Research Survey

Curriculum Vitae and Papers

Programa de Doctorat en Enginyeria Informàtica Industrial /

Tecnologies Avançades de Control del

Departament d'Electrònica, Informàtica i Automàtica

Universitat de Girona

Title of research survey:

Modelling Stereoscopic Systems for Robotic Applications

by:

Xavier Armangué Quintana

Director: Dr. Joaquim Salvi Mas

July 2001

Curriculum Vitae

CURRICULUM VITAE NORMALIZADO

APELLIDOS: **ARMANGUÉ QUINTANA**
NOMBRE: **XAVIER** SEXO: **V**
D.N.I.: **40 446 375** FECHA DE NACIMIENTO: **07 09 1976**
DIRECCION PARTICULAR: **DIAZ PACHECO, 70**
CIUDAD: **ROSES** DISTRITO POSTAL: **17480** TELEFONO: **646 03 46 13 / 972 25 65 50**

ESPECIALIZACION (CODIGO UNESCO):**1203.05-06 1203.17 2209.10 2209.15 2209.90 3304.06 3304.12 3304.17**

FORMACIÓN ACADÉMICA

DIPLOMA./LICENCIA./INGENIER.	CENTRO	FECHA
ING. TEC. EN INFORMÁTICA DE SISTEMAS	**EPS - UNIVERSIDAD DE GIRONA**	**10 07 1997**
INGENIERO EN INFORMÁTICA	**EPS - UNIVERSIDAD DE GIRONA**	**24 09 1999**

DOCTORADO

DIRECTOR(ES) DE TESIS:

SITUACION ACTUAL Y FECHA DE INICIO: **BECARIO FPI** **01-06-2000**
ORGANISMO: **UNIVERSITAT DE GIRONA**
CENTRO: **ESCOLA POLITÈCNICA SUPERIOR**
DEPT./SECC./UNIDAD ESTR.: **DEP. DE ELECTRÒNICA, INFORMÀTICA Y AUTOMÀTICA**
DIRECCION POSTAL: **Avda. Lluís Santaló s/n. 17003 Girona**
TELEFONO (indicar prefijo, número y extensión): **972 41 84 83**

CONTRATO	☐	OTRAS SITUACIONES ……		
PLANTILLA	☐	DEDICACION:	A TIEMPO COMPLETO	■
BECARIO	■		A TIEMPO PARCIAL	☐
INTERINO	☐		ESPECIFICAR ……	
EXCEDENCIA	☐			

ACTIVIDADES ANTERIORES DE CARACTER CIENTIFICO O PROFESIONAL

FECHAS	PUESTO	INSTITUCION
SEPT. 1997 – JUN. 1998	**Contrato a tiempo parcial**	**UdG - ANTEX S.A.**
OCT. 1998 – JUN. 1999	**Beca de Colaboración de Departamento**	**EPS (UdG)**
NOV. 1999 – MAY 2000	**Beca del CERTAP**	**EPS (UdG)**

IDIOMAS DE INTERES CIENTIFICO (R = Regular, B = Bien, C = Correctamente)

IDIOMA	HABLA	LEE	ESCRIBE
FRANCÉS	**B**	**C**	**B**
INGLÉS	**R**	**B**	**R**

FECHA CUMPLIMENTACION: **Julio 2001** FIRMA:

PROYECTOS DE INVESTIGACION

CLAVE: IP = Investigador Principal; I = Investigador; D = Director de Investigación; C = Colaborador; R = Realizador

ENTIDAD FINANCIADORA**: MINISTERIO DE EDUCACION Y CIENCIA**
PROGRAMA: **P.E.T.R.I.**
Título: **DESARROLLO DE UN EQUIPO DE VISIÓN POR COMPUTADOR PARA LA INSPECCIÓN EN TIEMPO REAL DE DEFECTOS MICROSCÓPICOS, EN CADENAS DE PRODUCCIÓN DE HILATURA O EN SISTEMAS DE PRODUCCIÓN CON NECESIDADES EQUIVALENTES.**
Número de Proyecto**: 95-0145-OP** Importe: **7.600.000** Año: **1997/1998**
INVESTIGADOR PRINCIPAL: **JOAN BATLLE**
CLAVE: **C**

ENTIDAD FINANCIADORA**: GENERALITAT DE CATALUNYA**
PROGRAMA: **CERTAP**
Título: **DESARROLLO DE UN PROTOTIPO INDUSTRIAL DE ROBOT MÓVIL PARA UN ENTORNO PRODUCTIVO Y UNO DE PREINDUSTRIAL PARA UN ENTORNO DE SERVICIOS.**
Número de Proyecto**:** Importe: **6.000.000** Año: **1999/2002**
INVESTIGADOR PRINCIPAL: **JOAQUIM SALVI** CLAVE: **C**

ENTIDAD FINANCIADORA**: MINISTERIO DE EDUCACION Y CIENCIA**
PROGRAMA: **CICYT**
Título: **DESARROLLO DE SISTEMAS DE VISIÓN POR COMPUTADOR ESPECÍFICOS PARA TAREAS DE RECONSTRUCCIÓN Y ANÁLISIS DE ESCENAS EN ROBÓTICA MÓVIL.**
Número de Proyecto**: TAP99-0443-CO5-01** Importe: **7.000.000** Año: **2000/2002**
INVESTIGADOR PRINCIPAL: **JOAN JOSÉ SERRANO MARTÍN (UPV)**
INVESTIGADOR PRINCIPAL SUBPROYECTO: **JOAQUIM SALVI** CLAVE: **C**

PUBLICACIONES
(no incluir proceedings, ni printings, ni abstracts de Congresos)

CLAVE: L = libro completo; M = mapa; CL = capítulo de libro; A = artículo; AP = artículo difusión/periodístico; R = review, E = editor; T = traductor

AUTORES (p.o. de firma): **J. SALVI, X. ARMANGUÉ, J. BATLLE**
TITULO: **A COMPARATIVE REVIEW OF CAMERA CALIBRATING METHODS WITH ACCURACY EVALUATION**
REF. REVISTA/LIBRO: **PATTERN RECOGNITION**
Volumen: **PENDIENTE DE PUBLICACIÓN** Número: Pág.: Págfi.: Año: **2001** Ciudad: País:
CLAVE: **A** NUMERO: ORDEN: ISSN/ISBN: **0031-3203** D. Legal:

PARTICIPACION EN CONTRATOS DE INVESTIGACION DE ESPECIAL RELEVANCIA CON EMPRESAS Y/O ADMINISTRACIONES

CLAVE: IR = Investigador Principal; I = Investigador; D = Director de Investigación; C = Colaborador; R = Realizador

EMPRESA/ADMINISTRACION FINANCIADORA: **UdG - ANTEX S.A.**
TITULO DEL CONTRATO: **COLABORADOR DE INVESTIGACIÓN**
DURACION DESDE: **01 09 1997** HASTA: **30 06 1998**
INVESTIGADOR RESPONSABLE: **JOAN BATLLE** CLAVE: **C**

ESTANCIAS EN CENTROS EXTRANJEROS (superiores a cuatro semanas)

CLAVE: D = Doctorando, P = Postdoctoral, I = Invitado, C = Contratado, O = Otras (especificar)

CENTRO: **INSTITUTE OF SYSTEMS AND ROBOTICS – UNIVERSITY OF COIMBRA**
LOCALIDAD: **COIMBRA** PAIS: **PORTUGAL** AÑO: **2001** DURACION: **4 MESES**
TEMA: **EGOMOTION ESTIMATION USING THE DIFFERENTIAL EPIPOLAR CONSTRAINT**
CLAVE: **D**

CONGRESOS
(Presentación de comunicaciones y ponencias)

AUTORES (p.o. de firma): **X. ARMANGUÉ, R. GARCÍA, J. BATLLE, X. CUFÍ, J.LL. DE LA ROSA, P. RIDAO**
TITULO: **IMPLEMENTACIÓN DE UN ALGORITMO PARALELO DE SEGUIMIENTO DE MÚLTIPLES MICROROBOTS UTILIZANDO VHDL.**
ORGANIZADOR: **PROCEEDINGS DEL 2ON CONGRÈS CATALÀ D'INTEL.LIGÈNCIA ARTIFICIAL, CCIA'99**
Carácter: Nacional ■ Internacional ☐
Volumen: Número: **18-19** Págin.: **356** Págfi.: **364** Año: **1999** Ciudad: **GIRONA**
NUMERO: ORDEN: PAIS: **ESPAÑA**

AUTORES (p.o. de firma): **X. ARMANGUÉ, J. SALVI, J. BATLLE**
TITULO: **A COMPARATIVE REVIEW OF CAMERA CALIBRATING METHODS WITH ACCURACY EVALUATION**
ORGANIZADOR: **V IBERO AMERICAN SYMPOSIUM ON PATTERN RECOGNITION, SIARP 2000**
Carácter: Nacional ☐ Internacional ■
Volumen: Número: Págin.: **183** Págfi.: **194** Año: **2000** Ciudad: **LISBON**
NUMERO: ORDEN: PAIS: **PORTUGAL**

AUTORES (p.o. de firma): **X. ARMANGUÉ, J. PAGÈS, J. SALVI, J. BATLLE**
TITULO: **COMPARATIVE SURVEY ON ESTIMATING THE FUNDAMENTAL MATRIX**
ORGANIZADOR: **IX SIMPOSIUM NACIONAL DE RECONOCIMIENTO DE FORMAS Y ANÁLISIS DE IMÁGENES, SNRFAI 2001**
Carácter: Nacional ■ Internacional ☐
Volumen: **1** Número: Págin.: **227** Págfi.: **232** Año: **2001** Ciudad: **CASTELLÓN**
NUMERO: ORDEN: PAIS: **ESPAÑA**

AUTORES (p.o. de firma): **J. PAGÈS, X. ARMANGUÉ, J. SALVI, J. FREIXENET, J. MARTÍ**
TITULO: **A COMPUTER VISION SYSTEM FOR AUTONOMOUS FORKLIFT VEHICLES IN INDUSTRIAL ENVIRONMENTS**
ORGANIZADOR: **THE 9TH. MEDITERRANEAN CONFERENCE ON CONTROL AND AUTOMATION, MEDS 2001**
Carácter: Nacional ☐ Internacional ■
Volumen: Número: Págin.: **62** Págfi.: **69** Año: **2001** Ciudad: **DUBROVNIK**
NUMERO: ORDEN: PAIS: **CROATIA**

AUTORES (p.o. de firma): **J. SALVI , X. ARMANGUÉ, J. PAGÈS**
TITULO: **A SURVEY ADDRESSING THE FUNDAMENTAL MATRIX ESTIMATION PROBLEM**
ORGANIZADOR: **IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, ICIP 2001**
Carácter: Nacional ☐ Internacional ■
Volumen: Número: Págin.: Págfi.: Año: **2001** Ciudad: **THESSALONIKI**
NUMERO: ORDEN: PAIS: **GREECE**

TESIS DE LICENCIATURA Y TRABAJOS FIN DE CARRERA DIRIGIDOS O CODIRIGIDOS

TITULO: **SISTEMA DE VISIÓ PER A ROBOTS MÒBILS. ANÀLISI D'IMATGES ESTÈREO PER DOTAR EL ROBOT AMB LA CAPACITAT D'ESQUIVAR I CARACTERITZAR OBJECTES**
DOCTORANDO: **ANNA BOSCH RUÉ**
UNIVERSIDAD: **UNIVERSIDAD DE GIRONA** FACULTAD/ESCUELA: **ESCUELA POLITÉCNICA SUPERIOR**
CALIFICACION: **PENDIENTE DE CUALIFICACIÓN** AÑO: **2001**

GRANDES EQUIPOS QUE UTILIZA O HA UTILIZADO

CLAVE: R = Responsable, UA = Usuario Asiduo, UO = Usuario Ocasional

EQUIPO: **VAX**
CLAVE: **UA** FECHA: **1994**

EQUIPO: **UNIX - WORK STATION**
CLAVE: **UA** FECHA: **1994-2001**

ACTIVIDADES CIENTIFICAS/TECNICAS

LINEAS DE INVESTIGACION O DESARROLLO EN QUE HA TRABAJADO

LINEA: **DISEÑO DE CIRCUITOS INTEGRADOS VLSI**
CENTRO: **UdG** FECHAS: **1997/1999**
PALABRAS CLAVE: **DC-VLSI, ABEL, VHDL**

LINEA: **VISIÓN POR COMPUTADOR**
CENTRO: **UdG** FECHAS: **1999/2000**
PALABRAS CLAVE: **CALIBRATION, STRUCTURED LIGHT, 3D RECONSTRUCTION.**

LINEA: **ROBÓTICA**
CENTRO: **UdG**. FECHAS: **1999/2000**
PALABRAS CLAVE: **UNDERWATER ROBOTS, MOBILE ROBOTS**

TECNICAS O ESPECIALIDADES QUE DOMINA

TEC/ESP:	**PROGRAMACIÓN EN LENGUAJES IMPERATIVOS C, PASCAL, BASIC …**
PALABRAS CLAVE:	**C, PASCAL, BASIC**
TEC/ESP:	**PROGRAMACIÓN EN LENGUAJES FUNCIONALES LISP**
PALABRAS CLAVE:	**LISP**
TEC/ESP:	**PROGRAMACIÓN EN LENGUAJES LOGICOS PROLOG**
PALABRAS CLAVE:	**PROLOG**
TEC/ESP:	**PROGRAMACIÓN EN LENGUAJES ORIENTADOS A OBJETOS C++, JAVA, VISUAL BASIC, DELPHI**
PALABRAS CLAVE:	**C++, JAVA, VISUAL BASIC, DELPHI**
TEC/ESP:	**PROGRAMACIÓN CONCURENTE Y PARALELA ADA**
PALABRAS CLAVE:	**ADA**
TEC/ESP:	**PROGRAMACIÓN EN LENGUAJE MATEMÁTICO MATLAB**
PALABRAS CLAVE:	**MATLAB**
TEC/ESP:	**PROGRAMACIÓN EN ENSAMBLADOR (x86, VAX)**
PALABRAS CLAVE:	**ASSEMBLER, COMPUTER STRUCTURE, COMPUTER ARCHITECTURE**
TEC/ESP:	**LENGUAJES DE PROGRAMACIÓN DE HARDWARE**
PALABRAS CLAVE:	**ABEL - VHDL**
TEC/ESP:	**ADQUISICIÓN Y PROCESAMIENTO DE IMAGENES**
PALABRAS CLAVE:	**IMAGE PROCESSING**
TEC/ESP:	**VISIÓN POR COMPUTADOR**
PALABRAS CLAVE:	**COMPUTER VISION**
TEC/ESP:	**ROBÓTICA**
PALABRAS CLAVE:	**ROBOTICS**

BECAS

CLAVE: F = Formación; P = Perfeccionamiento; C = Colaboración; E = Estancia; V = Viaje

OTORGANTE: **MINISTERIO DE EDUCACION Y CIENCIA**
AÑO: **1997/1998** IMPORTE: **350.000** CLAVE: **C**

OTORGANTE: **GENERALITAT DE CATALUNYA**
AÑO: **1997/1998** IMPORTE: **670.800** CLAVE: **C**

OTORGANTE: **MINISTERIO DE EDUCACION Y CIENCIA / MINISTERIO DE CIENCIA Y TECNOLOGIA**
AÑO: **2000/2001** IMPORTE: CLAVE: **F**

AYUDAS

CLAVE: I = Infraestructura; E = Edición

OTORGANTE:
AÑO: IMPORTE: CLAVE:

CURSOS Y SEMINARIOS RECIBIDOS

CURSO ■ SEMINARIO □ CREDITOS: **4** FECHA: **OCTUBRE 1999- ENERO 2000**
CENTRO: **ESCOLA POLITÈCNICA SUPERIOR (U.d.G.)**
ORGANITZADOR: **T. JOVÉ, J. BATLLE, R. FABREGAT, J. SALVI**
MATERIA: **INTEGRACIÓ DE SISTEMES: DE LES ARQUITECTURES A LES COMUNICACIONS**

CURSO ■ SEMINARIO □ CREDITOS: **4** FECHA: **OCTUBRE 1999- ENERO 2000**
CENTRO: **ESCOLA POLITÈCNICA SUPERIOR (U.d.G.)**
ORGANITZADOR: **J.L. DE LA ROSA, J.VEHÍ, J. MELENDEZ, J. COLOMER**
MATERIA: **SUPERVISIÓ DE PROCESSOS INDUSTRIALS**

CURSO ■ SEMINARIO □ CREDITOS: **4** FECHA: **FEBRERO 2000- JUNIO 2000**
CENTRO: **ESCOLA POLITÈCNICA SUPERIOR (U.d.G.)**
ORGANITZADOR: **X. CUFÍ, J. MARTÍ**
MATERIA: **APLICACIONS DE LA VISIÓ PER COMPUTADOR AL CONTROL DE QUALITAT**

CURSO ■ SEMINARIO □ CREDITOS: **4** FECHA: **FEBRERO 2000- JUNIO 2000**
CENTRO: **ESCOLA POLITÈCNICA SUPERIOR (U.d.G.)**
ORGANITZADOR: **SEGMENTACIÓ I INTERPRETACIÓ D'ESCENES**
MATERIA: **SEGMENTACIÓ I INTERPRETACIÓ D'ESCENES**

CURSO ■ SEMINARIO □ CREDITOS: **4** FECHA: **FEBRERO 2000- JUNIO 2000**
CENTRO: **ESCOLA POLITÈCNICA SUPERIOR (U.d.G.)**
ORGANITZADOR: **J. SALVI**
MATERIA: **PERCEPCIÓ 3D**

CURSO □ SEMINARIO ■ FECHA: **FEBRERO 2000**
CENTRO: **ESCOLA POLITÈCNICA SUPERIOR (U.d.G.)**
ORGANITZADOR: **J. SALVI**
MATERIA: **CODED STRUCTURED LIGHT**

OTRAS ACTIVIDADES RELEVANTES DE INTERES CIENTIFICO, TECNICO O ACADEMICO

ACTIVIDAD: **PUBLICACIÓN DOCENTE**
AUTORES : **J. Martí, L. Delgado, X. Armangué**
TITULO : **Disseny de circuits digitals amb dispositius FPGA - Prácticas**
Volumen : Num : Pag. Inicio : **1** Pag. Fin : **26** Año : **1998** Ciudad : **Girona** Pais : **España**

ACTIVIDAD: **REPORT DE INVESTIGACIÓN**
AUTORES : **J. Salvi, X. Armangué**
TITULO : **Mètode de Calibració de Càmeres.**
Volumen : Num : **IIiA 00-04-RR**
Pag. Inicio : **1** Pag. Fin : **32** Año : **2000** Ciudad : **Girona** Pais : **España**

Publications and Contributions in Conferences

Publications

- X. Armangué and J. Salvi. *Overall view regarding fundamental matrix estimation.* Submitted to Image and Vision Computing, **IVC**.

- J. Salvi, X. Armangué and J. Batlle. *A comparative review of camera calibrating methods with accuracy evaluation.* Pattern Recognition, **PR**. Accepted to be published.

Contributions Related with this Research Survey

- J. Salvi, X. Armangué and J. Pagès. *A survey addressing the fundamental matrix estimation problem.* IEEE International Conference on Image Processing, **ICIP 2001**, Thessaloniki, Greece, October 7–10, 2001.

- X. Armangué, J. Pagès, J. Salvi and J. Batlle. *Comparative survey on estimating the fundamental matrix.* IX Simposium Nacional de Reconocimiento de Formas y Análisis de Imágenes, **SNRFAI 2001**, pp 227–232, Castelló, Spain, May 14–18, 2001.

- X. Armangué, J. Salvi and J. Batlle. *A comparative review of camera calibrating methods with accuracy evaluation.* Proceedings of 5th Ibero-American Symposium on Pattern Recognition, **SIARP 2000**, pp 183–194, Lisboa, Portugal, September 11–13, 2000.

Others Research Contributions

- J. Pagès, X. Armangué, J. Salvi, J. Freixenet, J. Martí. *A computer vision system for autonomous forklift vehicles in industrial environments*. The 9th. Mediterranean Conference on Control and Automation, **MED 2001**, Dubrovnik, Croatia, June 27–29, 2001.

- X. Armangué, R. García, J. Batlle, X. Cufí, J.Ll. De La Rosa, P. Ridao. *Implementació d'un algorisme paral·lel de seguiment de múltiples microrobots mitjançant VHDL*. Proceedings del 2on Congrés Català d'Intel·ligència Artificial, **CCIA 1999**, Girona, Spain, Octubre 25–27, 1999.

Overall view regarding fundamental matrix estimation

Xavier Armangué and Joaquim Salvi

Computer Vision and Robotics Group
Institute of Informatics and Applications
University of Girona
Av. Lluís Santaló, s/n, E-17071 Girona (Spain)
{armangue,qsalvi}@eia.udg.es

Image and Vision Computing
Submitted in July 2001

Reception Letter

Date: Mon, 16 Jul 2001 17:20:48 +0100
From: Professor Keith Baker <keith.baker@reading.ac.uk>
To: 'qsalvi@eia.udg.es' <qsalvi@eia.udg.es>
Subject: IVC 2234 - Acknowledgement and PDF Request

Dear Dr Salvi,

IVC 2234
Overall View Regarding Fundamental Matrix Estimation
X Armangue and J Salvi

Thank you for sending me your above manuscript for consideration for possible publication in Image and Vision Computing Journal. In future communications please could I ask that you use the reference IVC 2234.

Please will you send me a PDF version of your manuscript.

Regards,
Keith Baker.

Professor Keith D Baker
Editor in Chief
Image and Vision Computing Journal
Email: Keith.Baker@Reading.ac.uk

Tel: +44 (0) 118 931 8602
Fax: +44 (0) 118 975 1822
Mobile: +44 (0) 7850 722 782

Overall View Regarding Fundamental Matrix Estimation

Xavier Armangué and Joaquim Salvi*

Computer Vision and Robotics Group. Institute of Informatics and Applications.
University of Girona. Av. Lluís Santaló, s/n, E-17071 Girona (Spain)
{armangue,qsalvi}@eia.udg.es

*Corresponding Author

Corresponding Author

Joaquim Salvi.
Computer Vision and Robotics.
Institute of Informatics and Applications.
University of Girona.
Avda. Lluís Santalo s/n.
E-17071 GIRONA (SPAIN)
Ph: +34 972 41 8483
FAX: +34 972 41 8098
e-mail: qsalvi@eia.udg.es

Summary

In this article we present a comparative study of the most commonly used fundamental matrix estimation methods of the last few decades. The techniques cover a wide range of methods from the classical linear estimation such as 7-points, least squares and Eigen analysis up to some of the most used robust methods such as M-Estimators, Least Median of Squares and Random Sampling. Hence, this study is presented describing a total of 10 methods and up to 15 different implementations. The different methods have been programmed and their accuracy analysed with both synthetic and real images. The code is available in Internet and the whole application can be download from `http://eia.udg.es/~armangue/research`.

Experimental results prove that: a) Linear methods are quite good if the points are well located in the image and the corresponding problem previously solved. Moreover, they require a reduced computation time. b) Iterative methods can cope with some gaussian noise in the localisation of image points due to both image blurring and segmentation. However, they became really inefficient in the presence of outliers given by false matching. c) Robust methods can cope with both gaussian noise in point localisation and outliers but they usually require a high computation time. Moreover, experimental results show that eigen analysis gives better result than a classic least-squares minimisation both in linear or iterative methods. We have also pointed out that data have to be normalised and the best results have been obtained by using the method proposed by Hartley in his book published last year. Finally, a rank-2 matrix is preferred because it models the epipolar geometry with a single epipole defined by the intersection of all the epipolar lines. The article describes every surveyed method in detail and its accuracy is compared with the others, giving a fresh look that may be useful for any reader who want to be introduced in the field or either is searching a sort of method for his application.

Fundamental matrix estimation is a key point in computer vision because an accurate estimation is required in order to compute the epipolar geometry of a stereoscopic system. Some applications of the epipolar geometry are: a) the simplification of the image matching which it is reduced to a searching along the epipolar line; b) the estimation of the camera motion mounted either to a robotic arm or a mobile robot with useful application in grasping and mobile navigation; and c) scene reconstruction with several applications in industrial inspection, prototyping and mould generation.

Overall View Regarding Fundamental Matrix Estimation[1]

Abstract

Epipolar geometry is a key point in computer vision and the fundamental matrix estimation is the only way to compute it. This article surveys several methods of fundamental matrix estimation which have been classified into linear methods, iterative methods and robust methods. All of these methods have been programmed and their accuracy analysed using real images. A summary, accompanied with experimental results, is given and the code is available in Internet:
`http://eia.udg.es/~armangue/research`.

Keywords: Epipolar Geometry, Fundamental Matrix, Performance Evaluation.

1 Introduction

The estimation of three-dimensional information is a crucial problem in computer vision. At present, there are two approaches to accomplish this task. The first approach is based on a previous camera calibration, which is the computation of the model of the imaging sensors relating 3D object points to their 2D projections on the image plane. A thorough survey on camera modelling and calibration was presented by Ito in 1991 [21] and this subject has been widely studied during the past decades. Actually, basic methods model the imaging sensor through a single transformation matrix [2, 11]. Other methods fix geometrical constraints in such matrix introducing a set of intrinsic and extrinsic camera parameters [10]. Moreover, lens distortion introduces two non-linear equations which model the image curvature obtaining a more accurate model. Some authors have considered only radial lens distortion [37], while others considered tangential distortion [38]. Finally, once the system is calibrated, the camera model can be used either to estimate the 2D projection of an object point or to compute the 3D optical ray passing through a given 2D image projection and onto the scene. Therefore, at least two optical rays are needed to compute the 3D position of the object point by means of triangulation.

Calibration can not be used in active systems due to its lack of flexibility. Note that in active systems, the optical and geometrical characteristics of the cameras might change dynamically depending on the imaging scene and camera motion. The second approach then is based on computing either the epipolar geometry between both imaging sensors [8] or an Euclidean reconstruction [16]. Euclidean reconstruction is achieved through previous knowledge of the scene [29] such as projective basis and invariants. However, this assumption is difficult to integrate into many computer vision applications, while epipolar geometry is based only on image correspondences.

[1]Work funded by Spanish project CICYT TAP99-0443-C05-01

An application of scene reconstruction using Epipolar geometry was first published by Longuet-Higgins in 1981 [25]. Since that time, a great deal of effort has been put into increasing the knowledge [8, 19]. Many articles have been presented on self-calibrated and uncalibrated systems as a result of the boom in the 1990's. For instance, in 1992 Faugeras published a brief survey on self-calibration and the derived Kruppa equations which are used to estimate the camera parameters from the epipolar geometry [9]. Basically, intrinsic parameters of both cameras and the position and orientation of one camera related to the other can be extracted by using Kruppa equations [18]. In the same year, Faugeras also gave an answer to the question *"What can be seen in three dimensions with an uncalibrated stereo rig?"* [7]. Hartley also did a lot of work with geometry and how it is contained in the essential and the fundamental matrix [13] as well as the estimation of the camera pose [15]. Two years later, Deriche et al. presented a robust method for recovering epipolar geometry based on a matching by correlation and detecting the outliers [6]. As a result, Hartley studied the geometry involved in a rotating camera [17] while Li studied the geometry of a head-eye system [24] and Luong et al. introduced a Canonic representation [28]. Also, in 1994, Luong and Faugeras published an interesting article on analyzing the stability of the Fundamental matrix due to uncertainty in the epipole computation, noise in the image point localization, camera motion, and so on [26].

Some applications of epipolar geometry are the simplification of the image matching in stereoscopic systems [4], the estimation of camera motion [22] and scene reconstruction [43]. It is important, therefore, to develop accurate techniques to compute it. Classic linear methods are mainly based on least-squares minimization [40] and eigen values minimization [34]. Other methods are based on optimizing linear methods by means of iteration [23]. Robust methods are based on computing a more accurate geometry detection and removing false matchings [31, 40]. Robust computation is still a subject for wide research focusing mainly on proposing new estimators to improve the accuracy of the fundamental matrix and on reducing computation expenses [3, 33, 36].

This article surveys up to fifteen of the most widely used techniques in computing the fundamental matrix such as the 7-point, least-squares and eigen analysis linear techniques among others and robust techniques such as M-estimators, LMedS, RANSAC and so on. All these techniques have been programmed and their accuracy analyzed in synthetic and real scenarios. This article is divided as follows. First, a brief introduction to epipolar geometry is presented. Then, all the surveyed methods are described in section 3 analyzing their advantages and drawbacks with respect to the previous methods. Section 4 deals with the experimental results obtained with both synthetic and real images. Finally, the article ends with conclusions.

2 Epipolar Geometry

Given a 3D object point $M = ({}^{W}X,\, {}^{W}Y,\, {}^{W}Z,\, 1)^{\mathrm{T}}$ expressed with respect to a world coordinate system $\{W\}$, and its 2D projection on the image plane in pixels $m = ({}^{I}X,\, {}^{I}Y)^{\mathrm{T}}$, both points are related to a projective transformation matrix as shown in equation (1),

$$s\, m = {}^{I}\mathbf{P}_{W}\, M \tag{1}$$

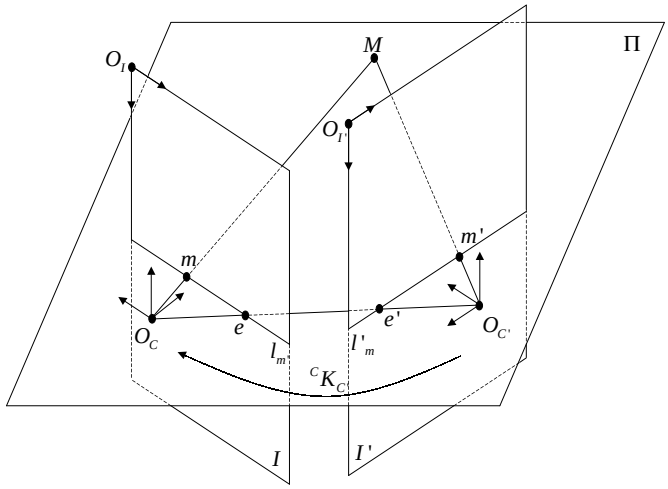


Figure 1: The geometric relation between two pinhole cameras.

in which s is a scale factor and ${}^I\mathbf{P}_W$ is a 3 $\times$ 4 matrix, which can be decomposed as

$$ {}^I\mathbf{P}_W = {}^I\mathbf{A}_C \, {}^C\mathbf{K}_W \tag{2} $$

in which ${}^I\mathbf{A}_C$ is a 3 $\times$ 4 matrix relating the metric camera coordinate system located at the focal point O_C to the image coordinate system located at the top-left corner of the image plane in pixels, that is the optical and internal geometry of the camera. Moreover, ${}^C\mathbf{K}_W$ is a 4 $\times$ 4 matrix which relates the camera coordinate system $\{C\}$ to the world coordinate system $\{W\}$, that is the position and orientation of the camera in the scene.

$$ {}^C\mathbf{K}_W = \begin{pmatrix} {}^C\mathbf{R}_W & {}^C t_W \\ 0_{1x3} & 1 \end{pmatrix} \tag{3} $$

Then, epipolar geometry defines the geometry between the two cameras creating a stereoscopic system or geometry between two different positions of a mobile camera. Given an object point M and its 2D projections m and m' on both image planes, the 3 points define a plane Π, which intersects both image planes at the epipolar lines $l_{m'}$ and l'_m respectively, as shown in Figure 1. Note that the same plane Π can be computed using both focal points O_C and $O_{C'}$ and a single 2D projection, which is the principle to reduce the correspondence problem to a single scanning along the epipolar line. Moreover, the intersection of all the epipolar lines defines an epipole on both image planes, which can also be obtained by intersecting the line defined by both focal points O_C and $O_{C'}$ with both image planes.

All the epipolar geometry is contained in the so called fundamental matrix as shown in equation (4).

$$ m^{\mathrm{T}} \mathbf{F} m' = 0 \tag{4} $$

The fundamental matrix contains the intrinsic parameters of both cameras and the rigid transformation of one camera related to the other, which depends on which camera has been considered as the origin of the world

coordinate system. In equation (5), the origin of the world coordinate system coincides with the coordinate system of the second camera, located at $O_{C'}$.

$$\mathbf{F} = {}^{I}\mathbf{A}_{C}{}^{-\mathrm{T}}[{}^{C}t_{C'}]_{\times}{}^{C}\mathbf{R}_{C'}\,{}^{I'}\mathbf{A}_{C'}{}^{-1} \tag{5}$$

A particular case of the fundamental matrix is the essential matrix. When the intrinsic camera parameters are known, it is possible to simplify equations (4) and (5) obtaining

$$q^{\mathrm{T}}\mathbf{E}\,q' = 0 \tag{6}$$

where,

$$q = {}^{I}\mathbf{A}_{C}{}^{-1}m, \qquad \mathbf{E} = [{}^{C}t_{C'}]_{\times}{}^{C}\mathbf{R}_{C'}, \qquad q' = {}^{I'}\mathbf{A}_{C'}{}^{-1}m' \tag{7}$$

The matrix $\mathbf{E}$ is called essential [19].

3 Estimating the Fundamental Matrix

In the last few years, several methods to estimate the fundamental matrix have been proposed, which can be classified into lineal, iterative and robust methods. Lineal and iterative methods can cope with bad point localization in the image plane due to noise in image segmentation. Robust methods can cope with both image noise and outliers, i.e. wrong matching between point correspondences in both image planes. All of these methods are based on solving a homogeneous system of equations which can be deduced from equation (4) rewriting it in the following way:

$$\mathbf{U}f = 0 \tag{8}$$

where,

$$f = (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^{\mathrm{T}} \tag{9}$$

$$U = \begin{pmatrix} {}^{I}X_1\,{}^{I}X'_1 & {}^{I}X_1\,{}^{I}Y'_1 & {}^{I}X_1 & {}^{I}Y_1\,{}^{I}X'_1 & {}^{I}Y_1\,{}^{I}X'_1 & {}^{I}Y_1 & {}^{I}X'_1 & {}^{I}Y'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ {}^{I}X_n\,{}^{I}X'_n & {}^{I}X_n\,{}^{I}Y'_n & {}^{I}X_n & {}^{I}Y_n\,{}^{I}X'_n & {}^{I}Y_n\,{}^{I}X'_n & {}^{I}Y_n & {}^{I}X'_n & {}^{I}Y'_n & 1 \end{pmatrix} \tag{10}$$

It is important to note that there are only 7 independent parameters and 9 unknowns. The 7 independent parameters are given by two independent columns and the scale factor forcing the fundamental matrix to be rank-2 [40].

3.1 Linear Methods

The linear method of the *seven points* is based on computing the fundamental matrix by using only seven point correspondences [40]. Due to the homogeneity of the equations, the solution is a set of matrices of the form

$$\mathbf{F} = \alpha\mathbf{F}_1 + (1-\alpha)\mathbf{F}_2 \tag{11}$$

By forcing the rank of the matrix to be equal to 2 and using the expression $\det[\alpha\mathbf{F}_1 + (1-\alpha)\mathbf{F}_2]$, a cubic polynomial is obtained which has to be solved to obtain α and then $\mathbf{F}$. The main advantage of this method is that

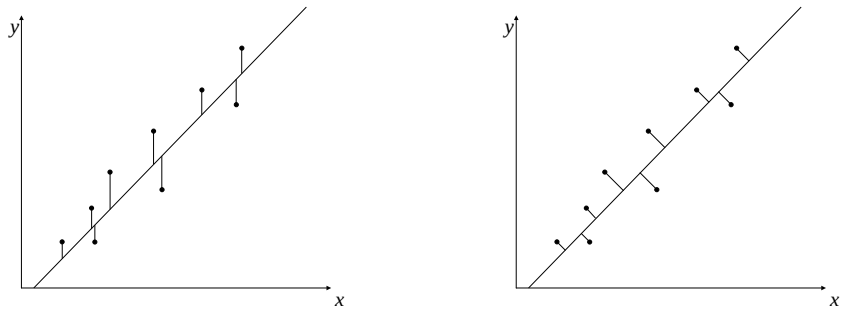


Figure 2: (a) Classical least-squares. (b) Orthogonal least-squares.

a fundamental matrix can be estimated by using only seven points. However this fact becomes a drawback when some points are poorly located. Moreover, the 7-points method cannot be applied in the presence of redundancy. Hence, it can not be applied using n points where $n > 7$.

Another interesting method is the *8-points method*, in which the redundancy of points permits minimizing the error in estimating $\mathbf{F}$. The equation to minimize in the 8-points method is the residual of equation (4), that is:

$$\min_{\mathbf{F}} \sum_{i} \left(m_i^{\mathrm{T}} \mathbf{F} m_i' \right)^2 \tag{12}$$

The classical method to solve such an equation is the *least-squares technique* of forcing one of the components of $\mathbf{F}$ to be the unity [27]. This simplification can be assumed because $\mathbf{F}$ is always defined up to a scale factor. Then, the equation to solve is

$$f' = \left(\mathbf{U'}^{\mathrm{T}} \mathbf{U'} \right)^{-1} \mathbf{U'}^{\mathrm{T}} c_9 \tag{13}$$

in which $\mathbf{U}'$ is a matrix containing the first eight columns of $\mathbf{U}$, c_9 is the last column of $\mathbf{U}$ (see also equation (10)) and f' is a vector containing the first eight elements of f. Note that the last element of f is 1.

A variant of the 8-points method can be applied if equation 12 is solved by using *eigen analysis*, also called *orthogonal least-squares technique* [34]. In this case $\mathbf{F}$ can be determined from the eigen vector corresponding to the smallest eigen value of $\mathbf{U}^{\mathrm{T}}\mathbf{U}$. The difference between this method and the classical least-squares resides in the form of calculating the error between correspondences and epipolar lines, as shown in Figure 2, where an orthogonal minimization is much more realistic.

The last linear method we surveyed is the *analytic method with rank-2 constraint* which imposes the rank-2 constraint in minimization [40]. Then, the matrix $\mathbf{U}'$ is defined as the composition of the first seven columns of $\mathbf{U}$ and c_8 and c_9 are defined as the eighth and ninth columns of $\mathbf{U}$ respectively, so that $\mathbf{F}$ can be computed as

$$f' = -f_8 \left(\mathbf{U'}^{\mathrm{T}} \mathbf{U'} \right)^{-1} \mathbf{U'}^{\mathrm{T}} c_8 - f_9 \left(\mathbf{U'}^{\mathrm{T}} \mathbf{U'} \right)^{-1} \mathbf{U'}^{\mathrm{T}} c_9 \tag{14}$$

in which f' is the vector containing the first seven elements of f, and f_8 and f_9 are the eighth and ninth elements of f. In order to obtain the values of f_8 and f_9, $\mathbf{F}$ is computed by using the seven points algorithm. Then, f is computed by selecting from any choice of pairs of $\mathbf{F}$, the one which minimizes $\|f\| = 1$. Although the *analytic*

method with rank-2 constraint obtains a rank-2 matrix, it does not greatly improve the results of the previously explained methods.

Concluding, the linear methods present an interesting reduced computing time but their accuracy is rather poor in the presence of false matching or if the points are only badly located due to image noise. In order to obtain better results, iterative algorithms have to be considered.

3.2 Iterative Methods

Iterative methods can be classified into two groups: those that minimize the distances between points and epipolar lines and those that are based on the gradient.

In the first classification, the iterative methods minimizing the distances between points and epipolar lines are based on solving the following equation

$$\min_{\mathbf{F}} \sum_i \left(d^2(m_i, \mathbf{F}m'_i) + d^2(m'_i, \mathbf{F}m_i) \right) \tag{15}$$

A first approach consists of directly applying an iterative method as *Newton-Raphson* or *Levenberg-Marquardt* in the equation (15) [32]. Another possibility consists of applying an *iterative linear method* [40], in which equation (15) has to be rewritten as

$$\min_{\mathbf{F}} \sum_i w_i^2 \left(m_i^{\mathrm{T}} \mathbf{F} m'_i \right)^2, \text{ where } w_i = \left(\frac{1}{{l_1}^2 + {l_2}^2} + \frac{1}{{l'_1}^2 + {l'_2}^2} \right)^{1/2} \tag{16}$$

The iterative linear method is based on computing the weight value w_i equivalent to the epipolar distances by using the previous $\mathbf{F}$ (in the first iteration $w_i = 1$) and then minimize by using *least-squares* in each iteration. Neither approach imposes the rank-2 constraint. However, the *nonlinear minimization in parameter space* [40] can solve this situation. This method is based on parameterizing the fundamental matrix, keeping in mind that it has a rank-2 in the following way,

$$\mathbf{F} = \begin{pmatrix} a & b & -ax_e - by_e \\ c & d & -cx_e - dy_e \\ -ax_{e'} - cy_{e'} & -bx_{e'} - dy_{e'} & (ax_e + by_e)x_{e'} + (cx_e + dy_e)y_{e'} \end{pmatrix} \tag{17}$$

in whitch (x_e, y_e) and $(x_{e'}, y_{e'})$ are the coordinates of the epipole in the first image plane and second image plane, respectively. Equation (17) is just one of the multiple parameterizations of $\mathbf{F}$ which must be computed. Finally, the estimated $\mathbf{F}$ becomes the parameterization which maximizes the following equation,

$$(ad - bc)^2 \sqrt{{x_e}^2 + {y_e}^2 + 1} \sqrt{{x_{e'}}^2 + {y_{e'}}^2 + 1} \tag{18}$$

The iteration of this method allows computing better rank-2 $\mathbf{F}$.

