Mechanical Systems and Signal Processing 124 (2019) 142–169

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

journal homepage: www.elsevier.com/locate/ymssp

Applications of image processing in robotics and instrumentation

Paulo Roberto Gardel Kurka *, Aldo André Díaz Salazar

School of Mechanical Engineering, University of Campinas, R. Mendeleyev 200, 13083860 Campinas, Brazil

ARTICLE INFO

Article history:
Received 9 February 2018
Received in revised form 17 December 2018
Accepted 6 January 2019

Keywords:
Image processing
Robotics
Instrumentation
Crankshaft dimensional verification
Wheel alignment
Visual odometry

ABSTRACT

Modern applications of industrial automation and robotics are increasingly relying on image processing techniques. This paper shows ways in which image processing can be applied to solve actual problems in robotics and instrumentation. The paper starts by presenting the fundamentals of camera models, digital acquisition of images and selected processing techniques, followed by examples of applications of such knowledge. Four examples of image processing applications are shown: rim detection in automotive wheel images, dimensional verification of crankshafts, measurement of wheel alignment angles of a car and a stereo visual odometry algorithm for mobile robotics. The examples not only illustrate the uses of different image processing techniques, but may also inspire the development of new robotic and industrial automation products.

1. Introduction

The engineering community is experiencing a dramatic growth in application of image processing as a tool to perform non-intrusive precision measurement, autonomous robot navigation or reliable verification of industrial automation processes. Image acquisition and processing techniques are familiar to physics, computer, electrical and mechanical engineering sciences for a relatively long time. Specialized literature on this field is usually dedicated to specific aspects such as the geometry of camera projections and computer vision [1–5], image acquisition and processing [6], or either the scientific and industrial applications of image processing and analysis [7].

This paper is both a review and a tutorial of applied image processing in industrial automation, aimed at researchers, postgraduate students and engineers. Digital images can be associated with geometric properties of objects, such as shapes and dimensions, which enables a myriad of applications and developments. In order to exploit such possibilities, the paper provides insight into some of the key techniques associated with image acquisition, processing and analysis. Theoretical aspects on the geometry of perspective projection, camera calibration, epipolar geometry and stereo image correspondence are further elaborated with regards to technical details. Such a mathematical framework is the core for the use of cameras as measuring devices in photogrammetry. Four applications of image processing are shown through examples in actual robotics and instrumentation scenarios: rim detection in automotive wheel images; dimensional verification of crankshafts; measurement of the wheel alignment angles of a car, and a stereo visual odometry algorithm for mobile robotics. The examples are innovative concepts that can inspire a wider range of image-related applications. Each example considers four aspects: (1) a motivation; (2) an explicit description of the system setup and calibration; (3) mathematical rationale and references to first principles that explicit the assumptions behind each application; (4) empirical results and discussion.

* Corresponding author.
E-mail address: kurka@fem.unicamp.br (P.R.G. Kurka).

https://doi.org/10.1016/j.ymssp.2019.01.015
0888-3270/

The text is organized as follows: Section 2 introduces the pinhole camera model, image sensing devices and color models, which are associated to the process of image acquisition and primary conditioning operations. Section 3 provides a detailed explanation of a selection of image processing techniques, namely, correction of lens distortion, histogram equalization, erosion and dilation, edge and shape detection, geometry of perspective projection, camera calibration, epipolar geometry and the Scale Invariant Feature Transform. All such topics correspond to essential techniques of image processing, followed by the rationale for practical analysis of image elements. Section 4 describes a method for rim detection in automotive wheel images, which illustrates the use of image processing techniques. Section 5 presents a 3-D crankshaft verification system, which illustrates the use of multiple camera images in the construction of a spatial point cloud model of a solid. Section 6 shows a wheel alignment measuring system, which illustrates the potential use of photogrammetry in precision engineering measurements. Section 7 introduces a stereo visual odometry algorithm which shows the potential use of cameras applied to autonomous mobile robotics.

2. Camera, image and color models

This section presents the primary concepts of image acquisition and processing that are the pinhole camera model, digital image formation and practical color models.

2.1. The pinhole camera model

The pinhole camera model [1,8] is the simplest and most useful representation of the process of capturing an image. Fig. 1 illustrates the pinhole camera model. The light reflected from an object is concentrated and projected through the pinhole in the camera's image plane, producing an inverted perspective image [6]. The image plane is located at a focal length f from the pinhole. The scale of the projected image depends on the focal distance. In actual cameras, the pinhole is substituted by optical lenses that perform the task of concentrating the light from objects in front of the camera, providing at the same time a bigger area for the passage of light.

2.2. Image sensing devices

Light that comes from the camera lens pass through primary color (red, green and blue) optical filters embedded in an electronic sensor, which can typically be charge-coupled devices (CCD) or complementary metal-oxide-semiconductors (CMOS). The device is comprised of a two-dimensional array of sensitive elements (pixels) that receive light filtered in the three basic colors and map their intensities into discrete numerical values that usually range from 0 (minimum intensity) to 255 (maximum intensity). Fig. 2 illustrates such an image sensing device. The levels of light intensity discerned by each pixel, in each of the three basic colors, yield a combination of 16,777,216 different color tones. Such a range of different intensities can be numerically represented by a digital number with 24 binary digits (bits). The output digital image is a matrix of M rows by N columns, where each element stores three 8-bit numbers proportional to the primary color intensities.

2.3. Color models

Although an image is usually captured in the primary red, green and blue colors, it can be represented using other color models or formats according to the application. The most popular color formats are presented below.

2.3.1. RGB

The *red*, *green*, and *blue* (RGB) color system, originally used in conventional photography [9], has current applications in television and computer displays [10]. Fig. 3 depicts the formation of colors from primary RGB intensities. In RGB, intensity weighted combinations of the projections of the three primary colors yield all the other possible colors.

Fig. 4 illustrates the RGB components of a colored image.

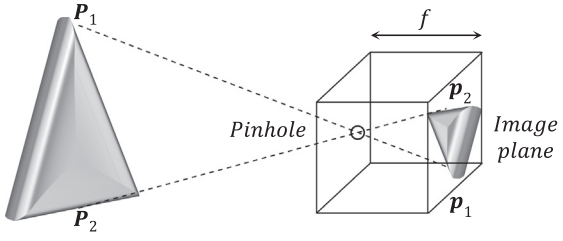


Fig. 1. The pinhole camera model.

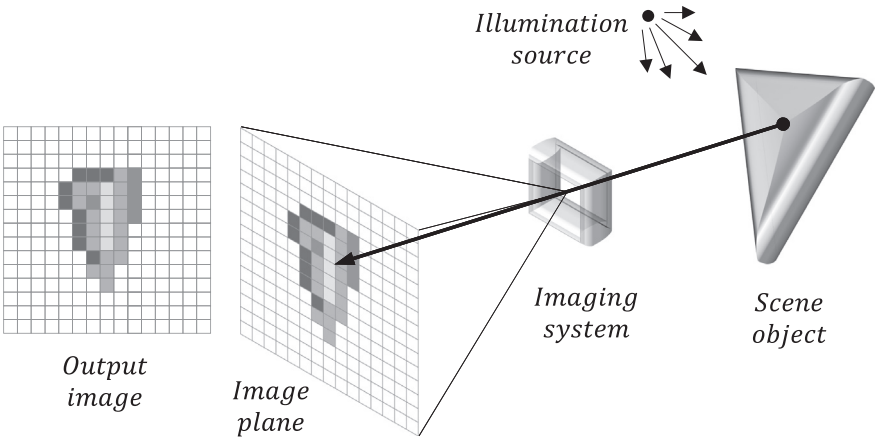
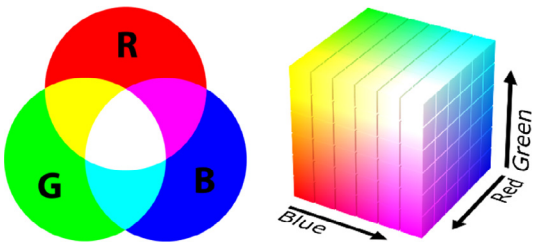


Fig. 2. Digital image formation.



(a) The RGB color mixing. (b) The RGB cube representation.

Fig. 3. The RGB color model. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.).

2.3.2. YUV

The YUV model was created as a re-coding of the RGB standard to minimize the bandwidth of analogue color television transmissions and to provide downwards compatibility with black-and-white television. The YUV standard comprises the *luminance* (Y) and the color differences (U, V) or *chrominance* components. The luminance is calculated as a weighted sum of red, green and blue light intensity components. The color difference components U and V are formed by subtracting luminance from blue and red color intensities, respectively. The Y, U and V components are computed as [11]

$$Y = 0.299R + 0.587G + 0.114B, \tag{1}$$

$$U = 0.492(B - Y), \tag{2}$$

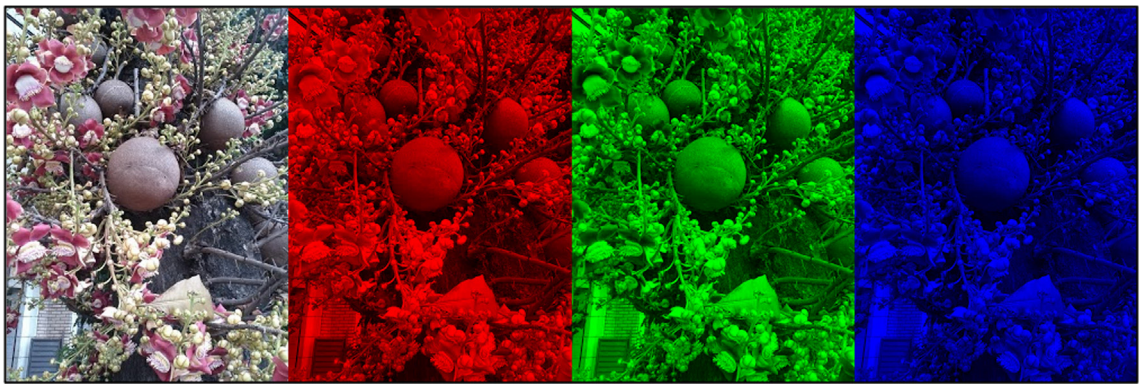


Fig. 4. The RGB components of an image.

$$V = 0.877(R - Y). \tag{3}$$

The Y component is important in computer vision applications because it provides a compact grayscale representation of RGB colors. Other color models used in video and television applications derive from the YUV standard, such as the YPbPr, YCbCr and YIQ standards [11].

2.3.3. CMYB

The *cyan*, *magenta*, *yellow* and *black* (CMYB) standard is used in color printing devices. The black color component is defined separately due to the high demand of such a color in text printing applications.

2.3.4. HLS and HSV

The *hue*, *lightness*, *saturation* (HLS) and the *hue*, *saturation*, *value* (HSV) color models were developed to approximate the human perception and interpretation of colors. Individual colors, in such models, can be represented by points in an inverted hexcone, as shown in Fig. 5. Hue is an angular measure that varies from 0 to 360, beginning and ending with red and running through green, blue and all intermediary colors. Saturation indicates the degree to which hue differs from a neutral gray. Saturation values run from 0, which means no color saturation, to 1, which is the highest saturation for a given hue at a given illumination. The intensity components, lightness in HLS and value in HSV, indicate the illumination level. Both vary from 0 (black, no light) to 1 (white, full illumination). The difference between them is that saturated colors have lightness 0.5 in the HSL model while in HSV their value is 1.

Although popular in most of digital image processing applications, the color models presented herein are only a small set of a greater variety of standards defined for different intents and purposes [11].

3. Image processing techniques

This section introduces a selection of popular image processing techniques including graphical examples that are fundamental to the robotics and instrumentation applications presented in Sections 4–7.

3.1. Correction of lens distortion

Camera lenses introduce non-linear geometric distortions to captured images, that can be numerically corrected by recalculation of the relative positions of the pixels in the image matrix. Radial and tangential distortion coefficients calibrate the horizontal and vertical positions of a pixel of coordinates (x_d, y_d) [12,13]. The distortion-corrected coordinates (x_u, y_u) of a pixel are calculated as

$$x_u = (x_d - \bar{x}_d)K(r) + \left[P_1\left(r^2 + 2(x_d - \bar{x}_d)^2\right) + 2P_2(x_d - \bar{x}_d)(y_d - \bar{y}_d)\right]P(r), \tag{4}$$

$$y_u = (y_d - \bar{y}_d)K(r) + \left[P_2\left(r^2 + 2(y_d - \bar{y}_d)^2\right) + 2P_1(x_d - \bar{x}_d)(y_d - \bar{y}_d)\right]P(r), \tag{5}$$

where

$$K(r) = 1 + \sum_{k=1}^{\infty} K_k r^{2k}, \tag{6}$$

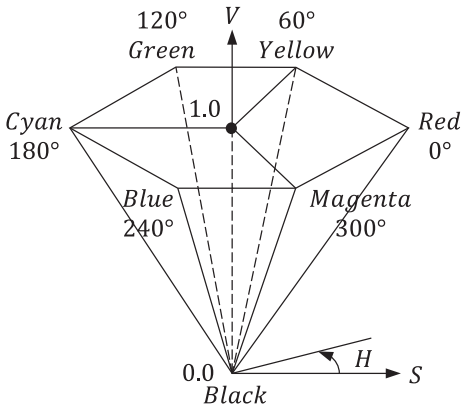


Fig. 5. The HSV hexcone representation.

$$P(r) = 1 + \sum_{k=3}^{\infty} P_k r^{2k(k-2)}, \tag{7}$$

$$r = \sqrt{(x_d - \bar{x}_d)^2 + (y_d - \bar{y}_d)^2}, \tag{8}$$

$\bar{x}_d$ and $\bar{y}_d$ are the coordinates of the *principal point* or image center and K_k and P_k are the radial and tangential distortion coefficients, respectively. Appropriate radial and tangential distortion coefficients can be determined by heuristic calibration. Fig. 6 depicts the result of correction of lens distortion of a dotted image pattern.

3.2. Histogram equalization

The different intensities of light captured by pixels in a digital image, should occupy preferably the whole range of available discrete levels, in order yield the best picture quality. In this sense, the minimum light intensities of an image should be ideally associated with the value zero and the maximum light intensities with the highest value of the discrete mapping range. An equally distributed image intensity range guarantees a good image contrast [14]. The histogram equalization of a pixel that occupies the position (x, y), with an initial associated intensity $I(x, y)$, is given by

$$I_e(x, y) = \frac{L-1}{N} C_f(I(x, y)), \tag{9}$$

where L is the number of possible intensity values, $C_f(I)$ is the cumulative frequency distribution of the pixel intensity I and N is the total number of pixels in the image. The cumulative frequency distribution is the accumulated sum of image histogram values in the range $[0, L-1]$.

