the lines segments are first categorised into groups according common VPs located relatively close from each other. Then, an estimated VP is located using lines from the line segment cluster. Similar to other methods, our proposed approach performs a line segment clustering, and then executes an estimation step. Specifically, the line classification problem is treated as a clustering problem solved by the TLBO algorithm. Then, the set of optimal VPs is computed within an estimation step.

Before we describe the proposed VP detection process in Chapter 4, we analyse some line clustering algorithms. There are two categories of line classification methods: i) algorithms that relies on accumulator spaces; and ii) methods that estimate line clustering directly on the image plane. For the methods within the first category, an accumulator space is generated. In this space, each cell accumulates votes of all the lines passing through a corresponding image point, i.e. a tentative VP. Then, for VP estimation, the cells that have accumulated the most line votes generate the detected VPs. For the VP-detection methods within the second category, the line clustering is typically executed by computing the distances among lines and points. Since computing every point-line distance is unattainable, heuristic methods are used.

Typical methods are based on the RANSAC approach, such as the J-Linkage algorithm and the multi-RANSAC method [48]. The methods based on RANSAC principles iteratively select a sampling set of image features to compute a tentative VP. To retrieve a VP, the more consistent feature set, i.e. the set with the more votes, is used. Conversely, rather than relying over a pure random selection as it is the case of RANSAC-like algorithms, our approach additionally uses the TLBO algorithm to improve the clustering search.

There is another difference between line clustering algorithms: the need for prior information regarding intrinsic camera parameters. Accumulator-space-based algorithms additionally require the knowledge of the calibration parameters. Also, the performance of these algorithms are strongly dependent on the selection of the accumulator space. Differently, RANSAC-like algorithms perform the clustering directly in the image plane, and are executed in an uncalibrated setting. Nevertheless, RANSAC-based algorithms do not assure an optimal solution to VP detection.

Another method to compute sets of lines meeting at multiple vanishing points is the Expectation-Maximisation (EM) algorithm [49]. EM alternates between expectation and maximisation steps. Within the E step, an estimation of the line clustering from candidate VPs is carried out. Within the M step, on the other hand, the VPs computation is performed using the data clusters found in the previous E step. A drawback for the EM algorithm is that it strongly depends on the set of originally found VPs.

Circular-marker detection can be used as well within a 3D reconstruction pipeline. For instance, triangulation methods can be performed after detection of the same marker viewed from two different perspectives. Thereby, we describe in the next section the problem of automatic circle detection.

2.3 Automatic Circular-Markers Detection

2.3.1 Introduction

The importance of automatic circle detection becomes evident considering the variety of computer vision applications for which circle extraction is a key element. We have mentioned that circular markers can be used as non-ambiguous features to perform, for instance, triangulation techniques. Also, sometimes circles are used as fiducial markers for augmented and mixed reality. Other applications for circle extraction are related to, for example, biometric authentication systems that rely largely in an accurate iris segmentation [50, 51]. In robotics, on the other hand, circular markers have been used for visual-based localization purposes [52, 53]. Similar applications seize on man-made objects such as traffic signs [54] for autonomous navigation tasks. Regarding industrial and manufacturing applications, circle extraction has been used for autonomous inspection [55, 56], and remote sensing processing used for oil circular depots detection [57,58]. Given the relevance to this problem, different approaches for circle extraction have been proposed to offer a solution; these methods are described in the next subsection.

2.3.2 Circular Markers Detection: Related Work

The methods based on the Hough transform (HT) have been extensively used due to their effectiveness. The classical Circular HT (CHT) performs an exhaustive exploration of the search space by mapping every point in the input image into a parametric space, replacing the difficult pattern detection in the image space into a local peak searching in the parameter space. This approach makes the CHT robust to noise, yet computationally inefficient. Consequently, some new approaches based on the CHT principles have been proposed to improve both storage and computational requirements. For instance, some methods modify the architecture of the traditional CHT in order to propose a multi-threaded implementation [59], or a FPGA implementation of the CHT [60]. Other approaches exploit the CHT principles in combination with gradient information and curvature analysis. For instance, the Curvature Aided Hough

transform for Circle Detection (CACD) algorithm estimates the circle center and radius by adaptively estimating curvature radius [61]. The CACD approach estimates the curvature on the edge-only image to perform a one-to-one vote using accumulator arrays of different radius ranges. Similarly, the work in [62] uses first-order and second-order derivatives to compute a one-to-one dense CHT; however, the method does not performs edge extraction as a preprocessing step.

There are some other recent approaches that also exploit gradient information along with other geometrical properties of circles to perform circle detection. For instance, some work use the fact that the center of a circle lies along the normal direction of edgels (edge pixels), and that the intersection of these normal lines is the center of the circle [63,64]. In the work of [65], on the other hand, an inverted gradient hash map is used to detect circles by using the fact that pairs of edgels on a circumference differ in orientation by 180 degrees. Another method implements an isosceles triangles sampling to estimate centers and radius [66]. The method performs a refinement process using chords and a linear compensation based on gradient information of edgels.

Other approaches rely on a sampling procedure to reduce the computational time for the circle detection task. An example is a recent method called GRCD-R (Gradient-based Randomized Circle Detection with Refinement) [67]. This method randomly selects four non-collinear edgels. Then, gradient orientation is used to test if the gradient vectors point to the center of a candidate circle. After that validation, three edgels are used to compute a circle, and the last one determines whether the candidate circle is a possible circle in function of its distance to the candidate circumference. Finally, a voting procedure is performed to decide whether possible circles are to be upgraded to detected circles. The work in [68] adds further constraints based on the curvature of the isophotes. Isophotes are curves connecting pixels in the image with equal intensity, whose properties make them particularly suitable for object detection.

Other methods use arc segments instead of edgels to find circles and ellipses in binary images [69, 70]. EDCircles [71], for instance, uses edges extracted by an edge segment extractor named EDPF (Edge Drawing Parameter Free). The edges are then converted into lines, and lines into

circular arcs, which are joined together using heuristic algorithms to fit circles and ellipses. To eliminate false detections, EDCircles uses a contrario validation step based on the Helmholtz principle. Their idea is to compute the level line orientation field (which is orthogonal to the gradient orientation field) of a given image, and look for a contiguous set of pixels having similar level line orientation.

The three problems explained above (Epipolar geometry estimation, Vanishing point detection, and circle extraction) are solved in this work by considering them as optimization problems. Thus, we now describe metaheuristic approaches for solving optimization and combinatorial problems.

2.4 Metaheuristics

2.4.1 Introduction

Metaheuristics refer to techniques utilized for solving combinatorial optimization problems. The functionality principle of metaheuristics is based on a guided heuristic search that uses predefined black box functions. These black box functions can be heuristic themselves, and are used to evaluate the function value (usually called objective function or fitness function) for a given state of the model (solution). Since metaheuristics make few ssumptions about the optimization problem, they are commonly employed when the faced problem is intractable or difficult to solve using one of the more classical optimization methods that are generally based on gradient descent.

Metaheuristics are bio-inspired, this means that the search strategy derives from mimicking the operations of biological systems [72]; e.g. ant colony [73, 74], bee colony [75], genetic chromosomes [76], differential evolution [77]. More recent metaheuristics are Harmoney Search [78], Firefly algorithm [79], Teaching Learning Based Optimization [80], among others.

Differently from metaheuristics, a global optimization approach may be considered deterministic when the objective function and constraints are not specified in probabilistic terms and there

is no element of randomization in the search algorithm. Metaheuristics, on the other hand, are suited for problems where a degree of randomness is needed due to probabilistic modeling of the problem. This is what makes metaheuristic approaches for approximation a popular method used to optimize unknown functions. In this thesis dissertation, several single and multiple view geometry problems are globally optimized. We explore these problem-specific aspects in Chapters 3, 4 and 5 of the dissertation.

Although metaheuristic approaches have common characteristics, they can be classified, according to the literature, into the following categories:

- **Evolutionary-based Computation Methods**. Evolutionary computation (EC) methods are derivative-free procedures, i.e. they are not based on gradient descent. Thereby, EC methods do not need the objective function to be neither two-times differentiable nor uni-modal. For this reason, EC techniques are used as global optimization algorithms since they can deal with multi-modal, non-linear, and non-convex problems, depite if they are subject to non-linear or linear constraints with discrete or continuous decision variables.

- **Swarm Intelligence**. These are algorithms that stems from the collective behavior of species. Usually, the metaphors are taken from birds, termite, firefly, fish, ants, bees among others. The so called swarm intelligence originates from the social behavior of those species that search and compete for nourishment. The main characteristics of swarm-intelligence-based algorithms are the use of simple particles and nonsophisticated agents, they cooperate by an indirect communication medium, and do movements in the decision space. Among the most successful swarm intelligence inspired optimization algorithms are ant colony and particle swarm optimization.

2.4.2 Common Concepts for Evolutionary Algorithms

The main search components for designing a metaheuristic are as follows: 1. representation; 2. Population initialization; 3. Objective function; 4. Selection strategy; 5. Reproduction or

individual-combination strategy; 6. Replacement strategy; and 7. Stopping criteria. We now describe these components as usually defined in the literature.

1. **Representation**. This is a common search component for all metaheuristics. In the EC (specially Genetic Algorithms (GA)) community, the encoded solution is referred as chromosome while the decision variables within a solution (chromosome) are genes. The possible values of variables (genes) are the alleles and the position of an element (gene) within a chromosome is named locus.

2. **Population initialization**. This is a common search component for all population-based metaheuristics.

3. **Objective Function**. This is a common search component for all metaheuristics. In the EC community, the term fitness refers to the objective function.

4. **Selection strategy**. The selection strategy addresses the following question: *Which parents (individuals) for the next generation are chosen with a bias toward better fitness?*

5. **Reproduction or individual-combination strategy**. The reproduction strategy consists in designing suitable mutation and crossover operator(s) to generate new individuals (offsprings).

6. **Replacement strategy**. The new offsprings compete with old individuals for their place in the next generation (survival of the fittest).

7. **Stopping criteria**. This is a common search component for all metaheuristics. Some stopping criteria are specific to population based metaheuristics.

In Chapters 3 through 5, different methods based on metaheuristics are proposed. Thus, we describe the different components of metaheuristic-based solutions focusing on the search space organization, the individual representation, and the objective-function definition.

2.4.3 Structure of a Metaheuristic Algorithm

As stated in the literature, a metaheuristic method is an algorithm that mimics at some level of abstraction a natural, social or biological system. A standard metaheuristic algorithm includes:

1. One or more populations of candidate solutions (models) are considered.
2. These populations change dynamically due to the production of new solutions considering previously tested candidate models.
3. A fitness function or objective function expresses the capability of an individual (model solution) to survive and reproduce.
4. Different operators are employed in order to exploit and explore appropriately the search space (space of solutions.)

The metaheuristical methodology states that, due to the evolution process, a set of individuals (candidate solutions) whose fitness (objective function to optimize) will, on average, lead to an improvement of their fitness. Consequently, the metaheuristic will steer the simulated population towards the global solution.

We now describe the Differential Evolution approach and the Teaching Learning Based Optimization algorithm in more detailed, since we use both approaches within the experiment sections.

Differential Evolution Algorithm

The Differential Evolution (DE) algorithm is a simple and direct search technique that is considered as a population-based method aiming to optimize multimodal functions [77]. Similar to the Genetic Algorithm, which is also a population-based method, the DE approach employs evolutionary operators such as the crossover and mutation operators, along with a selection mechanism. Despite the similarities between DE and GA, they differ in the importance given

to the operators. For instance, while the DE approach relies on the mutation operation as the main operator, the GA method relies on the crossover operator to exchange information among individuals (solutions). Also, the DE algorithm employs a non-uniform crossover. This allows the algorithm to take child vector parameters from one parent more frequently than others. In consequence, the DE method is able to focus the search on most promising areas of the solution space contributing to the exploitation capabilities of the DE algorithm. This is due to the fact that the non-uniform crossover operator efficiently interchanges information about successful solutions (combinations).

Classic DE algorithms begin by initializing a population of N_p and D-dimensional vectors considering parameter values that are randomly distributed between the pre-specified lower initial parameter bound $x_{j_{low}}$ and the upper initial parameter bound $x_{j_{high}}$ as follows:

$$x_{i,j,t} = x_{i,j_{low},t} + rand(0,1)\ (x_{i,j_{high},t} - x_{i,j_{low},t}), \tag{2.6}$$

$$j = 1, 2, \cdots, D; i = 1, 2, \cdots, N_p; t = 0.$$

The subscript t is the generation index, while j and i are the parameter and population indexes, respectively. Hence, $x_{j,i,t}$ is the jth parameter of the ith population vector in generation t. In order to generate a trial solution, DE algorithm first mutates the best solution vector $\boldsymbol{x}_{best,t}$ from the current population by adding the scaled difference of two vectors from the current population.

$$\boldsymbol{v}_{i,t} = \boldsymbol{x}_{best,t} + F(\boldsymbol{x}_{r1,t} - \boldsymbol{x}_{r2,t});\ r1, r2 \in \{1, 2, \cdots, N_p\}, \tag{2.7}$$

with $\boldsymbol{v}_{i,t}$ being the mutant vector. Vector indices r_1 and r_2 are randomly selected such that they are different $r_1 \neq r_2 \neq i$. Another important parameter is F, the mutation scale factor. $0 < F < 1$ is a positive real number. Figure 2.3 illustrates how the vector-generation is carried out by Equation 2.7.

After the above process is performed, the next step in the crossover operation is carried out.

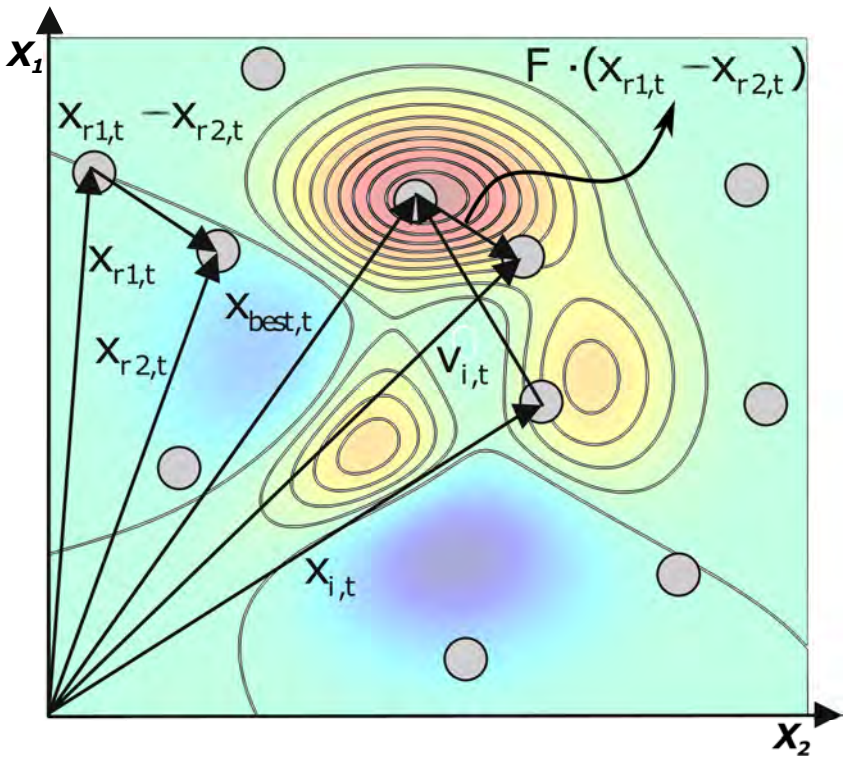

Figure 2.3: Objective function example. the two-dimensional objective function and the process for generating v from vectors of the current generation

Within this process, one or more parameter values of the mutant vector $\boldsymbol{v}_{i,t}$ are exponentially crossed with those belonging to the ith population vector $\boldsymbol{x}_{i,t}$. The result is the trial vector $\boldsymbol{u}_{i,t}$ which is computed by considering element to element as follows:

$$u_{j,i,t} = \begin{cases} v_{j,i,t} & \text{if } rand(0,1) < Cr \text{ or } j = j_{rand} \\ x_{j,i,t} & \text{otherwise} \end{cases}, \tag{2.8}$$

with $j_{rand} \in \{1, 2, \cdots, D\}$.

To control the contribution of the mutant vector, the crossover constant ($0.0 \leq Cr \leq 1.0$) is used. Also, the computed trial vector always inherits the mutant vector parameter in an heuristic fashion. Thus, the index j_{rand} is chosen randomly, assuring that the trial vector differs by at least one parameter from the vector to which it is being compared ($\boldsymbol{x}_{i,t}$).

Finally, the last step is performed to create better solutions. To this purpose, the selection operation is used. Within the selection process, if the corresponding objective function value

for the trial vector, $(\boldsymbol{u}_{i,t})$, is less than or equal to target vector, $(\boldsymbol{x}_{i,t})$, then $(\boldsymbol{u}_{i,t})$ replaces $(\boldsymbol{x}_{i,t})$ within the next generation. Otherwise, the target vector $(\boldsymbol{x}_{i,t})$ remains in the population for at least one more generation. This is done as:

$$x_{i,j,t+1} = \begin{cases} u_{i,j,t} & \text{if } f(\boldsymbol{u_{i,t}}) \leq f(\boldsymbol{x_{i,t}}) \\ x_{i,j,t} & \text{otherwise} \end{cases}, \tag{2.9}$$

where, the objective function is represented by f. These processes are repeated for finding an global minimum. To this purpose, the whole set of processes are carried until a pre-determined generation number is reached, or a termination criterion is attained.

Teaching Learning Based Optimization Algorithm

The TLBO algorithm, that was first proposed by Rao et al. in [80], is a method for solving complex optimization problems with a simple structure and an easy implementation. This metaheuristic is inspired in the traditional school learning process where students knowledge increases by two means: i) the effects of teaching of a teacher upon a student and ii) the knowledge transfer through the interaction between schoolmates. Given that the TLBO algorithm has been demonstrated to be a successful heuristic, many engineering and scientific applications have been published [80–83].

The TLBO algorithm is a recently developed population-based heuristic algorithm that is inspired by the teaching and learning process. The principle of the algorithm is based on two main ideas: one, a student is able to improve its knowledge by learning from the teacher. And two, a student knowledge could be improved through the interaction with other students. In this algorithm, a group of students is considered as the population and the different subjects offered are compared to the problem variables. Similarly, the objective function value is compared with student grades. And the most qualified student, i.e. the best solution, is viewed as the teacher.

The TLBO starts with an initialization procedure, where N random numbers of initial solutions

(students) $\boldsymbol{X}_i$ $i \in \{1, \cdots, N\}$ are generated within the search space. The algorithm continues with two consecutive phases: the teacher phase and the learner phase. During the first phase, the teacher $\boldsymbol{X}_{teacher}$ (best solution) tries to increase the knowledge level of the whole class and to help students individually to get better grades. To achieve this, the algorithm attempts to shift the value of students in the group towards the teacher value. After the phase consisting in learning from the teacher is completed, the second phase is performed. This final phase attempts knowledge increase through the interaction between students. During this phase any student can interact with any other student for knowledge transfer. The algorithm is terminated after a certain number of iterations are completed.

As explained above, the element $\boldsymbol{X}_i$ of the population represents a single possible solution to a particular optimization problem. $\boldsymbol{X}_i$ is a real-valued vector with D elements, where D is the dimension of the problem and is used to represent the number of subjects that an individual enrolls for within the TLBO context. The algorithm then tries to improve certain individuals by changing these individuals during teacher and learner phase.

During the teacher phase, the best individual is assigned as the teacher. The algorithm attempts to improve other individuals by moving their position towards the position of the teacher by taking into account the current mean value of the individuals. The student position $\boldsymbol{X}_i$ is updated by:

$$\boldsymbol{X}_{new} = \boldsymbol{X}_i + r \cdot (\boldsymbol{X}_{teacher} - T_F \cdot \boldsymbol{X}_{mean}). \tag{2.10}$$

The former equation indicates how the improvement of student $\boldsymbol{X}_i$ may be influenced by the difference between the knowledge of the teacher and the qualities of all students. In Eq. 2.10, r is a real random number between 0 and 1 and TF, called the teaching factor, can be either 1 or 2 and is decided randomly with equal probability. $\boldsymbol{X}_i$ is replaced by $\boldsymbol{X}_{new}$ if the latter gives better fitness value.