Besides, the minimization of equation (12) is not accurate enough to obtain a good estimation because the variance of points is not analogous and the least-square technique assumes they are comparable. In order to overcome this drawback, the second group of methods has to be considered.

The second group of methods is based on the *gradient-based* [14]. In this case, the equation to solve is

$$\min_{\mathbf{F}} \sum_i \left(m_i^{\mathrm{T}} \mathbf{F} m_i'\right)^2 / g_i^2 \tag{19}$$

where $g_i = \sqrt{{l_1}^2 + {l_2}^2 + {l_1'}^2 + {l_2'}^2}$.

The *gradient-based* technique obtains better results when compared with linear methods and iterative methods which minimize the distance between points and epipolar lines. Although iterative methods are more accurate than linear methods, they compute intensively and can not cope with potential outliers.

3.3 Robust Methods

In this section we present three robust methods: *M-Estimators*, *Least-Median-Squares* (LMedS) and *Random Sampling* (RANSAC), which can be used both in the presence of outliers and in bad point localization.

M-estimators [14] reduces the effect of outliers weighting the residual of each point. Consider r_i the residual of $m_i^{\mathrm{T}} \mathbf{F} m_i'$. Then, *M-estimators* are based on solving the following expression

$$\min_{\mathbf{F}} \sum_i w_i \left(m_i^{\mathrm{T}} \mathbf{F} m_i'\right)^2 \tag{20}$$

in which w_i is a weight function. A lot of different weight functions have been proposed so a new M-estimator is obtained for each one. A common weight function proposed by Huber [20] is the following

$$w_i = \begin{cases} 1 & |r_i| \leq \sigma \\ \sigma/|r_i| & \sigma < |r_i| \leq 3\sigma \\ 0 & 3\sigma < |r_i| \end{cases} \tag{21}$$

Another interesting weight function is proposed by Tukey [30],

$$w_i = \begin{cases} \left(1 - \left(\frac{r_i}{4.6851}\right)^2\right)^2 & |r_i| \leq 4.6851\sigma \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

In order to obtain σ, the robust standard deviation can be used (see [40]).

$$\sigma = 1.4826 \left(1 + 5/(n-7)\right) \operatorname{median}_i |r_i| \tag{23}$$

There are a lot of weight functions and for each one we obtained different results. The results given by this method are quite good in the presence of gaussian noise in image point localization, but they are rather limited in outlier detection.

LMedS [40] and *RANSAC* [34] techniques are quite similar. Both techniques are based on randomly selecting the set of points used to compute an approximation of $\mathbf{F}$ by using a linear method. The difference between both techniques is in the way they determine the chosen $\mathbf{F}$. LMedS calculates for each $\mathbf{F}$ the median distance between the points and epipolar lines, in which the chosen fundamental matrix has to minimize this median. RANSAC calculates for each $\mathbf{F}$ the number of inliers, in which the chosen $\mathbf{F}$ is the one that maximizes it. Once the outliers are removed, $\mathbf{F}$ is recalculated with the aim of obtaining a better approach.

Another difference between both methods is that LMedS is more restrictive than RANSAC, in that LMedS removes more points than RANSAC. However, the principal constraints of both techniques is their lack of repetitivity due to the aleatory way of selecting the points. Although experimental results show that LMedS gives better results in terms of accuracy, it does not always model the epipolar geometry properly.

3.4 Considerations in F Estimation

3.4.1 Normalizing data

Data normalization is a key point in fundamental matrix estimation. It has been proved that the computation should not be applied directly to raw data in pixels due to potential uncertainties given by huge numbers. The process of normalization consists of scaling and translating the data so that points m_i and m'_i are transformed to ($\hat{m}_i = \mathbf{T}m_i$ and $\hat{m}'_i = \mathbf{T}'m'_i$) by using two transformation matrices $\mathbf{T}$ and $\mathbf{T}'$ respectively. Then, the $\hat{\mathbf{F}}$ matrix is estimated from the normalized points and, finally, it has to be restored to obtain $\mathbf{F}$ using the following equation

$$\mathbf{F} = \mathbf{T}^{\mathrm{T}}\hat{\mathbf{F}}\mathbf{T}' \tag{24}$$

Basically there are two different methods of data normalization. The first method [40] normalizes the data between [-1,1]. The second was proposed by Hartley [12] and is based on two transformations. First, the points are translated so that their centroid is placed at the origin. Then, the points are scaled so that the mean of the distances of the points to the origin is $\sqrt{2}$. It has been proved that Hartley's method gives more accurate results than normalizing between [-1,1].

3.4.2 Rank-2 constraint

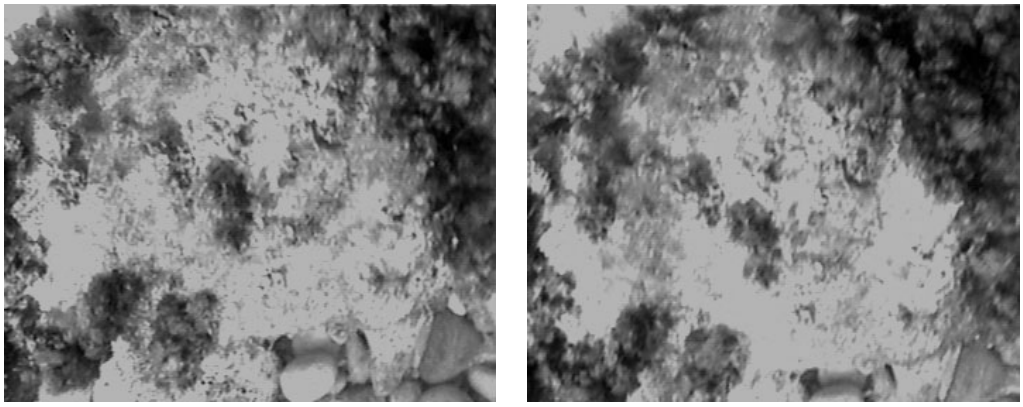
In most circumstances, the estimated $\mathbf{F}$ should be a rank-2 matrix in order to model the epipolar geometry with all the epipolar lines intersecting in a unique epipole. Although the rank-2 constraint is not imposed in most of the surveyed methods, there is a mathematical method which transforms a rank-n square matrix to the closest rank-($n-1$) matrix [14]. The $\mathbf{F}$ is decomposed in

$$\mathbf{F} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}} \tag{25}$$

by using singular value decomposition, where $\hat{\mathbf{S}} = \mathrm{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \sqrt{\lambda_3})$. The component with the smallest weight is removed obtaining
$\hat{\mathbf{S}} = \mathrm{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, 0)$. Then, $\mathbf{F}$ is recalculated in the following way:

$$\hat{\mathbf{F}} = \mathbf{U}\hat{\mathbf{S}}\mathbf{V}^{\mathrm{T}} \tag{26}$$

However, transforming the obtained $\mathbf{F}$ to a rank-2 matrix will give worse results because it has not been optimized. Then, we suggest using any method which imposes a rank-2 matrix in the computation of $\mathbf{F}$.



(a) (b)

Figure 3: Two views of an underwater scene: (a) left image; (b) right image.

(a) (b)

Figure 4: Urban scene courtesy of Zhengyou Zhang[2]: (a) left image; (b) right image.

4 Experimental Results

The surveyed methods have been programmed and their accuracy analyzed with synthetic images varying the gaussian noise and the number of outliers. Moreover, these methods have been tested in real images such as underwater images from the seabed (see Figure 3) grabbed by our underwater robot GARBI [1] and images obtained from public databases (see Figure 4). The corresponding points have been normalized by using the method proposed by Hartley [12] explained in the last section.

Table 1 shows the accuracy of each method computed as the mean and standard deviation of the distances between points and epipolar lines.

The *seven points* algorithm obtains a solution using only seven points. However, the accuracy depends greatly on the points used. The *least-squares technique* is based on using at least 8 points and its accuracy depends on the amount of badly located points used, usually obtaining better results by increasing the amount of points. The *eigen analysis* is the linear method that obtains the best results because an orthogonal least-squares minimization is more realistic than the classic one. However, all these methods obtain a rank-3 fundamental matrix, which means that the epipolar geometry is not properly modeled.

The *analytic method with rank-2 constraint* obtains a rank-2 fundamental matrix in which distances between

[2]Available at `http://www-sop.inria.fr/robotvis/demo/f-http/html`

Table 1: Methods Implemented with mean and std. of error: 1.- seven points; 2.- least-squares (LS) 3.- orthogonal LS; 4.- rank-2 constraint; 5.- iterative lineal using LS; 6.- iterative Newton-Raphson using LS; 7.- minimization in parameter space using eigen; 8.- gradient using LS; 9.- gradient using eigen; 10.- M-Estimator using LS; 11.- M-Estimator using eigen; 12.- M-Estimator proposed by Torr; 13.- LMedS using LS; 14.- LMedS using eigen; 15.- RANSAC using eigen.

Methods	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
$\sigma = 0.0$	0.000	0.000	0.000	0.102	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
outliers 0%	0.000	0.000	0.000	0.043	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$\sigma = 0.0$	22.125	339.562	17.124	30.027	161.684	27.035	17.871	187.474	18.224	73.403	4.909	4.714	0.000	0.000	9.449
outliers 10%	57.007	433.013	31.204	59.471	117.494	59.117	31.225	197.049	36.141	60.443	4.493	2.994	0.000	0.000	8.387
$\sigma = 0.1$	15.048	1.331	0.107	0.120	1.328	0.108	0.112	1.328	0.112	0.355	0.062	0.062	1.331	0.107	0.107
outliers 0%	14.498	0.788	0.088	0.091	0.786	0.088	0.092	0.786	0.092	0.257	0.042	0.041	0.788	0.088	0.088
$\sigma = 0.1$	26.136	476.841	19.675	50.053	158.671	70.530	19.549	183.961	15.807	73.354	4.876	4.130	0.449	0.098	9.148
outliers 10%	66.095	762.756	46.505	53.974	124.086	91.194	46.537	137.294	40.301	59.072	4.808	2.997	0.271	0.077	8.564
$\sigma = 0.5$	15.783	5.548	0.538	0.642	5.599	0.538	0.554	5.590	0.554	2.062	0.392	0.367	5.548	0.538	0.538
outliers 0%	14.837	3.386	0.362	0.528	3.416	0.366	0.361	3.410	0.361	1.466	0.237	0.207	3.386	0.362	0.362
$\sigma = 0.5$	117.534	507.653	19.262	26.475	161.210	47.884	18.933	217.577	19.409	143.442	3.887	3.147	47.418	0.586	10.723
outliers 10%	94.987	940.808	49.243	54.067	136.828	65.975	49.204	368.061	51.154	111.694	3.969	2.883	29.912	0.434	12.972
$\sigma = 1.0$	19.885	21.275	1.065	1.319	20.757	1.064	1.071	21.234	1.071	8.538	0.794	0.814	21.275	1.065	1.065
outliers 0%	16.485	12.747	0.744	0.912	12.467	0.747	0.745	12.719	0.745	6.306	0.463	0.463	12.747	0.744	0.744
$\sigma = 1.0$	138.554	629.326	21.264	61.206	158.849	79.323	20.277	152.906	18.730	120.012	3.921	4.089	25.759	1.052	8.657
outliers 10%	96.671	833.019	53.481	64.583	120.461	80.100	49.476	120.827	38.644	122.436	3.752	4.326	15.217	0.803	17.410
Underwater	3.833	4.683	1.725	5.242	3.068	2.584	1.643	2.949	1.581	0.557	0.650	0.475	1.485	1.039	1.725
Scene	4.440	3.941	2.138	4.286	2.804	4.768	2.109	2.798	2.056	0.441	0.629	0.368	1.134	0.821	2.138
Urban	4.9146	1.7238	0.4396	1.0227	0.4676	1.1020	2.9740	1.1088	0.4463	1.6678	0.3089	0.2790	0.9037	0.3193	0.4396
Scene	4.6250	1.1592	0.3343	1.0124	0.3414	0.7963	3.0658	0.8026	0.3675	0.9349	0.2280	0.1885	0.6171	0.2690	0.3343

points and epipolar lines are worse than in the linear methods. The *iterative linear method* improves considerably the least-squares technique but can not cope with outliers. The *iterative Newton-Raphson algorithm* gets even better results than the previous method if the presence of outliers is not considered. Although the *nonlinear minimization in parameter space* also obtains a rank-2 matrix, the distances of points to epipolar lines are the worst. The eighth and ninth methods are two diferent versions of the *gradient-based method* using least-squares and orthogonal least-squares, respectively. Both methods obtain better results than their equivalent linear methods. Nevertheless, the eigen analysis once more obtains better results than the other linear methods. Some of these methods obtain a rank-2 matrix but can not cope with outliers.

The last surveyed methods are classified into robust, which means that they might detect and remove potential outliers and compute the fundamental matrix by using only inliers. Three versions of the *M-estimators* based on the Huber weight function have been programmed: least-squares, eigen analysis and the method proposed by Torr [34]. The three methods start from a linear initial guess and become fully dependent on the linear method used to estimate it. Moreover, least-squares and eigen values get a rank-3 matrix, while Torr forces a rank-2 matrix in each iteration giving a more accurate geometry. Besides, two different versions of *LMedS* using again least-squares and eigen analysis have been studied. Although the accuracy of LMedS seems worse compared to M-estimators, LMedS removes the outliers more efficiently so that the epipolar geometry is properly obtained. *RANSAC* is the last surveyed method. However, RANSAC does not obtains any better results than LMedS with eigen analysis due to the method used to select the outliers which is quite permissive.

The last two rows of Table 1 show the results obtained from real images, i.e. an underwater scene grabbed

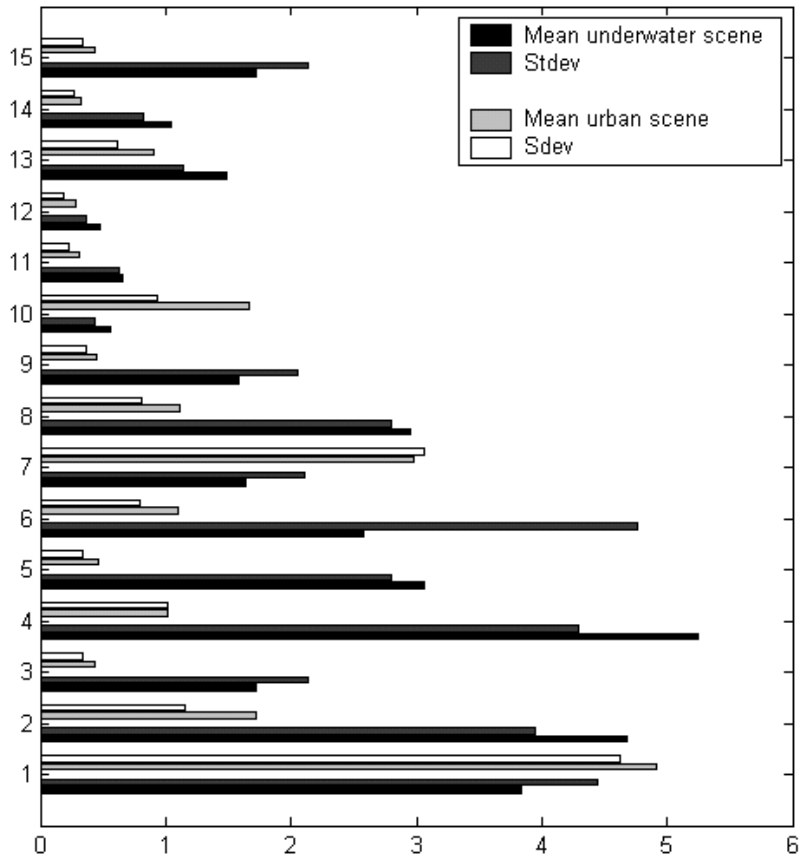


Figure 5: Experimental results obtained in the underwater scene and the urban scene[3].

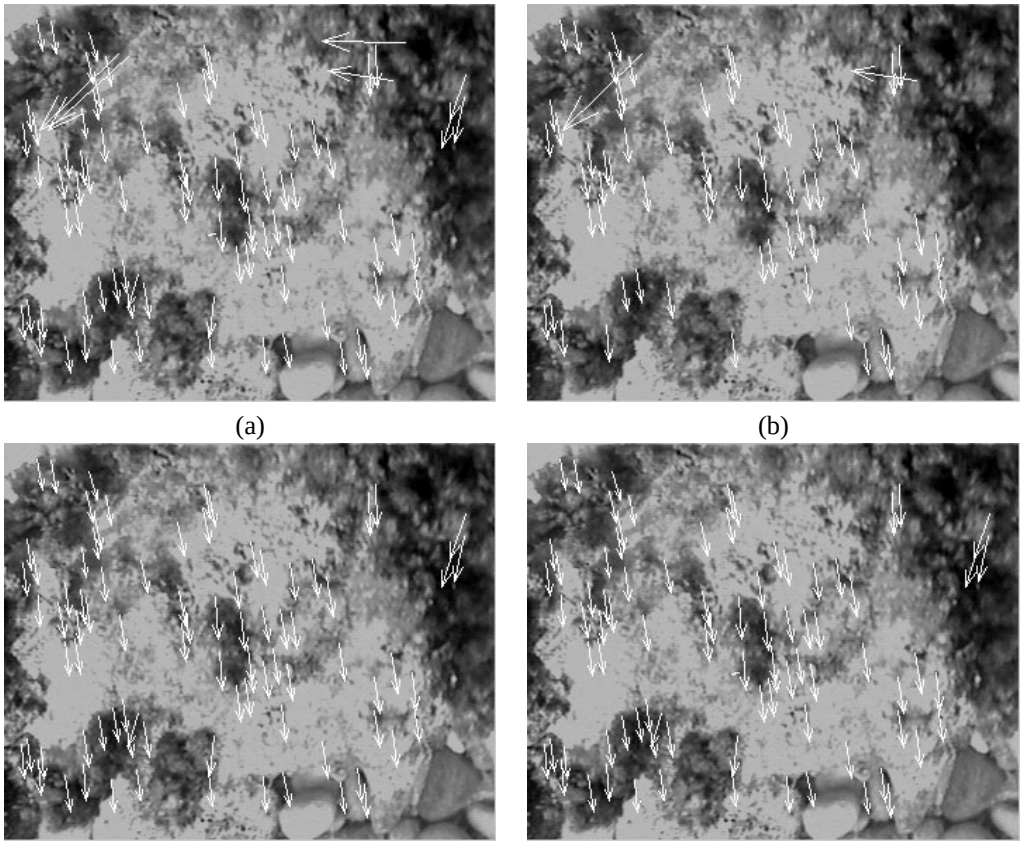
from our underwater robot and a well-known urban view taken from a public database. Figure 5 shows the results obtained by using every programmed method in the underwater and urban case. Note that accuracy is slightly worse in the underwater case because of image blurring due to water scattering which increases the noise in point localization and the number of outliers. Results take out the weakness of every method in such conditions.

Figures 6a and 7a show the matchings obtained by using the method proposed by Zhengyou Zhang [39, 42]. First, a Harris corner detector is applied to get a list of interesting points. Then the matching between both images is computed by using a pixel-based correlation. Note that matches might not be unique. Finally, a relaxation method is used to improve the local consistency of matches, reducing their ambiguity.

Figures 6b and 7b show the list of matchings kept by M-estimator based on eigen values. Depending on the weighting function, the removed matchings vary due to both noise and outliers. Note that some good matchings are also removed while potential outliers are kept as inliers. Figures 6c and 7c show the results obtained by LMedS, while Figures 6d and 7d show the results obtained by RANSAC. In both cases, every single outlier is detected and removed, obtaining comparatively the same results.

Also, the geometry modeled by every robust method is quite different. Figures 8a and 8b show the epipolar

[3]Methods: 1.- seven points; 2.- least-squares (LS) 3.- orthogonal LS; 4.- rank-2 constraint; 5.- iterative lineal; 6.- iterative Newton-Raphson; 7.- minimization in parameter space; 8.- gradient using LS; 9.- gradient using eigen; 10.- M-Estimator using LS; 11.- M-Estimator using eigen; 12.- M-Estimator proposed by Torr; 13.- LMedS using LS; 14.- LMedS using eigen; 15.- RANSAC



(a) (b)

(c) (d)

Figure 6: Underwater scene and matchings: (a) set of initial correspondences; and the matchings kept by: (b) M-Estimators; (c) LMedS; (d) RANSAC.

geometry given by M-Estimator based on eigen values, wherein it is shown how the epipolar lines do not cross in a single epipole due to the rank-3 matrix obtained. LMedS obtains a completely different geometry in which epipoles have been located outside the image plane, but they are unique (see Figure 8c and Figure 8d). RANSAC obtains a geometry with the epipole located near the image centre. Comparing the obtained geometries related to the position of the camera and its motion, the geometry modelled by RANSAC is the closest to reality.

The same study has been done considering the urban scene showing that the obtained results are a bit different. The reader can see these results in Figure 9, Figure 10 and Figure 11. The number of potential outliers is fewer than in the underwater scene and the location of image points is more accurate because of better image quality (see Figure 9a and Figure 10a). Figure 9b and Figure 10b show the poor results obtained by the eigen value M-Estimator, in which a lot of matchings are removed while some of the outliers are kept. In this case, LMedS is the only method which detects the set of outliers located in the right side of the image (see Figure 9c and Figure 10c). Besides, RANSAC does not detect any outlier so results are not accurate enough.

The geometry obtained in the urban scene largely depends on the method utilized. Figure 11 shows the three different geometries given by M-Estimator, LMedS and RANSAC. In this case, M-Estimator and RANSAC model a similar geometry in which the epipoles are located outside the image near the top-right corner, which is not the

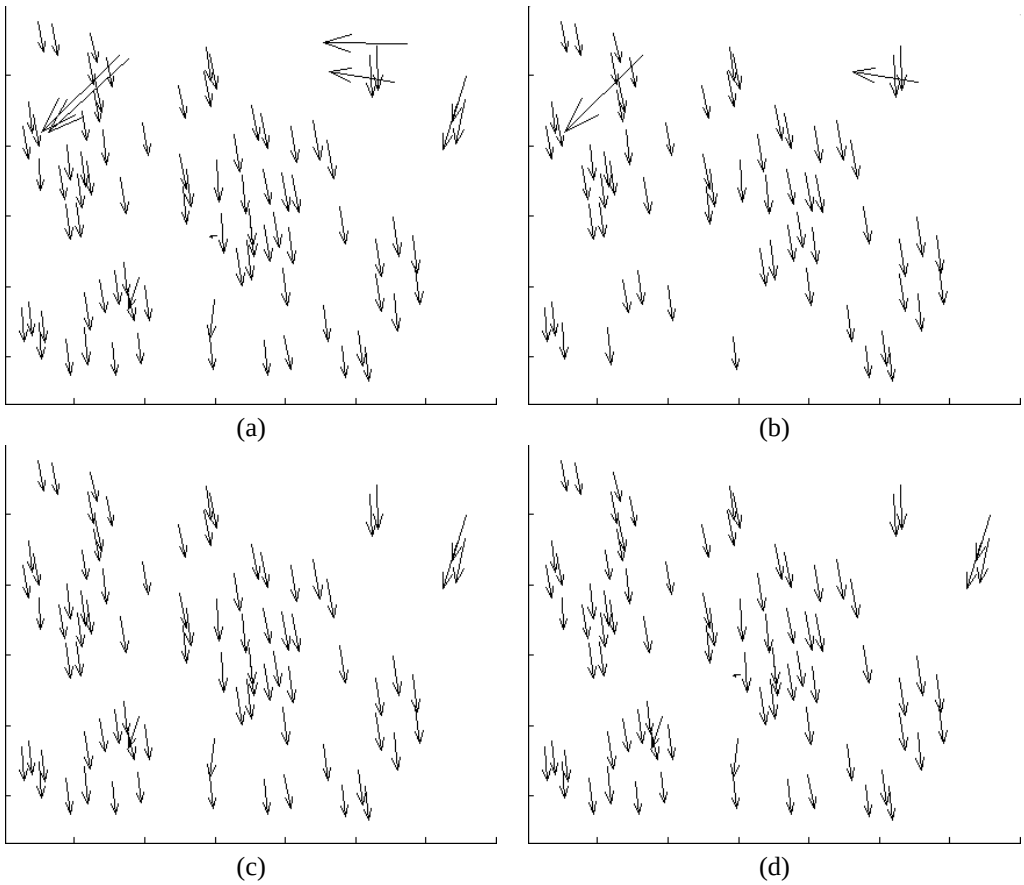


Figure 7: Matchings in the underwater scene: (a) set of initial correspondences; and the matchings kept by: (b) M-Estimators; (c) LMedS; (d) RANSAC.

right situation. LMedS obtains the right geometry with the epipoles located in the left side of the image.

5 Conclusions

This article surveys up to fifteen of the most used methods in fundamental matrix estimation. The different methods have been programmed and their accuracy analyzed with real images. Experimental results show that: a) linear methods are quite good if the points are well located in the image and the corresponding problem previously solved; b) iterative methods can cope with some gaussian noise in the localization of points, but they become really inefficient in the presence of outliers; and c) robust methods can cope with both discrepancy in the localization of points and false matchings.

Experimental results show that the orthogonal least-squares using eigen analysis gives better results than the classic least-squares technique of minimization. Moreover, a rank-2 method is preferred because it models the epipolar geometry with all the epipolar lines intersecting at the epipole. Moreover, experimental results show that the corresponding points have to be normalized and the best results have been obtained by using the method proposed by Hartley [14].

Summarizing, the best results are either obtained by using LMedS or RANSAC robust estimators, but their

accuracy depends on noise, the number of outliers and camera position and motion. The uncertainty in fundamental matrix computation was studied in detail by Csurka et al. [5] and Torr and Zisserman [35]. The surveyed methods model the epipolar geometry without considering lens distortion which considerably influences their discrepancy. Thus, some efforts have been made recently in presence of radial lens distortion [41]. In all, LMedS is the most appropriate for outlier detection and removal. However, with the aim of obtaining an accurate geometry, it is better to combine it with M-Estimator, which in our case has modelled a proper geometry in synthetic data, either in the presence of noise or outliers.

References

[1] J. Amat, J. Batlle, A. Casals, and J. Forest, *GARBI: A low cost ROV, constrains and solutions.*, in 6ème Seminaire IARP En Robotique Sous-Marine, Toulon-La Seyne, France, March 27-29, 1996, International Advanced Robotics Program, pp. 1–22.

[2] J. Batlle, E. Mouaddib, and J. Salvi, *A survey: Recent progress in coded structured light as a technique to solve the correspondence problem*, International Journal of Pattern Recognition, 31 (1998), pp. 963–982.

[3] M. Bober, N. Greorgis, and J. Kittler, *On accurate and robust estimation of fundamental matrix*, Computer Vision and Image Understanding, 72 (1998), pp. 39–53.

[4] M. J. Brooks, L. Agapito, D. Q. Huynh, and L. Baumela, *Direct methods for self-calibration of a moving stereo head*, in 4th Europena Conference on Computer Vision, vol. 2, Cambridge, UK, April 1996, pp. 415–426.

[5] G. Csurka, C. Zeller, Z. Zhang, and F. O. D., *Characterizing the uncertainty of the fundamental matrix*, Computer vision and Image Understanding, 68 (1997), pp. 18–36.

[6] R. Deriche, Z. Zhang, L. Q.-T., and F. O., *Robust recovery of the epipolar geometry for an uncalibrated stereo ring*, European Conference on Computer Vision, 800 (1994), pp. 567–576.

[7] O. D. Faugeras, *What can be seen in three dimensions with an uncalibrated stereo ring?*, European Conference on Computer Vision, 588 (1992), pp. 563–578.

[8] ——, *Three-Dimensional Computer Vision*, The MIT Press, Cambridge, Massachusetts, 1993.

[9] O. D. Faugeras, Q. T. Luong, and S. J. Maybank, *Camera self-calibration: Theory and experiments*, European Conference on Computer Vision, (1992), pp. 321–334.

[10] O. D. Faugeras and G. Toscani, *The calibration problem for stereo*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Los Alamitos, CA, June 1986, pp. 15–20.

[11] E. L. Hall, J. B. K. Tio, C. A. McPherson, and F. A. Sadjadi, *Measuring curved surfaces for robot vision*, Computer Journal, December (1982), pp. 42–54.

[12] R. Hartley, *In defence of the 8-point algorithm*, in Proceedings of the 5th International Conference on Computer Vision, Boston, 1995, IEEE Computer Society Press, pp. 1064–1070.

[13] R. Hartley, R. Gupta, and T. Chang, *Stereo from uncalibrated cameras*, Computer Vision and Pattern Recognition, (1992), pp. 761–764.

[14] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[15] R. I. Hartley, *Estimation of relative camera position for uncalibrated cameras*, European Conference on Computer Vision, (1992), pp. 579–587.

[16] ——, *Euclidean reconstruction from uncalibrated views*, Second European Workshop on Applications of Invariance in Computer Vision, (1993), pp. 237–257.

[17] ——, *Self-calibration from multiple views with a rotating camera*, European Conference on Computer Vision, 800 (1994), pp. 471–478.

[18] ——, *Kruppa's equations derived from the fundamental matrix.*, Pattern Analysis and Machine Intelligence, 19 (1997), pp. 133–135.

[19] T. S. Huang and O. D. Faugeras, *Some propieties of the E matrix in two-view motion estimation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 11 (1989), pp. 1310–1312.

[20] P. Huber, *Robust Statistics*, John Wiley and Sons, 1981.

[21] M. Ito, *Robot vision modelling - camera modelling and camera calibration*, Advanced Robotics, 5 (1991), pp. 321–335.

[22] J.-H. Jang and H. Ki-Sang, *Self-calibration of a stereo-camera by pure translational motion*, in IEEE International Conference on Image Processing, vol. 2, Laussane, Switzerland, 16-19 September 1996, pp. 297–300.

[23] F. Li, M. Brady, and C. Wiles, *Fast computation ot the fundamental matrix for active stereo vision system*, in 4th European Conference on Computer Vision, vol. 1, Cambridge, UK, April 1996, pp. 157–166.

[24] M. Li, *Camera calibration of a head-eye system for active vision*, European Conference on Computer Vision, 800 (1994), pp. 543–554.

[25] H. C. Longuet-Higgins, *A computer algorithm for reconstructing a scene from two projections*, Nature, 293 (1981), pp. 133–135.

[26] Q.-T. Luong and O. D. Faugeras, *A stability analysis of the fundamental matrix*, European Conference on Computer Vision, 800 (1994), pp. 577–588.

[27] ——, *The fundamental matrix: Theory, algorithms, and stability analysis*, International Journal of Computer Vision, 17 (1996), pp. 43–75.

[28] Q.-T. Luong and T. Viéville, *Canonic representation for geometries of multiple projective views*, European Conference on Computer Vision, 800 (1994), pp. 589–599.

[29] R. Mohr and E. Arbogast, *It can be done without camera calibration*, International Journal of Pattern Recognition Letters, 12 (1991), pp. 39–43.

[30] F. Mosteller and J. Turkey, *Data and Analysis and Regression*, Addison-Wesley, 1977.

[31] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*, John Wiley and Sons, New York, 1987.

[32] J. Salvi, *An Approach to Coded Structured Light to Obtain Three Dimensional Information*, PhD thesis, Universitat de Girona, Departament d'Electrònica, Informàtica i Automàtica, 1997.

[33] C. V. Stewart, *MINPRAN: A new robust estimator from computer vision*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 17 (1995), pp. 925–938.

[34] P. H. S. Torr and D. W. Murray, *The developement and comparision of robust methods for estimating the fundamental matrix*, International Journal Computer Vision, 24 (1997), pp. 271–300.

[35] P. H. S. Torr and A. Zisserman, *Robust detection of degenerate configurations while estimating the fundamental matrix*, Computer Vision and Image Understanding, 71 (1998), pp. 312–333.

[36] ——, *MLESAC: A new robust estimator with application to estimating image geometry*, Computer Vision and Image Understanding, 78 (2000), pp. 138–156.

[37] R. Y. Tsai, *A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses*, IEEE International Journal on Robotics and Automation, RA-3 (1987), pp. 323–344.

[38] J. Weng, P. Cohen, and M. Herniou, *Camera calibration with distortion models and accuracy evaluation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 14 (1992), pp. 965–980.

[39] Z. Zhang, *The matching problem: The state of the art*, Tech. Rep. 2146, Institut National de Recherche en Informatique et en Automatique, December 1993.

[40] ——, *Determining the epipolar geometry and its uncertainty: A review*, Tech. Rep. 2927, Institut National de Recherche en Informatique et en Automatique, July 1996.

[41] ——, *On the epipolar geometry between two images with lens distortion*, in International Conference on Pattern Recognition, vol. 1, 1996, pp. 407–411.

[42] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, *A robust technique for matching two uncalibred images through the recovery of the unknown epipolar geometry*, Tech. Rep. 2273, Institut National de Recherche en Informatique et Automatique, May 1994.

[43] Z. Zhang, Q.-T. Luong, and O. Faugeras, *Motion of an uncalibrated stereo ring: Self-calibration and metric reconstruction*, IEEE Transactions on Robotics and Automation, 12 (1996), pp. 103–113.

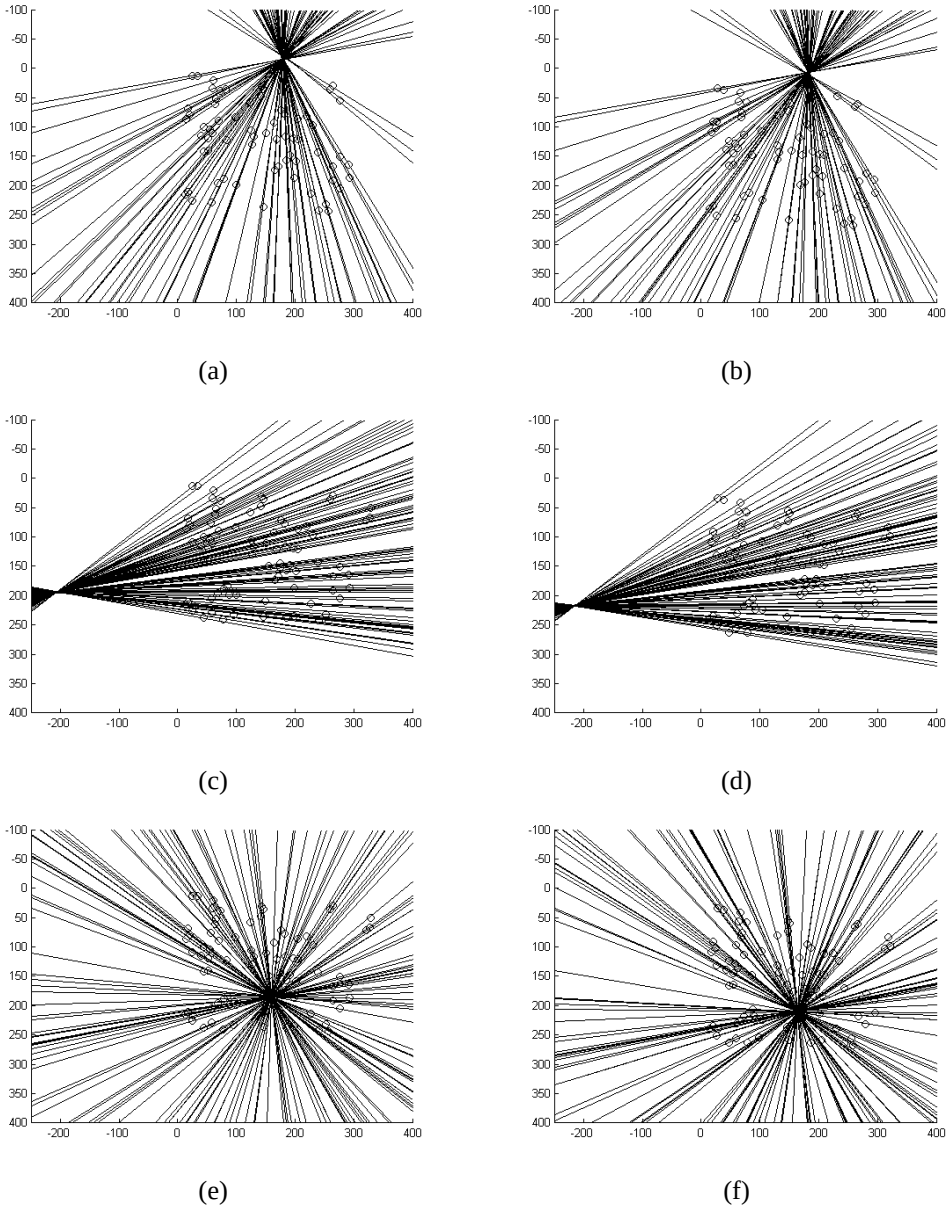
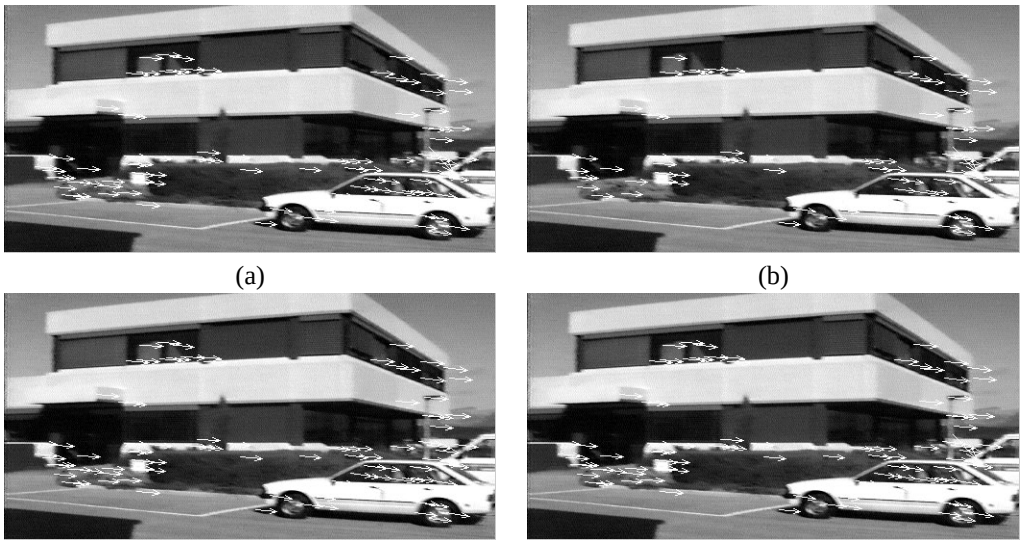


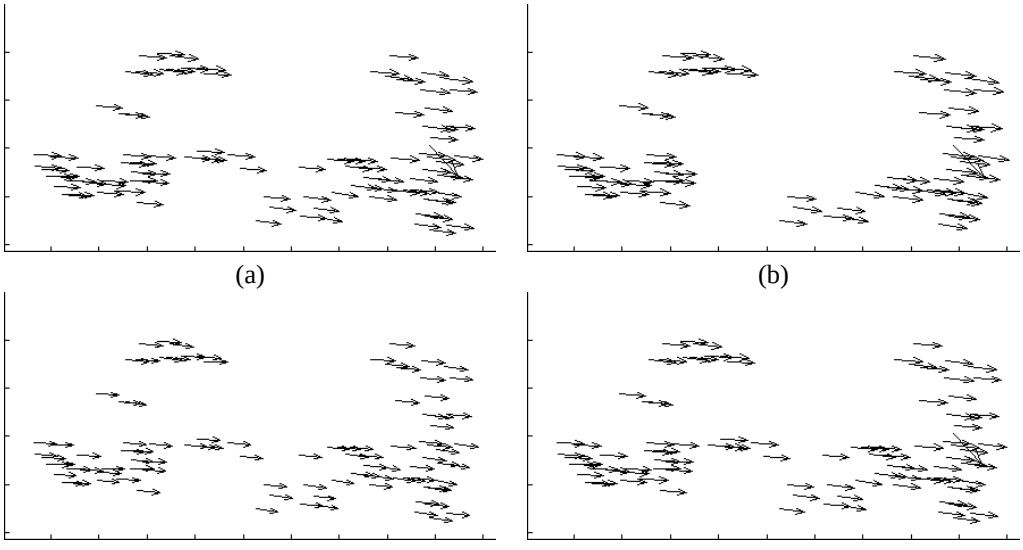
Figure 8: Points and epipolar lines in the underwater scene: (a) left and (b) right views obtained by M-Estimator; (c) left and (d) right views obtained by LMedS; (e) left and (f) right views obtained by RANSAC.



