Accurate Camera Calibration Algorithm Using a Robust Estimation of the Perspective Projection Matrix

R. Guerchouche\textsuperscript{a,b}, F. Coldefy\textsuperscript{a}, and T. Zaharia\textsuperscript{b}

\textsuperscript{a} FranceTelecom R&D, Technopole Anticipa 2, Avenue Pierre Marzin, 22307 Lannion, France; \textsuperscript{b} Institut National des Télécarmunications, ARTEMIS department, 9, Rue Charles Fourier, 91011 Evry, France

ABSTRACT

After a comprehensive overview of camera calibration algorithms, this paper introduces a robust calibration method based on contour matching of a pattern object. In contrast with state of art methods, no manual and fastidious selection of particular pattern points is required. In order to evaluate the accuracy of the proposed approach, an objective comparison with three well known methods (Faugeras and Toscani, Tsai, and Zhang) is proposed and discussed in detail. Experiments show that the proposed robust approach outperforms existing techniques and yields accurate results even when calibration is performed from a single image.

Keywords: camera calibration, robust estimation, perspective projection matrix.

1. INTRODUCTION

Camera calibration is an essential stage for any computer vision application requiring to establish the geometrical relation between the acquired image and the observed 3D scene. Typical applications include stereo image rectification, 3D reconstruction, object tracking ... Camera calibration aims at solving two basic problems: (i) estimating the intrinsic parameters of the camera, \textit{i.e.} focal length, principal point, aspect ratio, skew, and lens distortion; (ii) estimating the extrinsic parameters, \textit{i.e.} the camera pose in the 3D space.

The objective of this paper is twofold. The first one is to review the state of the art calibration methods and analyse their advantages and limitations. The second is to propose an accurate and robust algorithm for camera calibration. In contrast with state of the art, calibration methods, which need a tedious and error-prone manual selection of feature points, our technique requires only a minimal user interaction. A colored cube that is used as a pattern reference object. The user has only to select two faces of the cube. The calibration algorithm automatically computes the camera model by minimizing the distance between projected cube edges and image contours. The proposed approach is objectively compared with three well-known calibration methods (Faugeras-Toscani,\textsuperscript{1} Tsai,\textsuperscript{2,3} and Zhang\textsuperscript{4,5}), and validated within the framework of an application related to the estimation of a commercial camera baseline.

The key ingredients of our algorithm are the contour matching procedure involved and the robust estimation technique which makes it possible to minimize the influence of outliers. Let us note that robust estimation has been frequently used in many computer vision applications, including object tracking in image sequences. Bryant \textit{et al.}\textsuperscript{6} have developed a robust calibration method for cameras mounted on an autonomous underwater vehicle. They use the Karambar's reference pattern\textsuperscript{7} providing 96 points which contribute to estimate the projection matrix by the Tsai calibration algorithm.\textsuperscript{3} Marchand\textsuperscript{8} proposes an algorithm for tracking complex objects. The estimation of the 3D object pose and of the 2D affine transformation between two successive frames of the video sequence is formulated as an energy minimization problem. Armstrongand \textit{et al.}\textsuperscript{9} present a tracking algorithm robust to various ambient lightning conditions based on geometric primitives of the known object.

Further author information (Send correspondence to R. Guerchouche):
E-mail: rachid.guerchouche@orange-ft.com, Telephone: +33 (0)2 96 05 99 34, Fax: +33 (0)2 96 05 11 29
E-mail: francois.coldefy@orange-ft.com, Telephone: +33 (0)2 96 05 12 41, Fax: +33 (0)2 96 05 11 29
E-mail: titus.zaharia@int-evry.fr, Telephone: +33 (0)1 60 76 46 74, Fax: +33 (0)1 60 76 43 81
The paper is organized as follows. In the next section, we recall the mathematical formulation of a pinhole camera model with lens distortion. Section 3 provides a critical overview of camera calibration methods for monocular, multi-view, and self-acquisition systems. Section 4 introduces the proposed camera calibration method, and describes in detail extrinsic and intrinsic parameter estimation. Experimental results are presented and discussed in Section 5. Finally, Section 6 concludes the paper and opens perspectives of future work.

2. CAMERA MODEL

The projection model considered in this paper corresponds to a pinhole camera model, recalled here-below.

2.1. Pinhole camera model

Most of the calibration methods consider a pinhole camera model,\(^{10}\) which assumes that the camera performs a perfect perspective transformation \(M\) from 3D scene coordinates \([X\ Y\ Z]\) to image plane coordinates \((u, v)\):

\[
\begin{bmatrix}
\eta u \\
\eta v \\
\eta w
\end{bmatrix}^t = M \begin{bmatrix}
X \\
Y \\
Z \\
1
\end{bmatrix}^t,
\]

where \(\eta\) is an homogeneous factor and \([\cdot]^t\) is the transpose operator. The Perspective Projection Matrix (PPM) \(M = (m_{ij})_{i,j=1...4}\) is defined as the product of intrinsic and extrinsic parameter matrices, respectively denoted as \(I_c\) and \([R\ |\ t]\):

\[
M = I_c [R\ |\ t].
\]