Fig. 7 illustrates the difference in contrast after the equalization of a grayscale image. Fig. 8 shows the histogram and cumulative frequency distribution of the original and equalized images. In colored images, equalization is applied separately to the intensities of each primary composing color [14].

3.3. Erosion and dilation

Erosion and dilation are fundamental binary operations in the context of morphological image processing. *Mathematical morphology*, describes the algorithms used in the representation and description of geometric structures or region shapes. Morphological imaging is an effective tool in industrial applications, including product verification, precision measurement and robot vision. Morphological algorithms such as image boundary, skeleton and convex hull and image processing tools such as pruning, thinning and morphological filtering are based on primitive dilation and erosion operations. The concepts summarized in this section are based on the textbook of Gonzalez and Woods, Ref. [6].

Dilation and erosion are defined in terms of set translation and set reflection. The concept of *structuring element* is fundamental in these morphological definitions. A structuring element is a small image or sub-image with a specific shape used to test the structural properties of an image under study. The origin of a structuring element is commonly located at its center of gravity and is the reference point in the formulation of set reflection and translation.

The translation of a set B in $\mathbb{Z}^2$ by a point $z = (z_1,\ z_2)$, denoted by B_z, is defined as

$$B_z = \{p | p = b + z, \text{for } b \in B\}. \tag{10}$$

Fig. 9 illustrates the erosion operation. Given an image A and an arbitrary structuring element B, where A and B are sets in $\mathbb{Z}^2$, the erosion of A by B, denoted by $A \ominus B$, is defined as

$$A \ominus B = \{z | B_z \subseteq A\}. \tag{11}$$

The reflection of set B in $\mathbb{Z}^2$, denoted by $\widehat{B}$, is defined as

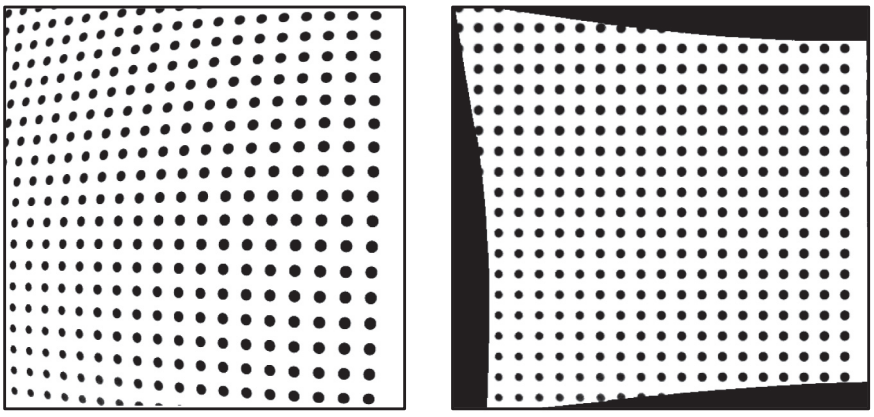
$$\widehat{B} = \{p | p = -b, \text{for } b \in B\}. \tag{12}$$

Fig. 10 depicts the dilation operation. The dilation of an image A by a structuring element B, denoted by $A \oplus B$, is defined as

$$A \oplus B = \left\{z | \widehat{B}_z \cap A \neq \varnothing\right\}. \tag{13}$$

3.4. Edge and shape detection

The detection of edges in an image is a convenient way to define regions of interest in order to perform identification and comparison of geometric forms, measurements, or to provide a simplified data structure to a complex image. Edge detection algorithms are based on the change of pixel intensities in specific image regions. The Canny's algorithm is one of the most popular and practical edge detection algorithms in digital image processing. Canny [15] has established important statistical characteristics of pixel intensity changes in order to precisely define the edge regions of an image. In Canny's edge detection,



(a) The dotted pattern. (b) The distortion-corrected image.

Fig. 6. Correction of lens distortion in a dotted pattern.

(a) Original image. (b) High contrast image.

Fig. 7. Histogram equalization in a grayscale image.

an image is convolved with a structuring element to obtain a blurred image that optimizes the edge detection process. The horizontal, vertical and diagonal changes of intensity in the blurred image provide the information about the edges of the original image. An efficient recursive implementation of Canny's edge detection can be found in an early work of Deriche [16].

Canny's edge detection requires a grayscale image and a Gaussian structuring element or *smoothing filter*. A typical Gaussian square structuring element of size 5 is shown in Eq. (14).

$$\mathbf{G} = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}. \tag{14}$$

Fig. 11 depicts the process of edge detection. Fig. 11a shows the grayscale image of a landscape. The image is convolved with a Gaussian square structuring element to obtain the smoothed image of Fig. 11b. The filtered image is mapped into a binary image by imposing a discriminant threshold on pixel intensities. Fig. 11c shows the output binary image. The regions with the most significant changes in intensity are the detected edges of the original image.

Shape detection is a complementary technique to extract information of shapes in an image. Shape detection methods search for regular patterns of interest, e.g. lines, circles and ellipses, that often appear in images of man-made structures (buildings, roads, rounded objects) and in nature (plants, fruits, cells, stars). The Hough transform [16,17] is one of the most popular and effective techniques for the detection of shapes in an image.

Hough transform is based on a *voting process* that identifies curves in a *space of parameters*. As an example, consider the parametrization of a line in the *normal form* $\rho = x\cos(\theta) + y\sin(\theta)$, where ρ is the length of a normal segment from the origin to the line and θ is the orientation of ρ with respect to the x-axis. A given point in the line of coordinates $(x_i,\ y_i)$ in the xy-

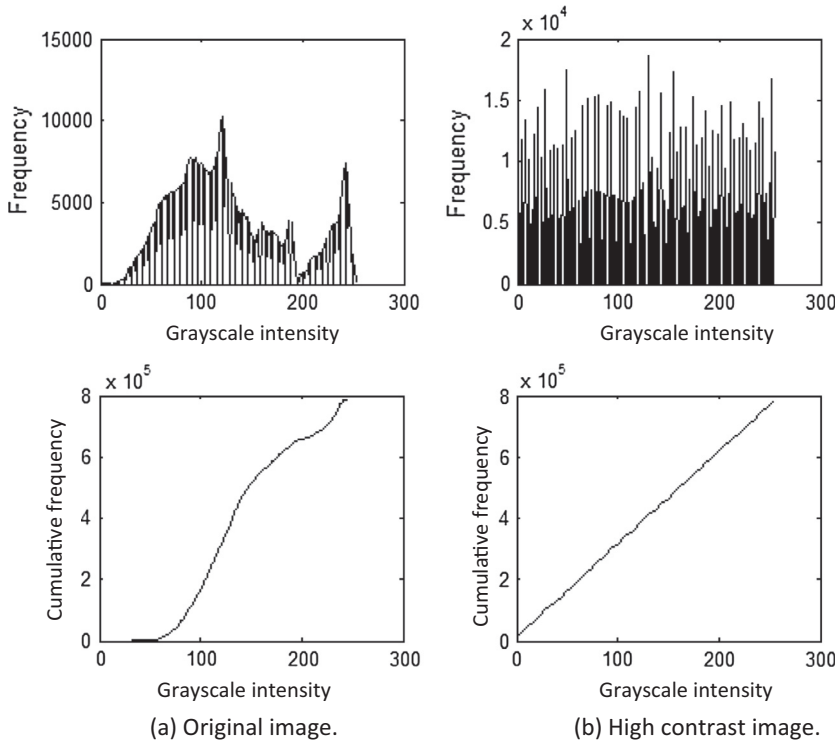


Fig. 8. Histogram and cumulative frequency distributions.

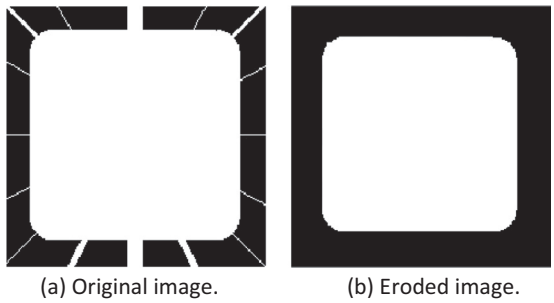


Fig. 9. Erosion of the radial lines with a square structuring element of 14-pixel size.

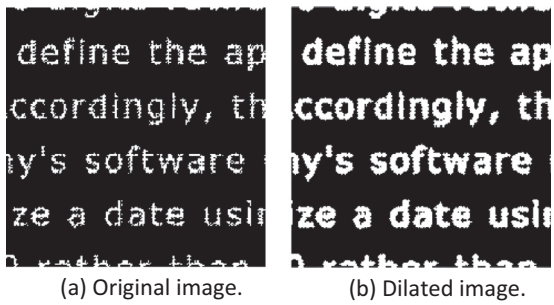


Fig. 10. Dilation of a text with gaps of 1 and 2-pixel size using a 3×3 cross-shaped structuring element.

(a) Original image. (b)Smoothed Image. (c) Canny's edges.

Fig. 11. The edge detection process.

space maps to a sinusoidal curve of parameters $(\theta_i,\ \rho_i)$ of the $\theta\rho$ space. A different point in the line of coordinates $(x_j,\ y_j)$ in the *xy*-space describes another sinusoidal curve of parameters $\left(\theta_j,\ \rho_j\right)$ in the $\theta\rho$ space. The curves in the $\theta\rho$ space will cross at a certain point of coordinates $\left(\theta',\ \rho'\right)$ that defines the parameters of the line that passes through both points in the *xy*-space. In computer realization, the $\theta\rho$ space is a square tessellation of fixed intervals or accumulator cells, usually $-90° < \theta < 90°$ and $-D < \rho < D$, where D is the main diagonal of the image. An accumulator cell increments its count (or vote) by one, each time a sinusoidal curve passes within. The cell with the highest *votes* defines the desired line parametrization.

Fig. 12 illustrates line detection using the Hough transform. Edges detected from the original image (Fig. 12a) are mapped to discretized sinusoidal curves in the parameter space, as shown in Fig. 12b. Two sets of crossing points with the greatest incidence in the parameter space, highlighted in Fig. 12b are selected to yield the identified lines shown in Fig. 12c (thicker lines).

3.5. Geometry of perspective projections

The *geometry of perspective projections* is a fundamental concept in photogrammetry for the use of cameras as measuring devices. The geometry of a perspective projection is based on a modified version of the pinhole camera model of Fig. 1, where the image plane is in front of the pinhole rather than behind it, to allow for non-inverted projections. Three reference frames are defined in such a context. The *world reference frame* is an arbitrary coordinates system that describes objects in the 3-D space. The *camera reference frame* is a 3-D coordinates system located at the projection center of the camera. The *image reference frame* is a 2-D coordinates system located at the upper-left corner of the image projection plane. The coordinates of the world and camera reference frames are expressed in units of length, whereas the coordinates of the image reference frame are expressed in units of pixel.

A point $\boldsymbol{P}$ in the 3-D space is expressed in the world reference frame as

$$\boldsymbol{P}_w^{\mathrm{T}} = [X \quad Y \quad Z]. \tag{15}$$

Point $\boldsymbol{P}$ is expressed in the camera reference frame as

$$\boldsymbol{P}_c^{\mathrm{T}} = [x \quad y \quad z]. \tag{16}$$

(a) Original image. (b) Parameter space $\theta\rho$. (c) Detected lines.

Fig. 12. Detection of lines with the Hough transform.

The perspective projection of $\boldsymbol{P}$, denoted by $\boldsymbol{p}$, is expressed in the camera reference frame as

$$\boldsymbol{p}^{\intercal} = [\,x' \quad y' \quad f\,], \tag{17}$$

where f is the focal length.

The pixel mapping of point $\boldsymbol{P}$, denoted by $\overline{\boldsymbol{p}}$, is expressed in the image reference frame as

$$\overline{\boldsymbol{p}}^{\intercal} = [\,u \quad v\,]. \tag{18}$$

Fig. 13 shows a representation of world, camera and image reference frames and point $\boldsymbol{P}$ with its perspective projection $\boldsymbol{p}$. Rotation $\mathbf{R}$ and translation $\boldsymbol{t}$ define the *extrinsic parameters* of the camera. Rotation $\mathbf{R}$ represents the orientation of the camera reference frame with respect to the world and is defined by the 3×3 rotation matrix

$$\mathbf{R} = \begin{bmatrix} \cos\theta_Z & \sin\theta_Z & 0 \\ -\sin\theta_Z & \cos\theta_Z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_Y & 0 & -\sin\theta_Y \\ 0 & 1 & 0 \\ \sin\theta_Y & 0 & \cos\theta_Y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_X & \sin\theta_X \\ 0 & -\sin\theta_X & \cos\theta_X \end{bmatrix}, \tag{19}$$

where θ_X, θ_Y and θ_Z are the angles of rotation of the axes of world coordinates. Translation $\boldsymbol{t}$ describes the position of the origin of world coordinates with respect to the camera reference frame.

The transformation of coordinates of a point in the world into the camera reference frame is given by

$$\boldsymbol{P}_c = \mathbf{R}^{\intercal}\boldsymbol{P}_w + \boldsymbol{t}. \tag{20}$$

The perspective projection of point $\boldsymbol{P}_c$ is expressed as

$$\boldsymbol{p} = \frac{f}{z}\boldsymbol{P}_c. \tag{21}$$

The pixel mapping of point $\boldsymbol{p}$ on the image plane is given by

$$\widehat{\boldsymbol{p}} = round\left(\frac{1}{f}\mathbf{A}\boldsymbol{p}\right), \tag{22}$$

where $\widehat{\boldsymbol{p}}$ is a vector of homogeneous coordinates. The vector of homogeneous coordinates is related to the pixel mapping vector as

$$\widehat{\boldsymbol{p}}^{\intercal} = [\,\overline{\boldsymbol{p}}^{\intercal} \quad 1\,] \tag{23}$$

and $\mathbf{A}$ is the matrix of *intrinsic parameters* of the camera, defined as

$$\mathbf{A} = \begin{bmatrix} S_x f & \gamma & u_0 \\ 0 & S_y f & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{24}$$

where S_x and S_y are the horizontal and vertical pixel densities (in units of pixel per length unit), u_0 and v_0 are the horizontal and vertical pixel coordinates of the principal optical axis respectively, and γ is the pixel skew parameter, which is zero for most of the digital cameras with a square pixel aspect ratio. Eq. (23) imposes the vector $\widehat{\boldsymbol{p}}$ to be of integer values through an integer rounding operation. This is because pixels in a digital image are discrete representations of the continuous intensities projected on the image plane of a camera. The theoretical aspects of perspective projection presented so far are fundamental for the definition of a camera calibration procedure described in the sequence.