(a) (b)

(c) (d)

Figure 9: Urban scene and matchings: (a) set of initial correspondences; and the matchings kept by: (b) M-Estimators; (c) LMedS; (d) RANSAC.



(a) (b)

(c) (d)

Figure 10: Matchings in the urban scene: (a) set of initial correspondences; and the matchings kept by: (b) M-Estimators; (c) LMedS; (d) RANSAC.

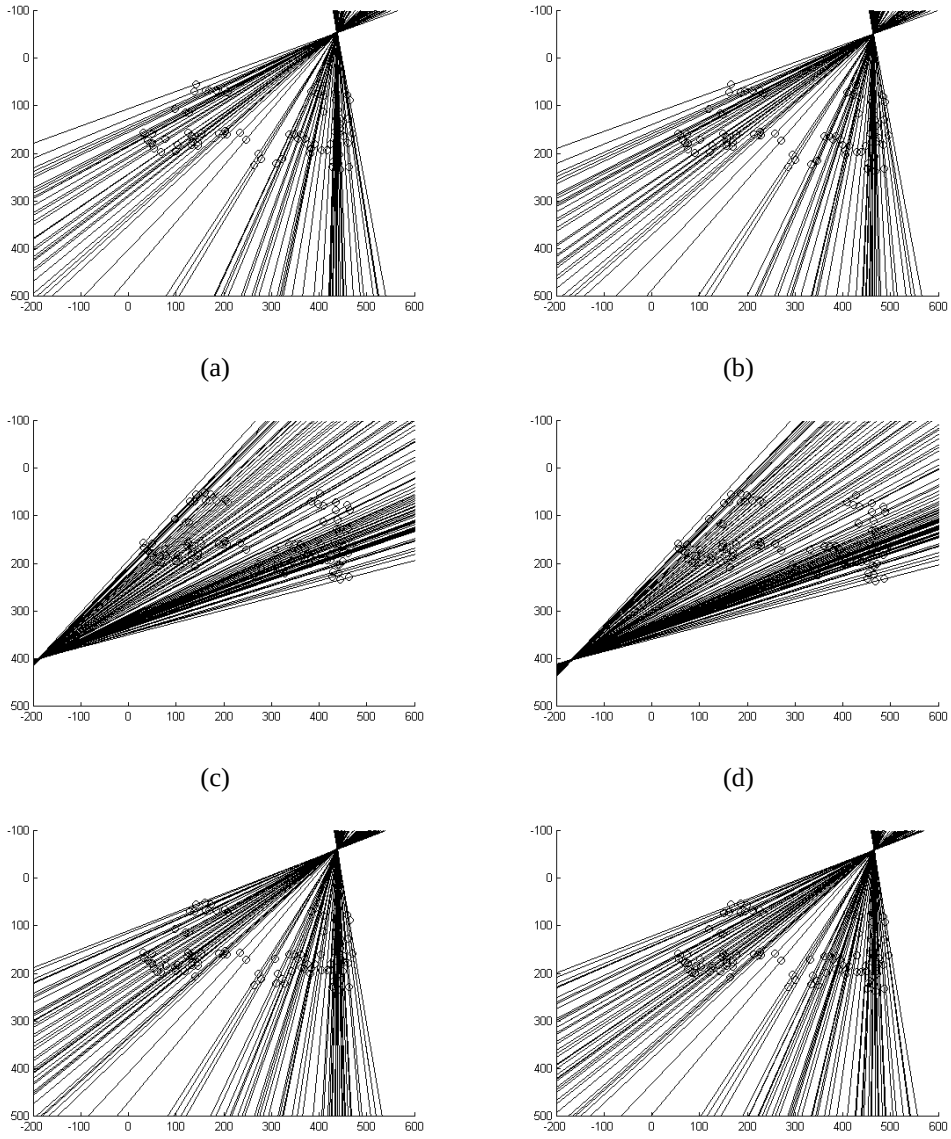


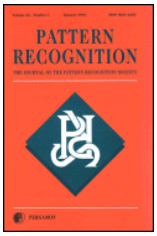
Figure 11: Points and epipolar lines in the urban scene: (a) left and (b) right views obtained by M-Estimator; (c) left and (d) right views obtained by LMedS; (e) left and (f) right views obtained by RANSAC.

A comparative review of camera calibrating methods with accuracy evaluation

Joaquim Salvi, Xavier Armangué and Joan Batlle

Computer Vision and Robotics Group
Institute of Informatics and Applications
University of Girona
Av. Lluís Santaló, s/n, E-17071 Girona (Spain)
{qsalvi,armangue,jbatlle}@eia.udg.es

Pattern Recognition Society
Pattern Recognition
Accepted to be published in 2001



Acceptance Letter

PATTERN RECOGNITION

Editor-in-Chief
Robert S. Ledley

Blaire V. Mossman, Manager Editor
P.O. Box 13177
Scottsdale, AZ 85267-3177
Phone (480) 991-4997 FAX (480) 991-4998

May 11, 2001

Joaquim Salvi
University of Girona
Computer Vision & Robotics Group
Av.Luis Santalo, s/n
E-17071 Girona
Spain

Dear Dr. Salvi:

We would be happy to publish your manuscript

A Comparative Review of Camera Calibrating Methods
with Accuracy Evaluation

in the journal provided that it is revised in accordance with the enclosed referee comments.

Kindly supply two copies of your modified paper and point out where all changes were made when resubmitting it for furthur consideration of publication in the journal.

Thank you for your interest in PATTERN RECOGNITION.

Sincerely yours,

Blaire V. Mossman

(Mrs.) Blaire V. Mossman
Managing Editor

P.S. Please comply with journal style requirements as marked in pink on the enclosure.

Dedicated to the Advancement of the Science of Pattern Recognition

• Theory • Application • Instrumentation •

A Comparative Review of Camera Calibrating Methods with Accuracy Evaluation

Joaquim Salvi*, Xavier Armangué and Joan Batlle

Computer Vision and Robotics Group. Institute of Informatics and Applications.
University of Girona. Av. Lluís Santaló, s/n, E-17071 Girona (Spain)
{qsalvi,armangue,jbatlle}@eia.udg.es

*Corresponding Author

Corresponding Author

Joaquim Salvi.
Computer Vision and Robotics.
Institute of Informatics and Applications.
University of Girona.
Avda. Lluís Santalo s/n.
E-17071 GIRONA (SPAIN)
Ph: +34 972 41 8483
FAX: +34 972 41 8098
e-mail: qsalvi@eia.udg.es

Biographical Sketch

JOAQUIM SALVI graduated in Computer Science in the Polytechnical University of Catalunya in 1993. He joined the Computer Vision and Robotics Group in the University of Girona, where he received the M.S. degree in Computer Science in July 1996 and the Ph.D in Industrial Engineering in January 1998. He received the best thesis award in Industrial Engineering of the University of Girona. At present, he is an associate professor in the Electronics, Computer Engineering and Automation Department of the University of Girona. His current interest are in the field of computer vision and mobile robotics, focusing on structured light, stereovision and camera calibration.

XAVIER ARMANGUÉ received the B.S. degree in Computer Science in the University of Girona in 1999 before joining the Computer Vision and Robotics Group. At preset he is engaged in the study of stereovision systems for mobile robotics and he is working for his Ph.D. in the Computer Vision and Robotics Group in the University of Girona and in the Institute of Systems and Robotics in the University of Coimbra.

JOAN BATLLE, graduated in Physics in the Autonomous University of Barcelona, received the Ph.D. in Computer Science in the Polytechnical University of Catalunya. At present, he is a professor in the Electronics, Computer Engineering and Automation Department; the leader of the Computer Vision and Robotics Group; and the director of the Institute of Informatics and Applications. His research activity is mainly focused on real-time vision and autonomous robots. He is actually involved in some governmental projects about underwater robots and technology transfer to industrial enterprises.

Summary

In this article, we present a comparative study of the most commonly used camera calibrating methods of the last few decades. These techniques cover a wide range of the classical hard calibration of image sensors which begin from a previous knowledge of a set of 3D points and their corresponding 2D projections on an image plane in order to estimate the camera parameters. Hence, this study is presented describing a total of 5 different camera calibrating techniques which include implicit vs. explicit calibration and linear vs. non-linear calibration.

A great deal of attention has been paid to use the same nomenclature and a standardized notation in the presentation of all the techniques. Actually, this is one of the greatest difficulties which appears when going into the details of any calibrating technique. This problem usually arises because each method defines a different set of coordinate systems and camera parameters. Therefore, all the techniques have been re-arranged so as to allow a comparative presentation. The reader is introduced to calibration with the implicit linear technique of the pseudo-inverse presented by Hall, afterwards the explicit linear calibration of Faugeras-Toscani is presented. Furthermore, the article describes an easy modification of the Faugeras method in order to include radial lens distortion, the well-known method of Tsai and Tsai optimized, and finally the complete method of Weng which models up to three different kinds of lens distortion.

In order to compare the accuracy provided by each technique surveyed, a brief description of accuracy evaluation is presented. Each calibrating technique has been implemented and its accuracy evaluated. The same set of test points has been used for all the techniques, which allows the results to be reliably compared. Hence, the reader can choose one or another method depending on the required accuracy. Moreover, once the calibrating method is chosen, the reader can take the equations directly from this article and easily use them in the desired calibrating algorithm.

There are numerous advantages thanks to an accurate calibration. For instance, dense reconstruction of 3D objects and surfaces has applications in visual inspection and medical imaging, such as quality control in industrial manufacturing and reconstruction of human backs and skulls for the detection of deformations or surgery. Another problem is the 3D pose estimation of an object in a scene, which has many applications such as obstacle avoidance, landmark detection and industrial part assembly, among others.

A Comparative Review of Camera Calibrating Methods with Accuracy Evaluation[1]

Abstract

Camera calibrating is a crucial problem for further metric scene measurement. Many techniques and some studies concerning calibration have been presented in the last few years. However, it is still difficult to go into details of a determined calibrating technique and compare its accuracy with respect to other methods. Principally, this problem emerges from the lack of a standardized notation and the existence of various methods of accuracy evaluation to choose from. This article presents a detailed review of some of the most used calibrating techniques in which the principal idea has been to present them all with the same notation. Furthermore, the techniques surveyed have been tested and their accuracy evaluated. Comparative results are shown and discussed in the article. Moreover, code and results are available in internet.

Keywords: Camera Calibration, lens distortion, parameter estimation, optimization, camera modelling, accuracy evaluation, 3D Reconstruction, Computer Vision.

1 Introduction

Camera calibration is the first step towards computational computer vision. Although some information concerning the measuring of scenes can be obtained by using uncalibrated cameras [1], calibration is essential when metric information is required. The use of precisely calibrated cameras makes the measurement of distances in a real world from their projections on the image plane possible [2, 3]. Some applications of this capability include:

1. *Dense reconstruction:* Each image point determines an optical ray passing through the focal point of the camera towards the scene. The use of more than a single view of a motionless scene (taken from a stereoscopic system, a single moving camera, or even a structured light emitter) permits crossing both optical rays to get the metric position of the 3D points [4, 5, 6]. Obviously, the correspondence problem has to be previously solved [7].

2. *Visual inspection:* Once a dense reconstruction of a measuring object is obtained, the reconstructed object can be compared with a stored model in order to detect any manufacturing imperfections such as bumps, dents or cracks. One potential application is visual inspection for quality control. Computerized visual in-

[1]This work has been supported by Spanish project CICYT TAP99-0443-CO5-01

spection allows automatic and exhaustive examination of products, as opposed to the slow human inspection which usually implies a statistical approach [8].

3. *Object localization:* When considering various image points from different objects, the relative position among these objects can be easily determined. This has many possible applications such as in industrial part assembly [9] and obstacle avoidance in robot navigation [10, 11], among others.

4. *Camera localization:* When a camera is placed in the hand of a robot arm or on a mobile robot, the position and orientation of the camera can be computed by locating some known landmarks in the scene. If these measurements are stored, a temporal analysis allows the handler to determine the trajectory of the robot. This information can be used in robot control and path planning [12, 13, 14].

Camera calibration is divided into two phases. First, camera modelling deals with the mathematical approximation of the physical and optical behavior of the sensor by using a set of parameters. The second phase of camera calibration deals with the use of direct or iterative methods to estimate the values of these parameters. There are two kinds of parameters in the model which have to be considered. On the one hand, the intrinsic parameter set, which models the internal geometry and optical characteristics of the image sensor. Basically, intrinsic parameters determine how light is projected through the lens onto the image plane of the sensor. The other set of parameters are the extrinsic ones. The extrinsic parameters measure the position and orientation of the camera with respect to a world coordinate system, which, in turn, provides metric information with respect to a user-fixed coordinate system instead of the camera coordinate system.

Camera calibration can be classified according to several different criteria. For instance, 1) Linear versus nonlinear camera calibration (usually differentiated depending on the modelling of lens distortion) [15]. 2) Intrinsic versus extrinsic camera calibration. Intrinsic calibration is concerned only with obtaining the physical and optical parameters of the camera [16, 17]. Besides, extrinsic calibration concerns the measurement of the position and orientation of the camera in the scene [18, 19]. 3) Implicit [20] versus explicit [21] calibration. Implicit calibration is the process of calibrating a camera without explicitly computing its physical parameters. Although, the results can be used for 3D measurement and the generation of image coordinates, they are useless for camera modelling as the obtained parameters do not correspond to the physical ones [22]. Finally, 4) the methods which use known 3D points as a calibrating pattern [23, 24] or even a reduced set of 3D points [25, 26], with respect to others which use geometrical properties in the scene such as vanishing lines [27] or other line features [28, 29].

These different approaches can also be classified regarding the calibration method used to estimate the parameters of the camera model:

1. *Non-linear optimization techniques.* A calibrating technique becomes non-linear when any kind of lens imperfection is included in the camera model. In that case, the camera parameters are usually obtained through iteration with the constraint of minimizing a determined function. The minimizing function is

usually the distance between the imaged points and the modelled projections obtained by iterating. The advantage of these iterating techniques is that almost any model can be calibrated and accuracy usually increases by increasing the number of iterations up to convergence. However, these techniques require a good initial guess in order to guarantee convergence. Some examples are described in classic photogrammetry [30] and Salvi [31].

2. *Linear techniques which compute the transformation matrix.* These techniques use the least squares method to obtain a transformation matrix which relates 3D points with their 2D projections. The advantage here is the simplicity of the model which consists in a simple and rapid calibration. One drawback is that linear techniques are useless for lens distortion modelling, entailing a rough accuracy of the system. Moreover, it is sometimes difficult to extract the parameters from the matrix due to the implicit calibration used. Some references related to linear calibration can be found in Hall [20], Toscani-Faugeras [23, 32] and Ito [15].

3. *Two-step techniques.* These techniques use a linear optimization to compute some of the parameters and, as a second step, the rest of the parameters are computed iteratively. These techniques permit a rapid calibration considerably reducing the number of iterations. Moreover, the convergence is nearly guaranteed due to the linear guess obtained in the first step. *Two-step techniques* make use of the advantages of the previously described methods. Some references are Tsai [24], Weng [33] and Wei [22].

This article is a detailed survey of some of the most frequently used calibrating techniques. The first technique was proposed by Hall in 1982 and is based on an implicit linear camera calibration by computing the 3x4 transformation matrix which relates 3D object points with their 2D image projections [20]. The latter work of Faugeras, proposed in 1986, was based on extracting the physical parameters of the camera from such a transformation technique, thus it is explained as the second technique [23, 32]. The following methods are based on non-linear explicit camera calibration, including the modelling of lens distortion. Hence, the first one is a simple adaptation of the Faugeras linear method with the aim of including radial lens distortion [31, 34]. The widely used method proposed by Tsai, which is based on a *two-step technique* modelling only radial lens distortion, is also detailed [24]. Finally, the complete model of Weng, which was proposed in 1992, including three different types of lens distortion, is explained as the last technique [33]. Note that one of the principal problems to understand a calibrating technique in detail is the lack of notation standardization in mathematical equations and the use of different sets of coordinate systems. Both limitations complicate the comparing of techniques, thus a great deal of effort has been made to present the survey using the same notation. All five techniques are explained herein and their 2D and 3D accuracy shown and discussed. A brief overview of camera accuracy evaluation [35] is included with the aim of using the same tools to compare different calibrating techniques implemented.

This article is structured as follows. Section two deals with camera modelling and how the camera model is gradually obtained by a sequence of geometrical transformations is explained. Section 3 describes the five different

techniques of camera calibration, which estimate the parameters of the camera model. Then, a few methods for accuracy evaluation of camera calibrating techniques are explained in section 4. Finally, both 2D and 3D accuracy of each calibration technique have been measured and their results are shown and compared. The paper ends with conclusions.

2 Camera Model

A model is a mathematical formulation which approximates the behavior of any physical device by using a set of mathematical equations. Camera modelling is based on approximating the internal geometry along with the position and orientation of the camera in the scene. There are several camera models to choose from depending on the desired accuracy [15]. The simplest are based on linear transformations without modelling the lens distortion. However, there are also some non-linear models which accurately model the lens. These are useful for some applications where greater precision is required.

The simplest model is the one proposed by Hall [20]. The goal is to find a linear relationship among the 3D points of the scene with their 2D projecting points on the image plane. This relationship is approximated by means of a transformation matrix[2], as shown in equation (1).

$$\begin{pmatrix} s^I X_d \\ s^I Y_d \\ s \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \end{pmatrix} \begin{pmatrix} {}^W X_w \\ {}^W Y_w \\ {}^W Z_w \\ 1 \end{pmatrix} \tag{1}$$

Then, given a 3D point P_w, expressed with respect to the metric world coordinate system (i.e. ${}^W P_w$), and applying the transformation matrix proposed by Hall, the 2D point P_d in pixels with respect to the image coordinate system is obtained, i.e. ${}^I P_d = ({}^I X_d, {}^I Y_d)$.

However, camera modelling is usually broken down into 4 steps, as is detailed in the following list (see also figure 1).

1. The first step consists of relating point ${}^W P_w$ from the world coordinate system to the camera coordinate system, obtaining ${}^C P_w$. This transformation is performed by using a rotation matrix and a translation vector.

2. Next, it is necessary to carry out the projection of point ${}^C P_w$ on the image plane obtaining point ${}^C P_u$, by using a projective transformation.

3. The third step models the lens distortion, based on a disparity with the real projection. Then, point ${}^C P_u$ is transformed to the real projection of ${}^C P_d$ (which should coincide with the points captured by the camera).

4. Finally, the last step consists of carrying out another coordinate system transformation in order to change from the metric coordinate system of the camera to the image coordinate system of the computer in pixels, obtaining ${}^I P_d$.

[2]The appendix at the end of the paper details the used nomenclature.

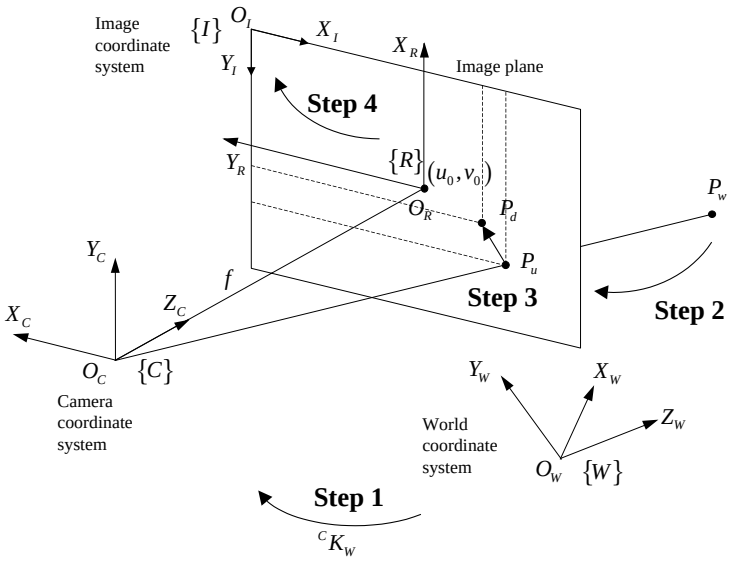


Figure 1: The geometric relation between a 3D object point and its 2D image projection.

In the following, the different camera models of Faugeras-Toscani [32], Faugeras-Toscani with distortion [34], Tsai [24] and Weng [33] are explained in detail with attention on how they carry out the above four steps.

2.1 Changing from the world to the camera coordinate system

Changing the world coordinate system to the camera coordinate system is carried out in the same way in all the surveyed models. This transformation is modelled using a translation vector and a rotation matrix, as shown in equation (2).

$$\begin{pmatrix} {}^{C}X_w \\ {}^{C}Y_w \\ {}^{C}Z_w \end{pmatrix} = {}^{C}R_W \begin{pmatrix} {}^{W}X_w \\ {}^{W}Y_w \\ {}^{W}Z_w \end{pmatrix} + {}^{C}T_W \tag{2}$$

Then, given a point ${}^{W}P_w$ related to the world coordinate system, and applying equation (2), the point ${}^{C}P_w$ in relation to the camera coordinate system is obtained. Note that ${}^{C}R_W$ expresses the orientation of the world coordinate system $\{W\}$ with respect to the axis of the camera coordinate system $\{C\}$, and that ${}^{C}T_W$ expresses the position of the origin of the world coordinate system measured with respect to $\{C\}$.

2.2 Projection of the 3D point on the image plane

Consider that any optical sensor can be modelled as a *pinhole camera* [2]. That is, the image plane is located at a distance f from the optical center O_C, and is parallel to the plane defined by the coordinate axis X_C and Y_C. Moreover, given an object point (${}^{C}P_w$) related to the camera coordinate system, if it is projected through the focal point (O_C), the optical ray intercepts the image plane at the 2D image point (${}^{C}P_u$). This relation is shown in equation (3).

$${}^{C}X_u = f\frac{{}^{C}X_w}{{}^{C}Z_w} \qquad {}^{C}Y_u = f\frac{{}^{C}Y_w}{{}^{C}Z_w} \tag{3}$$

All the various models reviewed solved the projective transformation by using the same equation (3).

2.3 Lens distortion

The third step is based on modelling the distortion of the lenses. However, each model surveyed required a different approach. Equations (4) transform the undistorted point ${}^{C}P_u$ to the distorted point ${}^{C}P_d$, where δ_x and δ_y represent the distortion involved.

$$ {}^{C}X_u = {}^{C}X_d + \delta_x \qquad {}^{C}Y_u = {}^{C}Y_d + \delta_y \tag{4} $$

The camera model proposed by Faugeras and Toscani [32] does not model the lens distortion, therefore, ${}^{C}P_u$ and ${}^{C}P_d$ are the same point. In this case δ_x and δ_y are zero, as shown in equation (5).

$$ \delta_x = 0 \qquad \delta_y = 0 \tag{5} $$

The Faugeras-Toscani model however can be improved by modelling the radial lens distortion [34]. Tsai [24] has modelled distortion in the same way. As shown in equations (6), δ_x and δ_y represent the radial distortion [30]. This type of distortion is mainly caused by flawed radial curvature of the lens. See also [33].

$$ \delta_x = \delta_{xr} \qquad \delta_y = \delta_{yr} \tag{6} $$

The displacement given by the radial distortion dr can be modelled by equations (7), which consider only k_1 the first term of the radial distortion series. It has been proven that the first term of this series is sufficient to model the radial distortion in most of the applications [24].

$$ \delta_{xr} = k_1 {}^{C}X_d \left({}^{C}X_d{}^2 + {}^{C}Y_d{}^2 \right) \qquad \delta_{yr} = k_1 {}^{C}Y_d \left({}^{C}X_d{}^2 + {}^{C}Y_d{}^2 \right) \tag{7} $$

The model of Weng [33] considers three types of distortion: radial distortion, decentering distortion and thin prism distortion. The total distortion will be the sum of these three distortions.

$$ \delta_x = \delta_{xr} + \delta_{xd} + \delta_{xp} \qquad \delta_y = \delta_{yr} + \delta_{yd} + \delta_{yp} \tag{8} $$

However, Weng proposed to model the lens distortion from the undistorted image point (${}^{C}X_u$,${}^{C}Y_u$) instead of the distorted one (${}^{C}X_d$,${}^{C}Y_d$). Although both approaches can be considered, it also has to be taken into account that the calibrating parameters will be different. Hence, equations (4) have to be substituted by equations (9).

$$ {}^{C}X_d = {}^{C}X_u + \delta_x \qquad {}^{C}Y_d = {}^{C}Y_u + \delta_y \tag{9} $$

The *radial distortion* is modelled in the same manner Tsai proposed, except that Weng used the undistorted points.

$$ \delta_{xr} = k_1 {}^{C}X_u \left({}^{C}X_u{}^2 + {}^{C}Y_u{}^2 \right) \qquad \delta_{yr} = k_1 {}^{C}Y_u \left({}^{C}X_u{}^2 + {}^{C}Y_u{}^2 \right) \tag{10} $$

The *decentering distortion* is due to the fact that the optical center of the lens is not correctly aligned with the center of the camera [33]. This type of distortion introduces a radial and tangential distortion [30], which can be described by the following equations,

$$\begin{aligned} \delta_{xd} &= p_1\left(3\,{}^{C}X_u{}^2 + {}^{C}Y_u{}^2\right) + 2p_2\,{}^{C}X_u\,{}^{C}Y_u \qquad & \delta_{yd} &= 2p_1\,{}^{C}X_u\,{}^{C}Y_u + p_2\left({}^{C}X_u{}^2 + 3\,{}^{C}Y_u{}^2\right) \end{aligned} \tag{11}$$

The *thin prism distortion* arises from imperfection in lens design and manufacturing as well as camera assembly. This type of distortion can be modelled by adding a thin prism to the optic system, causing radial and tangential distortions [33]. This distortion is modelled by,

$$\begin{aligned} \delta_{xp} &= s_1\left({}^{C}X_u{}^2 + {}^{C}Y_u{}^2\right) \qquad & \delta_{yp} &= s_2\left({}^{C}X_u{}^2 + {}^{C}Y_u{}^2\right) \end{aligned} \tag{12}$$

By adding the three equations (7), (11) and (12), and carrying out the following variable replacement: $g_1 = s_1 + p_1$, $g_2 = s_2 + p_2$, $g_3 = 2p_1$ and $g_4 = 2p_2$, equations (13) are obtained,

$$\begin{aligned} \delta_x &= (g_1 + g_3)\,{}^{C}X_u{}^2 + g_4\,{}^{C}X_u\,{}^{C}Y_u + g_1\,{}^{C}Y_u{}^2 + k_1\,{}^{C}X_u\left({}^{C}X_u{}^2 + {}^{C}Y_u{}^2\right) \\ \delta_y &= g_2\,{}^{C}X_u{}^2 + g_3\,{}^{C}X_u\,{}^{C}Y_u + (g_2 + g_4)\,{}^{C}Y_u{}^2 + k_1\,{}^{C}Y_u\left({}^{C}X_u{}^2 + {}^{C}Y_u{}^2\right) \end{aligned} \tag{13}$$

2.4 Changing from the camera image to the computer image coordinate system

This final step deals with expressing the ${}^{C}P_d$ point with respect to the computer image plane in pixels $\{I\}$. This change of coordinates can be made in two different ways according to the camera models surveyed.

The camera models proposed by Faugeras-Toscani, Faugeras-Toscani with distortion and by Weng use the following equations to carry out such a transformation:

$$\begin{aligned} {}^{I}X_d &= -k_u\,{}^{C}X_d + u_0 \qquad & {}^{I}Y_d &= -k_v\,{}^{C}Y_d + v_0 \end{aligned} \tag{14}$$

where: (k_u,k_v) are the parameters that transform from metric measures with respect to the camera coordinate system to pixels with respect to the computer image coordinate system; and (u_0, v_0) are the components that define the projection of the focal point in the plane image in pixels, i.e. the principal point. They are used to determine the translation between both coordinate systems.

The camera model of Tsai proposed other equations to carry out the same transformation. These equations are the following,

$$\begin{aligned} {}^{I}X_d &= -s_x d'_x{}^{-1}\,{}^{C}X_d + u_0 \qquad & {}^{I}Y_d &= -d_y{}^{-1}\,{}^{C}Y_d + v_0 \end{aligned} \tag{15}$$

where: (u_0, v_0) are the components of the principal point in pixels; s_x is the image scale factor; $d'_x = d_x \frac{N_{cx}}{N_{fx}}$; d_x is the center to center distance between adjacent sensor elements in the X direction; d_y is the center to center distance between adjacent sensor elements in the Y direction; N_{cx} is the number of sensor elements in the X direction; and N_{fx} is the number of pixels in an image row as sampled by the computer.

3 Calibrating Methods

The calibrating method depends on the model used to approximate the behavior of the camera. The linear models, i.e. Hall and Faugeras-Toscani, use a least-squares technique to obtain the parameters of the model. However, non-linear calibrating methods, as with Faugeras-Toscani with distortion, Tsai and Weng, use a two-stage technique. As a first stage, they carry out a linear approximation with the aim of obtaining an initial guess and then a further iterative algorithm is used to optimize the parameters. In this section, each calibrating method is explained detailing the equations and the algorithm used to calibrate the camera parameters.

3.1 The method of Hall

The method used to calibrate the model of Hall is based on expressing equation (1) in the following form,

$$\begin{aligned} {}^{I}X_u &= \frac{A_{11}{}^{W}X_w + A_{12}{}^{W}Y_w + A_{13}{}^{W}Z_w + A_{14}}{A_{31}{}^{W}X_w + A_{32}{}^{W}Y_w + A_{33}{}^{W}Z_w + A_{34}} \\ {}^{I}Y_u &= \frac{A_{21}{}^{W}X_w + A_{22}{}^{W}Y_w + A_{23}{}^{W}Z_w + A_{24}}{A_{31}{}^{W}X_w + A_{32}{}^{W}Y_w + A_{33}{}^{W}Z_w + A_{34}} \end{aligned} \tag{16}$$

By arranging the variables, the following expressions are obtained,

$$\begin{aligned} 0 &= A_{11}{}^{W}X_w - A_{31}{}^{I}X_u{}^{W}X_w + A_{12}{}^{W}Y_w \\ &\quad -A_{32}{}^{I}X_u{}^{W}Y_w + A_{13}{}^{W}Z_w - A_{33}{}^{I}X_u{}^{W}Z_w + A_{14} - A_{34}{}^{I}X_u \\ 0 &= A_{21}{}^{W}X_w - A_{31}{}^{I}Y_u{}^{W}X_w + A_{22}{}^{W}Y_w \\ &\quad -A_{32}{}^{I}Y_u{}^{W}Y_w + A_{23}{}^{W}Z_w - A_{33}{}^{I}Y_u{}^{W}Z_w + A_{24} - A_{34}{}^{I}Y_u \end{aligned} \tag{17}$$

Finally, the unknowns A_{ij} are arranged in a 12-parameter vector (A), obtaining the following equation:

$$Q\,A = 0 \tag{18}$$

where A is the vector of 12 unknowns of the equation (19). Q is a matrix of $2n \times 12$ where n is the number of pair points used to calibrate the camera. A pair of points is formed by a 3D point expressed with respect to the world coordinate system $\{W\}$ and its 2D projection expressed in coordinates from the image plane $\{I\}$.

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{21} & A_{22} & A_{23} & A_{24} & A_{31} & A_{32} & A_{33} & A_{34} \end{pmatrix}^{\mathrm{T}} \tag{19}$$

Each pair of points adds to the Q matrix the two following rows,

$$Q_{2i-1} = \begin{pmatrix} {}^{W}X_{ui} \\ {}^{W}Y_{ui} \\ {}^{W}Z_{ui} \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -{}^{I}X_{ui}{}^{W}X_{wi} \\ -{}^{I}X_{ui}{}^{W}Y_{wi} \\ -{}^{I}X_{ui}{}^{W}Z_{wi} \\ -{}^{I}X_{ui} \end{pmatrix}^{\mathrm{T}} \qquad Q_{2i} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ {}^{W}X_{ui} \\ {}^{W}Y_{ui} \\ {}^{W}Z_{ui} \\ 1 \\ -{}^{I}Y_{ui}{}^{W}X_{wi} \\ -{}^{I}Y_{ui}{}^{W}Y_{wi} \\ -{}^{I}Y_{ui}{}^{W}Z_{wi} \\ -{}^{I}Y_{ui} \end{pmatrix}^{\mathrm{T}} \tag{20}$$

Consider then that the 3D position of a set of n calibrating points and their corresponding 2D projection in the image are known (n should be bigger or equal to 6). Moreover, consider without loss of generality that $A_{34} = 1$. This approximation can be assumed since the transformation matrix is defined up to a scale factor [2]. Then, all the elements of the A vector can be obtained by using a linear least-squares technique as the pseudo-inverse [20]. With the aim of applying the pseudo-inverse, it becomes necessary to modify equation (18) considering that $A_{34} = 1$, obtaining:

$$Q'\,A' = B' \tag{21}$$

where,

$$A' = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{21} & A_{22} & A_{23} & A_{24} & A_{31} & A_{32} & A_{33} \end{pmatrix}^{\mathrm{T}} \tag{22}$$

and,

$$Q'_{2i-1} = \begin{pmatrix} {}^{W}X_{ui} \\ {}^{W}Y_{ui} \\ {}^{W}Z_{ui} \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -{}^{I}X_{ui}\,{}^{W}X_{wi} \\ -{}^{I}X_{ui}\,{}^{W}Y_{wi} \\ -{}^{I}X_{ui}\,{}^{W}Z_{wi} \end{pmatrix}^{\mathrm{T}} \qquad Q'_{2i} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ {}^{W}X_{ui} \\ {}^{W}Y_{ui} \\ {}^{W}Z_{ui} \\ 1 \\ -{}^{I}Y_{ui}\,{}^{W}X_{wi} \\ -{}^{I}Y_{ui}\,{}^{W}Y_{wi} \\ -{}^{I}Y_{ui}\,{}^{W}Z_{wi} \end{pmatrix}^{\mathrm{T}} \tag{23}$$

$$B'_{2i-1} = \left({}^{I}X_{ui}\right) \qquad B'_{2i} = \left({}^{I}Y_{ui}\right) \tag{24}$$

Finally, the vector of unknowns (A) is computed by applying the pseudo-inverse shown in the following equation (25).

$$A' = \left(Q'^{\mathrm{T}}Q'\right)^{-1} Q'^{\mathrm{T}}B' \tag{25}$$

3.2 The method of Faugeras

In order to obtain the complete model of the camera proposed by Faugeras and Toscani, it is necessary to combine equations (2), (3), (4), (5) and (14), obtaining (26).

$$\begin{aligned} {}^{I}X_u &= -k_u f \frac{r_{11}\,{}^{W}X_w + r_{12}\,{}^{W}Y_w + r_{13}\,{}^{W}Z_w + t_x}{r_{31}\,{}^{W}X_w + r_{32}\,{}^{W}Y_w + r_{33}\,{}^{W}Z_w + t_z} + u_0 \\ {}^{I}Y_u &= -k_v f \frac{r_{21}\,{}^{W}X_w + r_{22}\,{}^{W}Y_w + r_{23}\,{}^{W}Z_w + t_y}{r_{31}\,{}^{W}X_w + r_{32}\,{}^{W}Y_w + r_{33}\,{}^{W}Z_w + t_z} + v_0 \end{aligned} \tag{26}$$

Note that equations (26) can be expressed in a matricial form in the following manner,

$$\begin{pmatrix} s\,{}^{I}X_d \\ s\,{}^{I}Y_d \\ s \end{pmatrix} = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{W}X_w \\ {}^{W}Y_w \\ {}^{W}Z_w \\ 1 \end{pmatrix} \tag{27}$$

where $\alpha_u = -fk_u$ and $\alpha_v = -fk_v$. Then, by computing the product of both matrices, the transformation matrix A is obtained.

$$\begin{pmatrix} s^I X_d \\ s^I Y_d \\ s \end{pmatrix} = \begin{pmatrix} \alpha_u r_1 + u_0 r_3 & \alpha_u t_x + u_0 t_z \\ \alpha_v r_2 + v_0 r_3 & \alpha_v t_y + v_0 t_z \\ r_3 & t_z \end{pmatrix} \begin{pmatrix} {}^W X_w \\ {}^W Y_w \\ {}^W Z_w \\ 1 \end{pmatrix} \tag{28}$$

$$A = \begin{pmatrix} \alpha_u r_1 + u_0 r_3 & \alpha_u t_x + u_0 t_z \\ \alpha_v r_2 + v_0 r_3 & \alpha_v t_y + v_0 t_z \\ r_3 & t_z \end{pmatrix} \tag{29}$$

The camera parameters can be extracted from the symbolic matrix (A) by equalling it to the numeric matrix obtained by calibrating the camera with the technique of Hall. Note that the orientation of the vectors r_i must be orthogonal and that it is also known that the dot product between two vectors is equal to the multiplication of their norms multiplied by the cosine of the angle they form. Using these relationships, the four intrinsic parameters (α_u, α_v, u_0, v_0) and the six extrinsic ones (r_1, r_2, r_3, t_x, t_y, t_z) can be extracted from equation (29) in the following manner,

$$\begin{array}{lcllcl} u_0 & = & A_1 {A_3}^{\mathrm{T}} & v_0 & = & A_2 {A_3}^{\mathrm{T}} \\ \alpha_u & = & -\left(A_1 {A_1}^{\mathrm{T}} - u_0^2\right)^{1/2} & \alpha_v & = & -\left(A_2 {A_2}^{\mathrm{T}} - v_0^2\right)^{1/2} \\ r_1 & = & \frac{1}{\alpha_u}\left(A_1 - u_0 A_3\right) & t_x & = & \frac{1}{\alpha_u}\left(A_{14} - u_0 A_{34}\right) \\ r_2 & = & \frac{1}{\alpha_v}\left(A_2 - v_0 A_3\right) & t_y & = & \frac{1}{\alpha_v}\left(A_{24} - v_0 A_{34}\right) \\ r_3 & = & A_3 & t_z & = & A_{34} \end{array} \tag{30}$$

where the numerical matrix A is:

$$A = \begin{pmatrix} A_1 & A_{14} \\ A_2 & A_{24} \\ A_3 & A_{34} \end{pmatrix} \tag{31}$$

However, before estimating the camera parameters, the A matrix has to be calculated. Faugeras proposed a slightly different method of estimating A from the one proposed by Hall. Hence, the terms of equation (1) have been rearranged in the following way,

$$\begin{array}{rcl} A_1 {}^W P_w + A_{14} - {}^I X_u \left(A_3 {}^W P_w + A_{34}\right) & = & 0 \\ A_2 {}^W P_w + A_{24} - {}^I Y_u \left(A_3 {}^W P_w + A_{34}\right) & = & 0 \end{array} \tag{32}$$

Both equations are then factorized with respect to the unknowns, obtaining,

$$\begin{array}{rcl} {}^I X_u & = & \frac{A_1}{A_{34}} {}^W P_w + \frac{A_{14}}{A_{34}} - \frac{A_3}{A_{34}} {}^W P_w {}^I X_u \\ {}^I Y_u & = & \frac{A_2}{A_{34}} {}^W P_w + \frac{A_{24}}{A_{34}} - \frac{A_3}{A_{34}} {}^W P_w {}^I Y_u \end{array} \tag{33}$$

At this point, a set of 5 parameters is considered $X = (T_1, T_2, T_3, C_1, C_2)^{\mathrm{T}}$, which are $T_1 = \frac{A_1}{A_{34}}$, $T_2 = \frac{A_3}{A_{34}}$, $T_3 = \frac{A_2}{A_{34}}$, $C_1 = \frac{A_{14}}{A_{34}}$ and $C_2 = \frac{A_{24}}{A_{34}}$.

$$\begin{array}{rcl} {}^I X_u & = & T_1 {}^W P_w + C_1 - T_2 {}^W P_w {}^I X_u \\ {}^I Y_u & = & T_3 {}^W P_w + C_2 - T_2 {}^W P_w {}^I Y_u \end{array} \tag{34}$$

Then, the value of the vector X is obtained by using a linear least-squares technique.

$$B = Q\,X \tag{35}$$

where,

$$Q = \begin{pmatrix} & & \cdots & & \\ {}^{W}P_{w_i}{}^{\mathrm{T}} & -{}^{I}X_{u_i}{}^{W}P_{w_i}{}^{\mathrm{T}} & 0_{1x3} & 1 & 0 \\ 0_{1x3} & -{}^{I}Y_{u_i}{}^{W}P_{w_i}{}^{\mathrm{T}} & {}^{W}P_{w_i}{}^{\mathrm{T}} & 0 & 1 \\ & & \cdots & & \end{pmatrix} \qquad B = \begin{pmatrix} \cdots \\ {}^{I}X_{u_i} \\ {}^{I}Y_{u_i} \\ \cdots \end{pmatrix} \tag{36}$$

Hence, vector X is computed using equation (35).

$$X = \left(Q^{\mathrm{T}}Q\right)^{-1}Q^{\mathrm{T}}B \tag{37}$$

Finally, the camera parameters are extracted from X by using equation (28).

$$\begin{array}{lllllll} T_1 & = & \frac{r_3}{t_z}u_0 + \frac{r_1}{t_z}\alpha_u & & C_1 & = & u_0 + \frac{t_x}{t_z}\alpha_u \\ T_2 & = & \frac{r_3}{t_z} & & & & \\ T_3 & = & \frac{r_3}{t_z}v_0 + \frac{r_2}{t_z}\alpha_v & & C_2 & = & v_0 + \frac{t_y}{t_z}\alpha_v \end{array} \tag{38}$$

At this point, it has to be considered that the norm of the three orientation vectors r_i is equal to unity by definition. By using equations (38), the parameter t_z can then be computed. Hence, considering $r_3 = 1$,

$$t_z = \frac{1}{\|T_2\|} \tag{39}$$

The rest of the parameters can be obtained using the properties of the dot product and the cross product between vectors, which are,

$$v_1 v_2 = \|v_1\|\,\|v_2\|\cos\alpha \qquad v_1 \wedge v_2 = \|v_1\|\,\|v_2\|\sin\alpha \tag{40}$$

so that,

$$\begin{array}{llllll} r_i r_j{}^{\mathrm{T}} & = & 0 & i \neq j & r_i \wedge r_j & = & 1 & i \neq j \\ r_i r_j{}^{\mathrm{T}} & = & 1 & i = j & r_i \wedge r_j & = & 0 & i = j \end{array} \tag{41}$$

The intrinsic parameters can then be obtained in the following way,

$$\begin{array}{lllll} u_0 & = & \frac{T_1 T_2{}^{\mathrm{T}}}{\|T_2\|^2} & v_0 & = & \frac{T_1 T_3{}^{\mathrm{T}}}{\|T_2\|^2} \\ \alpha_u & = & -\frac{\left\|T_1{}^{\mathrm{T}} \wedge T_2{}^{\mathrm{T}}\right\|}{\|T_2\|^2} & \alpha_v & = & -\frac{\left\|T_2{}^{\mathrm{T}} \wedge T_3{}^{\mathrm{T}}\right\|}{\|T_2\|^2} \end{array} \tag{42}$$

Moreover, the extrinsic parameters which model the orientation are the following,

$$\begin{array}{lll} r_1 & = & -\frac{\|T_2\|}{\left\|T_1{}^{\mathrm{T}} \wedge T_2{}^{\mathrm{T}}\right\|}\left(T_1 - \frac{T_1 T_2{}^{\mathrm{T}}}{\|T_2\|^2}T_2\right) \\ r_2 & = & -\frac{\|T_2\|}{\left\|T_2{}^{\mathrm{T}} \wedge T_3{}^{\mathrm{T}}\right\|}\left(T_3 - \frac{T_2 T_3{}^{\mathrm{T}}}{\|T_2\|^2}T_2\right) \\ r_3 & = & \frac{T_2}{\|T_2\|} \end{array} \tag{43}$$

Finally, the extrinsic parameters that model the translation are also obtained from (38).

$$\begin{array}{lll} t_x & = & -\frac{\|T_2\|}{\left\|T_1{}^{\mathrm{T}} \wedge T_2{}^{\mathrm{T}}\right\|}\left(C_1 - \frac{T_1 T_2{}^{\mathrm{T}}}{\|T_2\|^2}\right) \\ t_y & = & -\frac{\|T_2\|}{\left\|T_2{}^{\mathrm{T}} \wedge T_3{}^{\mathrm{T}}\right\|}\left(C_2 - \frac{T_2 T_3{}^{\mathrm{T}}}{\|T_2\|^2}\right) \\ t_z & = & \frac{1}{\|T_2\|} \end{array} \tag{44}$$

By using the r_i vectors in equations (43), the rotation matrix ${}^{C}R_{W}$ is directly obtained. The three angles α, β and γ can then be computed by equalling the symbolic rotation matrix to the numeric matrix obtained by calibration. At this point, all the parameters of the linear model of Faugeras are obtained. These parameters determine the relationship between the 3D object points with their projections, as shown in equation (28). However, the model of Faugeras can be more accurate if radial lens distortion is included.