Here, \(R = (r_{ij})_{i,j=1...3}\) is an orthogonal matrix representing the camera orientation and the vector \(t = [t_x, t_y, t_z]^t\) the position of the camera in the 3D space:

\[
[R\ |\ t] = \begin{pmatrix}
r_{11} & r_{12} & r_{13} & t_x \\
r_{21} & r_{22} & r_{23} & t_y \\
r_{31} & r_{32} & r_{33} & t_z
\end{pmatrix}.
\]

\(I_c\) is a 3x3 matrix defined as:

\[
I_c = \begin{pmatrix}
\alpha_u & 0 & u_0 \\
0 & \alpha_v & v_0 \\
0 & 0 & 1
\end{pmatrix},
\]

where \(\alpha_u = f k_u\) and \(\alpha_v = f k_v\), \(f\) is the focal length in millimeter, \(k_u\) and \(k_v\) are the effective number of pixels per millimeter along the \(u\) and \(v\) axes. \((u_0, v_0)\) are the coordinates of the principal point, given by the intersection of the optical axis with the image plane, and \(\gamma\) is the skew factor that models non orthogonal \(u - v\) axes. The ratio \(s = \alpha_u/\alpha_v = k_u/k_v\) is called the aspect ratio. The image coordinates \((u, v)\) of a point \([X\ Y\ Z]\) in the world reference, with \(\eta = m_{31}X + m_{32}Y + m_{33}Z + m_{34}\), are given by:

\[
(u, v) = \left(\frac{m_{11}X + m_{12}Y + m_{13}Z + m_{14}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}}, \frac{m_{21}X + m_{22}Y + m_{23}Z + m_{24}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}}\right).
\]

Let us now consider a set of \(n\) 3D points \((X_k, Y_k, Z_k)\)\(^{n}\) and their corresponding projections onto the image plane \((u_k, v_k)\)\(^{n}\). For each point, the relation between world and image coordinates (equation (5)) can be re-written as a linear combination of coefficients \(m_{ij}\):

\[
\begin{align}
X_k m_{11} + Y_k m_{12} + Z_k m_{13} + m_{14} & = u_k X_k m_{31} + u_k Y_k m_{32} - v_k Z_k m_{33} = u_k m_{34}, \\
X_k m_{21} + Y_k m_{22} + Z_k m_{23} + m_{24} & = v_k X_k m_{31} - v_k Y_k m_{32} - v_k Z_k m_{33} = v_k m_{34}.
\end{align}
\]
By setting parameter $m_{34}$ to 1, the following linear system of equations is obtained:

$$
\mathbf{Km}_{ic[H|T]} = \mathbf{u},
$$

where $\mathbf{m}_{ic[H|T]}$ is the vector containing all $\mathbf{M}$ matrices’s elements except $m_{34}$.

### 2.2. Camera lens distortion

Cameras usually exhibit significant lens radial and tangential distortions.\(^{11}\) Radial distortion is point-symmetric at the optical centre of the lens and causes an inward or outward shift of image points from their initial perspective projection. About the optical center, radial distortion is generally modeled by a first and second order approximation, according to the following relation:

$$
(\tilde{u} - u, \tilde{v} - v) = (k_1 u(u^2 + v^2) + k_2 u(u^2 + v^2)^2, k_1 v(u^2 + v^2) + k_2 v(u^2 + v^2)^2),
$$

where $k_1$ and $k_2$ are the parameter model, $(\tilde{u}, \tilde{v})$ the observed image coordinates, $(u, v)$ the ideal (non observable distortion free) pixel image coordinates.

The tangential distortion is expressed as:

$$
(\tilde{u} - u, \tilde{v} - v) = (p_1 (3u^2 + v^2) + 2p_2 uv + s_1 (u^2 + v^2), p_2 (3u^2 + v^2) + 2p_1 uv + s_2 (u^2 + v^2)),
$$

where $p_1$ ($p_2$) denotes the first (second) order optical centre decentring coefficient, and $s_1$ and $s_2$ are coefficients describing the lens prism.

Camera calibration techniques aims at determining all the above presented parameters: PPM, intrinsic (including lens distortion) and extrinsic parameters. The calibration process requires a set of image acquisitions, knowledge on the observed scene (e.g. space coordinates of a set of points, shape of objects), and data extracted from the acquired images (e.g. coordinates of image points, point correspondences).

Let us now analyze how this issue is tackled by existing techniques of the literature.

### 3. CAMERA CALIBRATION: THE STATE OF THE ART

The issue of camera calibration has been extensively studied in the literature during the last few decades. The proposed solutions include single camera, multiple camera, as well as self (or auto) calibration systems. Let us analyze the most representative methods of each family of approaches.

#### 3.1. Single camera calibration

**Tsai’s method:**\(^2\) This technique makes it possible to determine the camera parameters including the first order radial lens distortion $k_1$. There are many simplifying assumptions limiting its relevance: the optical axis is supposed to be located at the image centre, and the skew is null. The calibration pattern may be but is not restricted to a plane. More than 200 measurement points on a single image are needed to achieve an accurate calibration. The camera has to be positioned so that its optical axis makes an angle of 30° with the normal to the calibration pattern plane.
The calibration is achieved in two steps. First, the distortion is ignored, \( t_z \) is set to 0 in the case of a plane pattern, and a subset of parameters (the orientation, the translations and the aspect ratio) are estimated by a linear minimization algorithm. In a second stage, the focal length \( f \) and the pattern distance \( t_z \) are coarsely computed by a linear technique. Then, the focal length, the pattern distance and the distortion coefficient \( k_1 \) are refined using a non-linear optimization algorithm (e.g. Levenberg-Marquardt, Newton-Raphson).

