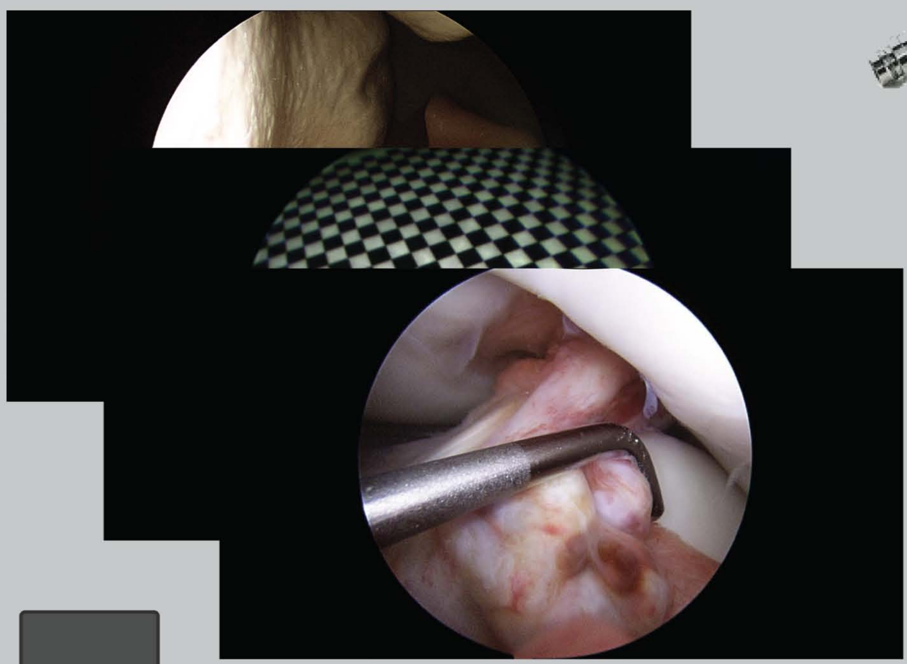
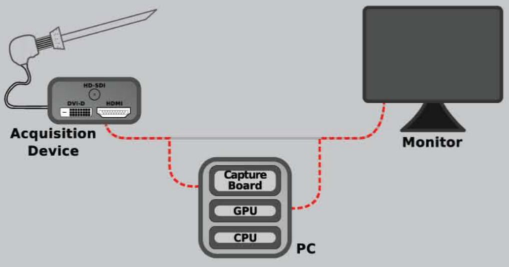




Rui Jorge Melo Teixeira

CAMERA CALIBRATION AND REAL-TIME IMAGE PROCESSING IN HETEROGENEOUS ARCHITECTURES - APPLICATION IN MEDICAL ENDOSCOPY

Tese de Doutoramento em Engenharia Electrotécnica e de Computadores, Ramo de Especialização em Automação e Robótica, orientada pelos Professor Doutor João Barreto e Professor Doutor Gabriel Falcão e apresentada ao Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Setembro 2014

· U C ·

UNIVERSIDADE DE COIMBRA

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

PhD Thesis

Camera Calibration and Real-Time Image Processing in Heterogeneous Architectures - Application in Medical Endoscopy

Rui Jorge Melo Teixeira

September 15, 2014

Camera Calibration and Real-Time Image Processing in Heterogeneous Architectures - Application in Medical Endoscopy

By

Rui Jorge Melo Teixeira

Advisor: Prof. Dr. João Barreto **Co-Advisor**: Prof. Dr. Gabriel Falcão

Department of Electrical and Computer Engineering
Faculty of Science and Technology,
University of Coimbra

September 15, 2014

Contents

Acknowledgements

The first and foremost person I would like to acknowledge is my advisor, João Barreto. It was only through the motivation, dedication, guidance and support he transmitted during this journey that this work was possible. To my co-advisor, Gabriel Falcão, I want to thank for the dedication and support, specially during the most critical periods.

Throughout these years I have always been always able to count with the help and support of my research colleagues and friends. To Michel, Miguel, Francisco, Abed, José, Luís, Vítor and Cristovão, among many others, a big thank you.

At a more personal level, a very special thank to Cátia, who walked that long walk beside me, supportive as always. It would have been impossible without her.

I also want to thank the Institute for Systems and Robotics (ISR) and the Department of Electrical and Computer Engineering for providing a wonderful research environment. Finally, a special acknowledge goes to the Portuguese Foundation for Science and Technology (FCT), that has funded this research work through the Ph.D. scholarship SFRH/BD/79359/2011.

List of Tables

List of Figures

Acronyms

MIS	Minimally Invasive Surgery	i
OR	Operating Room	i
FOV	Field of View	i
RD	Radial Distortion	i
HD	High-Definition	2
SD	Standard-Definition	2
CCD	Charge-Coupled Device	7
DM	Division Model	11
SIC	Single Image Calibration	13
DLT	Direct Linear Transformation	18
RANSAC	RANdom SAmple Consensus	20
DOF	Degrees of Freedom	23
UFL	Uncapacited Facility Location	30
HFL	Hierarchical Facility Location	30
GPU	Graphics Processing Unit	45
EKF	Extended Kalman Filter	45
RMS	Root Mean Square	59
CUDA	Compute Unified Device Architecture	61
DCT	Discrete Cosine Transform	62
GPGPU	General-Purpose Computing on Graphics Processing Units	62
VLSI	Very-large-scale Integration	63
FPGA	Field-programmable Gate Array	63
PC	Personal Computer	64
CPU	Central Processing Unit	64

Abstract

In the context of Minimally Invasive Surgery (MIS), developing image processing algorithms for endoscopic video poses several challenges over conventional camera systems. The endoscopic lens is a complex optical system that possesses some unique features, like the rotation of the scope for periscope-like view inside the anatomical cavity. Building a camera model that accounts for these changes in the image formation is of major importance for algorithms depending on accurate camera modeling, such as stereo reconstruction, visual robust localization and mapping or distortion correction for visualization improvement. In order to infer geometric information from the image the camera has to be properly calibrated. While calibration is a well known topic in the literature, few works have focused on the specificity and practicability issues of calibrating an endoscopic system in the Operating Room (OR). In endoscopy, the lens is usually assembled to the camera head right before the procedure and therefore current camera calibration approaches are not suited for use in the OR. Research on unsupervised camera calibration techniques, that potentially ease the burden of performing this procedure in a delicate environment, is of key importance to bring complex image processing algorithms to the medical endoscopy field.

Since lenses in MIS are usually very small with a wide Field of View (FOV), the Radial Distortion (RD) of the optical system is significant enough to further difficult camera calibration and affect the performance of surgeons during operation. The RD affects MIS procedures by interfering with the notion of relative depths in the imaged scene and changing the shape of anatomical landmarks and tools while the surgeon operate. The development of real-time systems for endoscopy in architectures, that allow a fast and scalable deployment of advanced image processing algorithms, is an important step for bringing practical software solutions for assisting the practitioner in the OR.

This thesis advances the state of the art in the modeling of endoscopic systems and calibration of perspective cameras with RD, with the results of such work being used in a practical real-time system for distortion correction in medical endoscopy that impacts in the performance of surgeons. The real-time requirements of image processing algorithms that aim at assisting the doctor during surgery/diagnosis lead to the development of efficient solutions in heterogeneous architectures to cope with the latest video resolutions and frame rates.

Resumo

No contexto da cirurgia minimamente invasiva, o desenvolvimento de algoritmos de processamento de imagem para vídeo endoscópico coloca vários desafios inexistentes em sistemas de câmera convencional. A lente endoscópica é um sistema óptico complexo, que possui características especiais, como a rotação da lente para possibilitar um vista similar a um periscópio no interior da cavidade anatômica. Desenvolver um modelo de câmera que tenha em atenção essas mudanças na formação da imagem é de grande importância para os algoritmos que dependem de um modelo da câmera preciso, tais como a reconstrução estéreo, a localização robusta visual e mapeamento ou a correcção de distorção para a melhoria da visualização. De forma a deduzir informações geométricas a partir da imagem, a câmera tem que ser devidamente calibrada. Enquanto que a calibração é um tema conhecido na literatura, apenas alguns trabalhos se focam na especificidade e questões práticas para calibrar um sistema endoscópico na sala de operação. Em endoscopia, a lente é geralmente montada na cabeça da câmera imediatamente antes do procedimento e portanto as técnicas atuais de calibração não são adequados para uso em bloco operatório. A investigação de técnicas de calibração da câmera sem supervisão, que potencialmente aliviam o impacto de realizar este procedimento num ambiente delicado, é de importância fundamental para levar algoritmos de processamento de imagens complexos o campo da endoscopia clínica.

As lentes em cirurgia minimamente invasivas são geralmente muito pequeno, com um amplo campo de visão, e a distorção radial do sistema óptico é significativa o suficiente para tornar ainda mais difícil a calibração da câmara e afetar o desempenho dos cirurgiões durante a operação. A distorção afeta os procedimentos interferindo com a noção de profundidade relativa na imagem e mudando a forma das referências anatômicas e ferramentas enquanto o cirurgião opera. O desenvolvimento de sistemas de tempo real para endoscopia em arquiteturas que permitão uma implantação rápida e escalável de algoritmos avançados de processamento de imagens é uma passo importante para trazer soluções práticas de software para ajudar o cirurgião no bloco operatório.

Esta tese avança o estado da arte na modelação de sistemas endoscópicos e calibração de câmeras em perspectiva com distorção radial, com os resultados a serem utilizados num sistema prático em tempo real para correção de distorção em endoscopia clinica, com impacto no desempenho dos cirurgiões. Os requisitos de tempo real dos algoritmos de processamento de imagem que visam auxiliar o médico durante a cirurgia/diagnóstico levaram ao desenvolvimento de soluções eficientes em arquiteturas heterogêneas para lidar com as últimas resoluções e frequências de vídeo.

Chapter 1

Introduction

1.1 Context and Motivation

Over the past twenty years, the computer vision community has made great advances in providing solutions to problems such as camera localization, visual tracking and stereo reconstruction for example. Many algorithms have been made tractable by the rapid increases in computational speed and memory size now available into a single computer [1]. Computer vision applied to medical imaging has appeared as a natural application of these techniques, aiming at improving health-care trough image segmentation, detection and tracking, classification and visualization ([2–6]).

One of the many medical imaging devices currently in use is the endoscope, which is used to perform MIS. MIS have gained increasing popularity in different clinical specialities, and is currently the “standard” to perform many of today’s diagnosis and surgical procedures. A non-exhaustive list of examples includes rhinoscopy, colonoscopy, laparoscopy, neurology and arthroscopy. All these specialities rely on the use of small lens to be inserted in the human body through small incisions or even natural orifices, as is the case of the recent natural orifice translumenal endoscopic surgery (NOTES) technique [7]. The captured endoscopic images allow the clinician to observe the interior of anatomical cavities and control the action of instruments manipulated from the outside.

Since visual information provides the only guidance to the practitioner, imaging quality is of extreme importance in clinical endoscopy, especially for non-experienced doctors. Over time, the MIS industry has been constantly working towards improving the visualization during medical procedures. From the introduction of digital video endoscopy in the 90’s to the current state-of-the-art visualization, we have seen remarkable improvements in light sources quality, camera technology and image resolution (with the shift from Standard-Definition (SD) to High-Definition (HD)). As HD video is becoming a standard in clinical endoscopy, state-of-the-art systems for medical endoscopy provide 1080p video streams at 60Hz. For such high resolutions and frame rates the real-time execution of image processing tasks is far from trivial, requiring careful algorithm design and optimization.

1.2 Research Topics and Contributions

In this thesis we will address the need of real-time image processing in MIS, aiming at obtaining a final system that can be used in the OR for improving the visualization in medical endoscopy. For this we will address the modeling and calibration of endoscopic cameras as a way of obtaining an accurate description of the projection model. One of the cornerstones of future MIS application based on geometric computer vision is to be able to have an accurate camera calibration across the entire surgical procedure. In this thesis we will develop a camera model that accommodates the lens periscope-like rotation, tackling the specific properties of such a complex optical system. During the research on camera calibration in high distortion camera systems, we also developed a calibration method based on the plumb-line approach that can be applied to any perspective camera with distortion, including the medical endoscope, and can ultimately leverage new applications in medical imaging technologies.