3.3 The method of Faugeras with radial distortion

When a bright accuracy is necessary, the linear method of Faugeras becomes useless. However, it can be easily modified by including the radial lens distortion as it has been shown in section 2.3. However, the equations become non-linear, and the linear least-squares technique has to be replaced by an iterative algorithm.

Note that by combining equations (2), (3), (4), (6) and (7), the equations (45) are obtained.

$$\begin{aligned} {}^{C}X_d + {}^{C}X_d k_1 r^2 &= f\frac{r_{11}{}^{W}X_w + r_{12}{}^{W}Y_w + r_{13}{}^{W}Z_w + t_x}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} \\ {}^{C}Y_d + {}^{C}Y_d k_1 r^2 &= f\frac{r_{21}{}^{W}X_w + r_{22}{}^{W}Y_w + r_{23}{}^{W}Z_w + t_y}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} \\ r &= \sqrt{{}^{C}X_d{}^{2} + {}^{C}Y_d{}^{2}} \end{aligned} \tag{45}$$

Moreover, equations (14) have to be used to transform from metric coordinates to pixels. Then, equation (46) defines the vector of unknowns which can be computed by using an iterative method as, for instance, the method of Newton-Raphson or Levenberg-Marquardt, among others [36].

$$x = (\alpha, \beta, \gamma, t_x, t_y, t_z, k_u, k_v, u_0, v_0, k_1)^{\mathrm{T}} \tag{46}$$

For example, the general method of Newton-Raphson minimizes the following equation,

$$G(x_k) \approx G(x_{k-1}) + J\Delta x_k \tag{47}$$

where x is the unknown vector, $G(x)$ is the minimization function, $G(x_k)$ is a value close to the solution, and J represents the jacobian matrix of the function $G(x)$. With the aim of finding a solution of Δx_k, it is necessary to equal $G(x_k)$ to zero.

$$G(x_k) = 0 \tag{48}$$

Note that one of the problems of convergence in iterative algorithms is the initial guess. However, an initial guess can be obtained by calibrating the linear method of Faugeras-Toscani without including lens distortion, and assuming $k_1 = 0$. Moreover, the difference between the initial value and the estimated parameters will be the error of the function. For each iteration it is necessary to calculate the value of Δx_k to know the new value of x.

$$J\Delta x_k = -G(x_{k-1}) \tag{49}$$

By, applying the equations (45) and (14), and passing all the terms from the equality to the same side, functions $U(x)$ and $V(x)$ are defined.

$$\begin{aligned} U\left(x\right) &= f\frac{r_{11}{}^{W}X_w + r_{12}{}^{W}Y_w + r_{13}{}^{W}Z_w + t_x}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} - \frac{\left({}^{I}X_d - u_0\right)}{-k_u} \\ &\quad -k_1\left(\left(\frac{\left({}^{I}X_d - u_0\right)}{-k_u}\right)^2 + \left(\frac{\left({}^{I}Y_d - v_0\right)}{-k_v}\right)^2\right)\frac{\left({}^{I}X_d - u_0\right)}{-k_u} \\ V\left(x\right) &= f\frac{r_{21}{}^{W}X_w + r_{22}{}^{W}Y_w + r_{23}{}^{W}Z_w + t_y}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} - \frac{\left({}^{I}Y_d - v_0\right)}{-k_v} \\ &\quad -k_1\left(\left(\frac{\left({}^{I}X_d - u_0\right)}{-k_u}\right)^2 + \left(\frac{\left({}^{I}Y_d - v_0\right)}{-k_v}\right)^2\right)\frac{\left({}^{I}Y_d - v_0\right)}{-k_v} \end{aligned} \tag{50}$$

In continuation, with the aim of solving the system, it is necessary to apply equations (50) to the n calibrating points. However, in order to apply equation (49), it is necessary to get the symbolic function $G(x)$ and its partial derivative matrix J, as it is shown in the following equations,

$$G(x_{k-1}) = \begin{pmatrix} U_1\left(x_{k-1}\right) \\ V_1\left(x_{k-1}\right) \\ \vdots \\ V_n\left(x_{k-1}\right) \end{pmatrix} \tag{51}$$

$$J = \begin{pmatrix} \frac{\partial U_1(x_{k-1})}{\partial \alpha} & \frac{\partial U_1(x_{k-1})}{\partial \beta} & \cdots & \frac{\partial U_1(x_{k-1})}{\partial k_1} \\ \frac{\partial V_1(x_{k-1})}{\partial \alpha} & \frac{\partial V_1(x_{k-1})}{\partial \beta} & \cdots & \frac{\partial V_1(x_{k-1})}{\partial k_1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial V_n(x_{k-1})}{\partial \alpha} & \frac{\partial V_n(x_{k-1})}{\partial \beta} & \cdots & \frac{\partial V_n(x_{k-1})}{\partial k_1} \end{pmatrix} \tag{52}$$

Finally, the parameters of the model are obtained by applying the pseudo-inverse of equations (53) in each iteration. The more iterations done, the higher the accuracy obtained until convergence is achieved.

$$\begin{aligned} \Delta x_k &= -\left(J^{\mathrm{T}}J\right)^{-1}J^{\mathrm{T}}G\left(x_{k-1}\right) \\ x_k &= x_{k-1} + \Delta x_k \end{aligned} \tag{53}$$

3.4 The method of Tsai

The non-linear method of Faugeras was based on fixing the initial guess without considering lens distortion. Moreover, a large number of iterations are usually necessary to obtain an accurate value of the camera parameters. The method of Tsai [24] also models the radial lens distortion but assumes that there are some parameters of the camera which are provided by manufacturers. This fact reduces the number of calibrating parameters in the first step where an initial guess is estimated. Moreover, although all the parameters are iteratively optimized in the last step, the number of iterations is considerably reduced by using the calibrating algorithm proposed by Tsai.

Firstly, by combining equations (2), (3), (4), (6), and (7), the equations (45) are obtained. Note that at this point model of Tsai is equivalent to the previous model of Faugeras with distortion (45).

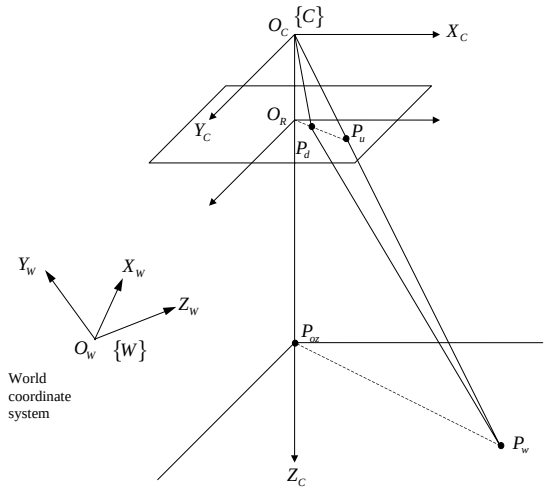


Figure 2: Illustration of the radial alignment constraint [24].

Once ${}^{C}X'_{d}$ and ${}^{C}Y'_{d}$ are obtained in metric coordinates by using equation (15), they can be expressed in pixels ${}^{I}X_{d}$ and ${}^{I}Y_{d}$ and the following equations are obtained.

$$ {}^{C}X'_{di} = -\left({}^{I}X_{di} - u_0\right) d'_x \qquad {}^{C}Y'_{di} = -\left({}^{I}Y_{di} - v_0\right) d_y \tag{54} $$

where,

$$ {}^{C}X'_{di} = {}^{C}X_{di} s_x \qquad {}^{C}Y'_{di} = {}^{C}Y_{di} \tag{55} $$

It is necessary therefore to find a relationship between the image point P_d (in metric coordinates) with respect to the object point P_w. Figure 2 shows how the radial distortion affects the camera model. It can be observed that the segment $\overline{O_R P_d}$ is parallel to the segment $\overline{P_{oz} P_w}$. Considering this constraint, the following relationship is established,

$$ \overline{O_R P_d} // \overline{P_{oz} P_w} \Rightarrow \overline{O_R P_d} \times \overline{P_{oz} P_w} = 0 \tag{56} $$

By using equation (56), the following equations are obtained.

$$ \overline{O_R P_d} \times \overline{P_{oz} P_w} = 0 \tag{57} $$

$$ \left({}^{C}X_d, {}^{C}Y_d\right) \times \left({}^{C}X_w, {}^{C}Y_w\right) = 0 \tag{58} $$

$$ {}^{C}X_d {}^{C}Y_w - {}^{C}Y_d {}^{C}X_w = 0 \tag{59} $$

Equation (59) can be arranged expressing the object point P_w with respect to the world coordinate system, instead of expressing it with respect to the camera coordinate system.

$$ {}^{C}X_d \left(r_{21} {}^{W}X_w + r_{22} {}^{W}Y_w + r_{23} {}^{W}Z_w + t_y\right) = {}^{C}Y_d \left(r_{11} {}^{W}X_w + r_{12} {}^{W}Y_w + r_{13} {}^{W}Z_w + t_x\right) \tag{60} $$

Operating equation (60) and arranging the terms,

$$ {}^{C}X_d = {}^{C}Y_d {}^{W}X_w \frac{r_{11}}{t_y} + {}^{C}Y_d {}^{W}Y_w \frac{r_{12}}{t_y} + {}^{C}Y_d {}^{W}Z_w \frac{r_{13}}{t_y} + {}^{C}Y_d \frac{t_x}{t_y} $$

$$-{}^{C}X_d{}^{W}X_w\frac{r_{21}}{t_y} - {}^{C}X_d{}^{W}Y_w\frac{r_{22}}{t_y} + {}^{C}X_d{}^{W}Z_w\frac{r_{23}}{t_y} \tag{61}$$

In order to compute equation (61) for the n points obtained from equations (54), it is necessary to combine equation (61) with the equations (55), obtaining,

$$\begin{aligned} {}^{C}X'_{di} &= {}^{C}Y'_{di}{}^{W}X_{wi}\frac{s_x r_{11}}{t_y} + {}^{C}Y'_{di}{}^{W}Y_{wi}\frac{s_x r_{12}}{t_y} + {}^{C}Y'_{di}{}^{W}Z_{wi}\frac{s_x r_{13}}{t_y} + {}^{C}Y'_{di}\frac{s_x t_x}{t_y} \\ &\quad -{}^{C}X'_{di}{}^{W}X_{wi}\frac{r_{21}}{t_y} - {}^{C}X'_{di}{}^{W}Y_{wi}\frac{r_{22}}{t_y} + {}^{C}X'_{di}{}^{W}Z_{wi}\frac{r_{23}}{t_y} \end{aligned} \tag{62}$$

At this point, a system with n equations and 7 unknowns is obtained, which can be expressed in the following form,

$$\begin{pmatrix} {}^{C}Y'_{di}{}^{W}X_{wi} \\ {}^{C}Y'_{di}{}^{W}Y_{wi} \\ {}^{C}Y'_{di}{}^{W}Z_{wi} \\ {}^{C}Y'_{di} \\ -{}^{C}X'_{di}{}^{W}X_{wi} \\ -{}^{C}X'_{di}{}^{W}Y_{wi} \\ -{}^{C}X'_{di}{}^{W}Z_{wi} \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} t_y^{-1}s_x r_{11} \\ t_y^{-1}s_x r_{12} \\ t_y^{-1}s_x r_{13} \\ t_y^{-1}s_x t_x \\ t_y^{-1}s_x r_{21} \\ t_y^{-1}s_x r_{22} \\ t_y^{-1}s_x r_{23} \end{pmatrix} = {}^{C}X'_{di} \tag{63}$$

In order to simplify the notation, the 7 unknown components of the vector can be renamed a_i.

$$\begin{array}{cc} a_1 = t_y^{-1}s_x r_{11} & a_5 = t_y^{-1}r_{21} \\ a_2 = t_y^{-1}s_x r_{12} & a_6 = t_y^{-1}r_{22} \\ a_3 = t_y^{-1}s_x r_{13} & a_7 = t_y^{-1}r_{23} \\ \multicolumn{2}{c}{a_4 = t_y^{-1}s_x t_x} \end{array} \tag{64}$$

Note that the a_i components can be easily computed by using a least-squares technique. Therefore, the point of interest is to extract the calibrating parameters of the camera from these a_i components. First t_y can be obtained by using equations (64) in the following manner,

$$t_y = \frac{\|r_2\|}{\|a_{5,6,7}\|} \tag{65}$$

and equation (65) is simplified because the norm of the vector r_2 is equal to the unity, obtaining the parameter t_y.

$$|t_y| = \frac{1}{\sqrt{{a_5}^2 + {a_6}^2 + {a_7}^2}} \tag{66}$$

However, equation (66) is insufficient since it does not provide the sign of the t_y component. In order to determine this sign, a point ($^{I}X_d$,$^{I}Y_d$) located at the periphery of the image, far from the center, is taken from the set of test points (its corresponding 3D point is also kept). It is then supposed that the t_y sign is positive, and the following equations are computed.

$$\begin{array}{cc} r_{11} = a_1 t_y / s_x & r_{21} = a_5 t_y \\ r_{12} = a_2 t_y / s_x & r_{22} = a_6 t_y \\ r_{13} = a_3 t_y / s_x & r_{23} = a_7 t_y \\ \multicolumn{2}{c}{t_x = a_4 t_y} \end{array} \tag{67}$$

By using the corresponding 3D point ($^{W}X_w$,$^{W}Y_w$,$^{W}Z_w$), the linear projection of this 3D point on the image plane (sans lens distortion) can be computed by using equations (68).

$$\begin{aligned} {}^{C}X_u &= r_{11}{}^{W}X_w + r_{12}{}^{W}Y_w + r_{13}{}^{W}Z_w + t_x \\ {}^{C}Y_u &= r_{21}{}^{W}X_w + r_{22}{}^{W}Y_w + r_{23}{}^{W}Z_w + t_y \end{aligned} \tag{68}$$

At this point the t_y sign can be verified. If both components of the point (${}^{C}X_u$,${}^{C}Y_u$) have the same sign as the point (${}^{I}X_d$,${}^{I}Y_d$), it means that the t_y sign was correctly chosen as positive. Otherwise, it has to be considered negative.

The second parameter to be extracted is the scale factor (s_x). Note that by arranging equations (64), the following equation is obtained,

$$s_x = \frac{\|a_{1,2,3}\|\, t_y}{\|r_1\|} \tag{69}$$

where it is known that the norm of r_1 is the unity and the scale factor is always positive. Then, s_x is obtained by using equation (70).

$$s_x = \sqrt{{a_1}^2 + {a_2}^2 + {a_3}^2}\,|t_y| \tag{70}$$

Furthermore, the 2D points, with respect to the camera coordinate system (${}^{C}X_d$,${}^{C}Y_d$), can be computed from the same point with respect to the image coordinate system, that is (${}^{I}X_d$,${}^{I}Y_d$), by using equations (55). Moreover, by using equations (67) the r_1 and r_2 vectors of the rotation matrix ${}^{C}R_W$, and the first element of the translation vector ${}^{C}Y_W$, i.e. t_x, can be calculated. Nevertheless, the third orientation vector (r_3) can be computed by a cross product between r_1 and r_2 because of the property of orthogonality, (note also that the determinant of any rotation matrix is the unity, i.e. $|{}^{C}R_W| = 1$). At this point, the first three steps of the method of Tsai are completed, see figure 3.

However, the following parameters are still unknown: the focal distance (f), the radial lens distortion coefficient (k_1), and the translation of the camera with respect to the Z axis (t_z). In order to compute these last three parameters, a linear approximation is first used without considering the k_1 parameter. The linear approximation is shown in equation (71), which was obtained from equations (45).

$$\begin{pmatrix} {r_{21}}^{W}X_{wi} + {r_{22}}^{W}Y_{wi} + {r_{23}}^{W}Z_{wi} + t_y & -{}^{C}Y_d \end{pmatrix} \begin{pmatrix} f \\ t_z \end{pmatrix} = \\ \left({r_{31}}^{W}X_{wi} + {r_{32}}^{W}Y_{wi} + {r_{33}}^{W}Z_{wi}\right) {}^{C}Y_d \tag{71}$$

Equation (71) has now been applied to the whole set of test points, obtaining a system of n equations and two unknowns. The linear approximation of both unknowns, f and t_z, is obtained by using a pseudo-inverse. However, in order to calculate a better approximation including the k_1 parameter, it is necessary to iterate equations (45) by using an optimization method considering the linear method with $k_1 = 0$ as an initial solution.

Finally, all the parameters are optimized iteratively with the aim of obtaining an accurate solution. The entire process is explained in figure 3.

3.5 The method of Weng

The method of Tsai is based on modelling radial lens distortion. The accuracy obtained by Tsai is sufficient for most applications. However, in some cases where the camera lens needs to be accurately modelled, a simple

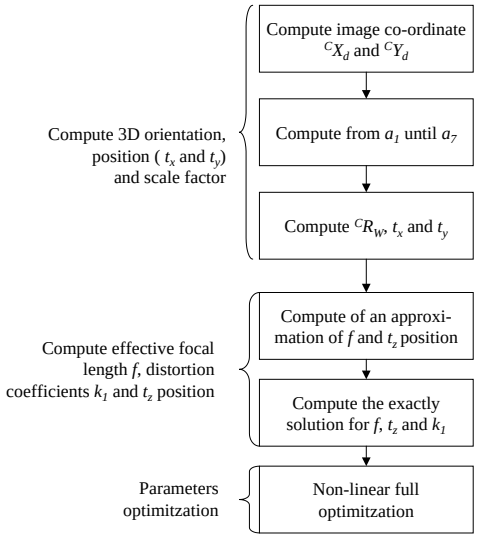


Figure 3: Flowchart of the method of Tsai.

radial approximation is not sufficient. In such situations, Weng [33] modifies the model proposed by Faugeras-Toscani [32] including up to three types of lens distortion, as has been explained in section 2.3. This fact increases the number of steps needed to calibrate the camera. A flowchart of the entire process is detailed in figure 4.

The first step is to obtain the complete model of Weng. However, Weng proposes to simplify the equations by introducing a variable substitution. Hence, equalling equations (9) and (14), equations (72) are obtained.

$$\begin{array}{rcl} {}^{C}X_u + \delta_x\left({}^{C}X_u, {}^{C}Y_u\right) & = & \left({}^{I}X_d - u_0\right)/-k_u \\ {}^{C}Y_u + \delta_y\left({}^{C}X_u, {}^{C}Y_u\right) & = & \left({}^{I}Y_d - v_0\right)/-k_v \end{array} \tag{72}$$

At this point, two new unknowns are introduced, in the following manner,

$${}^{C}\hat{X}_d = \left({}^{I}X_d - u_0\right)/\alpha_u \qquad {}^{C}\hat{Y}_d = \left({}^{I}X_d - v_0\right)/\alpha_v \tag{73}$$

A substitution is then applied to simplify equations (72), obtaining equations (74).

$$\frac{{}^{C}X_u}{f} = {}^{C}\hat{X}_d - \frac{\delta_x\left({}^{C}X_u, {}^{C}Y_u\right)}{f} \qquad \frac{{}^{C}Y_u}{f} = {}^{C}\hat{Y}_d - \frac{\delta_y\left({}^{C}X_u, {}^{C}Y_u\right)}{f} \tag{74}$$

This replacement of unknowns is necessary because the value of $({}^{C}X_u, {}^{C}Y_u)$ cannot be obtained by observation. This fact makes it necessary to compute the distortion from the observed points after representing them with respect to the camera coordinate system, that is from $({}^{C}\hat{X}_d, {}^{C}\hat{Y}_d)$ [30, 33]. This replacement is reasonable because the distortion on the image plane suffered by the point $({}^{C}X_u, {}^{C}Y_u)$ is approximately equal to the distortion suffered by the point$({}^{C}\hat{X}_d, {}^{C}\hat{Y}_d)$. Therefore, the distortion coefficients in δ'_x and δ'_y will be estimated from $({}^{C}\hat{X}_d, {}^{C}\hat{Y}_d)$, instead of δ_x and δ_y, which was estimated from $({}^{C}X_u, {}^{C}Y_u)$. As a result, the equations which relate distorted to undistorted points are the following,

$$\frac{{}^{C}X_u}{f} = {}^{C}\hat{X}_d + \delta'_x\left({}^{C}\hat{X}_d, {}^{C}\hat{Y}_d\right) \qquad \frac{{}^{C}Y_u}{f} = {}^{C}\hat{Y}_d + \delta'_y\left({}^{C}\hat{X}_d, {}^{C}\hat{Y}_d\right) \tag{75}$$

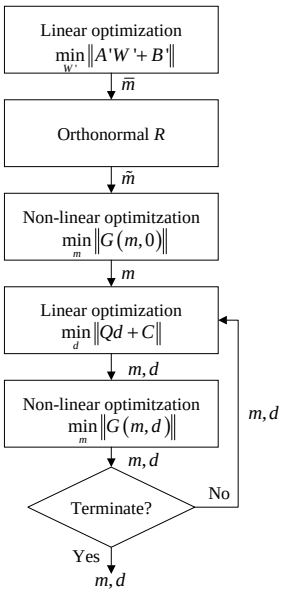


Figure 4: Flowchart of the method of Weng.

Finally, redefining the coefficients k_1 and g_1 up to g_4, and combining equations (2), (3) and (75) the complete camera model is obtained,

$$\begin{aligned}
&\frac{r_{11}{}^{W}X_w + r_{12}{}^{W}Y_w + r_{13}{}^{W}Z_w + t_x}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} = {}^{C}\hat{X}_d \\
&\quad + (g_1 + g_3)\,{}^{C}\hat{X}_d{}^{2} + g_4\,{}^{C}\hat{X}_d\,{}^{C}\hat{Y}_d + g_1\,{}^{C}\hat{Y}_d{}^{2} + k_1\,{}^{C}\hat{X}_d\left({}^{C}\hat{X}_d{}^{2} + {}^{C}\hat{Y}_d{}^{2}\right) \\
&\frac{r_{21}{}^{W}X_w + r_{22}{}^{W}Y_w + r_{23}{}^{W}Z_w + t_y}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} = {}^{C}\hat{Y}_d \\
&\quad + g_2\,{}^{C}\hat{X}_d{}^{2} + g_3\,{}^{C}\hat{X}_d\,{}^{C}\hat{Y}_d + (g_2 + g_4)\,{}^{C}\hat{Y}_d{}^{2} + k_1\,{}^{C}\hat{Y}_d\left({}^{C}\hat{X}_d{}^{2} + {}^{C}\hat{Y}_d{}^{2}\right)
\end{aligned} \tag{76}$$

In order to be able to calibrate all the parameters of the model, Weng proposes to obtain a first approximation of the linear parameters, i.e. the extrinsic and intrinsic parameters without distortion. The m vector is now defined containing these linear parameters.

$$m = (u_0, v_0, \alpha_u, \alpha_v, t_x, t_y, t_z, \alpha, \beta, \gamma)^{\mathrm{T}} \tag{77}$$

Furthermore, the non-linear parameters which model the lens define a new vector d.

$$d = (k_1, g_1, g_2, g_3, g_4)^{\mathrm{T}} \tag{78}$$

Moreover, the calibration is based on the 3D test points and their projections. Let us call F the camera model, Ω the set of 3D points, and ω the set of their projections. Then, the calibration problem is the same as optimizing the parameters (m^*,d^*) which minimize the equation F by using both sets of test points.

$$F(\Omega, \omega, m^*, d^*) = \min_{m,d} F(\Omega, \omega, m, d) \tag{79}$$

This problem of optimization can be solved by using a non-linear method, in the following manner:

1. Fix $d = 0$.

2. Calculate m, which minimizes F by fixing d, that is: $\min_{m} F(\Omega, \omega, m, d)$

3. Calculate d, which minimizes F by fixing m, that is: $\min_{d} F(\Omega, \omega, m, d)$

4. Return to step 2 until the minimization error is sufficiently tolerable.

This method of optimization is used to solve diverse problems. First, the vector d can be coupled with m making the minimization of F false. Second, the intrinsic parameters can not be optimized until a sufficient approximation of the extrinsic parameters is achieved. Third, since m corresponds to an approximation of the linear parameters, it cannot be the best solution if a significant distortion is presented.

With the aim of obtaining a good estimation of m with a non-linear optimization method, it is necessary to obtain an initial guess before iterating. Therefore, the initial guess is calculated supposing $d = 0$. Then, the model of Weng removing distortion, see equations (74), is applied to the n calibrating points, obtaining $2n$ equations of the form:

$$\begin{aligned} \left({}^{I}X_{ui} - u_0\right) {}^{W}X_{wi} r_{31} + \left({}^{I}X_{ui} - u_0\right) {}^{W}Y_{wi} r_{32} + \left({}^{I}X_{ui} - u_0\right) {}^{W}Z_{wi} r_{33} + \left({}^{I}X_{ui} - u_0\right) t_z \\ -\alpha_u {}^{W}X_{wi} r_{11} - \alpha_u {}^{W}Y_{wi} r_{12} - \alpha_u {}^{W}Z_{wi} r_{13} - \alpha_u t_x = 0 \\ \left({}^{I}Y_{ui} - v_0\right) {}^{W}X_{wi} r_{31} + \left({}^{I}Y_{ui} - v_0\right) {}^{W}Y_{wi} r_{32} + \left({}^{I}Y_{ui} - v_0\right) {}^{W}Z_{wi} r_{33} + \left({}^{I}Y_{ui} - v_0\right) t_z \\ -\alpha_v {}^{W}X_{wi} r_{21} - \alpha_v {}^{W}Y_{wi} r_{22} - \alpha_v {}^{W}Z_{wi} r_{23} - \alpha_v t_y = 0 \end{aligned} \tag{80}$$

By using equations (80), all the m parameters can be calculated. As the m vector has 10 unknowns, it is necessary to use at least 5 test points. Nevertheless, a large number of points is used in order to obtain a more accurate solution. The following parameters are then defined,

$$\begin{array}{lcllcl} W_1 & = & \alpha_u r_1 + u_0 r_3 & w_4 & = & \alpha_u t_x + u_0 t_z \\ W_2 & = & \alpha_v r_2 + v_0 r_3 & w_5 & = & \alpha_v t_y + v_0 t_z \\ W_3 & = & r_3 & w_6 & = & t_z \end{array} \tag{81}$$

where the vectors r_1, r_2 and r_3 correspond to each row of the matrix ${}^{C}R_{W}$, respectively. Moreover, the set of equations (80) is expressed in matricial form as

$$A\,W = 0 \tag{82}$$

where A is a matrix with $2n$ rows and 12 columns.

$$A = \begin{pmatrix} -{}^{W}P_{w1}{}^{\mathrm{T}} & 0_{1x3} & {}^{I}X_{u1}\,{}^{W}P_{w1}{}^{\mathrm{T}} & -1 & 0 & {}^{I}X_{u1} \\ 0_{1x3} & -{}^{W}P_{w1}{}^{\mathrm{T}} & {}^{I}Y_{u1}\,{}^{W}P_{w1}{}^{\mathrm{T}} & 0 & -1 & {}^{I}Y_{u1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -{}^{W}P_{wn}{}^{\mathrm{T}} & 0_{1x3} & {}^{I}X_{un}\,{}^{W}P_{wn}{}^{\mathrm{T}} & -1 & 0 & {}^{I}X_{un} \\ 0_{1x3} & -{}^{W}P_{wn}{}^{\mathrm{T}} & {}^{I}Y_{un}\,{}^{W}P_{wn}{}^{\mathrm{T}} & 0 & -1 & {}^{I}Y_{un} \end{pmatrix} \tag{83}$$

However, the vector $W = (W_1, W_2, W_3, w_4, w_5, w_6)^{\mathrm{T}}$ cannot be directly calculated because of the homogeneity of the system, which deals with multiple solutions. However, only one of these potential solutions satisfies the following conditions: a) The norm of the W_3 vector has to be the unity because it is the third row of the rotation

matrix; b) The w_6 sign has to coincide with the position of the optical center with respect to the image plane: Positive if the z-axis intersects the image plane, and negative if otherwise.

With the aim of avoiding the homogeneity of the system of equation (82), it is necessary to impose the following temporary restriction,

$$w_6 = t_z = 1 \tag{84}$$

Hence, equation (82) is modified, obtaining

$$A'\,W' + B' = 0 \tag{85}$$

where A' is the first 11 columns of the A matrix, B' is the last column of A and W' is a vector of the 11 unknowns, i.e. $W' = (W_1, W_2, W_3, w_4, w_5)$. Then, W' is computed by using the pseudo-inverse,

$$W' = \left(A'^{\mathrm{T}} A'\right)^{-1} A'^{\mathrm{T}} \left(-B'\right) \tag{86}$$

At this point, W' is the solution of the system shown in equation (85). However, in order to be a solution of equation (82) as well, it has to accomplish the two constraints. Therefore, the solution is divided by $\|W_3\|$, which forces the norm of W_3 to be the unity, and replaces the w_6 sign if necessary. See equation (87).

$$S = \begin{pmatrix} S_1 \\ S_2 \\ S_3 \\ s_4 \\ s_5 \\ s_6 \end{pmatrix} = \pm \frac{1}{\|W_3\|} \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ w_4 \\ w_5 \\ w_6 \end{pmatrix} \tag{87}$$

Moreover, knowing that the vectors r_1, r_2 and r_3 are orthogonal, equations (81) can be applied to obtain a first approximation of the m vector.

$$\begin{array}{lcllcl} \bar{u}_0 & = & S_1^{\mathrm{T}} S_3 & \bar{v}_0 & = & S_2^{\mathrm{T}} S_3 \\ \bar{\alpha}_u & = & -\|S_1 - \bar{u}_0 S_3\| & \bar{\alpha}_v & = & -\|S_2 - \bar{v}_0 S_3\| \\ \bar{t}_x & = & (s_4 - \bar{u}_0 s_6)/\bar{\alpha}_u & \bar{r}_1 & = & (S_1 - \bar{u}_0 S_3)/\bar{\alpha}_u \\ \bar{t}_y & = & (s_5 - \bar{v}_0 s_6)/\bar{\alpha}_v & \bar{r}_2 & = & (S_2 - \bar{v}_0 S_3)/\bar{\alpha}_v \\ \bar{t}_z & = & s_6 & \bar{r}_3 & = & S_3 \end{array} \tag{88}$$

However, this first approximation does not imply that the matrix ${}^C\bar{R}_W$ is orthonormal. The next step consists of calculating the orthonormal matrix ${}^C\tilde{R}_W$. The first step is to verify,

$$\left\| {}^C\tilde{R}_W - {}^C\bar{R}_W \right\| = \min_{{}^C R_W} \left\| {}^C\bar{R}_W - {}^C R_W \right\| \tag{89}$$

With the aim of solving equation (89), it is rewritten including a 3×3 identity matrix I.

$$\left\| {}^C\tilde{R}_W I - {}^C\bar{R}_W \right\| = \min_{{}^C R_W} \left\| {}^C\bar{R}_W - {}^C R_W \right\| \tag{90}$$

It is then defined a 4×4 matrix B,

$$B = \sum_{i=1}^{3} B_i^{\mathrm{T}} B_i \tag{91}$$

where,

$$B_i = \begin{pmatrix} 0 & (i_i - \bar{r}_i)^{\mathrm{T}} \\ \bar{r}_i - i_i & (\bar{r}_i + i_i)_{\times} \end{pmatrix} \tag{92}$$

and where $I = (i_1, i_2, i_3)^{\mathrm{T}}$, and $(x,y,z)_{\times}$ is the antisymmetric matrix of the vector (x,y,z), that is:

$$(x, y, z)_{\times} = \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix} \tag{93}$$

The vector $q = (q_0, q_1, q_2, q_3)^{\mathrm{T}}$ is then obtained by calculating the eigenvalues associated with matrix B, where q_i is an eigenvalue and $q_i \leq q_{i+1}$. Finally, the solution of the matrix ${}^{C}\tilde{R}_W$ is shown in the following equation,

$${}^{C}\tilde{R}_W = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2\left(q_1 q_2 - q_0 q_3\right) & 2\left(q_1 q_3 + q_0 q_2\right) \\ 2\left(q_2 q_1 + q_0 q_3\right) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2\left(q_2 q_3 - q_0 q_1\right) \\ 2\left(q_3 q_1 - q_0 q_2\right) & 2\left(q_3 q_2 + q_0 q_1\right) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \tag{94}$$

With the orthornomal rotation matrix, the rest of the parameters are recalculated once more, obtaining:

$$\begin{array}{rclcrcl} \tilde{u}_0 & = & S_1^{\mathrm{T}} \tilde{r}_3 & & \tilde{v}_0 & = & S_2^{\mathrm{T}} \tilde{r}_3 \\ \tilde{\alpha}_u & = & -\left\| S_1 - \tilde{u}_0 \tilde{r}_3 \right\| & & \tilde{\alpha}_v & = & -\left\| S_2 - \tilde{v}_0 \tilde{r}_3 \right\| \\ & & \tilde{t}_x & = & \left(s_4 - \tilde{u}_0 s_6\right)/\tilde{\alpha}_u & & \\ & & \tilde{t}_y & = & \left(s_5 - \tilde{v}_0 s_6\right)/\tilde{\alpha}_v & & \\ & & \tilde{t}_z & = & \bar{t}_z & & \end{array} \tag{95}$$

An iterative method is then used to recalculate, for the third time, the values of m, assuming zero distortion. Finally, a two-stage iterative method is used. In the first stage, the parameters of d are linearly obtained by using least-squares. The second stage computes the values of m iteratively. These stages are repeated as many times as needed depending on the desired accuracy.

3.5.1 Stage of non-linear optimization of m by fixing d.

The camera model of Weng is expressed in equation (96), see also (76).

$$\begin{array}{rcl} U(x) & = & \dfrac{r_{11}{}^{W}X_w + r_{12}{}^{W}Y_w + r_{13}{}^{W}Z_w + t_x}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} \\ & & -{}^{C}\hat{X}_d - \left(g_1 + g_3\right){}^{C}\hat{X}_d{}^2 - g_4\,{}^{C}\hat{X}_d\,{}^{C}\hat{Y}_d - g_1\,{}^{C}\hat{Y}_d{}^2 - k_1\,{}^{C}\hat{X}_d\left({}^{C}\hat{X}_d{}^2 + {}^{C}\hat{Y}_d{}^2\right) \\ V(x) & = & \dfrac{r_{21}{}^{W}X_w + r_{22}{}^{W}Y_w + r_{23}{}^{W}Z_w + t_y}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} \\ & & -{}^{C}\hat{Y}_d - g_2\,{}^{C}\hat{X}_d{}^2 - g_3\,{}^{C}\hat{X}_d\,{}^{C}\hat{Y}_d - \left(g_2 + g_4\right){}^{C}\hat{Y}_d{}^2 - k_1\,{}^{C}\hat{Y}_d\left({}^{C}\hat{X}_d{}^2 + {}^{C}\hat{Y}_d{}^2\right) \end{array} \tag{96}$$

Equation (97) shows the function of minimization that has to be used in optimization.

$$\sum_{i=1}^{n} \left\{ \left({}^{I}X_{di} - {}^{I}X_{di}\left(m, d\right)\right)^2 + \left({}^{I}Y_{di} - {}^{I}Y_{di}\left(m, d\right)\right)^2 \right\} \tag{97}$$

At this point any optimization algorithm such as Newton-Raphson or Levenberg-Marquardt can be used to optimize equations (96).