As stated in [80], the intuition behind the teaching factor is that in practice a teacher can only improve the quality of a class up to some extent depending on the capability of the class. Thus,

Algorithm 1: *Classical TLBO algorithm.*

```
Algorithm TLBO
Initialize the population of students X randomly.
Compute the fitness F(X) of each student in the population.
for each iteration do
    Calculate the mean of each student in the population: (X_mean).
    Find out the best solution, i.e. (X_teacher).
    r ← rand(0,1)
    TF ← round(1+rand(0,1))
    for each X_i in population do
        Teacher phase:
        X_new ← X_i + r · (X_teacher − TF · X_mean)
        if (F(X_new) > F(X_i)) then
            X_i ← X_new
        Learning phase:
        Select randomly another different student X_j with i ≠ j
        if (F(X_i) > F(X_j)) then
            X_new ← X_i + r · (X_i − X_j)
        else
            X_new ← X_i + r · (X_j − X_i)
        if (F(X_new) > F(X_i)) then
            X_i ← X_new
Find out the best solution, i.e. X_teacher.
Select X_teacher as the final solution.
```

the teaching factor is decided randomly for heuristic purposes, and can be either 1 or 2, thus emphasizing the importance of student quality. Also, in Equation 2.10, r and TF contribute to the exploration capabilities of the algorithm.

The teacher phase aims to increase the mean of the class through the teacher, who put maximum effort into teaching its students, but students gain knowledge according to the quality of the imparted teaching. However, the possibility for students to improve their knowledge is not completely lost. During the second and final phase, the learner phase, a student learns with the help of other students. In general terms, the quantity of knowledge transferred to a student does not only depend on its teacher but also on interactions amongst students.

During the learner phase, student $\boldsymbol{X}_i$ tries to improve its knowledge by learning from an arbitrary student $\boldsymbol{X}_j$. In the case that $\boldsymbol{X}_j$ is better than $\boldsymbol{X}_i$, $\boldsymbol{X}_i$ is moved towards $\boldsymbol{X}_j$ according to:

$$\boldsymbol{X}_{new} = \boldsymbol{X}_i + r \cdot (\boldsymbol{X}_j - \boldsymbol{X}_i). \tag{2.11}$$

Otherwise, it is moved away from $\boldsymbol{X}_j$ according to:

$$\boldsymbol{X}_{new} = \boldsymbol{X}_i + r \cdot (\boldsymbol{X}_i - \boldsymbol{X}_j). \tag{2.12}$$

The objective of this phase is to attain knowledge transfer from a more qualified student to a less qualified student. To this end, $\boldsymbol{X}_i$ is replaced by $\boldsymbol{X}_{new}$ if the latter gives better fitness value. The process above, involving the two phases, is repeated until a certain termination criterion is met. Algorithm 1 describes the simplest form of the TLBO algorithm.

In this subsection, we described two different metaheuristcs implemented for this work. Another important concept to describe is the number of objective function evaluations. We now described this concept.

2.4.4 Number of Objective Function Evaluations

When the number of FEs is the same for all methods, a fair comparison is guaranteed since all algorithms sample the search space an equal number of times. Otherwise, an algorithm with more FEs has, a priori, more chances to provide a better solution, since it gathers more information of the problem. The number of FEs for each approach is now described.

Given M individuals and N iterations, the number of FEs for the TLBO algorithm is

$$FEs_{TLBO} = (2N + 1) \cdot M. \tag{2.13}$$

The above equation is easily deduced considering that in the initialization step for the TLBO algorithm, the objective function is evaluated once for each individual; and that at each iteration, two function evaluations are carried out for each individual (one per step: teacher and

learner). Distinctively, the DE-based detector only requires one objective function evaluation at each epoch for each individual, and another within the initialization step. Thus, the number of FEs for the DE algorithm is

$$FEs_{ED} = (N + 1) \cdot M. \tag{2.14}$$

The final case to describe is the GA-based detector, with a number of FEs found by

$$FEs_{GA} = N \cdot M. \tag{2.15}$$

The former equation is true, since within the GA approach, there is no need for function evaluations in the initialization step. Function evaluations are perfomed once for each individual at each generation in the selection step.

Having explained the main concepts of metaheuristic approaches, we now present in the next subsection why metaheuristic techniques have been proposed for solving the problems presented in the previous sections (Epipolar geometry estimation, vanishing point detection, circle extraction). We also explain the advantages of our proposed methods.

2.4.5 The Need of Metaheuristic Approaches for Combinatorial Problems

As explained in previous subsections, both deterministic and heuristic approaches have been utilized to solve the problem of estimating the epipolar geometry, the detection of vanishing points, and the extraction of circular markers. The more popular of the heuristic methods is RANSAC. However, even considering that RANSAC is a robust and simple algorithm, it has some disadvantages. We discuss this issue for each problem in the following subsections.

Metaheuristics for Epipolar Geometry Estimation

Two main difficulties for the RANSAC algorithm occur with a high multi-modality problem and a noisy dataset [84]. For a better estimation under such circumstances, the number of samples, N_{RANSAC}, and the threshold error, Th_{error}, must be tunned. Ideally, these parameters should be set with consideration of the relationship between the dataset and the model. In other words, optimal settings are chosen with information about the number of outliers. Unfortunately, this is not a simple task; usually it is assumed that the measurement error is Gaussian with zero mean [1]. The problem arises with a poor chose of N_{RANSAC} and Th_{error}. A small sample set, N_{RANSAC}, makes the algorithm fast, but leads to inaccuracies. A larger N_{RANSAC} improves accuracy, but at the cost of a higher computational burden. Likewise, a low Th_{error} value increases the accuracy of the model but degrades its generalization ability to tolerate noisy data. By contrast, a high Th_{error} value improves the noise tolerance of the model, but adversely drives the process to false detections.

The aforementioned RANSAC drawbacks make way for recent approaches utilized to solve engineering problems that usually are ill-posed and complex. This approaches applied modern optimization methods such as swarm and evolutionary algorithms described in Section 2.4, which have delivered better solutions in comparison with classical methods. The use of metaheuristics for estimating multiple view relations has been reported using the Harmony Search algorithm (HS) [85, 86], and the Clonal Selection Algorithm (CSA) [84]. However, when using these metaheuristics, the need of many parameter tunning still remains. For instance, the CSA needs the tunning of the mutation rate along with the clonal size, the length of the antibody and others. HS, on the other hand, requires the harmony memory consideration rate, pitch adjusting rate, and the number of improvisations.

Differently from previous metaheuristc approaches, in this work we propose the utilization of the TLBO algorithm as explained in Chapter 3. By using a swarm-based algorithm, we provide our method with a better guided search while keeping the parameter tunning to a minimum, since only the number of iterations and the size of the population are needed, and no other algorithm-specific parameters need to be tuned. As a result, our method iteratively

builds candidate solutions by considering the quality of previously generated models, rather than being a pure random selection as the RANSAC algorithm, and also requiring a minimum parameter tunning.

Metaheuristics for Vanishing Point Estimation

As already stated, main difficulties for the RANSAC algorithm occur with a high multi-modality problem and a noisy dataset. Regarding vanishing point estimation, VP detectors have been proposed using an Artificial Bee Colony (ABC) [87], the Particle Swarm Optimization (PSO) [88], and the Binary PSO (BPSO) [89]. However, when using these metaheuristics, the need of many parameter tunning still remains. For instance, ABC requires a proper tuning of parameters such as the number of bees (employed, scout, and onlookers), limit, and others. The same is the case with PSO, which uses inertia weight, social and cognitive parameters, along with number of iterations and population size. The BPSO, on its part, additionally requires the length of the binary string.

In the interest of improving the autonomy of the metaheuristic-based detector, yet preserving its capabilities, in this work we propose, similar with the estimation of epipolar geometry, a VP detector based on the Teaching Learning Based Optimization (TLBO) algorithm.

In this work, as described in Chapter 4, an automatic VP detection on images using the TLBO algorithm is presented. To our knowledge, the TLBO algorithm has not been applied for this task before. Our method treats VP detection as a clustering problem with little supervision and without any camera internal parameter input. Further, it does not use any accumulator space technique, since our method works directly in the image plane.

Metaheuristics for Circle Detection

The methods already described for circle detection are efficient, but sometimes they produce false and missing detections. Thence, evolutionary and swarm-based algorithms have been studied as well as a meta-heuristic manner to perform circle extraction. For example, circle detectors

have been proposed using the Differential Evolution approach [90], the Electromagnetism-like optimization (EMO) algorithm [91], the Artificial Bee Colony (ABC) method [92], the Collective Animal Behavoir (CAB) approach [93], the Harmony Search (HS) algorithm [94], and the Clonal Selection Algorithm (CSA) [95]. Unfortunately, despite the fact that many meta-heuristics have been used for the circle detection task, to our knowledge, less effort has been done into exploring the use of gradient information within the search process. In particular, the pattern present in circular shapes within the gradient orientation field has not been used to better guide the search with the objective function.

If the objective function is improperly defined, it can lead to non-acceptable solutions whatever meta-heuristic is used. Most of the reported work evaluates the fitness of an individual by essentially counting the number of pixels in the target edge map that coincide with the perimeter of candidate circles. The fitness value is given by the ratio of candidate to target pixels. They also punish points not exactly on the perimeter of a candidate circle using a displacement factor. A drawback of this approach becomes evident while avoiding small false positive circles and working in noisy conditions. The reason is that this approach tends to favor smaller circles since fewer pixels are required to define a high ratio.

Another drawback of circle detectors based on meta-heuristics is that they need the tuning of several parameters. For these methods, a set of algorithm-specific parameters must be tuned appropriately to accomplish good accuracy and high detection rate when dealing with different image conditions, such as variations in illumination and blurring boundaries. For instance, the DE approach requires the tunning of the differential weight, and the crossover probability. The EMO algorithm, on its part, needs the tunning of the step length. ABC requires tuning of the number of bees (employed, scout, and onlookers), limit, and others. CAB, on its part, requires the probability of attraction, the probability of random movement, and the elite size. Similarly, HS requires tunning of the harmony memory and pitch adjusting rates, and the number of improvisations. In the case of CSA, the mutation rate is to be tuned, along with the clonal size, the length of the antibody and others.

To overcome the aforementioned issues, in this work we propose in Chapter 5, differently from

previous meta-heuristic-based circle detectors, the utilization of gradient information to better customize the search space, and to improve the search guidance provided by the objective function. Additionally, in the interest of reducing the number of parameters to tune, we use another natural phenomena inspired optimizer, the already presented Teaching Learning Based Optimization (TLBO) algorithm [80]. As already mentioned, the TLBO algorithm does not require the tuning of many parameters in comparison with other meta-heuristics, since only the number of iterations and the size of the population are needed, and no other algorithm-specific parameters need to be tuned.

Chapter 3

Multiple-View Relations Considering Metaheuristic Approaches

3.1 Introduction

In this chapter we describe the estimation of geometric relations between two different views of the same scene. Once having this geometric relations, one can apply them in visual tracking, pose estimation and more, and later used them for 3D sparse reconstruction. Different heuristic techniques have been proposed to solve the modeling problem related to the estimation of multiple view geometry. The most popular of these methods is the Random Sample Consensus (RANSAC). However, as mentioned in the last chapter, the RANSAC algorithm presents a drawback regarding the tunning of two parameters: The number of samples, and the threshold error. A small sample set leads to inaccuracies; but a larger set, to a higher computational burden. Likewise, a low threshold error improves the accuracy of the model, but degrades its ability to tolerate noisy data; whereas a high threshold error increases the noise tolerance, but adversely drives the process to false detections. To solve this problem, we use the TLBO algorithm in the problem of estimating the multiple view relations given by the Homography $\boldsymbol{H}$ and the Fundamental matrix $\boldsymbol{F}$. To improve the method, a more accurate objective function is incorporated to accurately evaluate the quality of a candidate model. To validate the efficacy

of the proposed approach, several tests and a comparison with the RANSAC algorithm and other metaheuristics were executed.

The rest of the chapter is organized as follows. The next section, Section 3.2, describes our proposed method. Section 3.3 exposes the results; and finally, Section 3.4 states the conclusions and future work.

3.2 Epipolar Geometry Estimation using TLBO

In this section, we describe the proposed approach for the estimation of the fundamental matrix and homography, that encapsulate the epipolar geometry (See Chapter 2). To this purpose, we depict the utilization of the TLBO algorithm to find multiple view relations, which is a novel task for this metaheuristic. Since our estimator is based on a metaheuristic, we describe the following elements: The search space organization, the individual representation, and the objective-function definition.

3.2.1 Search Space

To generate the search space where the TLBO algorithm optimizes, two gray-scale images are processed to find key-features with its corresponding descriptors as described in Chapter 2. Then, the descriptors are matched using the Euclidean distance in order to find matching pair points $\boldsymbol{x_i} \leftrightarrow \boldsymbol{x'_i}$. Finally, every pair is added in a set $U = \{\boldsymbol{x_1} \leftrightarrow \boldsymbol{x'_1}, \cdots, \boldsymbol{x_M} \leftrightarrow \boldsymbol{x'_M}\}$, where M is the number of matches found. For our algorithm to work, it is assumed that the tentative matches in U are consistent with at least one geometric relation model, i.e. a fundamental matrix or a homography. In other words, the two input images are supposed to view the same scene from different view points.

3.2.2 Individual Representation

In the estimation process, each candidate student S encodes either a homography $\boldsymbol{H}$, or a fundamental matrix $\boldsymbol{F}$. In order to construct a candidate solution or individual S_i that encodes a homography, four indexes are selected from the set of correspondences U. Likewise, in the case of the fundamental matrix, eight indexes are selected from U. A transformation from indexes to either $\boldsymbol{H}$ or $\boldsymbol{F}$ is done according to [1].

3.2.3 Objective Function

In this work, the problem to solve consists in estimating the parameters of $\boldsymbol{F}$ or $\boldsymbol{H}$ through a set of M different noisy correspondences. To find a solution using the TLBO algorithm, this problem is treated as an optimization procedure.

Our method implements the TLBO to generate samples or candidate solution based on information about their quality, rather than pure randomness as in the case of the RANSAC algorithm. In the traditional RANSAC method, the objective function corresponds only to the number of inliers. Distinctively, we improve results by using a different objective function to accurately evaluate the quality of a candidate model.

The objective function fuse the number of inliers and their residual error into a simple quotient by the following expression:

$$F(S) = \frac{\sum_{j=1}^{N} \theta\big(e_j^2(h_i)\big)}{\sum_{j=1}^{N} e_j^2(h_i)}, \tag{3.1}$$

where $e_j^2(h_i)$ represents the quadratic errors EF_j^2 or EH_j^2 (see Eq. 2.4 and Eq. 2.5 in Chapter 2) produced by the jth correspondence considering the candidate transformation $\boldsymbol{F_i}$ or $\boldsymbol{H_i}$, whereas $\theta\big(e_j^2(h_i)\big)$ is defined as follows:

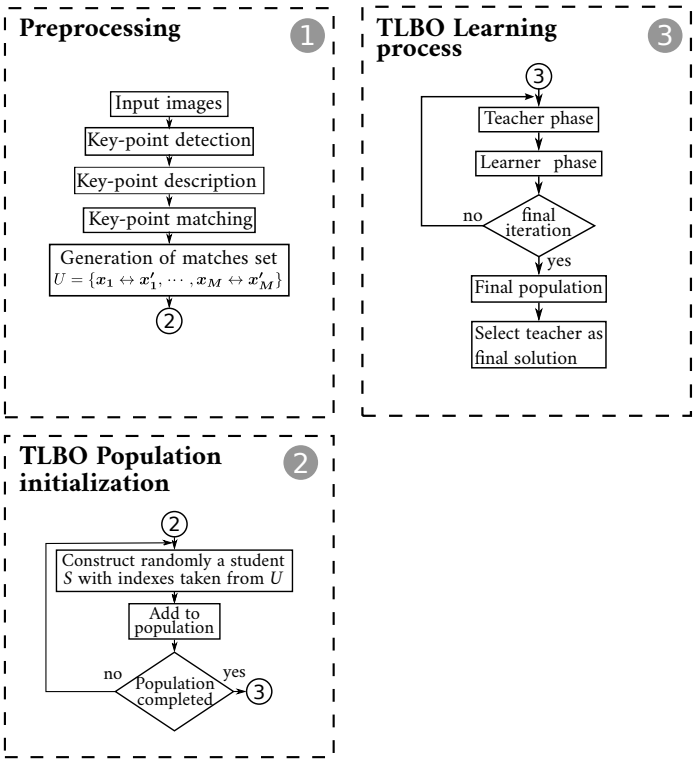

Figure 3.1: Summary of the proposed approach. First, the preprocessing step aims to generate and customize the search space. Then, different solutions (a.k.a population) are generated within the TLBO initialization procedure. Finally, to propose a solution, the metaheuristic optimization is performed by the TLBO learning process. The main contributions of this solution reside in the utilization (and adaptation of the search space and individual coding) of a novel metaheuristic; and the preprocessing procedure to construct the search space

$$\theta\big(e_j^2(h_i)\big) = \begin{cases} 0 & e_j^2(h_i) > Th \\ 1 & e_j^2(h_i) \leq Th \end{cases}. \tag{3.2}$$

Therefore, the maximization of $F(\cdot)$ implies to obtain the candidate solution S holding the highest number of inlier and the lowest residual error, simultaneously.

The objective function $F(\cdot)$ evaluates the quality of a candidate transformation. Guided by the values of this objective function, the set of encoded candidate solutions are modified by using the TLBO process so that they can improve their quality as the optimization process evolves. We now describe the whole process proposed in this work.

3.2.4 TLBO for Epipolar Geometry Estimation

The presented approach can be summarized as shown in Figure 3.1. First, a preprocessing step is carried out to construct the search space. Then, within the initialization procedure of the TLBO algorithm, candidate solutions are randomly generated. After that, the teaching and learning processes of the TLBO algorithm are iteratively executed to improve the quality of the population. Finally, at the end of the defined number of iterations, the best individual ($S_{teacher}$) is selected as the final solution. Within the next section, Experimental Results, we show how our proposed solution actually obtains accurate results.

3.3 Experimental Results

In this section, results of the application of the proposed method on real images are reported, and compared with five different approaches: RANSAC as implemented in OpenCV, MLESAC as implemented in the Matlab computer vision toolbox, CSA-RANSAC as implemented in [84], and an implementation using a Genetic Algorithm (GA), and the Differential Evolution approach (DE) with the same search space and objective function as the TLBO-based estimator.

All the experiments were executed on a 2.80GHz Intel Core i7-7700HQ CPU, with a C++ implementation of the algorithm; key-point detection and description were perfomed by OpenCV.

Table 3.1: Parameter setup for the TLBO-based estimator

Parameter	Value
Iterations	100
Population size	50

Table 3.2: Parameter setup for the GA-based estimator

Parameter	Value
Number of generations	201
Population size	50
Crossover rate	0.85
Mutation rate	0.10
Selection method	Roulette with sigma scaling
Crossover method	1-point crossover

Table 3.3: Parameter setup for the DE-based estimator

Parameter	Value
Number of epochs	200
Population size	50
Differential weight	0.25
Crossover probability	0.80

The experimental setup includes the use of real images commonly used in the literature. These images are shown in Figures 3.2 through 3.4. Room_1 (640×480) and Room_2 (640×480) belong to the CIMAT-NAO-A dataset, which was acquired with a NAO humanoid robot, and is available in http://personal.cimat.mx:8181/ hmbecerra/CimatDatasets.zip. The dataset contains 399 different images with blur effects and low textures. Street_1 (1241×376) belongs to the Kitty dataset which is usually used for autonomous driving experiments [96]; this dataset is publicly available in http://www.cvlibs.net/datasets/kitti/eval_odometry.php. Street_2 (1348×374), on its part, is the work of [97], and can be found in http://www.cvlibs.net/datasets/karlsruhe_sequences/ Finally, Book_1 (671×503) and Book_2 (671×503) were capture by a generic cellphone camera.