Correcting optical aberrations in endoscopy, such as the radial distortion, presents a step forward into bringing computer vision algorithms closer to real-life applications in the life sciences domain. Computer vision algorithms are complex programs that require an intuitive and scalable development platform. Current image processing software for medical endoscopy is usually deployed in dedicated hardware units and, although the high performance achieved, those platforms are usually closed and difficult to develop and customize. In this thesis we tackle the problem of real-time image processing and develop a system for endoscopic image processing that uses a hybrid CPU+GPU architecture and is able to leverage the GPU computational power to run a demanding visualization application. This results in a system providing a visualization improvement feature that can be deployed as an add-on to current endoscopic equipment and will pave the way to the deployment of other real-time image processing algorithms for endoscopy.

The contribution of this thesis can be summarized in the following points:

- **Modeling of the endoscopic lens** [5]: We derived a new projection model for the rigid endoscope that models the lens rotation as a rotation of the image plane around a fixed axis perpendicular to the camera sensor. This new model allows to better describe this complex optical system's projection and therefore it improves the performance of geometric-based algorithms that require accurate camera calibration.

- **Plumb-line calibration from a single image** [6]: Throughout the study of calibration solutions for medical applications we developed the first plumb-line calibration method that can estimate the camera intrinsics from a minimum of 3 image lines. This algorithm is able to calibrate a camera in a completely unsupervised manner and can be applied to any camera with significant radial distortion even for other domains.

- **System for radial distortion correction that accommodates lens rotation** [5]: In this thesis we created a system for radial distortion correction specially tailored for use in medical endoscopy. This system accurately tracks the lens rotation (through the tracking of the image's useful region) and updates the calibration parameters to accurately compensate the radial distortion correction in all situations.

- **Heterogeneous implementation for real-time execution** [8]: While developing the system mentioned above, we paid special attention to the architecture and implementation details in order to handle the high processing demand required for this application. We demonstrate that an hybrid solution, where the workload is distributed across CPU and GPU concurrently and efficient memory access patterns are implemented, allows the system to process a HD video stream (input resolution of 1920×1080 at a frequency of $60Hz$) at a frame-rate of 250*fps*.

In addition, we actively contributed to the planning, execution and analysis in a study of the impact of the proposed system in arthrosocpic surgery (the branch of endoscopy that relates to procedures in the anatomical joints) simulation using a knee model [9]. The radial distortion correction system was used to validate the influence of radial distortion in the surgical performance of surgeons undergoing training. This study showed that the depth perception is increased with the use of the developed system and that this translates into a better surgical outcome in a basic arthroscopic task.

1.3 Thesis Outline

This thesis is outlined as follows: Chapter 2 provides an overview of the image formation model and radial distortion, presenting the new endoscopic projection model that takes into account the rotation between the lens and the camera sensor. Chapter 3 presents the camera calibration problem and details the unsupervised calibration from the image of lines developed to recover the camera intrinsics from a single image. Chapter 4 addresses the problem of radial distortion correction in endoscopy, presenting the developed system that copes with the specific needs in endoscopy. Chapter 5 details the developed system's architecture, outlining the implementation details that allow the system to outperform dedicated hardware solutions in terms of image processing throughput. Chapter 6 presents an experimental study in a population of doctors about the system's clinical benefits during endoscopic procedures. Finally Chapter 7 draws the conclusions of this thesis and presents some possible future research directions.

1.4 Notation

Throughout this thesis all the mathematical derivations respect a common notation. Vectors and vector functions are represented by bold symbols, e.g $\mathbf{x}$, $\mathbf{F}(\mathbf{x})$, scalars and scalar functions are indicated by plain letters, e.g. r, $f(x)$, $g(r)$, matrices and image signals are respectively denoted by capital letters in sans serif and typewriter fonts, e.g. the matrix M and the image $\mathtt{I}$. Points, lines and conics are represented in homogeneous coordinates. We do not distinguish between a projective transformation and the matrix representing it. Points in the plane are typically represented using homogeneous coordinates, with the symbol $\sim$ denoting the equality up to scale, e.g. $\mathbf{x} \sim \begin{pmatrix} x_1 & x_2 & 1 \end{pmatrix}^{\mathsf{T}}$. I_m indicates the $m \times m$ identity matrix, while $\mathsf{0}_{m\times n}$ denotes a rectangular matrix with zeros. If a matrix R is a function of a scalar α and/or a vector $\mathbf{q}$, then the input parameters are indicated in subscript, e.g. $\mathsf{R}_{\alpha,\mathbf{q}}$.

Considering a point $\mathbf{q} \sim (q_1\ q_2\ q_3)^T$ in $\mathbb{P}^3$ and a conic that can be represented either by a 3×3 symmetric matrix:

$$\Omega \sim \begin{pmatrix} \omega_1 & \frac{\omega_2}{2} & \frac{\omega_4}{2} \\ \frac{\omega_2}{2} & \omega_3 & \frac{\omega_5}{2} \\ \frac{\omega_4}{2} & \frac{\omega_5}{2} & \omega_6 \end{pmatrix} \tag{1.1}$$

or by a 6×1 homogeneous vector of the parameters:

$$\boldsymbol{\omega} = \left(\begin{array}{cccccc} \omega_1 & \omega_2 & \omega_3 & \omega_4 & \omega_5 & \omega_6 \end{array} \right)^T ,$$

point $\mathbf{q}$ belongs to the conic Ω *iff* $\mathbf{q}^T\Omega\mathbf{q} = 0$. For the particular case of Ω being a circle, $w_1 = w_3$ and $w_2 = 0$, and we can write that the circle passes by point $\mathbf{q}$ as:

$$\mathring{\boldsymbol{\omega}}^T \mathring{\mathbf{q}} = 0$$

with $\mathring{\boldsymbol{\omega}}$ being a 4×1 homogeneous vector that can be interpreted as a point in $\mathbb{P}^3$, the 3D projective space.

$$\mathring{\boldsymbol{\omega}} \sim \left(\begin{array}{cccc} \omega_1 & \omega_4 & \omega_5 & \omega_6 \end{array} \right)^T ,$$

and $\mathring{\mathbf{q}}$ is a lifted representation of $\mathbf{q}$ [10]:

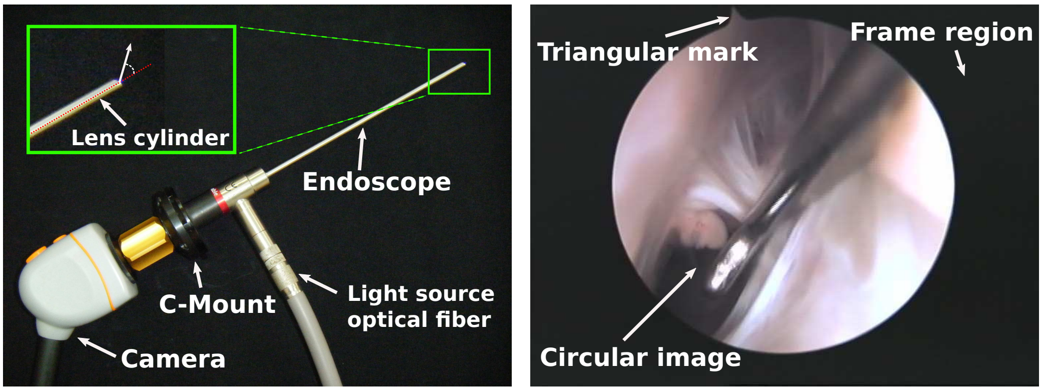
$$\mathring{\mathbf{q}} \sim \left(\begin{array}{cccc} q_1^2 + q_2^2 & q_1 q_3 & q_2 q_3 & q_3^2 \end{array} \right)^T$$

Chapter 2

Modeling Endoscopic Cameras

Rigid medical endoscopes typically combine an endoscopic lens with a Charge-Coupled Device (CCD) camera, as shown in Figure 2.1a. The borescope is a tubular device, with an objective lens and an eyepiece linked through an optical relay system. The relay works as an optical repeater that moves the projection center from the objective lens to a virtual point on the back end of the rigid tube [11]. The lens covers the CCD sensor that acquires images with the dark aperture (circular region) being visible due to the optics converter used for mounting the lens on the camera (Figure 2.1b). Two types of rigid endoscopes are commonly used in medicine: the *forward viewing endoscopes*, where the optical axis is aligned with the cylindrical probe, and the *oblique viewing endoscopes*, where the optical axis makes an angle of 30°, 70° or 90° with respect to the symmetry axis of the probe. The viewing direction of the latter can be changed without moving the camera head by simply rotating the endoscope around its symmetry axis [12–14]. This relative rotation is typically inferred by observing the position of a triangular mark on the periphery of the circular region (Figure 2.1b). As pointed out in [15], the motion of the lens probe with respect to the camera head changes the projection parameters and the position and shape of the circular boundary separating the two image regions. Oblique viewing endoscopes are specially useful in inspecting narrow cavities, such as the articulations (arthroscopy) or the sinus (rhinoscopy), where the space to maneuver the probe is very limited.

In oblique viewing endoscopes the motion between optics and camera sensor changes the projection parameters and the position and shape of the circular boundary separating the two image regions [15]. Having an accurate description of the image formation is fundamental for many image processing algorithms, including the



(a) Rigid endoscopic camera

(b) Medical endoscopic image

Figure 2.1: The rigid endoscope combines a lens with a CCD camera (a). The system enables the visualization of human body cavities whose access is difficult or limited, and is widely used for both surgery and diagnosis. (b) is an image of the interior of a knee joint, acquired by the 30° oblique viewing arthroscope.

radial distortion correction which is specially relevant in the context of this thesis. Since it is not feasible to calibrate the endoscope for every possible lens position, the calibration parameters must be updated according to a camera model that accounts for this relative motion.

By the aforementioned we can expect that modeling such a complex and advanced optical system presents some challenges over conventional lenses. In this chapter we will introduce the radial distortion model used throughout this thesis and depict the proposed projection model that accounts for the possible lens rotation.

2.1 Modeling the Radial Distortion

To capture an image with a greater visible range, the convex lens of a wide-angle camera (such as an endoscope) reflects more light from captured objects. The reflected light projects on the fixed size image sensor array of the camera. However, the refracted angle of the light within the convex lens increases from the center to the surrounding, which distorts the captured image in the back of the convex lens. This optical aberration, commonly known as RD, is usually encountered in wide angle lenses and/or very small optics, as the ones used in endoscopy. The barrel-type distortion addressed in this thesis, henceforth simply referred as RD, causes a nonlinear geometric deformation of the image, with the points being moved radially

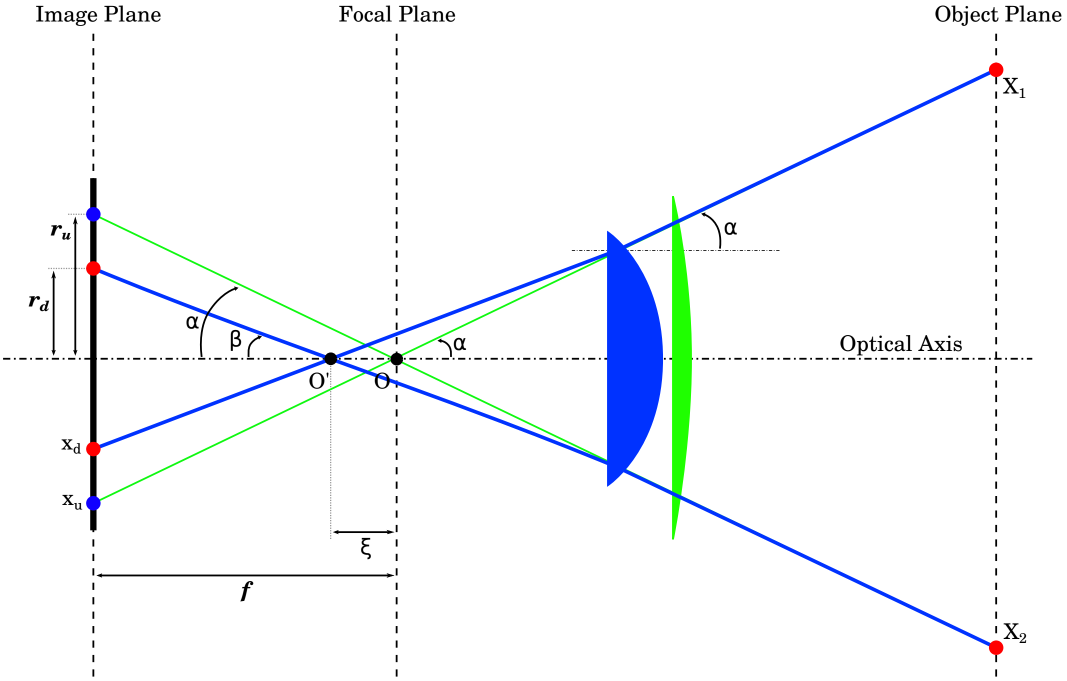


Figure 2.2: The imaging of two scene points X_1 and X_2 into the image sensor through a wide angle lens with distortion (blue). The physical properties of the lens causes the incident light rays in the periphery to bend, creating an effective viewpoint O' displaced from the viewpoint O along the optical axis. Using the Division Model to model the RD, this viewpoint shift can be described by a single parameter ξ, providing an intuitive understanding of the distortion effect. Under the RD effect, point X_1 is imaged to x_d and x_u represents the undistorted version of x_d as if it were imaged through a perspective lens without distortion (green).

towards the center [16], as shown in Figure 2.2.