3.5.2 Stage of linear optimization of d by fixing m.

Note that by arranging equations (14) and (76), the equations which have to be optimized become linear. Therefore, they can be optimized by using the pseudo-inverse technique. The linear equations obtained are the following,

$$\begin{aligned} {}^{I}X_d\,(m,d) - {}^{I}X_d &= u_0 + \alpha_u\,{}^{C}\hat{X}_d - {}^{I}X_d \\ &= u_0 + \alpha_u \left(\frac{r_{11}\,{}^{W}X_w + r_{12}\,{}^{W}Y_w + r_{13}\,{}^{W}Z_w + t_x}{r_{31}\,{}^{W}X_w + r_{32}\,{}^{W}Y_w + r_{33}\,{}^{W}Z_w + t_z} \right. \\ &\quad \left. - (g_1 + g_3)\,{}^{C}\hat{X}_d{}^{2} - g_4\,{}^{C}\hat{X}_d\,{}^{C}\hat{Y}_d - g_1\,{}^{C}\hat{Y}_d^2 - k_1\,{}^{C}\hat{X}_d \left({}^{C}\hat{X}_d{}^{2} + {}^{C}\hat{Y}_d{}^{2} \right) \right) - {}^{I}X_d \end{aligned} \tag{98}$$

$$\begin{aligned} {}^{I}Y_d\,(m,d) - {}^{I}Y_d &= v_0 + \alpha_v\,{}^{C}\hat{Y}_d - {}^{I}Y_d \\ &= v_0 + \alpha_v \left(\frac{r_{21}\,{}^{W}X_w + r_{22}\,{}^{W}Y_w + r_{23}\,{}^{W}Z_w + t_y}{r_{31}\,{}^{W}X_w + r_{32}\,{}^{W}Y_w + r_{33}\,{}^{W}Z_w + t_z} \right. \\ &\quad \left. - g_2\,{}^{C}\hat{X}_d{}^{2} - g_3\,{}^{C}\hat{X}_d\,{}^{C}\hat{Y}_d - (g_2 + g_4)\,{}^{C}\hat{Y}_d^2 - k_1\,{}^{C}\hat{Y}_d \left({}^{C}\hat{X}_d{}^{2} + {}^{C}\hat{Y}_d{}^{2} \right) \right) - {}^{I}Y_d \end{aligned}$$

where the function to minimize is expressed in equation (99):

$$\min_{d} \|Q\,d + C\| \tag{99}$$

where,

$$C = \begin{pmatrix} u_0 + \alpha_u \left(\frac{r_{11}\,{}^{W}X_{w1} + r_{12}\,{}^{W}Y_{w1} + r_{13}\,{}^{W}Z_{w1} + t_x}{r_{31}\,{}^{W}X_{w1} + r_{32}\,{}^{W}Y_{w1} + r_{33}\,{}^{W}Z_{w1} + t_z} \right) - {}^{I}X_{d1} \\ v_0 + \alpha_v \left(\frac{r_{21}\,{}^{W}X_{w1} + r_{22}\,{}^{W}Y_{w1} + r_{23}\,{}^{W}Z_{w1} + t_y}{r_{31}\,{}^{W}X_{w1} + r_{32}\,{}^{W}Y_{w1} + r_{33}\,{}^{W}Z_{w1} + t_z} \right) - {}^{I}Y_{d1} \\ \vdots \\ u_0 + \alpha_u \left(\frac{r_{11}\,{}^{W}X_{wn} + r_{12}\,{}^{W}Y_{wn} + r_{13}\,{}^{W}Z_{wn} + t_x}{r_{31}\,{}^{W}X_{wn} + r_{32}\,{}^{W}Y_{wn} + r_{33}\,{}^{W}Z_{wn} + t_z} \right) - {}^{I}X_{dn} \\ v_0 + \alpha_v \left(\frac{r_{21}\,{}^{W}X_{wn} + r_{22}\,{}^{W}Y_{wn} + r_{23}\,{}^{W}Z_{wn} + t_y}{r_{31}\,{}^{W}X_{wn} + r_{32}\,{}^{W}Y_{wn} + r_{33}\,{}^{W}Z_{wn} + t_z} \right) - {}^{I}Y_{dn} \end{pmatrix} \tag{100}$$

$$Q = \begin{pmatrix} -\alpha_u\,{}^{C}\hat{X}_{d1} \left({}^{C}\hat{X}_{d1}{}^{2} + {}^{C}\hat{Y}_{d1}{}^{2} \right) & -\alpha_u \left({}^{C}\hat{X}_{d1}{}^{2} + {}^{C}\hat{Y}_{d1}{}^{2} \right) \\ -\alpha_v\,{}^{C}\hat{Y}_{d1} \left({}^{C}\hat{X}_{d1}{}^{2} + {}^{C}\hat{Y}_{d1}{}^{2} \right) & 0 \\ \vdots & \vdots \\ -\alpha_u\,{}^{C}\hat{X}_{dn} \left({}^{C}\hat{X}_{dn}{}^{2} + {}^{C}\hat{Y}_{dn}{}^{2} \right) & -\alpha_u \left({}^{C}\hat{X}_{dn}{}^{2} + {}^{C}\hat{Y}_{dn}{}^{2} \right) \\ -\alpha_v\,{}^{C}\hat{Y}_{dn} \left({}^{C}\hat{X}_{dn}{}^{2} + {}^{C}\hat{Y}_{dn}{}^{2} \right) & 0 \end{pmatrix}$$
$$\begin{pmatrix} 0 & -\alpha_u\,{}^{C}\hat{X}_{d1} & -\alpha_u\,{}^{C}\hat{X}_{d1}\,{}^{C}\hat{Y}_{d1} \\ -\alpha_v \left({}^{C}\hat{X}_{d1}{}^{2} + {}^{C}\hat{Y}_{d1}{}^{2} \right) & -\alpha_v\,{}^{C}\hat{X}_{d1}\,{}^{C}\hat{Y}_{d1} & -\alpha_v\,{}^{C}\hat{Y}_{d1} \\ \vdots & \vdots & \vdots \\ 0 & -\alpha_u\,{}^{C}\hat{X}_{dn} & -\alpha_u\,{}^{C}\hat{X}_{dn}\,{}^{C}\hat{Y}_{dn} \\ -\alpha_v \left({}^{C}\hat{X}_{dn}{}^{2} + {}^{C}\hat{Y}_{dn}{}^{2} \right) & -\alpha_v\,{}^{C}\hat{X}_{dn}\,{}^{C}\hat{Y}_{dn} & -\alpha_v\,{}^{C}\hat{Y}_{dn} \end{pmatrix} \tag{101}$$

The solution for d can now be obtained by using the pseudo-inverse in the following way,

$$d = -\left(Q^{\mathrm{T}}Q\right)^{-1} Q^{\mathrm{T}}C \tag{102}$$

4 Accuracy Evaluation

The systems used to evaluate the accuracy of camera calibration can be classified in two groups. The first group is based on analyzing the discrepancy between the real position of the 3D object point with respect to the 3D position estimated from its 2D projection. The second group compares the real position in pixels of a 2D image point with the calculated projection of the 3D object point on the image plane. In the following text, some of the most frequently used methods of accuracy evaluation are described.

4.1 3D Measurement

1. *3D position obtained from stereo triangulation.* In the first step, two images are acquired from a set of 3D test points whose 3D coordinates are known. In the second, the estimated 3D coordinates of the same points are computed from their projections using the calibrated parameters. Finally, the discrepancy between real and estimated positions is compared.

2. *Radius of ambiguity in the calibrating plane.* First, a set of 3D test points, which lay on test plane and whose coordinates in the world coordinate system are known, is acquired. Second, for each image point, the calibrated model is used to project the optical ray back from the focal point through the 2D projection. The transverse of the optical ray with the test plane determines the intersection point. The distance from the 3D test point to this intersection point defines a radius of ambiguity around the 3D point.

3. *Distance with respect to the optical ray.* This method is a generalization of the previous method. In this case, the discrepancy to be measured is the distance of the 3D test points from the optical ray generated from their projections.

4. *Normalized Stereo Calibration Error* [33]. The array of pixels in an image is projected back to the scene so that each back-projected pixel covers a certain area of the object surface. This area indicates the uncertainty of the basic resolution at this distance. The orientation of the surface has been fitted to a plane which is orthogonal to the optical axis. Let the depth of this plane be equal to ${}^{C}Z_{w}$, and the row and column focal lengths be α_u and α_v. The back projection of the pixel on this plane is a rectangle of $a \times b$ size. Let the real coordinates of the i_{th} 3D object points $({}^{C}X_{wi}, {}^{C}Y_{wi}, {}^{C}Z_{wi})$ be represented in the camera coordinate system, and let its coordinates obtained by back-projecting the pixel and intersecting it with the surface plane $({}^{C}\hat{X}_{wi}, {}^{C}\hat{Y}_{wi}, {}^{C}\hat{Z}_{wi})$ be also represented in the camera coordinate system. With these givens, the Normalized Stereo Calibration Error (NSCE) is defined as,

$$\mathrm{NSCE} = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{\left({}^{C}\hat{X}_{wi} - {}^{C}X_{wi}\right)^2 + \left({}^{C}\hat{Y}_{wi} - {}^{C}Y_{wi}\right)^2}{{}^{C}\hat{Z}_{wi}{}^{2}\left(\alpha_u^{-2} + \alpha_v^{-2}\right)\Big/12}\right]^{1/2} \tag{103}$$

4.2 2D Measurement

1. *Accuracy of distorted image coordinates.* First, take an image of a set of 3D test points. Then, calculate the 2D position of each 3D point on the image plane, taking into account lens distortion. Accuracy is obtained by measuring the discrepancy between the real 2D points (obtained from image segmentation) and the estimated ones (obtained by using the camera model).

2. *Accuracy of undistorted image coordinates.* First, take an image of a set of 3D test points. Calculate the linear projection of the 3D points on the image plane, without taking lens distortion into account. Continue by determining the real 2D points through image segmentation and remove the lens distortion by using the camera model to obtain a set of undistorted points. Finally, accuracy is obtained by measuring the discrepancy between the linear projections and the undistorted points.

5 Experimental Results

Instead of using our own experimental setup, we decided to download a list of corresponding points from the well-known Tsai's Camera Calibration Software Webpage (`http://www.cs.cmu.edu/~rgw/TsaiCode.html`). Actually, results are always conditioned to the structure of the 3D points and the image processing tools used in segmentation and further points extraction. Hence, this decision was just taken to allow the scientific community to reproduce the same conditions. Then, the surveyed calibrating techniques have been implemented and their accuracy measured using the following criteria: a) Distance with respect to the optical ray; b) Normalized Stereo Calibration Error; c) Accuracy of distorted image coordinates; and d) Accuracy of undistorted image coordinates.

The two first criteria calculate the accuracy with respect to a world coordinate system. The other two calculate the discrepancy on the image plane. First, table 2 show the accuracy measured by using the first criteria and the second criteria, respectively. Note that the NSCE method is not applicable to Hall because the method of Hall does not provide the camera parameters. Second, table 4 show the results of calculating the accuracy by using the third and fourth criteria, respectively. Note that the first three calibrating methods which do not include the modelling of lens distortion (i.e. Hall, Faugeras-Toscani and iterative Faugeras-Toscani without distortion) obviously give the same accuracy with distorted and undistorted 2D points as has been considered $P_d = P_u$.

These tables show the accuracy obtained by each of the camera calibration techniques surveyed. It can be observed that the techniques, which do not model lens distortion (the first three rows in the tables) provide less accuracy than the others, which do model the lens. Moreover, the technique of Hall appears as the best undistorted lens method because it is based on computing the transformation matrix without including any constraint. The other two techniques are based on a model which imposes a determined form of the transformation matrix. This fact ill effects the calibration. However, the discrepancy between their accuracy is not significant. Furthermore, the results show that the use of an iterative algorithm does not improve the accuracy obtained by using the pseudo-

Table 1: Accuracy of 3D Coordinate Measurement.

	3D position (mm)			NSCE
	Mean	σ	Max	
Hall	0.1615	0.1028	0.5634	n/a
Faugeras	0.1811	0.1357	0.8707	0.6555
Faugeras NR[3] without distortion	0.1404	0.9412	0.0116	0.6784
Faugeras NR with distortion	0.0566	0.0307	0.1694	0.2042
Tsai	0.1236	0.0684	0.4029	0.4468
Tsai optimized	0.0565	0.0306	0.1578	0.2037
Tsai with principal point of Tsai optimized	0.0593	0.0313	0.1545	0.2137
Tsai optimized with principal point of Tsai optimized	0.0564	0.0305	0.1626	0.2033
Weng	0.0570	0.0305	0.1696	0.2064

Table 3: Accuracy of 2D Coordinate Measurement.

	2D distorted image (pix.)			2D undistorted image (pix.)		
	Mean	σ	Max	Mean	σ	Max
Hall	0.2676	0.1979	1.2701	0.2676	0.1979	1.2701
Faugeras	0.2689	0.1997	1.2377	0.2689	0.1997	1.2377
Faugeras NR without distortion	0.2770	0.2046	1.3692	0.2770	0.2046	1.3692
Faugeras NR with distortion	0.0840	0.0458	0.2603	0.0834	0.0454	0.2561
Tsai	0.1836	0.1022	0.6082	0.1824	0.1011	0.6011
Tsai optimized	0.0838	0.0457	0.2426	0.0832	0.0453	0.2386
Tsai with principal point of Tsai optimized	0.0879	0.0466	0.2277	0.0872	0.0463	0.2268
Tsai optimized with principal point of Tsai optimized	0.0836	0.0457	0.2500	0.0830	0.0454	0.2459
Weng	0.0845	0.0455	0.2608	0.0843	0.0443	0.2584

inverse in the technique of Faugeras-Toscani without distortion. This fact demonstrates that pseudo-inverse is the best approximation in undistorted models. In order to improve accuracy it has to go to lens modelling.

It can be observed from the tables that the non-linear techniques, which model lens distortion (the last 6 rows of the tables), obviously obtain better results than the undistorted techniques. However, the improvement obtained by the method of Tsai without optimization (fifth row) is not very significant because only a few parameters are iteratively optimized (i.e. f, t_z and k_1). Nevertheless, when the whole set of parameters is optimized, the method of Tsai (sixth row) shows the best accuracy obtainable despite needing more computing time. Note that accuracy is limited due to image segmentation and also that the model used always approximates the real behavior of the image sensor. However, if a real principal point is known instead of the image center approximation, the Tsai method without optimization is as accurate as any iterative method, and allows a rapid computation. Note that the use of the Tsai optimized method by using the real principal point in the initial guess does not suggest an important improvement in the obtained accuracy. Finally, the results show that any iterative method which models lens distortion provides the same accuracy without depending on the kind of modelled lens. That is, the complete method of Weng does not obtain a better accuracy than the simple iterative method of Faugeras modelling only

[3]Newton-Raphson.

radial distortion. Even so, the accuracy is slightly less due to the complexity of this model which ill effects the calibration. The modelling of a camera including a large quantity of parameters does not imply that the accuracy obtained will be better.

6 Conclusions

This article surveys some of the most frequently used calibrating techniques. Effort has been made to unify the notation among these different methods, and they have been presented in a way the reader can easily understand. We can see that the differences among these techniques are mainly in the step concerning lens modelling. Also, the transformation from camera to image coordinates is slightly different in the method proposed by Tsai.

Furthermore, a survey on accuracy evaluation has been done. The methods surveyed have been implemented and their accuracy has been analyzed. Results show that only non-linear methods obtain a 3D accuracy smaller than 0.1 mm. with a reasonable standard deviation. Moreover, the accuracy of non-linear methods on the image plane is much better than linear methods. Results show moreover that the modelling of radial distortion is quite sufficient when high accuracy is required. The use of more complicated models does not improve the accuracy significantly. It should be kept in mind that segmentation introduces a discrepancy between observable and modelled projections which poses conditions on the accuracy. Moreover, when a low accuracy is sufficient, the fast and simple method of Hall is sufficient for most applications.

When comparing the obtained results, it can be seen that a relationship exists between the different criteria. Accuracy measuring methods obtain similar results if they are relatively compared. That is, good calibrating algorithms obtain acceptable accuracy results independently from the accuracy evaluation method used. Obviously, the results only prove something already demonstrated by the authors. However, in this article the accuracy has been measured by using the same test points for all the methods so results can be reliably compared. Hence, the reader can choose one or another method depending on the accuracy required. Moreover, once the calibrating method is chosen, the reader can take equations directly from this article to use in the desired calibrating algorithm.

Appendix

This appendix synthesizes the nomenclature used to express coordinate systems and camera parameters in the article.

$\{H\}$ defines a coordinate system H, which is composed of an origin O_H and either two $\{X_H, Y_H\}$ or three $\{X_H, Y_H, Z_H\}$ axis, depending on the number of dimensions defined.

The article defines the following coordinate systems:

- $\{W\}$=$\{O_W$,X_W,Y_W,$Z_W\}$ defines the world coordinate system.
- $\{C\}$=$\{O_C$,X_C,Y_C,$Z_C\}$ defines the camera coordinate system located at the focal point O_C.

- $\{R\}$=$\{O_R,X_R,Y_R\}$ defines the retinal coordinate system located at the principal point $O_R = (u_0, v_0)$.
- $\{I\}$=$\{O_I,X_I,Y_I\}$ defines the computer image coordinate system located in the upper-left corner of the image plane.

Each point P is always related to a coordinate system. Hence, ${}^H P$ relates the point P with respect to $\{H\}$, where ${}^H P = ({}^H X,{}^H Y,{}^H Z)$. Each point can be related to any coordinate system. However, the following notations are the only ones used:

- ${}^W P_w$=(${}^W X_w$,${}^W Y_w$,${}^W Z_w$) expresses a 3D test point from the world (scene) expressed with respect to $\{W\}$.
- ${}^C P_w$=(${}^C X_w$,${}^C Y_w$,${}^C Z_w$) expresses a 3D test point from the world (scene) expressed with respect to $\{C\}$.
- ${}^C P_u$=(${}^C X_u$,${}^C Y_u$,f)=(${}^C X_u$,${}^C Y_u$) expresses the linear projection of a point ${}^C P_w$ on the image plane related to $\{C\}$, without including lens distortion.
- ${}^C P_d$=(${}^C X_d$,${}^C Y_d$,f)=(${}^C X_d$,${}^C Y_d$) expresses a 2D image point, including lens distortion, related to $\{C\}$.
- ${}^I P_d$=(${}^I X_d$,${}^I Y_d$) expresses a 2D image point related to the image coordinate system $\{I\}$, in pixels. This point is the observable point from image acquisition.

In order to distinguish a single point from a set, i.e. the set of test points, a second sub-index is used. Then, P_{ui} indicates the i-th point on a set, where $i = 1...n$.

A rigid transformation between a two coordinate system is expressed by a transformation matrix, i.e. ${}^J K_H$ expresses the coordinate system $\{H\}$ with respect to $\{J\}$. Moreover,

$$ {}^J K_H \quad = \quad \begin{pmatrix} {}^J R_H & {}^J T_H \\ 0_{1\times 3} & 1 \end{pmatrix} $$

where $R = (r_1, r_2, r_3)^{\mathrm{T}}$ expresses the orientation of $\{H\}$ measured with respect to the axis of $\{J\}$. R can also be given related to the three rotation angles, i.e. α, β and γ. Moreover, $T = (t_x, t_y, t_z)^{\mathrm{T}}$ expresses the position of the origin of $\{H\}$ with respect to $\{J\}$.

Finally, the following camera parameters are used:

- k_1 is the first coefficient of a series which models the radial lens distortion.
- g_1 up to g_4 are the coefficients which model the decentering and thin prism lens distortion.
- f is the focal distance, i.e the distance from the focal point O_C to the image plane.
- (u_0,v_0) are the two components of the principal point, i.e. the projection of O_C on the image plane.
- k_u, k_v are the two components which permit to transform a point from metric coordinates to pixels.
- α_u, α_v are defined as $\alpha_u = f\, k_u$ and $\alpha_v = f\, k_v$.

- s_x is the scale factor.
- $d'_x = d_x \frac{N_{cx}}{N_{fx}}$
- d_x, d_y are the center to center distances between adjacent sensor elements with respect to X direction and Y direction of the CCD sensor, respectively.
- N_{cx} is the number of sensor elements in the X direction of the CCD sensor.
- N_{fx} is the number of pixels in an image row as sampled by the computer.

References

[1] R. I. Hartley, *Euclidean reconstruction from uncalibrated views*, Second European Workshop on Applications of Invariance in Computer Vision, (1993), pp. 237–257.

[2] O. D. Faugeras, *Three-Dimensional Computer Vision*, The MIT Press, Cambridge, Massachusetts, 1993.

[3] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. 2, Addison-Wesley Publishing Company, 1993.

[4] R. Ahlers and J. Lu, *Stereoscopic vision - an application oriented overview*, SPIE - Optics, Illumination, and Image Sensing for Machine Vision IV, 1194 (1989), pp. 298–307.

[5] J. Batlle, E. Mouaddib, and J. Salvi, *A survey: Recent progress in coded structured light as a technique to solve the correspondence problem*, International Journal of Pattern Recognition, 31 (1998), pp. 963–982.

[6] R. A. Jarvis, *A perspective on range finding techniques for computer vision*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 5 (1983), pp. 122–139.

[7] Z. Zhang, *The matching problem: The state of the art*, Tech. Rep. 2146, Institut National de Recherche en Informatique et en Automatique, 1993.

[8] T. S. Newman, *A survey of automated visual inspection*, Image Understanding, 61 (1995), pp. 231–262.

[9] A. Casals, *Sensor Devices and Systems for Robotics*, vol. 52, Springer-Verlag. NATO ASI Series, Berlin Heidlberg, 1989.

[10] A. Broggi, *Vision-based driving assistance in vehicles of the future*, IEEE Intelligent Systems, (1998), pp. 22–23.

[11] L. Charbonnier and A. Fournier, *Heading guidance and obstacles localization for an indoor mobile robot*, IEEE International Conference on Advanced Robotics, (1995), pp. 507–513.

[12] D. Khadraoui, G. Motyl, P. Martinet, J. Gallice, and F. Chaumette, *Visual servoing in robotics scheme using a Camera/Laser-stripe sensor*, IEEE International Journal on Robotics and Automation, 12 (1996), pp. 743–750.

[13] R. K. Lenz and R. Y. Tsai, *Calibrating a cartesian robot with eye-on-hand configuration independent of eye-to-hand relationship*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 11 (1989), pp. 916–928.

[14] M. Li, *Camera calibration of a head-eye system for active vision*, European Conference on Computer Vision, (1994), pp. 543–554.

[15] M. Ito, *Robot vision modelling - camera modelling and camera calibration*, Advanced Robotics, 5 (1991), pp. 321–335.

[16] R. K. Lenz and R. Y. Tsai, *Techniques for calibration of the scale factor and image center for high accuracy 3D machine vision metrology*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 10 (1988), pp. 713–720.

[17] M. Penna, *Camera calibration: A quick and easy way to detection the scale factor*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13 (1991), pp. 1240–1245.

[18] Y. Liu, T. S. Huang, and O. D. Faugeras, *Determination of camera location from 2-d to 3-d line and point correspondences*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 12 (1990), pp. 28–37.

[19] C. C. Wang, *Extrinsic calibration of a vision sensor mounted on a robot*, IEEE Int. Journal on Robotics and Automation, 8 (1992), pp. 161–175.

[20] E. L. Hall, J. B. K. Tio, C. A. McPherson, and F. A. Sadjadi, *Measuring curved surfaces for robot vision*, Computer Journal, December (1982), pp. 42–54.

[21] J. Batista, H. Araujo, and A. T. de Almeida, *Iterative multistep explicit camera calibration*, IEEE International Journal on Robotics and Automation, 15 (1999), pp. 897–916.

[22] G.-Q. Wei and S. De Ma, *Implicit and explicit camera calibration: Theory and experiments*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 16 (1994), pp. 469–480.

[23] O. D. Faugeras and G. Toscani, *The calibration problem for stereo*, Proceedings of IEEE Computer Vision and Pattern Recognition, (1986), pp. 15–20.

[24] R. Y. Tsai, *A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses*, IEEE International Journal on Robotics and Automation, RA-3 (1987), pp. 323–344.

[25] Z. Hong and J. Yang, *An algorithm for camera calibration using a three-dimensional reference point*, Pattern Recognition, 26 (1993), pp. 1655–1660.

[26] S. Kamata, R. O. Eason, M. Tsuji, and E. Kawaguchi, *A camera calibration using 4 points targets*, International Conference on Pattern Recognition, 1 (1992), pp. 550–553.

[27] L. L. Wang and W. Tsai, *Camera calibration by vanishing lines for 3-D computer vision*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13 (1991), pp. 370–376.

[28] S. Chen and W. Tsai, *A systematic approach to analytic determination of camera parameters by line features*, Pattern Recognition, 23 (1990), pp. 859–877.

[29] T. Echigo, *A camera calibration technique using three sets of parallel lines*, Machine Vision and Applications, 3 (1990), pp. 159–167.

[30] *Manual of Photogrammetry*, American Society of Photogrammetry, 4 ed., 1980.

[31] J. Salvi, *An Approach to Coded Structured Light to Obtain Three Dimensional Information*, PhD thesis, Universitat de Girona, Departament d'Electrònica, Informàtica i Automàtica, 1997.

[32] G. Toscani, *Systèmes de Calibration et Perception du Movement en Vision Artificielle*, PhD thesis, Université Paris Sud, 1987.

[33] J. Weng, P. Cohen, and M. Herniou, *Camera calibration with distortion models and accuracy evaluation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 14 (1992), pp. 965–980.

[34] J. Salvi, J. Batlle, and E. Mouaddib, *A robust-coded pattern projection for dynamic 3D scene measurement*, International Journal of Pattern Recognition Letters, 19 (1998), pp. 1055–1065.

[35] J. Z. C. Lai, *On the sensitivity of camera calibration*, Journal of Image and Vision Computing, 11 (1993), pp. 656–664.

[36] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, 1980.

A survey addressing the fundamental matrix estimation problem

Joaquim Salvi, Xavier Armangué and Jordi Pagès

Computer Vision and Robotics Group
Institute of Informatics and Applications
University of Girona
Av. Lluís Santaló, s/n, E-17071 Girona (Spain)
{qsalvi,armangue,jpages}@eia.udg.es

IEEE Signal Processing Society
International Conference on Image Processing
ICIP 2001, Thessaloniki, Greece
October 7-10, 2001

Acceptance Letter

```
Date: Thu, 12 Apr 2001 17:24:20 -0500
From: icip2001web@securecms.com
To: armangue@eia.udg.es, jpages@eia.udg.es, qsalvi@eia.udg.es,
     qsalvi@eia.udg.es
Cc: icip2001web@securecms.com
Subject: ICIP 2001 Paper Acceptance 4429
```

Dear Dr. Armangué, Pagès, Salvi,

We are pleased to confirm that your paper: (4429) *A Survey Addressing the Fundamental Matrix Estimation Problem* has been accepted for presentation at ICIP-2001. The program committee was very selective in putting together an outstanding program for this year's ICIP. We rely on your continuing efforts, through your presentation and published paper, to maintain ICIP's tradition as the premier conference in the field of image processing.

To help ensure the quality of the Proceedings and sessions, we ask that you strictly adhere to the schedule and the procedures. Please note that for your paper to appear in the ICIP 2001 Proceedings, your camera-ready document file must be received by the staggered deadlines posted on the website (May 15-18, 2001). Papers must not exceed four pages.

Detailed instructions will be posted on the conference web site (`http://icip01.ics.forth.gr`). If the reviewers provided any comments for the authors, these are appended to the end of this message; they may help you in preparing your final paper.

All published papers are required to be presented on-site at the conference by one of the authors. As you are no doubt aware, ICIP is the most important technical conference in the field of Image Processing in the world. Your paper presentation is a critical part of making the conference a success and maintaining its high quality. If you will be unable to attend the conference and present your paper, please inform us immediately so that the conference program can be adjusted accordingly.

We look forward to working with you over the next few months on ICIP 2001. The first steps have been taken in putting together what promises to be a stimulating program; ultimately, it will be the quality of your work and presentation that will make ICIP 2001 truly noteworthy.

We look forward to seeing you in Thessaloniki!

Technical Program Co-Chairs

Billene Mercer
Conference Manager

A SURVEY ADDRESSING THE FUNDAMENTAL MATRIX ESTIMATION PROBLEM

J. Salvi, X. Armangué and J. Pagès

University of Girona.
Institute of Informatics and Applications.
Av. Lluís Santaló, s/n, E-17071 Girona (Spain)

ABSTRACT

Epipolar geometry is a key point in computer vision and the fundamental matrix estimation is the only way to compute it. This article surveys several methods of fundamental matrix estimation which have been classified into linear methods, iterative methods and robust methods. All of these methods have been programmed and their accuracy analysed using real images. A summary, accompanied with experimental results, is given and the code is available in Internet (`http://eia.udg.es/~armangue/research`).

1. INTRODUCTION

The estimation of three-dimensional information in active systems is a crucial problem in computer vision because the camera parameters may change dynamically depending on the scene. In such situations, only epipolar geometry, which is contained in the fundamental matrix, can be computed. Basically, the intrinsic parameters of both cameras and the position and orientation of one in relation to the other can be extracted by using Kruppa equations [1]. Moreover, the fundamental matrix can be used to reduce the matching process among the viewpoints [2], therefore, it is very interesting to develop accurate techniques to compute it.

This article surveys fifteen of the most frequently used techniques in computing the fundamental matrix and is organized as follows. First, a brief introduction of epipolar geometry is presented. Then, all the techniques to estimate **F** are presented describing their advantages and drawbacks as opposed to previous ones. Section three deals with the experimental results obtained with real images and finally, the article ends with conclusions.

2. EPIPOLAR GEOMETRY

Consider a 3D object point M expressed with respect to a world coordinate system $M = ({}^{W}X,\, {}^{W}Y,\, {}^{W}Z,\, 1)^{\mathrm{T}}$ and its 2D projection m on the image plane in pixels $m = ({}^{I}X,\, {}^{I}Y)^{\mathrm{T}}$. Both points are related with a projective transformation matrix ${}^{I}\mathbf{P}_{W}$, that is $s\, m = {}^{I}\mathbf{P}_{W}\, M$, where s is a scale factor. The ${}^{I}\mathbf{P}_{W}$ matrix can be broken down into ${}^{I}\mathbf{P}_{W} = {}^{I}\mathbf{A}_{C}\, {}^{C}\mathbf{K}_{W}$, where ${}^{I}\mathbf{A}_{C}$ is a 3×4 matrix which relates the metric camera coordinate system C located at the focal point O_C with the image coordinate system I located at the north-west corner of the image plane in pixels, and ${}^{C}\mathbf{K}_{W}$ is a 4×4 matrix, see equation (1), which relates the camera coordinate system $\{\mathrm{C}\}$ with the world coordinate system W.

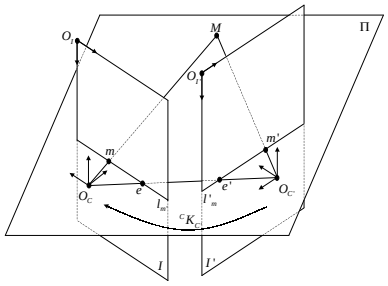


Fig. 1. The geometric relation between two cameras.

$$ {}^{C}\mathbf{K}_{W} = \begin{pmatrix} {}^{C}\mathbf{R}_{W} & {}^{C}t_{W} \\ 0_{1x3} & 1 \end{pmatrix} \qquad (1) $$

Then, the epipolar geometry concerns the relationship between both cameras of a stereoscopic system. Given the object point M and its 2D projections m and m' on both image planes, these 3 points define a plane Π which intersects both image planes at the epipolar lines $l_{m'}$ and l'_{m} respectively, as shown in Figure 1. Note that the same plane Π can be computed using both focal points O_C and $O_{C'}$ and a single 2D projection (m or m'), which is the principle to reduce the correspondence problem to a single scanning along the epipolar line. Moreover, the intersection of all the epipolar lines define an epipole on both image planes, which can also be extracted by intersecting the line defined from both focal points O_C and $O_{C'}$ with both image planes. All the epipolar

Work funded by Spanish project CICYT TAP99-0443-C05-01

geometry is contained in the so called Fundamental matrix $\mathbf{F}$, where $m^{\mathrm{T}}\mathbf{F}m' = 0$.

The fundamental matrix contains the intrinsic parameters of both cameras and the rigid transformation between both cameras which depends on the camera that has been considered as the origin of the world coordinate system. In this article, the origin of the world coordinate system coincides with the coordinate system of the second camera, located at $O_{C'}$, as shown in equation (2).

$$\mathbf{F} = {}^{I}\mathbf{A}_{C}{}^{-\mathrm{T}}[{}^{C}t_{C'}]_{\times}{}^{C}\mathbf{R}_{C'}{}^{I'}\mathbf{A}_{C'}{}^{-1} \quad (2)$$

3. ESTIMATING THE FUNDAMENTAL MATRIX

In the last few years, several methods to estimate the fundamental matrix have been proposed and can be classified into lineal methods and iterative methods. These deal with bad point localization due to noise in image segmentation and robust techniques whitch eliminate the outliers due to false matchings.

3.1. Linear methods

The linear method of *seven points* is based on computing the fundamental matrix by using only seven point correspondences [3]. Due to the homogeneity of the equations, the solution is a set of matrices of the form $\mathbf{F} = \alpha\mathbf{F}_1 + (1-\alpha)\mathbf{F}_2$. Then, by forcing the rank of the matrix to 2 and using the expression $\det[\alpha\mathbf{F}_1 + (1-\alpha)\mathbf{F}_2]$, a cubic polynomial is obtained, which has to be solved to obtain α and therefore $\mathbf{F}$. The main advantage of this method is that a fundamental matrix can be estimated by using only seven points, but this fact becomes a drawback when some points are badly located. Moreover, the *7-points method* cannot be applied in the presence of redundancy. Hence, it can not be applied using n points if $n > 7$.

Another interesting method is *8-points method* in witch redundancy of points permits minimizes the error of estimating $\mathbf{F}$. The minimizing equation is $\min_{\mathbf{F}} \sum_i \left(m_i^{\mathrm{T}}\mathbf{F}m_i'\right)^2$. The classical method of solving such equation is the *least-squares technique* which forces one of the components of $\mathbf{F}$ to be the unity [4]. This simplification can be assumed because $\mathbf{F}$ is always defined using a scale factor. Then the system to solve is $f' = (\mathbf{U'}^{\mathrm{T}}\mathbf{U'})^{-1}\mathbf{U'}^{\mathrm{T}}c_9$ in which $\mathbf{U}'$ is a matrix containing the first eight columns of $\mathbf{U}$, c_9 is the last column of $\mathbf{U}$ and f' is a vector containing the first eight elements of f. Note that the ninth element of f is 1. A variant of the *8-points method* can be applied if the equation is solved by using *eigen analysis*, also called *orthogonal least-squares technique* [5]. In this case, $\mathbf{F}$ can be determined from the eigen vector corresponding to the smallest eigen value of $\mathbf{U}^{\mathrm{T}}\mathbf{U}$. The difference between this method and the classical *least-squares* resides in the form of calculating the error between correspondences and epipolar lines in which an orthogonal distance to the epipolar line is much more realistic.

The last linear method we surveyed is the *analytic method with rank-2 constraint* [3] which imposes the rank-2 constraint during the minimization. The matrix $\mathbf{U}'$ is defined as the composition of the first seven columns of $\mathbf{U}$ and c_8 and c_9 are defined as the eighth and ninth columns of $\mathbf{U}$ respectively so that $\mathbf{F}$ can be computed as

$$f' = -f_8(\mathbf{U'}^{\mathrm{T}}\mathbf{U}')^{-1}\mathbf{U'}^{\mathrm{T}}c_8 - f_9(\mathbf{U'}^{\mathrm{T}}\mathbf{U}')^{-1}\mathbf{U'}^{\mathrm{T}}c_9 \quad (3)$$

in which f' is the vector containing the first seven elements of f, and f_8 and f_9 are the eighth and ninth elements of f, respectively. In order to obtain the values of f_8 and f_9, an $\mathbf{F}$ is computed by using the *seven points algorithm*. Then, f is computed by selecting from the various pairs of $\mathbf{F}$ the one which minimizes $\|f\| = 1$. This method provides a rank-2 matrix. However, the *analytic method with rank-2 constraint* does not improve the results of the previously explained methods to any great extent.

The linear methods are very fast but their accuracy is rather poor in the presence of noise. In order to obtain better results the iterative algorithms have to be considered.

3.2. Iterative methods

The iterative methods can be classified into two groups. The first group of techniques is based on minimizing the distances between points and epipolar lines, that is

$$\min_{\mathbf{F}} \sum_i \left(d^2(m_i, \mathbf{F}m_i') + d^2(m_i', \mathbf{F}m_i)\right) \quad (4)$$

A first approach consists of directly applying an iterative method as *Newton-Raphson* [6]. Another possibility is the *iterative linear method* [3] which is based on computing the weight value w_i equivalent to the epipolar distances by using the previous $\mathbf{F}$ (in the first iteration $w_i = 1$) and then minimize by using *least-squares* in each iteration. Neither approach imposes the rank-2 constraint. The *nonlinear minimization in parameter space* [3] can solve this situation. This method is based on parameterizing the fundamental matrix keeping in mind that it must have rank 2 by fixing just one of the multiple parameterizations. The iteration of this method permits computing a better rank-2 $\mathbf{F}$. However, obtaining a good estimation alone is not enough because the variance of points are not analogous and the least-square technique assumes they are comparable. In order to overcome this drawback, the second group of methods have to be considered.

The second group of methods are based on the *gradient-technique* [7]. In this case, the equation to solve is

$$\min_{\mathbf{F}} \sum_i \left(m_i^{\mathrm{T}}\mathbf{F}m_i'\right)^2 / g_i^2 \quad (5)$$

in which $g_i = ({l_1}^2 + {l_2}^2 + {l'_1}^2 + {l'_2}^2)^{1/2}$.

The *gradient-based* technique obtains better results compared to linear methods and the iterative methods of the first group. Although iterative methods are more accurate than linear ones, they are also time consuming and can not eliminate the potential outliers. Hence, robust methods have to be used in the presence of outliers.

3.3. Robust methods

This paper surveys three robust methods: *M-Estimators*, *Least-Median-Squares* (LMedS) and *Random Sampling* (RANSAC), which can be used in the presence of either outliers or bad point localization.

The *M-estimators* [7] try to reduce the effect of outliers by weighting the residual of each point. A lot of different weight functions have been proposed and each one gives a new variant of the M-estimator method. The results obtained are quite good in the presence of outliers but they are rather bad if the points are badly located. The *LMedS* [3] and *RANSAC* [5] techniques are very similar. First, both techniques are based on a random selection of a set of points which are then used to compute $\mathbf{F}$ by using a linear method. The difference between these techniques is the way they determining the best $\mathbf{F}$. The *LMedS* calculates the median of distances between points and epipolar lines for each $\mathbf{F}$. The chosen fundamental matrix has to minimize such a median. The *RANSAC* calculates the number of inliers for each $\mathbf{F}$ and the chosen $\mathbf{F}$ is the one which maximizes it. Once the outliers are eliminated, the $\mathbf{F}$ is recalculated with the aim of obtaining a better approach. Another difference is that *LMedS* is more restrictive than *RANSAC* so that it eliminates more points. However, the main constraint of both techniques is their lack of repetitivity due to the aleatory way of selecting the points.

3.4. Considerations in F estimation

Data normalization is a key point in fundamental matrix estimation. It has been proved that the computation should not be applied directly to the raw data in pixels due to potential uncertainties when using by huge numbers. Basically, there are two different methods of data normalization. The first method [3] normalizes the data between $[-1, 1]$. The second, proposed by Hartley [8], is based on two transformations. In the first, the points are translated so that their centroid is placed at the origin. In the second, the points are scaled so that the mean of the distances of the points to the origin is $\sqrt{2}$. It has been proved that the method proposed by Hartley gives more accurate results than the previous one.

Another interesting fact is that the estimated $\mathbf{F}$ should be a rank-2 matrix in order to model the epipolar geometry with all the epipolar lines intersecting in the epipole. Although the rank-2 constraint is not imposed in all the surveyed methods, there is a mathematical method which transforms a rank-n square matrix to the closest rank-$(n-1)$ matrix [7]. However, the obtained rank-2 $\mathbf{F}$ give worse results because it has not been optimized. In this case, we propose to using any method which imposes a rank-2 matrix in the computation of $\mathbf{F}$ instead of further transforming it.

4. EXPERIMENTAL RESULTS

The surveyed methods have been programmed and their accuracy analyzed with synthetic and real data, such as underwater images from the seabed obtained by our underwater robot GARBI. Image points have been normalized by using Hartley [8] explained in section two. Table 1 shows the accuracy of each method as the mean and standard deviation of the distances between points and epipolar lines.

The accuracy of the *seven points algorithm* extremely depends on the seven points used. The *least-squares technique* depends inversely on the amount of bad-located points, obtaining usually better results by increasing the amount of points. The *eigen analysis* is the linear method that obtains the best results because an orthogonal least-squares minimization is more realistic than the classic least-squares. However, all these methods obtain a rank-3 matrix, which means that the epipolar geometry is not properly modeled.

The *analytic method with rank-2 constraint* obtains a rank-2 fundamental matrix. However, the distances between points and epipolar lines are worse than in the linear methods. The *iterative linear method* improves considerably the least-squares technique but can not cope with the outliers problem. The *iterative Newton-Raphson algorithm* obtains even better results than the previous method if there is no outliers present. Although the *nonlinear minimization in parameter space* obtains also a rank-2 matrix, but his computational cost is very high. The eighth and ninth methods are two versions of the *gradient-based method* using least-squares and orthogonal least-squares, respectively. Both methods obtain better results than their equivalent linear methods. Furthermore, the eigen analysis, once again, obtains better results than the other linear methods. Although some of these methods obtain a rank-2 matrix, they can not cope with outliers.

The last surveyed methods are known as "robust", which means they might detect and remove the outliers and compute the fundamental matrix using only the inliers. Three versions of the *M-estimators* have been programmed using least-squares, eigen analysis and the method proposed by Torr [5], respectively. The three methods use a linear initial guess and they become really dependent on the linear method used to estimate it. The following two methods are two versions of LMedS using again least-squares and eigen analysis, respectively. Although the accuracy of

Table 1. Methods Implemented with mean and std. of error: 1.- seven points; 2.- least-squares (LS) 3.- orthogonal LS; 4.- rank-2 constraint; 5.- iterative lineal using LS; 6.- iterative Newton-Raphson using LS; 7.- minimization in parameter space using eigen; 8.- gradient using LS; 9.- gradient using eigen; 10.- M-Estimator using LS; 11.- M-Estimator using eigen; 12.- M-Estimator proposed by Torr; 13.- LMedS using LS; 14.- LMedS using eigen; 15.- RANSAC using eigen.