**Batista’s Method:**\(^{12}\) In comparison with the Tsai’s method, this approach offers the advantages that (i) it can be used with monoplane calibration data with no restriction for the pose geometry of the camera; (ii) no restrictive hypotheses relative to the intrinsic parameters are imposed. Batista’s method is based on a single image of a known planar pattern. The algorithm requires a coarse initialization of the intrinsic parameters \( f, k_u \) and \( k_v \) according to the camera manufacturer documentation. In order to avoid the singularities due to the calibration with monoplane points, the authors propose a four alternate steps method. At each step, only some subsets of calibration parameters are updated, the other ones being fixed. More precisely, the method proceeds as follows.

- **Step 1** The translation vector \( t \) is initialized to the null vector, while the intrinsic parameters are obtained from the camera manufacturer documentation.
- **Step 2** Compute the values of \( t_z \) and \( t_y \) as well as the aspect ratio \( s \).
- **Step 3** Compute the focal length, the radial distortion \( k_1 \) and the \( t_z \) component of \( t \).
- **Step 4** Compute the image scale factors, the image centre coordinates and the skew angle.

The steps 2-4 are iterated until the square error between the projection of the 3D points and their observed position in the image stops decreasing.

**Vanishing Points Method:** This method was originally proposed by Caprile and Torre\(^{13}\) for calibrating a system of two (or more) cameras. The initial method was extended by Cipolla *et al.*\(^{14}\) which estimate the PPM for one camera. The technique requires the acquisition of at least two images of a known cube from two different angles. Each face of the cube contains a set of parallel lines, the orientation of the lines on each face is different, so that three distinct vanishing points can be obtained. The set of the vanishing points in the two images makes it possible to define an orthocentre. A first value of \( f \) is calculated using a linear approach by supposing that the optical axis is passing through the image centre. This initial value is then refined by a nonlinear method integrating the radial distortion. The skew is also estimated. The rotation matrix is estimated using point correspondences in the two images.

**Heikkilä Method:**\(^{15}\) The calibration is performed in four steps. First, a linear method (Direct Linear Transform algorithm) is used to estimate the PPM, which gives the intrinsic (the focal length, the principal point as well as the radial and tangential distortions) and extrinsic parameters using a QR decomposition of the PPM. In the second step, a nonlinear parameter estimation is performed, based on the Levenberg-Marquardt algorithm, in order to improve the accuracy of the estimation. A correction of the points projection is realized in the third step. Finally, an image correction is performed to estimate the radial and tangential distortions. This method requires at least one image of a non-coplanar target, and allows the use of coplanar targets assuming \( t_z = 0 \).

**Faugeras and Toscani Method:**\(^{1}\) In the original formulation, this technique estimates the PPM, while the lens distortions are not modeled. The method requires the availability of a non-planar calibration pattern. Two different approaches are considered. The first one is based on a linear least squares (LS) solution of the system of equations defined by (8b), under the assumption that in equations (6) and (7) \( m_{34} = t_z = 1 \). A minimum number of 6 points is required. The projections are manually selected or automatically detected. The second approach involves a nonlinear solution, based on the orthogonality constraints of the rotation matrix which does not undergo the influence of the intrinsic parameters. The normalization constraint of the vector \( m_3 = (m_{31}, m_{32}, m_{33}) \) representing the last row of the rotation matrix \( R \), leads to the Faugeras-Toscani constraint:\( \|m_3\|^2 = 1 \).

More recent works improve the original technique by introducing distortion estimation. In,\(^{16}\) Weng improves the model by including radial and tangential lens distortion. Salvi *et al.*\(^{17}\) introduce radial distortion to obtain more accuracy.

**Zhang’s Methods:**\(^{4,5}\) This method requires at least two different views of a planar pattern (an accurate estimation is obtained using a larger number of views, approximately 20). The displacements between the views
are unknown. The algorithm consists of the following four steps:

Step 1) For each image acquisition, the homography between the planar pattern and its projection onto the image plane is computed from the detected feature points.

Step 2) The intrinsic parameter matrix $I_c$ and the camera poses are computed from the estimated homographies taking advantages of the geometrical constraints imposed by the planar shape of the pattern.

Step 3) Using a least square minimization technique, an initial estimation of the radial distortion is iteratively calculated, the five other intrinsic parameters being fixed.

Step 4) All the parameters are refined using a non-linear optimization technique (Levenberg-Marquardt algorithm) that minimizes the average of the image disparity.

Another method based on the displacement of a one dimensional object was proposed by Zhang. A set of collinear points (three at least) is used as a target. The distances between the points is supposed to be known. A high number of images of the object at different positions, is required. This technique is appropriate for video sequences with at least several hundreds of frames. However, the camera lens distortions are not taken into account.

**Neural Networks Camera Calibration:** Ahmed et al. propose a method based on a feedforward Neural Network (NN), called Neurocalibration. The world point coordinates are the input and the image coordinates are the output of the NN. The algorithm computes the PPM using one image of a non-planar pattern. A high number of feature points should be used. The PPM is then decomposed into the 11 parameters (extrinsic and intrinsic). The distortions are not estimated. In, authors extend their method to multi-image camera calibration. Other works also propose NNs for camera calibration, but without specifying the calibration parameters of the camera model. Mendonça proposes a technique based on the back propagation perceptron NN.