The barrel-type spatial distortion made by the convex lens is traditionally corrected up to a certain limit by a multi-piece lens group within a certain distortion range. This type of optical structure has a large size, and manufacturing precise optical lenses is very expensive. Hence, many studies have proposed various mathematical models of image distortion, and techniques to obtain the optimal model parameters for correcting wide-angle distortion [17, 18].

2.1.1 Polynomial Model

In the camera calibration method proposed by Zhang [19], the author model the lens distortion function by a Taylor expansion of the form:

$$L(r) = \alpha_0 + \alpha_1 r + \alpha_2 r^2 + \alpha_3 r^3 + \cdots , \tag{2.1}$$

with r being the distance of point x in the image to the principal point. Zhang limits this expansion to second order and forth order terms, yielding the even order polynomial:

$$f(r) = k_1 r^2 + k_2 r^4, \tag{2.2}$$

where k_i denote the parameters of this distortion model.

Later, Alvarez et al. [20] proposed a similar model to [19], but the authors did not restrict the polynomial to terms of an even-order degree and modeled the distortion as a polynomial of order four:

$$f(r) = k_0 + k_1 r^2 + k_2 r^2 + k_3 r^3 + +k_4 r^4, \tag{2.3}$$

2.1.2 Parameter-free Model

Hartley et al. [21] proposed a parameter-free distortion correction method. They motivated their approach by the fact that the Taylor expansion does not work well for strong distortions (such as the ones observed in fish-eye and wide angle lenses) since, in the periphery, the distortion approximates infinity and cannot be well modeled by a polynomial. As a consequence the authors propose a parameter-free model which uses a high number of point correspondences between distorted and undistorted points. Based on these correspondences a scatter plot is generated, which plots the radial distance between distorted and undistorted points against the distorted radius. Then for each point in the scatter plot the right scale factor is computed for each available distorted point. In case of points which are not available, the respective values are interpolated to some precision. The values computed this way are stored in a lookup table T for the distortion correction process. Using T the authors can also compute a second table T', to obtain distortion strengths for computing the distorted coordinates from the undistorted points.

2.1.3 Division Model

In contrast to the polynomial models, the so-called Division Model (DM) [22, 23] only needs one parameter to be estimated in order to correct radial distortions. Let $\mathbf{X}$ be the vector of homogeneous coordinates of a 3D point represented in a world reference frame. Point $\mathbf{X}$ is projected into point $\mathbf{x}$ in the image such that

$$\mathbf{x} \sim \mathsf{K}_0\, \boldsymbol{\Gamma}_\xi(\mathsf{P}\,\mathbf{X})\,. \tag{2.4}$$

P denotes the standard 3×4 projection matrix [16], $\boldsymbol{\Gamma}_\xi$ is a nonlinear projective function that accounts for the image radial distortion with ξ being the distortion parameter, and K_0 is the matrix of intrinsic parameters with the following structure

$$\mathsf{K}_0 \sim \begin{pmatrix} a\,f & s\,f & c_x \\ 0 & a^{-1}\,f & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.5}$$

Where f denotes the focal distance, a stands for the aspect ratio, s for the skew angle and (c_x, c_y) is the image principal point in pixels [16]. The world undistorted projected points $\mathbf{x}_u \sim (x_u\, y_u\, z_u)^\mathsf{T}$ are mapped into world distorted points $\mathbf{x}_d$ according to:

$$\boldsymbol{\Gamma}_\xi(\mathbf{x}_u) \sim \left(2\,x_u \quad 2\,y_u \quad z_u + \sqrt{z_u^2 - 4\xi(x_u^2 + y_u^2)}\right)^\mathsf{T}. \tag{2.6}$$

The amount of nonlinear distortion is quantified by a single scalar parameter ξ that always takes a non-positive value ($\xi \leq 0$). The mapping in the projective plane $\mathbb{P}^2$ induced by $\boldsymbol{\Gamma}_\xi$ is bijective, and the inverse function $\boldsymbol{\Gamma}_\xi^{-1}$ transforms distorted points $\mathbf{x}_d \sim (x_d\, y_d\, z_d)^\mathsf{T}$ back into undistorted points $\mathbf{x}_u$ [24]:

$$\boldsymbol{\Gamma}_\xi^{-1}(\mathbf{x}_d) \sim \left(x_d z_d \quad y_d z_d \quad z_d^2 + \xi(x_d^2 + y_d^2)\right)^\mathsf{T}. \tag{2.7}$$

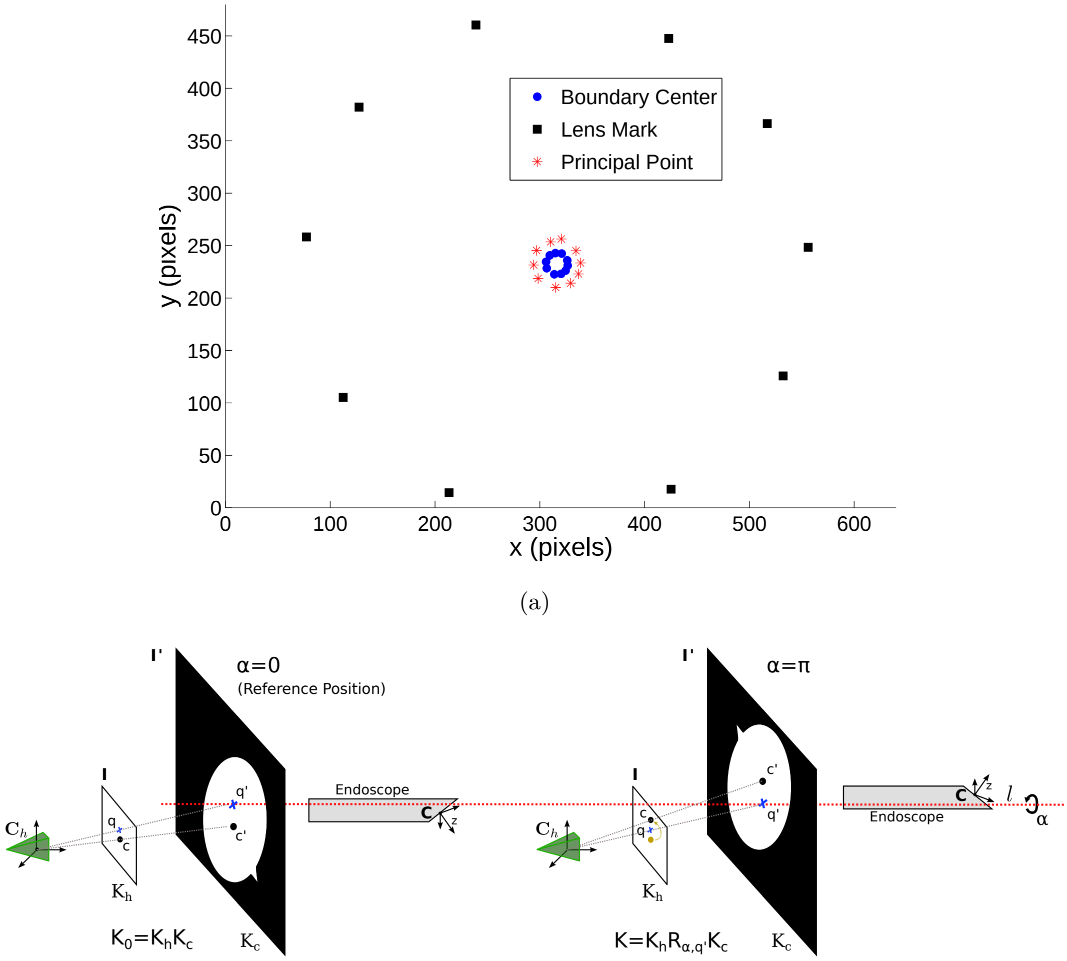
If the distance between $\mathbf{x}_d$ and the origin $\mathbf{O} \sim (0\,0\,1)^\mathsf{T}$ is

$$r_d = \sqrt{\frac{x_d^2}{z_d^2} + \frac{y_d^2}{z_d^2}}\,, \tag{2.8}$$

then it follows from the inverse mapping of equation 2.7 that the distance between $\mathbf{x}_u$ and $\mathbf{O}$ is

$$r_u = \frac{r_d}{1 + \xi\, r_d^2}\,. \tag{2.9}$$

The DM has been used mainly because of its simplicity and good accuracy when



(a)

(b)

Figure 2.3: Understanding and modeling the lens rotation. (a) is a plot of the principal point **c**, triangular mark **p**, and boundary center **w**, when the lens undergoes a full rotation (we took 10 calibration images as we rotated the lens). (b) illustrates the model and assumptions that are considered for updating the intrinsic matrix of the endoscopic camera. The lens projects a virtual image onto a plane $\mathtt{I}'$ that is imaged into $\mathtt{I}$ by a CCD camera. The relative rotation of the lens is around an axis **l** that is parallel to the optical axis and orthogonal to planes $\mathtt{I}$ and $\mathtt{I}'$. **c** represents the principal point and **q** is the center of the induced image rotation.

dealing with highly distorted images, as shown in the following chapters. Describing the distortion using the DM also enables the robust calibration from a single image presented by [24] and detailed in chapter 3.

2.2 Dealing with the endoscopic probe rotation

A camera equipped with a rigid endoscope is a compound system with a complex optical arrangement, where the projection is central [15]. As shown in [5], the user can rotate the lens relatively to the camera head, allowing a periscope-like view in oblique-viewing endoscopes. This changes the projection model, as demonstrated in Figure 2.3a, where we study this effect by acquiring 10 calibration images while rotating the lens probe for a full turn. The calibration was estimated for each angular position using the Single Image Calibration (SIC) of [24], and we plot the estimated principal point, as well as both the boundary $\mathbf{\Omega}$ and the triangular mark. Since the three points describe almost perfect concentric trajectories it seems reasonable to model the effect of the lens rotation on the camera intrinsics by means of a rotation around an axis orthogonal to the image plane. This idea has already been advanced by Wu et al. [13], but they consider that the axis always goes through the principal point, an assumption that in general does not hold, as shown by the experiment of Figure 2.3a.