Methods	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
$\sigma = 0.0$	0.000	0.000	0.000	0.102	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
outliers 0%	0.000	0.000	0.000	0.043	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$\sigma = 0.0$	22.125	339.562	17.124	30.027	161.684	27.035	17.871	187.474	18.224	73.403	4.909	4.714	0.000	0.000	9.449
outliers 10%	57.007	433.013	31.204	59.471	117.494	59.117	31.225	197.049	36.141	60.443	4.493	2.994	0.000	0.000	8.387
$\sigma = 0.1$	15.048	1.331	0.107	0.120	1.328	0.108	0.112	1.328	0.112	0.355	0.062	0.062	1.331	0.107	0.107
outliers 0%	14.498	0.788	0.088	0.091	0.786	0.088	0.092	0.786	0.092	0.257	0.042	0.041	0.788	0.088	0.088
$\sigma = 0.1$	26.136	476.841	19.675	50.053	158.671	70.530	19.549	183.961	15.807	73.354	4.876	4.130	0.449	0.098	9.148
outliers 10%	66.095	762.756	46.505	53.974	124.086	91.194	46.537	137.294	40.301	59.072	4.808	2.997	0.271	0.077	8.564
$\sigma = 0.5$	15.783	5.548	0.538	0.642	5.599	0.538	0.554	5.590	0.554	2.062	0.392	0.367	5.548	0.538	0.538
outliers 0%	14.837	3.386	0.362	0.528	3.416	0.366	0.361	3.410	0.361	1.466	0.237	0.207	3.386	0.362	0.362
$\sigma = 0.5$	117.534	507.653	19.262	26.475	161.210	47.884	18.933	217.577	19.409	143.442	3.887	3.147	47.418	0.586	10.723
outliers 10%	94.987	940.808	49.243	54.067	136.828	65.975	49.204	368.061	51.154	111.694	3.969	2.883	29.912	0.434	12.972
$\sigma = 1.0$	19.885	21.275	1.065	1.319	20.757	1.064	1.071	21.234	1.071	8.538	0.794	0.814	21.275	1.065	1.065
outliers 0%	16.485	12.747	0.744	0.912	12.467	0.747	0.745	12.719	0.745	6.306	0.463	0.463	12.747	0.744	0.744
$\sigma = 1.0$	138.554	629.326	21.264	61.206	158.849	79.323	20.277	152.906	18.730	120.012	3.921	4.089	25.759	1.052	8.657
outliers 10%	96.671	833.019	53.481	64.583	120.461	80.100	49.476	120.827	38.644	122.436	3.752	4.326	15.217	0.803	17.410
Real	3.833	4.683	1.725	5.242	3.068	2.584	1.643	2.949	1.581	0.557	0.650	0.475	1.485	1.039	1.725
Image	4.440	3.941	2.138	4.286	2.804	4.768	2.109	2.798	2.056	0.441	0.629	0.368	1.134	0.821	2.138

LMedS seems worse than the one given by *M-estimators*, *LMedS* removes the outliers much more correctly since the epipolar geometry is better modeled. The *RANSAC* is the last surveyed method, which does not obtain better results than *LMedS* with eigen analysis because the method is too permissive selecting the outliers.

5. CONCLUSIONS

The objective of this article is a comparative survey of fifteen of the most frequently used methods in fundamental matrix estimation. The different methods have been programmed and their accuracy analyzed with real images. Experimental results show that: a) linear methods are quite good if the points are well located in the image and the correspondence problem previously solved; b) iterative methods can cope with some gaussian noise in the localization of points, but they become quite inefficient in the presence of outliers; c) robust methods can cope with both discrepancy in the localization of points and false matchings.

The experimental results point out that the orthogonal least-squares using eigen analysis gives better results than the classic least-squares technique of minimization. Moreover, a rank-2 method is preferred because it models the epipolar geometry with all the epipolar lines intersecting at the epipole. Finally, experimental results show that the corresponding points have to be normalized and the best results using this sort of method have been obtained by using the method proposed by Hartley [7]. In conclusion, the best results were obtained with the *LMedS* method forcing the matrix to be rank-2 once the outliers have been removed.

6. REFERENCES

[1] Richard I. Hartley, "Kruppa's equations derived from the fundamental matrix.," *Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 133–135, 1997.

[2] Zhengyou Zhang, "The matching problem: The state of the art," Tech. Rep. 2146, Institut National de Recherche en Informatique et en Automatique, 1993.

[3] Zhengyou Zhang, "Determining the epipolar geometry and its uncertainty: A review," Tech. Rep. 2927, Institut National de Recherche en Informatique et en Automatique, July 1996.

[4] Quan-Tuan Luong and Olivier D. Faugeras, "The fundamental matrix: Theory, algorithms, and stability analysis," *International Journal of Computer Vision*, vol. 17, no. 1, pp. 43–75, January 1996.

[5] P. H. S. Torr and D. W. Murray, "The developement and comparision of robust methods for estimating the fundamental matrix," *International Journal Computer Vision*, vol. 24, no. 3, pp. 271–300, September 1997.

[6] Joaquim Salvi, *An Approach to Coded Structured Light to Obtain Three Dimensional Information*, Ph.D. thesis, Universitat de Girona, Departament d'Electrònica, Informàtica i Automàtica, 1997.

[7] Richard Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[8] Richard Hartley, "In defence of the 8-point algorithm," in *Proceedings of the 5th International Conference on Computer Vision*, Boston, 1995, pp. 1064–1070, IEEE Computer Society Press.

Comparative survey on estimating the fundamental matrix

Xavier Armangué, Jordi Pagès, Joaquim Salvi and Joan Batlle

Computer Vision and Robotics Group
Institute of Informatics and Applications
University of Girona
Av. Lluís Santaló, s/n, E-17071 Girona (Spain)
{armangue,jpages,qsalvi,jbatlle}@eia.udg.es

Associación Española de Reconocimiento de Formas y Análisis de Imágenes
IX Simposium Nacional de Reconocimiento de Formas y
Análisis de Imágenes
SNRFAI 2001, Castelló, Spain
May 14-18, 2001

Acceptance Letter

Assumpte: SNRFAI 2001 Review results
Data: Thu, 22 Feb 2001 20:08:27 +0100
De: Congreso SNRFAI2001 <info@snrfai2001.uji.es>
A: undisclosed-recipients:;

Benicàssim (Castelló)
May 14-18,2001

Castelló, 15th Febraury 2001

Dear authors:

We are pleased to inform you that according to the reviewing result, your paper entitled:

Title: *Comparative Survey on Fundamental Matrix Estimation*
Code: 0161

has been accepted for Oral presentation at SNRFAI 2001.

If the reviewers have made suggestions for improvement or changes to your submission, you will find their comments attached. Please, take into account their suggestions when preparing the final submission. The document style files are available from the conference web site. Bear in mind that papers have to be in camera-ready format by March 15, 2001. Papers must be sent in postscript format by electronic mail or through the conference web page.

Please, note that, in order for a paper to be published in the conference proceedings, at least one registration (with payment or proof of payment) per paper is required to arrive to the SNRFAI 2001 Secretariat by April 15, 2001. Conference registration, accommodation, and travel information is available from the conference web site.

Thank you very much for your participation, and we look forward to seeing you in Benicàssim at the SNRFAI 2001.

Sincerely yours

SNRFAI'2001 Organizing Committee
Universitat Jaume I
E-12071 Castelló - Spain.

Telephone: +34 964 728348 / +34 964 728350
Fax: +34 964 728435
E-mail: info@snrfai2001.uji.es

http://www.snrfai2001.uji.es

Comparative Survey on Fundamental Matrix Estimation*

X. Armangué, J. Pagès, J. Salvi and J. Batlle
Universitat de Girona.
Institut de Informàtica i Aplicacions.
Av. Lluís Santaló, s/n, E-17071 Girona (Spain)

Abstract

The epipolar geometry is a key point in computer vision and the fundamental matrix estimation the unique way to compute it. This article surveys several methods of fundamental matrix estimation. All the different methods have been programmed and their accuracy analysed using synthetic and real images. A discussion justified with experimental results is given and the code is available in Internet (`http://eia.udg.es/~armangue/research`).

Keywords: Epipolar Geometry, Fundamental Matrix, Performance Evaluation.

1 Introduction

The estimation of three-dimensional information in active systems is a crucial problem in computer vision because the camera parameters might change dynamically depending on the scene. In such situation, only the epipolar geometry, which is contained in the fundamental matrix, can be computed. Basically, the intrinsic parameters of both cameras and the position and orientation of one from the other can be extracted by using Kruppa equations [1]. Moreover, the fundamental matrix can be used to reduce the matching process among the viewpoints [2]. Thus, it is very interesting to develop accurate techniques to compute it. This article surveys fifteen of the most used techniques in computing the fundamental matrix.

The article is organized as follows. First, section two describes all the techniques in a sequence analyzing their advantages and drawbacks with the previous ones. Then, section three deals with the experimental results obtained by using synthetic and real data. The article ends with conclusions.

*Work funded by Spanish project CICYT TAP99-0443-C05-01

2 Estimating the fundamental matrix

In the last few years, several methods to estimate the fundamental matrix have been proposed, which can be classified into lineal methods and iterative methods that deal with bad point localization due to noise in image segmentation, and robust techniques that eliminate the outliers due to false matchings.

Linear methods:

The linear method of the *seven points* is based on computing the fundamental matrix by using only seven point correspondences [3]. Due to the homogeneity of the equations, the solution is a set of matrices of the form $\mathbf{F} = \alpha\mathbf{F}_1 + (1-\alpha)\mathbf{F}_2$. Then, by forcing the rank of the matrix to 2 and using the expression $\det[\alpha\mathbf{F}_1 + (1-\alpha)\mathbf{F}_2]$ a cubic polynomial is obtained, which has to be solved to obtain α and therefore $\mathbf{F}$. The main advantage of this method is that a fundamental matrix can be estimated by using only seven points, but this fact becomes a drawback when some points are bad located. Moreover, the *7-points method* cannot be applied in the presence of redundancy. Hence, it can not be applied using n points if $n > 7$.

Another interesting method is the *8-points method*, where the redundancy of points permits to minimize the error on estimating $\mathbf{F}$. The equation to minimize is: $\min_{\mathbf{F}} \sum_i \left(m_i^{\mathrm{T}}\mathbf{F}m_i'\right)^2$. The most classical method to solve such equation is the *least-squares technique* by forcing one of the components of $\mathbf{F}$ to be the unity [4]. This simplification can be assumed because $\mathbf{F}$ is always defined up to a scale factor. Then the system to solve is: $f' = (\mathbf{U'}^{\mathrm{T}}\mathbf{U'})^{-1}\mathbf{U'}^{\mathrm{T}}c_9$, where $\mathbf{U'}$ is a matrix containing the first eight columns of $\mathbf{U}$, c_9 is the last column of $\mathbf{U}$ and f' is a vector containing the first eight elements of f. Note that the last element of f is 1. A variant of the *8-points method* can be applied if the equation is solved by using *eigen analysis*, also called *orthogonal least-squares technique* [5]. In this case $\mathbf{F}$ can be determined from the eigen vector corresponding to the smallest eigen value of $\mathbf{U}^{\mathrm{T}}\mathbf{U}$. The difference between this method and the classical *least-squares* resides in the form of calculating the error between correspondences and epipolar lines, where an orthogonal distance to the epipolar line is much more realistic.

The last linear method we surveyed is the *analytic method with rank-2 constraint* [3], which imposes the rank-2 constraint during minimization. The matrix $\mathbf{U'}$ is defined as the composition of the first seven columns of $\mathbf{U}$ and c_8 and c_9 are defined as the eighth and ninth columns of $\mathbf{U}$ respectively, so that $\mathbf{F}$ can be computed as $f' = -f_8(\mathbf{U'}^{\mathrm{T}}\mathbf{U'})^{-1}\mathbf{U'}^{\mathrm{T}}c_8 - f_9(\mathbf{U'}^{\mathrm{T}}\mathbf{U'})^{-1}\mathbf{U'}^{\mathrm{T}}c_9$, where f' is the vector containing the first seven elements of f, and f_8 and f_9 are the eighth and ninth elements of f, respectively. In order to obtain the values of f_8 and f_9, a $\mathbf{F}$ is computed by using the *seven points algorithm*. Then, f is computed by selecting for any choices of

pairs of $\mathbf{F}$ the one that minimizes $\|f\| = 1$. This method obtains a rank-2 matrix. However, the *analytic method with rank-2 constraint* does not improve considerably the results of the previously explained methods.

The linear methods are very fast but their accuracy is rather poor in the presence of noise. Better results might be obtain using iterative algorithms.

Iterative methods:

The iterative methods can be classified into two groups. The first group of techniques is based on minimizing the distances between points and epipolar lines, that is $\min_{\mathbf{F}} \sum_i \left(d^2(m_i, \mathbf{F}m'_i) + d^2(m'_i, \mathbf{F}m_i)\right)$.

A first approach consists in applying directly an iterative method as *Newton-Raphson* [6] using *least-squares technique* as initial solution. Another possibility is the *iterative linear method* [3] that is based on computing a weight value w_i equivalent to the epipolar distances by using the previous $\mathbf{F}$ (in the first iteration $w_i = 1$) and then minimize by using *least-squares* in each iteration. Both approaches do not impose the rank-2 constraint. Then, the *nonlinear minimization in parameter space* [3] can solve this situation. This method is based on parameterizing the fundamental matrix keeping in mind that it has rank 2 by fixing just one of the multiple parameterizations. The iteration of this method permits to compute a better rank-2 $\mathbf{F}$. However, it is not enough to obtain a good estimation because the variance of points are not analogous and the least-square technique assume they are comparable. In order to overcome this drawback, the second group of methods have to be considered, which are based on the *gradient-technique* [7]. In this case, the equation to solve is $\min_{\mathbf{F}} \sum_i \left(m_i^{\mathrm{T}}\mathbf{F}m'_i\right)^2 / g_i^2$, where $g_i = ({l_1}^2 + {l_2}^2 + {l'_1}^2 + {l'_2}^2)^{1/2}$. The *gradient-based* technique obtains better results compared to linear methods and the iterative methods of the first group. Although iterative methods are more accurate than linear ones, they are also hard time consuming and they can not eliminate potential outliers. Hence, robust methods have to be considered.

Robust methods:

This paper surveys up to three robust methods: *M-Estimators*, *Least-Median-Squares* (LMedS) and *Random Sampling* (RANSAC), which can be used in the presence of either outliers and bad point localization.

The *M-estimators* [7] tries to reduce the effect of outliers weighting the residual of each point. A lot of different weight functions have been proposed and each one gives a new variant of the M-estimator method. The results obtained are quite good in the presence of outliers, but they are rather bad if the points are bad located. The *LMedS* [3] and *RANSAC* [5] techniques are very similar. First, both techniques are based on selection randomly the set of points that are used to compute $\mathbf{F}$ by using a

linear method. The difference between both techniques is in the way of determining the best **F**. The LMedS calculates for each **F** the median of distances between points and epipolar lines and the chosen fundamental matrix has to minimize such a median. The RANSAC calculates for each **F** the number of inliers and the chosen **F** is the one that maximizes it. Once the outliers are eliminated, the **F** is recalculated with the aim of obtaining a better approach. Moreover, another difference is that LMedS is more restrictive than RANSAC, so that it eliminates more points. However, the main constraint of both techniques is their lack of repetitivity due to the aleatory way of selecting the points.

Considerations in F estimation:

Data normalization is a key point in fundamental matrix estimation. It has been proved that the computation should not be applied directly to the raw data in pixels due to potential uncertainties given by huge numbers. Basically, there are two different methods of data normalization. The first method [3] normalize the data between $[-1, 1]$. The second was proposed by Hartley [8] and it is based on two transformations: a) First, the points are translated so that their centroid is placed at the origin; Secondly, the points are scaled so that the mean of the distances of the points to the origin is $\sqrt{2}$. It has been proved that the method proposed by Hartley gives more accurate results than the previous one.

Another interesting fact is that the estimated **F** should be a rank-2 matrix in order to model the epipolar geometry with all the epipolar lines intersecting in the epipole. Although the rank-2 constraint is not imposed in all the surveyed methods, there is a mathematical method that transforms a rank-n square matrix to the closest rank-$(n-1)$ matrix [7]. However, the obtained rank-2 **F** give worse results because it has not been optimized. Then, we propose to use any method which impose a rank-2 matrix in the computation of **F** instead of further transforming it.

3 Experimental Results

The surveyed methods have been programmed and their accuracy analyzed with synthetic and real data, such as underwater images from the seabed obtained by our underwater robot GARBI. Image points have been normalized by using Hartley [8] explained in section two. Table 1 shows the accuracy of each method as the mean and standard deviation of the distances between points and epipolar lines.

The accuracy of the *seven points algorithm* extremely depends on the seven points used. The *least-squares technique* depends inversely on the amount of bad-located points, obtaining usually better results by increasing the amount of points. The *eigen analysis* is the linear method that obtains the best results because an

Table 1: Methods Implemented with mean and std. of error: 1.- seven points; 2.- least-squares (LS) 3.- orthogonal LS; 4.- rank-2 constraint; 5.- iterative lineal using LS; 6.- iterative Newton-Raphson using LS; 7.- minimization in parameter space using eigen; 8.- gradient using LS; 9.- gradient using eigen; 10.- M-Estimator using LS; 11.- M-Estimator using eigen; 12.- M-Estimator proposed by Torr; 13.- LMedS using LS; 14.- LMedS using eigen; 15.- RANSAC using eigen.

Methods	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
$\sigma = 0.0$	0.000	0.000	0.000	0.102	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
outliers 0%	0.000	0.000	0.000	0.043	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$\sigma = 0.0$	22.125	339.562	17.124	30.027	161.684	27.035	17.871	187.474	18.224	73.403	4.909	4.714	0.000	0.000	9.449
outliers 10%	57.007	433.013	31.204	59.471	117.494	59.117	31.225	197.049	36.141	60.443	4.493	2.994	0.000	0.000	8.387
$\sigma = 0.1$	15.048	1.331	0.107	0.120	1.328	0.108	0.112	1.328	0.112	0.355	0.062	0.062	1.331	0.107	0.107
outliers 0%	14.498	0.788	0.088	0.091	0.786	0.088	0.092	0.786	0.092	0.257	0.042	0.041	0.788	0.088	0.088
$\sigma = 0.1$	26.136	476.841	19.675	50.053	158.671	70.530	19.549	183.961	15.807	73.354	4.876	4.130	0.449	0.098	9.148
outliers 10%	66.095	762.756	46.505	53.974	124.086	91.194	46.537	137.294	40.301	59.072	4.808	2.997	0.271	0.077	8.564
$\sigma = 0.5$	15.783	5.548	0.538	0.642	5.599	0.538	0.554	5.590	0.554	2.062	0.392	0.367	5.548	0.538	0.538
outliers 0%	14.837	3.386	0.362	0.528	3.416	0.366	0.361	3.410	0.361	1.466	0.237	0.207	3.386	0.362	0.362
$\sigma = 0.5$	117.534	507.653	19.262	26.475	161.210	47.884	18.933	217.577	19.409	143.442	3.887	3.147	47.418	0.586	10.723
outliers 10%	94.987	940.808	49.243	54.067	136.828	65.975	49.204	368.061	51.154	111.694	3.969	2.883	29.912	0.434	12.972
$\sigma = 1.0$	19.885	21.275	1.065	1.319	20.757	1.064	1.071	21.234	1.071	8.538	0.794	0.814	21.275	1.065	1.065
outliers 0%	16.485	12.747	0.744	0.912	12.467	0.747	0.745	12.719	0.745	6.306	0.463	0.463	12.747	0.744	0.744
$\sigma = 1.0$	138.554	629.326	21.264	61.206	158.849	79.323	20.277	152.906	18.730	120.012	3.921	4.089	25.759	1.052	8.657
outliers 10%	96.671	833.019	53.481	64.583	120.461	80.100	49.476	120.827	38.644	122.436	3.752	4.326	15.217	0.803	17.410
Real	3.833	4.683	1.725	5.242	3.068	2.584	1.643	2.949	1.581	0.557	0.650	0.475	1.485	1.039	1.725
Image	4.440	3.941	2.138	4.286	2.804	4.768	2.109	2.798	2.056	0.441	0.629	0.368	1.134	0.821	2.138

orthogonal least-squares minimization is more realistic than the classic least-squares. However, all these methods obtain a rank-3 matrix, which means that the epipolar geometry is not properly modeled.

The *analytic method with rank-2 constraint* obtains a rank-2 fundamental matrix. However, the distances between points and epipolar lines are worse than in the linear methods. The *iterative linear method* improves considerably the least-squares technique but can not cope with the outliers problem. The *iterative Newton-Raphson algorithm* obtains even better results than the previous method if there is no outliers present. Although the *nonlinear minimization in parameter space* obtains also a rank-2 matrix, but his computational cost is very high. The eighth and ninth methods are two versions of the *gradient-based method* using least-squares and orthogonal least-squares, respectively. Both methods obtain better results than their equivalent linear methods. Furthermore, the eigen analysis, once again, obtains better results than the other linear methods. Although some of these methods obtain a rank-2 matrix, they can not cope with outliers.

The last surveyed methods are known as "robust", which means they might detect and remove the outliers and compute the fundamental matrix using only the inliers. Three versions of the *M-estimators* have been programmed using least-squares, eigen analysis and the method proposed by Torr [5], respectively. The three methods use a linear initial guess and they become really dependent on the linear method used to estimate it. The following two methods are two versions of LMedS using again least-squares and eigen analysis, respectively. Although the accuracy of *LMedS* seems worse than the one given by *M-estimators*, *LMedS* removes the outliers

much more correctly since the epipolar geometry is better modeled. The *RANSAC* is the last surveyed method, which does not obtain better results than *LMedS* with eigen analysis because the method is too permissive selecting the outliers.

4 Conclusions

This article surveys up to fifteen of the most used methods in fundamental matrix estimation. The different methods have been programmed and their accuracy analyzed with synthetic and real images. Experimental results show that: a) linear methods are quite good if the points are well located in the image and the correspondence problem previously solved; b) iterative methods can cope with some gaussian noise in the localization of points, but they become really inefficient in the presence of outliers; c) robust methods can cope with both discrepancy in the localization of points and false matchings.

The experimental results brings out that the orthogonal least-squares using eigen analysis gives better results than the classic least-squares technique of minimization. Moreover, a rank-2 method is preferred because it models the epipolar geometry with all the epipolar lines intersecting at the epipole. Finally, experimental results show up that the corresponding points have to be normalized and the better results have been obtained by using the method proposed by Hartley [7]. Concluding, the best results were obtained with the *LMedS* method forcing the matrix to be rank-2 once the outliers have been removed.

References

[1] Richard I. Hartley. Kruppa's equations derived from the fundamental matrix. *Pattern Analysis and Machine Intelligence*, 19(2):133–135, February 1997.

[2] Zhengyou Zhang. The matching problem: The state of the art. Technical Report 2146, Institut National de Recherche en Informatique et en Automatique, 1993.

[3] Zhengyou Zhang. Determining the epipolar geometry and its uncertainty: A review. Technical Report 2927, Institut National de Recherche en Informatique et en Automatique, July 1996.

[4] Quan-Tuan Luong and Olivier D. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17(1):43–75, January 1996.

[5] P. H. S. Torr and D. W. Murray. The developement and comparision of robust methods for estimating the fundamental matrix. *Int. Journal Computer Vision*, 24(3):271–300, 1997.

[6] Joaquim Salvi. *An Approach to Coded Structured Light to Obtain Three Dimensional Information*. PhD thesis, Universitat de Girona, 1997.

[7] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[8] Richard Hartley. In defence of the 8-point algorithm. In *Proceedings of the 5th International Conference on Computer Vision*, pages 1064–1070, 1995. IEEE Computer Society Press.

A comparative review of camera calibrating methods with accuracy evaluation

Xavier Armangué, Joaquim Salvi and Joan Batlle

Computer Vision and Robotics Group
Institute of Informatics and Applications
University of Girona
Av. Lluís Santaló, s/n, E-17071 Girona (Spain)
{armangue,jpages,qsalvi,jbatlle}@eia.udg.es

International Association for Pattern Recognition
5th Ibero-American Symposium on Pattern Recognition
SIARP 2000, Lisboa, Portugal
September 11-13, 2000

Acceptance Letter

SIARP
V IBERO-AMERICAN SYMPOSIUM ON PATTERN RECOGNITION, LISBON 2000

Subject: Acceptance of paper entitled ***A Comparative Review of Camera Calibrating Methods with Accuracy Evaluation*** for presentation at the 5th Iberoamerican Symposium on Pattern Recognition.

June 30, 2000

We are pleased to inform you that your paper has been accepted for presentation at the *5th Iberoamerican Symposium on Pattern Recognition*.

We would now ask you to prepare a final version of your manuscript, according to the guidelines described at:

http://alfa.ist.utl.pt/~siarp

Some particularly important comments of the referees are also enclosed and we would ask you to pay particular attention to these when finalizing your paper.

Please, send the final version of your manuscript in an electronic format (by e-mail or in a floppy disk) and in paper, by mail, to the following address, until July 15:

SIARP 2000
CVRM – Geosystems Center
Instituto Superior Técnico
Av. Rovisco Pais
1049-001Lisboa
PORTUGAL

Thank you for your contribution.

Yours sincerely,

F. Muge
M. Piedade
R. Caldas Pinto
(The Chairs of the SIARP)

SIARP 2000
CVRM-GEOSYSTEMS CENTER - INSTITUTO SUPERIOR TÉCNICO
AV. ROVISCO PAIS – 1049-001 LISBON • PORTUGAL
PHONE: +351-21-841 72 47 • FAX: +351-21-841 74 42 • E-MAIL: siarp@alfa.ist.utl.pt

A COMPARATIVE REVIEW OF CAMERA CALIBRATING METHODS WITH ACCURACY EVALUATION[1]

X. Armangué, J. Salvi and J. Batlle
Institut de Informàtica i Aplicacions
Universitat de Girona
Av. Lluís Santaló, s/n, E-17071 Girona (Spain)
Mail: {armangue,qsalvi,jbatlle}@eia.udg.es

Abstract

Camera modelling and calibrating is a crucial problem for further metric measuring of a scene. A lot of different calibrating techniques and surveys about calibration have been presented in last years. However, it is still difficult to go into details of a determined calibrating technique and to compare it with respect to the others. Mainly, this problem emerges due to the lack of notation standardization and the different existing methods of accuracy evaluation. This article presents a detailed review about five of the most used calibrating techniques and a great deal of effort has been done to present them by using the same notation. Moreover, the techniques have been implemented and the accuracy evaluation results are shown in the article.

1. INTRODUCTION

Camera calibration is the first step toward computational computer vision. Although some information from the measuring scenes can be obtained from uncalibrated cameras, calibration is essential when metric information is required. Some applications of this capability includes:

1. *Dense reconstruction:* Each image point determines an optical ray passing through the focal point of the camera toward the scene. Use of more than a view of a motionless scene permits to cross both optical rays and get the metric position of the 3D point.

2. *Visual inspection:* Once a dense reconstruction of a measuring object is obtained, it can be compared with an stored model in order to detect some manufacturing imperfections as bumps, dents and cracks.

[1]This work has been supported by Spanish project CICYT TAP99-0443-C05-01

3. *Object localization:* From points of different objects, the position relation among these objects can be easily determined, which has many application in industrial part assembly and obstacle avoidance in robot navigation, among others.

4. *Camera localization:* When a camera is placed at the hand of a robot arm or on a mobile robot, the position and orientation of the camera can be computed through the localization of some known landmarks in the scene. If these measures are stored, a temporal analysis permits to obtain the trajectory of the robot, which can be used in robot control and path planning.

Camera calibration can be classified according to several different criteria. For instance, 1) Linear versus nonlinear camera calibration (usually differentiate by the modelling or not of the lens distortion [1]). 2) Implicit versus explicit calibration. Implicit calibration is the process of calibrating a camera without explicitly computing its physical parameters. Although, the results can be used for 3D measurement and generation of image coordinates, they are useless for camera modelling as the obtained parameters do not correspond to the physical ones [2]. And 3), methods that use known 3D points as a calibrating pattern [3], and others that use some geometrical properties of the scene such as vanishing lines. Moreover, the approaches can be classified with respect to the calibrating method used to estimated the parameters of the camera model:

1. *Non linear optimization techniques.* The camera parameters are obtained through iteration with the constraint of minimizing a determined function. The advantage of these techniques is that it can calibrate nearly any model and the accuracy usually increase by increasing the number of iterations. However, these techniques require a good initial guess in order to guarantee the convergence. Examples: classic photogrammetry and Salvi [4].

2. *Linear techniques which compute the transformation matrix.* These techniques use the least squares method to obtain a transformation matrix which relates 3D points with their projections. However, these techniques can not model lens distortion. Moreover, it's sometimes difficult to extract the parameters from the matrix due to the implicit calibration used. Examples: Hall [5], Toscani-Faugeras [3] and Ito [1].

3. *Two-step techniques.* These techniques use a linear optimization to compute some of the parameters and, as a second step, the rest of parameters are iteratively computed. These techniques permit a rapid calibration reducing considerably the number of iterations. Moreover, the convergence

is nearly guarantee due to the linear guess obtained in the first step. Examples: Tsai [6], Weng [7] and Wei [2].

This article is a detailed survey of some of the most used calibrating techniques. A great deal of effort has been done to present the survey using the same notation. Moreover, the techniques have been implemented and comparative results are shown and discussed.

The article is structured as follows: Section 2 deals with camera modelling. Section 3 describes some techniques of camera calibrating. Section 4 explains some methods for the accuracy evaluation of camera calibrating techniques. The paper ends with conclusions.

2. CAMERA MODEL

The model is a mathematical formulation which approximates the behaviour of any physical device, i.e a camera. In such a case, the internal geometry and the position and orientation of the camera in the scene is modelled. There are several camera models depending on the desired accuracy. The simplest model is the one proposed by Hall [5]. The goal is to find a relationship among the 3D points of the scene with their 2D projecting point in the plane image. This relationship is approximated by means of a transformation matrix,

$$\begin{pmatrix} s^I X_d \\ s^I Y_d \\ s \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \end{pmatrix} \begin{pmatrix} {}^W X_w \\ {}^W Y_w \\ {}^W Z_w \\ 1 \end{pmatrix} \tag{1}$$

Then, given a 3D point P_w with respect to the metric world coordinate system, applying the transformation matrix proposed by Hall, the 2D point P_u in pixels with respect to the image coordinate system is obtained. However, other more complex models decompose the transformation of the point P_w into the point P_d in 4 steps, explained in the following (see also figure 1):

1. The first step consists on relating the point P_w from the world coordinate system to the camera coordinate system.
2. Next it is necessary to carry out the projection of the point P_c on the image plane obtaining the point P_u, by using a projective transformation.
3. The third step models the lens distortion, based on a disparity of the real projection. Then, the point P_u is transformed to the real projection P_d.
4. Finally, the last step consists on carrying out another coordinate system transformation in order to go from the metric coordinate system of the camera to the image coordinate system of the computer in pixels.

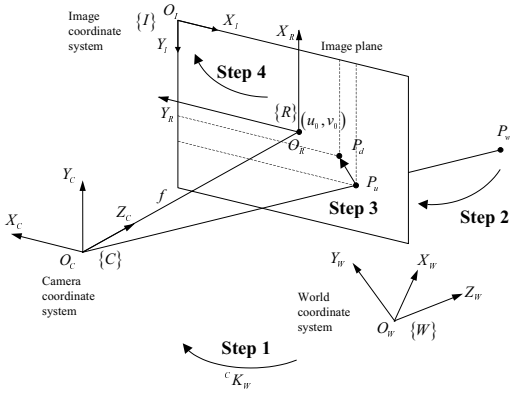


Figure 1: The geometric relation between a 3D point and its 2D projection.

In the following four different camera models (Faugeras-Toscani [3], Faugeras-Toscani with distortion [4], Tsai [6] and Weng [7]) are explained in detail considering how they carry out this four steps.

2.1. Changing from the world to the camera coordinate system

Changing the system of coordinates of the world to the system of coordinates of the camera is carried out in the same way in the 4 surveyed models. This transformation is modelled using a translation vector and a rotation matrix,

$$\begin{pmatrix} {}^CX_w \\ {}^CY_w \\ {}^CZ_w \end{pmatrix} = {}^CR_W \begin{pmatrix} {}^WX_w \\ {}^WY_w \\ {}^WZ_w \end{pmatrix} + {}^CT_W \tag{2}$$

2.2. Projection of the point 3D on the image plane

Consider that any optical sensor can be modelled as a *pinhole camera*. That is, the image plane is located at a distance f from the optical centre O_C, and it is parallel to the plane defined by the coordinate axis X_C and Y_C. Moreover, given an object point P_w related to the camera coordinate system, if it is projected through the focal point O_C, the optical ray intercepts the image plane at the 2D image point P_u. This relation is shown in equation. The four reviewed models solved the projective transformation by using equation (3).

$${}^CX_u = f\frac{{}^CX_w}{{}^CZ_w} \qquad {}^CY_u = f\frac{{}^CY_w}{{}^CZ_w} \tag{3}$$

2.3. Lens distortion

The third step is based on modelling of the distortion of the lenses. Each surveyed model proposed different approaches. Equation (4) transforms the point P_u without distortion to the point P_d with distortion, δ_x and δ_y represents the applied distortion.

$$ {}^{C}X_u = {}^{C}X_d + \delta_x \qquad {}^{C}Y_u = {}^{C}Y_d + \delta_y \tag{4} $$

The camera model proposed by Faugeras-Toscani [3] does not model the lens distortion, therefore P_u and P_d are the same point. In this case δ_x and δ_y are zero.

The model of Faugeras-Toscani can be improved by modelling the lens distortion [4] as also considered Tsai [6]. This displacement can be modelled by equation (5), which considers only the first coefficient k_1.

$$ \delta_x = k_1 {}^{C}X_d \left({}^{C}X_d{}^{2} + {}^{C}Y_d{}^{2} \right) \qquad \delta_y = k_1 {}^{C}Y_d \left({}^{C}X_d{}^{2} + {}^{C}Y_d{}^{2} \right) \tag{5} $$

The model of Weng [7] considers three types of distortion: radial distortion, decentering distortion and thin prism distortion. The total distortion will be the sum of the three distortions like it is shown in the following equations,

$$ \delta_x = \delta_{xr} + \delta_{xd} + \delta_{xp} \qquad \delta_y = \delta_{yr} + \delta_{yd} + \delta_{yp} \tag{6} $$

The radial distortion is modelled in the same way Tsai did. The decentering distortion is due to that the optic centre of the lenses is not correctly aligned with the centre of the camera [7]. This type of distortion introduces a radial and tangential distortion. Moreover, the thin prism distortion arises from imperfection in lens design and manufacturing as well camera assembly. This type of distortion can be modelled adding a thin prism in the optic system, causing radial and tangential distortions [7]. Adding the three types of distortions the following equations are obtained,

$$ \begin{aligned} \delta_x &= (g_1 + g_3) {}^{C}X_d{}^{2} + g_4 {}^{C}X_d {}^{C}Y_d + g_1 {}^{C}Y_d{}^{2} + k_1 {}^{C}X_d \left({}^{C}X_d{}^{2} + {}^{C}Y_d{}^{2} \right) \\ \delta_y &= g_2 {}^{C}X_d{}^{2} + g_3 {}^{C}X_d {}^{C}Y_d + (g_2 + g_4) {}^{C}Y_d{}^{2} + k_1 {}^{C}Y_d \left({}^{C}X_d{}^{2} + {}^{C}Y_d{}^{2} \right) \end{aligned} \tag{7} $$

2.4. Changing from the camera to the computer image coordinate system

The last step deals with relating the P_d point with respect to the computer image plane in pixels. This change of coordinates can be made in two different ways according to the surveyed camera models.

The camera models proposed by Faugeras-Toscani, Faugeras-Toscani with distortion and Weng use the following equations,

$$ {}^{I}X_d = -k_u {}^{C}X_d + u_0 \qquad {}^{I}Y_d = -k_v {}^{C}Y_d + v_0 \tag{8} $$

where, (k_v,k_u) are these parameters transforms from metric measures with respect to the camera coordinate system to pixels with respect to the computer image coordinate system; and (u_0, v_0) are these components defines the projection of the focal point in the plane image in pixels, i.e. the principal point.

The camera model of Tsai proposed another equations to carry out the same transformation. The equations are the following,

$$ {}^{I}X_d = s_x d'^{-1}_x {}^{C}X_d + u_0 \qquad {}^{I}Y_d = d_y^{-1} {}^{C}Y_d + v_0 \tag{9} $$

where, (u_0, v_0) are the coordinates of the projection of the focal point in the plane image in pixels; s_x is the image scale factor; d_x is the centre to centre distance between adjacent sensor elements in the X direction; d_y is the centre to centre distance between adjacent sensor elements in the Y direction; N_{cx} is the number of sensor elements in the X direction; N_{fx} is the number of pixels in a line as sampled by the computer; and $d'_x = d_x \frac{N_{cx}}{N_{fx}}$.

3. CALIBRATING METHODS

The calibrating method depends on the model used to approximate the behaviour of the camera. The linear models, i.e Hall and Faugeras-Toscani, use a least-squares technique to obtain the parameters of the model. However, non-linear calibrating methods as: Faugeras-Toscani with distortion, Tsai and Weng; use a two stages technique. They carry out a linear approximation with the aim of obtaining an initial guess for a further iterative optimization.

3.1. Method of Hall

The method used to calibrate the model of Hall is based on expressing equation (1) in the following form,

$$ \begin{array}{l} {}^{I}X_u = \frac{A_{11} {}^{W}X_w + A_{12} {}^{W}Y_w + A_{13} {}^{W}Z_w + A_{14}}{A_{31} {}^{W}X_w + A_{32} {}^{W}Y_w + A_{33} {}^{W}Z_w + A_{34}} \\ {}^{I}Y_u = \frac{A_{21} {}^{W}X_w + A_{22} {}^{W}Y_w + A_{23} {}^{W}Z_w + A_{24}}{A_{31} {}^{W}X_w + A_{32} {}^{W}Y_w + A_{33} {}^{W}Z_w + A_{34}} \end{array} \tag{10} $$

Then, consider that the 3D position of a set of n calibrating points and their corresponding 2D projection in the image are known (n should be bigger or equal to 6). Moreover, if we consider without lost of generality that $A_{34} = 1$, all the elements of the transformation matrix A can be obtained by using a linear least-squares technique as the pseudo-inverse [5].

3.2. Method of Faugeras-Toscani

In order to obtain the complete model of the camera, it is necessary to combine equations (2), (3), (4) and (8), obtaining (11).

$$\begin{array}{l} {}^{I}X_u = -k_u f \frac{r_{11}{}^{W}X_w + r_{12}{}^{W}Y_w + r_{13}{}^{W}Z_w + t_x}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} + u_0 \\ {}^{I}Y_u = -k_v f \frac{r_{21}{}^{W}X_w + r_{22}{}^{W}Y_w + r_{23}{}^{W}Z_w + t_y}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} + v_0 \end{array} \tag{11}$$

Note that equation (11) can be expressed in matricial form in the following way, where $\alpha_u = -fk_u$ and $\alpha_v = -fk_v$.

$$\begin{pmatrix} s\,{}^{I}X_d \\ s\,{}^{I}Y_d \\ s \end{pmatrix} = \begin{pmatrix} \alpha_u r_1 + u_0 r_3 & \alpha_u t_x + u_0 t_z \\ \alpha_v r_2 + v_0 r_3 & \alpha_v t_y + v_0 t_z \\ r_3 & t_z \end{pmatrix} \begin{pmatrix} {}^{W}X_w \\ {}^{W}Y_w \\ {}^{W}Z_w \\ 1 \end{pmatrix} \tag{12}$$

Then, equation (12) is combined in the following way,

$$\begin{array}{rcl} {}^{I}X_u & = & \frac{A_1}{A_{34}}{}^{W}P_w + \frac{A_{14}}{A_{34}} - \frac{A_3}{A_{34}}{}^{W}P_w\,{}^{I}X_u \\ {}^{I}Y_u & = & \frac{A_2}{A_{34}}{}^{W}P_w + \frac{A_{24}}{A_{34}} - \frac{A_3}{A_{34}}{}^{W}P_w\,{}^{I}Y_u \end{array} \tag{13}$$

At that time, a set of 5 parameters are considered $X = (T_1, T_2, T_3, C_1, C_2)^{\mathrm{T}}$, which are $T_1 = \frac{A_1}{A_{34}}$, $T_2 = \frac{A_3}{A_{34}}$, $T_3 = \frac{A_2}{A_{34}}$, $C_1 = \frac{A_{14}}{A_{34}}$ and $C_2 = \frac{A_{24}}{A_{34}}$. Then, the value of the vector X is obtained by using a linear least-squares technique. Finally, the camera parameters are extracted from X by using equation (12).

