COMPUTER VISION METHODS FOR AUTOMOTIVE APPLICATIONS

VISIÓN COMPUTACIONAL PARA APLICACIONES AUTOMOTIVAS

Aldo A. Díaz-Salazar, Paulo R. G. Kurka

School of Mechanical Engineering, University of Campinas, Campinas, Brazil

* *E-mail: aldodiaz64@gmail.com*

Recibido (Received): 05 / 12 / 2019 | Aceptado (Accepted): 03/ 10 / 2020

ABSTRACT

Recent advances in computer vision are leveraging many technological developments in modern industry and automation. In this tutorial, it is presented a review of computer vision methods and applications relevant to the use of cameras as measurement devices in the automotive industry and robotics. The methods include algorithms for edge and ellipse detection, camera calibration, 3-D reconstruction and stereo vision. The applications are elaborated through simulations of three key problems: Detection of rims in automotive wheels; estimation of the calibration angles of vehicles and; trajectory reconstruction using stereo vision. These applications allow to demonstrate the potential of vision-based technologies in solving complex engineering problems in an automated fashion using cameras as sensors. As a result, three general purpose methodologies are proposed for solving problems of industrial need that would serve as guidelines for further developments in current and other related areas.

Keywords: computer vision, wheel rim detection, vehicle calibration angles, stereo odometry

RESUMEN

Avances recientes en visión computacional vienen favoreciendo el desarrollo tecnológico en la industria y automatización modernas. En este tutorial, se presenta una revisión de métodos de visión computacional y aplicaciones relevantes para el uso de cámaras como dispositivos de medición en la industria automotiva y robótica. Los métodos incluyen algoritmos para detección de bordes y elipses, calibración de cámaras, reconstrucción 3-D y visión estéreo. Las aplicaciones son elaboradas utilizando simulaciones en tres problemas fundamentales: Detección de aros en ruedas automotivas; estimativa de los ángulos de calibración en vehículos y; reconstrucción de la trayectoria de un vehículo usando visión estéreo. Estas aplicaciones demuestran el potencial de los métodos de visión computacional para resolver problemas de ingeniería complejos de manera automatizada utilizando cámaras como sensores. Como resultado, son propuestas tres metodologías de propósito general para solución a problemas con demanda industrial y que pueden servir como guías para desarrollos futuros en áreas afines o relacionadas.

Palabras Clave: Visión computacional, detección de aros, ángulos de calibración de vehículos, adometría estéreo

1. INTRODUCTION

Computer vision is the science of extracting information and knowledge from images in order to provide an understanding of the world based on camera observations. Currently, the research on autonomous vehicles has driven many technological advances in computer vision methods and applications. Some of the state-of-the-art computer vision algorithms are 3-D reconstruction from multiple camera views, visual odometry (VO) and visual SLAM (simultaneous localization and mapping or structure from motion), which enable many applications such as autonomous navigation, scene reconstruction, map creation and exploration, etc.

In this work we present a tutorial on the use of computer vision applied to solve actual problems in the automotive industry. More specifically, we present algorithms for the use cameras as measuring devices including edge and ellipse detection, camera calibration, 3-D reconstruction and stereo vision. We motivate these algorithms with three industrial applications: Detection of wheel rims; a system to estimate the calibration angles of vehicles and; the reconstruction of the trajectory of a vehicle using stereo vision.

This tutorial is referenced on a previous work in computer vision [1], but with dedicated attention to the applications in the automotive field including extended theory and results. Therefore, it can be used as a self-contained introduction for engineers and practitioners to motivate further developments that use computer vision in the automotive industry and

related fields. The methods were carefully selected according to the challenges presented on each application. For instance, objection detection tasks on Section 2.1 and Section 2.2 are solved using shape detection techniques. Section 2.3 to Section 2.5 introduce the fundamental techniques for the use of cameras as measuring devices allowing metrology applications such as measuring angles and trajectories.

The applications provide computer vision methodologies at an entry level to important problems in Advance Driver Assistance Systems (ADAS), Wheel Alignment and Mobile Robotics. More specifically, the first application discusses how to detect rounded shapes in images, which is important for obstacle avoidance [2], the second application shows an automatic approach for detecting the wheel alignment angles of vehicles using cameras as a low cost alternative to laser-based devices [3] and, finally, the third application shows an application of stereo vision in visual odometry to recover the trajectory of a car-like robot in adverse scenarios where satellite information from the GPS is not accessible [4].

The text is organized as follows: Section 2 presents the theory and foundations of computer vision methods; Section 3 presents three applications of these methods in the automotive industry and robotics; Section 4 outlines the final remarks and conclusions.

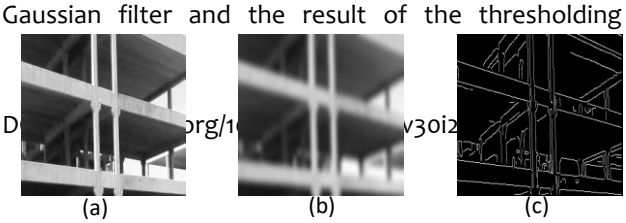
2. Computer vision methods

In this section we make a summary of selected computer vision methods and theory required for the applications introduced in the following sections.

2.1. Edge detection

The edges of an image are regions characterized by high changes of pixel intensities [5]. The image edges provide a rich source of information and a simplified representation of a complex image. Edges can be used to detect shapes in object detection tasks such as the lines of buildings or roads, or either to isolate geometrical shapes such as rounded objects (e.g. cells, wheels, balls).

One of the most popular edge detectors was proposed by Canny in [6]. The Canny's algorithm consists of convolving a grayscale image with a smoothing filter, typically a Gaussian square subimage, to obtain a coarse depiction of the original image. A binary threshold is applied to the smoothed intensities and the regions with the most salient changes are identified as the image edges. Figure 1a shows the image of a building with pillars showing predominant straight-line edges. The image is convolved with a Gaussian filter and the result of the thresholding process is shown on Figure 1c with the image edges depicted as white lines.