The scheme in Figure 2.3b aims to give the idea of the proposed model for describing the effect of the lens rotation in the intrinsic parameters. Let us assume that the endoscopic lens projects a virtual image onto a plane $\mathtt{I}'$ placed at the far end. We can think of $\mathtt{I}'$ as the image that would be seen by looking directly through the endoscope eye-piece. K_c is the intrinsic matrix of this virtual projection, and $\mathbf{c}'$ is the point location where the optical axis meets the plane. Now assume that a camera head is connected to the eye-piece, such that the CCD plane $\mathtt{I}$ is perfectly parallel to $\mathtt{I}'$ and orthogonal to the optical axis. The projection onto I has intrinsics K_h, with the principal point $\mathbf{c}$ being the image of $\mathbf{c}'$. So, if the camera is skewless (s=0) with square pixels (a=1) and K_0 is the intrinsic matrix estimate

$$\mathsf{K}_0 \sim \begin{pmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix}, \tag{2.10}$$

then it can be factorized as

$$\mathsf{K}_0 \sim \mathsf{K}_\mathsf{h}\,\mathsf{K}_\mathsf{c} \sim \begin{pmatrix} f_h & 0 & c_x \\ 0 & f_h & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f_c & 0 & 0 \\ 0 & f_c & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{2.11}$$

with f_c being the focal length of the endoscopic lens, and f_h being the focal length of the camera head that converts metric units into pixels.

Let us now consider that the lens probe is rotated around an axis $\mathbf{l}$ by an angle α (Figure 2.3b left). $\mathbf{l}$ is assumed to be orthogonal to the virtual plane $\mathtt{I}'$, but not necessarily coincident with the lens axis. In this case the point $\mathbf{c}'$ describes an arc of circle with amplitude α and, since $\mathtt{I}$ and $\mathtt{I}'$ are parallel, the same happens with its image $\mathbf{c}$. The intrinsic matrix of the compound optical system formed by the camera head and the rotated endoscope becomes

$$\mathsf{K} \sim \mathsf{K}_h \, \mathsf{R}_{\alpha,\mathbf{q}'} \, \mathsf{K}_c \,, \tag{2.12}$$

with $\mathsf{R}_{\alpha,\mathbf{q}'}$ being a plane rotation by α around the point $\mathbf{q}'$, where the axis $\mathbf{l}$ intersects $\mathtt{I}'$.

$$\mathsf{R}_{\alpha,\mathbf{q}'} = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) & (1-\cos(\alpha))q'_x - \sin(\alpha)q'_y \\ -\sin(\alpha) & \cos(\alpha) & \sin(\alpha)q_x + (1-\cos(\alpha))q_y \\ 0 & 0 & 1 \end{pmatrix} \,, \tag{2.13}$$

The position of $\mathbf{q}'$ is obviously unchanged by the rotation, and the same is true for its image $\mathbf{q} \sim \mathsf{K}_h \, \mathbf{q}'$. Taking into account the particular structure of K_h, we can re-write equation 2.12 in the following manner[1]

$$\begin{aligned} \mathsf{K} &\sim \mathsf{R}_{\alpha,\mathbf{q}} \, \mathsf{K}_h \, \mathsf{K}_c \\ &\sim \mathsf{R}_{\alpha,\mathbf{q}} \, \mathsf{K}_0 \end{aligned} \,. \tag{2.14}$$

With the reasoning above, we have derived a projection model for the endoscopic camera that accommodates the rotation of the lens probe and is consistent with the observations of Figure 2.3a. Given the camera calibration K_0, at an arbitrary reference position (α=0), the matrix of intrinsic parameters at a certain frame time instant i becomes:

$$\mathsf{K}_i \sim \mathsf{R}_{\alpha_i,\mathbf{q}_i} \, \mathsf{K}_0 \,, \tag{2.15}$$

where α_i is the relative angular displacement of the lens, and $\mathbf{q}_i$ is the image point that remains fixed during the rotation. Since the RD is a characteristic of the lens, the parameter ξ is unaffected by the relative motion with respect to the camera-head.

2.3 Closure

In this chapter we have provided a comprehensive model of the image formation in the endoscopic camera. Such modeling is specially relevant when dealing with every

[1] The assumption of square pixels and zero skew is valid for most CCD cameras. If $a \neq 1$ or $s \neq 0$, then equations 2.12 and 2.14 are no longer strictly equivalent. However, the latter is typically a good approximation of the former, and the consequences in terms of modeling accuracy are negligible.

day user requirements, such as the lens probe rotation in arthroscopy, and will be used later on to incorporate the rotation probe compensation in the radial distortion correction of the final system. Remark that, although the presented camera model successfully deals with the probe rotation, other changes in the model such as optical zoom are not addressed. Lourenco et. al [25] has conducted some exploratory work that uses endoscopic image properties (the image boundary) as well as robust feature tracking in order to automatically detect zoom variation and update the calibration accordingly. This work presents a step forward into a complete endoscopic camera modeling that will allow the use of more advanced computer vision algorithms in endoscopy.

Chapter 3

Unsupervised Calibration from a Single Image

Geometric camera calibration is a well studied topic in the literature, and currently several methods and software are available for accomplishing the task [26]. The objective is to determine the intrinsic calibration matrix K_0, this is, the position of the image center in the image (c_x, c_y), the focal length f, the different scaling factors for rows and column pixels a, the skew factor s and the lens distortion ξ. Geometric camera calibration also aims at determining the rigid transform between the camera and the calibration target. Planar regular patterns are widely used as a calibration object because they are readily available and simplify the problem of establishing point correspondences. Bouguet's toolbox [27] is a popular software that implements Zhangs method [19] for calibrating a generic camera from a minimum of 3 images of a checkerboard. The toolbox is also able to accommodate non-linear lens distortions [28] by considering a 7th order polynomial model in a global iterative optimization step.

Auto-calibration techniques rely in point correspondences across views for determining both motion and camera parameters [29, 30]. Very recently, Zhang et al. presented a technique for calibrating a camera using one or more images of a pattern that is unknown but sufficiently structured to be considered a low rank texture matrix [31]. Given a single image, the algorithm enables to recover the distortion parameters and the principal point whenever the scene has two patterns orthogonal to each other. This requirement considerably limits the number of cases where the camera parameters can be recovered. Moreover the approach requires the user to roughly indicate the localization of the patterns which precludes fully automatic applications.

Unfortunately, the before mentioned methods for geometric camera calibration do not meet the usability requirements for the calibration of endoscopes in the operating room. In practice the number of input images for achieving accurate results is way above 3, and the detection of grid corners in images with RD needs substantial user intervention. Several authors addressed the specific problem of intrinsic calibration or RD correction in medical endoscopes [32–37]. However, these methods are either impractical for use in the OR, or they employ circular dot patterns to enable the automatic detection of calibration points, which precludes the accurate determination of the centers of the dots in an image with radial distortion and therefore undermines the results accuracy [38]. Barreto et al. showed in [24] that it is possible to fully calibrate a camera with distortion using a single image of a chessboard pattern.

In the context of this thesis, automation of the camera calibration procedure aiming at its simplification from the user standpoint is of major importance in order to achieve our usability requirements. This is particularly relevant in the context of medical endoscopy, where the complicated apparatus in the OR and the short available time for camera calibration precludes the use of conventional calibration methods. In this chapter we will address two methods for unsupervised camera calibration that can be used in the medical endoscopy context: (i) the SIC, that was proposed in [24] and uses a single image of a calibration pattern to recover the full set of camera parameters (intrinsics and extrinsics), and (ii) the unsupervised plumb-line calibration from a single image, that is a contribution of this thesis [6] and uses a single natural image to recover the intrinsics of the camera up to a global scale factor.

3.1 Single Image Calibration from a Calibration Pattern

Barreto et al. [24] showed that a camera following the projection model described in the previous chapter can be calibrated from a single image of a planar checkerboard pattern acquired from an arbitrary position. The authors use the lifted representation of image-plane correspondences to estimate the homography in $\mathbb{P}^5$ using a minimum of 12 points in a Direct Linear Transformation (DLT)-like approach.

Let $\mathbf{g}$ be a point in the checkerboard expressed in plane homogeneous coordinates, and $\mathbf{H}$ be the homography encoding the relative pose between plane and camera [16].

From equation 2.4, it follows that the image of $\mathbf{g}$ is

$$\mathbf{x} \sim \mathsf{K}_0 \, \mathbf{\Gamma}_\xi(\mathsf{H}\,\mathbf{g})\,.$$

Let $\widehat{\mathbf{g}}$ and $\widehat{\mathbf{x}}$ be homogeneous vectors with dimension 6 corresponding to the lifted representation of points $\mathbf{g}$ and $\mathbf{x}$ according to a second order Veronese map [10]. It can be proved that

$$\widehat{\mathbf{x}} \sim \widehat{\mathsf{H}}_{6\times 6}\, \widehat{\mathbf{g}} \tag{3.1}$$

with $\widehat{\mathsf{H}}_{6\times 6}$ being a 6×6 matrix. Since each image-plane correspondence imposes 3 linear constraints on the lifted homography, the matrix $\widehat{\mathsf{H}}_{6\times 6}$ can be estimated from a minimum of 12 point correspondences $(\mathbf{x}\,, \mathbf{g})$ using a DLT-like approach. Given $\widehat{\mathsf{H}}_{6\times 6}$, the matrix of intrinsic parameters K_0, the distortion parameter ξ, and the original homography H, can be factorized in a straightforward manner. This initial camera calibration can be further refined using standard iterative non-linear optimization.

User intervention is limited to the acquisition of a calibration image and the pose of the camera with respect to the checkerboard plane is arbitrary. It is important to note that locating corners in an image with strong radial distortion can be highly problematic because of the bending of the straight lines. After the frame acquisition the processing steps are:

1. Localization of a minimum of 12 corners in the center image region where the distortion is less pronounced. This operation is carried out by a heuristic algorithm that uses standard image processing techniques. The consistent correspondence between image corners $\mathbf{x}$ and grid points $\mathbf{g}$ is accomplished by counting the squares of the checkerboard pattern.

2. The image-plane correspondences are used for estimating $\widehat{\mathsf{H}}_{6\times 6}$ using a DLT-like approach (equation 3.1).

3. The checkerboard pattern is projected onto the image plane using the homography for generating corner hypotheses in the image periphery. These hypotheses are confirmed and refined by applying a standard image corner finder.

4. The lifted homography is re-estimated using the entire set of image-plane correspondences. The intrinsic matrix K_0, the distortion parameter ξ, and the pose of the checkerboard are computed by factorizing $\widehat{\mathsf{H}}_{6\times 6}$ [24].

Table 3.1: Comparison between SIC, Hartley's and Bouguet's calibration.

	c_x (px)	c_y (px)	f (mm)	ξ
SIC (mean)	595.77	500.14	558.88	-0.527
SIC (std)	7.069	4.889	34.935	0.0066
Hartley (mean)	618.15	522.38	-	-
Hartley (std)	42.96	63.16	-	-
Hartley RANSAC (mean)	599.23	499.99	-	-
Hartley RANSAC (std)	6.8715	7.0680	-	-
Bouguet	599.32	497.08	541.90	-0.497

5. The calibration results are refined by an iterative nonlinear optimization that assumes zero skew and a unitary aspect ratio.

In order to assess the robustness and repeatability of the SIC we conducted an experiment where we ran the algorithm in 10 calibration images with resolution 1280 × 960. We report a comparison with the well known Bouguet's method [27] and the non-parametric approach proposed by Hartley et al. [21] (both the original approach and a robust variant that applies RANdom SAmple Consensus (RANSAC) for estimating the fundamental matrix that encodes the distortion center). The results are presented in table 3.1 and show a very good repeatability, providing mean values for the calibration parameters that are similar to the ones achieved with [27] and a distortion parameter consistent with [21]. Note that since Bouguet assumes a polynomial model, the estimated distortion profile is fitted with the division model for comparison purposes.

The relatively high standard deviation in f and ξ is justified by the fact that they are coupled parameters. As explained in [24], whenever camera and plane are close to a fronto-parallel configuration, the decoupling of f and ξ becomes inaccurate and the results is correct up to a global scale factor. However, this scale factor does not affect an eventual RD correction for visualization purposes. Moreover, and since Bouguet models the RD using a 3 parameter polynomial, the comparison with the division model was carried out by interpolating the distortion profile curves, which can also explain minor differences between the values of f and ξ.