**Cuboid Structures for Camera Calibration:** In Wilczkowiak et al., parallelepipeds and parallelograms structures are used for camera calibration and reconstruction from a few images. The technique takes advantage of some geometric constraints such as parallelism and orthogonality present in man-made environments to estimate the camera calibration parameters. Authors demonstrate that intrinsic parameters of the camera are uniquely determined by the image of a known parallelepiped. Their method can be used with one ore more images. Some prior knowledge related to the intrinsic parameters of the camera or to the parallelepiped pattern is requested. In the case of a single image, solely the intrinsic parameters can be determined, while in the case of multiple cameras, the pose parameters can be estimated as well.

The same principle has been reconsidered and used by Debevec et al. as well as Chen within the context of augmented reality under the name of PCS (parametrized cuboid structure). However, the main limitations relate to constraints such as: scale factor equal to 1, prior knowledge on the angles of the parallelepiped pattern.

All of the mentioned methods require a tedious manual interaction with error-prone selection of image points. In addition, strong constraints related to the image acquisition process and use of more or less sophisticate patterns are needed. In order to overcome such limitations, multiple camera calibration methods consider systems composed by two or more cameras.

### 3.2. Multiple camera calibration

In the case of multi-camera calibration, at least two cameras are supposed to be available. The objective is then to determine the intrinsic and extrinsic parameters of each camera.

**Vanishing Points Method:** Caprile and Torre proposed a calibration method that exploits the properties of the vanishing points. First, the intrinsic parameters (focal length and principal point) are recovered from a single image of the cube. Then, the extrinsic parameters of two cameras are estimated from an image stereo pair of a suitable planar pattern. The matching of corresponding vanishing points in the two images makes it possible to determine the rotation matrix. The translation vector is estimated with the help of a triangulation. The method does not estimate the distortions and supposes that the skew is null.

**Multi-Camera System Self Calibration:** Svoboda et al. propose an interesting method to calibrate virtual immersive environments or telepresence setups. The minimum number of cameras required is three. The calibration object considered is a freely bright spot waving through the working volume, which yields a set of virtual 3D points projected on each camera image plane. RANSAC analysis is used to ensure that the
projection of the 3D points is determined with sub-pixel precision. The method also takes into account the nonlinear distortion. The cameras do not have to see all points, but only some "reasonable" overlap between camera subgroups is necessary. Projective structures are computed via a rank-4 factorization and a so-called Euclidean stratification is performed by imposing geometric constraints. The method is fully automatic.

**Dynamic Silhouettes for Camera Calibration:** The authors propose a method for calibrating a network of synchronized cameras from silhouettes. This method is interesting in the case of shape-from-silhouette or visual-hull systems. The epipolar geometry is robustly computed from the dynamic silhouettes. The fundamental matrices of each pair of camera estimated with the RANSAC algorithm allows to determine the projective reconstruction of the complete camera configuration. The method does not require the acquisition of specific calibration data, and works in the absence of texture, which makes it well-suited for applications dealing with widely separated views such as surveillance camera networks.

All multiple-camera calibration methods cannot be used to calibrate only one camera. In addition, the estimated parameters depend strongly on the number of cameras.

### 3.3. Self or calibration methods

In the self-calibration (or auto-calibration) techniques, the world’s Euclidean structure (the calibration target) is unknown. This family of approaches is also called 0D calibration, because no calibration pattern is needed.

The concept of self calibration for perspective camera was introduced by Maybank and Faugeras. The calibration is performed by tracking a set of particular points in the image sequence as the camera moves. The camera motion is not necessarily known. First, the epipolar transformation between each pair of successive frames is determined by using either a generalization of the essential matrix or projective invariant. Then, the solution is found by solving a set of Kruppa equations, which requires a minimum of three images acquired at different camera positions. The same principle was also adopted by Quan for calibrating an affine camera from multiple views. Furthermore, Faugeras et al. reconsidered the above described principle to self calibrate a 2D projective camera using a 1D projective camera self-calibration.

Gurdjos and Strum propose a nonlinear approach, which is able to deal with possibly varying focal length. In, Lavest proposes an auto-calibration method using a virtual pattern. The method requires a preliminary calibration of a camera, then the calibration of a projector using this camera. Any other camera with a similar focal length can then be calibrated using the projector which generates virtual patterns.

Horaud et al. proposed a method for auto-calibration of a stereo rig using a planar object that performs arbitrary, unknown motion. The auto-calibration is achieved using correspondences between points. The method requires at least five images, and about ten images are needed to guarantee an accurate result.

Planar structures were also used by Noirfalise et al. to auto-calibrate cameras. Their approach is iterative, and each new image in the sequence allows the projective reconstruction of the observed plane. The reconstruction process Gradually updates the metric structure of the scene. As in the case of Horaud et al. the method requires at least five images, and approximatively ten images are needed to obtain an accurate result.

All the above-described autocalibration methods require that:

(i) the user specifies the correspondences between feature points in the image sequence, or
(ii) a set of point correspondences between points on calibration target and the camera image(s) must be found, usually a manual or manually guided process.