3.3. Method of Faugeras-Toscani with radial distortion

When a good accuracy is necessary, the linear method of Faugeras-Toscani becomes useless. However, it can be easily modified including the radial lens distortion as it has been shown in section 2.3. Nevertheless, the equations turns into non-linear and the least-squares techniques has to be replaced by an iterative algorithm.

Note that combining equation (2), (3), (4) and (5), equation (14) is obtained. Moreover, equation (8) has to be used to transform from metric coordinates to pixels.

$$\begin{array}{rcl} {}^{C}X_d + {}^{C}X_d k_1 r^2 & = & f \frac{r_{11}{}^{W}X_w + r_{12}{}^{W}Y_w + r_{13}{}^{W}Z_w + t_x}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} \\ {}^{C}Y_d + {}^{C}Y_d k_1 r^2 & = & f \frac{r_{21}{}^{W}X_w + r_{22}{}^{W}Y_w + r_{23}{}^{W}Z_w + t_y}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} \\ r & = & \sqrt{{}^{C}X_d{}^{2} + {}^{C}Y_d{}^{2}} \end{array} \tag{14}$$

where $x = (\alpha, \beta, \gamma, t_x, t_y, t_z, k_u, k_v, u_0, v_0, k_1)^{\mathrm{T}}$ is the vector of unknowns which can be computed by using an iterative method as, for instance, the method of Newton-Raphson or Levenberg-Marquardt, among others.

Note that one of the problems of convergence in iterative algorithms is the initial guess. However, an initial guess can be obtained calibrating the linear method of Faugeras-Toscani without lens distortion, and assuming $k_1 = 0$.

3.4. Method of Tsai

The non-linear method of Faugeras-Toscani is based on fixing the initial guess without considering lens distortion. Moreover, a large number of iterations is usually necessary to obtain an accurate value of the camera parameters. The method of Tsai [6] models also the radial lens distortion but assumes that there are some parameters of the camera that are distributed by manufacturers. This fact reduces the number of calibrating parameters in the first steps. However, all of them are iteratively optimized in the last step.

Firstly, by combining equation (2), (3), (4) and (5), equation (14) is obtained. Note that, at this point Tsai's model is equivalent to the previous one of Faugeras-Toscani with distortion. Note also, that the points have been expressed in metric coordinates instead of pixels for easier comprehension (use equation (9) to express the points in pixels)

Once ${}^{C}X_d$ and ${}^{C}Y_d$ are obtained in metric coordinates by using equation (9), they can be expressed in pixels ${}^{I}X_d$ and ${}^{I}Y_d$, assuming that $s_x = 1$ and the principal point is located in the physical centre of the image plane. Moreover, by using equation (14) and considering a radial lens distortion, some geometrical principles can be applied, which leads us to the following substitution of unknowns.

$$\begin{array}{lll} a_1 = t_y^{-1} s_x r_{11} & & a_5 = t_y^{-1} r_{21} \\ a_2 = t_y^{-1} s_x r_{12} & a_4 = t_y^{-1} s_x t_x & a_6 = t_y^{-1} r_{22} \\ a_3 = t_y^{-1} s_x r_{13} & & a_7 = t_y^{-1} r_{23} \end{array} \tag{15}$$

The a_i unknowns are obtained by using least-squares technique. Moreover, assuming that the orientation vectors of a rotation matrix are orthogonal, t_y, t_x and ${}^{C}R_W$ are obtained. This complete the first three steps of the method of Tsai. However, two more steps are programmed in order to calculate f, t_z and k_1. Firstly, f, t_z are approximated by least-squares, considering $k_1 = 0$. Then, this approximation determines an initial guess, which is used by an iterative algorithm to obtain f, t_z and k_1 accurately. Finally, all the parameters are optimized iteratively with the aim of obtaining an accurate solution.

3.5. Method of Weng

The method of Tsai is based on modelling radial lens distortion. The accuracy obtained by Tsai is very good and enough for most of the applications. However, in same cases where the lens of the camera need to be accurately modelled a simple radial approximation is not enough. Weng [7] improves the

model proposed by Faugeras-Toscani [3] including three types of lens distortion as it is explained in section 2.3. Then, by using equations (2), (3), (4) and (7), equation (16) is obtained. As in the previous models, the 2D point coordinates are expressed in metric (use equation (8) to transform them to pixels).

$$\begin{aligned} f\frac{r_{11}{}^{W}X_w + r_{12}{}^{W}Y_w + r_{13}{}^{W}Z_w + t_x}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} &= {}^{C}X_d + (g_1 + g_3)\,{}^{C}X_d{}^{2} \\ &\quad + g_4\,{}^{C}X_d\,{}^{C}Y_d + g_1\,{}^{C}Y_d{}^{2} + k_1\,{}^{C}X_d\left({}^{C}X_d{}^{2} + {}^{C}Y_d{}^{2}\right) \\ f\frac{r_{21}{}^{W}X_w + r_{22}{}^{W}Y_w + r_{23}{}^{W}Z_w + t_y}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} &= {}^{C}Y_d + g_2\,{}^{C}X_d{}^{2} \\ &\quad + g_3\,{}^{C}X_d\,{}^{C}Y_d + (g_2 + g_4)\,{}^{C}Y_d{}^{2} + k_1\,{}^{C}Y_d\left({}^{C}X_d{}^{2} + {}^{C}Y_d{}^{2}\right) \end{aligned} \tag{16}$$

Then, equations (17) are obtained.

$$\begin{array}{lcllcl} W_1 & = & \alpha_u r_1 + u_0 r_3 & w_4 & = & \alpha_u t_x + u_0 t_z \\ W_2 & = & \alpha_v r_2 + v_0 r_3 & w_5 & = & \alpha_v t_y + v_0 t_z \\ W_3 & = & r_3 & w_6 & = & t_z \end{array} \tag{17}$$

At this point, $W = (W_1, W_2, W_3, w_4, w_5, w_6)^{\mathrm{T}}$ is computed by using a linear least-squares technique. Then, w_6 is assumed to be the unity in order to extract from W the values of all the parameters of the vector $m = (u_0, v_0, \alpha_u, \alpha_v, t_x, t_y, t_z, \alpha, \beta, \gamma)^{\mathrm{T}}$, without considering distortion. Furthermore, ${}^{C}R_W$ is recalculated in order to keep orthogonality by using eigen values, and the rest of parameters are recalculated.

Then, an iterative method is used to recalculate, for the third time, the values of m, assuming zero distortion. Finally, a two stages iterative method is used. In the first stage, the parameters of $d = (k_1, g_1, g_2, g_3, g_4)^{\mathrm{T}}$ are linearly obtained by using least-squares, and the second stage computes the values of m iteratively. This stages are repeated as many times as needed depending on the desired accuracy.

4. ACCURACY EVALUATION

The systems used to evaluate the accuracy of camera calibrating methods can be classified into two groups depending on whether the accuracy is measured from the 3D or 2D points. In the following some of the most used methods of accuracy evaluation are described.

4.1. 3D Measurement

1. *3D position obtained from stereo triangulation.* First, acquire two images of a set of 3D test points whose 3D coordinates are known. Second, com-

pute the estimated 3D coordinates of the same points from their projections using the calibrated parameters. Finally, compare the discrepancy between real and estimated positions.

2. *Radius of ambiguity in the calibrating plane.* First, acquire a set of 3D test points, lying on a plane of test, whose coordinates in the world coordinate system are known. Second, for each image point use the calibrated model to back project the ray from the focal point through the 2D projection. The intersection of the optical ray with the plane of test determines a point. Then, the distance from the 3D test point to the intersecting points defines a radius of ambiguity around the 3D test point.

3. *Distance with respect to the optical ray.* This method is a generalization of the previous one. In this case, the discrepancy to be measured is the distance of the 3D test points from the optical ray generated by their projections.

4. *Normalized Stereo Calibration Error.* The array of pixels in an image is projected back to the scene so that each back-projected pixel covers a certain area of the object surface. This area indicates the uncertainty of the basic resolution at this distance. Where $({}^{C}X_{wi}, {}^{C}Y_{wi}, {}^{C}Z_{wi})$ is the real coordinates of the i_{th} 3D object point and $({}^{C}\hat{X}_{wi}, {}^{C}\hat{Y}_{wi}, {}^{C}\hat{Z}_{wi})$ is the coordinates obtained by back-projecting the pixel and intersecting it with the surface plane. See Weng [7] for detailed information. Then, it is defined the Normalized Stereo Calibration Error (NSCE) as show equation 18.

$$\text{NSCE} = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{\left({}^{C}\hat{X}_{wi}-{}^{C}X_{wi}\right)^2+\left({}^{C}\hat{Y}_{wi}-{}^{C}Y_{wi}\right)^2}{{}^{C}\hat{Z}_{wi}{}^2\left(\alpha_u^{-2}+\alpha_v^{-2}\right)/12}\right]^{1/2} \tag{18}$$

4.2. 2D Measurement

1. *Accuracy of distorted image coordinates.* First, take an image of a set of 3D test points. Then, calculate the 2D position on the image plane of each 3D point, taking into account lens distortion. Accuracy is obtained measuring the discrepancy between the real 2D points and the estimated one.

2. *Accuracy of undistorted image coordinates.* First, take an image of a set of 3D test points. Then, calculate the linear projection of the 3D points on the image plane. Besides, detect the real 2D points through image segmentation and remove the distortion of lens by using the camera model. Finally, accuracy is obtained measuring the discrepancy between the linear projection and the undistorted points.

5. EXPERIMENTAL RESULTS

The five calibrating techniques have been implemented and their accuracy measured by using the following criteria: 1) Distance with respect to the optical ray; 2) Normalized Stereo Calibration Error; 3) Accuracy of distorted image coordinates; 4) Accuracy of undistorted image coordinates.

Using a set of object ponits proposed by Tsai[2]. Table 1 shows the accuracy measured by using the first criteria and second criteria, respectively. Comparing the mean of the results obtained in both tables, we can see that a relationship exists among the two methods. Both methods obtained similar results if they are relatively compared that is, good calibrating algorithms obtain acceptable accuracy results not depending on the evaluation method used. Moreover, table 2 shows the results of calculating the accuracy by using the third and fourth criteria, respectively.

	3D position (mm)			NSCE	N° of
	Mean	σ	Max		iterations
Hall	0.1615	0.1028	0.5634	n/a	n/a
Faugeras	0.1811	0.1357	0.8707	0.6555	n/a
Faugeras NR[3] without distortion	0.1404	0.9412	0.0116	0.6784	20
Faugeras NR with distortion	0.0566	0.0307	0.1694	0.2042	20
Tsai	0.1236	0.0684	0.4029	0.4468	57
Tsai optimized	0.0565	0.0306	0.1578	0.2037	499
Tsai with principal point of Tsai optimized	0.0593	0.0313	0.1545	0.2137	52
Tsai optimized with principal point of Tsai optimized	0.0564	0.0305	0.1626	0.2033	355
Weng	0.0570	0.0305	0.1696	0.2064	200

Table 1: Accuracy of 3D Coordinate Measurement

	2D distorted image (pix.)			2D undistorted image (pix.)		
	Mean	σ	Max	Mean	σ	Max
Hall	0.2676	0.1979	1.2701	0.2676	0.1979	1.2701
Faugeras	0.2689	0.1997	1.2377	0.2689	0.1997	1.2377
Faugeras NR without distortion	0.2770	0.2046	1.3692	0.2770	0.2046	1.3692
Faugeras NR with distortion	0.0840	0.0458	0.2603	0.0834	0.0454	0.2561
Tsai	0.1836	0.1022	0.6082	0.1824	0.1011	0.6011
Tsai optimized	0.0838	0.0457	0.2426	0.0832	0.0453	0.2386
Tsai with principal point of Tsai optimized	0.0879	0.0466	0.2277	0.0872	0.0463	0.2268
Tsai optimized with principal point of Tsai optimized	0.0836	0.0457	0.2500	0.0830	0.0454	0.2459
Weng	0.0845	0.0455	0.2608	0.0843	0.0443	0.2584

Table 2: Accuracy of 2D Coordinate Measurement

[2] `http://www.cs.cmu.edu/~rgw/TsaiCode.html`

[3] Newton-Raphson

6. CONCLUSION

This article surveys five of the most used calibrating techniques. Effort has been done to unify the notation among the five methods, so that they have been presented in a way that the reader can understand them easily. We can see that the differences among them are mainly in the step concerning the modelling of lens.

Moreover, a survey on accuracy evaluation has been done. The five methods have been implemented and their accuracy analyzed. Results show that only non-linear methods obtain a 3D accuracy smaller than 0.1 mm. with a very good standard deviation. Moreover, the accuracy on the image plane of non-linear methods is much better than linear methods. However, non-linear methods are more time-consuming than linear ones. Obviously, the results only prove something already demonstrated by authors. However, in this article the five methods have been compared among them, so that the reader can choose one or another method depending on their applications. Future work. it would be useful if they could compare the same methods by artificially including noise.

References

[1] M. Ito, "Robot vision modelling - camera modelling and camera calibration," *Advanced Robotics*, vol. 5, no. 3, pp. 321–335, 1991.

[2] G.-Q. Wei and S. De Ma, "Implicit and explicit camera calibration: Theory and experiments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 469–480, May 1994.

[3] O. D. Faugeras and G. Toscani, "The calibration problem for stereo," *Proceedings of IEEE Computer Vision and Pattern Recognition*, pp. 15–20, 1986.

[4] J. Salvi, J. Batlle, and E. Mouaddib, "A robust-coded pattern projection for dynamic 3D scene measurement," *Int. Journal of Pattern Recognition Letters*, vol. 19, pp. 1055–1065, September 1998.

[5] E. L. Hall, J. B. K. Tio, C. A. McPherson, and F. A. Sadjadi, "Measuring curved surfaces for robot vision," *Computer Journal*, vol. December, pp. 42–54, 1982.

[6] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Int. Journal on Robotics and Automation*, vol. RA-3, pp. 323–344, August 1987.

[7] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 965–980, October 1992.

A computer vision system for autonomous forklift vehicles in industrial environments

Jordi Pagès, Xavier Armangué, Joaquim Salvi,
Jordi Freixenet and Joan Martí

Computer Vision and Robotics Group
Institute of Informatics and Applications
University of Girona
Av. Lluís Santaló, s/n, E-17071 Girona (Spain)
{jpages,armangue,qsalvi,jordif,joanm}@eia.udg.es

IEEE Control Systems Society - IEEE Robotics and Automation Society
9th Mediterranean Conference on Control and Automation
MED 2001, Dubrovnik, Croatia
June 27-29, 2000

Acceptance Letter

Assumpte: MED01
Data: Sat, 31 Mar 2001 12:56:17 +0200
De: Zdenko Kovacic <zdenko.kovacic@fer.hr>
A: <jpages@eia.udg.es>

Dear Dr. Jordi Pages

On behalf of the Program Committee of the 9th Mediterranean Conference on Control and Automation MED'01 it is my pleasure to inform you that your paper MED01-071

A Computer Vision System for Autonomous Forklift Vehicles in Industrial Environments

has been accepted for presentation at the MED'01 Conference in Dubrovnik. The instructions on how to prepare your final camera-ready paper can be found on the web address: `http://med01.rasip.fer.hr/authors.html`. Paper(s) should be submitted in standard IEEE Transactions format. A one page abstract in one column format should be also uploaded for the printed Book of Abstracts. We greatly appreciate your attention to these instructions while preparing your paper for publication in the conference materials. Paper(s) and abstract(s) have to be submitted electronically through already adopted upload procedure (more details can be found on the conference web site) no later than April 15, 2001. In case you encounter problems with upload, please contact the Publication Chair (`zeljko.ban@fer.hr`). The Preliminary Program will be announced soon on the conference web page: `http://med01.rasip.fer.hr`.

Please do not hesitate to contact me if you have any questions about your paper or the Conference.

We thank you for your valuable contribution and are looking forward to welcome you at the MED'01 Conference.

Yours sincerely,

Prof. Dr. Zdenko Kovacic
Program Chair

P.S. - Please acknowledge the receipt of this message.

A Computer Vision System for Autonomous Forklift Vehicles in Industrial Environments[1].

J. Pagès, X. Armangué, J. Salvi, J. Freixenet and J. Martí.

***Abstract*— This paper describes a vision system to detect the 3D position of pallets for autonomous forklift vehicles. An accurated image segmentation method based on colour and geometric characteristics of the pallet is proposed. Moreover, the application computes the 3D position and orientation of the pallet and generates the vehicle trajectory to fork it. The system has been tested and experimental results are shown.**

***Index Terms*— 3D Image Reconstruction, Computer Vision, Feature Selection, Industrial Application and Segmentation.**

I. INTRODUCTION

This article describes a complete computer vision system designed to segment and locate the pallets in an industrial environment with the aim of automating forklift vehicles. Usually, autonomous navigation is based on using ultrasonic sensors and odometers. Computer vision has not been used in general applications due to its lack of robustness in the presence of illumination variations and also due to computer complexity. However, computer vision shows a great performance in specific applications. For instance, it has been widely used in industrial inspection [1]. Moreover, some efforts have been done using structured light in order to reduce computer complexity [2]. However, it remains difficult to apply computer vision in industrial navigation.

This paper addresses a specific application: the forklift vehicle (see Fig. 1) has to detect the pallet, which is situated on the floor, and fork it autonomously. The problem has been solved using computer vision. Although it is not the purpose of this article, the vehicle uses also ultrasonic sensor to avoid obstacles and odometers to keep the trajectory.

The article is divided as follows. First, section II deals with the pallet modeling and segmentation. Section III is based on detecting the pallet in the 2D image once it has been segmented. Then, section IV describes the methodology used to obtain the 3D position of the pallet with respect to the vehicle using a single camera attached on its top. Finally, section V describes the method used to generate the trajectory to fork the pallet. The article discusses the experimental results and ends with conclusions.

Fig. 1. The forklift that is being automated

II. OBJECT MODELLING

A. Introduction

Models constitute representations of the real world, and thus, modelling implies the choice of a suitable set of parameters in order to build those representations. Obviously, the selection of the parameters will affect how well the models fit reality, and this becomes a central issue in any object recognition system. Here, a method for learning such models in training images is proposed.

Object modelling has been designed as a supervised task, where a teacher presents representative examples of objects in training images. Moreover, the presentation of which parts do not belong to the object of interest is also required. Afterwards, the application fulfil a stage of calculating or abstracting in order to find out which features can be used to identify whether an image pixel belongs to the target object. That is, the modelling process is planned as a features selection problem, in which the goal is to find the subset of features that best captures the representation of a given object. In this work, only colour features have been treated, however, the methodology here described is valid for whatever kind of features.

B. Feature selection

In this stage a combination of features which makes possible to identify an object must be found. The solution to this problem is entirely statistical due to the complexity and the amount of data to be treated. If all the combinations of features are tested, a problem of huge execution time has to be faced. Besides, there is another problem, more subjective, which is how to decide which set of features is the most appropriated. With the aim of showing the problem of feature evaluation, a simple example based only on two single features is proposed. Such a simplification has been considered in order to reduce the complexity of

Institute of Informatics and Applications. EIA-PII. University of Girona, Av. Lluis Santaló s/n, 17071 Girona (Spain). Jpages@eia.udg.es

1 This work has been supported by the CeRTAP consortium of the local government of Catalonia.

graphical representation of spaces with multiple features. In Fig. 2 two couples of colour features of a set of pixel samples from two different objects have been calculated.

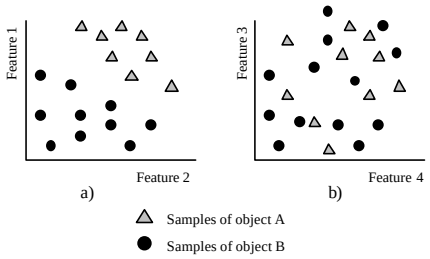


Fig. 2. Object samples distribution based on colour features couples.

The result is a 2D distribution of the given samples which allows us to see whether exists some relationship between them based on their feature values. The sample pixels corresponding to each object form two differentiated clusters, as shown on Fig. 2a, which means that these features can separate the pixels of both objects. Now, a procedure is needed to find out if these two features are more useful than the used in Fig. 2b, where there is no clear relationship between the features and the object samples.

The selection of an optimal subset of features will always rely on a certain evaluation function. Typically, an evaluation function tries to measure the discriminating ability of a feature or a subset of features, to distinguish the different objects [3]. Following the "divide and conquer" paradigm, the problem of feature evaluation has been solved by using a decision tree that separates the samples in a recursive way [4]. Fig. 3 shows such a decision tree operating in a two-dimensional feature space. The decision trees considered in our approach are binary trees with multivariate decision functions, where each node is a binary test represented by a linear function. The Fisher Linear Discriminant Function (LDF) has been demonstrated as a powerful classifier that maximizes the ratio between the inter-class covariance and the intra-class covariance [5]. Each node of the tree attempts to separate, in a set of known instances (the training set), a target (i.e., *pallet*) mapped as △, from non-target instances (*no-pallet*) mapped as ●. However, this is achieved only in a certain ratio, because realistic data is not always linearly separable. The resulting two subsets of samples are again subdivided into two parts using two new calculated linear functions. This process is extended along the binary tree structure, until an appropriate misclassification ratio is achieved. The result is a tree of hyperplane nodes that recursively try to divide the feature space into target and non-target samples. Every leaf has a group of samples with an associated misclassification. So that, all the subgroups of object samples can be found with multiple evaluation functions. To avoid an uncontrolled tree growing, the deep of its branches is usually limited.

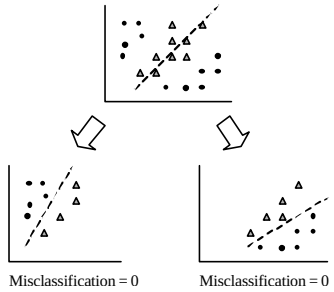


Fig. 3. Binary tree using Fisher recursive evaluation.

Once an evaluation function is chosen to compare the performance of different feature sets, a method to find the best set is required. As it was said before, all possible combinations of features should be tested. However other algorithms are usually used to avoid an exponential execution time. These methods are all based on reducing the search space, using for example AI algorithms like heuristics, genetics, etc. The technique chosen in this work is based on genetics [6]. In summary, it creates an initial population of chromosomes where every one of them represents a different combination of features. Each chromosome is a vector as long as the number of features chosen. Every position of the vector can be zero or one, indicating whether the feature associated to that position is present. Fitness is calculated for every chromosome, which is the misclassification produced for the activated features. A series of iterations are applied based on the initial population, trying to create new chromosomes by using cross techniques, mutating some parts of the existing and removing the worst of them. This process is repeated till a chromosome of certain fitness is generated. The performance of genetics, although its lack of determination, is quite good and the timing calculus is much better than an exhaustive combination method [6].

C. Real time image processing

One of the main traits of industrial applications is the capacity to work in real time. Therefore, the image processing, as part of our application, should be as fast as possible. The evaluation of a decision tree based on Fisher function for each pixel of an image requires a substantial processing time. To reduce it, a look up table is used. The procedure to create this table is simple: every possible combination of RGB colours are evaluated offline using the calculated Fisher tree. Thus, it is known *a priori*, which pixel have to be considered part of the target (pallet) and which must be filtered. All this information is stored in the look up table, that is a simple memory structure of three dimensions (one for each colour component). When an image has to be segmented, it is only necessary to calculate the RGB components of every pixel and consult the look up table to know whether is part of the target object. The use of a look up table is possible because only colour features have been used. If other features like textures are considered, a look up table is useless because of the required neighbourhood operation.

D. Requirements for a good colour segmentation

The study of the pallet colour has reflected that there are many grey-scale components in a large variety. This fact implies that a robust segmentation of the pallet is not easy to achieve if there are objects in the environment with important variety of grey components. As the aim of this work is for an industrial application in a controlled environment the adaptation of it to our requirements is truly possible. Therefore, the theoretical warehouse where this application will be used can be adapted so the colours of the walls and floor make easier the segmentation of the pallets.

III. Localization of the Pallet

The goal of locating the pallet in a complex environment is not an easy task. Thus, the images are pre-processed to remove the most part of the scene except the pallet, which will appear as a non-uniform region. The pre-processing task consists on a colour-based segmentation. Once the pallet is the only remarkable blob in the image, some techniques are applied to get the positional information and orientation of the pallet. An example of pallet segmentation is shown in Fig. 4. Note the image contains some noise that is further removed. The global shape of the pallet can be observed.

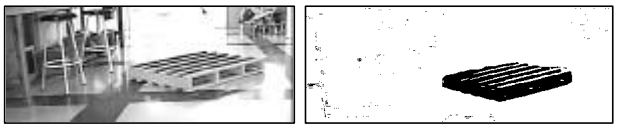


Fig. 4. Pallet colour segmentation sample.

Now, in order to fork the pallet with the forklifter, its 3D position and orientation have to be computed. Moreover, it is necessary to identify the nearest forking side of the pallet. Both tasks are explained in the following sections.

A. Pallet vertex detection

In order to find the pallet orientation, the Hough transform is proposed. The Hough transform can be applied to digital images, using the adapted algorithm introduced by Duda and Hart [7]. In summary, this method gathers information about the lines that can be found in the image. Besides, the pallet has a rectangular shape that, at maximum, two of its lateral sides can be observed in a 2D image depending on its orientation. Therefore, the lines of both sides that are in contact with the floor should be detected with Hough transform. Once these lines have been detected the position of three pallet vertexes can be easily found, as shown in Fig. 5. Images shown are in negative.

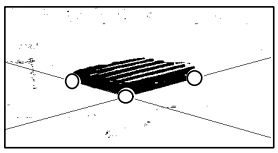


Fig. 5. Significative vertexes of a pallet

The way used to find the vertexes is here briefly described: both lines given by Duda-Hart are explored by using four cursors located at the four line ends. Each cursor explores the line going to the image centre until detects part of the pallet region, which determines one of the four vertexes and finishes its exploration. Obviously, a certain tolerance for this process is recommended in order to skip noise. Then, the exploration consists on using a 1x8 window centred in the line (instead of a 1x1 window) and moving it along the line. This is strongly recommended because of the inherent error of the discrete Hough transform, which can cause that the detected line differs a little from the right one.

It has been observed, during the study, that a single segment of the pallet is only required because the whole pallet can be reconstructed from it, as the pallet model is known.

The main constraint in the line detection process is the interpretation of the information given by Hough transform. The output of the Hough transform is a rectangular matrix in polar coordinates, where each cell represents one of the potential lines that might be present in the image. Every matrix cell contains a number expressing the amount of pixels that have been found in the source image that could be part of the same line. The cell with the highest number determines the largest line in the image. However, the main problem result from finding such a cell that represents the desired line among a non-uniform distribution with local maximums. The complexity of this problem has been described in some articles [8]. In this paper, an intermediate stage is applied to reduce the amount of local maximums. The process is described in the following section.

1) Image filtering process

The intermediate process is based on cleaning out the images before applying Hough by using filtering and morphologic operators. The source images in this process have been already segmented so they are binary coded.

The first step consists in a *close* morphological operation to grow the inner pallet region with the goal of producing a single blob. Secondly, an *open* operator is applied with the aim of removing the noise. After most noise is eliminated, a *sobel* filter is applied to enhance the edges of the image. Afterwards, a thining operation is executed to obtain the skeleton of every line. The result of the three steps can be observed in Fig. 6a. As it can be seen, most part of the noise has been removed. However, a lot of lines are still detected if Hough is now applied. As our aim is the detection of a single edge of the pallet that is in contact with floor, the rest of lines should be firstly removed. In order to achieve this goal, another step is applied: every column of the image is explored keeping only the pixel with the highest Y component and removing the rest. The effects of this filter can be seen in Fig. 6b.

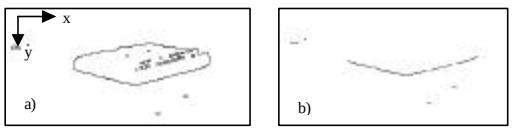


Fig. 6. a) Image filtering; b) Edge pallet detection.

As result of this procedure most part of pixels forming the two searched lines have overcome the filtering. Although some of them are not detected because of noise, the pallet edges still have an important pixel contribution so they are now easily detected by using Hough transform.

2) Maximum searching in Hough matrix

As it has been discussed in previous sections, a single pallet edge is required. It has been seen that, at least, one of the visible sides has a positive angle in polar coordinates, related to a coordinate system located at the left-bottom corner of the image. This could be used to restrict the scanning area of the hough matrix. However, both lines are searched in order to achive better results. This decission is due to the fact that some discrete lines of certain slopes have not a clear linear appearance. The scanning areas for maximums in the Hough matrix are determined by ρ = [1920, 3839], which represents the possible distances between the lines and the origin, and θ = ([159, 312], [316, 469] and [473,626]), representing the line angles. Fig. 7 shows the scanning areas and the correspondence between matrix indexes and angles. These ranges have been found with the analysis of a set of images where all boundary slopes of each side appeared. All this process could be easily automatized for any linear shape (not only rectangular) giving the number of sides and angles among them. A security thin blank space has been defined between each consecutive area with the aim of avoiding the discontinuities given by upright slopes. Finally, the maximum search starts.

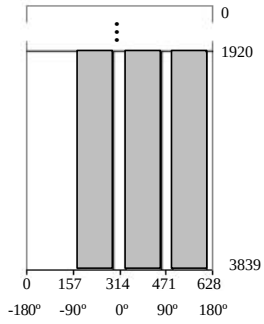


Fig. 7. Scanning areas of hough matrix of a 768x576 image.

An exhaustive scanning is done all over the area. Maximums do rarely appear as a single cell with a maximum value, but as a set of adjacent cells with the same value. Therefore, for each region of adjacent cells with the same maximum value, the gravity centre is calculated and stored as a local maximum. However, only the maximums that overcome a fixed threshold are considered. This minimum value has been fixed to 20 (what it means that lines with less than 20 pixels are not considered). Finally, the local maximum with the highest value is selected as the largest line in the image. Thus, a single edge of both pallet sides is presumed to be detected.

IV. 3D RECONSTRUCTION

A. Coplanar Tsai calibration

When the segment of one of the sides of the pallet, which are in contact with floor, has been detected in the image, its 3D real position in the world has to be computed in order to get the 3D position and orientation of the pallet. The method chosen is a transformation algorithm to get the 3D coordinates of 2D points from a single image. A camera calibration algorithm is required to calculate the relationship between the 2D points and the corresponding 3D optical rays. The Coplanar Tsai calibration method has been used [9]. This algorithm presumes that all the reconstructing points are placed in a plane, which coincides with the calibrating plane. Therefore, this method only gives accurate results under these conditions. The problem treated in this work can easily accomplish this requirement because the pallet is always on the floor and it can be used as the calibrating plane. In order to apply the calibration algorithm a 3D world coordinate system is required. Although it can be virtually situated everywhere, it is interesting to set it in the forklift adequately. Moreover, the camera is located as far as possible from the floor and on the forklift.

A set of 3D sample points (minimum 6, due to the number of variables involved) from the floor must be measured referred to the origin of the world coordinate system, trying to spread them to cover all the image scope, where the pallet can be located. The larger is the number of sample points the more accurate the transformation between 2D and 3D will be. Then, the 2D correspondences of every 3D point is measured in the image. Once the set of 3D points and their 2D correspondences are known, the camera can be calibrated by using the co-planar algorithm proposed by Tsai [9]. When the camera is calibrated, two main transformations can be applied: a) given a 3D point, its 2D projection on the image plane can be predicted; b) given a 2D point, its optical ray starting from the optical center of the image and passing through the 2D projection and the 3D point can be computed. The 3D position of every point lying on the floor can be computed intersecting the optical ray with such a floor. The, the position of the pallet related to the forklift (world coordinates) can be easily computed.

B. 3D Reconstruction of the pallet

As result of the 2D-edge localization process, both end points of the main segment in the image are known. Besides, it is known that both points lie on the floor, so the equation of the floor with respects to the world coordinate system is also known. This information is enough to calculate the rest of pallet points. Once we have the 3D coordinates of the detected points, the distance between them can be calculated. This distance represents the longitude of one of the pallet sides. The average dimensions of the pallets must be known and stored as part of the pallet model. The pallets used in our application have the largest side in the forking part, which is about 1 meter long. The

shorter side is about 0.8 m. Therefore, analyzing computed distances, the forking side of the pallet can be detected. The rest of the pallet can be reconstructed by simple scaling and rotating adequately the edges of the pallet.

V. TRAJECTORY CALCULATION

Once the 3D information of the pallet relative to the vehicle is obtained, a trajectory strategy to fork the pallet has to be appointed. The simplest trajectory consists on dividing it into two linear segments. The goal is to fork the pallet from the forking side close to the vehicle (Note every pallet has two forking sides). In Fig. 8 a schema of this kind of trajectory is shown [10].

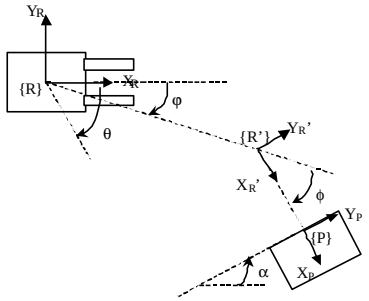


Fig. 8. Linear trajectory to fork the pallet.

In order to achieve such a trajectory, two couples of rotation-displacement operations are defined. The rotations are defined around de Z-axis, which is orthogonal to the floor, following the rules of a counter-clockwise system. The displacements are defined along the X-axis of the vehicle co-ordinate system. Then, three intermediate steps are considered to achieve the trajectory. The first step {R} is the position of the vehicle once the pallet has been detected. The second step {R'} represents the vehicle position when the first rotation and displacement and the second rotation have been made. Last step {P} represents the position and orientation of the vehicle when the pallet has been forked. Both rotations are identified by the angles j and f. The angle q represents the rotation that the vehicle might do in the initial step to get orthogonal to the target side of the pallet. The relationship of these angles is:

$$q = j + f$$

where θ can be calculated with the following rules:

$$\alpha > 0 \rightarrow \theta = \alpha - \pi/2$$
$$\alpha < 0 \rightarrow \theta = \alpha + \pi/2$$

where $\tan(\alpha)$ is the slope of the target side referred to X_R. α is expressed in the range of [π/2, -π/2]. The intermediate point of the trajectory {R'} is computed so that X_R' is pointing to the target and located at a predefined distance with respect to it. Thus, the second segment displacement is always constant. In order to calculate the motion parameters, the transformation matrices between the three co-ordinate systems are used:

$$^{R}A_{R'} = {}^{R}A_{P} \cdot ({}^{R}A_{P})^{-1}$$

Where $^{(*)}A_{(\#)}$ defines the co-ordinate system (#) referred with respect to (*). Observing Fig. 8 matrix $^{R}A_{R'}$ can be derived in the following way:

$$^{R}A_{R'} = \begin{pmatrix} \cos q & -\mathrm{sen}\, q & 0 & {}^{R}Px \\ \mathrm{sen}\, q & \cos q & 0 & {}^{R}Py \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} =$$
$$= \begin{pmatrix} \cos q & -\mathrm{sen}\, q & 0 & -d\cos q + {}^{R}P_x \\ \mathrm{sen}\, q & \cos q & 0 & -d\,\mathrm{sen}\, q + {}^{R}P_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Where $({}^{R}P_X, {}^{R}P_Y)$ are the co-ordinates of the central point of the target side of the pallet (origin of {P}) referred to {R}, d is the predefined distance between the origin of {P} and the origin of {R'}. Therefore, the co-ordinates of the intermediate point of the trajectory (origin of {R'}) are

$$(-d\cdot\cos q + {}^{R}\mathrm{Px}, -d\cdot\mathrm{sen}\, q + {}^{R}\mathrm{Py})$$

Finally, the rest of parameters can be computed:

$$j = \mathrm{ATAN2}(-d\cdot\mathrm{sen}\, q + {}^{R}\mathrm{Py} / -d\cdot\cos q + {}^{R}\mathrm{Px}) \qquad f = q - j$$
$$t1 = \sqrt{(-d\cdot\cos q + {}^{R}P_x)^2 + (-d\cdot\mathrm{sen}\, q + {}^{R}P_y)^2} \qquad t2 = d$$

where *t1* and *t2* are the two displacements along the X_R and $X_{R'}$ axis of the vehicle, respectively. The sequence of transformations to achieve the trajectory is: rotation φ, displacement *t1*, rotation ϕ and displacement *t2*.

VI. EXPERIMENTAL RESULTS

A Visual C++ application has been developed to study the performance of the whole process. All the different steps have been implemented from colour segmentation till trajectory calculation. With the aim of reducing the design time, the forklift has been implemented in a mobile robot model Pioneer2® of ActivMedia™, and a camera has been assembled on its top. This article describes only a part of a huge project financed by the local government, which integrates up to 7 research groups. Due to the difficulty to have the autonomous forklift all the time available in our university, as a first approach we have simulated its behavior in such a mobile robot.

A set of sample situations (varying the position and orientation of the pallet with respect to the vehicle) have been tested in the laboratory, where the environmental conditions have been adapted to simulate an industrial warehouse. Walls and floor colours have been chosen (white for the walls and green for the floor due to their easy segmentation), and the illumination system is light controlled. Such conditions have been discussed and accepted by the enterprises supporting the project and considered as really low-cost measures.

The study has addressed that the discrepancy between defined and real intermediate points depends on an inherent error produced by the control architecture of the robot, but also an error generated in the pallet localization step. The first component of the error has not been treated because it was not the aim of this work (other research groups are working in control architectures and sensor data fusion) and it is assumed to be small. The error due to the imprecision

of the pallet localization method has been studied. An obvious result is that the larger is the distance between the pallet and the robot, the higher is the error between predicted and real pallet position.

Table I shows the real and the calculated world coordinates of both points of the detected side in a set of images, as well as the real and calculated longitude of the side, besides the errors produced by the system (all in mm).

TABLE I
PALLET 3D LOCALIZATION RESULTS

Real points				**Calculated points**			
X1	*Y1*	*X2*	*Y2*	*X1*	*Y1*	*X2*	*Y2*
1909	1105	2254	169	1916	1084	2335	180
2049	-40	1909	-804	2078	-33	1928	-801
2311	0	2533	-748	2373	-1	2560	-747
2408	703	2136	-28	2409	702	2115	-59
2522	166	2311	-804	2565	178	2353	-809
2713	402	2755	-595	2722	392	2766	-577
2713	402	3087	-502	2742	399	3135	-540
3153	180	3014	-804	3093	186	3033	-810
3416	378	3416	-402	3467	376	3460	-410
3571	763	3416	0	3677	765	3439	-34
4689	374	4220	-506	4333	250	4521	-538

Errors				**Side longitude**	
X1	*Y1*	*X2*	*Y2*	*Real*	*Calculated*
-7	21	-81	-11	1000	996
-29	-7	-19	-3	800	808
-62	1	-27	-1	800	793
-1	1	21	31	800	842
-43	-12	-42	5	1000	1009
-9	10	-11	-18	1000	969
-29	3	-48	38	1000	1018
60	-6	-19	6	1000	997
-51	2	-44	8	800	812
-106	-2	-23	34	800	860
356	124	-301	77	1000	810

This table is only a summary of a bigger test where 50 different pallet orientation and positions were proved. If there is not an important loss of the pallet geometry in segmentation, the discrepancy between real and calculated points has an average of 31.5mm along the x-axis and 12.5mm along the y-axis, with a standard deviation of 21.7mm and 10.1mm, respectively (see Fig. 8). The discrepancy has been computed when the pallet is close to the vehicle (up to 3 m). The error of the x component is larger because the range of distances along this axis is also bigger. Note that the difference between both pallet sides is 200 mm, so the system can keep an accurate identification of the forking side of the pallet. Besides, if the segmentation process erodes the pallet edges, the forking side identification decreases considerably. However, this occurs only when the pallet is far from the vehicle. Then, the strategy consists in approaching the vehicle to the pallet and, then, recalculating the trajectory when the pallet is close to 3 meters. The idea of this strategy is that the orientation of the pallet is not important until it is separated less than a certain distance from the vehicle, while if the distance is larger, the application only tries to calculate the mass center of the pallet. Before calculating the mass center, the statistical technique of the median is applied to remove possible remaining noise. The mass centre is used to calculate the approaching trajectory. Once the distance between pallet and vehicle is less than 3m, the detection process is used. If the dimensions of the detected pallet do not fit the pallet model, the processed image is rejected and a new one is grabbed, introducing a sort of feedback.

VII. CONCLUSIONS

This article describes a vision system useful for autonomous forklift vehicles in industrial environments. The application is based on accurate pallet segmentation based on its colour and geometric characteristics. A trajectory to fork the pallet is calculated based on its 3D position and orientation.

The pallet segmentation is one of the key points of the process. There are more powerful algorithms of feature selection and better discrimination functions than the linear ones that have been used. Although the results obtained with linear tools have been quite accurate, we are interested in testing other segmentation methods, like textures, that could improve the pallet segmentation when it is far from the vehicle. Another aspect related with the vision problem is the scene illumination. Moreover, the system was tested in light uncontrolled environments with the aim of observing its robustness, but the obtained results were really discouraging. Once more, a good illumination system is really required to obtain accurate results.