We also conducted an experiment to evaluate the repeatability of the SIC in a variety of endoscopic lenses. The results showed a good repeatability across different lenses used in several specialities, as depicted in Figure 3.1. We developed a toolbox based on the SIC proposed in [24] dubbed EasyCamCalib, which has been used in the experiments presented in this thesis. This toolbox allows the calibration of

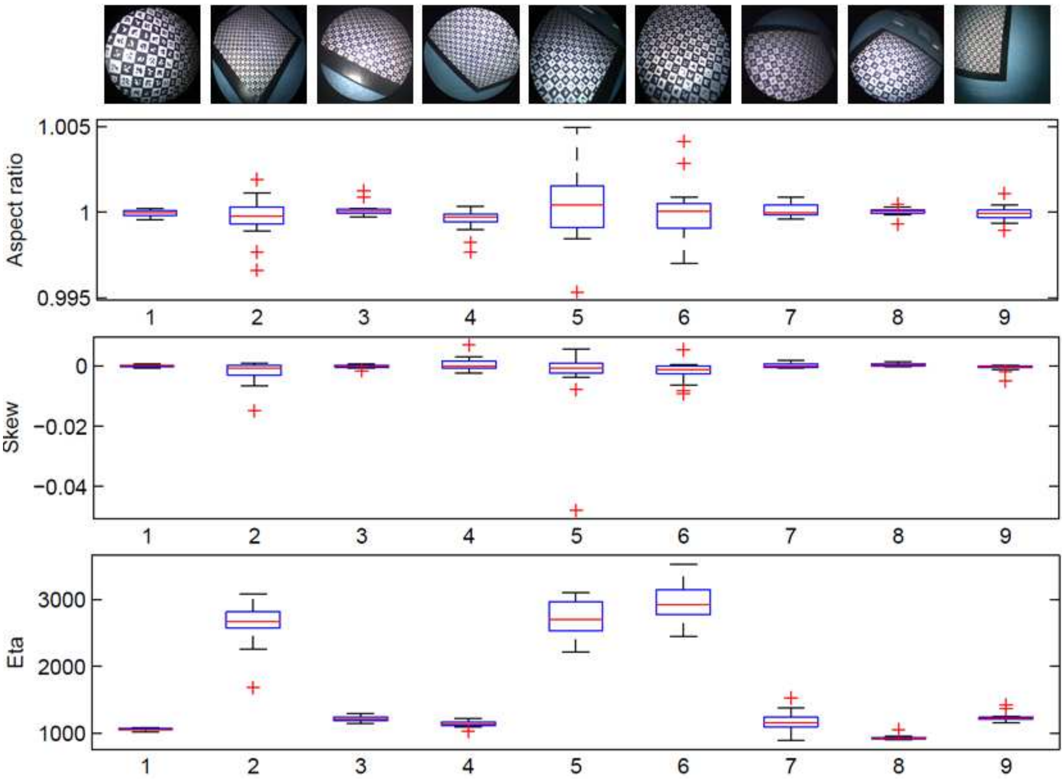
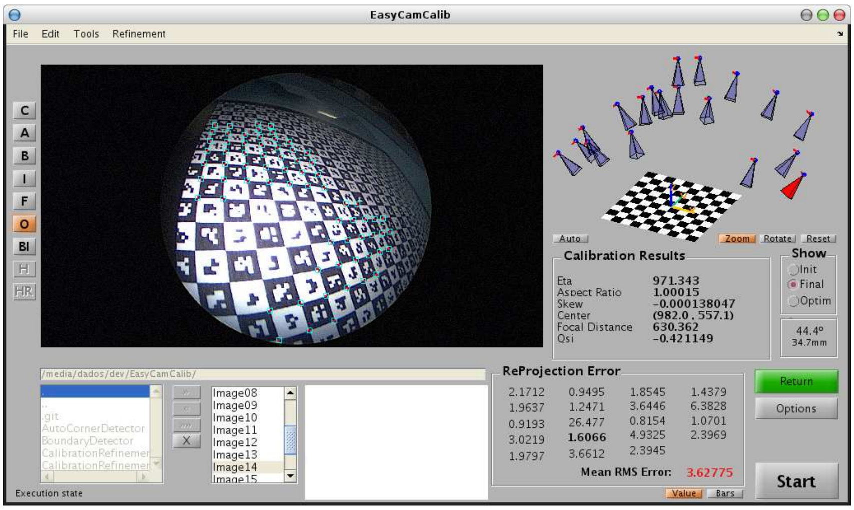


Figure 3.1: Comparing the repeatability of the SIC [24] in different endoscopic lenses. 16 images per lens are used to conduct the comparison.



Calibration Results

```
Eta              971.343
Aspect Ratio     1.00015
Skew             -0.000138047
Center           (982.0 , 557.1)
Focal Distance   630.362
Osi              -0.421149
```

ReProjection Error

```
2.1712    0.9495    1.8545    1.4379
1.9637    1.2471    3.6446    6.3828
0.9193    26.477    0.8154    1.0701
3.0219    1.6066    4.9325    2.3969
1.9797    3.6612    2.3945

          Mean RMS Error:   3.62775
```

Figure 3.2: The EasyCamCalib toolbox used to quickly calibrate a camera from a single image of a calibration pattern in an arbitrary position.

any perspective camera with distortion, presenting useful non-linear optimization routines and results visualization that help the quick calibration of a camera in the research lab (see Figure 3.2). For increased robustness we adopted the special patterns used in [39] and optimized the calibration target points detection provided by the authors for the case of moderate to high radial distortion.

3.2 Calibration from the Image of Lines

Contrary to what happens with conventional perspective cameras, in the case of cameras with distortion it is possible to recover the intrinsic calibration from the projection of 3D lines in random position [26].

In [6], we investigate the problem of fully calibrating an image with significant distortion without requiring any type of manual supervision. A solution for automatic, single frame calibration is specially relevant in the case of images mined from the internet, for which knowing the camera parameters can be useful for multiple tasks. Possible applications include the distortion correction via image warping for visualization purposes [5], the normalization of images for subsequent use in content retrieval frameworks [40], the estimation of camera rotation by aligning vanishing directions with a manhattan world scene [41], or the inference of 3D metric information from the image [26].

We consider the case of cameras with distortion that can be described by the 1-parameter DM [22,23], and assumes that the imaged scene has a reasonable number of straight lines. This work proposes for the first time a calibration algorithm that, given the image of 3 lines, it estimates the distortion, principal point, aspect ratio, and skew. Such result is not surprising if we consider that the DM has obvious resemblances with the stereographic projection used to describe the para-catadioptric sensor [10], and that para-catadioptric cameras can be fully calibrated from a minimum of 3 line images [42]. It is also shown that there are some differences with respect to the para-catadioptric case, namely the possibility of recovering aspect-ratio and skew from a single line projection, and the fact that the distortion can only be computed in pixels, being proved that line information is insufficient to decouple the focal length parameter from the distortion parameter in millimeters. Such decoupling requires additional information that is often available (e.g. the EXIF tag, the nominal field-of-view) and only knowing the distortion in pixels still enables accurate distortion compensation with the focal length being chosen as a function of the desired resolution for the output perspective [5].

3.2.1 Plumb-line Calibration

Since lines are features that often appear in natural images, with special relevance in the case of man-made environments, line-based calibration is an appealing proposition. The first contributions in camera calibration using the so-called "plumb-line" constraint go back to the 70's when Brown suggested to model the distortion by a polynomial and estimate its parameters by straightening up the lines in the image [43]. Latter on, Brauer-Burchardt et al. [23] and Fitzgibbon [22] simultaneously proposed to describe the image distortion by a 1-parameter rational function. The geometry of line projection considering the DM was investigated by Barreto in [10]. He concluded that, similarly to para-catadioptric cameras, the lines in 3D are projected into a family of conic curves that intersect in two points and satisfy an harmonic conjugate relation with two other points [42]. He also showed that the conic where a line is projected has only two independent Degrees of Freedom (DOF) and that, if the center is known, then it is possible to estimate and correct the image distortion using a single line. More recently Wang et al. proposed an algorithm for computing both the distortion parameter and the principal point from an image of 3 lines [44] using an algebraic interpretation of the DM. In all these works the user is required to manually select the image contours corresponding to the projection of lines. Very recently Bukhari et al. [45] suggested an algorithm for automatically detecting lines and accomplishing the calibration following the methods of [44] and provided a detailed study of the circle fitting methods applied to the calibration problem.

3.2.2 Intrinsic calibration using a "plumb-line" approach

In this section we present our geometric interpretation of the plumb-line calibration problem and derive a minimal solution for the calibration of the camera, up to a global scale factor, using 3 images of lines. We start by giving a quick overview of the geometry for projection of lines in the image plane, pointing the relation between conics in the image and world straight lines under the division model and providing the minimal solution for recovering the instrinsics from the image of 3 lines.

Modelling the camera distortion using the DM [22, 23], where ξ is the negative parameter that quantifies the amount of distortion, we define $\mathbf{h}()$ as the radial distortion function that maps undistorted points $\mathbf{u}$ in $\mathbb{P}^2$ into distorted points $\mathbf{d}$ in

$\mathbb{P}^2$:

$$\mathbf{d} \sim \mathbf{h}(\mathbf{u}) \sim \begin{pmatrix} 2u_1 & 2u_2 & u_3 + \sqrt{u_3^2 - 4\xi(u_1^2 + u_2^2)} \end{pmatrix}^T \tag{3.2}$$

Let $\mathbf{n} \sim \begin{pmatrix} n_1 & n_2 & n_3 \end{pmatrix}$ be the projection of a 3D line onto the projective plane according to the conventional pinhole model. A point $\mathbf{x}$ lying on line $\mathbf{n}$ is transformed into image point $\mathbf{x}'$ by the non-linear function $\mathbf{h}^{-1}(\mathbf{x})$. Since $\mathbf{n}^T\mathbf{x} = 0$ and $\mathbf{x} = \mathbf{h}(\mathbf{x}')$ it comes that $\mathbf{n}^T\mathbf{h}(\mathbf{x}') = 0$. After some algebraic manipulation, we can write the previous equation as $\mathbf{x'}^T\Omega\mathbf{x}'$, with Ω being given by:

$$\Omega = \begin{pmatrix} \xi n_3 & 0 & \frac{n_1}{2} \\ 0 & \xi n_3 & \frac{n_2}{2} \\ \frac{n_1}{2} & \frac{n_2}{2} & N_3 \end{pmatrix}. \tag{3.3}$$

This shows that the distortion function 3.2 transforms a line $\mathbf{n}$ into the conic section Ω [10]. It has been shown in [10] that Ω is the distorted image of a world line *iff* it passes through the circular points $\mathbf{I}$ and $\mathbf{J}$, and points $\mathbf{r}_+$ and $\mathbf{r}_-$ are harmonic conjugates [46] with respect to Ω:

$$\begin{cases} \mathbf{I}^T\Omega\mathbf{I} = 0 & \text{with} \quad \mathbf{I} = \begin{pmatrix} 1 & i & 0 \end{pmatrix}^T \\ \mathbf{J}^T\Omega\mathbf{J} = 0 & \text{with} \quad \mathbf{J} = \begin{pmatrix} 1 & -i & 0 \end{pmatrix}^T \\ \mathbf{r}_+^T\Omega\mathbf{r}_- = 0 & \text{with} \quad \mathbf{r}_\pm = \begin{pmatrix} 1 & 0 & \pm\sqrt{\xi} \end{pmatrix}^T \end{cases} \tag{3.4}$$

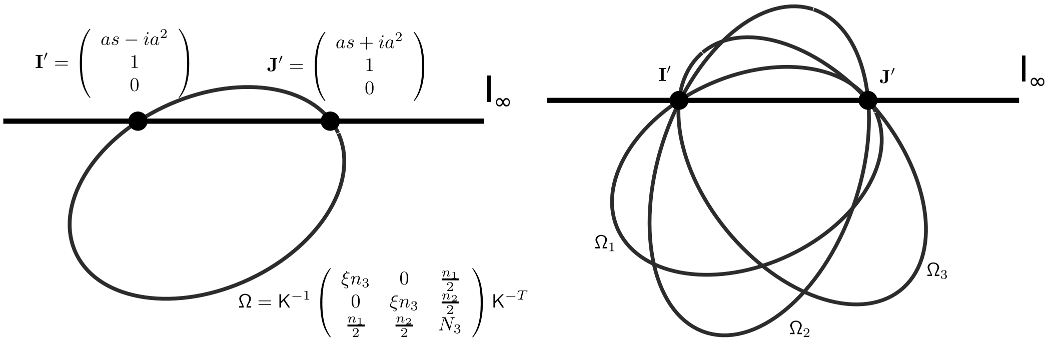
For convenience we will use, whenever needed, lifted representations of points and conics, so for example the condition for a point belonging to a conic section can be written in the following manner:

$$\mathbf{q}^t \underbrace{\begin{bmatrix} \omega_1 & \frac{\omega_2}{2} & \frac{\omega_4}{2} \\ \frac{\omega_2}{2} & \omega_3 & \frac{\omega_5}{2} \\ \frac{\omega_4}{2} & \frac{\omega_5}{2} & \omega_6 \end{bmatrix}}_{\Omega} \mathbf{q} = 0 \Leftrightarrow$$

$$\Leftrightarrow \underbrace{\begin{pmatrix} \omega_1 & \omega_2 & \omega_3 & \omega_4 & \omega_5 & \omega_6 \end{pmatrix}}_{\boldsymbol{\omega}^T} \widehat{\mathbf{q}}^T = 0 \tag{3.5}$$

where $\widehat{\mathbf{q}}$ denotes the lifted coordinates of $\mathbf{q}$ according to a second order Veronese map [10].