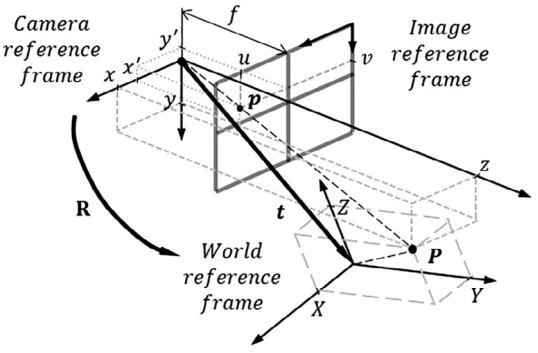


Fig. 13. The three coordinate systems of perspective projection.

3.6. Camera calibration

Camera calibration consists in identifying the intrinsic and extrinsic parameters of a camera with respect to a global reference frame. Calibration methods are often based on reference points of a calibration pattern to estimate the camera parameters. A calibration pattern is an object that defines a metric in a global reference frame. The projections of reference points in the camera's image plane enable the calibration process. A number of camera calibration approaches are found in literature [18], including a popular calibration method that uses a planar calibration chessboard, available as an open-source package [19].

The following method is based on the work of Kurka et al. [20], which uses a solid box of known dimensions as the calibration pattern. Fig. 14 depicts an image of the box captured by a digital camera. The box vertices are the calibration points, whose positions are determined using image processing techniques such as edge detection and geometry of perspective projection, as detailed in previous sections. The global coordinates system located at the box vertex 7 in Fig. 14 is arbitrarily regarded as the global reference frame.

Vertex points are represented by $\boldsymbol{B}_k$, with $k = 1, 2, \cdots 8$, and described in world coordinates as

$$\boldsymbol{B}_1 = \begin{bmatrix} 0 \\ H \\ 0 \end{bmatrix}, \boldsymbol{B}_2 = \begin{bmatrix} D \\ H \\ 0 \end{bmatrix}, \boldsymbol{B}_3 = \begin{bmatrix} D \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{B}_4 = \begin{bmatrix} D \\ 0 \\ W \end{bmatrix}, \boldsymbol{B}_5 = \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix}, \boldsymbol{B}_6 = \begin{bmatrix} 0 \\ H \\ W \end{bmatrix}, \boldsymbol{B}_7 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{B}_8 = \begin{bmatrix} D \\ H \\ W \end{bmatrix}, \tag{25}$$

where D, H and W are the known parameters of depth, height and width of the box, respectively.

The global referenced box vertex point $\boldsymbol{B}_k$, expressed in camera coordinates, $\boldsymbol{P}_k$, is written as

$$\boldsymbol{P}_k = \mathbf{R}^{\mathrm{T}}\boldsymbol{B}_k + \boldsymbol{t}. \tag{26}$$

The same point $\boldsymbol{B}_k$, expressed in homogeneous image coordinates $\widehat{\boldsymbol{p}}_k$ using Eqs. (22) and (26), is given by

$$\widehat{\boldsymbol{p}}_k = \frac{1}{z_k}\mathbf{A}(\mathbf{R}^{\mathrm{T}}\boldsymbol{B}_k + \boldsymbol{t}), \tag{27}$$

where z_k is the depth or z-axis coordinate of point $\boldsymbol{p}_k$ in the camera frame of references.

In a practical calibration procedure, the only known quantities of Eq. (27) are the integer valued pixel coordinates u_k and v_k of vector $\widehat{\boldsymbol{p}}_k$, and the world coordinates X_k, Y_k and Z_k of vertex $\boldsymbol{B}_k$. Camera calibration consists in estimating the unknown quantities of Eq. (27), which are, the depth z_k, intrinsic calibration matrix $\mathbf{A}$, rotation matrix $\mathbf{R}$, and translation vector $\boldsymbol{t}$, which is the origin position of the world reference frame.

It can be seen in [20], that the basic parameter identification relation can be drawn for a set of three different box vertices, namely points $\boldsymbol{B}_{ka}$, $\boldsymbol{B}_{kb}$ and $\boldsymbol{B}_{kc}$, and their respective homogenous image projections $\widehat{\boldsymbol{p}}_{ka}$, $\widehat{\boldsymbol{p}}_{kb}$, $\widehat{\boldsymbol{p}}_{kc}$, as

$$\left[\boldsymbol{B}_{k_I}^{\mathrm{T}} \otimes \mathbf{I} - \boldsymbol{B}_{k_{II}}^{\mathrm{T}} \otimes \widehat{\mathbf{P}}_{k_I}\widehat{\mathbf{P}}_{k_{II}}^{-1}\right]\mathrm{vec}(\mathbf{A}_R) = 0, \tag{28}$$

where matrices $\widehat{\mathbf{P}}_{kj}$ are defined as

$$\widehat{\mathbf{P}}_{k_I} = \begin{bmatrix} 0.5\widehat{\boldsymbol{p}}_{k_a} & -\widehat{\boldsymbol{p}}_{k_b} & 0.5\widehat{\boldsymbol{p}}_{k_c} \end{bmatrix}, \tag{29}$$

$$\widehat{\mathbf{P}}_{k_{II}} = \begin{bmatrix} -\widehat{\boldsymbol{p}}_{k_a} & 0.5\widehat{\boldsymbol{p}}_{k_b} & 0.5\widehat{\boldsymbol{p}}_{k_c} \end{bmatrix}. \tag{30}$$

Vectors $\boldsymbol{B}_{kj}$ are defined as

$$\boldsymbol{B}_{k_I} = (0.5\boldsymbol{B}_{k_a} - \boldsymbol{B}_{k_b} + 0.5\boldsymbol{B}_{k_c}) \tag{31}$$

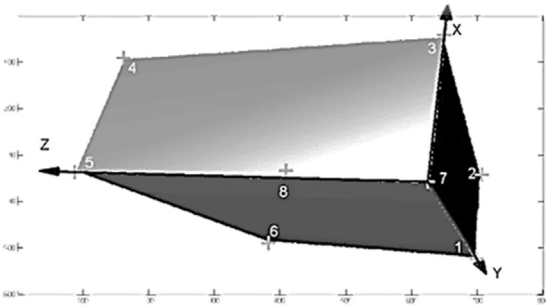


Fig. 14. Calibration box and the world reference frame.

and

$$\boldsymbol{B}_{k_{II}} = \left(-\boldsymbol{B}_{k_a} + 0.5\boldsymbol{B}_{k_b} + 0.5\boldsymbol{B}_{k_c}\right). \tag{32}$$

The Kronecker product is denoted by $\otimes$, matrix $\mathbf{A}_R$ is defined as

$$\boldsymbol{A}_R = \boldsymbol{AR}, \tag{33}$$

term $\mathbf{I}$ is the identity matrix of dimension 3×3, and the operation that stacks the columns of matrix $\boldsymbol{A}_R$, turning it into a vector of dimension 9×1, is denoted by vec( ).

Eq. (28) can be expanded, in the presence of $n \geq 3$ different sets of vertex coordinates vectors, and image coordinate projections, yielding the vertex related matrices $\boldsymbol{B}_{k_I}$ and $\boldsymbol{B}_{k_{II}}$, and their corresponding homogeneous image coordinates based matrices $\hat{\boldsymbol{P}}_{k_I}$ and $\hat{\boldsymbol{P}}_{k_{II}}$, resulting in

$$\boldsymbol{P}\bar{\mathbf{e}} = \boldsymbol{0}, \tag{34}$$

where

$$\boldsymbol{P} = \begin{bmatrix} \left(\boldsymbol{B}_{1_I}^{\mathsf{T}} \otimes \mathbf{I} - \mathbf{B}_{1_{II}}^{\mathsf{T}} \otimes \hat{\mathbf{P}}_{1_I}\hat{\mathbf{P}}_{1_{II}}^{-1}\right) \\ \left(\boldsymbol{B}_{2_I}^{\mathsf{T}} \otimes \mathbf{I} - \mathbf{B}_{2_{II}}^{\mathsf{T}} \otimes \hat{\mathbf{P}}_{2_I}\hat{\mathbf{P}}_{2_{II}}^{-1}\right) \\ \vdots \\ \left(\boldsymbol{B}_{n_I}^{\mathsf{T}} \otimes \mathbf{I} - \mathbf{B}_{n_{II}}^{\mathsf{T}} \otimes \hat{\mathbf{P}}_{n_I}\hat{\mathbf{P}}_{n_{II}}^{-1}\right) \end{bmatrix}, \tag{35}$$

and $\bar{\mathbf{e}}$ is the representation for vec($\mathbf{A}_R$).

Eq. (35) is a system of $3n \times 9$ homogeneous equations, which finds solutions in the vectors of the right null space of $\mathbf{P}$. In practical cases, where the image projections of the spatial vertex points are discrete pixel positions, matrix $\mathbf{P}$ will not have a null space, and $\bar{\boldsymbol{e}}$ will be estimated as the right singular vector of $\mathbf{P}$ associated with its smallest singular value.

The estimation vector is reshaped into a matrix of dimension 3×3, which is an approximation of matrix $\mathbf{A}_R$, that is

$$\bar{\mathbf{A}}_R = \text{reshape}(\bar{\boldsymbol{e}}, 3 \times 3). \tag{36}$$

Matrix $\bar{\mathbf{A}}_R$ is also an estimation of the product of the intrinsic calibration and rotation matrices, according to Eq. (33). The orthogonal property of the rotation matrix allows manipulation of Eq. (33) to yield an estimation of the self-transpose product of the calibration matrix, that is

$$\mathbf{ARR}^{\mathsf{T}}\mathbf{A}^{\mathsf{T}} = \mathbf{A}^{\mathsf{T}} \simeq \widetilde{\mathbf{A}}_{\mathrm{R}}\widetilde{\mathbf{A}}_{\mathrm{R}}^{\mathsf{T}} \tag{37}$$

Matrix $\widetilde{\mathbf{A}}_R$ of Eq. (37) is the normalized version of $\bar{\mathbf{A}}_R$, forcing its third row, third column element to be 1, similar to the likewise element of $\mathbf{AA}^{\mathsf{T}}$.

Estimations of the intrinsic camera parameters are derived from the self-transpose product of matrix $\widetilde{\mathbf{A}}_R$, according to Eqs. (37) and (24)

$$\widetilde{S}_x f \simeq \sqrt{\left(\text{element of row 1, column 1 of } \widetilde{\mathbf{A}}_R\widetilde{\mathbf{A}}_R^{\mathsf{T}}\right) - \widetilde{\mu}_0^2}, \tag{38}$$

$$\widetilde{S}_y f \simeq \sqrt{\left(\text{element of row 2, column 2 of } \widetilde{\mathbf{A}}_R\widetilde{\mathbf{A}}_R^{\mathsf{T}}\right) - \widetilde{v}_0^2}, \tag{39}$$

$$\widetilde{\mu}_0\widetilde{S}_x f \simeq \text{element of row 1, column 3 of } \widetilde{\mathbf{A}}_R\widetilde{\mathbf{A}}_R^{\mathsf{T}}, \tag{40}$$

$$\widetilde{v}_0\widetilde{S}_y f \simeq \text{element of row 2, column 3 of } \widetilde{\mathbf{A}}_R\widetilde{\mathbf{A}}_R^{\mathsf{T}}, \tag{41}$$

A preliminary estimate for the rotation matrix can be obtained from Eq. (33) as

$$\widetilde{\mathbf{R}}' = \widetilde{\mathbf{A}}^{-1}\widetilde{\mathbf{A}}_R, \tag{42}$$

where

$$\widetilde{\mathbf{A}} = \begin{bmatrix} \widetilde{S}_x f & 0 & \widetilde{\mu}_0 \\ 0 & \widetilde{S}_y f & \widetilde{v}_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{43}$$

The final estimate for the rotation matrix, denoted by $\widetilde{\mathbf{R}}$, is obtained by normalizing the singular values of $\widetilde{\mathbf{R}}'$, that is

$$\widetilde{\mathbf{R}} = \mathbf{U}_R\mathbf{V}_R^{\mathsf{T}}, \tag{44}$$

where $\mathbf{U}_R$ and $\mathbf{V}_R$ are matrices with the left and right singular vectors of $\widetilde{\mathbf{R}}'$.

The translation vector $\boldsymbol{t}$ can be estimated simultaneously with a vector of depth projection values by applying Eq. (27) to a number $n \geq 2$ of pairs of spatial and projected points, via the least-squares solution of a linear system of equations

$$\begin{bmatrix} \hat{\boldsymbol{p}}_1 & 0 & \dots & 0 & -\widetilde{\mathbf{A}} \\ 0 & \hat{\boldsymbol{p}}_2 & \dots & 0 & -\widetilde{\mathbf{A}} \\ \vdots & \vdots & \ddots & \vdots & -\widetilde{\mathbf{A}} \\ 0 & 0 & \dots & \hat{\boldsymbol{p}}_n & -\widetilde{\mathbf{A}} \end{bmatrix} \begin{bmatrix} \widetilde{\boldsymbol{\lambda}} \\ \widetilde{\boldsymbol{t}} \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{A}}_R \boldsymbol{B}_1 \\ \widetilde{\mathbf{A}}_R \boldsymbol{B}_2 \\ \vdots \\ \widetilde{\mathbf{A}}_R \boldsymbol{B}_n \end{bmatrix}, \tag{45}$$

where $\widetilde{\boldsymbol{\lambda}}$ is the vector of estimated depth projections defined as

$$\widetilde{\boldsymbol{\lambda}}^{\mathsf{T}} = [\,\mathrm{z}_1 \quad \mathrm{z}_2 \quad \dots \quad \mathrm{z}_n\,]. \tag{46}$$

3.7. Epipolar geometry

Epipolar geometry is the geometry of stereoscopic vision. Stereo computer vision refers to the imaging system consisting of a pair of cameras that emulate the human binocular vision capability. The visual perception of depth in stereo systems is possible due to *disparity*, which is the difference in apparent position of objects in a scene observed by two independent cameras. Epipolar geometry relates the concept of visual disparity to objects or physical points on a 3-D environment. The process of retrieving depth information is known as *stereo image matching*, which consists of finding pixel correspondences in two images and converting their 2-D positions into 3-D coordinates. The textbook of Trucco and Verri [5] dedicates a chapter to stereo vision and its geometric relationships. The main aspects of such subjects are introduced in the sequence.