The sets of parameters for the TLBO, GA and DE are shown in Table 3.1, Table 3.2 and Table 3.3, respectively. The parameters have been chosen empirically in favor of performing a fair comparison. First, the settings for the TLBO algorithm were experimentally defined. Then, the parameters of the DE algorithm and GA that control the number of objective function evaluations were set specifically to equate this number for all approaches. Finally, the rest of the DE and GA parameters were set after experimentation. Also, to compute the objective function the parameter Th was set to 5. It is important to recall that the parameter setup of each metaheuristic remained fixed for every image. The parameter setups, as explained above, were set empirically as usually done for these type of techniques. In this case, empirical tunning means that different setups were tested to finally select the setup with the best results.

To show the capabilities of the proposed method for the task of estimating the fundamental matrix $\boldsymbol{F}$, we present Figure 3.2 and Figure 3.3. Our method also estimates the homography $\boldsymbol{H}$, as shown by the results of Figure 3.4. Ground-truth odometry and calibration parameters are available for images in Figure 3.2 and Figure 3.3.

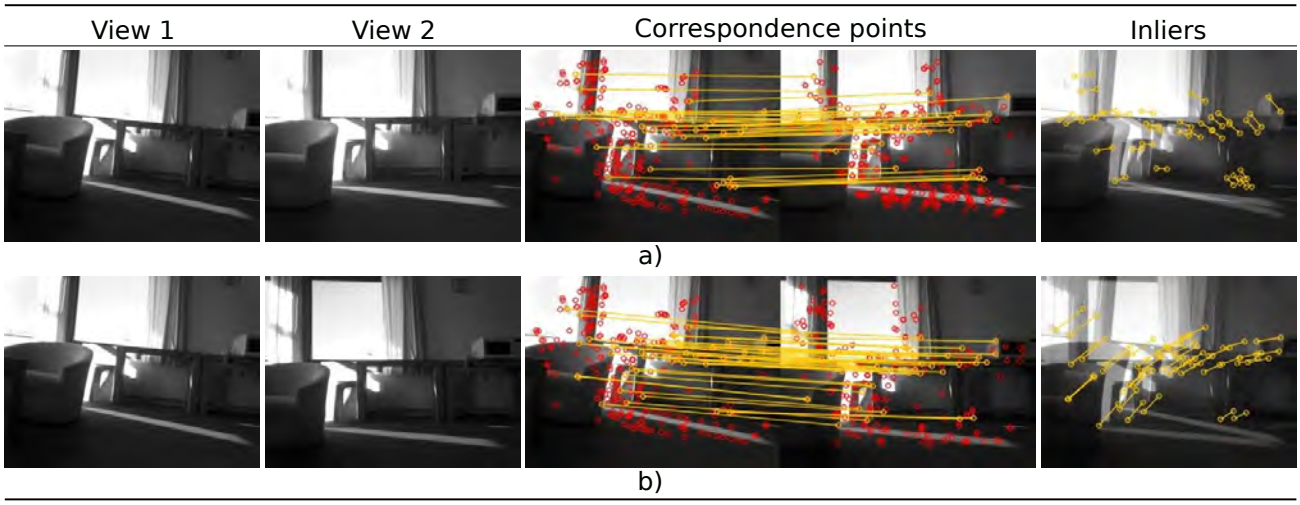

Figure 3.2: Results for the fundamental-matrix estimation from test images a) Room_1 and b) Room_2. The first and second columns show the first and second view respectively. The third column shows the correspondence points along with outliers contained in the dataset. The fourth column depicts a blended image of the two views with inliers found by our method.

To quantitatively test all methods and compare results, two values are studied: the inlier number IN, and the error E_r. We now describe these values as used in the related work [84,86]. The inlier number IN is used to express the number of detected inliers within the set I_s. The E_r error, on the other hand, is used to provide a quantitative measure for the quality of the estimated geometric relation. E_r is assessed from the standard deviation of only the inliers. Thus, the error E_r is computed as follows:

$$E_r = \left(\sum_i \frac{e_i^2}{IN} \right)^{1/2}, \qquad i \in I_s, \tag{3.3}$$

where e_i^2 is the quadratic error produced by the ith inlier. The term e_i^2 corresponds to EF_i^2 or EH_i^2 as described in Chapter 2, and represent the errors produced by the ith inlier considering the final fundamental matrix $\boldsymbol{F}$ or homography $\boldsymbol{H}$, respectively.

Quantitative results are shown in Table 3.4 and Table 3.5. Table 3.4 exposes the number of inliers and the error E_r for all methods used in images 3.2 and 3.3. Table 3.5, on the other hand, shows the results given by the homography for the images in Figure 3.4.

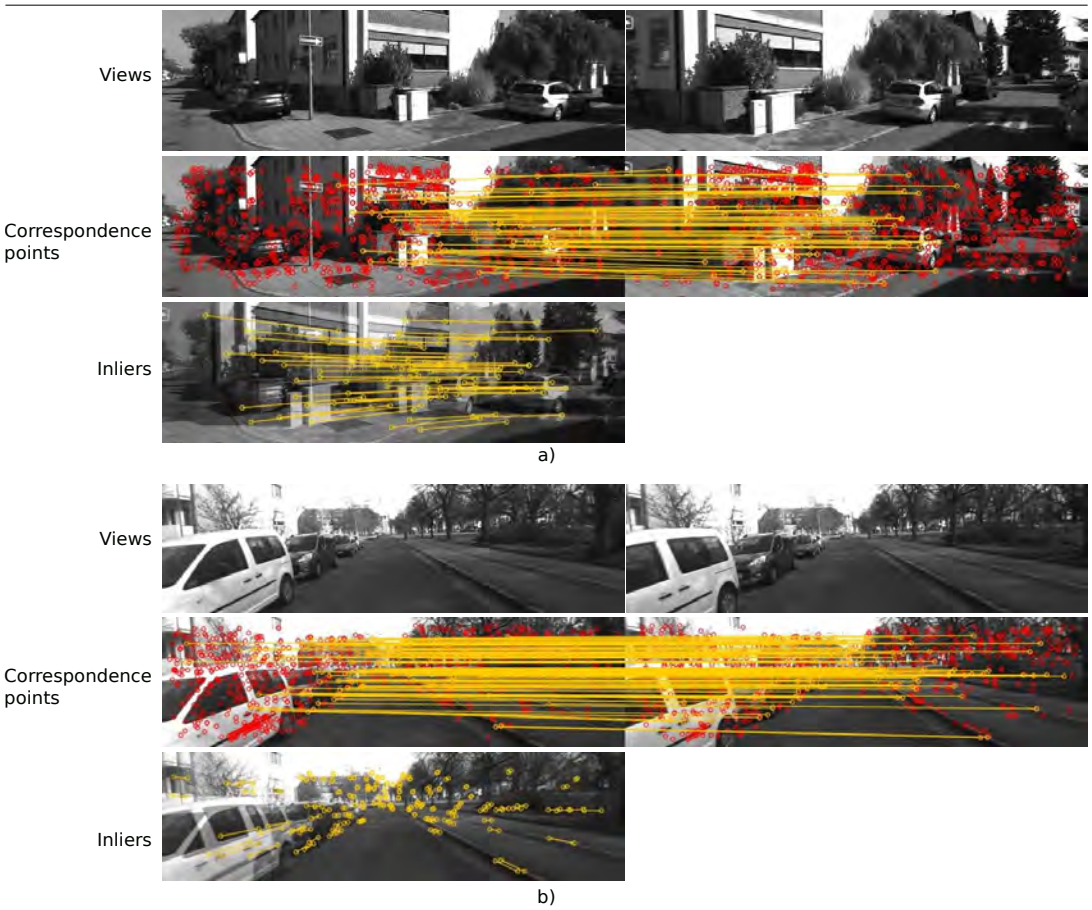

Figure 3.3: Results for the fundamental-matrix estimation from test images a) Street_1 and b) Street_2. The first row shows the first and second view. The second row shows the correspondence points along with outliers contained in the dataset. The third row depicts a blended image of the two views with inliers found by our method.

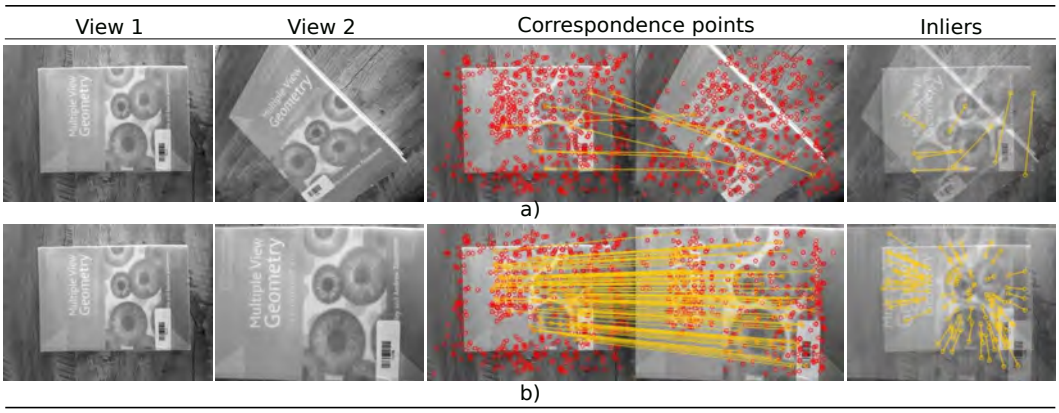

Figure 3.4: Results for the homography estimation from the test images a) Book_1 and b) Book_2. The first and second columns show the first and second view respectively. The third column shows the correspondence points along with outliers contained in the dataset. The fourth column depicts a blended image of the two views with inliers found by our method. Ground-truth position of cameras is not available for this experiment.

Table 3.4: Number of inliers IN and error E_r for all methods, considering the test images in Figures 3.2 and 3.3

	TLBO-based		GA-based		DE-based		CSA-based		RANSAC		MLESAC	
Image	IN	E_r error	IN	E_r error	IN	E_r error	IN	E_r error	IN	E_r error	IN	E_r error
Room_1	47	0.86	42	0.85	45	1.12	47	0.78	46	2.80	43	2.36
Room_2	42	0.75	45	1.24	45	1.03	45	1.29	43	2.45	39	1.76
Street_1	64	0.61	59	0.91	58	0.83	62	1.03	62	1.75	67	1.85
Street_2	127	0.91	115	1.01	110	1.24	110	1.52	98	2.54	95	2.67

Table 3.5: Number of inliers IN and error E_r for all methods, considering the test images in Figure 3.4

	TLBO-based		GA-based		DE-based		CSA-based		RANSAC		MLESAC	
Image	IN	E_r error	IN	E_r error	IN	E_r error	IN	E_r error	IN	E_r error	IN	E_r error
Book_1	8	0.25	8	0.78	8	0.85	8	0.64	8	1.068	8	1.84
Book_2	71	0.42	73	0.63	71	0.51	71	0.73	68	1.07	73	1.36

3.4 Discussions

This work has presented a novel application of the TLBO algorithm to the task of estimation of multiple view relations, i.e. the fundamental matrix and homography. We presented a method for the robust estimation of the epipolar geometry from point correspondences. Rather than relying over a pure random selection as it is the case of the RANSAC algorithm, our approach additionally uses the TLBO algorithm to improve the search. Our method adopts a different sampling strategy in comparison with RANSAC to generate putative solutions. Guided by the TLBO algorithm, our proposed approach builds iteratively new potential solutions considering the quality of the solutions that have been generated by previous candidate individuals (solutions). Likewise, we presented an accurate objective function to certainly evaluate the quality of a candidate model (solution). One result of this approach is a substantial reduction of the number of iterations (in comparison with RANSAC), yet maintaining the robustness capability of RANSAC.

Chapter 4

Vanishing Points Estimation Considering Metaheuristic Approaches

4.1 Introduction

The objective of this chapter is to describe our proposed approach for estimating image Vanishing Points (VPs). Detecting VPs has many applications regarding robotic navigation, camera calibration, image understanding, visual measurement, 3D reconstruction, among others. Similar to the estimation of the epipolar geometry, different methods for detecting vanishing points employ a heuristic approach such as RANSAC; however, others relies on accumulator space techniques. Nevertheless, these type of methods suffer from low accuracy or high computational cost. In this chapter, we continue to exploring metaheuristic approaches for computer vision problems; specifically, VP estimation. Thus, for improving the efficiency of vanishing point estimation, we use the TLBO algorithm (TLBO). In our proposed approach, after a line segment detection, the TLBO algorithm is used to cluster line segments according to their more optimal vanishing point. Therefore, our algorithm detects both orthogonal and nonorthogonal vanishing points. To validate the performance of our proposed approach, tests and comparison with other approaches were carried out. The results verify the efficiency and accuracy of our proposed algorithm.

The rest of the chapter is organized as follows. Section 4.2 depicts the TLBO-based VP detector. Section 4.3, on its part, shows the experimental results. Finally, Section 4.4 states the conclusions and future work.

4.2 TLBO Algorithm for Vanishing Point Detection

In this section, we describe our proposed method for VP detection. Our proposed approach aims to have good accuracy, to detect both orthogonal and nonorthogonal vanishing points, and to overcome the limitations associated with the classical RANSAC-like clustering algorithms. To attain these objectives, a TLBO-based detector is proposed. Since our VP detector implements a metaheuristic, we describe in this section the following elements: the search space organization, the individual representation, and the objective-function definition.

4.2.1 Search Space

We construct the search space following the next ideas: i) line segments contribute no information for the VP detection if they are considered too short; ii) line segments need to be sorted according to their directions since a VP depends only on the direction of a line, not on its position [1]. The TLBO method constructs better students (solutions) in each iteration by moving them towards the position of the best one. Thus, ordering line segments according to their directions allows our algorithm to perform better.

To construct the search space, we execute the following operations. First, we perform edge extraction from the gray-scale input image to later extract line segments. Then, lines shorter than 5% of the diagonal length of the input image are removed. Finally, the remaining line segments are sorted according to its directions from 0 to π. Thus, a sorted set of line segments is generated: $E = \{\xi_1, \xi_2, \cdots, \xi_{N_E}\}$, being N_E the total number of the remaining line segments. The line segment ξ_i is defined by its two endpoints $p_{i,1}$ and $p_{i,2}$ in homogeneous coordinates. Also, its corresponding inclined angle θ_i is stored.

The result of the above process is the set E of sorted edges long enough to be significant. We now describe how an individual represents a solution within this search space.

4.2.2 Individual Representation

As already stated, in this work we treat vanishing point estimation as an optimisation problem where line segments are to be clustered. Therefore, we use a TLBO-based method to solve this optimisation problem. In the context of clustering, a valid line cluster belongs to a VP. Thus, a single student X_i represents the N_c cluster centroids (VPs). That is, each student is constructed as follows:

$$X_i = \{(p_j, q_j) \mid \quad j = \{1, 2, \cdots, N_c\}\}, \tag{4.1}$$

where p_j and q_j are index positions of the set E for the segments ξ_{p_j} and ξ_{q_j}, respectively. The set E generated as in Subsection 4.2.1 encodes the search space.

For the student (solution) X_i as defined in Equation 4.1, the indexes p_j and q_j compute a candidate VP $vp_{i,j}$ that refers to the j-th cluster centroid of the i-th student. To define $vp_{i,j}$, the two segments ξ_{p_j} and ξ_{q_j} are randomly selected from E. Then $vp_{i,j}$ (in homogeneous coordinates) is determined by the cross-product of the lines (in homogeneous coordinates) passing the segments. Therefore, a student represents a number of candidate clustering for the current line segments. The fitness of a student is easily measured as the quantization error. Once the whole initial population is initialised, the objective function guides the search to a set of optimal VPs.

4.2.3 Deviation Function and Objective Function

To define the objective function, we first define a deviation function.

Deviation function

Before computing the objective function, the deviation function needs to be calculated. The deviation function aims to quantify the deviation that exists between a VP $vp_{i,j} \in X_i$ and a line segment $\xi_k \in E$. To this purpose, as propose in [87], our algorithm considers the orientation error, this is also consistently with the measuring method proposed in [98, 99]. Thereby, the deviation $d(\xi_k, vp_{i,j})$ is computed with the absolute value of the sine of the angle formed by the line l_k that passes through the segment ξ_k, and the line $\hat{l}_k$ that connects $vp_{i,j}$ with the center of l_k. To compute the deviation function, both the VP $vp_{i,j}$ and the line segment ξ_k need to be expressed in homogeneous coordinates. Remember that the segment ξ_k is defined by its two endpoints $p_{k,1}$ and $p_{k,2}$ in homogeneous coordinates. Therefore the line l_k passing the segment ξ_k is found by:

$$l_k = p_{k,1} \times p_{k,2} \ , \tag{4.2}$$

while $\hat{l}_k$, on the other hand, can be computed by:

$$\hat{l}_k = vp_{i,j} \times \frac{1}{2}(p_{k,1} + p_{k,2}). \tag{4.3}$$

Finally, the deviation $d(\xi_k, vp_{i,j})$ is calculated as follows:

$$d(\xi_k, vp_{i,j}) = \left| \frac{-e_2\hat{e}_1 + e_1\hat{e}_2}{\sqrt{e_1^2 + e_2^2}\sqrt{\hat{e}_1^2 + \hat{e}_2^2}} \right|, \tag{4.4}$$

where $l_k = (e_1, e_2, e_3)^T$ and $\hat{l}_k = (\hat{e_1}, \hat{e_2}, \hat{e_3})^T$ are the homogeneous forms of l_k and $\hat{l}_k$ respectively.

In Eq. 4.4, the absolute value is used, since only the relative deviation between the orientations of the two lines is of interest, and not its sign. Thus, the deviation angle is restricted between 0 and π.

To identify outliers, a distance threshold Th_c is specified. Thereby, a line segment is identified

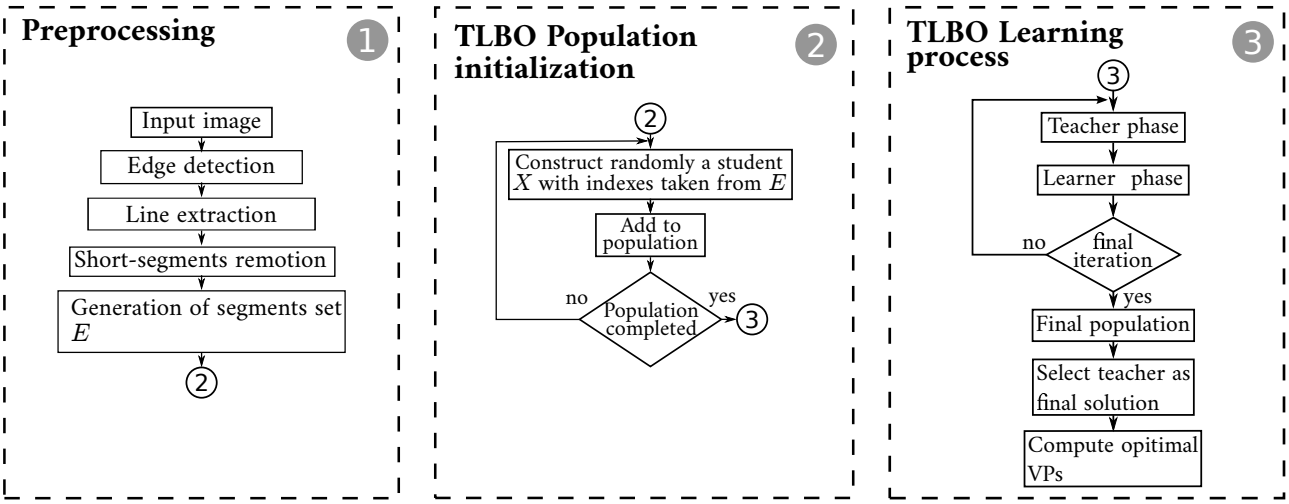

Figure 4.1: Summary of the propose metaheuristic-based vanishing point detection

as an outliers if the segment is separated by more than Th_c from all the tentatives VPs.