The division model of equation 3.2 is usually studied as a bijective mapping in image coordinates. In this work we assume the mapping to be before the intrinsics.


$\mathbf{I}' = \begin{pmatrix} as - ia^2 \\ 1 \\ 0 \end{pmatrix}$
$\mathbf{J}' = \begin{pmatrix} as + ia^2 \\ 1 \\ 0 \end{pmatrix}$
l_∞
$\Omega = \mathsf{K}^{-1} \begin{pmatrix} \xi n_3 & 0 & \frac{n_1}{2} \\ 0 & \xi n_3 & \frac{n_2}{2} \\ \frac{n_1}{2} & \frac{n_2}{2} & N_3 \end{pmatrix} \mathsf{K}^{-T}$
$\mathbf{I}'$
$\mathbf{J}'$
l_∞
Ω_1
Ω_3
Ω_2

(a) (b)

Figure 3.3: Intersecting projections of 3D lines with the line at infinity.

In this case, a point $\mathbf{Q}$ in 3D is projected onto a point $\mathbf{q}$ in the image by:

$$\mathbf{q} \sim \mathsf{K}\mathbf{h}(\mathsf{P}\mathbf{Q}),$$

with K being the matrix of intrinsic parameters of the camera:

$$\mathsf{K} = \begin{pmatrix} af & sf & c_x \\ 0 & a^{-1}f & c_y \\ 0 & 0 & 1 \end{pmatrix},$$

P being the standard 3×6 projection matrix [16], and $\mathbf{h}()$ being the radial distortion function with ξ being now expressed in millimeters instead of pixels.

The conic Ω where a line is imaged is now given by transforming the result of equation 3.3 by the intrinsic parameters K, as shown in Figure 3.3a. Since projective transformations preserve incidence and cross-ratio relations, the conic Ω must intersect the line at infinity in points $\mathbf{I}' \sim \mathsf{K}\mathbf{I}$ and $\mathbf{J}' \sim \mathsf{K}\mathbf{J}$, and must verify an harmonic relation with respect to points $\mathbf{r}'_\pm \sim \mathsf{K}\mathbf{r}_\pm$. Therefore, a conic is the image of a line *iff* it verifies $\mathsf{\Phi}\boldsymbol{\omega} = 0$, with $\boldsymbol{\omega}^T$ being its representation in $\mathbb{P}^5$ and $\mathsf{\Phi}$ being the 3×6 matrix:

$$\mathsf{\Phi} = \underbrace{\begin{bmatrix} (as - ia^2)^2 & as - ia & 1 & 0 & 0 & 0 \\ (as + ia^2)^2 & as + ia & 1 & 0 & 0 & 0 \\ c_x^2 - \frac{a^2}{\eta} & c_x c_y & c_y^2 & c_x & c_y & 1 \end{bmatrix}}_{\mathbf{\Lambda}} \tag{3.6}$$

$$\text{with} \qquad \eta = \frac{\xi}{f^2}.$$

The two first rows are the lifted representation of $\mathbf{I}'$ and $\mathbf{J}'$ respectively, and the last row is the joint lifted representation of $\mathbf{r}_+$ and $\mathbf{r}_-$ [47].

If the calibration parameters of the camera are known, then the conic Ω can be estimated from $N \geq 2$ image points using constrained least squares [48], with equation 3.6 giving the set of 3 linear constraints. From equation 3.6, we observe that the images of lines lie in a 2D subspace $\mathbb{S}$ of $\mathbb{P}^5$ that encodes the calibration. We now show how to recover the calibration parameters from 3 line images $\boldsymbol{\omega}_1$, $\boldsymbol{\omega}_2$ and $\boldsymbol{\omega}_3$ (Figure 3.3b). If the projection of a 3D line is correctly estimated in the image plane, intersecting it with the line at infinity defines points $\mathbf{I}'$ and $\mathbf{J}'$, whose coordinates encode a and s as shown in Figure 3.3a. Solving with respect to these parameters it can be shown that:

$$a = \sqrt{\frac{\omega_3}{\omega_1} - \frac{1}{4}\frac{\omega_2^2}{\omega_1^2}} \quad \text{and} \quad s = -\frac{\omega_2}{\omega_1} a^{-1}, \tag{3.7}$$

where ω_i are the entries of $\boldsymbol{\omega}$. We can therefore retrieve a and s from the image of a single line. The principal point (c_x, c_y) and distortion parameter are encoded in the third orthogonal vector to the subspace of lines, $\mathbb{S}$. Given tree line images, we can determine this subspace and compute $\boldsymbol{\Lambda}$ by parametrizing the null space of the lines as follows:

$$\mathbb{N}(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \boldsymbol{\omega}_3) = K_1\mathbf{V}_1 + K_2\mathbf{V}_2 + K_3\mathbf{V}_3 \tag{3.8}$$

with $\boldsymbol{\Phi} \in \mathbb{N}(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \boldsymbol{\omega}_3)$. Considering the third row of $\boldsymbol{\Phi}$, $\boldsymbol{\Lambda}$, the following conditions must hold:

$$\begin{cases} \Lambda_6 = 1 \\ \Lambda_2 - \Lambda_5\Lambda_4 = 0 \\ \Lambda_5^2 - \Lambda_3 = 0 \end{cases}$$

After solving the first constraint in order to K_3 and replacing it in the other two, the parameters K_1 and K_2 can be computed by intersecting the two conic curves given by the 2^{nd} and 3^{rd} constraints (system of two second order homogeneous polynomials). From the 4 possible solutions we choose the one with physical meaning.

In summary, we have shown that three world lines are sufficient to calibrate a camera. However, two important remarks are done:

1. We are only able to determine the ratio η between ξ and f^2, that can be understood as the distortion parameter expressed in pixels rather than in mil-

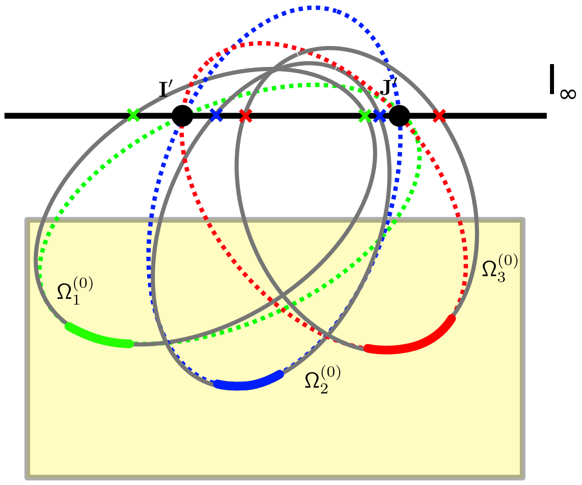


Figure 3.4: 3 line estimations do not intersect at the same point with the line at infinity due to the discrete nature of image edges and noise in the image.

limeters. Nevertheless this coupled parameter enables to rectify the image distortion (as shown in the experiments).

2. We can verify that, considering a fourth line projection $\boldsymbol{\omega}_4$, puts no further constraints to the calibration problem. The conic curve must satisfy two linear constraints since it must pass by points $\mathbf{I}'$ and $\mathbf{J}'$, and must belong to the subspace orthogonal to $\boldsymbol{\Lambda}$. This means that only 2 of the 5 DOF of the conic curve are really independent and they refer to the orientation of the plane containing the original line in 3D (see [10]). Thus, we conclude that line images $\boldsymbol{\omega}_i$, with $i > 3$, bring no additional information about the camera calibration and it is impossible to decouple the focal length f from the distortion parameter ξ using exclusively line features.

The calibration solution demonstrated above enables to determine the back-projection directions up to an angular multiplicative factor. Determining this multiplicative factor requires knowing, directly or indirectly, the absolute angle between two back-projection rays (e.g. know 'a priori' the field-of-view (FOV) of the camera or extract focal distance from EXIF tag).

3.2.3 Calibration algorithm

The previous section demonstrates how to calibrate a camera from a minimum of 3 imaged lines. In practice, for each conic, we are only able to extract points belonging to a small arc of the conic. The joint effect of noise and strong partial occlusion makes the estimation of the conic very uncertain [47]. In Figure 3.4 we can see that the arrangement of the initial conics $\Omega_i^{(0)}$ does not comply with the constrains derived in section 3.2.2. In this case the conics do not intersect the line at infinity in two unique points and the harmonic relations with respect to points $\mathbf{r}'$ and $\mathbf{t}'$ are not consistent. Hence, it is of the utmost relevance to correctly estimate arc segments in the image that are likely to be the projection of lines and are used as input for the remaining of the estimation algorithm.

We start by extracting edges in the image using the canny edge detector [49] and connect them into continuous segments. Those segments present various shapes depending on the edges connectivity and therefore have to be divided into probable circular arcs. While the authors in [45] use RANSAC [50] for fitting circles to the edges in each segment, preserving all the non-overlapping models (candidate circular arcs) that have more supporting points, we recursively fit circles to edges in a segment using Taubin's iterative minimization [51] and cut the segments at the maximum error points. With this method, outlier edges caused by noise are likely to originate segment division, creating a larger but more accurate amount of segments.

Table 3.2 summarizes the unsupervised calibration algorithm. We start by estimating the likely location of the conics intersection with the line at infinity (steps 1 to 3) and then the conics are re-estimated from the corresponding image points enforcing the incidence with points $\mathbf{I}'$ and $\mathbf{J}'$ (step 4). The principal point (c_x, c_y) and η are estimated by computing the vector $\mathbf{\Lambda}$ lying in the null space $\mathbb{N}(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \boldsymbol{\omega}_3)$ (steps 5 to 6).

The calibration estimation of steps 1 to 6 is sub-optimal and is used as initialization for a final iterative optimization step. From the analytical form of Ω (see Fig:3.3a), and after some algebraic manipulation, we derive the following equation:

$$\underbrace{\begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \\ \omega_5 \\ \omega_6 \end{pmatrix}}_{\boldsymbol{\omega}} \sim \underbrace{\begin{pmatrix} 0 & 0 & \eta a^{-2} \\ 0 & 0 & -2\eta a^{-1} s \\ 0 & 0 & \eta(a^2+s^2) \\ a^{-1} & 0 & -2\eta c_x a^{-1} \\ -s & a & 2\eta(c_x s + c_y a) \\ -c_x & -c_y & 1+\eta(c_x^2+c_y^2) \end{pmatrix}}_{\mathsf{G}} \cdot \underbrace{\begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix}}_{\mathbf{m}} \tag{3.9}$$

with G being a 6×3 matrix that encodes the calibration parameters and $\mathbf{m}$ being the 3×1 vector encoding the orientation of the plane that contains the line [10].