Fig. 15 shows the geometry of a stereo vision system. The stereo system consist of two pinhole cameras (left and right cameras) with projection centers O_l and O_r and image planes π_l and π_r. The 3-D reference frame of each camera is located at the corresponding projection center and the z-axis coincides with the optical axis in both cases. The line that connects both camera projection centers is called the *baseline*. The baseline intersects the image planes π_l and π_r at points $\boldsymbol{e}_l$ and $\boldsymbol{e}_r$, called *epipoles*. A given point $\boldsymbol{P}$ in the 3-D space is described by vector $\boldsymbol{P}_l{}^{\mathsf{T}} = [x_l, y_l, z_l]$, or $\boldsymbol{P}_r{}^{\mathsf{T}} = [x_r, y_r, z_r]$ and generates the projections $\boldsymbol{p}_l{}^{\mathsf{T}} = [x_l', y_l', f_l]$ and $\boldsymbol{p}_r{}^{\mathsf{T}} = [x_r', y_r', f_r]$ of camera coordinates, where f_l and f_r are focal lengths. The epipolar lines $\boldsymbol{u}_l$ and $\boldsymbol{u}_r$ connect the epipoles to the projections. Such a geometry defines the *epipolar constraint*, which restricts the match of a point in the image of a camera to the epipolar line in the image of the opposite camera. The epipolar plane π_P is defined by point $\boldsymbol{P}$ and the centers of projection O_l and O_r of the cameras.

Translation vector $\boldsymbol{t} = (O_r - O_l)$ and rotation matrix $\mathbf{R}$ are the extrinsic parameters that relate the reference frames of the left and right cameras. The relation between $\boldsymbol{P}_l$ and $\boldsymbol{P}_r$ is given by

$$\boldsymbol{P}_l = \mathbf{R}^{\mathsf{T}} \boldsymbol{P}_r + \boldsymbol{t}. \tag{47}$$

An interest point $\boldsymbol{P}$ and its projections are related by the perspective projection equations

$$\boldsymbol{p}_l = \frac{f_l}{z_l} \boldsymbol{P}_l \tag{48}$$

and

$$\boldsymbol{p}_r = \frac{f_r}{z_r} \boldsymbol{P}_r. \tag{49}$$

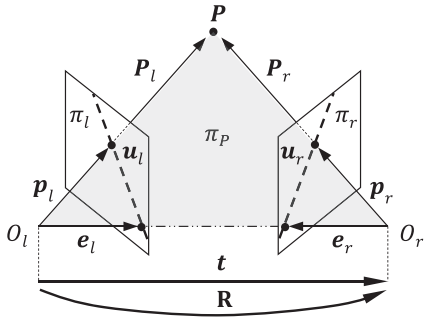


Fig. 15. The epipolar geometry.

The essential matrix **E**, in camera coordinates, establishes the epipolar constraint between left and right projections, such as

$$\boldsymbol{p}_r^{\intercal}\mathbf{E}\boldsymbol{p}_l = 0, \tag{50}$$

$$\mathbf{E} = \widehat{\mathbf{T}}\mathbf{R}^{\intercal}, \tag{51}$$

where $\widehat{\mathbf{T}}$ is the skew-symmetric matrix representation of the cross product operator for $\boldsymbol{t}$, that is $''\boldsymbol{t}\times''$. The product $\mathbf{E}\boldsymbol{p}_l$ can be thought of as a vector perpendicular to the epipolar plane, projected in the right camera system of coordinates. The dot product of vector $\boldsymbol{p}_r$, in the epipolar plane π_P, and $\mathbf{E}\boldsymbol{p}_l$, is thus null.

Points $\widehat{\boldsymbol{p}}_l$ and $\widehat{\boldsymbol{p}}_r$ are the pixel mappings of homogenous coordinates corresponding to $\boldsymbol{p}_l$ and $\boldsymbol{p}_r$, such as

$$\boldsymbol{p}_l = f_l\mathbf{A}_l^{-1}\widehat{\boldsymbol{p}}_l, \tag{52}$$

$$\boldsymbol{p}_r = f_r\mathbf{A}_r^{-1}\widehat{\boldsymbol{p}}_r, \tag{53}$$

where $\mathbf{A}_l$ and $\mathbf{A}_r$ are the matrices of intrinsic parameters of the left and right cameras, respectively.

The fundamental matrix **F**, of image coordinates, establishes the epipolar constraint between left and right projections as

$$\widehat{\boldsymbol{p}}_r^{\intercal}\mathbf{F}\widehat{\boldsymbol{p}}_l = 0, \tag{54}$$

$$\mathbf{F} = \mathbf{A}_r^{-\intercal}\mathbf{E}\mathbf{A}_l^{-1}. \tag{55}$$

Matrices **E** and **F** can be estimated using the eight-point algorithm described in [5].

3.7.1. Extrinsic parameters of the stereo system

Fig. 16 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 **R** and $\boldsymbol{t}$ are calculated with respect to the reference frame of the left camera, such as

$$\mathbf{R} = \mathbf{R}_r^{\intercal}\mathbf{R}_l, \tag{56}$$

$$\boldsymbol{t} = \boldsymbol{t}_l - \mathbf{R}^{\intercal}\boldsymbol{t}_r. \tag{57}$$

3.7.2. Stereo correspondence

The reconstruction of the 3-D scene depends on the solution of the correspondence problem or disparity estimation. Fig. 17 depicts the stereo correspondence problem. The position of a point $\boldsymbol{P}$ in space is recovered from its projections $\boldsymbol{p}_l$ and $\boldsymbol{p}_r$. The coordinates x_l' and x_r' are the projections referred to the principal points c_l and c_r, t is the baseline and z is the straight-line distance from the point $\boldsymbol{P}$ to the baseline.

From the construction of the problem,

$$z = f\frac{t}{d}, \tag{58}$$

$$d = x_l' - x_r', \tag{59}$$

where the disparity d is the difference in horizontal image position between corresponding points in both images and is inversely proportional to the depth z, for parallel cameras, as shown in Fig. 17.

Algorithms to find correspondence pairs, according to [5], can be classified in two main classes: correlation-based and feature-based. Correlation-based algorithms operate on the complete set of image points and feature-based methods on a sparse set of image features. A comprehensive review on stereo correspondence methods is found in [2].

3.7.2.1. Correlation-based methods. Correlation-based methods search for correspondences over image windows of fixed size. The similarity criterion is established by a measure of the inter-window correlation in the two images. The window that maximizes this criterion contains corresponding elements.

3.7.2.2. Feature-based methods. Feature-based methods focus on the search of *image descriptors* that describe particular characteristics of the images. The similarity criterion is established through a measure of the distance between descriptors. The corresponding elements are given by the pair of descriptors with minimum distance. Appropriate constraints, such as the epipolar restriction, improve the performance of correspondence methods by imposing a condition on the search and reducing false matches and occlusions (points with no counterpart in the opposite image). The correlation technique discussed herein are present in a feature-based method called the Scale Invariant Feature Transform (SIFT), which is presented in Section 3.8.

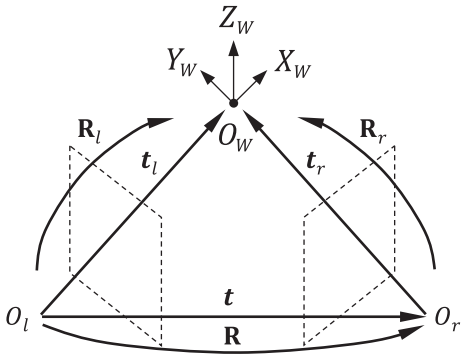


Fig. 16. The extrinsic parameters of a stereo system.

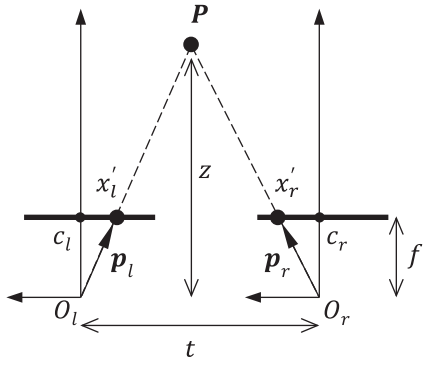


Fig. 17. Depth estimation from disparity of corresponding points.

3.7.3. 3-D reconstruction by triangulation

Fig. 18 depicts the 3-D reconstruction of a point $\boldsymbol{P}$ in space by triangulation, that is, by finding the closest point between projective directions $\boldsymbol{l} = (O_l, p_l)$ and $\boldsymbol{r} = (O_r, p_r)$. Although the projection lines of a point in two different cameras should intersect in space, it does not occur in a practical 3-D reconstruction application. This is due to the fact that, in a practical application, the position of the corresponding pair $\boldsymbol{p}_l$ and $\boldsymbol{p}_r$ and the camera parameters are only approximate estimations. Therefore, the 3-D reconstruction problem can be solved by calculating depths z_l and z_r, in an optimal way, such as to minimize a spatial position difference error vector, that is

$$\boldsymbol{e} = z_l \boldsymbol{p}_l - z_r \mathbf{R}^\top \boldsymbol{p}_r - \boldsymbol{t}, \tag{60}$$

$$\begin{bmatrix} z_l \\ z_r \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_l^\top \boldsymbol{p}_l & -\boldsymbol{p}_l^\top \mathbf{R}^\top \boldsymbol{p}_r \\ -\boldsymbol{p}_l^\top \mathbf{R}^\top \boldsymbol{p}_r & \boldsymbol{p}_r^\top \boldsymbol{p}_r \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{p}_l^\top \boldsymbol{t} \\ -\boldsymbol{p}_r^\top \mathbf{R} \boldsymbol{t} \end{bmatrix}, \tag{61}$$

where $\boldsymbol{e}$ is the position difference error vector and $z_l \boldsymbol{p}_l$ and $z_r \mathbf{R}^\top \boldsymbol{p}_r$ parametrize projection lines $\boldsymbol{l}$ and $\boldsymbol{r}$, respectively. The depth values calculated according to Eq. (61), minimize the squared error $\boldsymbol{e}^\top \boldsymbol{e}$. The optimally reconstructed 3-D intersection point is $\boldsymbol{P}'$, located at the middle of error vector Other reconstruction scenarios are discussed in [2,5].

3.8. Scale invariant feature transform

The Scale Invariant Feature Transform (SIFT) is a feature-based method for stereo correspondence developed by Lowe [21]. The SIFT algorithm searches for image points of interest or keypoints based on a scale-space [22]. The keypoints are invariant to affine transformations, illumination changes and additive Gaussian noise [23]. The SIFT algorithm consists of four major stages described in the sequence. The scale-space extrema detection stage searches for candidate keypoints in the scale-space of a Difference of Gaussian (DOG) function. The keypoint localization stage filters out candidates based on criteria such as minimum contrast or minimum ratio of principal curvature. The orientation assignment stage adds a vector of orientation to a keypoint in order to achieve rotational invariance. The keypoint descriptor stage calculates a local image descriptor that remains invariant to additional changes in illumination and 3-D viewpoint.

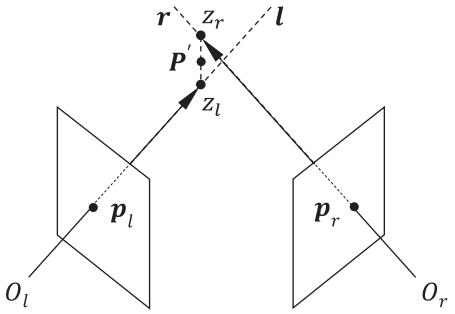


Fig. 18. The 3-D reconstruction by triangulation.

3.8.1. Scale-space extrema detection

The SIFT algorithm starts by detecting candidate keypoint locations that preserve their geometric characteristics, despite changes in image scale. The keypoint search is performed on a DOG scale-space, which is a function containing all possible image scales. A scale-space $L(x, y, \sigma)$ is defined by the convolution of image $I(x, y)$ with a Gaussian smoothing filter $G(x, y, \sigma)$, that is

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \tag{62}$$

where

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\left(x^2+y^2\right)/2\sigma^2}, \tag{63}$$

and σ is the standard deviation or blurring factor. The DOG function is an approximation of the Laplacian of the Gaussian function $\sigma^2\nabla^2 G$, which is an appropriate space to search for image invariance to scale [22]. The DOG between two images with distortion scales $k\sigma$ and σ is given by

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma). \tag{64}$$

The DOG is also calculated for scale spaces of *octave* resampled versions of the original image. Octaves are calculated by doubling the value of the distorting factor σ. The image is resampled by 50% in the x and y dimensions, when moving through octaves. Keypoint identification or *extrema detection* is performed on a *3-D Moore neighborhood*, which is the local pixel neighborhood at the same scale-space and up and down the octaves. Fig. 19 illustrates the local pixel search on a 3-D Moore neighborhood of unitary radius. A pixel is considered *extrema* or keypoint candidate if its intensity value is a minimum or maximum compared with its neighbors.

3.8.2. Keypoint localization

The location of candidate keypoints is refined to discard low contrast or edge points. Keypoint location is interpolated using the quadratic Taylor expansion of the DOG at sample point $\boldsymbol{x}^\intercal = [x, y, \sigma]$, such as

$$D(\boldsymbol{x}) = D + \left(\frac{\partial D}{\partial \boldsymbol{x}}\right)^\intercal \boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^\intercal \frac{\partial^2 D}{\partial \boldsymbol{x}^2}\boldsymbol{x}. \tag{65}$$

An optimal keypoint offset $\widehat{\boldsymbol{x}}$ is obtained by minimizing Eq. (65), which results in

$$\widehat{\boldsymbol{x}} = -\left(\frac{\partial^2 D}{\partial \boldsymbol{x}^2}\right)^{-1} \frac{\partial D}{\partial \boldsymbol{x}}. \tag{66}$$

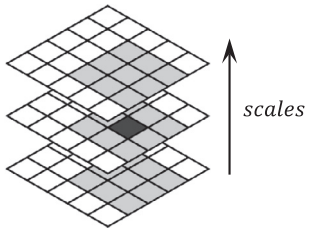


Fig. 19. The 3-D Moore neighborhood (marked in gray) of a pixel (marked in black).