(a) (b) (c)

Figure 1. Edge detection in the image of a building.

2.2. Ellipse detection

Shape detection methods are instrumental for identifying the geometrical shapes of objects in images. The detection of shapes has a wide variety of applications such as robot localization [5], object measurement [7], counting [8] and identification [9]. In the automotive industry, detecting wheel shapes may be useful for applications such as measuring wheel dimensions for detecting anomalies or identifying wheeled vehicles for obstacle avoidance on the road such as bicycles, motorcycles and cars.

One of the most popular shape detectors is the so called Hough transform [10], [11]. The Hough transform operates over the edges of an image and is based on a voting principle which consists in parametrizing the shape to be detected (e.g., a line or an ellipse) to produce a mapping of the edge intensities to a *space of parameters*. An accumulator is incremented on each point coincidence of the curves in the parameter space and the voting is to choose among accumulator cells with the highest count representing the parameters of the detected shape.

The parametrization of an ellipse of coordinates (u, v) is given by

$$\frac{(u_e)^2}{a^2} + \frac{(v_e)^2}{b^2} = 1\,, \tag{1}$$

$$u_e = (u - u_0)\cos\Theta + (v - v_0)\sin\Theta\,, \tag{2}$$

$$v_e = -(u - u_0)\sin\Theta + (v - v_0)\cos\Theta\,, \tag{3}$$

where u_0 and v_0 are the coordinates of the ellipse center, a and b are the lengths of the semi-major and the semi-minor axes, respectively, and Θ is the ellipse rotation angle.

2.3. Camera calibration

Camera calibration is the process of determining the parameters of a camera required to use it as a measuring device in computer vision and metrology applications. The camera parameters consist of the intrinsics (focal length, principal point, lens distortion) and the extrinsics (rotation and translation with respect to world coordinates). The following camera calibration procedure, based on the work in [12], uses a solid box as calibration pattern as depicted in Fig. 2.

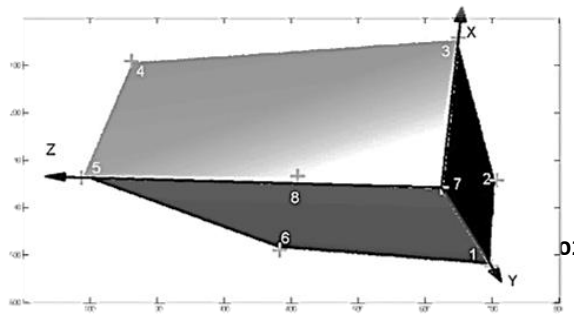


Figure 2. Solid calibration box and the world coordinates.

The reference points for calibration are the positions of the eight vertices of the box in world coordinates $\boldsymbol{B}_k$ and their pixel mappings $\widehat{\boldsymbol{p}}_k$, with $k = 1, \ldots, 8$; they are related by

$$z_k \widehat{\boldsymbol{p}}_k = \mathbf{K}(\mathbf{R}\boldsymbol{B}_k + \boldsymbol{t}) \,. \tag{4}$$

Calibration consists in estimating the unknowns in (4), which are the depths z_k, the matrix of intrinsic parameters $\mathbf{K}$, the rotation $\mathbf{R}$, and translation $\boldsymbol{t}$. The procedure consists of grouping sets of three different vertices, denoted $\boldsymbol{B}_{k_a}$, $\boldsymbol{B}_{k_b}$ and $\boldsymbol{B}_{k_c}$, and their corresponding pixel mappings $\widehat{\boldsymbol{p}}_{k_a}$, $\widehat{\boldsymbol{p}}_{k_b}$ and $\widehat{\boldsymbol{p}}_{k_c}$

$$\left[\boldsymbol{B}_{k_I}^T \otimes \mathbf{I} - \boldsymbol{B}_{k_{II}}^T \otimes \widehat{\mathbf{P}}_{k_I}\widehat{\mathbf{P}}_{k_{II}}^{-1}\right] \mathrm{vec}(\mathbf{K}_R) = 0 \,, \tag{5}$$

where $\otimes$ denotes the Kronecker product, $\mathbf{I}$ is the identity matrix of size 3, $\mathrm{vec}(\cdot)$ denotes the matrix to vector operator which stacks the columns of matrix $\mathbf{K}_R$ to yield a vector of size 9, $\boldsymbol{B}_{k_I}$ and $\boldsymbol{B}_{k_{II}}$ are defined as

$$\boldsymbol{B}_{k_I} = \left(0.5\boldsymbol{B}_{k_a} - \boldsymbol{B}_{k_b} + 0.5\boldsymbol{B}_{k_c}\right) , \tag{6}$$

$$\boldsymbol{B}_{k_{II}} = \left(-\boldsymbol{B}_{k_a} + 0.5\boldsymbol{B}_{k_b} + 0.5\boldsymbol{B}_{k_c}\right) , \tag{7}$$

and the matrices $\widehat{\mathbf{P}}_{k_I}$ and $\widehat{\mathbf{P}}_{k_{II}}$ are defined as

$$\widehat{\mathbf{P}}_{k_I} = \left[0.5\,\widehat{\boldsymbol{p}}_{k_a} \quad -\widehat{\boldsymbol{p}}_{k_b} \quad 0.5\widehat{\boldsymbol{p}}_{k_c}\right] , \tag{8}$$

$$\widehat{\mathbf{P}}_{k_{II}} = \left[-\,\widehat{\boldsymbol{p}}_{k_a} \quad 0.5\widehat{\boldsymbol{p}}_{k_b} \quad 0.5\widehat{\boldsymbol{p}}_{k_c}\right] , \tag{9}$$

and matrix $\mathbf{K}_R$ is defined as

$$\mathbf{K}_R = \mathbf{K}\mathbf{R} \,. \tag{10}$$

Equation (5) is extended adding $n \geq 3$ different sets of box vertices and their corresponding pixel mappings to form an overdetermined system of equations