On the contrary, in the camera can be automatically calibrated by passing it in front of a panel of so-called self-identifying patterns. The method uses an array of fiducial markers (or tags) which can be detected with a high degree of confidence. Each detected marker provides one or four correspondence points. The detected points are finally used for calibration using the Tsai’s or Zhang’s methods.

In this section, we presented a review of the most known methods for camera calibration, which are classified onto three categories: single, multiple and self methods. Each category present limitations: high number of images, constraints on parameters depending on the number of camera, error-prone feature points selection and tedious user interaction. To deal with this limitations, the next section presents an accurate and algorithm to estimate camera parameters.
4. ROBUST ESTIMATION OF THE CAMERA INTRINSIC PARAMETERS

In this section, we propose an accurate algorithm to estimate the intrinsic camera parameters. Our goal is to design a method requiring a minimum amount of user interaction and a simple calibration pattern. The pattern object considered is a cube with six differently colored faces. The user has to acquire one or several images of the cube at different depths and positions in the camera field. The sole constraint is that two adjacent faces of the cube have to be visible in all the images. Theoretically, a single image is sufficient for performing the calibration. However, we expect that taking up to 5 or 6 images of the cube may improve the accuracy of the camera intrinsic parameters.

The user has to select two adjacent faces of the cube in one of the calibration images. The algorithm then automatically computes the intrinsic parameters of the camera. No precise and fastidious selection of corresponding calibration points is needed.

An image segmentation procedure is applied for determining the cube’s faces. For each image, we automatically detect the 6 vertices of the two selected adjacent faces of the cube. A coarse estimation of the Perspective Projection Matrix (PPM) is then computed by applying the Faugeras-Toscani method.\(^1\) This initial PPM estimation is then refined by minimizing the distance between the projection of the cube edges and the detected contours. This distance is computed as a Chamfer distance\(^40\) to the image contours. A robust minimization technique, based on an Iterative Reweighted Least Square\(^41\) is carried out in order to take into account eventual inaccuracies of the detected edges.

By minimizing the distances between projected edges and observed contours, the algorithm provides a more precise estimation than traditional methods that involve a selection of the pattern points. Indeed, a fine manual estimation or automatic detection of the position of such points in the image cannot guarantee error-free localization.

Once the PPM determined, the intrinsic and extrinsic camera parameters are directly computed. In the case of a single calibration image, we apply the procedure described in Horaud and Monga.\(^10\) Otherwise, we use a Levenberg-Marquardt algorithm to compute the intrinsic parameters.

Let us now detail each step of the proposed method.

4.1. Detection of cube vertices

In order to identify the cube’s faces, we have adopted a meanshift segmentation algorithm,\(^42\) which takes into account both color and spatial information. For the color representation, the Luv color space has been considered because of its well-known nice metric properties.

First, the mean color of each cube face is computed. This makes it possible to identify the two faces selected by the user in the set of all images, by determining the regions with most similar colors.

A Harris corner detector\(^43\) is then applied to the binary image of each face region. More precisely, for each pixel, we compute the Hessian determinant of a smoothed version of the binary image. A pixel \(p\) is labeled as a corner if the determinant value at \(p\) is a local minimum with absolute value lower than a predefined threshold. However, because of eventual segmentation errors, some false alarms may occur. In order to eliminate such errors, the morphological skeleton of the binary face region is also computed. As the projected face is a parallelogram, the morphological skeleton has an ‘X’ form, whose extremities correspond to its corners. All the pixels previously labeled as corner points located too far from the obtained skeleton are removed. For each image, we thus detect six corners corresponding to the vertices of the two adjacent faces.

Figure 1 shows two examples of corner detection corresponding to two acquisitions with different focal lengths. In the first example, the cube pattern is the unique object in the scene, while in the second one the cube has been placed in a more complex environment which includes other objects and textures.

Figure 1.a presents the original image (left) and the obtained segmentation (right). Figure 1.b presents the binary images of each cube face. Let us note that in the second case, the segmentation contains some erroneous that are removed by a morphological opening operator.\(^44\) Figure 1.c shows the skeletons. Finally, figure 1.d shows, with red crosses, the detected vertices on the original images. In both examples, corner detection performs quite well. In both examples, the results were obtained without any parameter tuning neither for the segmentation procedure nor for the corner detection algorithm.
4.2. Estimation of the PPM

From the six detected vertices, a coarse estimation of the PPM is computed with Faugeras-Toscani method. However, the corner detection does not provide accurate positions of the cube vertices projections on each image. In order to obtain a more precise PPM estimation, we exploit the image contour information. The principle consists of minimizing the distance between projected cube edges and detected image contours. The image contours are detected by using a Canny-Derriche operator. Starting from the image contours, a Chamfer distance map is computed.