Objective function

According to the deviation criterion, the line segments can be partitioned into different clusters. The objective function is used to quantify the quality of the partitioning. Quality is given by the compactness and separation measures. We search for high compactness and low separation. High compactness means that the line segments present within the same cluster share a high degree of similarity, which means in the context of this work that the probability of having a common VP is high. On the other hand, a high separation indicates that the line segments present in different clusters are dissimilar, or that the probability of sharing a common VP is low. We measure similarity of the clustering by:

$$f_i = \frac{1}{N_c} \sum_{j=1}^{N_c} \sum_{l_k \in E} d(l_k, vp_{i,j}). \tag{4.5}$$

The objective function uses the Eq. 4.5. Thereby, the solution encoded by the student X_i is evaluated by the following equation:

$$f(X_i) = 1 - \frac{1}{1 - f_i}. \tag{4.6}$$

4.2.4 TLBO for Vanishing Points Detection

The whole VP detection algorithm is summarized in Fig. 4.1. First, a preprocessing step is carried out. The line segments detection is executed and short segments are removed from the search space. Then, the remaining segments are sorted according to its inclination. This last process contribute to a higher probability that neighboring edges belong to a common VP. The second main step is the initialisation of the TLBO population as described in Subsection 4.2.2. Then, the TLBO process iterates, and line clustering is improved guided by the objective function. Finally, and optimal set of VPs is computed for clusters with more than two inliers since two lines is the minimum information to compute a VP.

VP Estimation

In the final step, our proposed approach refine the VP locations. After the line segments have been clustered based on an estimated VP, we proceed to compute optimal VPs. To this purpose, for each cluster C_j we minimise the sum of the distances of the lines to an optimal VP vp_o.

$$vp_o = \arg\min_{p} \sum_{\xi_k \in C_j} d(\xi_i, p). \tag{4.7}$$

We perform such minimisation by a least-squares algorithm.

4.3 Experimental Results

In this section, we report results of the application of the proposed VP detector in this section. Experiments were carried out on real images commonly used in the literature. Also, comparisons were carried out with two different approaches: The work in [36], where line detection is done by Canny detector and flood fill, and VP estimation by the J-linkage algorithm [48]; a more recent VP detection algorithm [100] that computes line detection with PClines [101] and VP estimation as in [102] with a posterior refinement. For each method, we use the source code

as provided by its authors. All the experiments shown in this section were executed on a 2.80GHz Intel Core i7-7700HQ CPU, with a C++ implementation of the proposed algorithm. To perform line extraction and edge detection, subroutines from OpenCV were first utilized. The set of parameters for the TLBO algorithm are shown in Table 4.1.

Table 4.1: Parameters setup for the TLBO-based circle detector

Parameter	Value
N_c	10
Iterations	80
Population size	30

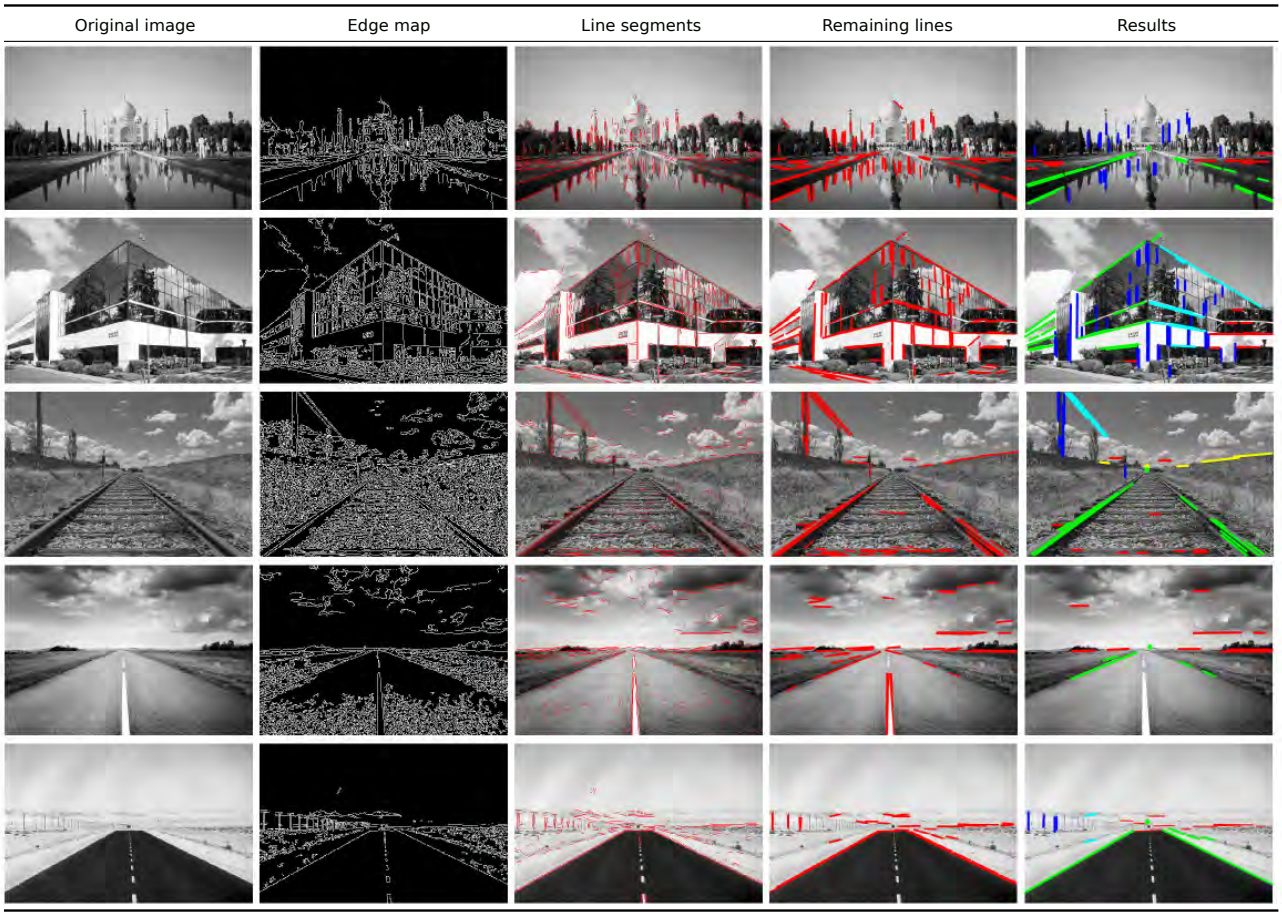

Figure 4.2: The proposed TLBO-based VP detector estimates both orthogonal and nonorthogonal VPs. In the first column, the original image is shown. The second column depicts the edge map. The third column shows the line segments detected. Fourth column indicates the remaining line segments after the preprocessing step. Finally, the fifth column shows the results of our approach.

4.3.1 Image performance

The capability of our proposed approach to detect both orthogonal an nonorthogonal Vps is shown in Fig. 4.2. In the figure, the original image, the edge map, the detected line segments, the remaining line segments after the preprocessing step, and the results of our approach are shown.

In the results shown in Fig. 4.2, we differentiate by color the found line segments clustered according to their most common VPs. It is noted that our approach detect both orthogonal and nonorthogonal vanishing points. Thus, the results in Fig. 4.2 show all line classifications for both nonorthogonal and the three Manhattan orthogonal directions.

4.3.2 Performance evaluation

To evaluate the performance of our proposed method, we compute quantitative errors. We estimate the focal length error, and the angular error. First, we define both errors, and then we compare the results. For this particular experiment, we use images form the York Urban database [2], since ground truth is available.

Focal error length

The authors of the York Urban database provide the intrinsic parameters of the camera: focal length$f$, pixel dimension (m_x, m_y), principal point (p_x, p_y), and skew factor ς. With this information, we can construct the camera intrinsic matrix:

$$\boldsymbol{K} = \begin{bmatrix} f/m_x & \varsigma & p_x \\ 0 & f/m_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4.8}$$

The focal length error measures the accuracy of the focal length that is estimated with the vanishing points found by our proposed approach. We compute the internal parameters using

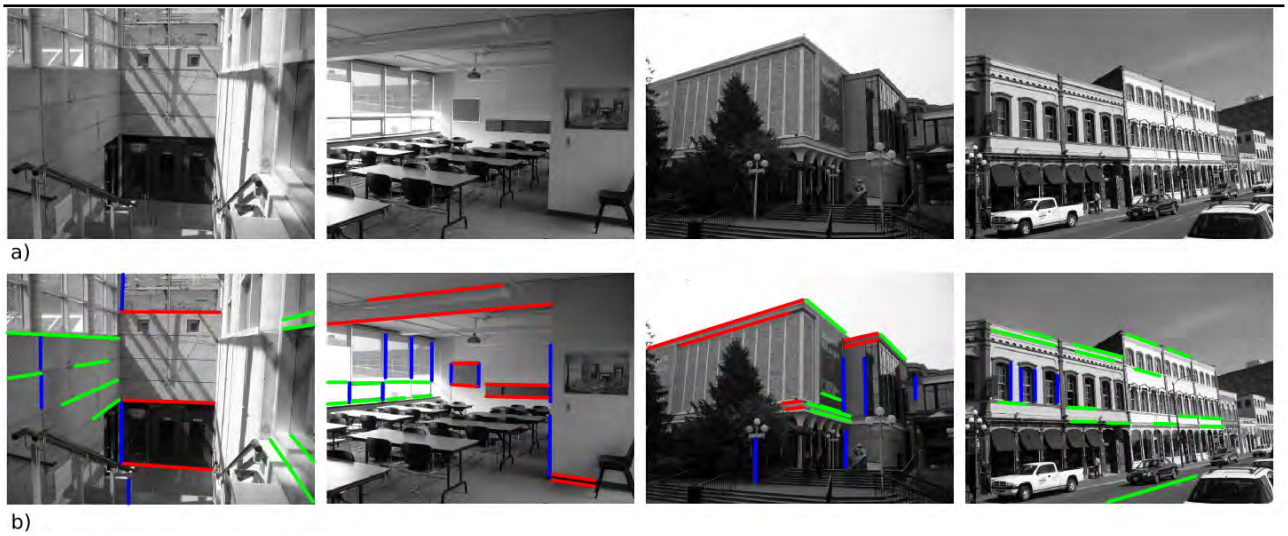

Figure 4.3: Results from the proposed TLBO-based detector in images from the York Urban dataset [2]. a) Original images. b) line segments in the Manhattan directions found by our proposed algorithm.

the linear solution in [1].

Angular error

The accuracy of the detected VPs is also measured with the angular deviation of the detected VPs. To compute the error, we select the three VPs for the Manhattan directions. Then, the 2D coordinates of the VPs are mapped into their corresponding 3D directions using the focal length given by the ground-truth. Finally, we compare the computed directions with the ones given by the ground-truth; their angular errors were measured. To select the orthogonal VP triplet, we use the method in [36].

The focal length error and the angular error are used to obtain quantitative information regarding the accuracy of all the VP detectors. The focal length and angular errors are a score metric to quantify their performance. Figure 4.3 shows some images form the York Urban data set and the corresponding line clusters found by our algorithm. The dataset include ground truth information of lines in Manhattan directions. Thus, in Figure 4.3, only lines segments corresponding to Manhattan directions found by the proposed TLBO-based detector are shown.

To present the results, the cumulative histograms of both the focal error length and the angular

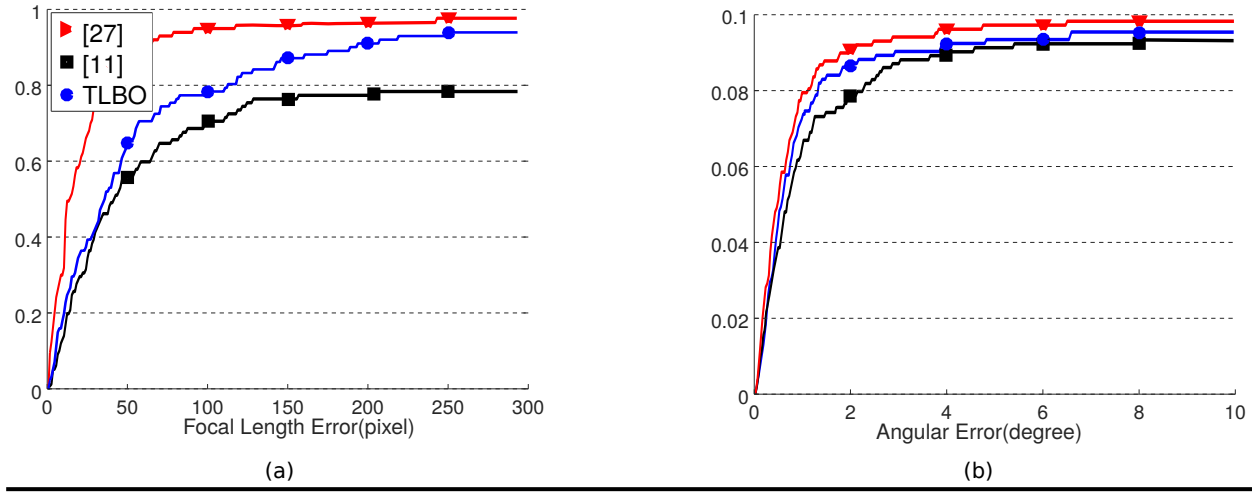

Figure 4.4: Quantitative analysis. a) Focal error length in pixels, b) Angular error in degrees

Table 4.2: Processing time for all approaches

Computational time	TLBO-based	[100]	[36]
min (seg)	0.8	0.3	1.2
mean (seg)	1.4	0.7	1.3
max (seg)	1.7	1.1	1.5

error, as described above, are shown in Figure 4.4. In the figure, it is shown by the results that the TLBO-based VP estimator can be considered as a good alternative to line clustering. The computation time for all approaches is shown in Table 4.2.

From the results, both qualitative and quantitative, it is shown that the estimator based on the TLBO algorithm can be consider as an alternative to line clustering for VP detection. While the work in [36] obtains lightly more accuracy, but it is also the method with the highest computational time. On the contrary, the work in [100] is fast, but not precise. Our TLBO-based detector is a good alternative for balancing between accuracy and computational time.

4.4 Discussions

In this chapter, we propose a metaheuristic VP-detector based on a population-based method: the TLBO algorithm. VP detection is a novel task for this metaheuristic. Our proposed technique differs from RANSAC-like methods in the way that our algorithm uses additional

information provided by previous candidate solutions, rather than a pure random search. To this purpose, a more appropriate objective function was utilised to guide the exploration of the search space.

When compared with other methods, the results obtained by our TLBO-based estimator required less effort, since there are no algorithm-specific parameters to tune. This is clear advantage in comparison to other metaheuristic techniques, and even to RANSAC algorithms. We used the focal length error and angular deviation error as score metrics to evaluate the mismatch between ground-truth line segments in Manhattan directions and the detected segments by the proposed method.

Our proposed method for VP estimation accurately detects vanishing points in real images. This is shown from the results given by comparing the errors between the TLBO-estimated VPs and the ground-truth VPs. Regarding future work, we will make efforts to explore the use of our method to computer vision problems that need vanishing point estimation.

Chapter 5

Circular Markers Extraction Considering Metaheuristic Approaches

5.1 Introduction

Circle extraction is usually a previous task used in different applications related to biometrics, robotics, medical image analysis among others. For sparse 3D reconstruction, circular markers might be used as unambiguous features for triangulation or bundle adjustment processes. Solutions based on meta-heuristic approaches, such as evolutionary and swarm-based algorithms, have been adopted in order to overcome the main deficiencies of Hough Transform methods. In this work, the task of circle detection is presented as an optimization problem, where each circle represents an optimum within the feasible search space. To this end, a circle detection method is proposed based on the Teaching Learning Based Optimization algorithm, which is a population-based technique that is inspired by the teaching and learning processes. Additionally, improvements to the evolutionary approach for circle detection are obtained by exploiting gradient information for the construction of the search space and the definition of the objective function. To validate the efficacy of the proposed circle detector, several tests using noisy and complex images as input were carried out, and the results compared with different approaches for circle detection.

The rest of the chapter has been organized as follows. Section 5.2 depicts our proposed approach for circle extraction. Section 5.3, on its part, present the results obtained by our method. Finally, Section 5.4 states the discussion of the work presented in this chapter.

5.2 Circle Detection Using the TLBO Algoritm

In this section, we describe the proposed approach for circle extraction. Our method seeks different objectives: To produce a few or no false detections; to work with real images; to detect multiple circles; and to have good accuracy and high detection rate in real images with low contrast and occluded circles, cluttered textures, and blurring boundaries.

In order to achieve these objectives, the TLBO algorithm is used, and gradient information is exploited. Since our detector is based on a meta-heuristic, we describe, as usual, the following elements: The search space organization, the individual representation, and the objective-function definition. Further, we explain the use of gradient orientation for the particular case of circle detection.

5.2.1 Search Space

Regarding the organization of the search space, in previous works [90–93,95], edge extraction is first performed, to then organize the search space in a set where all edgel locations are indexed one after the other as the edgels are visited in the edge map from left to right and top to bottom. A different approach involves storing edgel locations randomly within the set [103]. These organizations, as stated in [104], do not ensure that nearby pixels in the edge map are neighbors in the search space. This is an important issue to our approach, since the TLBO algorithm locates better individuals in each iteration by shifting them towards the position of the best one. Thus, neighboring individuals within the search space are expected to belong to the same circle and have similar fitness value. Consequently, we perform a different procedure to enhance the search space for the TLBO algorithm. This improvement is achieved by reducing

the size of the search space and by customizing its organization, so that its shape better suits the geometric meaning of the TLBO algorithm.

For the search space definition, we use gradient information extracted from image derivatives. First, image noise is reduced by using a 3×3 Gaussian filter. Then, edge extraction is performed in order to obtain one-pixel-wide edges. Image derivatives in horizontal (g_x) and vertical (g_y) directions are performed to extract gradient information, both magnitude and orientation. Gradient magnitude $M(x, y)$ is computed by:

$$M(x, y) = \sqrt{g_x(x, y)^2 + g_y(x, y)^2}, \tag{5.1}$$

and gradient orientation $O(x, y)$ is obtained by:

$$O(x, y) = atan2\big(g_y(x, y), g_x(x, y)\big). \tag{5.2}$$

Where $atan2(\cdot)$ is the function that returns the value of the arc tangent of g_y/g_x in radians; the function takes into account the sign of both arguments in order to determine the quadrant.

To perform the image derivatives g_x and g_y, we convolve the image with derivative kernels. To reduce the error in the gradient computation, the 3×3 Sobel operators are used since the accuracy of these operators can be up to 0.12 radians in noiseless conditions [105, 106].

We use gradient magnitude to achieve one of the objectives of the search space enhancement, which is to reduce the search space. This reduction is attained in two steps. Within the first step, all pixels with gradient magnitude lower than a threshold T_m are removed. The threshold T_m is computed as follows:

$$T_m = \tau \cdot M_{mean} \tag{5.3}$$

where M_{mean} is the mean magnitude value for the image as computed by Eq. 5.1, and $\tau \in [0, 1]$

is the factor that controls the distinctiveness of the magnitude values. When τ is near 1, more pixels are discarded. Conversely, when τ is near 0, less pixels are discarded. Finally, within the second step, the remaining pixels are connected into contours using the border-following algorithm that constructs the adjacency tree of the image [107]. Then, all the contours that are considered too small (i.e. below a threshold T_c pixels) to form a significant part of a circle are removed. To define the threshold T_c, we define a minimum radius r_{min} and compute T_c as follows:

$$T_c = ceiling(2\pi r_{min}). \tag{5.4}$$

In consequence of the previous steps, all edgels with small magnitude are removed as well as the short linked contours. Hence, the result is a smaller edge map organized into connected contours that are sufficiently large to be part of a circle. Also, gradient orientation of the remaining edgels is available. Figure 5.1 summarizes the operations of the preprocessing steps for the proposed approach.