$$\underbrace{\begin{bmatrix} \left(\boldsymbol{B}_{1_I}^T \otimes \mathbf{I} - \boldsymbol{B}_{1_{II}}^T \otimes \widehat{\mathbf{P}}_{1_I}\widehat{\mathbf{P}}_{1_{II}}^{-1}\right) \\ \left(\boldsymbol{B}_{2_I}^T \otimes \mathbf{I} - \boldsymbol{B}_{2_{II}}^T \otimes \widehat{\mathbf{P}}_{2_I}\widehat{\mathbf{P}}_{2_{II}}^{-1}\right) \\ \vdots \\ \left(\boldsymbol{B}_{n_I}^T \otimes \mathbf{I} - \boldsymbol{B}_{n_{II}}^T \otimes \widehat{\mathbf{P}}_{n_I}\widehat{\mathbf{P}}_{n_{II}}^{-1}\right) \end{bmatrix}}_{\mathbf{P}} \underbrace{\mathrm{vec}(\mathbf{K}_R)}_{\boldsymbol{x}} = 0 \,, \tag{11}$$

Equation (11) is a system of $3n \times 9$ homogeneous equations with solutions in the right null space of $\mathbf{P}$. However, the matrix $\mathbf{P}$ may not have a null space since the pixel positions of the box vertices are only calculated approximately. Hence, the vector $\boldsymbol{x}$ will be estimated as the right-singular vector of $\mathbf{P}$ associated to the least non-zero value. The vector $\boldsymbol{x}$ is reshaped into a matrix $\overline{\mathbf{K}}_R$ of size 3, which is an estimate of matrix $\mathbf{K}_R$,

$$\overline{\mathbf{K}}_R = \mathrm{reshape}(\boldsymbol{x}, 3 \times 3) \,. \tag{12}$$

By exploiting the orthogonal property of the rotation in (10), an estimate of the product of the matrix of intrinsic parameters $\mathbf{K}$ and its transpose $\mathbf{K}^T$ can be derived as

$$\mathbf{K}_R\mathbf{K}_R^T = \begin{bmatrix} (S_x f)^2 + u_0^2 & u_0 v_0 & u_0 \\ u_0 v_0 & \left(S_y f\right)^2 + v_0^2 & v_0 \\ u_0 & v_0 & 1 \end{bmatrix} \cong \widetilde{\mathbf{K}}_R\widetilde{\mathbf{K}}_R^T \,, \tag{13}$$

where matrix $\widetilde{\mathbf{K}}_R\widetilde{\mathbf{K}}_R^T$ is the normalized version of $\overline{\mathbf{K}}_R\overline{\mathbf{K}}_R^T$, obtained by imposing its third row, third column element to be 1, as in the same element of $\mathbf{K}\mathbf{K}^T$. The estimate of the matrix of intrinsic parameters, denoted $\widetilde{\mathbf{K}}$, is obtained from the elements of $\widetilde{\mathbf{K}}_R\widetilde{\mathbf{K}}_R^T$, as

$$\widetilde{\mathbf{K}} = \begin{bmatrix} \tilde{S}_x f & 0 & \tilde{u}_0 \\ 0 & \tilde{S}_y f & \tilde{v}_0 \\ 0 & 0 & 1 \end{bmatrix} , \tag{14}$$

$$\tilde{u}_0 \cong \text{row 1, column 3 of } \widetilde{\mathbf{K}}_R\widetilde{\mathbf{K}}_R^T \,, \tag{15}$$

$$\tilde{v}_0 \cong \text{row 2, column 3 of } \widetilde{\mathbf{K}}_R\widetilde{\mathbf{K}}_R^T \,, \tag{16}$$

$$\tilde{S}_x f \cong \sqrt{\text{row 1, column 1 of } \widetilde{\mathbf{K}}_R\widetilde{\mathbf{K}}_R^T - \tilde{u}_0^2} \,, \tag{17}$$

$$\tilde{S}_y f \cong \sqrt{\text{row 2, column 2 of } \widetilde{\mathbf{K}}_R\widetilde{\mathbf{K}}_R^T - \tilde{v}_0^2} \,, \tag{18}$$

A first approximation of the rotation is derived from (10) and the results of (14) and (12), as

$$\mathbf{R} = \mathbf{K}^{-1}\mathbf{K}_R \cong \widetilde{\mathbf{K}}^{-1}\overline{\mathbf{K}}_R \,. \tag{19}$$

To satisfy the orthogonal property of rotations, the final rotation estimate, denoted $\overline{\mathbf{R}}$, is obtained from the singular value decomposition (SVD) of matrix $\widetilde{\mathbf{K}}^{-1}\overline{\mathbf{K}}_R$

$$\overline{\mathbf{R}} = \mathbf{U}\mathbf{V}^T \,, \tag{20}$$

where $\mathbf{U}$ and $\mathbf{V}$ are matrices of the left and right singular vectors of $\widetilde{\mathbf{K}}^{-1}\overline{\mathbf{K}}_R$, respectively.

The estimate of the vector of depth reprojections, denoted $\boldsymbol{\lambda}$, and the translation vector, denoted $\boldsymbol{t}$, are obtained simultaneously by extending (4) to $n \geq 2$ different pairs of vertices and their pixel mappings, and solving for

$$\begin{bmatrix} \widehat{\boldsymbol{p}}_1 & 0 & \ldots & 0 & -\widetilde{\mathbf{K}} \\ 0 & \widehat{\boldsymbol{p}}_2 & \ldots & 0 & -\widetilde{\mathbf{K}} \\ \vdots & \vdots & \ddots & \vdots & -\widetilde{\mathbf{K}} \\ 0 & 0 & \ldots & \widehat{\boldsymbol{p}}_n & -\widetilde{\mathbf{K}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{t} \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{K}}_R\boldsymbol{B}_1 \\ \overline{\mathbf{K}}_R\boldsymbol{B}_2 \\ \vdots \\ \overline{\mathbf{K}}_R\boldsymbol{B}_n \end{bmatrix} , \tag{21}$$

where $\boldsymbol{\lambda}$ is defined as

$$\boldsymbol{\lambda} = \left[z_1 \quad z_2 \quad \ldots \quad z_n\right]^T \,. \tag{22}$$