The cube's 3D edges are first discretized by sampling them uniformly within a 3D points. Let $E_i^j$ be the $j^{th}$ 3D point of the $i^{th}$ discrete cube edge and $e_i^j$ its projection onto the image. We have $e_i^j = K_i^j \cdot m$ where $K_i^j$ is defined by equation (7) and $m = m_{t_e \cdot [R_i]}$. The objective is to determine $m$ minimizing the following functional:

$$U (m) = \sum_{(i,j) \in V} \rho (f_{i,j}) ,$$

where $V$ is the set of visible projected edge points, $f_{i,j} = I (e_i^j) = I (K_i^j \cdot m)$ is the the value of the Chamfer map $I$ at the point $e_i^j$ and $\rho$ is the Tukey function. The Tukey function makes it possible to decrease the influence of erroneous or undetected image contours.
The minimization of $m$ is performed by an applying the Iterative Reweighted Least Square Algorithm (IRLS) procedure. At step $l$, an increment $dm$ is determined by minimizing the following linearized functional:

$$U(m_l + dm) = \sum_{(i,j) \in V} \rho(f_{i,j} + <J_{i,j}, dm>),$$

where $m_l$ is the current value of $m$, $J_{i,j}$ is the Jacobian in $m$ of $f_{i,j}$ and $< . , >$ stands for the scalar product. The increment $dm$ is computed as the solution of the equation:

$$\hat{dm} = \left[ \sum_{(i,j) \in V} J_{i,j}^t J_{i,j} \right]^{-1} \sum_{(i,j) \in V} J_{i,j} f_{i,j} \omega_{i,j},$$

$$\omega_{i,j} = \rho''(f_{i,j} + <J_{i,j}, dm>) f_{i,j} + <J_{i,j}, dm>$$

where $\rho''(.)$ stands for the first derivative of $\rho(.)$.

The vector of PPM coefficients $m$ is updated following the rule: $m_{l+1} = m_l + \hat{dm}$.

The algorithm stops when the relative variation of energy $U(m)$ becomes less than a predefined threshold $\tau$ such that: $\frac{U(m_{l+1}) - U(m_l)}{U(m_l)} \leq \tau$. More details on IRLS can be found in Hong and in the Appendix.

Figure 2 shows initial and final projections of the cube edges onto the image contours. The initial detected corners do not match exactly the projected cube’s vertices. The proposed algorithm significantly improves the initial matching.

Let us denote by $U_i$ the initial value of the energy $U(m)$, corresponding to Faugeras-Toscani estimation, and by $U_f$ the finale value, after robust estimation. Both values are normalized to the number of edge pixels. The values obtained are $U_i = 3.27$ and $U_f = 0.92$. This shows that the distance between projected edges and image contours is minimized by the robust estimation procedure.

Figure 2. PPM estimation algorithm: Initial (left) and final (right) projection of the cube edges onto image contours.

4.3. Estimation of the intrinsic camera parameters and of the cube position and orientation

At this stage, a PPM has been estimated for each image. However, such projection matrices do not lead to a single, coherent intrinsic parameter matrix $I_c$. Let us denote by $[R_k | t_k]$ the cube orientation and position matrix expressed in the camera world coordinate system for the image $k$. $R_k$ depends on the three Cardan angles. Let us also denote by $(C_i)_{i=1...6}$ the 3D cube’s vertices of the two selected adjacent faces and $c^k_i$ their projections onto the image $k$ according to the PPM. Let $n$ be the total number of calibration images. The goal is to determine the intrinsic parameter matrix $I_c$ and the position and orientation matrices $[R_k | t_k]_{k=1...n}$ which minimize the following functional:

$$\sum_{k=1}^{n} \sum_{i=1}^{6} ||c^k_i - m_{I_c, [R_k | t_k]}(C_i)||^2,$$

where $m_{I_c, [R_k | t_k]}(C_i)$ is the projection of the point $C_i$ onto the image $k$ (equation (4)). The minimization of (13) is a non linear problem which is solved by applying the Levenberg-Marquardt algorithm. A first initialization of $I_c$ can be obtained by applying from the Faugeras-Toscani method.
5. EXPERIMENTAL RESULTS

The objective evaluation of camera calibration algorithms encounters several difficulties. For a constant camera setup, the intrinsic parameters are supposed to be fixed. However, in practice, the estimation of the intrinsic parameters varies from one calibration process to another. The accuracy of the camera calibration depends on:

(i) The accuracy of locating the pattern reference points (which can be manually or automatically detected) in the images.

(ii) The size of the pattern in the acquired images.

(iii) The pose of the pattern, which is determinant for the depth repartition of the considered reference points.

The influence of these elements in the camera calibration process has been poorly addressed in the literature. Only a few papers compare existing camera calibration method and underline such problems. González et al.\textsuperscript{47} present a comparative analysis of eight camera calibration methods. Their study concentrates on the stability and on the accuracy of the methods when the pattern is relocated or when the camera configuration varies. Zollner and Sablatnig\textsuperscript{48} investigate the performances of the three most widely used plane-based calibration algorithms: planar DLT (Direct Linear Transform), Tsai, and Zhang algorithms. In,\textsuperscript{49} Hemayed proposed a survey of self-calibration methods and presents two algorithms for both constant or varying intrinsic camera parameters. Salvi et al.\textsuperscript{17} present a detailed review about five of the mostly used calibration techniques. A noticeable effort has been done to present them under a unified evaluation framework. Their results show, with no surprise, that non-linear methods lead to a more accurate calibration than the linear ones.

In this paper, we have considered the issue of objective evaluation within the framework of a specific application, related to the estimation of the baseline parameter for commercial stereo vision system.