The search space is finally organized in a set V_c with edgel locations stored together and ordered according to their respective contour c. Hence, the search space is encoded within the set $V_c = \{c_1, c_2, \cdots, c_n\}$ with n the number of linked contours, and $c_i = \{p_{i_1}(x_{i_1}, y_{i_1}), \cdots, p_{i_{m_i}}(x_{i_{m_i}}, y_{i_{m_i}})\}$ with m_i the number of edgels connected in the i-contour. One result of this organization is an increase of the probability that nearby edgels in the search space belong to the same circle within the edge map, since connected edgels are stored together.

5.2.2 Individual Representation

Each student X of the population encodes a candidate circle C that passes through three different edgels. Thus, to define student X, three different elements are randomly chosen from V_c. Hence, $X = \{i, j, k\}$, with i, j and k being the index positions in the search space encoded in V_c for the edgel positions $p_i(x_i, y_i)$, $p_j(x_j, y_j)$ and $p_k(x_k, y_k)$ in the edge map, respectively.

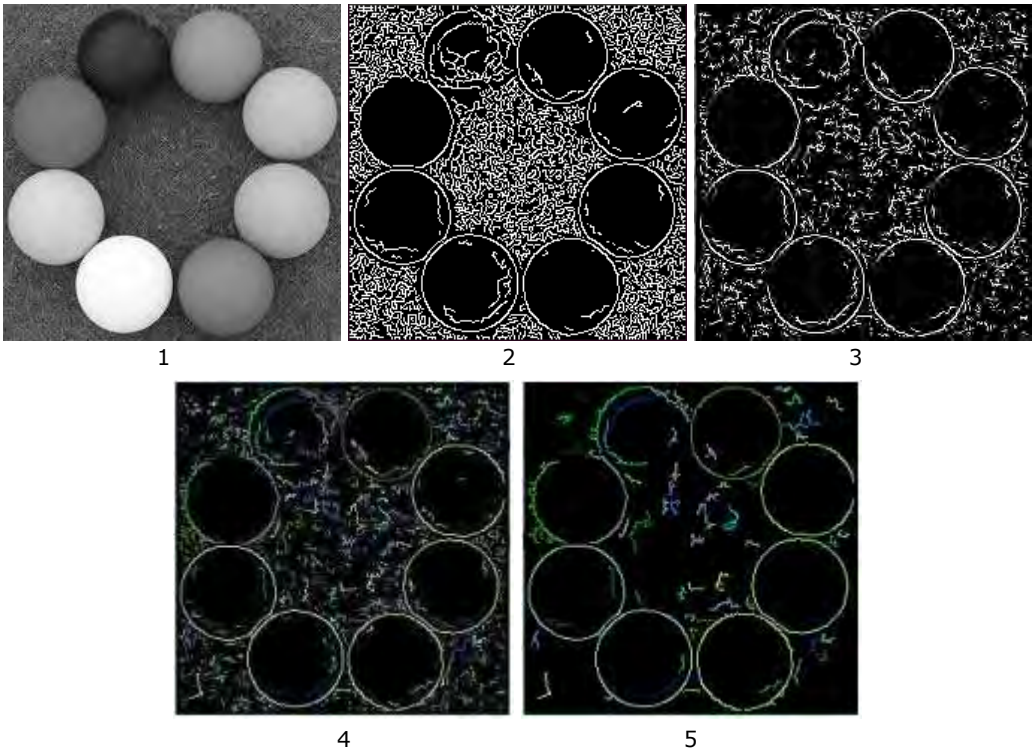

Figure 5.1: Preprocessing performed to reduce the search space and customize its organization. A gray-scale image (1) as input is processed to extract one-pixel-width edges as shown in 2. Then, edgels (edge pixels) with low gradient magnitude are removed; the result is the image in 3. Then, contours are obtained by connecting the remaining edgels as depicted in 4. However, short contours are removed since they are not considered to be part of the search space. Thereby, the input data to the proposed approach, as shown in 5, is organized into connected contours large enough to be considered as part of a circle

From student X, the center (x_0, y_0) and the radius r of a candidate circle C are calculated by the following equations:

$$x_0 = \frac{\begin{vmatrix} x_j^2 + y_j^2 - (x_j^2 + y_i^2) & 2(y_j - y_i) \\ x_k^2 + y_k^2 - (x_i^2 + y_i^2) & 2(y_k - y_i) \end{vmatrix}}{4((x_j - x_i)(y_k - y_i) - (x_k - x_i)(y_j - y_i))} , \tag{5.5}$$

$$y_0 = \frac{\begin{vmatrix} 2(x_j - x_i) & x_j^2 + y_j^2 - (x_j^2 + y_i^2) \\ 2(x_k - x_i) & x_k^2 + y_k^2 - (x_i^2 + y_i^2) \end{vmatrix}}{4((x_j - x_i)(y_k - y_i) - (x_k - x_i)(y_j - y_i))} , \tag{5.6}$$

and

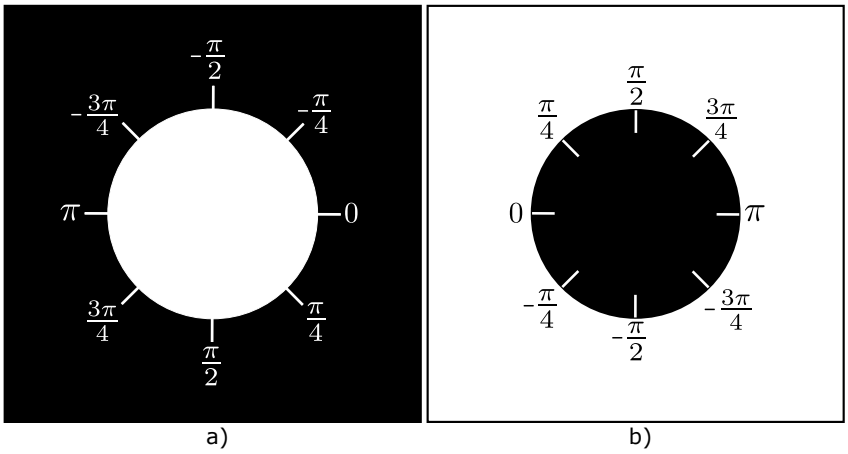

Figure 5.2: Gradient orientation inconsistency problem. Two different orientation patterns are obtained from image derivatives for: a) A circle brighter than the background, and b) A circle darker than the background

$$r = \sqrt{(x_0 - x_d)^2 + (y_0 - y_d)^2} \ , \tag{5.7}$$

where $d \in \{i, j, k\}$. Therefore, a candidate circle $C(x_0, y_0, r)$ can be represented by V_c indexes i, j, and k belonging to the student $X = \{i, j, k\}$, using Equations 5.5 through 5.7.

5.2.3 Objective Function

The last element to define is the objective function that guides the search. Usually, reported meta-heuristic approaches for circle detection only use edgel locations in the objective function [90–93, 95]. Conversely, our proposed method improve results by using additional data provided by the gradient orientation. However, the pattern of the gradient orientations within a circumference is not the same for a circle with less intensity than the background and a circle with more intensity than the background. This results in a gradient orientation inconsistency problem that needs to be tackled to reliably use gradient orientation in the definition of the objective function.

Orientation alignment

The gradient orientation inconsistency problem is shown in Figure 5.2. This problem entails that for the same circle parameters (x_0, y_0, r), one pixel orientation, say $\pi/2$, may belong to a concave segment as in Fig. 5.2 a); or to a convex segment as in Fig. 5.2 b). In other words, the circle parameters encoded by a student X are not enough information to assume whether the orientation vectors should point towards or against the center of the circle.

To solve the orientation inconsistency problem, the work in [67] uses a set of four 9×9 masks to verify whether an edgel belongs to a concave or convex segment. A different mask is used in different sections of the same circumference. Another used strategy involves the computation of second derivatives in order to calculate the contour curvature and decide whether a curve is concave or convex, consequently finding the correct orientation pattern [64]. These techniques are computational expensive or sensitive to noise. Hence, in order to reduce the computational burden, the proposed method partially verifies orientation alignment as now explained.

Instead of verifying whether a pixel belongs to a concave or convex segment, the proposed strategy validates the gradient orientation of a pixel within a circumference *regardless* it lies on a concave or a convex segment. In regard to Figure 5.2, it is clear that for a circle C with parameters (x_0, y_0, r), the pixel $p_1(x_0+r, y_0)$ within the perimeter of C has gradient orientation of either 0 as in Fig. 5.2 a), or π as in Fig. 5.2 b). Likewise, the pixel $p_2(x_0, y_0 + r)$ has an orientation of $\pi/2$ as in Fig. 5.2 a), or $-\pi/2$ as in Fig. 5.2 b). For all pixels in the perimeter of C, the absolute difference of the two facing valid orientations is always π. We use this fact to verify orientation alignment.

Given the circle parameters (x_0, y_0, r) encoded by a student X, two valid patterns of gradient orientations can be computed. Taking the case of Fig. 5.2 a) for convenience, it is noted that for each pixel $p_i(x_i, y_i)$ within the circumference, the *ideal* gradient orientation $\overline{O}_i$ can be computed by:

$$\overline{O}_i = atan2\big(y_i - y_0, x_i - x_0\big). \tag{5.8}$$

Thus, the proposed algorithm computes the degree of orientation alignment of the *actual* orientation O_i of the edgel $p_i(x_i, y_i)$ w.r.t the *ideal* orientation $\overline{O}_i$ using the following metric:

$$M(O_i) = |cos(|O_i - \overline{O}_i|)|. \tag{5.9}$$

This metric approximates to 1 whether the real gradient vector of p_i points towards or against the center of the circle, as long as its orientation approximates to either one of the two valid orientations of the concave and convex cases. On the other hand, the metric approaches 0 as the real gradient vector becomes perpendicular to either of the two valid orientations. According with this metric, the proposed algorithm classifies orientation alignment into three categories:

$$O_i \text{ alignment} = \begin{cases} \theta\text{-aligned} & \text{if } M(O_i) > cos(\theta) \\ \gamma\text{-aligned} & \text{else if } M(O_i) > cos(\gamma) \\ \text{not aligned} & \text{otherwise} \end{cases} \tag{5.10}$$

Within this classification, the orientation O_i is θ-aligned when it deviates less than θ radians from either of the two valid orientations. Similarly, the orientation O_i is γ-aligned when it deviates less than γ radians (but more than θ) from either of the two valid orientations. An illustration of the three classifications is shown in Figure 5.3.

The gradient orientation pattern is preserved even in the presence of noise. In Figure 5.4, the error between an ideal and a real orientation pattern is shown. To compute the error, the circle parameters are manually detected. Then, a valid gradient orientation $\overline{O}_i$ is computed for each edgel within the circumference. The real gradient orientation O_i, on the other hand, is obtained for the same edgels with derivative kernels. As shown in the figure, for this particular image with different circles of different intensities, the majority of edgels are θ-aligned even in the presence of different levels of Gaussian noise.

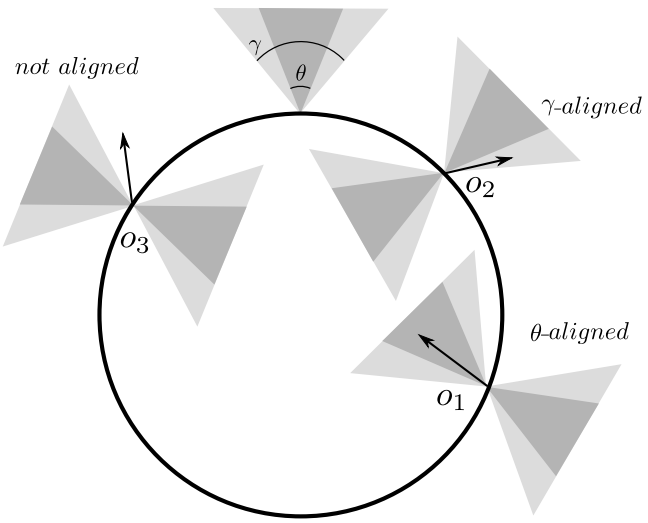

Figure 5.3: Proposed orientation alignment classification. The proposed approach categorizes the orientation alignment of each edgel (edge pixel) into three different classifications according to their gradient orientation

Objective-function definition

Having defined the orientation alignment, we now describe how it is used in the objective function computation. A well fitted student is the one that encodes a candidate circle that actually exists within the edge map and has a valid gradient orientation pattern. To score a solution, reported meta-heuristic approaches for circle detection do not use gradient information and proceed as follows. For a candidate circle C, a certain amount of test points in the perimeter of C is generated. The value of the classical objective function, $F_{classical}$, is the ratio of the total number of test points appearing in the edge map to the total number of test points. Variations to this conventional objective function include different penalizations as the edge points recede from the test circle. Unfortunately, when in noisy conditions, the quantity of false positive circles may increase using this approach due to the fact that small circumferences may contain enough pixels in the edge map to give a high ratio. For this reason, in the proposed method, additional information provided by the gradient orientation is used in order to surmount the problem.

Given a candidate circle $C(x_0, y_0, r)$ encoded by a student X, N_c test triplets $T = \{t_1(x_1, y_1, \overline{O}_1), \cdots, t_{N_c}(x_{N_c}, y_{N_c}, \overline{O}_{N_c})\}$ are generated. In the interest of reducing the computational burden and improving the sub-pixel accuracy, the locations of test points are generated

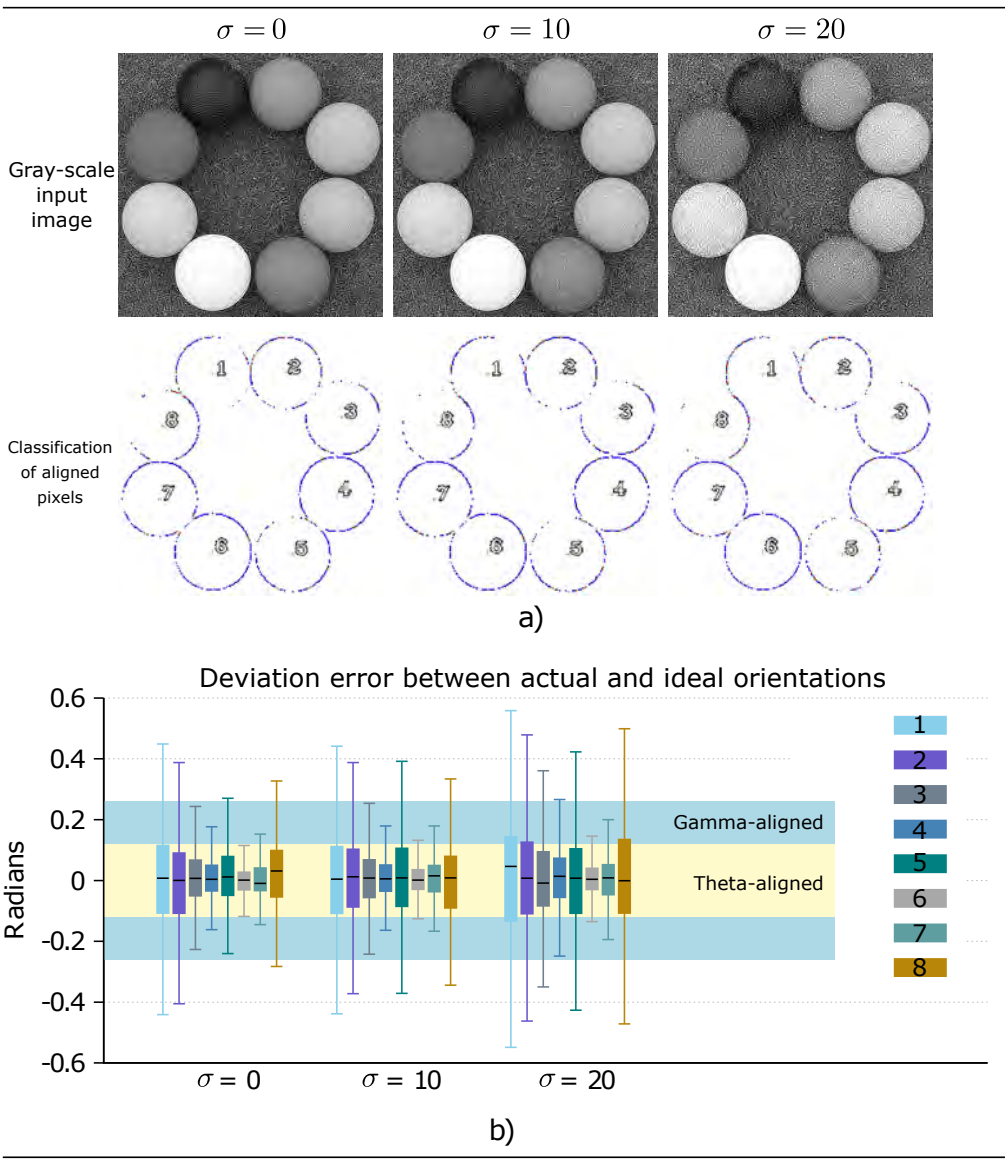

Figure 5.4: Orientation alignment test in real images. a) Original image corrupted with different levels of Gaussian noise. theta-aligned edgels are shown in blue, while gamma-aligned edgels in red. b) Boxplots showing the orientation alignment error of pixels within the circumference of each circle

using the Midpoint Circle Algorithm (MCA) [108]. The MCA aims to minimize the error between the pixel discrete positions and the continuous candidate circumference. Additionally, it reduces the computational burden of the algorithm by only computing pixel positions within the first octant of the circumference and mirroring the rest. Gradient orientation of each test point, on the other hand, is computed using Equation 5.8. The MCA is described in more detailed in the next subsection, then the objective function is defined.

Midpoint Circle Algorithm

The MCA is a established algorithm used to determine the required points to draw a circle in a digital image. The algorithm entries are the parameters of the circle, i.e. the centre (x_0, y_0) and radius r. The MCA aims to minimize the error between the pixel discrete positions and the continuous candidate circumference. Additionally, it reduces the computational burden of the algorithm by only computing pixel positions within the first octant of the circumference and mirroring the rest. For these reasons, the MCA is considered as the quickest providing a subpixel precision.

The MCA starts at point $(x_0 + r, 0)$ and proceeds upwards-left by using integer additions and subtractions. Let p_i be a pixel within the first octant, i.e. within the part of the circumference starting at $(x_0 + r, 0)$ and ending at (x, y) with $x < y$. If the pixel coordinates of p_i are (x_k, y_k), the coordinates of the next pixel p_{i+1} to be chosen are either (x_k, y_{k+1}) or (x_{k-1}, y_{k+1}). The selection is conducted as:

$$p_{i+1} = \begin{cases} (x_k, y_{k+1}) & \text{if } (x_{k\text{-}0.5}\text{-}x_0)^2 + (y_{k+1}\text{-}y_0)^2\text{-}r \leq 0 \\ (x_{k\text{-}1}, y_{k+1}) & \text{otherwise} \end{cases} \tag{5.11}$$

The above selection is performed iteratively until the coordinates of the chosen pixel satisfy $y > x$, which indicates that the whole first octant has been visited. The MCA process is outlined in Figure 5.5.