2.4. 3-D reconstruction

Figure 3 depicts the 3-D reconstruction of a point in the space $\boldsymbol{P}^\star$ by *triangulation*. Triangulation consists of finding the closest point between the rays $\boldsymbol{l} = (O_l, p_l)$ and $\boldsymbol{r} = (O_r, p_r)$ from the origins of left and right cameras to the point $\boldsymbol{P}$. An optimal solution to the 3-D reconstruction problem is obtained by minimizing the geometric error in spatial position and solving for depths z_l and z_r

$$z_l \boldsymbol{p}_l - z_r \mathbf{R}^T \boldsymbol{p}_r - \boldsymbol{t} = \boldsymbol{e}\,, \tag{23}$$

$$\begin{bmatrix} z_l \\ z_r \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_l^T \boldsymbol{p}_l & -\boldsymbol{p}_l^T \mathbf{R}^T \boldsymbol{p}_r \\ -\boldsymbol{p}_l^T \mathbf{R}^T \boldsymbol{p}_r & \boldsymbol{p}_r^T \boldsymbol{p}_r \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{p}_l^T \boldsymbol{t} \\ -\boldsymbol{p}_r^T \mathbf{R} \boldsymbol{t} \end{bmatrix}, \tag{24}$$

where $\boldsymbol{e}$ is the error vector in spatial position of crossing rays, $\boldsymbol{p}_l$ and $\boldsymbol{p}_r$ are normalized pixel coordinates (focal length $f = 1m$) and $(\mathbf{R}, \boldsymbol{t})$ are the extrinsic parameters denoting the transformation between camera views.

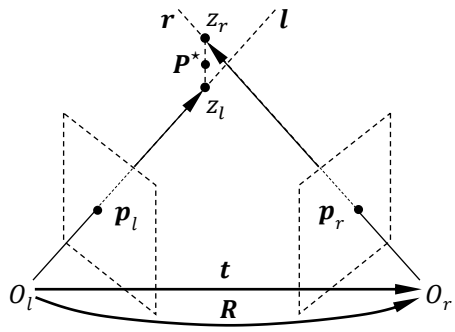


Figure 3. The 3-D reconstruction by triangulation.

2.5. Stereo vision

Two camera views are related by *epipolar geometry* or the geometry of stereo vision. Figure 4 illustrates the epipolar relations. The line connecting the camera projection centers is called baseline. The baseline intersects the image planes π_l and π_r at special points called epipoles $\boldsymbol{e}_l$ and $\boldsymbol{e}_r$. A point in the 3-D space $\boldsymbol{P}$ is described by the vectors $\boldsymbol{P}_l = [x_l \quad y_l \quad z_l]^T$ and $\boldsymbol{P}_r = [x_r \quad y_r \quad z_r]^T$ and the camera projections $\boldsymbol{p}_l = [x_l' \quad y_l' \quad f_l]^T$ and $\boldsymbol{p}_r = [x_r' \quad y_r' \quad f_r]^T$, where f_l and f_r are the focal lengths. The point $\boldsymbol{P}$ and the projection centers O_l and O_r describe a plane π_P called *epipolar plane* (in gray color). The lines connecting the epipoles to the camera projection centers are called *epipolar lines*, denoted $\boldsymbol{u}_l$ and $\boldsymbol{u}_r$. The stereo configuration also imposes the *epipolar constraint* which restricts the match of a point in an image to the epipolar line on the opposite image.

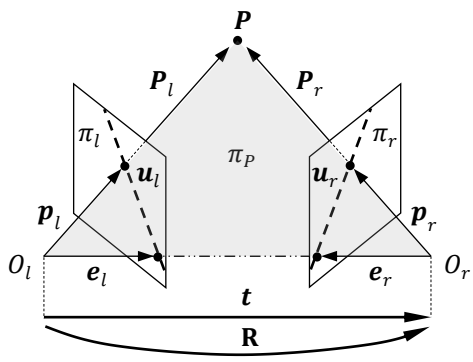


Figure 4. The epipolar geometry.

The translation vector $\boldsymbol{t}$ and rotation matrix $\mathbf{R}$ are the extrinsic parameters that relate the reference frames of the left and right cameras. Then, the points $\boldsymbol{P}_l$ and $\boldsymbol{P}_r$ are related by the rigid body transformation

$$\boldsymbol{P}_l = \mathbf{R}^T \boldsymbol{P}_r + \boldsymbol{t}\,. \tag{25}$$

The point $\boldsymbol{P}$ is related to its perspective projection $\boldsymbol{p}$ using the standard pinhole camera model

$$z_l \boldsymbol{p}_l = f_l \boldsymbol{P}_l\,, \tag{26}$$

$$z_r \boldsymbol{p}_r = f_r \boldsymbol{P}_r\,. \tag{27}$$

The *essential matrix* $\mathbf{E}$ of camera coordinates establishes an epipolar constraint between the left and right projections

$$\boldsymbol{p}_r^T \mathbf{E} \boldsymbol{p}_l = 0\,, \tag{28}$$

$$\mathbf{E} = \mathbf{R}\widehat{\mathbf{T}}\,, \tag{29}$$

where $\widehat{\mathbf{T}}$ is the skew-symmetric matrix expressing the cross product with the translation vector $\boldsymbol{t} \times (\cdot)$. The points $\widehat{\boldsymbol{p}}_l$ and $\widehat{\boldsymbol{p}}_r$ are the pixel mappings in homogenous coordinates corresponding to $\boldsymbol{p}_l$ and $\boldsymbol{p}_r$

$$\boldsymbol{p}_l = f_l \mathbf{K}_l^{-1} \widehat{\boldsymbol{p}}_l\,, \tag{30}$$

$$\boldsymbol{p}_r = f_r \mathbf{K}_r^{-1} \widehat{\boldsymbol{p}}_r\,, \tag{31}$$

where $\mathbf{K}_l$ and $\mathbf{K}_r$ are the calibration matrices of the left and right cameras, respectively.