If any component of $\hat{\boldsymbol{x}}$ is greater than 0.5, it means that the extrema is closer to a different sample point. In such a case, the interpolation is calculated again at the new sample point. The final keypoint location results from adding the offset $\hat{\boldsymbol{x}}$ to the original location of corresponding sample point. Furthermore, low contrast extrema are discarded by using the minimum threshold discrimination criteria $D(\hat{\boldsymbol{x}}) < 0.03$. Such a threshold expects image pixel values on the interval $[0, 1]$. Poorly defined responses at the edges are also neglected for instability by setting a threshold on the ratio of principal curvatures

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}, \tag{67}$$

where $r = \alpha/\beta$, is the ratio between the largest and the smallest magnitude eigenvalues of matrix $\mathbf{H}$, and $\mathbf{H}$ is the 2×2 Hessian matrix computed at the scale and location of the keypoint, such as

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}, \tag{68}$$

where the matrix elements are the second-order partial derivatives of $D(\boldsymbol{x})$ estimated by difference of adjacent sample points.

3.8.3. Orientation assignment

The keypoint receives an orientation angle that describes its local pixel neighborhood in terms of direction and provides invariance to rotation. Keypoint orientation is calculated from the local gradients of the smoothed image L at the initial scale σ. The local gradients of magnitude $m(x, y)$ and orientation $\theta(x, y)$ are calculated as

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}, \tag{69}$$

$$\theta(x,y) = \arctan\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right). \tag{70}$$

An orientation histogram is calculated from the orientation gradients of the keypoint and its neighborhood. The orientation histogram is a vector of 36 components corresponding to 10 degree equal divisions of a unit circle. The vector elements are weighted by the corresponding gradient of magnitude m and a Gaussian circular window with smoothing factor $\sigma = 1.5\sigma_{k_i}$, where σ_{k_i} belongs to the scale i of keypoint k. The final keypoint orientation is the peak value of the orientation histogram. Histogram peaks above 80% of the highest value are also considered, in order to create additional keypoints of same scale and location, but different orientation.

3.8.4. Keypoint descriptor

The keypoint descriptor defines local image regions that remain invariant to scale, orientation, illumination and 3-D viewpoint changes. The local gradients of magnitude and orientation of the keypoint and its neighborhood are rotated by the keypoint orientation angle to obtain rotation invariant metrics. The new gradients are weighted by a Gaussian window that emphasizes the effect of most influential gradients closer to the keypoint. The resulting keypoint descriptor is a square matrix whose elements are the magnitudes of eight different orientation histograms, calculated over local keypoint regions. Fig. 20 shows the 2×2 orientation histograms corresponding to a descriptor array calculated over a smoothed 8×8 keypoint sub region.

The SIFT feature vector is computed by concatenating the magnitude of all orientation histograms. The result is a vector of rn^2 components, where r is the number of histogram orientations and n is the size of the square features matrix. For instance, a keypoint vector of 128 components is the result of considering a 4×4 descriptor and 8 orientations. The output is converted to a unit vector, for further affine invariance to illumination, provided that contrast and illumination transitions

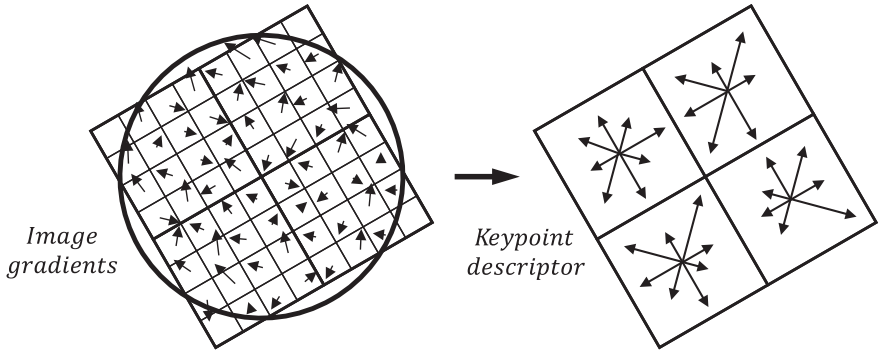


Fig. 20. Local image gradients weighted by a Gaussian window (overlaid circle) and the keypoint descriptor.

are considered as linear operations. Other non-linear changes in illumination are neglected because the main information of the keypoint descriptor is preserved in the distribution of the orientations rather than in its magnitude.

4. Application 1: Wheel rim detection

The detection of shapes in digital images has a wide variety of applications such as robot localization [6], object measurement [4], counting [5] and identification [3]. In this application, a method for the detection of rims in automotive wheel images is described, which is based on the Hough transform.

4.1. System setup

The system consists of a digital camera of 2976 × 1672 pixel resolution that captures images of an automotive wheel. Fig. 21a depicts a typical camera image. It can be noticed that the wheel image is affected by illumination (low contrast) and lens distortion (curved objects at edges). Histogram equalization (Section 3.2) and correction of lens distortion (Section 3.1) are performed sequentially to correct such effects. Fig. 21b shows the result after applying both corrections. The high contrast image results in more evenly distributed pixel intensities in the histograms, which are shown beneath the images in Fig. 21. The effect of correction of lens distortion is observed at the edges of the side objects in Fig. 21b.

4.2. Rim ellipse detection

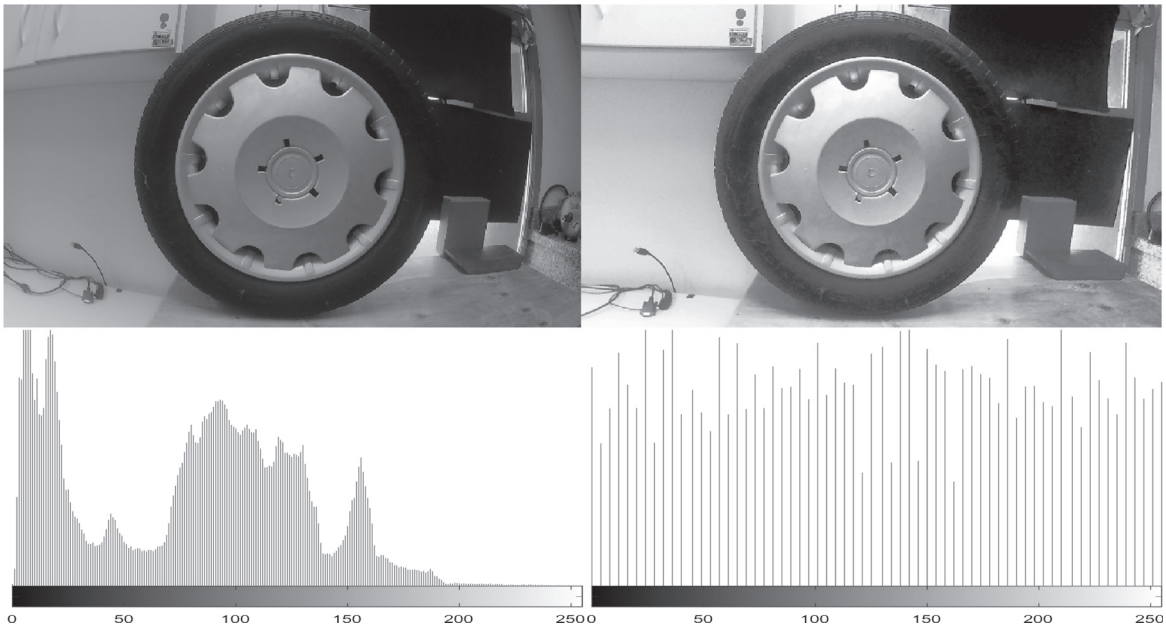
Circular wheel rims are projected as ellipses in the image plane due to the camera's viewpoint. The wheel's rim can thus be detected using a variation of the Hough transform (Section 3.4) applied to elliptical shapes [23,24]. Firstly, the RBG wheel image is converted to grayscale intensities using Eq. (1). Secondly, the Canny edge detector is computed over the grayscale image. Thirdly, the Hough transform is applied to detect the parameters of the rim ellipse of coordinates (u, v), defined as

$$\frac{(u_e)^2}{a^2} + \frac{(v_e)^2}{b^2} = 1, \tag{71}$$

$$u_e = (u - u_0)\cos\Theta + (v - v_0)\sin\Theta, \tag{72}$$

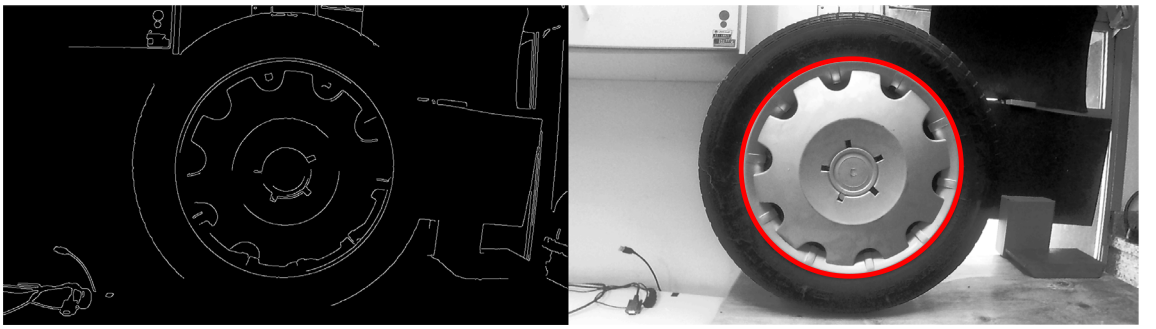
$$v_e = -(u - u_0)\sin\Theta + (v - v_0)\cos\Theta, \tag{73}$$

where u_0 and v_0 are the center coordinates, a and b are the lengths of semi-major and semi-minor axes and Θ is the axis rotation angle. Computational implementations of Hough's ellipse detection usually require estimates of minimum and maximum input range for the semi-major axis in order to optimize the search in the parameter space.



(a) Original image. (b) Histogram equalization and distortion correction.

Fig. 21. Wheel image preprocessing.



(a) Edge detection. (b) Ellipse detection.

Fig. 22. Wheel rim detection.

4.3. Experimental test results

Fig. 22 illustrates the wheel rim detection method. The image size is reduced to 40% of the total resolution to simplify the calculations. Fig. 22a shows the resulting image after applying Canny's edge detection. The semi-major axis of the ellipse on the edge of the rim is in the range $[430, 460]$ (pixels), which is a useful information to restrict the search of the ellipse in the parameter space. Fig. 22b represents the detected wheel rim ellipse in a colorful thick line. Results of wheel rim detection enable applications such as the measurement of wheel alignment angles, as will be shown in Section 6.

5. Application 2: Automatic 3-D crankshaft verification system

Image processing can be applied in industrial automation to solve complex tasks required for product quality assurance, such as pattern recognition, counting and verification. One of the advantages of computer-aided visual inspection is the thorough automatic examination of every manufactured component as opposed to human-based inspection, which often relies on statistical sampling. The following application introduces an automatic system for dimensional verification of automotive crankshafts using a multiple camera setup. The system is based on the results of a preliminary study [25] and a patent [37]. Fig. 23 characterizes the system, comprised of six cameras that capture simultaneous images of the crankshaft. The crankshaft has a length of 1.068 m, a width of 0.281 m, and a height of 0.242 m. System tests are performed on a simulated environment.

5.1. System setup and calibration

Fig. 24 depicts the multiple camera setup. Six cameras of 960×540 pixel resolution and 35 mm focal length are arranged in two parallel rows and oriented towards the crankshaft surface.

The system is comprised of the camera projection centers O_i, image planes π_i, focal lengths f_i translation vectors $\boldsymbol{t}_i$ and rotation matrices $\mathbf{R}_i$. The cameras are positioned at a 1.06 m height from the crankshaft base and rotated $\theta_x = 40°$ about their local x-axes towards the crankshaft surface. The origin of world coordinates O_W is arbitrarily located at the projection center of camera 1. The design parameters have the typical values $||\boldsymbol{t}_1|| = ||\boldsymbol{t}_2|| = ||\boldsymbol{t}_4|| = ||\boldsymbol{t}_5|| = 300$ mm, $||\boldsymbol{t}_3|| = 1600$ mm, $\mathbf{R}_1 = \mathbf{R}_2 = \mathbf{R}_4 = \mathbf{R}_5 = \mathbf{I}_3$, where $\mathbf{I}_3$ is the identity matrix of size 3, and $\mathbf{R}_3$ is a matrix of successive rotations $\theta_x = -80°$ and $\theta_z = 180°$ about local x and z axes.

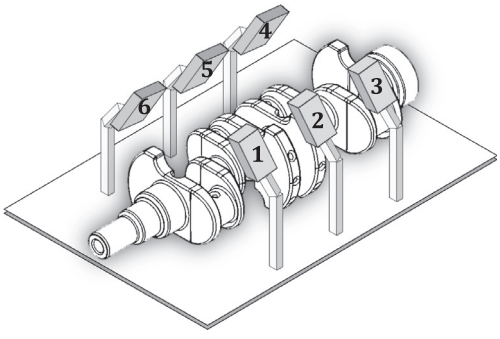


Fig. 23. The crankshaft verification system.