Proposed Objective Function

Candidate test points within the perimeter of C (found with MCA) must exist in the edge map, and their gradient orientation must be aligned. Thus, the proposed fitness function is computed by:

$$F_{proposed}(X) = \frac{\sum_{i=1}^{N_c} E(x_i, y_i, \overline{O}_i)}{N_c}, \tag{5.12}$$

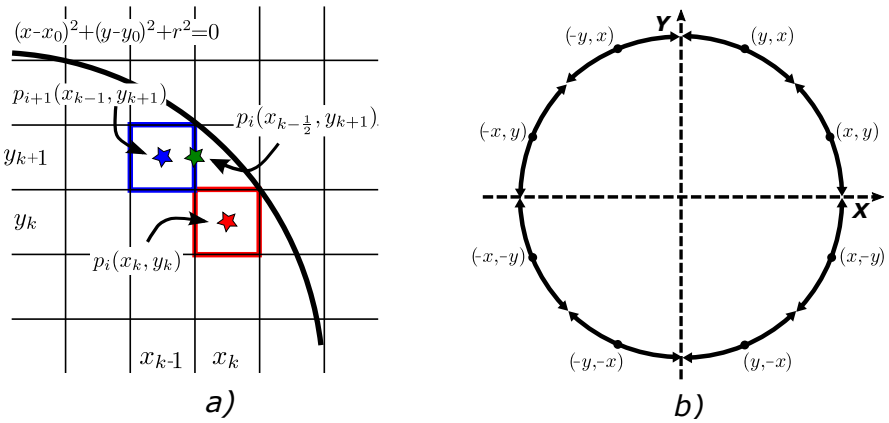

Figure 5.5: To generate the test points needed to evaluate the objective function, the Midpoint Circle Algorithm (MCA) is performed. a) Given a pixel $p_i(x_k, y_k)$ (colored in red), the only two possible points for the next chosen pixel p_{i+1} are: i) the upper-left and ii) the upper-middle neighbouring pixels. Noting that the midpoint $M_p(x_{k\text{-}1/2}, y_{k+1})$ (colored in green) lies inside the candidate circle, the selected pixel is the upper-left neighbour (colored in blue), b) To reduce computational burden, every point (x, y) in the first octant is mirrored on the other seven.

where $E(\cdot)$ is the function that verifies the pixel existence in $p_i(x_i, y_i)$, and its orientation alignment O_i w.r.t $\overline{O}_i$. E returns zero when the test point $p_i(x_i, y_i)$ is not an edgel. On the other hand, when the pixel p_i belongs to the edge map, E returns:

$$E(x_i, y_i, \overline{O}_i) = \begin{cases} \varphi_1 \cdot M(O_i) & \text{if } \theta\text{-aligned} \\ \varphi_2 \cdot M(O_i) & \text{if } \gamma\text{-aligned} \\ \kappa & \text{if not aligned,} \end{cases} \tag{5.13}$$

where $M(O_i)$ is the metric for orientation alignment in Equation 5.9, and the weights φ_1, φ_2 and κ can be set according to the desired accuracy. We set and fixed the weights empirically to: $\varphi_1 = 1$, $\varphi_2 = 0.5$ and $\kappa = 0.25$; and the threshold angles to $\theta = 0.12$ and $\gamma = 0.26$ radians. Different values modify the relevance of orientation alignment within the search process.

The example of Figure 5.6 illustrates the evaluation of two test points p_{T1} and p_{T2} belonging to a circle encoded by student $X = \{i, j, k\}$. Since p_{T1} is θ-aligned, P_{T1} contributes to $F_{proposed}$ by a factor of $\varphi_1 M(o_{T1})$. On the contrary, since the test point P_{T2} is neither θ-aligned nor γ-aligned, it only contributes by a factor of κ. Recall that both P_{T1} and P_{T2} belong to the edge map, however P_{T1} meets the additional orientation test.

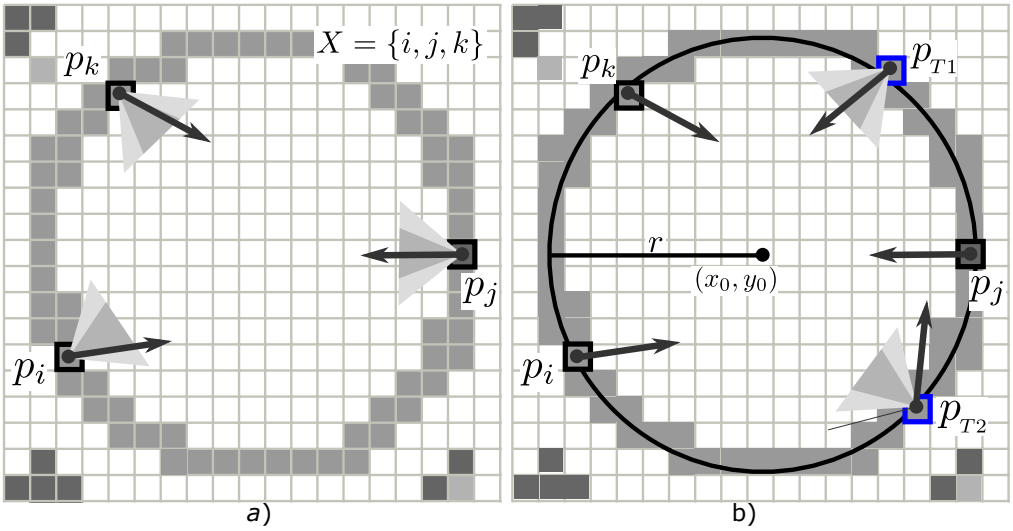

Figure 5.6: Fitness evaluation. a) A candidate solution $X = \{i, j, k\}$ represents a candidate circle $C(x_0, y_0, r)$. b) Two pixels p_{T1} and p_{T2} within the perimeter of C are evaluated. Both pixels contribute to the fitness of X. However, p_{T1} contributes to a greater extent than p_{T2}, since the former meets the additional orientation test

5.2.4 TLBO for Circular-Marker Extraction

The whole circle detection algorithm is summarized in Fig. 5.7. We build on our published work to demonstrate that the proposed circle detection method is accurate enough to be suggested in fiducial marker detection like the ones shown in [109]. First, the preprocessing step is performed in order to obtain the edge map, gradient information, and the search space encoded in V_c. Then, the TLBO algorithm is executed. The first step of the TLBO algorithm involves generating the initial population. All the solutions within the first population must consist of three non-collinear points. To achieve this, the algorithm first verifies if the denominator in Equations 5.5 or 5.6 is different from zero. If this is not the case, at least two of the points are collinear, and that particular combination of points is not added to the population. Later in the process, in the Teacher an Learner phases, the algorithm prevents individuals consisting in collinear points by assigning a fitness equal to zero if the denominator in Equations 5.5 or 5.6 is equal to zero. At the end, the teacher of the last iteration of the TLBO algorithm is selected as the solution.

Our proposed method is able to achieve multiple-circle detection. To this end, the proposed method is executed as many times as needed. After one execution, the detected circle is

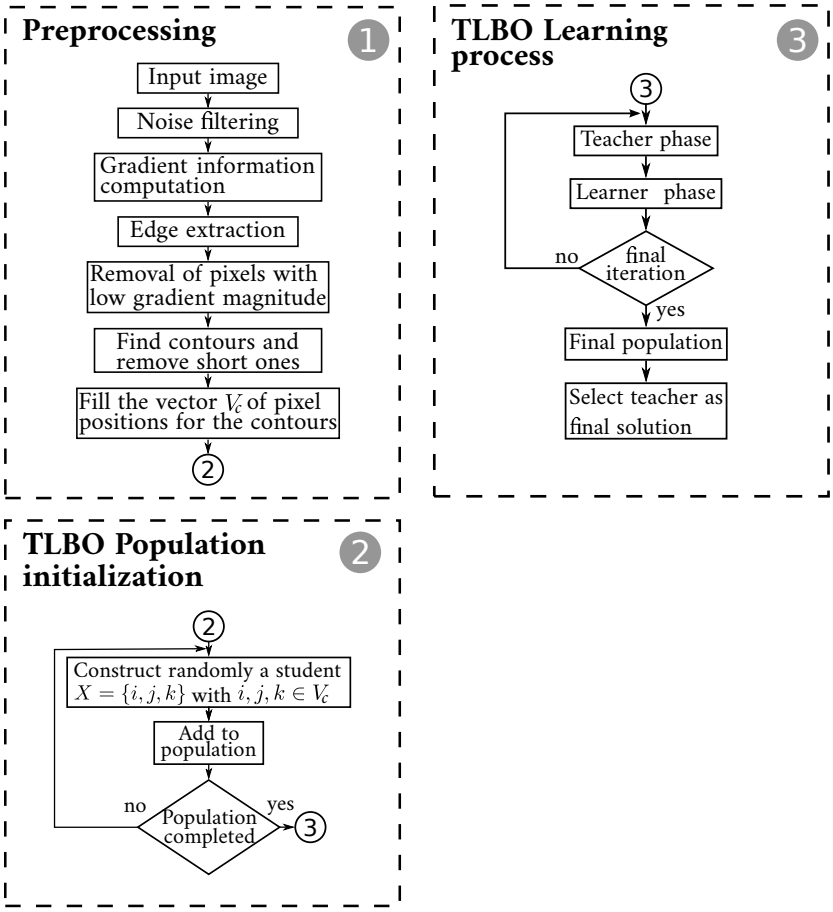

Figure 5.7: General procedure of the proposed meta-heuristic-based circle detector

removed from the search space and the edge map. Then, the process is carried out again over the modified search space and edge map. Multiple executions of the whole process are carried out until the algorithm returns a final solution whose fitness value is below a predefined threshold f_{min}. Finally, every detected circle is validated by analyzing the continuity of the detected circumference segments as in [90].

5.3 Experimental Results

In this section, results of the application of the proposed method on real images are reported, and compared with five different approaches that also use gradient information: 1) The OpenCV implementation of a HT-based circle detector; 2) The isophotes RCD algorithm with the same parameters as in [68]; 3) The on-line demo of EDCircles provided by the authors; 4) the Matlab implementation of CACD provided by the authors; and 5) The classical DE algorithm

Table 5.1: Parameter setup for the DE-based circle detector

Parameter	Value
Number of epochs	200
Population size	50
Differential weight	0.25
Crossover probability	0.80

with the same search space and objective function as the TLBO algorithm. We have selected these methodologies in the interest of contrasting our method with other approaches that also use gradient information, while at the same time are based on different methodologies. The OpenCV implementation is a one-to-many CHT, while the CACD is a one-to-one CHT. Isophotes RCD, on its part, is based on an iterative sampling methodology. EDCircles is an arc-based circle detector; and the DE-based circle detector relies on a meta-heuristic.

The experimental setup includes the use of real images commonly used in the literature: Streetlight (544×509), Plates (504×489), Bowling (467×480), Gobang (234×231), Cells (519×459), Cell (701×664), Watch (467×480), Cookies (295×292), Eye (190×143), and Fiducial (600×600). All the experiments were executed on a 2.80GHz Intel Core i7-7700HQ CPU, with a C++ implementation of the algorithm. To perform edge extraction and border following, subroutines from OpenCV were first utilized.

5.3.1 Parameter Setup

The sets of parameters for the meta-heuristics, shown in Table 5.1 and Table 5.2, have been chosen empirically in favor of performing a fair comparison. First, the settings for the TLBO algorithm were experimentally defined. Then, the parameters of the DE algorithm that control the number of objective function evaluations (FEs) (See Chapter 2) were set specifically to equate this number for both approaches. Finally, the rest of the DE parameters were set after experimentation. On the other hand, the thresholds for the preprocessing step and the multiple-circle detection process were fixed at $\tau = 0.05$, $T_c = 2\pi \cdot 5$ pixels, and $f_{min} = 0.20$, respectively. The parameters and thresholds for both meta-heuristics remained fixed for every experiment.

Table 5.2: Parameter setup for the TLBO-based circle detector

Parameter	Value
Iterations	100
Population size	50

We now describe results related to the convergence of the algorithm, since our approach is based on a meta-heuristic. Then, in the following subsections, we discuss qualitative and quantitative results.

5.3.2 Convergence Analysis

The evolution of the best individual, the mean fitness value and the similarity of the population are examined for every generation. To compute a measure of similarity S for a population P, the student $X = \{i, j, k\}$ is considered as a vector. Thus, the L2-norm of student X is $|X| = \sqrt{i^2 + j^2 + k^2}$. To calculate S, we first compute the L2-norm of each student within P. Then, each output value is divided by the maximum L2-norm value within P. The output of the last computation is a set of values between 0 and 1. We thus define S as the standard deviation of these values. For convenience, S is plotted along with the mean fitness value and the best individual in Figure 5.8.

Within the convergence analysis, the TLBO and DE algorithms optimize over two different representations of the search space: 1) Edgels locations are indexed in a set V one after the other as they are visited within the edge map from left to right and top to bottom; and 2) Edgels locations are indexed in a set V_c with linked contours as proposed in Subsection 5.2.1. Further, to guide the search, two different objective functions are used: 1) The classical objective function $F_{clasical}$ that only considers edgel locations; and 2) The objective function $F_{proposed}$ as described in Subsection 5.2.3.

Results demonstrate that the proposed combination of an objective function that considers gradient orientation, an adequate search space, and the TLBO algorithm, results in an efficient circle detector. First, we discuss the performance of the proposed objective function. Figure 5.8 shows the detected circles after one execution of each meta-heuristic. From the figure, it

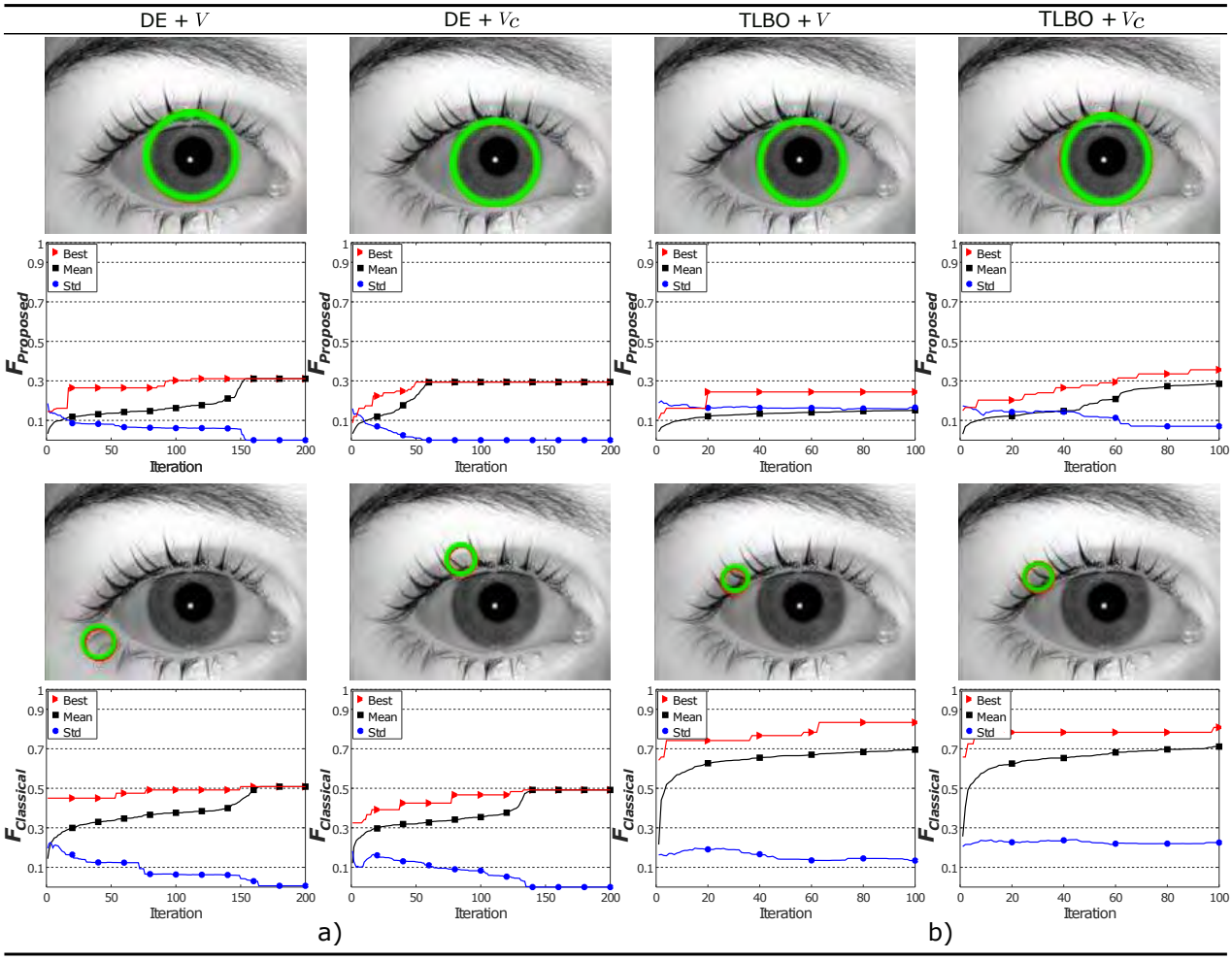

Figure 5.8: Comparison between different combinations of meta-heuristic, search-space organization, and objective function. a) Obtained results with DE optimizing over different search-space organizations and guided by different objective functions. b) Results for the TLBO algorithm optimizing over different search-space organizations and guided by different objective functions

is clear that when $F_{classical}$ is used, false positives are detected because they have high fitness value. Conversely, when guided by the proposed objective function, a true circle is detected.

To better explain why our proposed objective function guides more efficiently the search, Figure 5.9 depicts that, oppositely to $F_{clasical}$, $F_{proposed}$ clearly differentiates good from bad solutions, since it scores better a true circle than a false positive circle. The last statement is true due to the fact that $F_{proposed}$ does not only verifies edgels existence in the perimeter of a candidate circumference, but gradient orientation as well.

By additionally organizing the search space into linked contours, the fitness of the final solution is better, and the TLBO tends to converge. The graphs in Figure 5.8 b) show for the TLBO

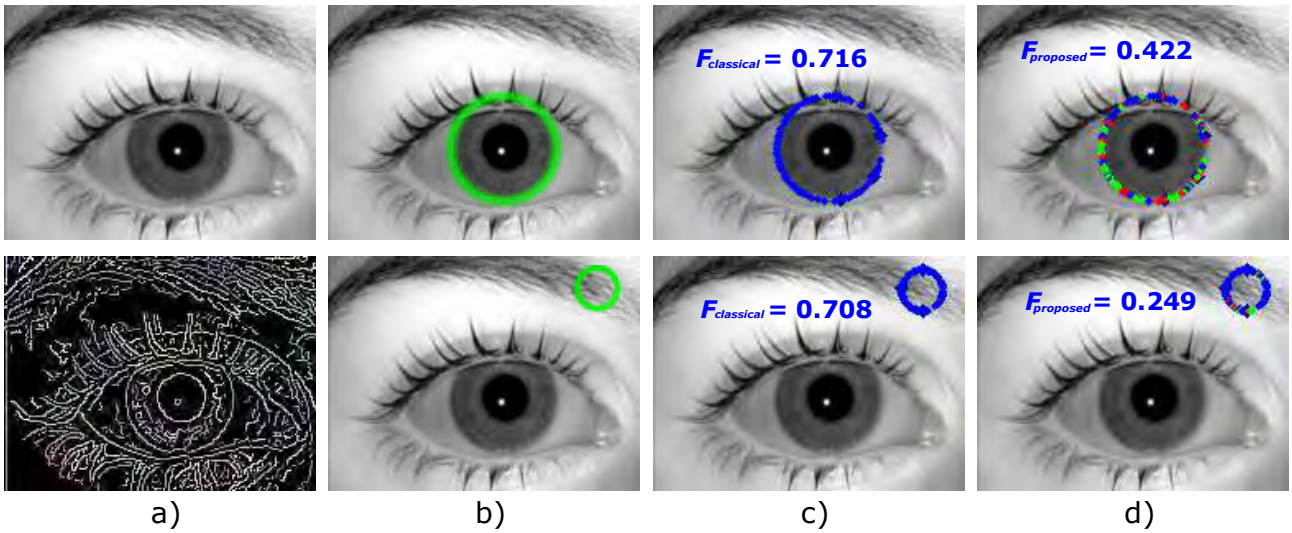

Figure 5.9: Comparison between objective functions. a) Test image and linked contours. b) Two different candidate solutions are to be scored. c) Fitness values computed with the classical objective function. d) Fitness values obtained with the proposed objective function. $F_{proposed}$ truly differentiates good from bad solutions by additionally evaluating orientation alignment (θ-aligned, green; γ-aligned, red; and not aligned, blue), and not only the existence of edgels within the edge map like $F_{classical}$

algorithm that, when optimizing over the unorganized search space V and guided by either of the two objective functions, the search is random, and in consequence the best solution does not evolve. This is true since the similarity of the population remained almost equal throughout the iterations. Conversely, it is only when using the proposed search space representation V_c along with the proposed objective function $F_{proposed}$, that a clear evolution and convergence of the population is seen. As a result, the final solution has a higher fitness because the detected circle fits better a true circle. In other words, more edgels with a valid orientation pattern are truly present within the perimeter of the detected circle. Another advantage of the proposed search space representation is that it permits to obtain accuracy while keeping a simple coding.