The *fundamental matrix* $\mathbf{F}$ of image coordinates establishes the epipolar constraint between the left and right projections as

$$\widehat{\boldsymbol{p}}_r^T \mathbf{F} \widehat{\boldsymbol{p}}_l = 0\,, \tag{32}$$

$$\mathbf{F} = \mathbf{K}_r^{-T} \mathbf{E} \mathbf{K}_l^{-1}\,. \tag{33}$$

Matrices $\mathbf{E}$ and $\mathbf{F}$ can be estimated using the eight-point algorithm [9].

Figure 5 shows the geometric relations between the cameras and the world. Rotations $\mathbf{R}_l$ and $\mathbf{R}_r$ and translations $\boldsymbol{t}_l$ and $\boldsymbol{t}_r$ describe the origin of world coordinates O_w in terms of the reference frames of the left and right cameras, respectively. The extrinsic parameters of the stereo system $\mathbf{R}$ and $\boldsymbol{t}$ are calculated with respect to the reference frame of the left camera, such as

$$\mathbf{R} = \mathbf{R}_r \mathbf{R}_l^T\,, \tag{34}$$

$$\boldsymbol{t} = \boldsymbol{t}_l - \mathbf{R}^T \boldsymbol{t}_r\,. \tag{35}$$

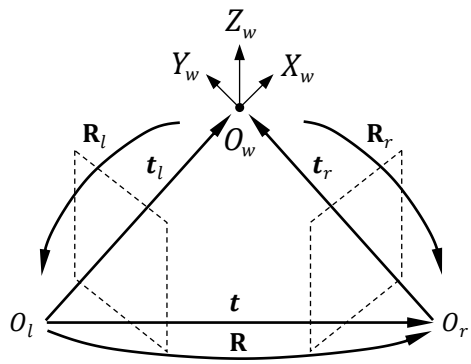


Figure 5. The extrinsic parameters of a stereo system.

3. Automotive applications

In this section we present three applications of computer vision methods in the automotive industry and robotics. In order to highlight the potential of the methods, we make use of simulations using Blender, which is an open-source platform used for generating virtual scenarios as ground truth data for evaluation [13], [14]. In Section 3.1, real images of a wheel were used. In Section 3.2, we used the 3-D model of a vehicle and an array of eight cameras grouped in pairs, as in Fig. 8. In Section 3.3, we used a 3-D indoors environment comprised of texturized furniture, chairs, floor, ceiling, windows, natural and artificial illumination to enable the detection of visual keypoints. A 3-D model of an automotive vehicle with a mounted pair of stereoscopic cameras is also introduced in the environment. The car-like robot simulates an autonomous vehicle following a predefined trajectory inside a virtual environment, capturing stereo images. The ground truth path contains straight and diagonal trajectories with rotations at different angles.

3.1. Detection of wheel rim

In this application we describe a method for detecting rims in automotive wheel images using the Hough transform. Rim detection may be useful to identify wheeled vehicles in tracking applications [2]. Figure 6a depicts a wheel image originally affected by poor illumination and lens distortion which are corrected using the histogram equalization and correction of lens distortion. Figure 6b shows the resulting image including both corrections.

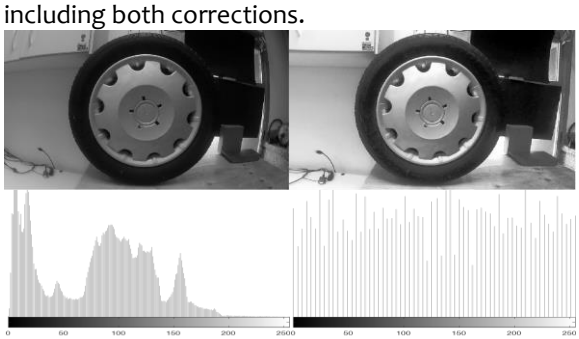
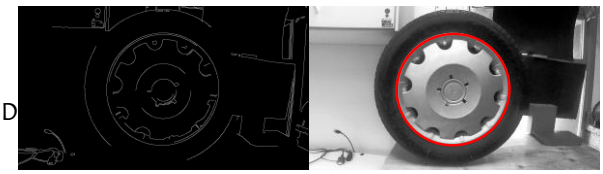


Figure 6. Original and distortion-free wheel images.

The ellipse containing the wheel rim is detected using the Hough transform applied to elliptical shapes [15], [16]. The procedure starts by converting the RBG image to grayscale intensities

$$Y = 0.299R + 0.587G + 0.114B\,, \qquad (36)$$

The Canny algorithm is computed over the grayscale image in order to detect the edges of the wheel. The rim ellipse is searched over the image edges using the ellipse parametrization given in (37-38). Figure 7 illustrates the detection results. Figure 7b represents the detected wheel rim ellipse in red thick line.



(a) (b)

Figure 7. Wheel rim detection

3.2. Toe and camber calibration angles

In this application we describe a system to recover the calibration angles of a vehicle. The system can be used as an automatic system to measure the wheel geometry and also to determine the toe and camber calibration angles. Figure 8 depicts the measuring system consisting of four stereo camera subsystems associated to wheels 1, 2, 3 and 4. The stereo subsystems are attached to a support of fixed baseline to have the center of the wheels and the optical axes of the cameras aligned at approximately the same height. The cameras are oriented towards the wheels and operate in *precision* and *reference* positions. *Precision cameras* 1, 3, 5 and 7 are perpendicular to the wheels to capture the largest possible images of the rims. *Reference cameras* 2, 4, 6 and 8 are oriented towards the geometric center of the system to capture images of the rims.