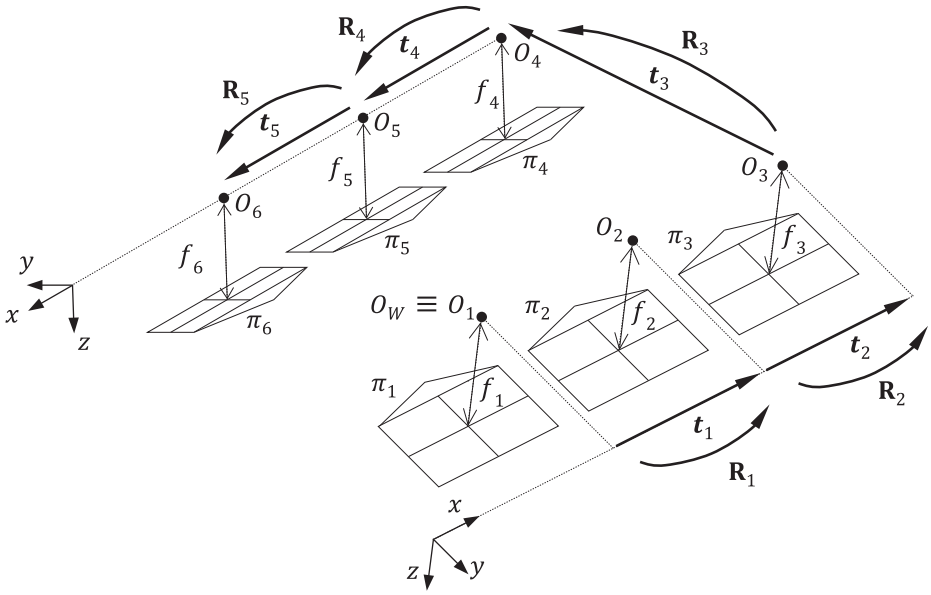


Fig. 24. The multiple camera setup.

Local calibrations are performed for the stereo pairs of cameras 1 and 2, 2 and 3, 4 and 5, and 5 and 6 to identify the local intrinsic and extrinsic camera parameters using the calibration procedure described in Section 3.6. The extrinsic parameters of the stereo subsystems $\boldsymbol{t}_i$ and $\mathbf{R}_i$ are calibrated from the latter local extrinsic parameters using Eqs. (56) and (57), as in the stereo system of Fig. 16. Tables 1 and 2 summarize the results of the calibration processes. Translations and rotations are expressed as vectors of relative coordinates for subsets of two consecutive cameras.

5.2. Image matching

Fig. 25 depicts the process of image matching between pairs of adjacent camera images to reconstruct the 3-D crankshaft surface using epipolar geometry, as described in Section 3.7. Stereo matching is evaluated specifically for the pairs of cameras 1 and 2, 2 and 3, 4 and 5, and 5 and 6 using the SIFT computational implementation of Vedaldi and Fulkerson [26].

Table 3 shows the result of the image matching process for each stereo subsystem. A total of 1319 keypoint correspondences were identified in such a particular scenario.

5.3. Crankshaft surface reconstruction

The reconstruction problem consists in recovering the geometric characteristics of the crankshaft, such as shape and dimensions, from the images of six cameras at different viewpoints. The 3-D local coordinates of an interest point in the object, denoted $\boldsymbol{P}_i$, are calculated by triangulation of a pair of corresponding keypoints using Eq. (61). The coordinates of $\boldsymbol{P}_i$, which are referred to the reference frame of the leftmost camera of the corresponding stereo pair, are converted to global coordinates in the world reference frame, denoted $\boldsymbol{B}$, by the composition of successive rotations and translations, according to

$$\boldsymbol{B} = \left(\prod_{k=1}^{i-1} \mathbf{R}_k^{\mathrm{T}}\right)\boldsymbol{P}_i + \sum_{k=1}^{i-1} \boldsymbol{t}_k, i = 2, 4, 5. \tag{74}$$

5.4. Simulation test results

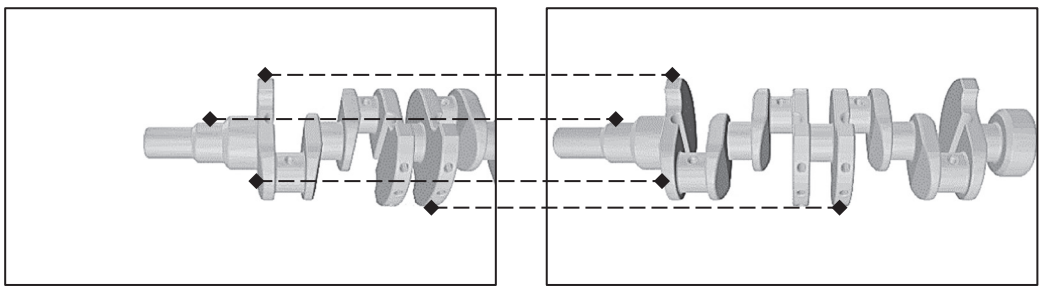
The camera setup and the crankshaft are simulated in the 3-D graphics freeware platform, Blender. The software enables the construction of a 3-D virtual model of a solid with a texturized surface, whose geometry is imported from a CAD solid model. The software also allows the creation of illumination sources and virtual cameras that capture rendered images of the solid. The present example uses virtual models of a texturized automotive cranckshaft, placed on a measuring table and seen by 6 virtual cameras with artificial illumination. Fig. 26 depicts the 3-D crankshaft reconstruction plotted as a point cloud (left) and selected points of the reference CAD model (right). The reconstructed crankshaft is compared with a ground truth

Table 1
Calibration of intrinsic parameters.

Camera	Focal length(mm)
1	34.729
2	37.439
3	34.921
4	33.314
5	34.217
6	35.380

Table 2
Calibration of extrinsic parameters.

Index(i)	$t_i^\intercal = [t_x \quad t_y \quad t_z]$ (mm)	$R_i^\intercal = [\theta_x \quad \theta_y \quad \theta_z]$ (deg)
1	[302.984 −0.859 0.000]	[0.000 0.000 0.000]
2	[302.995 0.800 0.000]	[0.000 0.000 0.000]
3	[0.022 −12252.321 10287.235]	[−82.984 0.000 177.016]
4	[302.954 0.272 0.000]	[0.000 0.000 0.000]
5	[303.003 0.000 0.000]	[0.000 0.000 0.000]



(a) Image of camera 1. (b) Image of camera 2.

Fig. 25. Image matching of two crankshaft images.

Table 3
Keypoint correspondences in the stereo subsystems.

Input	Keypoint matches
Cameras 1 and 2	320
Cameras 2 and 3	357
Cameras 4 and 5	335
Cameras 5 and 6	307

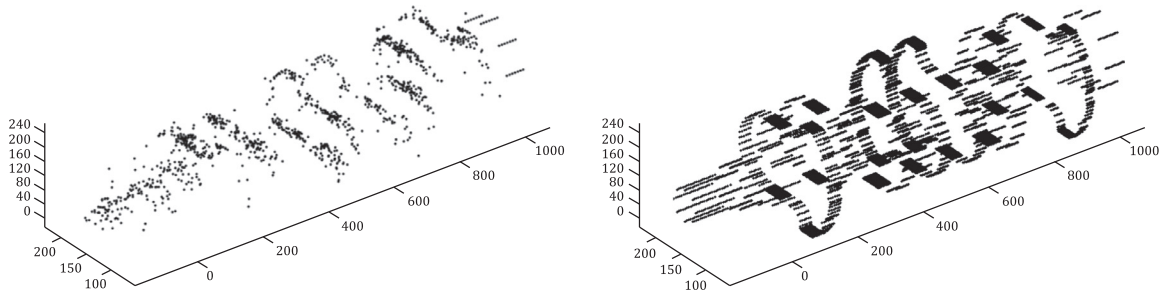
CAD cloud of 11,130 points. The reconstruction error is calculated as a measure of the nearest distances between the reconstructed and CAD points. The result yields a Gaussian approximation of the reconstruction error distribution with $\mu = 1.594$ mm and $\sigma = 2.703$ mm. Mean error and standard deviation are the controlling parameters to verify the quality of the reconstructed crankshaft solid.

6. Application 3: Wheel alignment measuring system

The following design is a system to measure the alignment angles of automotive wheels. The system is based on the study of Kurka and Mingoto [30] and patent [27]. The specific goal is to estimate the *camber* and *toe* alignment angles of the front wheels of a car, depicted in Fig. 27. The third alignment angle, *caster*, can be measured indirectly [28] by turning the wheels of the vehicle in a process called *caster sweep* [29]. The acquisition devices are eight cameras arranged in four stereo subsystems that capture simultaneous wheel images. System calibration is based on an adaptation of the algorithm introduced in Section 3.6. System tests are performed on a virtual set of automotive wheels.

6.1. System setup and calibration

Fig. 28 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, high enough to have the center of the wheels and the optical axes of the cameras at the same height. Cameras with 1280×720 pixel resolution are separately oriented towards the



(a) 3-D reconstructed crankshaft. (b) CAD crankshaft cloud of 11,130 points.

Fig. 26. The 3-D crankshaft reconstruction.

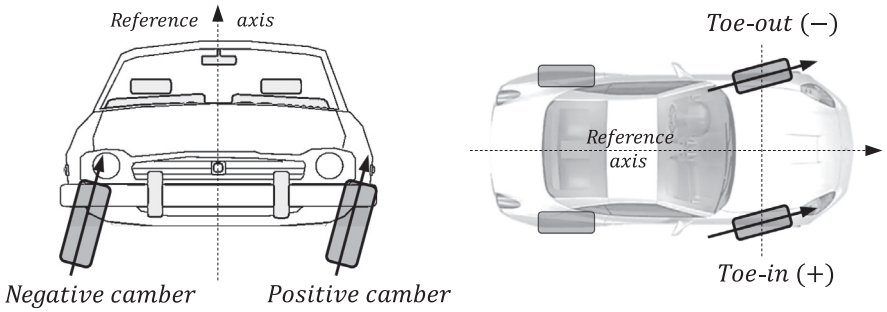


Fig. 27. The camber and toe wheel alignment angles.

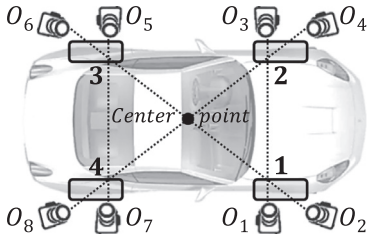


Fig. 28. The wheel alignment measuring system.

wheels to operate in *precision* and *reference* dispositions. Precision cameras 1, 3, 5 and 7 are perpendicular to the wheels to capture the largest possible images of the rims, as per recommendation of a previous sensitivity analysis of the system [30]. Reference cameras 2, 4, 6 and 8 are oriented towards the geometric center of the system to capture images of the rims as well as to facilitate subsequent calibration procedures.

System calibration is divided in local and global modified versions of the algorithm introduced in Section 3.6. The calibration patterns are hollow cubes of known dimensions with spherical vertices of different colors. This modification simplifies the calculus of the positions of the calibration points [20]. The calibration points are the image plane projections of the centers of the eight spherical cube vertices. The algorithm to identify such calibration points is described in [31].

Local calibration identifies the intrinsic and extrinsic parameters of the stereo subsystems. Four stereo calibrations are performed using a small calibration calliper, as shown in Fig. 29a. The local extrinsic parameters of the stereo subsystems are calculated with respect to corresponding reference cameras using Eqs. (56) and (57).

Global calibration process identifies the positions of the stereo subsystems with respect to the world reference frame. The origin of the world coordinates, denoted by O_W, is arbitrarily located at the projection center of camera 2. A bigger calibration caliper is positioned at the center point of the measuring system, as shown in Fig. 29b. Rotations and translations are calculated with respect to corresponding reference cameras. The global extrinsic parameters of the stereo subsystems are calculated with respect to the world origin using Eqs. (56) and (57).

6.2. Image matching

Fig. 27a shows wheel images of cameras 7 and 8 and the identified rim contour ellipses (in dotted lines). The rim contour points in the left image are associated to epipolar lines in the right image due to the epipolar constraint, as described in Section 3.7. An epipolar line intersects the rim ellipse at two points, as shown in Fig. 30b. Because the cameras are located at the same height, the pair of corresponding ellipse points will be the one of closest distance in the y-axis of image coordinates. The image matching process is repeated for all the ellipse points in the left image to determine their corresponding rim contour points in the right image, linked by a straight line as depicted in Fig. 30c. Due to the resolution of the image or other optical uncertainties, some epipolar lines of the upper and lower sections of the rim may intersect the ellipse at more than two points or may not intersect it at all. Consequently, the ellipse points in the upper and lower sections of the rim are neglected by considering the discriminant threshold of 10% of the rim diameter, which results in a smaller set of corresponding contour points, as shown in Fig. 30d.

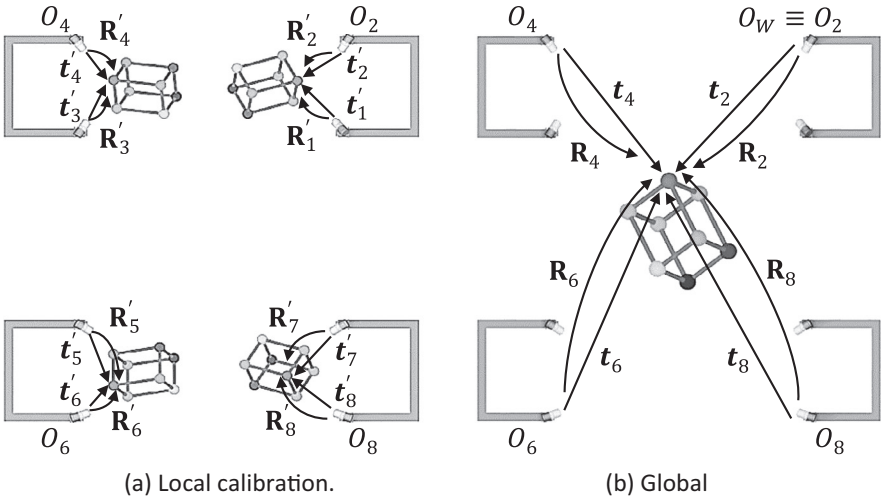


Fig. 29. The system calibration processes.