It is clear that optimizing over V_c and guided by $F_{proposed}$, good accuracy is obtained. Furthermore, by also utilizing the TLBO algorithm, results are improved in comparison with the DE approach. We discuss the advantages of the TLBO algorithm by analyzing its exploration and exploitation behavior. In comparison with DE, TLBO explores more the search space as shown by the similarity of the populations in Fig. 5.8. The DE approach usually converges faster (the similarity becomes zero); this means that the DE algorithm stops searching. Differently, the

TLBO algorithm makes more exploration of the search space, since there are more variability within the population while keeping the best solution. By allowing more iterations to DE, no changes in the results would exist. To change the exploration and exploitation capabilities of the DE approach, a different setting of parameters is needed. Conversely, the TLBO algorithm does not require the tunning of algorithm-specific parameters to modify its convergence behavior.

5.3.3 Image Performance

To evaluate the qualitative performance of the TLBO-based detector, several tests were carried out regarding different tasks involving:

1. Circle localization.
2. Multiple circle detection.
3. Blurred and low contrast circle detection.
4. Circular approximation.
5. Circle detection in presence of Gaussian noise.

The relevance of such tasks comes from the fact that they are commonly found in typical computer vision applications.

Results shown in Figures 5.10 and 5.11 demonstrate that the proposed circle detector achieves efficiently every task. Our circle detector found all circles present in the test images despite cluttered textures, blurred and low-contrast circles, occluded circles, and Gaussian noise.

To test the capabilities of the proposed method for the first task, image Streetlight is used. For this task, the TLBO-based detector outperforms the OpenCV implementation and EDCircles. The OpenCV implementation does not fit accurately the circle present in the image; while EDCircles falsely detects a circular shape in the background.

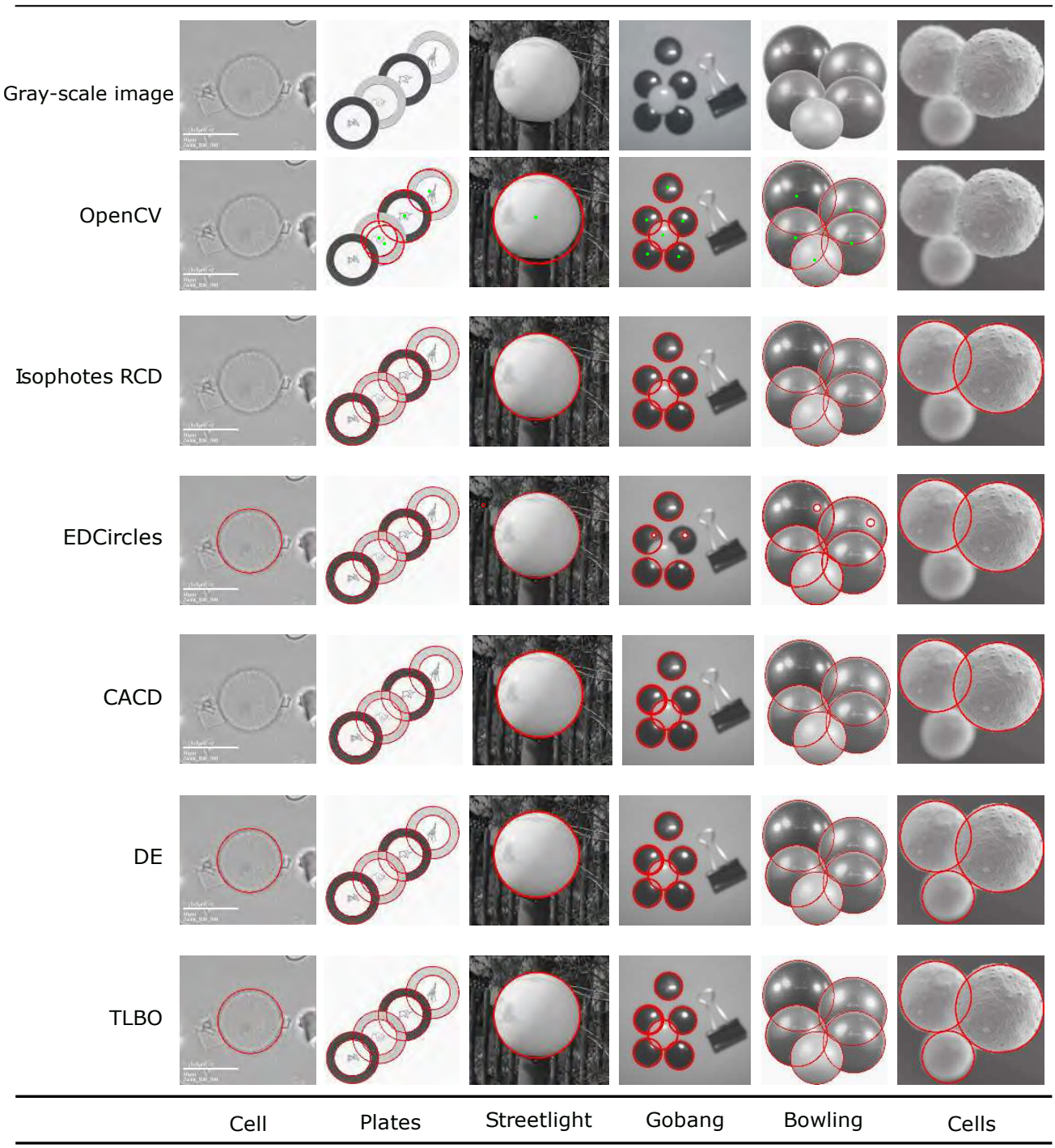

Figure 5.10: Performance of the six methods for the tasks of circle localization with cluttered textures (column 3), multiple-circle detection (columns 2, 3-5), and blurred or low-contrast circle detection (columns 1, 6)

Our method also detects multiple and concentric circles, as shown by the results of the second task. Images Plates, Bowling and Gobang show a better performance of our method in comparison with the Opencv implementation and EDCircles. For instance, EDCircles detects reflections as circles in Bowling and Gobang; while the Opencv implementation does not detect

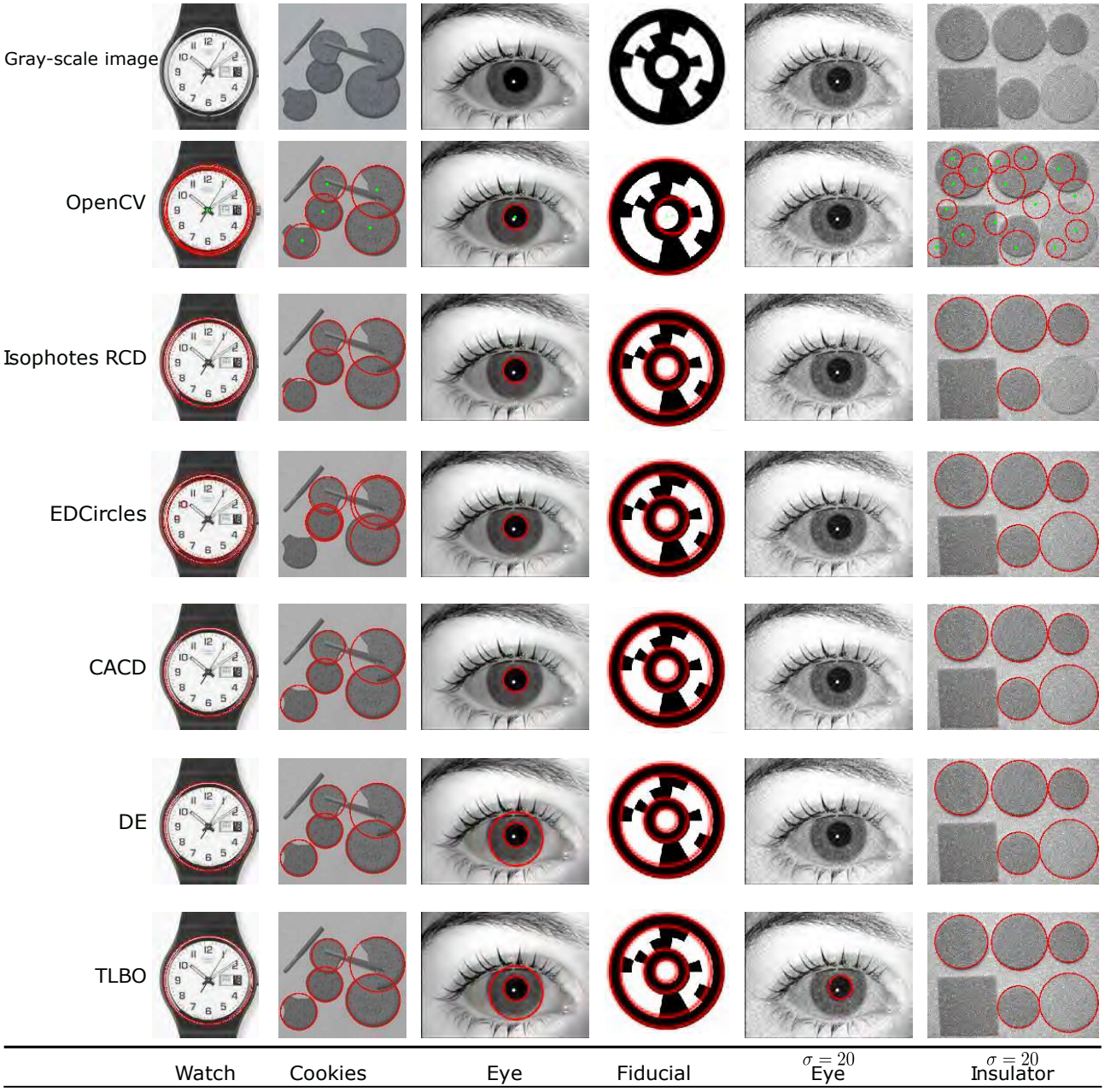

Figure 5.11: Performance of the six methods in regard to the tasks of circular approximation (columns 1-3), circular fiducial extraction (column 4) and circle detection under Gaussian noise (column 5-6)

all circles in Plates. Regarding the task of dealing with blurred and low-contrast circle detection, images Cells and Cell are presented. For this task, the TLBO-based detector outperforms the the Opencv implementation, and the CACD algorithm.

For the rest of the tasks, figure 5.11 shows the results involving circular approximation and noisy images. In comparison with other approaches, the proposed detector obtains better

results. For instance, the proposed detector outperforms EDCircles in images Watch, Cookies and Eye. EDCircles tends to classify as circles small curves like the number “0” and “9” in image Watch. On the other hand, the Isophotes RCD approach is outperformed by our method in images Eye and Insulator with Gaussian noise.

It is noted that the DE algorithm has a similar performance to the TLBO for the all images, except the image Eye. This is due to the fact that they share the search space and objective function. However, quantitative results described in the next subsection show that the TLBO algorithm obtained better quantitative accuracy.

5.3.4 Performance Evaluation

To obtain quantitative information of the accuracy of all detectors, a score metric is used to quantify their performance. Results from the methods are compared to a manually detected ground-truth circle using this score metric. Further, to compute a statistical analysis, the meta-heuristic-based circle detectors were executed 100 times for each test image.

The metric to evaluate a quantitative performance looks at the percent overlap of the circle area. The overlap error E_o is calculated as:

$$E_o = 1 - \frac{\overline{C_1 \cap C_2}}{\overline{C_1 \cup C_2}}, \tag{5.14}$$

where the second term is the overlap percentage, and is the area of the overlap between circle 1, C_1 and circle 2, C_2, divided by the area of the union of the circles. For multiple circle evaluation, the average value of the E_o errors, M_{E_o}, is calculated by:

$$M_{E_o} = \left(\frac{1}{NC}\right) \cdot \sum_{i=1}^{NC} E_{o_i}, \tag{5.15}$$

where NC is the total number of detected circles in the test image.

For the meta-heuristic approaches, we also compute a success rate S_r as now explained. Given an specific image, let C_{GT} be the set of ground truth circles, and C_D the set of circles detected by the method. For each circle in C_D, we find the match circle in C_{GT} subject to $E_o < 0.20$. Only one circle in C_D is accepted as a valid match per ground truth circle. Then, all the accepted circles are considered as true positives and stored in a set C_{TP}. Finally, we compute the success rate S_r as follows:

$$S_r = \frac{card(C_{TP})}{max(card(C_{GT}), card(C_D))}, \tag{5.16}$$

where $card(\cdot)$ returns the cardinality of a set. It is noted that S_r is equal to 1 when all the ground truth circles are found and no false positives are detected. Differently, when all ground truth circles are found, but false positives are detected, S_r decreases. The more false positives are detected or the less ground truth circles are found, the more S_r decreases and approximates to 0. Also, S_r is equal to 0 when there are no found circles by the method that match the ground truth circles. S_r penalizes even multiple detections of circles with similar centers and radii.

Table 5.3: Average time, average E_o error, and success rate for the meta-heuristic methods, considering the test images in Figures 5.10 and 5.11. Bold values indicate the best result for a particular experiment

	DE			TLBO		
Image	Time	E_o eror	Succes rate	Time	E_o error	Succes rate
Streetlight	0.420	0.107	**100**	**0.396**	**0.088**	**100**
Plates	**1.260**	**0.073**	95.87	1.589	0.084	**96.55**
Bowling	0.527	**0.173**	95.32	**0.423**	0.179	**97.80**
Gobang	**0.443**	0.147	90.33	0.557	**0.062**	**92.35**
Cells	**0.794**	0.182	78.33	0.812	**0.175**	**72.66**
Cell	0.359	0.112	59	**0.318**	**0.098**	**65**
Watch	0.540	**0.091**	**97.50**	**0.417**	0.152	96.20
Cookies	0.473	0.057	95.60	**0.460**	**0.032**	**98.33**
Eye	0.276	0.191	76	**0.250**	**0.105**	**89.50**
Eye ($\sigma = 20$)	0.352	-	-	**0.367**	**0.112**	**43**
Fiducial	0.708	0.173	99.25	**0.98**	**0.044**	**100**
Insulator($\sigma = 20$)	0.887	**0.187**	90.20	**0.792**	0.193	**92.35**

Table 5.4: Time and E_o error for the non-meta-heuristic methods, considering the test images in Figures 5.10 and 5.11

	OpenCV		Isophotes RCD		EDCircles	CACD	
Image	Time	E_o eror	Time	E_o error	E_o error	Time	E_o error
Streetlight	0.368	0.086	0.625	0.083	0.052	1.907	0.031
Plates	0.927	0.092	2.837	0.076	0.083	3.015	0.068
Bowling	1.017	0.318	1.605	0.152	0.065	2.216	0.105
Gobang	0.757	0.235	1.092	0.198	0.071	2.291	0.117
Cells	1.158	-	1.027	0.106	0.073	2.972	0.087
Cell	0.709	-	0.968	-	0.026	2.397	-
Watch	1.415	0.173	1.072	0.194	0.091	2.737	0.112
Cookies	1.157	0.025	2.142	0.177	0.124	2.471	0.089
Eye	0.831	0.104	1.793	0.052	0.087	1.916	0.062
Eye ($\sigma = 20$)	1.751	-	2.611	-	-	3.041	-
Fiducial	0.72	0.061	1.892	0.071	0.083	1.972	0.082
Insulator($\sigma = 20$)	12.365	-	2.407	0.098	0.019	3.973	0.091

The succes rate S_r allows us to obtain a measure of the performance of the algorithm to detect all ground truth circles and discard false positives. The error E_o, on the other hand, when applied only to circles in C_{TP}, allow us to obtain a measure of the accuracy of the detected true-positives circles.

Quantitative results are shown in Tables 5.3 and 5.4. Table 5.3 exposes the average time, average error, and success rate for the meta-heuristic approaches. Table 5.4, on its part, shows the time and error for the non-meta-heuristic approaches. In the case of EDCircles, computational time is omitted since the on-line demo provided by the authors was used.

5.4 Discussions

In this section, we presented a meta-heuristic circle detector based on the TLBO algorithm. The proposed method differs from previous meta-heuristic approaches in the way that it uses additional information provided by the gradient to customize the search space, and to better guide the search with a novel objective function.

In comparison with other approaches, the results obtained by the TLBO-based detector required less effort from the user because there are less parameters to tune for this particular meta-

heuristic. This is an advantage in comparison to other meta-heuristic approaches, and even to the OpenCV CHT and the isophotes RCD algorithms that also require the set of various parameters. To test the performance of the proposed approach, computation time and accuracy have been compared with five different approaches based on different principles: The OpenCV CHT, the isophotes RCD algorithm, EDCircles, CACD, and a meta-heuristic based on DE. We used a score metric to evaluate the mismatch between a ground-truth circle and the detected circle. Also, since our proposed circle detector is based on a meta-heuristic, an evaluation of the convergence of the algorithm was performed.

Our approach accurately detects circles in real images despite the presence of cluttered textures, circle occlusions, blurred effects, and Gaussian noise. Further, the detected circles hold a sub-pixel accuracy due to the use of the circle equation and the MCA method.

Chapter 6

Conclusion

In this chapter, we briefly discuss how the objectives of the this thesis were met. Also, we remark the possible future work.

6.1 Summary of Thesis Achievements

The work in this thesis has explored a novel application of the TLBO algorithm to different tasks related to the problem of sparse 3D reconstruction. The problems included in some pipeline for 3D reconstruction are: i) Estimation of multiple view relations, such as the fundamental matrix and homography; ii) Vanishing points estimation; and iii) Circle extraction.

Typical methods for solving the aforementioned problems are based on the RANSAC algorithm. However, our approaches additionally uses the TLBO algorithm to improve the search. This means that, the proposed methods do not rely over a mere random selection (case of RANSAC-like techniques), but they also exploit previous information from former candidate solutions.

In comparison with other approaches, the results obtained by the TLBO-based detector required less effort from the user because there are less parameters to tune for this particular meta-heuristic, which is a clear advantage.

6.2 Future Work

Regarding future works, we will make efforts to expand every method to solve different problems. For instance, more work can be done to solve the non-linear cases of the epipolar geometry. Furthermore, we will make efforts to expand our method for circle extraction to detect ellipses and reduce the computation time; and also to explore the use of gradient magnitude, and not only orientation in the objective function, since usually pixels of the same circumference have similar magnitude values. Additionally, it could be interest to make efforts to explore the use of our method to computer vision problems that need vanishing point estimation.

Glossary

- **Barzilai-Borwein method**: Two-point step gradient method.
- **CSA**: Clonal Selection Algorithm; metaheuristic which is inspired by the response of the inmune sytem to antigens.
- **Combinatorial problem**: Problems involving finding groupings, ordering or assignments of a discrete, finite set of objects that satisfy certain conditions or constraints.
- **DE**: Differential Evolution; stochastic optimization method.
- **Epipolar geometry**: Intrinsic projective geometry between two views.
- **Epipolar line**: In the geometry of two views, intersection of the image plane with the plane given by a 3D point and the two camera centers.
- **Epipole**: In the geometry of two views, intersection point of an image plane and the line joining the two camera centers (baseline.)
- **Feature descriptor**: Vector of real or binary values representing a single image feature.
- **Feature detector**: Algorithm to find key-points in digital images.
- **Fundamental matrix**: 3×3 matrix that encapsulates the epipolar geomtry.
- **GA**: Genetic Algorithm; metaheuristic based on the behaviour of crhomosomes.
- **Homography**: A 3×3 matrix that maps the perspective transformation of a plane.
- **Hough Transform**: Feature extraction technique; usually lines and circles.