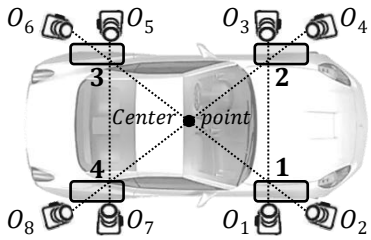
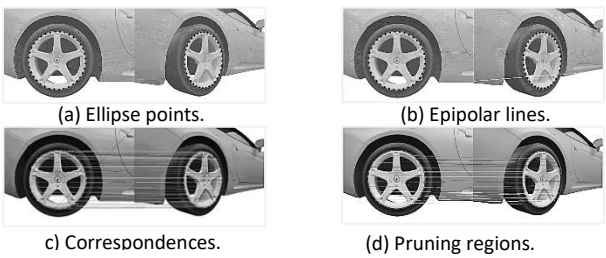


Figure 8. The wheel measuring system.

Rim ellipses are detected using the procedure described in previous section. Figure 9a shows the wheel images of cameras 7 and 8 and the identified rim ellipses (in dotted lines). The rim points on the left image are associated to epipolar lines on the right image due to the epipolar constraint. Epipolar lines intersect the rim ellipse at two points, as shown in Fig. 9b. The epipolar lines of the upper and lower sections of the rim at 10% of the rim diameter are filtered out, as shown in Fig. 9d.

Figure 9. The epipolar lines of a wheel contour and the matching process.

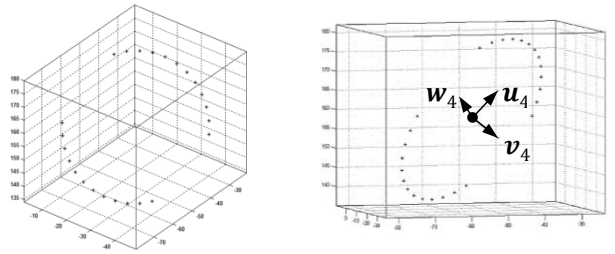


(a) Ellipse points. (b) Epipolar lines.

c) Correspondences. (d) Pruning regions.

a illustrates the result of the 3-D wheel rim reconstruction. The 3-D wheel rim is reconstructed by triangulation of corresponding 2-D rim points using (24). The *wheel axis system* is an auxiliary reference frame defined as the orthogonal 3-D basis resulting from the Principal Component Analysis (PCA) of the 3-D wheel rim, as illustrated in Fig. 10b. The vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ are located on the wheel rim plane and the vector $\boldsymbol{w}$ is collinear with the wheel spin axis. The local PCA coordinates are converted to global coordinates in the world reference frame using the extrinsic parameters of the global calibration process of the corresponding stereo subsystem.

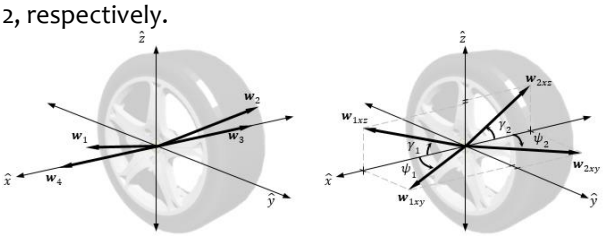
The camber and toe alignment angles of the front wheels are measured in a 3-D global system called the *car reference frame* defined by axis x' in the direction of the line that connects the centers of wheels 3 and 4, axis y' defined in the direction of the line that connects the centers of wheels 3 and 2 and axis z' given by the cross product of x' and y'. The triad x', y' and z' is normalized to yield the unit vectors $\hat{x}$, $\hat{y}$ and $\hat{z}$ denoting the car reference frame.



(a) 3-D wheel rim edge reconstruction. (b) The wheel axis system.

Figure 10. 3-D reconstruction of wheel 4 and its associated PCA directions.

Figure 11a depicts the location of the wheel spin axes in the car reference frame. Vectors $\boldsymbol{w}_1$, $\boldsymbol{w}_2$, $\boldsymbol{w}_3$ and $\boldsymbol{w}_4$ are collinear with the wheel spin axes and perpendicular to wheels 1, 2, 3 and 4, respectively. Vectors $\boldsymbol{w}_3$ and $\boldsymbol{w}_4$ are approximately collinear with the $\hat{x}$-axis. Vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are perpendicular to the misaligned wheels 1 and 2, respectively.



(a) Location of the wheel spin axes in the car reference frame. (b) Location of the camber and toe alignment angles of the front wheels in the car reference frame.

Figure 11. The car reference frame and the wheel alignment angles.

Figure 11b depicts the location of the wheel alignment angles in the car reference frame. Vector $\boldsymbol{w}_{1xz}$ is the projection of $\boldsymbol{w}_1$ onto the $\hat{x}\hat{z}$ plane. The camber angle of wheel 1, denoted γ_1, is the angle from $\boldsymbol{w}_{1xz}$ to the $\hat{x}$-axis and is calculated as

$$\gamma_1 = \arccos\left(\frac{\boldsymbol{w}_{1xz}\cdot\boldsymbol{w}_4}{\|\boldsymbol{w}_{1xz}\|\|\boldsymbol{w}_4\|}\right). \tag{39}$$

Vector $\boldsymbol{w}_{2xz}$ is the projection of $\boldsymbol{w}_2$ onto the $\hat{x}\hat{z}$ plane. The camber angle of wheel 2, denoted γ_2, is the angle from $\boldsymbol{w}_{2xz}$ to the $-\hat{x}$-axis and is calculated as

$$\gamma_2 = \arccos\left(\frac{\boldsymbol{w}_{2xz}\cdot\boldsymbol{w}_3}{\|\boldsymbol{w}_{2xz}\|\|\boldsymbol{w}_3\|}\right). \tag{40}$$

Vector $\boldsymbol{w}_{1xy}$ is the projection of $\boldsymbol{w}_1$ onto the $\hat{x}\hat{y}$ plane. The toe angle of wheel 1, denoted ψ_1, is the angle from $\boldsymbol{w}_{1xy}$ to the $\hat{x}$-axis and is calculated as

$$\psi_1 = \arccos\left(\frac{\boldsymbol{w}_{1xy}\cdot\boldsymbol{w}_4}{\|\boldsymbol{w}_{1xy}\|\|\boldsymbol{w}_4\|}\right). \tag{41}$$

Vector $\boldsymbol{w}_{2xy}$ is the projection of $\boldsymbol{w}_2$ onto the $\hat{x}\hat{y}$ plane. The toe angle of wheel 2, denoted ψ_2, is the angle from $\boldsymbol{w}_{2xy}$ to the $-\hat{x}$-axis and is calculated as

$$\psi_2 = \arccos\left(\frac{\boldsymbol{w}_{2xy}\cdot\boldsymbol{w}_3}{\|\boldsymbol{w}_{2xy}\|\|\boldsymbol{w}_3\|}\right). \tag{42}$$

3.3. Visual odometry

Visual odometry (VO) is a technique to estimate the ego-motion of a vehicle using imaging sensors. VO has gained recent attention as a technique to achieve autonomous navigation in mobile robotics. In the following application the path of robot moving in an unstructured environment is recovered using a stereo camera system in a simulated scenario [17].