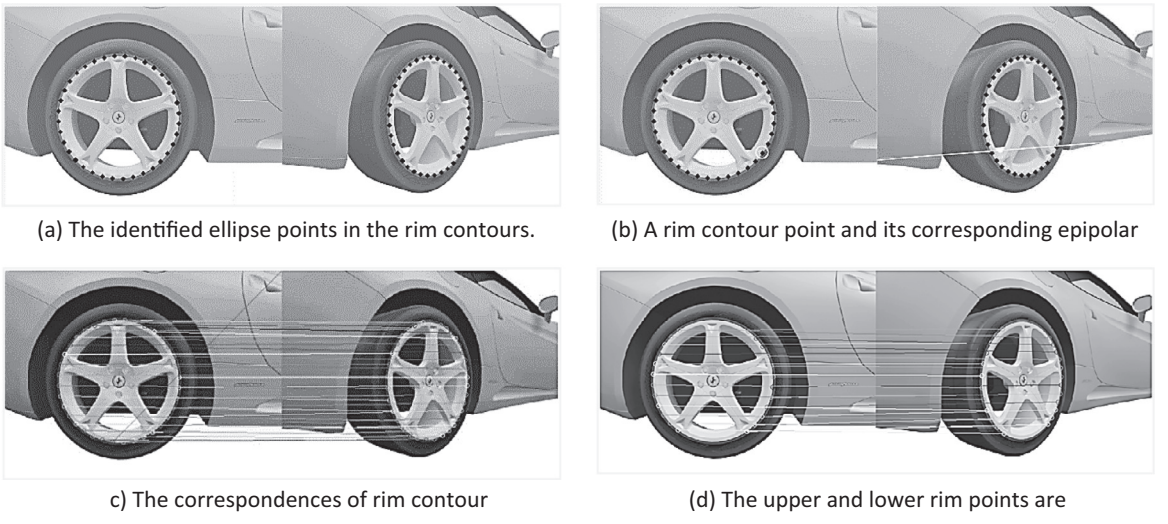


Fig. 30. The epipolar lines of a wheel contour and the matching process.

6.3. Wheel rim edge reconstruction

Fig. 31a illustrates the result of the 3-D wheel rim edge reconstruction. The 3-D wheel rim edge is reconstructed by triangulation of the corresponding rim contour points using Eq. (61). In order to be able to measure the wheel alignment angles, an auxiliary reference frame called the *wheel axis system* is firstly defined. The wheel axis system is the orthogonal 3-D basis resulting from the Principal Component Analysis (PCA) of the 3-D wheel rim edge points, as illustrated in Fig. 31b. Vectors u and v are located in the wheel rim edge plane and 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.

6.4. Camber and toe angle estimation

The camber and toe alignment angles of the front wheels are measured in the *car reference frame*, which is a special reference frame calculated as follows. A vector $\boldsymbol{x}'$ is defined in the direction of the line that connects the centers of wheels 3 and 4. A second vector $\boldsymbol{y}'$ is defined in the direction of the line that connects the centers of wheels 3 and 2. A third vector $\boldsymbol{z}'$ is calculated by the cross product of $\boldsymbol{x}'$ and $\boldsymbol{y}'$. The triad $\boldsymbol{x}'$, $\boldsymbol{y}'$ and $\boldsymbol{z}'$ is normalized to yield the basis of unit vectors $\hat{x}$, $\hat{y}$ and $\hat{z}$, denoted the car reference frame.

Fig. 32a 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.

Fig. 32b 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 plane $\widehat{x}\widehat{z}$. The camber angle of wheel 1, denoted by γ_1, is the angle from $\boldsymbol{w}_{1xz}$ to the $\hat{x}$-axis and is calculated as

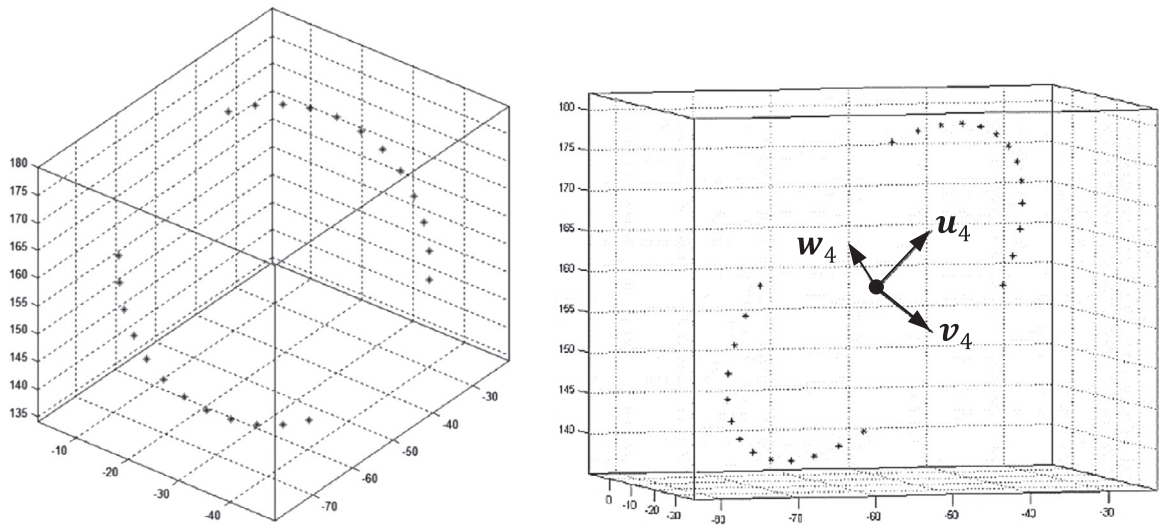
$$\gamma_1 = \arccos\left(\frac{\boldsymbol{w}_{1xz}\cdot\boldsymbol{w}_4}{\|\boldsymbol{w}_{1xz}\|\|\boldsymbol{w}_4\|}\right). \tag{75}$$

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

$$\gamma_2 = \arccos\left(\frac{\boldsymbol{w}_{2xz}\cdot\boldsymbol{w}_3}{\|\boldsymbol{w}_{2xz}\|\|\boldsymbol{w}_3\|}\right) \tag{76}$$

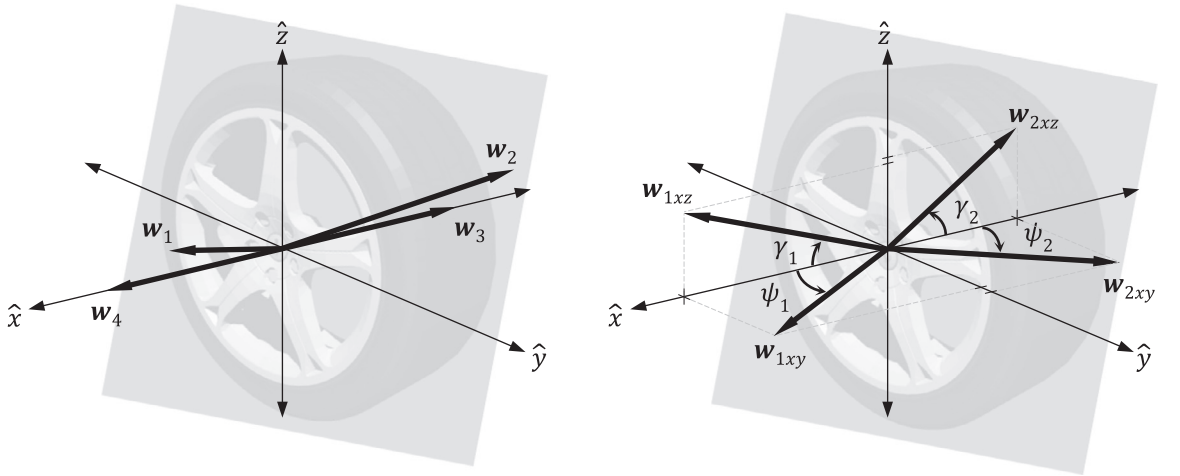
Vector $\boldsymbol{w}_{1xy}$ is the projection of $\boldsymbol{w}_1$ onto the $\widehat{x}\widehat{y}$ plane. The toe angle of the wheel 1, denoted ψ_1, is the angle from $\boldsymbol{w}_{1xy}$ to the $\hat{x}$-axis and is calculated as

$$\psi_1 = \arccos\left(\frac{\boldsymbol{w}_{1xy}\cdot\boldsymbol{w}_4}{\|\boldsymbol{w}_{1xy}\|\|\boldsymbol{w}_4\|}\right). \tag{77}$$



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

Fig. 31. 3-D reconstruction of wheel 4 and its associated PCA directions.



(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.

Fig. 32. The car reference frame and wheel alignment angles.

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

$$\psi_2 = \arccos\left(\frac{\boldsymbol{w}_{2xy}\cdot\boldsymbol{w}_3}{\|\boldsymbol{w}_{2xy}\|\|\boldsymbol{w}_3\|}\right) \tag{78}$$

6.5. Simulation test results

The system is simulated in the 3-D graphics freeware Blender, described in Section 5.4. For this simulation, the 3-D model of a complete vehicle is used, as well as all virtual cameras and illumination needed to emulate the environmental conditions of the measurement test. Table 4 shows the dimensions of the calibration patterns used in local and global calibration processes.

Table 4
Dimensions of the calibration boxes.

Dimensions(*cm*)	Local calibration pattern	Global calibration pattern
Width	17	65
Height	17	75
Depth	27	85
Ball radius	2.5	7.5
Side radius	1.25	2.5

Table 5
Ground truth and measuring systemmeasurements.

Ground truth (*deg*)				Measuring system (*deg*)			
Wheel 1		Wheel 2		Wheel 1		Wheel 2	
Camber	Toe	Camber	Toe	Camber	Toe	Camber	Toe
0	0	0	0	−1.07	0.86	0.12	−0.31
0	20	0	0	−0.74	19.21	0.12	−0.31
0	−20	0	0	−1.42	−18.34	0.12	−0.31
20	0	0	0	18.52	0.45	0.12	−0.31
−20	0	0	0	−20.81	1.03	0.12	−0.31
0	0	0	20	−1.07	0.86	−0.18	19.04
0	0	0	−20	−1.07	0.86	0.39	−20.01
0	0	20	0	−1.07	0.87	19.78	−0.42
0	0	−20	0	−1.08	0.85	−19.64	−0.62
0	0	20	−20	−1.07	0.87	20.89	−20.09
Average difference				**1.11**	**0.95**	**0.35**	**0.44**
Maximum difference				**1.48**	**1.66**	**0.89**	**0.96**

Table 5 summarizes the results of the measurements obtained with the measuring system against the ground truth wheel angles. A total of ten misaligned camber and toe angles were simulated. The first five correspond to wheel 1 and next five to wheel 2.

The results demonstrate congruent angular measurements with 1.11 degrees of maximum average difference. The maximum angular difference of all the measurements was 1.66 degrees.

7. Application 4: Stereo visual odometry for mobile robotics

Visual odometry is a technique to estimate the pose of a vehicle using imaging sensors. It is widely used in mobile robotics to complement wheel encoders odometry estimations in autonomous navigation. The following application example is inspired in a previous work [32], which introduces a stereo visual odometry algorithm to reconstruct the path of a robot moving in an unstructured environment. Robot motion is reconstructed using a stereo camera system and a feature-based image matching approach. Visual odometry is calculated by integrating the estimations of small successive displacement movements of the robot.

7.1. System setup and calibration

The system setup consists of two cameras of 720×480 pixel resolution and fixed baseline assembled in a mobile robotic platform. The parameters of the stereo system are the focal lengths f_l and f_r, the camera's relative rotation matrix $\mathbf{R}$ and translation vector $\boldsymbol{t}$, as shown in Fig. 16. Calibration is performed using the algorithm introduced in Section 3.6 to determine the intrinsic and extrinsic parameters of the cameras. Local robot motion will be estimated in the *robot reference frame*, which is a coordinate system located at the projection center of the left camera. The world reference frame is arbitrarily located at the initial position of the robot.

7.2. Image matching

The image matching algorithm is based on a practical implementation of the SIFT algorithm described in Section 3.8 [26]. Four stereo images are captured at consecutive robot steps $i-1$ and i, where $i \in \mathbb{N}$, and are analyzed as shown in the sequence illustrated in Fig. 33. The image planes π_l and π_r refer to the left and right cameras, respectively. Numbered arrows in the figure represent the sequenced operation of finding correspondence keypoints. 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 will be the input data for visual odometry estimation.

7.3. Stereo visual odometry algorithm

Fig. 34 illustrates the correspondences of keypoints at two successive robot steps. Local robot motion is represented by the rotation matrix $\mathbf{R}_i$ and the translation vector $\boldsymbol{t}_i$.

Local robot motion is retrieved from the 3-D keypoint reprojections $\boldsymbol{P}_k$ and $\boldsymbol{Q}_k$ of the left camera at corresponding robot steps $i-1$ and i, respectively, where k is the index of a detected keypoint in the scene. The 3-D keypoint reprojections are calculated by triangulation, as in Eq. (61). The reprojections are arranged in the matrices of homogeneous coordinates $\hat{\boldsymbol{P}}$ and $\hat{\boldsymbol{Q}}$ of size $4 \times k$, according to

$$\hat{\boldsymbol{P}} = \begin{bmatrix} \boldsymbol{P}_1 & \boldsymbol{P}_2 & \dots & \boldsymbol{P}_k \\ 1 & 1 & \dots & 1 \end{bmatrix}, \tag{79}$$

$$\hat{\boldsymbol{Q}} = \begin{bmatrix} \boldsymbol{Q}_1 & \boldsymbol{Q}_2 & \dots & \boldsymbol{Q}_k \\ 1 & 1 & \dots & 1 \end{bmatrix}. \tag{80}$$

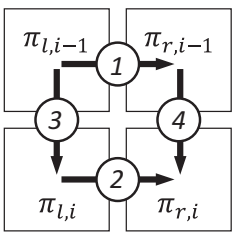


Fig. 33. Stereo matching sequence.

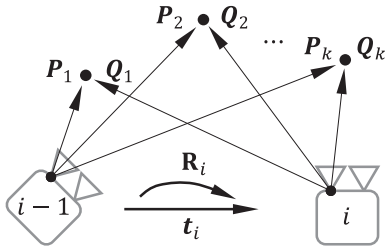


Fig. 34. Keypoint correspondences and the robot motion model.

Local robot motion is estimated by least-squares solution of the transformation matrix $\mathbf{M}$ that relates both matrices $\widehat{\mathbf{P}}$ and $\widehat{\mathbf{Q}}$, such as

$$\mathbf{M} = \widehat{\boldsymbol{P}}\widehat{\boldsymbol{Q}}^{\dagger}, \tag{81}$$

$$\mathbf{M} = \begin{bmatrix} \overline{\mathbf{R}}^{\mathsf{T}} & \bar{\mathbf{t}}_i \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}, \tag{82}$$

where the estimated 3×3 robot's rotation matrix $\overline{\boldsymbol{R}}_i$ and the 3×1 translation vector $\bar{\boldsymbol{t}}_i$ are submatrices of the block matrix $\mathbf{M}$, and $\boldsymbol{0}$ is a vector of null elements and size 3×1.