Table 3.2: Unsupervised plumb lines calibration algorithm. Steps 1 to 3 apply when there is no prior knowledge of the camera aspect ratio and skew angle.

1. Given 3 contours $\mathsf{e}_{i=1\cdots3}$, estimate the conics $\Omega^{(0)}_{i=1\cdots3}$ using a standard conic fitting algorithm. For our experiments we use Taubin's method [51] due to its better performance with small arcs [47].

2. Intersect each conic $\Omega^{(0)}_i$ with the line at infinity and obtain $\mathbf{I}'_i$ and $\mathbf{J}'$ estimates [47].

3. Estimate the pair of complex conjugate points $\mathbf{I}'$, $\mathbf{J}'$ from the pairs $\mathbf{I}'_i$, $\mathbf{I}'_i$ by averaging the real and imaginary parts.

4. (Re)-estimate the conics using constrained least squares [48], forcing them to intersect $\mathbf{I}'$ and $\mathbf{J}'$.

5. Given the conics $\Omega_{i=1\cdots3}$ compute a basis for the null space $\mathbb{N}$ and determine the calibration vector.

6. Extract the principal point (c_x, c_y) and η from the calibration vector.

7. Refine the calibration result by using iterative optimization.

Given the conics $\boldsymbol{\omega}_i$ and the matrix G, computed with the calibration initialization, the corresponding vector $\mathbf{m}_i$ is determined linearly. Let $\mathbf{q}_j^{(i)}$ be a contour point $j = 1 \cdots N_i$ belonging to the i^{th} conic $\boldsymbol{w}_i$. The bundle adjustment of the calibration parameters is carried by minimizing the function of equation 3.10:

$$f = \min_{a,s,c_x,c_y,\eta,\mathbf{m}_i} \sum_{i=1}^{3} \sum_{j=0}^{N_i} \left(\mathbf{m}_i^T \mathsf{G}^T \widehat{\mathbf{q}}_j^{(i)} \right)^2 , \tag{3.10}$$

Note that, if there is no prior knowledge about a and s, it is possible to estimate $\mathbf{I}'$ and $\mathbf{J}'$ from steps 1 to 3 in Table 3.2. However, if there is some prior knowledge such that reasonable assumptions can be made about a and s (e.g. the camera is skewless and has square pixels), then the 3 first steps can be skipped and the calibration carried trough 4 to 7.

While standard "plumb-line" calibration requires user intervention for selecting the image edges that are projection of lines, we carry this operation in a fully automatic manner. The line contours are detected in the clutter by evaluating their geometric consistency with putative calibrations. This leads to a complex problem of multi-model fitting that is difficult to solve in practice, since we aim at simultaneously detecting the line contours and finding the camera calibration. In the next section we depict two approaches to solve this fitting problem:

- Contours are used as input in a robust estimation framework that uses our geometric derivations to perform exemplar-based clustering and robust inlier detection. We will dub this as the **MAPSAC** approach.

- Line detection can be conveniently cast as an Uncapacited Facility Location (UFL) problem [52] that enables correct selection and clustering of contours. Contour triplets are used to establish different calibration hypothesis that give raise to different UFL instances. These multiple UFL instances can be combined in a large Hierarchical Facility Location (HFL) [53] problem for whose solution is the solution of the UFL instance with lowest energy. We will dub this as the **RANSAC-UFL** approach.

3.2.4 Model Fitting using an Exemplar Based Clustering and MAPSAC

Fitting a calibration to a set of curves in the $\boldsymbol{\omega}$ can be taken as the problem of fitting a 3D plane to a set of points. We first transform the conic curves in the image into circles using a suitable transform based on a and s and then derive the equation of the calibration plane, as depicted in the formulation bellow.

Knowing a and s (equation 3.7) we can apply a projective transformation H, that makes the image skewless and with unitary aspect ratio. There are several such transformations but for the sake of convenience we will consider

$$\mathsf{H} = \begin{pmatrix} a^{-1} & -s & 0 \\ 0 & a & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

meaning that the intrinsic camera parameters are now given by:

$$\mathsf{K}' \sim \mathsf{HK} \sim \begin{pmatrix} f & 0 & a^{-1}c_x - sc_y \\ 0 & f & ac_y \\ 0 & 0 & 1 \end{pmatrix} \sim \begin{pmatrix} f & 0 & c_x' \\ 0 & f & c_y' \\ 0 & 0 & 1 \end{pmatrix}$$

Note that by determining (c_x', c_y'), the original principal point (c_x, c_y) can be easily recovered using a and s.

The transformation H maps any line image Ω into a circle Ω' by:

$$\Omega' \sim \mathsf{H}^{-T}\Omega\mathsf{H}^{-1} \tag{3.11}$$

Knowing that points $\mathbf{r}_+$ and $\mathbf{r}_-$ are conjugate with respect to Ω (equation 3.4), points

$$\mathbf{r}_+' \sim \mathsf{K}'\mathbf{r}_+ \sim \begin{pmatrix} \frac{f}{\sqrt{\xi}} + c_x' & c_y' & 1 \end{pmatrix}^T$$
$$\mathbf{r}_-' \sim \mathsf{K}'\mathbf{r}_- \sim \begin{pmatrix} -\frac{f}{\sqrt{\xi}} + c_x' & c_y' & 1 \end{pmatrix}^T$$

must be harmonic conjugate with respect to Ω' such that $\mathbf{r}_+'^T\Omega'\mathbf{r}_-' = 0$. Using the vector notation for Ω', and after some algebraic manipulation, it comes that

$$\underbrace{\begin{pmatrix} c_x'^2 + c_y'^2 - \frac{1}{\eta} & c_x' & c_y' & 1 \end{pmatrix}}_{\mathbf{\Gamma}} .\mathring{\boldsymbol{\omega}}' = 0 \tag{3.12}$$

with

$$\eta = \frac{\xi}{f^2}.$$

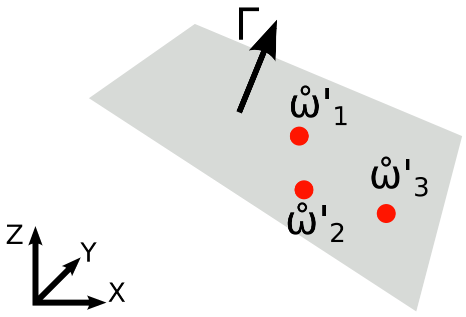


Figure 3.5: World line representation as points in $\mathbb{P}^3$, where the points lie on a plane defined by the calibration parameters η, c'_x, c'_y.

Where $\mathring{\boldsymbol{\omega}}'$ is the lifted representation of a circle ($\boldsymbol{\omega}_1 = \boldsymbol{\omega}_3$ and $\boldsymbol{\omega}_2 = 0$). Equation 3.12 shows that all the circles Ω', that result of transforming line images Ω by K', can be represented by points $\mathring{\boldsymbol{\omega}}'$ in $\mathbb{P}^3$ that must lie in a plane $\boldsymbol{\Gamma}$ that encodes the calibration parameters η, c'_x, c'_y, as shown in Figure 3.5.

Since world lines are often imaged as disconnected segments, due to edge noise or occlusion by objects in the scene, we aim at accurately merging disconnected segments belonging to the same circle and discard circles inconsistent with a line projection. This requires a multi-model fitting approach where the input data – the segments – have to be clustered into circles whose unknown parameters must be estimated at the same time. Since there is no smoothness prior over the input data, we can formulate the problem as an exemplar based clustering. The affinity propagation framework [54] is a robust solver for sparse exemplar based clustering that takes as input measures of similarity between pairs of data points and simultaneously considers all data points as potential exemplars for a cluster. In affinity propagation, real-valued messages are exchanged between data points until a high-quality set of exemplars and corresponding clusters gradually emerges.

We define the similarity S between two segments i and j as a function of the root mean square (RMS) error e for fitting a circle to the pair of segments using [51]. Segment pairs that produce a RMS error above a certain threshold σ are considered unconnected and to pairs that have low error we assign a similarity value based on the error itself. Additionally, an extra data point is created, the outlier exemplar Υ, that is connected to all other data points with a constant similarity υ.

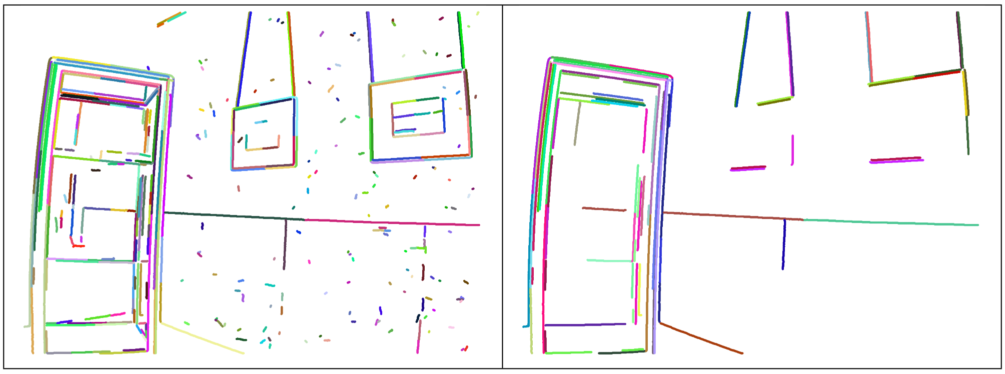


Figure 3.6: Exemplar based clustering using affinity propagation. The left-hand image shows the detected segments and the right-hand image shows the clustering result where we removed the outlier cluster for better visualization.

$$S_{i,j} = \begin{cases} 0, & \text{if } e > \sigma \ \vee \eta_l < \frac{\omega_1}{\omega_6} < \eta_h \ \vee r < r_{min} \\ \frac{1}{e}, & \text{if } e \leq \sigma \\ \gamma, & \text{if } i = j \\ \upsilon, & \text{if } i \vee j = \Upsilon \end{cases}$$

Using this exemplar based clustering approach we can also define geometric restrictions over the clustered segments. From the observation of vector $\mathbf{\Lambda}$, approximating $a = 1$, $s = 0$ and the principal point as the image center, we can observe that $\eta = \frac{\omega_1}{\omega_6}$. Defining a plausible high and low threshold for the distortion parameter, η_h and η_l respectively, we constrain the assignment of similarities to segment pairs that form plausible images of world straight lines. Additionally, we also consider no similarity between pairs that produce a circle with radius bellow $r_{min} = \sqrt{1/\eta_l}$. To increase the circle fitting accuracy of each cluster, we optimize each circle parameters by minimizing the geometric distance to the edges, as proposed in [45]:

$$\mathring{\boldsymbol{\omega}}_{(x_0,y_0,r)} = \sum_{i=1}^{n} d(x_i, y_i, x_0, y_0, r)^2$$

Figure 3.6 shows the result of the exemplar based clustering on a set of detected segments where we can see the correct clustering of segments belonging to the same line and the suppression of improbable circle segments.

From the set of circles retrieved from the exemplar based clustering above, we have ensured that all circles are hypothetically images of straight world lines. Representing each detected circle $\mathring{\boldsymbol{\omega}}_i$ as points in $\mathbb{P}^3$, we now aim at robustly determining the calibration plane that best fits the circles and is consistent with a valid calibration parameters solution. This can be achieved by a sample consensus framework, such as MAPSAC [55], that adds a model prior and maximizes the a posteriori estimate.

Approximating $a = 1$, $s = 0$ and the principal point as the image center, the camera parameters of equation 3.12 define a plane at $x = \eta$. Using this model prior we will use MAPSAC to find the set of circles that best describe a plane in $\mathbb{P}^3$ and are close to the plane $x = \eta$. Models are computed from 3 samples using the a DLT approach on equation 3.12. The likelihood of a circle belonging to a plane is given by the algebraic distance between plane and point, and the model prior is a function of the angle between the plane hypothesis and the vertical plane YoZ.