5.1. Estimation of a stereo camera baseline

In our experiments, we have used a Bumblebee camera. This two lens commercial stereo vision camera system is currently extensively used in many computer vision applications including people tracking, gesture recognition, mobile robotics... The system was calibrated using the following four methods: Tsai\textsuperscript{*}, Zhang\textsuperscript{†}, Faugeras-Toscani and the robust calibration proposed in this paper. In the case of the Tsai and Faugeras-Toscani methods, a set of 200 points of a non-planar target have been manually selected. For the Zhang method, 20 images of a planar target have been acquired. Two different calibrations have been here performed, one based on only 3 images, the other one using all the available images.

Figure 3 shows the two images of the cube used in our algorithm and two images from the twenty used in Zhang calibration.

![Figure 3. Illustrates the images used for calibration. (a) Two sample images used for Zhang calibration. (b) The pair of images used for robust calibration.](image)

In order to evaluate the performances, we have computed a two camera system baseline. The baseline $\delta$ of a stereo system is defined as the distance between the two camera centres. The accuracy of the baseline is essential for many stereo computer vision applications. In particular, the camera baseline determines the relation between image disparity $\text{disp}$ and object depth $Z$, according to the following equation: $\text{disp} = \frac{\delta f}{pZ}$, where $f$ is the focal length and $p$ the pixel size.

\textsuperscript{*}using Reg Willson CMU implementation: http://www.cs.cmu.edu/~rgw/TsaiCode.html

\textsuperscript{†}using OpenCV implementation: http://www.intel.com/technology/computing/opencv/index.htm
For the robust estimation method, we have computed the intrinsic parameters with automatic detection (AD) of the cube’s vertices, and manual selection (MS) of the cube’s vertices. In the case of manual selection, the experiment has been repeated five times. The same procedure has been adopted for Tsai and Faugeras-Toscani methods: the calibration was performed five times, with different manually selected points. The baseline values reported for the Tsai, Faugeras-Toscani and Robust Calibration (MS) correspond to the mean value averaged over the set of five experiments. The results are summarized in table 1.

<table>
<thead>
<tr>
<th>Calibration Method</th>
<th>$\delta$ (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tsai</td>
<td>12.55</td>
</tr>
<tr>
<td>Zhang (3 images)</td>
<td>12.48</td>
</tr>
<tr>
<td>Zhang (20 images)</td>
<td>11.97</td>
</tr>
<tr>
<td>Faugeras-Toscani</td>
<td>12.36</td>
</tr>
<tr>
<td>Robust calibration (MS)</td>
<td>12.05</td>
</tr>
<tr>
<td>Robust calibration (AD)</td>
<td>12.023</td>
</tr>
</tbody>
</table>

The value of the baseline provided by the manufacturer and used as ground truth is 12 cm. The proposed robust estimation with automatic detection of the reference points provides the most accurate result, with $\delta = 12.023$. When the reference points are manually selected, the results strongly degrade with a value of 12.05. However, the dispersion in this case is low, of only 1.8%. This shows the insensitivity of the proposed approach with respect to errors of the reference points. This nice propriety was tested on three different cameras with various resolutions: (a) Bumblebee (320x240 pixels), (b) BlueFox (1024x768 pixels) and (c) Sony (384x288 pixels). The dispersions $\sigma_u$ and $\sigma_v$ of the $u$ and $v$ image coordinates of the projected feature points are presented in table 2.

<table>
<thead>
<tr>
<th>Image (Resolution)</th>
<th>Bumblebee (320x240)</th>
<th>BlueFox (1024x768)</th>
<th>Sony (384x288)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_u$ (pixels)</td>
<td>0.367</td>
<td>0.429</td>
<td>0.443</td>
</tr>
<tr>
<td>$\sigma_v$ (pixels)</td>
<td>0.126</td>
<td>0.212</td>
<td>0.491</td>
</tr>
</tbody>
</table>

In all cases, the dispersions are inferior to 1 pixel, which proves the excellent stability of the solution.

Tsai and Faugeras-Toscani methods lead to inaccurate estimations. In addition, for these two methods, the dispersion is quite high with respective values of 7.3 and 8.2%. This shows the high sensitivity of such techniques to the manual selection of reference points.

Zhang’s method provides accurate results with a great number of images (in this case, 20) are used. However, performance rapidly degrades when the number of images is reduced. Thus, for three images, the error between the estimated and grand-truth baseline values is of 0.48 cm.

Experimental results show that the proposed method lead to the accurate estimation, with the advantages that calibration can be performed from a single image and no user interaction is required for selecting reference points.

Let us now analyse the behavior of the proposed method in the case of calibration from multiple images.

### 5.2. Intrinsic parameter estimation using multiple images

In order to test the performances of our algorithm in a multiple image estimation framework, we have considered a compact industrial MV BlueFOX camera, with image resolution 1024x768. Four different images of the pattern
object have been acquired, with the cube positioned at different depths within camera visual field. The four images are in figure 4.