Related to 3D computation, other calibration algorithms should be tested to survey their accuracy. Moreover, the trajectory proposed is robust but quite simple so that some other more complex, like cubic splines, should be studied.

Finally, we are also thinking in a continuous vision feedback system to adapt the trajectory dynamically. Thus, erroneous pallet detection could be isolated. In summary, a first approximation to the problem has been developed. Hereafter, the aim is to develop a robust system based on the methods here described.

VIII. REFERENCES

[1] B.G. Batchelor, F.M. Waltz, *Machine Vision Systems Integration in Industry*, The International Society for Optical Engineering, 1991, vol. 1386.

[2] J.Salvi, E. Mouaddib and J. Batlle, "An overview of the advantages and constraints of coded pattern projection techniques for autonomous navigation," *IEEE Int. Conf. on Intelligent Robots and Systems*, sept. 1997, vol. III, pp. 1264-1271.

[3] M. Dash and H. Liu, "Feature selection for classification", *Intelligent data analysis*, vol. 1(3), 1997.

[4] J.Freixenet, J.Martí, X.Cufí and X.Lladó, "Use of decision trees in colour feature selection. Application to object recognition in outdoor scenes," in *Proc. of the IEEE Int. Conference on Image Processing (ICIP)*, Sept. 2000, vol. 1 pp. 800-803.

[5] G.J. McLachlan, *Discriminant analysis and statistical pattern recognition*, Wiley-InterScience Publication, 1992.

[6] X.Lladó,J.Martí,J.Freixenet and X.Muñoz, "Features selection using genetic algorithms," *3rd Catalan Conference on Artificial Intelligence*, Oct. 2000, pp. 152-156.

[7] W.K.Pratt, *Digital Image Processing*, 2on Edition. Wiley-InterScience Publication, 1991.

[8] R.C. Aggarwal, R.K. Shevgaonkar and S.C Sahasrabudhe, "A fresh look at the Hough transform," *Pattern Recognition Letters*, vol. 17, pp 1065-1068, 1996.

[9] R.Y. Tsai. "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE Journal of Robotics and Automation*, aug. 1987, vol. 3, no.4, pp. 323-344.

[10] O. Faugeras, Three-Dimensional Computer Vision: a Geometric Viewpoint, The MIT Press, 1993.

Implementació d'un algorisme paral·lel de seguiment de múltiples microrobots mitjançant VHDL

Xavier Armangué, Rafael García, Joan Batlle, Xavier Cufí,
Josep Lluís de La Rosa, Pere Ridao

Dep. Electronics, Informatics and Automation
Institute of Informatics and Applications
University of Girona
Av. Lluís Santaló, s/n, E-17071 Girona (Spain)
{armangue,rafa,jbatlle,xcuf,peplluis,pere}@eia.udg.es

Associació Catalana per a la Intel·ligència Artificial
2on Congrés Català d'Intel·ligència Artificial
CCIA 1999, Girona, Spain
October 25-27, 1999

Acceptance Letter

Assumpte: CCIA'99: Formato Poster
Data: Wed, 06 Oct 1999 16:19:40 +0200
De: Orlando C. Contreras N. <orlando@eia.udg.es>
A: armangue@eia.udg.es

Benvolgut company/a,

El comitè organitzador del CCIA'99 l'informa que vostè disposa d'un plafó per la presentació del vostre article. El plafó és vertical amb una superfície útil de 1 mts d'amplada i 1,80 mts d'altura. Els plafons es muntaran agrupats formant estructures.

Cordialment,

Orlando
Comitè Organitzador CCIA'99

Implementació d'un algorisme paral·lel de seguiment de múltiples microrobots mitjançant VHDL

X. Armangué, R. García, J. Batlle, X. Cufí, J.Ll. de la Rosa, P. Ridao

Institut d'Informàtica i Aplicacions
Dep. d'Electrònica, Informàtica i Automàtica
Universitat de Girona, Av. Lluís Santaló, s/n, E-17003 Girona (España)
e-mail: {armangue, rafa, jbatlle, xcuf, peplluis, pere}@eia.udg.es

Resum

Utilitzant una placa de processament d'imatges que disposa d'una FPGA programable amb VHDL, s'ha implementat un algorisme que permet el seguiment de 5 microrobots i d'una pilota. S'utilitza aquesta placa per poder localitzar els jugadors 50 vegades/segon. Aquest mateix algorisme implementat per software dóna un rendiment molt baix degut al seu alt cost computacional.
La placa realitza la tasca d'etiquetar els píxels i l'algorisme s'encarrega de localitzar els robots. Quan es té un robot localitzat es centra una finestra de seguiment sobre el jugador i es calcula el seu centre de masses. Les coordenades dels jugadors són enviades a un sistema de control encarregat de prendre les decisions necessaris per cada un dels jugador.

1 Introducció

Dins el Departament d'Electrònica, Informàtica i Automàtica de la Universitat de Girona s'ha desenvolupat una placa de processament d'imatge [1], que permet la implementació d'algorismes paral·lels mitjançant un llenguatge de desenvolupament de hardware: el VHDL. Aquest llenguatge permet la programació concurrent mitjançant la creació de processos que s'executen en paral·lel. Les passes a seguir per tal de desenvolupar un algorisme són molt semblants a la programació tradicional, però diferenciant-se en que finalment, en lloc d'un codi executable, es sintetitza una lògica que queda mapejada en hardware sobre un dispositiu programable. Aquest sistema de visió s'utilitzarà per realitzar el seguiment d'un equip de microrobots cooperants que jugen a futbol [2].

És un requeriment indispensable del sistema el que funcioni en temps real, de tal manera que la única limitació del seu rendiment vingui fixada per l'estàndard de vídeo (50 imatges/segon). Aquest sistema serà utilitzat per l'equip de la Universitat de Girona que participa a la RoboCup F-180 League on competeixen 5 contra 5 robots [3].
Per arribar a aquest objectiu cal realitzar un preprocessament de la imatge, a continuació aplicar un algorisme per trobar i fer els seguiment dels jugadors i finalment enviar les coordenades al sistema que s'encarrega del control dels jugadors. En la literatura es poden trobar diferents implementacions per solucionar aquest problema [4,5]. Nosaltres hem optat per un preprocessat de la imatge consistent en etiquetar la imatge en funció del valor RGB de cada píxel. La forma d'etiquetar la imatge està lligat a l'arquitectura de la nostra placa de processament.
Les dimensions i colors del camp i dels robots estan fixats per les normes de la competició. El sistema que utilitza l'equip de la Universitat de Girona per realitzar la competició consta de:

- Un camp de joc.
- Una camera 3 CCD situada a sobre del camp sobre la vertical del centre del camp.
- Un PC encarregat del sistema de visió per trobar la posició dels jugadors i de la pilota. Aquest PC té instal·lada la tarja de processament.
- Una segon PC que llegeix les coordenades dels jugadors i de la pilota i fent servir tècniques d'intel·ligència artificial genera les consignes que s'han d'enviar a cada robot via radio.
- Un transmissor de radio.

La carcassa dels jugadors té forma cilíndrica amb un diàmetre de 12,5 cm. A la tapa s'hi ha col·locat unes taques de colors per identificar els jugadors. Al centre hi ha un cercle de color amb el diàmetre equivalent de 4,1 cm que pot ser de color groc o blau.

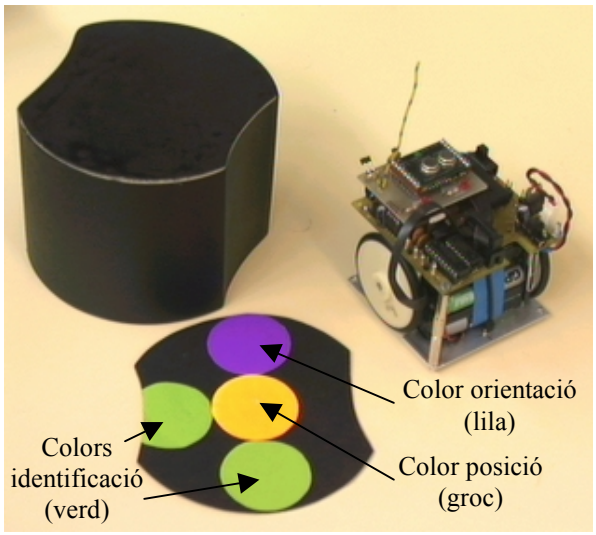


Figura 1 De esquerra de dreta: carcassa del robot; tapa la carcassa; robot

Per determinar quina és la orientació del jugador s'utilitza un cercle de color lila que és comú a tots els jugadors del mateix equip. Per fer la identificació de cada un dels jugadors s'utilitza un tercer color, el verd, en funció del número de cercles de color verd i de la seva posició es pot conèixer quin jugador és. De la identificació de cada un dels jugadors se n'encarrega una aplicació software i es per això que el sistema de seguiment no cal que segmenti el color verd. A la Figura 1 es pot observar un robot amb la seva carcassa i els colors que l'identifiquen.

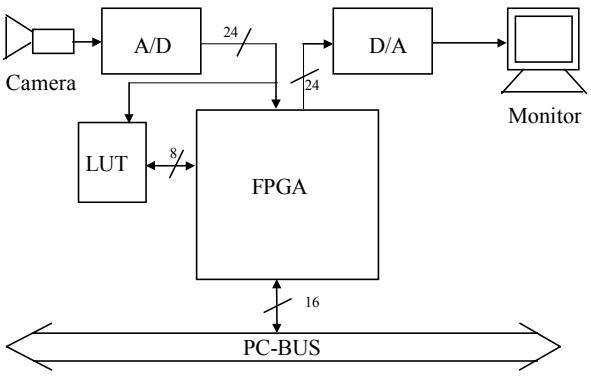


Figura 2 Diagrama de blocs de la tarja de processament

2 Definició de subsistemes

Per realitzar el seguiment dels 5 jugadors cal particionar el sistema per dividir les diverses funcions a realitzar.

- *Senyals de control*.
- *Comunicació amb el PC*.
- *Segmentació de la imatge*.
- *Seguiment de 5 jugadors i la pilota*.

2.1 Senyals de control

Aquest subsistema s'encarrega, bàsicament de generar les senyals de sincronisme horitzontal (SYNC_H), i de sincronisme vertical (SYNC_V) i les coordenades del píxel actual. La senyal SYNC_H serveix per indicar a la resta de subsistemes que ho necessitin quan s'ha processat una línia sencera de píxels. La senyal SYNC_V és l'equivalent a la senyal SYNC_H però ens indica quan s'ha processat un quadre sencer de la imatge. Aquest procés també s'encarrega d'anar comptant els píxels a mesura que van arribant per la camera per saber en tot moment quina és la coordenada del píxel actual.

2.2 Comunicació amb el PC

EL procés encarregat de la comunicació amb el PC realitza 3 funcions:

- Llegir les senyals de control enviades pel PC i enviar-les al subsistema afectat.
- Recollir les dades calculades pels altres subsistemes i enviar-les al PC quan aquest llegeixi de la tarja.
- Emplenar la memòria de la placa (LUT) amb la taula de segmentació generada pel PC.

2.3 Segmentació de la imatge

Aquest subsistema és l'encarregat de processar la imatge d'entrada provinent de la camera. Per realitzar la segmentació s'utilitza una taula de segmentació que s'emmagatzema a una memòria (LUT) de la tarja de processament. Per a cada valor RGB s'assigna una etiqueta. Per aquesta aplicació cal utilitzar 3 etiquetes: per la posició del jugador, per l'orientació dels jugador i per posició de la pilota. Per eliminar soroll les etiquetes són filtrades. El filtre utilitzat és una operació morfològica open 2x2.

A més a més, aquest subsitema es l'encarregat de generar una imatge de sortida per poder visualitzar el resultat del seguiment. S'encarrega dibuixar les finestres de seguiment dels jugadors i de la pilota per

tal de tenir una realimentació visual del funcionament del sistema.

2.4 Algorisme de seguiment

L'algorisme utilitzat per localitzar el cinc jugadors és el següent:

1. A mesura que arriben píxels es mira si estan etiquetats amb l'etiqueta del color de l'equip. Quan arriben un conjunt de píxels suficients per poder afirmar que s'ha localitzat un jugador es finalitza la busqueda en aquesta imatge i es guarda la coordenada del jugador.
2. Quan es comença la captura de la següent imatge es centra una finestra sobre la posició on s'havia localitzat el primer jugador. Per trobar el segon jugador es torna a buscar píxels etiquetats amb el color de l'equip però que quedin fora de la finestra de seguiment del primer jugador. En el moment que el nombre de píxels amb l'etiqueta de l'equip sigui superior a un llindar establert es guarden les coordenades del segon jugador.
3. Quan arribin els píxels de dins de la finestra del primer jugador cal calcular el centre de masses dels píxels etiquetats amb el color de la posició del jugador i el centre de masses dels píxels etiquetats amb el color de l'orientació del jugador. Amb aquest dos punts es té una mesura de la posició i de l'orientació del jugador.
4. Aquest procés es repeteix fins a localitzar tots els jugadors tenint en compte que per trobar un nou jugador no es tenen en compte els píxels que estan dins de les finestres dels jugadors ja localitzats.
5. Passades 5 imatges, si tot ha anat bé, ja s'han localitzat tots els jugadors i a la sisena imatge ja es disposa de la posició i orientació de tots els jugadors. Si en algun moment es perd la posició d'algun jugador es tornen a buscar píxels etiquetats amb la posició del jugador a dins de tot el camp i fora de les finestres dels altres jugadors.

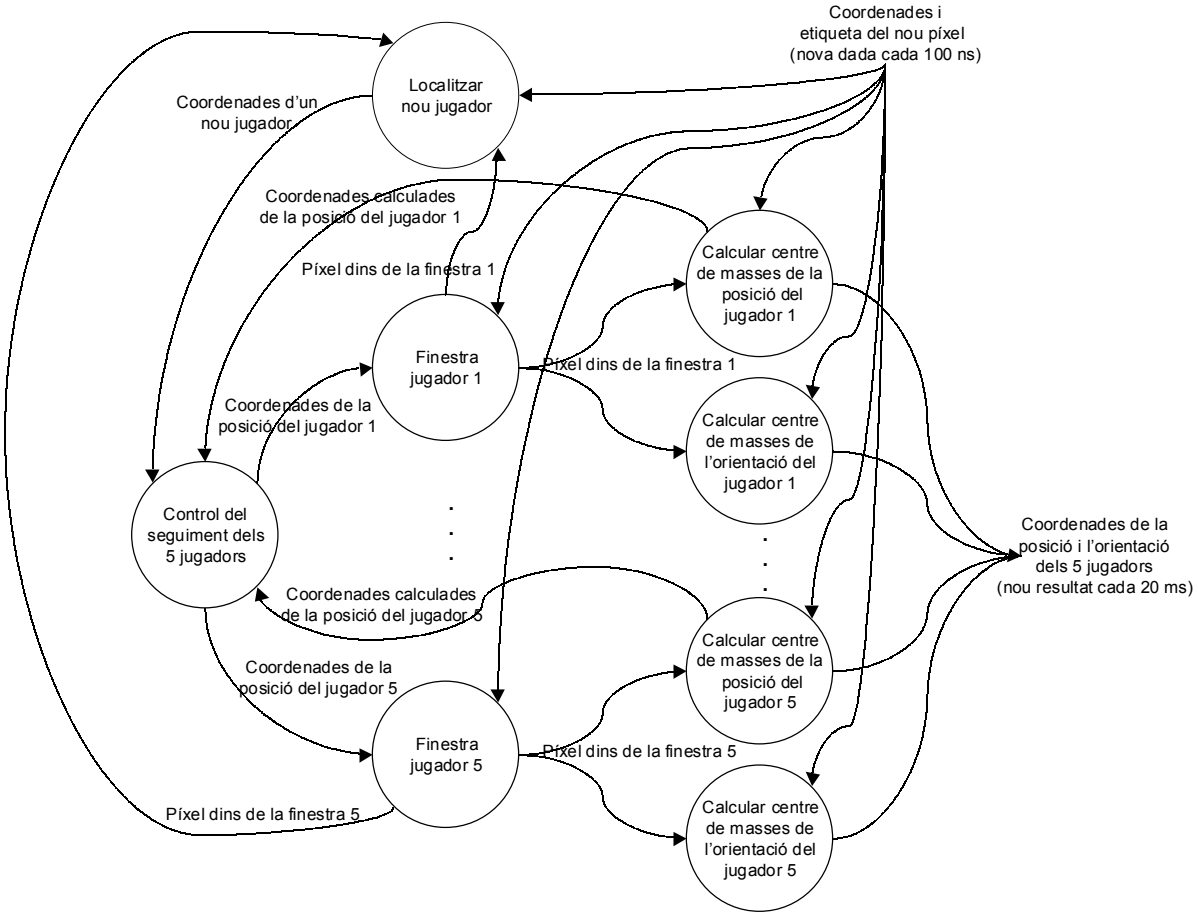


Figura 3 Diagrama de flux de dades pel seguiment del 5 jugadors. Tots els blocs treballen en paral·lel.

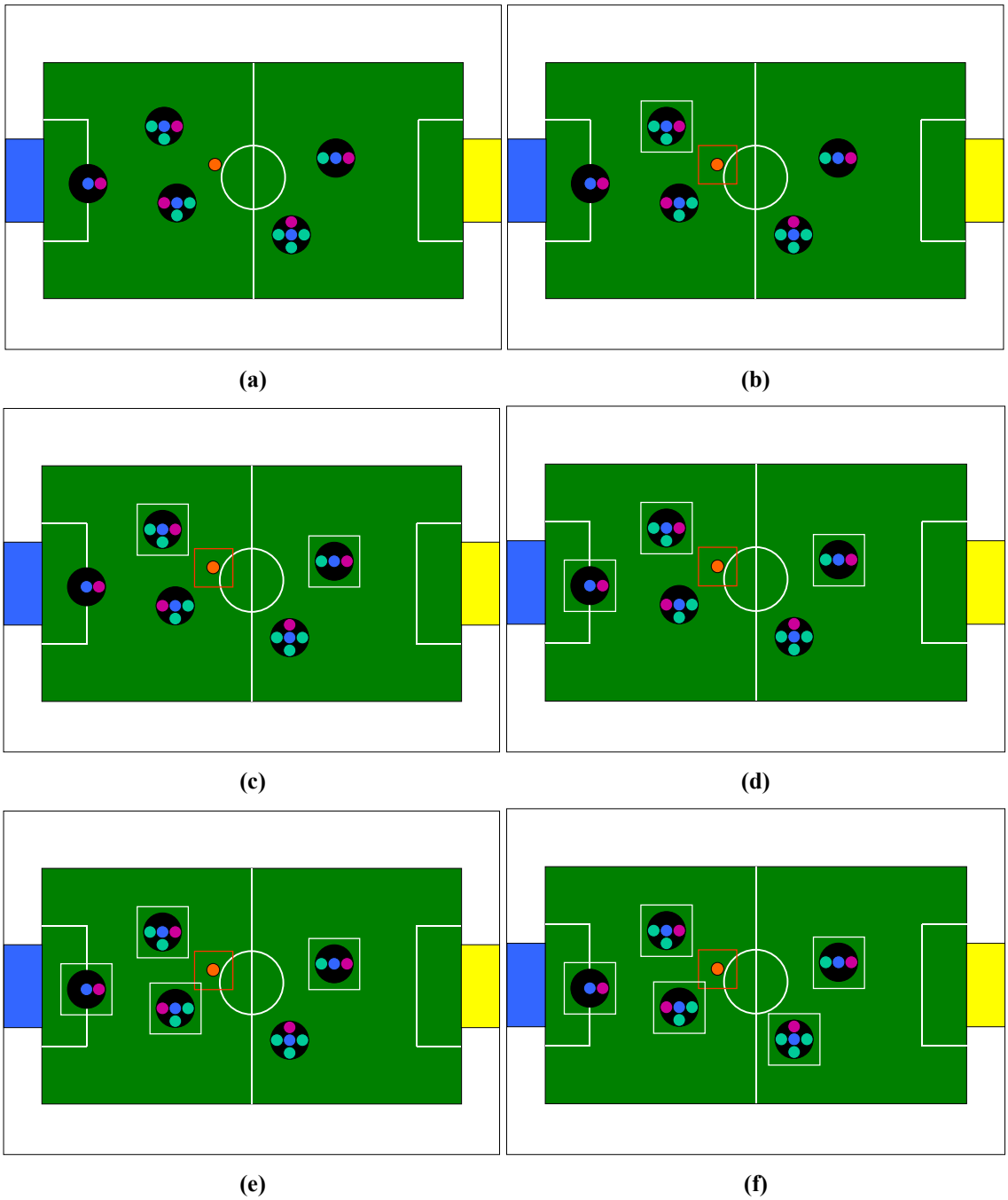


Figura 4 Evolució temporal de l'algorisme: localització inicial dels jugadors i la pilota. (a) A la primera imatge cal localitzar el primer jugador i la pilota (b) A la següent imatge ja s'ha trobat el primer jugador i la posició de la pilota (c) A la tercera imatge es troba el segon jugador (d) Després es localitza el quart jugador (e) Passades 5 imatges ja es tenen els 5 jugadors localitzats (f) A la sisena imatge ja s'està realitzant el seguiment.

3 Visualització de la imatge processada

Utilitzant la taula de segmentació de la LUT la imatge es segmentada i només ens quedem amb els píxels del color de que indiquen quina és la posició i orientació dels jugador i amb el color de la pilota. Visualitzant la imatge etiquetada s'aconsegueix que els píxels que són de color fosc quedin ressaltats i es vegi millor si la imatge té soroll. Per l'eliminació del soroll s'ha d'activar el filtratge morfològic.

Si observem una imatge de tot el camp com la Figura 5 es por observar que hi ha tota una zona on no cal realitzar el seguiment dels jugadors perquè queda fora dels límits del camp, es per això que cal indicar a la tarja de processament on està situat el camp. Els píxels que queden fora del camp són depreciats, no s'utilitzen per fer el seguiment dels jugadors.

Figura 5 Seguiment dels 5 jugadors i de la pilota per la tarja de processament

Un cop es té la imatge segmentada, etiquetada i filtrada es pot començar a aplicar l'algorisme de seguiment.

Sobre cada jugador que es realitza el seguiment es dibuixa una finestra de color blanc. Sobre la pilota es dibuixa una finestra de color vermell (veure Figura 5).

El host (ordinador on està connectada la tarja de processament) llegeix les coordenades de les posicions i orientacions dels jugadors i de la posició de la pilota. Aquestes dades es passen al sistema de control per decidir quina acció ha de prendre cada jugador. Les dades llegides pel host es representen visualment a la Figura 6.

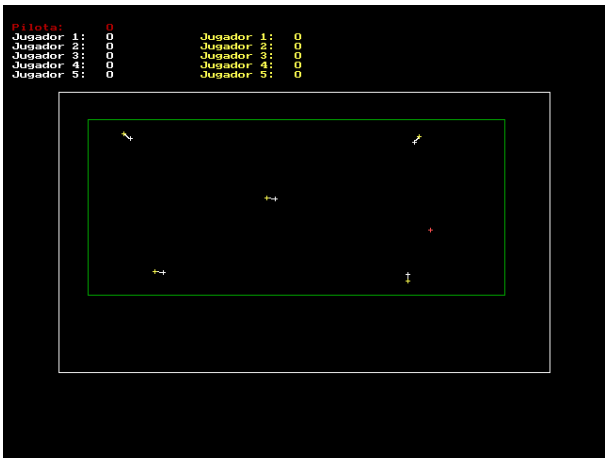


Figura 6 Seguiment de 5 jugadors i de la pilota pel host

4 Anàlisi temporal

A la Figura 7 es pot observa quin és e diagrama de temps de l'algorisme.

- Captura de la imatge: A l'instant de temps t=0 es comença de capturar la imatge, després de capturar les 287 línies amb informació de vídeo es finalitza la captura (t=18368 μs). La mitja línia que també conté informació de vídeo no es té en compte per no complicar la lògica.
- Etiquetatge: L'etiquetatge de la imatge va desfasada respecte la captura 20ns aquest retard és produeix perquè cal accedir a la LUT per obtenir les etiquetes i el temps d'accés a la LUT és de 20 ns. L'etiquetatge de la imatge finalitza al mateix tems que la captura.
- Filtratge: Una operació morfològica *open* 2x2 implica aplicar un *erode* a la imatge original, i a continuació un *dilate* a la imatge resultant. La concatenació d'aquestes dues operacions genera el mateix retard (en termes dels píxels que ens va passant la càmera) que si passéssim un únic filtre 3x3. Per poder passar un filtre 3x3 no es pot donar una valor al píxel central fins que no es coneguin els valors de tots els elements de la matriu 3x3. El fet de passar el filtre fa que es produeixi un retard d'una línia i un píxel respecte a la imatge capturada (64,1 μs). El filtratge finalitza quan acaba la captura de la imatge.
- Acumulació de coordenades: a mesura que les etiquetes són filtrades, quan es troba una etiqueta que pertany algun jugador o a la pilota les seves coordenades són acumulades i s'incrementa el número de píxels

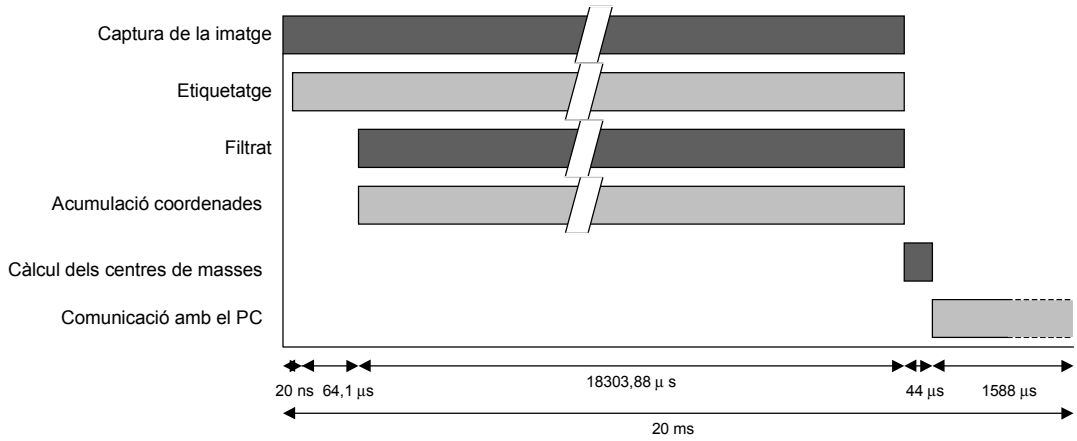


Figura 7 Diagrama de temps

- Càlcul del centre de masses: Al finalitzar l'acumulació de les coordenades ja es pot començar a calcular els centres de masses. Per calcular-lo cal dividir la coordenada acumulada per número de píxels de l'objecte corresponent. Com que es fa el seguiment de 5 jugador i de la pilota cal trobar 11 coordenades (5*posició jugador+5*orientació jugador+1*posició pilota) i com que cada coordenada està formada per 2 valors cal realitzar 22 divisions. Per seleccionar els valors a dividir i realitzar la divisió són necessaris 20 cicles del rellotge de 10 MHz, per tant per realitzar les 22 divisions calen 44 µs (22 divisions*20 cicles*100 ns).
- Comunicació amb el PC: quan ja s'han calculat els centres de masses el PC els ha de llegir. El temps que pot tardar a llegir les dades depèn completament de l'ocupació del PC i del sistema operatiu. Si el PC està completament dedicat a llegir les dades de la tarja i tenint en compte que el BUS ISA funciona a 10 MHz, la lectura del centre de masses pot tardar 3,3 µs (22 dades*100 ns+11 dades per alliberar el BUS*100ns). Donat que treballem amb un sistema operatiu multitasca però no temps real (Windows 98), és difícil estimar el temps necessari perquè el host porti a terme la lectura de les dades. Això dependrà no només de la velocitat de la CPU, sinó també de la planificació de l'execució dels threads, de la prioritat d'aquests i de la càrrega del sistema, generalment el host està realitzant altres tasques com visualitzar la imatge de vídeo, filtrar les dades obtingudes, realitzar prediccions sobre el moviment dels jugadors i de la pilota. Tot això pot retardar la lectura de les dades. El retard màxim que es pot permetre el host és de 20 ms, si tardés més es perdrien dades. Les proves realitzades han demostrat que no hi ha pèrdua rendiment a menys que es despleguin els menús de l'aplicació.

5 Experiments realitzats

Per comprovar quina és la precisió del sistema s'han realitzat tota una sèrie proves per comprovar la precisió de les dades. Primer de tot cal calcula quina és la relació entra les dades obtingudes i el món real. Les distàncies dels objectes d'una imatge es mesuren en píxels, coneixen les mides del camp en píxels i les mides del camp amb centímetres es pot trobar la següent correspondència:

$$\text{Mides del camp en píxels}: 444\text{x}187\,\text{píxels}$$

$$\text{Mides del camp en centimetres}: 274\text{x}152{,}5\,\text{cm}$$

$$\frac{1}{444} = \frac{x}{274} \Rightarrow x = 0{,}617\,\text{cm}$$

$$\frac{1}{187} = \frac{y}{152{,}5} \Rightarrow y = 0{,}815\,\text{cm}$$

Veient aquestes dades, una variació d'un píxel és de 0,617 cm en coordenades X i de 0,815 cm en coordenades Y.

Un altre aspecte a tenir es compte és el problema del càlcul d'angles. La imatge que genera la camera té una relació de 4:3, en canvi la imatge processada per la tarja i sobre la qual es realitzen les mesures té 522x287 píxels. Cal realitzar una correcció de les dades per corregir aquesta deformació.

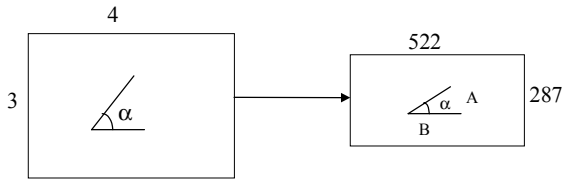


Figura 8 Error en el càlcul dels angles

S'ha de escalar la imatge de la tarja perquè compleixi les proporcions de 4:3. Les operacions a realitzar per corregir la mesura dels angle són les següents:

$$\frac{4}{3}=\frac{X}{288} \qquad \frac{x}{384}=\frac{B}{522} \qquad y = A$$

$$X=\frac{4*288}{3}=384 \qquad x=\frac{384B}{522}$$

$$\operatorname{tg}\alpha=\frac{y}{x} \qquad (1)$$

$$\alpha=\operatorname{tg}^{-1}\frac{y}{x}$$

S'ha realitzar una aproximació per treballar amb el mateix número de línies d'una quadre d'una imatge de l'estàndard de vídeo, se suposa que la tarja de processament mesura 288 línies.

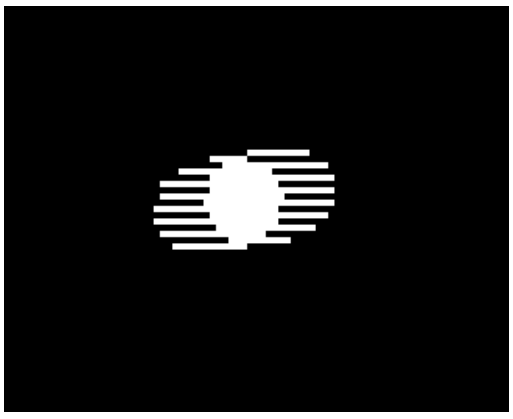


Figura 9 Pilota en moviment: zoom sobre una imatge entrellaçada de 768×576 pixels.

La imatge de vídeo que està entrellaçada, cada imatge la formen dos quadres, un amb les línies imparells i l'altre amb les línies parells. Quan es captura una imatge d'un objecte en moviment i s'entrellacen els dos quadres el resultat és que s'obté un objecte deformat perquè els dos quadres s'han capturat en instant de temps diferents. A la Figura 9 es pot observar quin és el resultat de capturar una imatge amb la pilota en moviment. Si es calcula el centre de masses d'aquesta imatge en donarà un punt intermig entre el centres de masses del primer i del segon quadre. Per evitar aquest problema es calcula el centre de masses a cada quadre, d'aquesta manera s'obté una mesura més exacta de la posició de la pilota en moviment i encara que es perdi resolució vertical.

6 Resultats

Per veure com són les dades obtingudes pel seguiment s'han recollit dades sobre la posició de la pilota i dels jugadors.

Si es col·loca la pilota en una posició estàtica, en teoria, la mesura hauria de ser sempre la mateixa. A la pràctica es produeixen petites variacions. A la Taula 1 es poden observar els principals valors estadístics de les dades obtingudes. Observant aquestes dades es veu que la variància de les coordenades és molt petita.

	Coordenada X	***Coordenada Y***
Màxim	419	199
Mínim	416	195
Mitjana	416,84	196,676
Variància	0,3679	0,3002

Taula 1 Estadístics de la posició de la pilota (mesures en píxels)

Realitzant la mateixa prova amb un jugador els resultats obtinguts són mot similars (veure Taula 2 i Taula 3).

	Coord. X posició	***Coord. Y posició***
Màxim	*415*	*118*
Mínim	*413*	*117*
Mitjana	*414,0689*	*117,9655*
Variància	0,0785	0,03352

Taula 2 Estadístics de la posició del jugador (mesures en píxels)

	Coord. X orientació	***Coord. Y orientació***
Màxim	*417*	*124*
Mínim	*415*	*121*
Mitjana	*416,2827*	*122,8069*
Variància	0,3153	0,2124

Taula 3 Estadístics de l'orientació del jugador (mesures en píxels)

A continuació s'ha comprovat quina és la resposta del sistema de seguiment amb objectes mòbils. A la Figura 10 es representen les dades obtingudes de la pilota, inicialment està parada, un moment donat se li dona un cop, avança, realitza un doble rebot contra una cantonada del camp i surt més esmorteïda. Els resultats

són molt bons ja que la pilota segueix trajectòries rectilínies i les dades obtingudes descriuen trajectòries rectilínies.

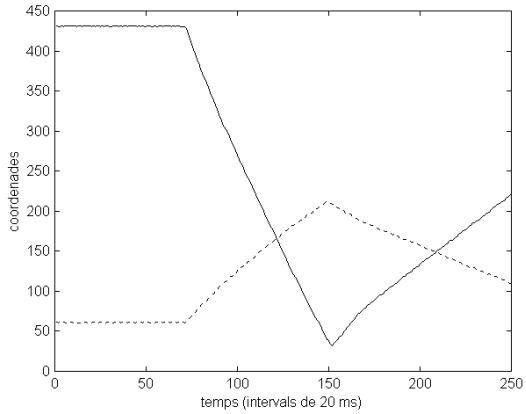


Figura 10 Coordenades d'una pilota en moviment: la línia contínua correspon a la coordenada X, la discontínua correspon a la Y

La prova realitzada amb el jugador ha consistit en fer-li donar una volta i recollir dades sobre les coordenades de la posició, de l'orientació i realitzar el càlcul de l'angle de l'orientació. La Figura 11 estan representades les dades d'un jugador realitzant una trajectòria circular, amb aquestes dades s'ha calculat l'angle de l'orientació de la Figura 12. Les coordenades tenen una variació molt petita però l'angle resultant té unes oscil·lacions molt grans, que farien difícil el control dels robots.

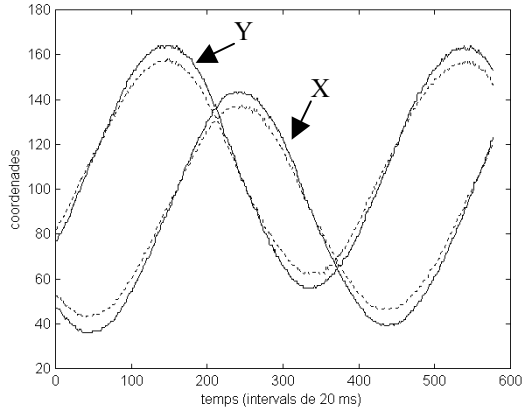


Figura 11 Coordenades d'un jugador realitzant una trajectòria circular: la línia contínua correspon a les coordenades de posició, i les discontínues corresponen a l'orientació

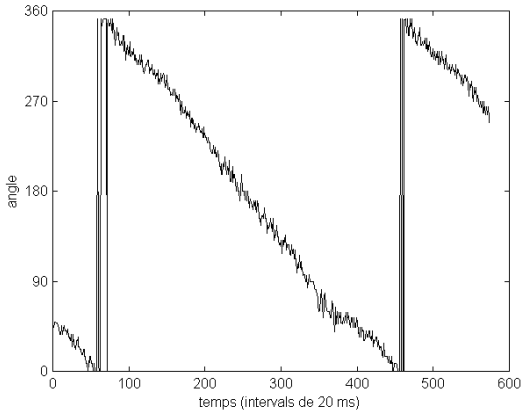


Figura 12 Angle d'un jugador realitzant una trajectòria circular

Per mirar de suavitzar aquests errors en el càlcul de l'angle s'ha aplicat un filtre d'ordre 1 [5] a les mesures de la coordenades. L'equació d'aquest filtre és la següent:

$$\widetilde{X}_k = \widetilde{X}_{k-1} + k \cdot \left(X_k - \widetilde{X}_{k-1}\right) \qquad (2)$$

L'estimació d'una coordenada en un instant temps *k* és funció de l'estimació anterior i de la diferencia entre la dada actual i l'estimació anterior. El paràmetre k s'utilitza per suavitzar més o menys l'efecte del filtre, per valors a prop de 1 es dona més importància a la dada actual i per valors a prop del 0 es dona més importància a l'estimació anterior. La Figura 13 s'observa quin és el resultat de l'aplicació d'aquest filtre a les coordenades d'orientació i posició amb una k=0.2.

Si es comparen les dades obtingudes del càlcul de l'angle de la Taula 4 es pot observar que la variància disminueix notablement utilitzant el filtre.

	Angle	***Angle Filtrat K=0.2***
Màxim	90	75,3767
Mínim	61,1134	66,1365
Mitjana	71,2723	71,3421
Variància	34,4243	3,5360

Taula 4 Estadístics de l'angle del jugador (mesures en graus)

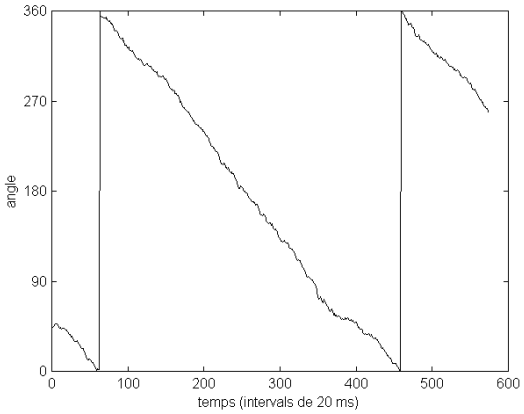


Figura 13 Angle de l'orientació filtrat

7 Conclusions

Els principals problemes que han sorgit durant la implementació són la limitació d'espai de programació de la FPGA i la necessitat de que el sistema de seguiment funcioni de 50 imatges/segon. Com a conseqüència d'aquesta limitació d'espai les seguiment dels jugadors contraris no es pot realitzar amb aquesta FPGA.

Els objectius que s'han complert són:

- Segmentar la imatge.
- Realitzar el seguiment dels 5 jugadors de l'equip propi per trobar-ne la posició i l'orientació.
- Realitzar el seguiment de la pilota.
- Enviar totes les coordenades al PC.
- Visualitzar els resultats.
- Aconseguir un temps de cicle de 20 ms per tot l'algorisme.

Agraïments

Els autors voldrien agrair la seva ajuda i col·laboració de tots els companys del laboratori, i en especial a Josep Tomàs, Lluís Magí i Tomàs Medel.

Referències

[1] Garcia, R., De La Rosa, J.Ll., Oller, A., Batlle, J., Figueras, A., Ramon, J.A. "A Fast Vision System for Tracking in Robotic Soccer" *Proceedings on Micro-Robot World Cup Soccer Tournament*, (MIROSOT'98), Paris, Francia,1998.

[2] Kitano, H., Kuniyoshi, Y., Noda, I., Asada, M., Matsubara, i Osawa, E-I. "Robocup: a challenge problem for AI," *Artificial Intelligence Magazine*, vol.18(1), pp.73–85, 1997.

[3] Sargent, R., Bailey, B., Witty, C., i Wright, A. Dynamic object capture using fast vision tracking, *Artificial Intelligence Magazine*, vol. 18:1, pp. 65–72, 1997.

[4] Hong, C.S., Chun, S.M., Lee, J.S., and Hong K.S., "A Vision-Guided Object Tracking and Prediction Algorithm for Soccer Robots*", IEEE International Conference on Robotics and Automation*, vol.1, pp. 346-351, 1997.

[5] Matsumoto,Y., Shibata, T., Sakai, K., Ibana, M., and Inoue, H., "Real-Time Color Stereo Vision System for a Mobile Robot based on Field Multiplexing", *in Proceedings of the IEEE International Conference on Robotics and Automation,* vol.3, pp. 1934-1939, 1997

[6] Isermann, R. "Digital control systems," Springer-Verlag, 1991.