- **Image feature**: Characteristic non-ambiguous image key-points; e.g. Harris corner.
- **Levenberg–Marquardt algorithm**: Model-based optimization technique.
- **Metaheuristic**: Stochastic method for solving optimization/combinatorial problems.
- **Pose**: of a camera, position and orientation.
- **RANSAC**: Random Sample Concensus
- **TLBO**: Teaching Learning Based Optimization algorithm; metaheursitic inspired by the teaching and learning processes.
- **Vanishing point**: Convergence point of projected 3D/real-world parallel lines.

Bibliography

[1] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision.* Cambridge university press, 2003.

[2] Patrick Denis, James H Elder, and Francisco J Estrada. Efficient edge-based methods for estimating manhattan frames in urban imagery. In *European conference on computer vision*, pages 197–210. Springer, 2008.

[3] Christopher G Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[4] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[6] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.

[7] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010.

[8] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, volume 11, page 2. Citeseer, 2011.

[9] Du Q Huynh, Amritpal Saini, and Wei Liu. Evaluation of three local descriptors on low resolution images for robot navigation. In *2009 24th International Conference Image and Vision Computing New Zealand*, pages 113–118. IEEE, 2009.

[10] RI Hartley. In defense of the eight point algorithm ieee transactions on pattern aalysis and machine intelligence. 1997.

[11] Richard Szeliski, Heung-Yeung Shum, Heung-Yeung Shum, and Heung-Yeung Shum. Creating full view panoramic image mosaics and environment maps. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 251–258. ACM Press/Addison-Wesley Publishing Co., 1997.

[12] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73, 2007.

[13] Menghan Xia, Jian Yao, Renping Xie, Li Li, and Wei Zhang. Globally consistent alignment for planar mosaicking via topology analysis. *Pattern Recognition*, 66:239–252, 2017.

[14] Hua Huang, Huiting He, Xin Fan, and Junping Zhang. Super-resolution of human face image using canonical correlation analysis. *Pattern Recognition*, 43(7):2532–2543, 2010.

[15] AydıN Akyol and Muhittin GöKmen. Super-resolution reconstruction of faces by enhanced global models of shape and texture. *Pattern Recognition*, 45(12):4103–4116, 2012.

[16] Jianbo Su, Ronald Chung, and Liang Jin. Homography-based partitioning of curved surface for stereo correspondence establishment. *Pattern Recognition Letters*, 28(12):1459–1471, 2007.

[17] Tiziana D'Orazio and Cataldo Guaragnella. A survey of automatic event detection in multi-camera third generation surveillance systems. *International Journal of Pattern Recognition and Artificial Intelligence*, 29(01):1555001, 2015.

[18] Eduardo Montijano, Eric Cristofalo, Dingjiang Zhou, Mac Schwager, and Carlos Saguees. Vision-based distributed formation control without an external positioning system. *IEEE Transactions on Robotics*, 32(2):339–351, 2016.

[19] Marián Uherčík, Jan Kybic, Yue Zhao, Christian Cachard, and Hervé Liebgott. Line filtering for surgical tool localization in 3d ultrasound images. *Computers in biology and medicine*, 43(12):2036–2045, 2013.

[20] Yue Zhao, Yi Shen, Adeline Bernard, Christian Cachard, and Hervé Liebgott. Evaluation and comparison of current biopsy needle localization and tracking methods using 3d ultrasound. *Ultrasonics*, 73:206–220, 2017.

[21] G López-Nicolás, JJ Guerrero, and C Sagüés. Visual control of vehicles using two-view geometry. *Mechatronics*, 20(2):315–325, 2010.

[22] Josafat Delfin, Hector M Becerra, and Gustavo Arechavaleta. Humanoid localization and navigation using a visual memory. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 725–731. IEEE, 2016.

[23] Josafat Delfin, Hector M Becerra, and Gustavo Arechavaleta. Humanoid navigation using a visual memory with obstacle avoidance. *Robotics and Autonomous Systems*, 109:109–124, 2018.

[24] S Morteza Safdarnejad, Yousef Atoum, and Xiaoming Liu. Temporally robust global motion compensation by keypoint-based congealing. In *European Conference on Computer Vision*, pages 101–119. Springer, 2016.

[25] Yueqiang Zhang, Langming Zhou, Yang Shang, Xiaohu Zhang, and Qifeng Yu. Contour model based homography estimation of texture-less planar objects in uncalibrated images. *Pattern Recognition*, 52:375–383, 2016.

[26] Muhamad Risqi U Saputra, Andrew Markham, and Niki Trigoni. Visual slam and structure from motion in dynamic environments: A survey. *ACM Computing Surveys (CSUR)*, 51(2):37, 2018.

[27] Shumin Liu, Jiajia Chen, Chip-Hong Chang, and Ye Ai. A new accurate and fast homography computation algorithm for sports and traffic video analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2993–3006, 2018.

[28] Zhengyou Zhang, Rachid Deriche, Olivier Faugeras, and Quang-Tuan Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial intelligence*, 78(1-2):87–119, 1995.

[29] Marc Pollefeys, Reinhard Koch, and Luc Van Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 32(1):7–25, 1999.

[30] Richard Hartley and Andrew Zisserman. Epipolar geometry and the fundamental matrix. *Multiple view geometry*, 2000.

[31] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2010.

[32] Richard Roberts, Sudipta N Sinha, Richard Szeliski, and Drew Steedly. Structure from motion for scenes with large duplicate structures. In *CVPR 2011*, pages 3137–3144. IEEE, 2011.

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

[34] Peyman Moghadam, Janusz A Starzyk, and W Sardha Wijesoma. Fast vanishing-point detection in unstructured environments. *IEEE Transactions on Image Processing*, 21(1):425–430, 2012.

[35] Kyung-Seok Seo, Jung-Hwa Lee, and Heung-Moon Choi. An efficient detection of vanishing points using inverted coordinates image space. *Pattern Recognition Letters*, 27(2):102–108, 2006.

[36] Jean-Philippe Tardif. Non-iterative approach for fast and accurate vanishing point detection. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1250–1257. IEEE, 2009.

[37] Long Quan and Roger Mohr. Determining perspective structures using hierarchical hough transform. *Pattern Recognition Letters*, 9(4):279–286, 1989.

[38] Robert T Collins and Richard S Weiss. Vanishing point calculation as a statistical inference on the unit sphere. In *[1990] Proceedings Third International Conference on Computer Vision*, pages 400–403. IEEE, 1990.

[39] Bo Li, Kun Peng, Xianghua Ying, and Hongbin Zha. Vanishing point detection using cascaded 1d hough transform from single images. *Pattern Recognition Letters*, 33(1):1–8, 2012.

[40] Elena Tretyak, Olga Barinova, Pushmeet Kohli, and Victor Lempitsky. Geometric image parsing in man-made environments. *International Journal of Computer Vision*, 97(3):305–321, 2012.

[41] Zengshi Huang, Naijie Gu, Chuanwen Lin, Jie Shen, and Jie Chang. Real time vanishing points detection on smartphones under manhattan world assumption. *Pattern Recognition Letters*, 115:117–127, 2018.

[42] Michel Antunes, Joao P Barreto, Djamila Aouada, and Bjorn Ottersten. Unsupervised vanishing point detection and camera calibration from a single manhattan image with radial distortion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4288–4296, 2017.

[43] Antonio Paolillo, Angela Faragasso, Giuseppe Oriolo, and Marilena Vendittelli. Vision-based maze navigation for humanoid robots. *Autonomous Robots*, 41(2):293–309, 2017.

[44] Yong Li, Weili Ding, XuGuang Zhang, and Zhaojie Ju. Road detection algorithm for autonomous navigation systems based on dark channel prior and vanishing point in complex road scenes. *Robotics and Autonomous Systems*, 85:1–11, 2016.

[45] Johannes Gräter, Tobias Schwarze, and Martin Lauer. Robust scale estimation for monocular visual odometry using structure from motion and vanishing points. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 475–480. IEEE, 2015.

[46] Federico Camposeco and Marc Pollefeys. Using vanishing points to improve visual-inertial odometry. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 5219–5225. IEEE, 2015.

[47] Zihan Zhou, Farshid Farhat, and James Z Wang. Detecting dominant vanishing points in natural scenes with application to composition-sensitive image retrieval. *IEEE Transactions on Multimedia*, 19(12):2651–2665, 2017.

[48] Roberto Toldo and Andrea Fusiello. Real-time incremental j-linkage for robust multiple structures estimation. In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, volume 1, page 6, 2010.

[49] Marcos Nieto and Luis Salgado. Simultaneous estimation of vanishing points and their converging lines using the em algorithm. *Pattern Recognition Letters*, 32(14):1691–1700, 2011.

[50] Marco Leo, Tommaso De Marco, and Cosimo Distante. Highly usable and accurate iris segmentation. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 2489–2494. IEEE, 2014.

[51] Hau T Ngo, Ryan N Rakvic, Randy P Broussard, and Robert W Ives. Resource-aware architecture design and implementation of hough transform for a real-time iris boundary detection system. *IEEE Transactions on Consumer Electronics*, 60(3):485–492, 2014.

[52] Zhuoman Wen, Yanjie Wang, Jun Luo, Arjan Kuijper, Nan Di, and Minghe Jin. Robust, fast and accurate vision-based localization of a cooperative target used for space robotic arm. *Acta Astronautica*, 136:101–114, 2017.

[53] Martin Saska, Tomas Baca, Justin Thomas, Jan Chudoba, Libor Preucil, Tomas Krajnik, Jan Faigl, Giuseppe Loianno, and Vijay Kumar. System for deployment of groups of unmanned micro aerial vehicles in gps-denied environments using onboard visual relative localization. *Autonomous Robots*, 41(4):919–944, 2017.

[54] Selcan Kaplan Berkaya, Huseyin Gunduz, Ozgur Ozsen, Cuneyt Akinlar, and Serkan Gunal. On circular traffic sign detection and recognition. *Expert Systems with Applications*, 48:67–75, 2016.

[55] Steffen Scholz, Tobias Mueller, Matthias Plasch, Hannes Limbeck, Raphael Adamietz, Tobias Iseringhausen, Daniel Kimmig, Markus Dickerhof, and Christian Woegerer. A modular flexible scalable and reconfigurable system for manufacturing of microsystems based on additive manufacturing and e-printing. *Robotics and Computer-Integrated Manufacturing*, 40:14–23, 2016.

[56] Luciano da Fontoura Costa and Roberto Marcond Cesar Jr. *Shape analysis and classification: theory and practice*. CRC press, 2010.

[57] Ali Ozgun Ok and Emre Başeski. Circular oil tank detection from panchromatic satellite images: a new automated approach. *IEEE Geoscience and Remote Sensing Letters*, 12(6):1347–1351, 2015.

[58] Chaoyang Li, Hong Huo, and Tao Fang. Oil depots detection from high resolution remote sensing images based on salient region extraction. In *Audio, Language and Image Processing (ICALIP), 2016 International Conference on*, pages 285–288. IEEE, 2016.

[59] Virendra Kumar Yadav, Munesh Chandra Trivedi, Shyam Singh Rajput, and Saumya Batham. Approach to accurate circle detection: Multithreaded implementation of modified circular hough transform. In *Proceedings of International Conference on ICT for Sustainable Development*, pages 25–34. Springer, 2016.

[60] Vineet Kumar, Abhijit Asati, and Anu Gupta. Memory-efficient architecture of circle hough transform and its fpga implementation for iris localization. *IET Image Processing*, 2018.

[61] Zhenjie Yao and Weidong Yi. Curvature aided hough transform for circle detection. *Expert Systems with Applications*, 51:26–33, 2016.

[62] Antoine Manzanera, Thanh Phuong Nguyen, and Xiaolei Xu. Line and circle detection using dense one-to-one hough transforms on greyscale images. *EURASIP Journal on Image and Video Processing*, 2016(1):46, 2016.

[63] Li-qin Jia, Cheng-zhang Peng, Hong-Min Liu, and Zhi-Heng Wang. A fast randomized circle detection algorithm. In *Image and Signal Processing (CISP), 2011 4th International Congress on*, volume 2, pages 820–823. IEEE, 2011.

[64] Hancheng Yu and Tuo Wang. Vision-based technique for circle detection and measurement using lookup table and bitwise center accumulator. *JOSA A*, 34(3):415–423, 2017.

[65] Ruben Gonzalez. Fast line and circle detection using inverted gradient hash maps. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 1354–1358. IEEE, 2015.

[66] Hanqing Zhang, Krister Wiklund, and Magnus Andersson. A fast and robust circle detection method using isosceles triangles sampling. *Pattern Recognition*, 54:218–228, 2016.

[67] Kuo-Liang Chung, Yong-Huai Huang, Shi-Ming Shen, Andrey S Krylov, Dmitry V Yurin, and Ekaterina V Semeikina. Efficient sampling strategy and refinement strategy for randomized circle detection. *Pattern recognition*, 45(1):252–263, 2012.

[68] Tommaso De Marco, Dario Cazzato, Marco Leo, and Cosimo Distante. Randomized circle detection with isophotes curvature analysis. *Pattern Recognition*, 48(2):411–421, 2015.

[69] Michele Fornaciari, Andrea Prati, and Rita Cucchiara. A fast and effective ellipse detector for embedded vision applications. *Pattern Recognition*, 47(11):3693–3708, 2014.

[70] Yipeng Li and Chunhui Zhao. Fast ellipse detection by elliptical arcs extracting and grouping. In *Sixth International Conference on Graphic and Image Processing (ICGIP 2014)*, volume 9443, page 94430C. International Society for Optics and Photonics, 2015.

[71] Cuneyt Akinlar and Cihan Topal. Edcircles: A real-time circle detector with a false detection control. *Pattern Recognition*, 46(3):725–740, 2013.

[72] NA Barricelli. Numerical examples of evolution processes. *Methodos*, 1954:45–68, 1954.

[73] Simon Goss, Serge Aron, Jean-Louis Deneubourg, and Jacques Marie Pasteels. Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, 76(12):579–581, 1989.

[74] Marco Dorigo and Luca Maria Gambardella. Ant colonies for the travelling salesman problem. *biosystems*, 43(2):73–81, 1997.

[75] Dervis Karaboga. An idea based on honey bee swarm for numerical optimization. Technical report, Technical report-tr06, Erciyes university, engineering faculty, computer . . . , 2005.

[76] Alex S Fraser. Simulation of genetic systems by automatic digital computers i. introduction. *Australian Journal of Biological Sciences*, 10(4):484–491, 1957.

[77] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[78] Zong Woo Geem, Joong Hoon Kim, and Gobichettipalayam Vasudevan Loganathan. A new heuristic optimization algorithm: harmony search. *simulation*, 76(2):60–68, 2001.

[79] Xin-She Yang. Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms*, pages 169–178. Springer, 2009.

[80] Ravipudi V Rao, Vimal J Savsani, and DP Vakharia. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43(3):303–315, 2011.

[81] R Venkata Rao, Vimal J Savsani, and DP Vakharia. Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. *Information Sciences*, 183(1):1–15, 2012.

[82] R Rao. Review of applications of tlbo algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems. *Decision science letters*, 5(1):1–30, 2016.

[83] Shom Prasad Das and Sudarsan Padhy. A novel hybrid model using teaching–learning-based optimization and a support vector machine for commodity futures index forecasting. *International Journal of Machine Learning and Cybernetics*, 9(1):97–111, 2018.

[84] Erik Cuevas, Daniel Zaldívar, and Marco Perez-Cisneros. Estimation of multiple view relations considering evolutionary approaches. In *Applications of Evolutionary Computation in Image Processing and Pattern Recognition*, pages 107–138. Springer, 2016.

[85] Erik Cuevas and Margarita Díaz. A method for estimating view transformations from image correspondences based on the harmony search algorithm. *Computational intelligence and neuroscience*, 2015:4, 2015.

[86] Erik Cuevas, Valentín Osuna, and Diego Oliva. Estimation of view transformations in images. In *Evolutionary Computation Techniques: A Comparative Perspective*, pages 181–204. Springer, 2017.

[87] Lei Han, Tanghuai Fan, Shengnan Zheng, and Chenrong Huang. Vanishing point detection based on an artificial bee colony algorithm. *Journal of Electronic Imaging*, 24(3):033024, 2015.

[88] Xun Pan, Wa Si, and Harutoshi Ogai. Fast vanishing point estimation based on particle swarm optimization. *IEICE TRANSACTIONS on Information and Systems*, 99(2):505–513, 2016.

[89] Lei Han, Chenrong Huang, Shengnan Zheng, Zhen Zhang, and Lizhong Xu. Vanishing point detection and line classification with bpso. *Signal, Image and Video Processing*, 11(1):17–24, 2017.

[90] Erik Cuevas, Daniel Zaldivar, Marco Pérez-Cisneros, and Marte Ramírez-Ortegón. Circle detection using discrete differential evolution optimization. *Pattern Analysis and Applications*, 14(1):93–107, 2011.

[91] Diego Oliva and Erik Cuevas. Detection of circular shapes in digital images. In *Advances and Applications of Optimised Algorithms in Image Processing*, pages 113–134. Springer, 2017.

[92] Erik Cuevas, Valentín Osuna, and Diego Oliva. Multi-circle detection on images. In *Evolutionary Computation Techniques: A Comparative Perspective*, pages 35–64. Springer, 2017.

[93] Erik Cuevas and Mauricio González. Multi-circle detection on images inspired by collective animal behavior. *Applied intelligence*, 39(1):101–120, 2013.

[94] Jaco Fourie. Robust circle detection using harmony search. *Journal of Optimization*, 2017, 2017.

[95] Margarita-Arimatea Díaz-Cortés, Erik Cuevas, and Raúl Rojas. Clonal selection algorithm applied to circle detection. In *Engineering Applications of Soft Computing*, pages 143–164. Springer, 2017.

[96] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[97] Emilio Garcia-Fidalgo and Alberto Ortiz. Probabilistic appearance-based mapping and localization using visual features. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 277–285. Springer, 2013.

[98] Marcos Nieto and Luis Salgado. Non-linear optimization for robust estimation of vanishing points. In *2010 IEEE International Conference on Image Processing*, pages 1885–1888. IEEE, 2010.

[99] Lei Han, Chenrong Huang, Shengnan Zheng, Zhen Zhang, and Lizhong Xu. Vanishing point detection and line classification with bpso. *Signal, Image and Video Processing*, 11(1):17–24, 2017.

[100] José Lezama, Gregory Randall, and Rafael Grompone von Gioi. Vanishing point detection in urban scenes using point alignments. *Image Processing On Line*, 7:131–164, 2017.

[101] Markéta Dubská, Adam Herout, and Jiří Havel. Pclines—line detection using parallel coordinates. In *CVPR 2011*, pages 1489–1494. IEEE, 2011.

[102] Michel Antunes and Joao P Barreto. A global approach for the detection of vanishing points and mutually orthogonal vanishing directions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1336–1343, 2013.

[103] Erik Cuevas, Fernando Wario, Valentın Osuna-Enciso, Daniel Zaldivar, and Marco Pérez-Cisneros. Fast algorithm for multiple-circle detection on images using learning automata. *IET Image Processing*, 6(8):1124–1135, 2012.

[104] Alan Lopez-Martinez and Francisco Cuevas. Automatic multi-circle detection on images using the teaching learning based optimization algorithm. *IET Computer Vision*, 2018.

[105] ER Davies. The effect of noise on edge orientation computations. *Pattern recognition letters*, 6(5):315–322, 1987.

[106] Josef Kittler. On the accuracy of the sobel edge detector. *Image and Vision Computing*, 1(1):37–42, 1983.

[107] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985.

[108] Jerry R Van Aken. An efficient ellipse-drawing algorithm. *IEEE Computer Graphics and Applications*, 4(9):24–35, 1984.

[109] Andrew C Rice, Alastair R Beresford, and Robert K Harle. Cantag: an open source software toolkit for designing and deploying marker-based vision systems. In *Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'06)*, pages 10–pp. IEEE, 2006.