The system setup consists of two cameras of 720×480 resolution and fixed baseline mounted on a mobile robotic platform. The extrinsic parameters of the stereo system are rotation matrix $\mathbf{R}$ and translation vector $\boldsymbol{t}$, as in Fig. 4. Calibration is performed using the algorithm introduced in previous section to determine the intrinsic and extrinsic camera parameters. Local robot motion is estimated in the *robot reference frame* located at the projection center of the left camera. The world reference frame is arbitrarily located at the origin of the left camera at the initial position of the robot.

The image matching is based on a practical implementation of the SIFT algorithm [18]. Four stereo images are captured at consecutive robot steps $i-1$ and i, where $i \in \mathbb{N}$. The images are analyzed as in the sequence of Fig. 12. The image planes π_l and π_r represent the left and right camera images, respectively. The main assumption is that consecutive images preserve keypoint correspondences whenever the robot undergoes a small displacement. The set of common keypoints between the four images is the input data for VO estimation.

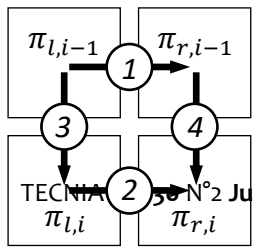


Figure 12. Stereo matching sequence.

Figure 13 illustrates the keypoint correspondences at two successive robot steps. Local robot motion is represented by rotation matrix $\mathbf{R}_i$ and translation vector $\boldsymbol{t}_i$. Local robot motion is recovered from the 3-D reprojections of the keypoints on the left camera, denoted $\boldsymbol{P}_k$ and $\boldsymbol{Q}_k$, at robot steps $i-1$ and i, respectively, where k is the keypoint index. The 3-D reprojections are calculated by triangulation, as in (24). The results are arranged in matrices of homogeneous coordinates $\widehat{\mathbf{P}}$ and $\widehat{\mathbf{Q}}$ of size $4\times k$, according to

$$\widehat{\mathbf{P}} = \begin{bmatrix} \boldsymbol{P}_1 & \boldsymbol{P}_2 & \dots & \boldsymbol{P}_k \\ 1 & 1 & \dots & 1 \end{bmatrix}, \tag{43}$$

$$\widehat{\mathbf{Q}} = \begin{bmatrix} \boldsymbol{Q}_1 & \boldsymbol{Q}_2 & \dots & \boldsymbol{Q}_k \\ 1 & 1 & \dots & 1 \end{bmatrix}. \tag{44}$$

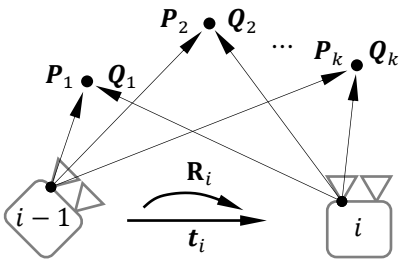


Figure 13. Keypoint correspondences and the robot motion model.

Local robot motion is estimated from the submatrices of the block matrix $\mathbf{M}$ (transformation matrix) that relates matrices $\widehat{\mathbf{P}}$ and $\widehat{\mathbf{Q}}$, such as

$$\mathbf{M} = \widehat{\mathbf{P}}\widehat{\mathbf{Q}}^{\dagger}, \tag{45}$$

$$\mathbf{R}_t = \begin{bmatrix} \overline{\mathbf{R}}_i^T & \bar{\boldsymbol{t}}_i \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{46}$$

where $\overline{\mathbf{R}}_i$ is the 3×3 rotation matrix and $\bar{\boldsymbol{t}}_i$ is the 3×1 translation vector, and $\mathbf{0}^T$ is a zero matrix of size 3×1.

Visual odometry is calculated by the composition of successive rotations and translations. The robot pose, denoted $\boldsymbol{B}_i$, can be defined recursively as

$$\boldsymbol{B}_i = \boldsymbol{B}_{i-1} + \left(\prod_{n=0}^{i-1} \overline{\mathbf{R}}_n^T\right) \bar{\boldsymbol{t}}_i, \quad i > 0, \tag{47}$$

or as in the explicit version

$$\boldsymbol{B}_i = \boldsymbol{B}_0 + \sum_{m=1}^{i} \left(\prod_{n=0}^{m-1} \overline{\mathbf{R}}_n^T\right) \bar{\boldsymbol{t}}_m, \quad i > 0, \tag{48}$$

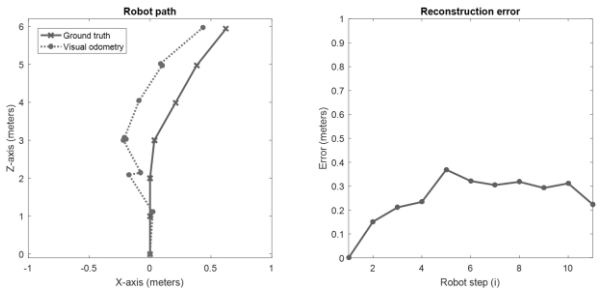
where $\boldsymbol{B}_0$ is the initial robot pose, $\overline{\mathbf{R}}_0 = \mathbf{I}_3$, and $\mathbf{I}_3$ is the identity matrix of size 3.