![The four images acquired using the mvBlueFox camera.](image)

**Figure 4.** The four images acquired using the mvBlueFox camera.

The initialization of the intrinsic parameters was performed using the first image with Faugeras-Toscani method (results on column *Init* of the table 3). Let $p_i$ denote vector of intrinsic parameters $[\alpha_u \alpha_v u_0 v_0 \gamma]$ estimated uniquely from image $i$. Let $h$ denote the vector of intrinsic parameters estimated from all the four images. The mean absolute value difference (MAD) between multiple images and single image estimations is defined as: $MAD = \frac{1}{n} \sum_{i=1}^{n} |h - p_i|$, with $n$ being the number of images (here, $n = 4$). Estimation results are summarized in table 3.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Init</th>
<th>$p_1$</th>
<th>$p_2$</th>
<th>$p_3$</th>
<th>$p_4$</th>
<th>$h$</th>
<th>MAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_u$</td>
<td>1906.53</td>
<td>1832.20</td>
<td>1799.33</td>
<td>1864.61</td>
<td>1836.12</td>
<td>1812.37</td>
<td>20.69</td>
</tr>
<tr>
<td>$\alpha_v$</td>
<td>1939.15</td>
<td>1829.08</td>
<td>1807.21</td>
<td>1853.5</td>
<td>1844.43</td>
<td>1845.12</td>
<td>11.56</td>
</tr>
<tr>
<td>$u_0$</td>
<td>415.20</td>
<td>598.33</td>
<td>507.44</td>
<td>498.32</td>
<td>601.66</td>
<td>502.92</td>
<td>48.51</td>
</tr>
<tr>
<td>$v_0$</td>
<td>233.10</td>
<td>310.10</td>
<td>388.14</td>
<td>407.33</td>
<td>356.03</td>
<td>370.72</td>
<td>49.68</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>0.013</td>
<td>0.033</td>
<td>0.081</td>
<td>0.044</td>
<td>0.052</td>
<td>0.027</td>
<td>0.0035</td>
</tr>
<tr>
<td>$U$</td>
<td>8.56</td>
<td>4.69</td>
<td>5.13</td>
<td>5.02</td>
<td>4.12</td>
<td>3.05</td>
<td>-</td>
</tr>
</tbody>
</table>

The obtained results show that using multiple images increases the accuracy of the estimation. The value of the energy functional $U$ (equation (10)) is significantly reduced when multiple image estimation is performed.

**6. CONCLUSION AND FUTURE WORK**

In this paper, we have first presented an overview of the camera calibration methods. Three categories of approaches have been identified and discussed, including mono, multi and self (or auto) calibration methods. The analysis of the literature highlighted open issues in the field of camera calibration, which concern the minimization of the number of input images, the improvement of the accuracy and robustness, as well as the reduction of the amount of human interaction involved.

In order to respond to such problems, we proposed an accurate camera calibration algorithm based on a robust estimation of the projection perspective matrix. The method requires a minimum user intervention, and only one input image. A real application of stereoscopic camera estimation is used to show the usefulness of the proposed approach. The proposed method is compared to three well-known calibration techniques (Faugeras & Toscani, Zhang, and Tsai). Experimental results show that the robust calibration method achieves a high accuracy with only one input image and a minimal amount of human interaction.

Our future work will concern the extension of our method for taking into account distortion parameters. In a second time, we will study the pertinence of our approach within the framework of an application concerning 3D object reconstruction for collaborative interactions.
APPENDIX A. M-ESTIMATORS

Let us consider the linear model \( y_i = \langle X_i, \theta \rangle + \epsilon_i \) where \( y_i \) are the observed data, \( \epsilon_i \) is a white gaussian noise of standard deviation \( \sigma \), \( X_i \) are the system model known input vectors of dimension \((n, 1)\) and \( \theta \) is the unknown parameter vector.

The M-Estimator technique reduces the effect of outliers present in the data yielding to the minimization of the cost function: \( C(\theta) = \sum \rho(\epsilon_i) \) where \( \rho(.) \) is a symmetric, positive-define function with an unique minimum at zero, and chosen to be less increasing than the square function. In the case of least square estimation \( (\rho(x) = x^2) \), the estimation of \( \theta \) is impaired by the outliers data. The M-estimator lessens the outlier’s impact on the parameter estimation by assigning a smaller weight to any observation with atypical value.

Among the M-estimator functions, let us cite Huber and Tukey bisquare’s. In our algorithm, we use the Tukey function given by:

\[
\rho (r) = \begin{cases} 
\frac{c^2}{6} \left(1 - \left(\frac{r}{c}\right)^2\right)^3 & \text{if } |r| \leq c \\
\frac{c^2}{6} & \text{if } |r| \geq c 
\end{cases}
\]  

(14)

where the parameter \( c \) is a tuning parameter, commonly set to \( c = 4.685\sigma \).

The minimization of \( C(\theta) \) is performed by the Iterative Reweighted Least Square algorithm. Derivation of the cost function in \( \theta \) yields to the following equation:

\[
\sum_i \rho'(\epsilon_i) X_i = 0. 
\]  

(15)

Let us note \( \omega_i = \frac{\rho'(\epsilon_i)}{\epsilon_i} \), the weight factors. \( \omega_i \) takes its values between 0 and 1. It is near 0 when \( \epsilon_i \) has a high value, corresponding to an outlier and it is near 1 in the other case. Equation (15) gives:

\[
\left[ \sum_i \omega_i X_i X_i^t \right] \theta = \sum_i \omega_i y_i X_i. 
\]  

(16)

and \( \theta \) is the solution of the fixed point equation

\[
\theta = \left[ \sum_i \omega_i X_i X_i^t \right]^{-1} \sum_i \omega_i y_i X_i. 
\]  

(17)

REFERENCES