In practical scenarios, robot motion may be estimated more accurately through Principal Component Analysis (PCA) of the keypoint clouds $\boldsymbol{P}_k$ and $\boldsymbol{Q}_k$. The PCA representation increases robustness to the estimates of 3-D keypoint positions, provides invariance to scale of the 3-D clouds, and reduces the dimension of the original problem. The robot motion parameters are estimated using PCA, such as

$$\bar{\boldsymbol{R}}_{\boldsymbol{i}} = \boldsymbol{U}_Q \boldsymbol{U}_P^{-1}, \tag{83}$$

$$\bar{\boldsymbol{t}}_{\boldsymbol{i}} = \bar{\boldsymbol{P}} - \bar{\boldsymbol{R}}_i^{\mathsf{T}} \bar{\mathbf{Q}}, \tag{84}$$

where $\mathbf{U}_P$ and $\mathbf{U}_Q$ are 3×3 eigenvector matrices with the principal directions of the keypoint clouds $\boldsymbol{P}_k$ and $\boldsymbol{Q}_k$, respectively, and $\overline{\mathbf{P}}$ and $\overline{\mathbf{Q}}$ are the 3×1 vectors of corresponding cloud centroids.

Fig. 35 illustrates the flow of the stereo visual odometry algorithm. Visual odometry is calculated by the composition of successive rotations and translations. The robot pose, denoted $\boldsymbol{B}_i$, is defined by the recursive formula

$$\boldsymbol{B}_i = \boldsymbol{B}_{i-1} + \left(\prod_{n=0}^{i-1} \overline{\mathbf{R}}_n^{\mathsf{T}} \right) \bar{\mathbf{t}}_i, i > 0, \tag{85}$$

or in the explicit version of the formula as

$$\boldsymbol{B}_i = \boldsymbol{B}_0 + \sum_{m=1}^{i} \left(\prod_{n=0}^{m-1} \overline{\mathbf{R}}_n^{\mathsf{T}} \right) \bar{\mathbf{t}}_m, i > 0, \tag{86}$$

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

7.4. Simulation test results

The example is based on 3-D simulations using the freeware platform Blender. In this application the 3-D (virtual) model of the inside environment of a commercial office is used. The model is comprised of texturized furniture and equipment, floor, ceiling, windows, natural and artificial illumination. A 3-D model of a small differential wheeled robot with a pair

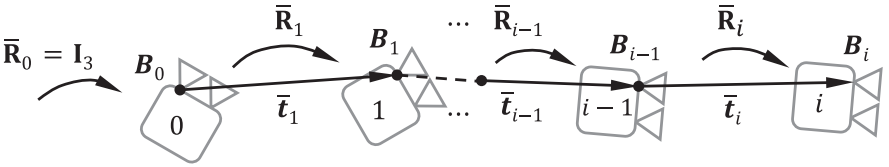


Fig. 35. Stereo visual odometry in a mobile robot.

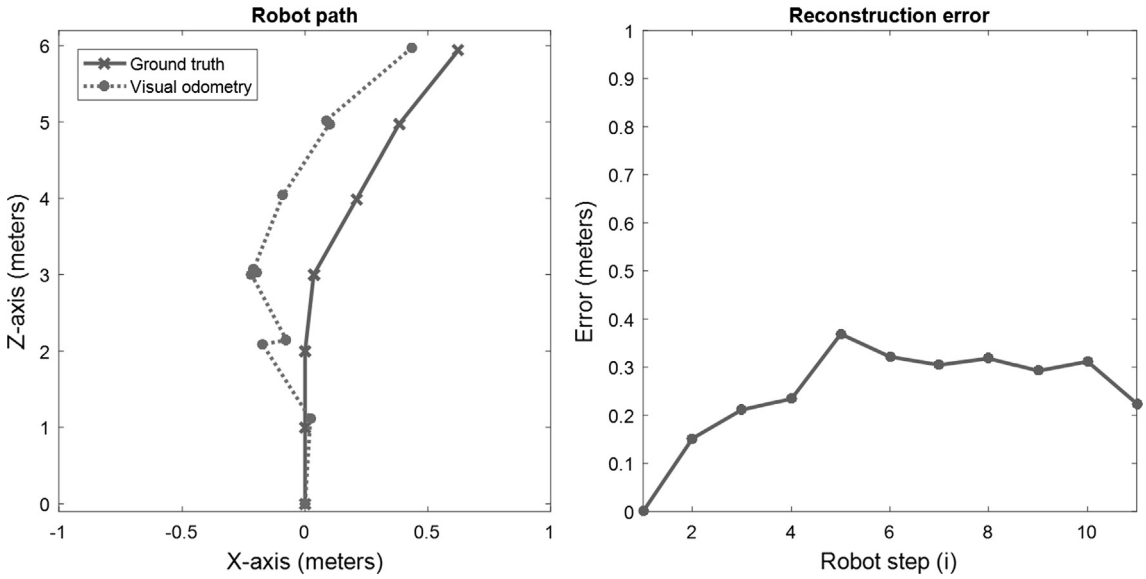


Fig. 36. The original robot path and the stereo visual odometry reconstruction.

of stereoscopic cameras is also introduced in the indoor environment. The simulated robot moves in a predefined trajectory in the virtual environment, capturing images from both cameras as it travels small increments of distance along the path. Fig. 36 depicts the reconstruction of a simulated robot trajectory with $i = 11$ steps. The ground truth robot path contains straight and diagonal lines and rotations at different angles. The deviation error of the reconstructed path is measured in units of length as

$$e_i = ||\boldsymbol{B}_i - \widetilde{\boldsymbol{B}}_i||, \tag{87}$$

where $\boldsymbol{B}_i$ and $\widetilde{\boldsymbol{B}}_i$ are the actual and estimated robot poses at step i, respectively. Visual odometry estimation yields an error of mean, $\mu = 0.2488m$ and standard deviation, $\sigma = 0.1035m$. Such a visual odometry algorithm is plausible of being implemented in real-time scenarios [33] and can be further improved in combination with probabilistic sensor fusion algorithms and additional perception devices as in current state-of-the-art in visual navigation [34–36,38,39].

8. Conclusion

This paper reviews the fundamentals of image acquisition and processing within the scope of industrial automation, robotics and instrumentation applications. The text presents the relevance and rationale behind the most popular and useful techniques of digital image processing in a comprehensive manner. The technical details of photogrammetry discussed here, such as the geometry of perspective projection, camera calibration, epipolar geometry and stereo correspondence form a collection of concepts and techniques that can be readily applied by scientists and engineers. Practical applications of the techniques described in this work are shown in the form of four different examples of scientific and industrial relevance.

Acknowledgment

This work was supported by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil, grant 88881.187827.

References

[1] D.A. Forsyth, J. Ponce, Computer Vision: A Modern Approach, 2nd ed., Prentice Hall, New Jersey, 2012.
[2] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, New York NY, 2011, doi:10.1007/978-1-84882-935-0.
[3] Y. Ma, S. Soatto, J. Košecká, S.S. Sastry, An Invitation to 3-D Vision, Springer, New York, NY, 2004, doi: 10.1007/978-0-387-21779-6.
[4] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, Cambridge, 2004, doi:10.1017/CBO9780511811685.
[5] E. Trucco, A. Verri, Introductory Techniques for 3-D Computer Vision, Prentice Hall, Upper Saddle River, NJ, 1998.
[6] R.C. Gonzalez, R.E. Woods, Digital Image Processing, 3rd ed., Pearson, Upper Saddle River, NJ, 2008.
[7] P. Corke, Robotics, Vision and Control,, Springer, Berlin, Heidelberg, 2011, doi: 10.1007/978-3-642-20144-8.
[8] R. Jain, R. Kasturi, B.G. Schunck, Machine Vision, Mcgraw-Hill, New York, NY, 1995.
[9] R. Hirsch, Exploring Colour Photography: A Complete Guide, 1st ed., Laurence King Publishing, London, 2004.

[10] D.L. Morton, A History of Electronic Entertainment, IEEE Press, New York, 1999.
[11] K. Jack, Video Demystified: A Handbook for the Digital Engineer, 5th ed., Elsevier, Burlington, MA, 2007, doi:10.1016/B978-0-7506-8395-1.X5000-7.
[12] J.G. Fryer, D.C. Brown, Lens distortion for close-range photogrammetry, Photogramm. Eng. Remote Sensing. 52 (1986) 51–58.
[13] D.C. Brown, Decentering distortion of lenses, Photogramm. Eng. 32 (1966) 444–462.
[14] J.-H. Han, S. Yang, B.-U. Lee, A novel 3-D color histogram equalization method with uniform 1-D gray scale histogram, IEEE Trans. Image Process. 20 (2011) 506–512, https://doi.org/10.1109/TIP.2010.2068555.
[15] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8 (1986) 679–698, https://doi.org/10.1109/TPAMI.1986.4767851.
[16] R.O. Duda, P.E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Commun. ACM. 15 (1972) 11–15, https://doi.org/10.1145/361237.361242.
[17] D.H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, Pattern Recognit. 13 (1981) 111–122, https://doi.org/10.1016/0031-3203(81)90009-1.
[18] J. Salvi, X. Armangué, J. Batlle, A comparative review of camera calibrating methods with accuracy evaluation, Pattern Recognit. 35 (2002) 1617–1635, https://doi.org/10.1016/S0031-3203(01)00126-1.
[19] J.-Y. Bouguet, Camera calibration toolbox for Matlab, Camera Calibration Toolbox for Matlab. (2015). http://www.vision.caltech.edu/bouguetj/calib_doc/. (accessed 1 January 2016).
[20] P.R.G. Kurka, J.V. Delgado, C.R. Mingoto, O.E.R. Rojas, Automatic estimation of camera parameters from a solid calibration box, J. Brazilian Soc. Mech. Sci. Eng. 35 (2013) 93–101, https://doi.org/10.1007/s40430-013-0013-2.
[21] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2004) 91–110, https://doi.org/10.1023/B:VISI.0000029664.99615.94.
[22] T. Lindeberg, Scale-space theory: a basic tool for analyzing structures at different scales, J. Appl. Stat. 21 (1994) 225–270, https://doi.org/10.1080/757582976.
[23] Yonghong Xie, Qiang Ji, A new efficient ellipse detection method, in: Object Recognit. Support. by User Interact. Serv. Robot., IEEE Comput. Soc, n.d.: pp. 957–960. doi:10.1109/ICPR.2002.1048464.
[24] C.A. Basca, M. Talos, R. Brad, Randomized Hough Transform for Ellipse Detection with Result Clustering, in: EUROCON 2005 - Int. Conf. 'Computer as a Tool', IEEE, 2005: pp. 1397–1400. doi:10.1109/EURCON.2005.1630222.
[25] R.G. Lins, P.R.G. Kurka, Architecture for multi-camera vision system for automated measurement of automotive components, in: 2013 IEEE Int. Syst. Conf., IEEE, 2013: pp. 520–527. doi: 10.1109/SysCon.2013.6549931.
[26] A. Vedaldi, B. Fulkerson, Vlfeat: An Open and Portable Library of Computer Vision Algorithms, in: Proc. 18th ACM Int. Conf. Multimed., ACM, New York, NY, USA, 2010: pp. 1469–1472. doi: 10.1145/1873951.1874249.
[27] P.R.G. Kurka, C.R. Mingoto, Method and device for verification of vehicular wheel suspension angles (translation from Portuguese). (2011), Instituto Nacional da Propriedade Industrial, patent deposit PI1105356 Brasil.
[28] D.B. January, Steering Geometry and Caster Measurement, in: SAE 1985 Trans., Society of Automotive Engineers, Warrendale, PA, 1985: pp. 176–185. doi: 10.4271/850219.
[29] J.D. Halderman, Alignment Diagnosis and Service, in: Automot. Steering, Suspens. Alignment, 7th ed., Pearson Education, New Jersey, 2017: p. 379. http://jameshalderman.com/books_a4.html.
[30] C.R. Mingoto, P.R.G. Kurka, Guidelines to Specify a Camera for Computer Vision Measurement of Vehicle Suspension Alignment Angles Based on the Desired Sensibility, in: 2012 Symp. Photonics Optoelectron., IEEE, 2012: pp. 1–3. doi:10.1109/SOPO.2012.6270430.
[31] O.E.R. Rojas, P.R.G. Kurka, Algoritmo de Procura de Círculos em uma imagem RGB, in: Proc. SPS 2012, Campinas, 2012: pp. 1–3. http://www.sps.fee.unicamp.br/sps2012/proceedings_sps2012/Oscar_RGB_SPS2012.pdf.
[32] J. V. Delgado, P.R.G. Kurka, E. Cardozo, Visual odometry in mobile robots, in: IX Lat. Am. Robot. Symp. IEEE Colomb. Conf. Autom. Control. 2011 IEEE, IEEE, 2011: pp. 1–4. doi:10.1109/LARC.2011.6086813.
[33] J. V. Delgado, P.R.G. Kurka, The Use of a Graphic Processing Unit (GPU) in a Real Time Visual Odometry Application, in: 2015 IEEE Int. Conf. Dependable Syst. Networks Work., IEEE, 2015: pp. 141–146. doi:10.1109/DSN-W.2015.32.
[34] M. Roth, G. Hendeby, F. Gustafsson, Nonlinear kalman filters explained: a tutorial on moment computations and sigma point methods, J. Adv. Inf. Fusion. 11 (2016) 47–70.
[35] S.Y. Chen, Kalman filter for robot vision: a survey, IEEE Trans. Ind. Electron. 59 (2012) 4409–4420, https://doi.org/10.1109/TIE.2011.2162714.
[36] F. Gustafsson, Particle filter theory and practice with positioning applications, IEEE Aerosp. Electron. Syst. Mag. 25 (2010) 53–82, https://doi.org/10.1109/MAES.2010.5546308.
[37] Paulo R.G. Kurka, R.G. Lins, Dimensional measuring device for automated analysis and inspection (translation from portuguese). (2014), Instituto Nacional da Propriedade Industrial, patent deposit BR10201400946, Brasil.
[38] D. Scaramuzza, F. Fraundorfer, Visual Odometry [Tutorial], IEEE Robot. Autom. Mag. 18 (2011) 80–92, https://doi.org/10.1109/MRA.2011.943233.
[39] F. Fraundorfer, D. Scaramuzza, Visual Odometry: Part II: Matching, Robustness, Optimization, and Applications, IEEE Robot. Autom. Mag. 19 (2012) 78–90, https://doi.org/10.1109/MRA.2012.2182810.