The system is simulated in the 3-D graphics software Blender. In this application, a 3-D virtual indoors environment is used. The environment is comprised of textured office furniture, floor, ceiling, windows, and artificial illumination. A mobile robot is simulated by a stereo camera system of fixed baseline. The robot moves inside the virtual environment on a predefined trajectory, capturing images from both cameras as it traverses small increments of the distance along the path. Figure 14 depicts the reconstruction of a robot trajectory with $i = 11$ steps. The ground truth trajectory contains straight and diagonal paths and rotations at different angles. The deviation error of the reconstructed path is measured in units of length as

$$e_i = \left\| \widetilde{\boldsymbol{B}}_i - \boldsymbol{B}_i \right\|, \tag{49}$$

where $\widetilde{\boldsymbol{B}}_i$ and $\boldsymbol{B}_i$ are the estimated and the actual robot poses at step i, respectively. VO yields a Gaussian error distribution of mean $\mu = 0.2488m$ and standard deviation $\sigma = 0.1035m$. The VO algorithm is plausible of being implemented in real-time scenarios [19]. The results may be improved in combination with state-of-the-art techniques in visual navigation for autonomous vehicles [4], [20].

Figure 14. The original robot path and the stereo visual



odometry reconstruction.

Conclusions

In this paper we reviewed computer vision methods applied to automotive applications. The methods presented included edge and ellipse detection, camera calibration, 3-D reconstruction and stereo vision which in group constituted the foundations for the use of cameras as measuring devices.

The automotive applications were the detection of wheel rim, the estimation of toe and camber calibration angles and a visual odometry system using a pair of cameras. The results demonstrated the potential of computer vision methods in solving actual problems of industrial relevance in automotive industry.

Acknowledgments

The authors are grateful to the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES, Brazil for funding this project.

References

[1] P. R. G. Kurka and A. A. Díaz Salazar, "Applications of image processing in robotics and instrumentation", *Mech. Syst. Signal Process.*, vol. 124, p. 142–169, jun. 2019, doi: 10.1016/j.ymssp.2019.01.015.

[2] T. Ardeshiri, F. Larsson, F. Gustafsson, T. B.

Schön, and M. Felsberg, "Bicycle tracking using ellipse extraction", 2011.

[3] C. R. Mingoto and P. R. G. Kurka, "Dispositivo, método e uso para conferência de alinhamento de suspensão veicular [Device and method for verification of the alignment of vehicle suspension]", BR PI 1105356-9 A2, 2014.

[4] D. Scaramuzza and F. Fraundorfer, "Visual Odometry [Tutorial]", *IEEE Robot. Autom. Mag.*, vol. 18, nº 4, p. 80–92, dez. 2011, doi: 10.1109/MRA.2011.943233.

[5] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3º ed. Upper Saddle River, NJ: Pearson, 2008.

[6] J. Canny, "A Computational Approach to Edge Detection", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, nº 6, p. 679–698, nov. 1986, doi: 10.1109/TPAMI.1986.4767851.

[7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge: Cambridge University Press, 2004.

[8] E. Trucco ans A. Verri, *Introductory Techniques for 3-D Computer Vision*. Upper Saddle River, NJ: Prentice Hall, 1998.

[9] Y. Ma, S. Soatto, J. Košecká, and S. S. Sastry, *An Invitation to 3-D Vision*, vol. 26. New York, NY: Springer New York, 2004.

[10] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures", *Commun. ACM*, vol. 15, nº 1, p. 11–15, jan. 1972, doi: 10.1145/361237.361242.

[11] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes", *Pattern Recognit.*, vol. 13, nº 2, p. 111–122, 1981, doi: 10.1016/0031-3203(81)90009-1.

[12] P. R. G. Kurka, J. V. Delgado, C. R. Mingoto, and O. E. R. Rojas, "Automatic estimation of camera parameters from a solid calibration box", *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 35, nº 2, p. 93–101, jun. 2013, doi: 10.1007/s40430-013-0013-2.

[13] M. Ramezani, D. Acharya, F. Gu, and K. Khoshelham, "INDOOR POSITIONING by VISUAL-INERTIAL ODOMETRY", 2017, doi: 10.5194/isprs-annals-IV-2-W4-371-2017.

[14] Z. Zhang, H. Rebecq, C. Forster, and D. Scaramuzza, "Benefit of large field-of-view cameras for visual odometry", in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, maio 2016, p. 801–808, doi: 10.1109/ICRA.2016.7487210.

[15] Yonghong Xie and Qiang Ji, "A new efficient ellipse detection method", in *Object recognition supported by user interaction for service robots*, vol. 2, p. 957–960, doi: 10.1109/ICPR.2002.1048464.

[16] C. A. Basca, M. Talos, and R. Brad, "Randomized Hough Transform for Ellipse Detection with Result Clustering", in *EUROCON 2005 - The International Conference on "Computer as a Tool"*, 2005, p. 1397–1400, doi: 10.1109/EURCON.2005.1630222.

[17] V. J. Delgado, P. R. Kurka, and E. Cardozo, "Visual odometry in mobile robots", in *IX Latin American Robotics Symposium and IEEE Colombian Conference on Automatic Control*, out. 2011, p. 1–4, doi: 10.1109/LARC.2011.6086813.

[18] A. Vedaldi and B. Fulkerson, "Vlfeat: An Open and Portable Library of Computer Vision Algorithms", in *Proceedings of the 18th ACM International Conference on Multimedia*, 2010, p. 1469–1472, doi: 10.1145/1873951.1874249.

[19] J. V. Delgado and P. R. G. Kurka, "The Use of a Graphic Processing Unit (GPU) in a Real Time Visual Odometry Application", in *2015 IEEE International Conference on Dependable Systems and Networks Workshops*, jun. 2015, p. 141–146, doi: 10.1109/DSN-W.2015.32.

[20] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part II: Matching, robustness, optimization, and applications", *IEEE Robot. Autom. Mag.*, vol. 19, nº 2, p. 78–90, jun. 2012, doi: 10.1109/MRA.2012.2182810.