In this way we ensure that circles used to estimate the final camera parameters form a consensus around a plausible calibration plane. This approach may fail if multiple outlier circles form a consensus around a plane other than the pretended calibration, which happens for example in Figure 3.8, where the curves associated with the power wires form a valid calibration hypotheses.

Figure 3.7 shows the result of the unsupervised calibratixon using the MAPSAC approach in some challenging environments.

3.2.5 Model Fitting using a RANSAC-UFL Approach

The previous section formulated the calibration problem as the fitting of a 3D plane using a MAPSAC approach. This section presents another approach that outperforms the previous one, as we will show in the experimental results section.

Let us assume a calibrated image with distortion for which we want to detect projections of world straight lines. We start by applying a standard edge detector [49], followed by a connected components algorithm in order to obtain several contours e_i that are line projection candidates. We aim at identifying the contours e_i that support conics $\boldsymbol{\omega}_j$ lying on the 2D subspace $\mathbb{S} \in \mathbb{P}^5$ defined by the calibration parameters (equation 3.6). This can be seen as a multi-model fitting problem where the models are the conics $\boldsymbol{\omega}_j$ consistent with the calibration and we want to assign to each contour e_i a model (or discard the contour in case it does not fit any model). Following this, we formulate the detection of lines as an optimal labelling problem that is cast as an UFL instance [52].

Suppose that we need to open a set of facilities ω_j^0 to serve N customers $\mathsf{e}_i \in \mathcal{E}$ whose locations are known. Consider a set $\mathcal{V}_0$ comprising M possible facility locations, the cost c_{ij}^0 for assigning the facility $\boldsymbol{\omega}_j^0$ to the customer e_i and the cost

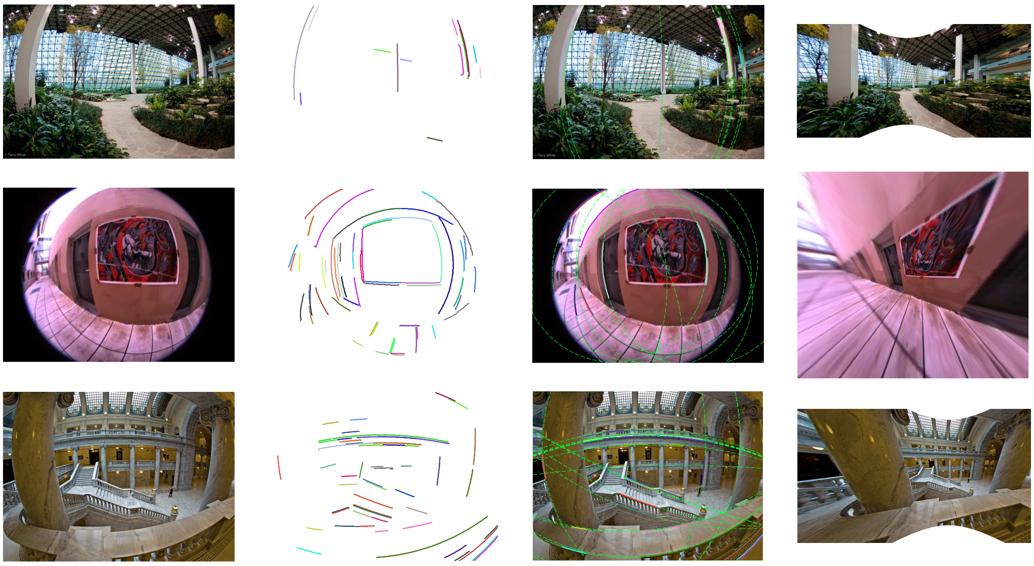
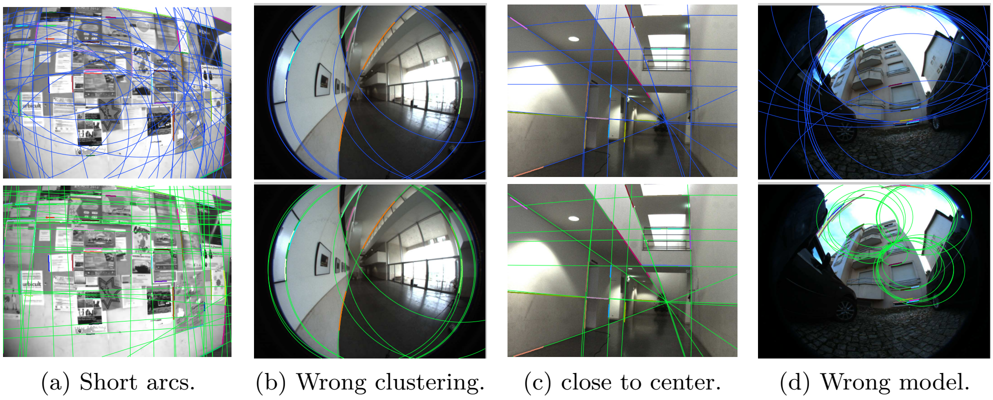


Figure 3.7: Results of the unsupervised calibration using the MAPSAC approach in challenging images. The first column shows the tested image, the second column presents the clustered circular arcs, the third column shows the selected arcs with MAPSAC and corresponding re-projection using the estimated parameters. The last column shows the distortion correction results.



(a) Short arcs. (b) Wrong clustering. (c) close to center. (d) Wrong model.

Figure 3.8: Failure cases of the MAPSAC approach. The top row shows the used circles and the bottom row shows the conic re-projections using the estimated parameters. In (a), the low number of support points for each circle induces a wrong parameters estimation. In (b) we can see that the exemplar based clustering merge the two brown circle candidates at the floor and balconee. In (c), the dominance of circular arcs passing close to the principal point raises ambiguity in the calibration estimation. In (d), the power cable lines in the upper part of the image, that are curves in the world, induce a wrong calibration model selection.

v_j^0 for opening the particular facility $\boldsymbol{\omega}_j^0$. The goal of the UFL problem is to select a subset of $\mathcal{V}_0$ such that each customer is served by one facility and the sum of the customer-facility costs plus the sum of facility opening costs is minimized. This leads to an integer programming problem that is usually formulated using unary indicator variables y_j^0 and binary indicator variables x_{ij}^0, and whose objective is to find the vector $\mathbf{x}^0 = \{x_{11}^0 ... x_{ij}^0 ... x_{NM}^0\}$ such that :

$$\min_{\mathbf{x}^0} \sum_{i=1}^{N} \sum_{j=1}^{M} c_{ij}^0 \, x_{ij}^0 + \sum_{j=1}^{M} v_j^0 y_j^0 \tag{3.13}$$

$$\text{subject to } \begin{cases} x_{ij}^0, \, y_j^0 \, \in \, \{0, 1\}, \, \forall_{i,j} \\ \sum\limits_{j=1}^{M} x_{ij}^0 = 1, \, \forall_i \\ y_j^0 \, \geq \, x_{ij}^0, \, \forall_{i,j} \end{cases} \tag{3.14}$$

The second constraint in equation 3.14 ensures that each customer is assigned to one facility, while the last constraint guarantees that each customer is only served by open facilities. For solving the UFL problem, we use the local message passing approach proposed by Lazic et al [52, 56].

Let $\mathsf{e}_i \in \mathcal{E}$ with $i = 1...N$ be the ith connected component in the image. The objective is to assign to each segment e_i an image conic $\omega_j^0 \in \mathcal{V}_0$ using as few unique models as possible. Consider that segments e_i are the customers and the putative conics ω_j^0 are the facilities. Let cost c_{ij}^0 be the root mean square geometric distance between points of e_i and conic ω_j^0. Let v_j^0 be a constant cost for adding ω_j^0 in the final models assignment. The goal is to select a subset of conics in $\mathcal{V}_0$ such that sum of the consistency measures c_{ij}^0 and the costs $\mathbf{v}_j^0$ is minimized, which corresponds to the minimization of equation 3.13. Figure 3.9 shows the result of the UFL formulation applied to an image with considerable amount of clutter and where straight line projections are often separated. It can be seen that the line extraction algorithm successfully identifies the correct lines and clusters disconnected segments in the same line.

The previous section showed that, given a calibration, we can detect and estimate distorted world line projections. This section considers the unsupervised calibration of the camera, which consists in simultaneously determining a suitable set of calibration parameters along with the corresponding world line projections in the image. This problem can be seen as a HFL instance.

Consider that the facilities ω_j^0 described previously need to be stocked by addi-

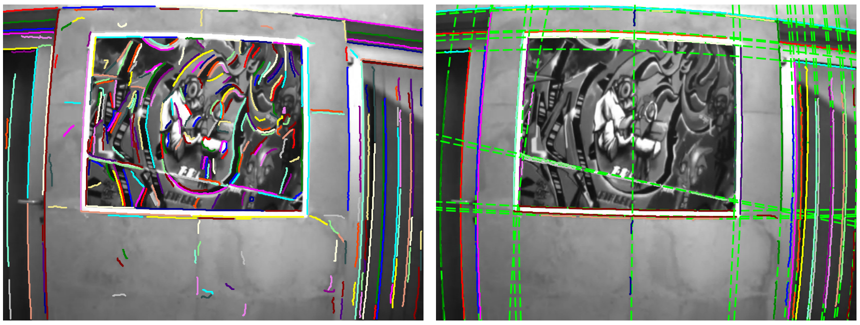


Figure 3.9: UFL for finding lines according to calibration.

tional storage facilities $\mathbf{\Gamma}^1_k$. Consider a set of M facility locations $\mathcal{V}_0$ and L storage facilities $\mathcal{V}_1$. In addition to the costs v^0_j and c^0_{ij} described in the previous section, we now add the cost v^1_j for opening the storage facility $\mathbf{\Gamma}^1_k$, and the cost c^1_{jk} associated with the facility $\mathbf{\Gamma}^1_k$ supplying the facility ω^0_j. The goal of this two layer HFL problem is to find the vector $\mathbf{x} = \{\mathbf{x}^0, \mathbf{x}^1\}$ that minimizes the following function:

$$\min_{\mathbf{x}} \quad \sum_{i=1}^{N}\sum_{j=1}^{ML} c^0_{ij} x^0_{ij} + \sum_{j=1}^{ML}\sum_{k=1}^{L} c^1_{jk} x^1_{jk} + \sum_{j=1}^{ML} f^0_j y^0_j + \sum_{k=1}^{L} f^1_k y^1_k$$

$$\text{s.t.:} \quad \begin{cases} x^0_{ij}, x^1_{jk}, y^0_j, y^1_k & \in \quad \{0,1\} \\ \sum_{j=1}^{M} x^0_{ij} = 1, \forall_i & \wedge \quad \sum_{k=1}^{L} x^1_{jk} = y^0_j, \forall_j \\ y^0_j \geq x^0_{ij}, \forall_{i,j} & \wedge \quad y^1_k \geq x^1_{jk}, \forall_{j,k} \end{cases}$$

The additional restrictions compared to equation 3.14 are that if a facility ω^0_j is closed in layer 0, then ω^0_j will not need to be stocked by a storage facility $\mathbf{\Gamma}^1_j$, whereas if a facility ω^0_j is open, then it must be stocked by a facility in layer 1.

Given a set of connected components $\mathsf{e}_i \in \mathcal{E}$, a set of image conics $\omega^0_j \in \mathcal{V}^0$, and a set of calibration hypotheses $\mathbf{\Gamma}^1_k \in \mathcal{V}^1$, the objective is to assign an image conic to each segment e_i, minimizing the number of conics as well as the number of calibration hypotheses. This problem is cast as a HFL instance with two different layers (Figure 3.10). In addition to the costs c^0_{ij} and v^0_j, we add a new penalization v^1_k for every $\mathbf{\Gamma}^1_k$ contained in the solution. Since only one calibration $\mathbf{\Gamma}^1_k$ is desirable, the penalization v^1_k should be very high.

Calibration hypotheses $\mathbf{\Gamma}^1_k$ are generated from segment e_i triplets using the method described in Table 3.2. For each generated $\mathbf{\Gamma}^1_k$ we compute the M conics ω^0_j that are consistent with the calibration $\mathbf{\Gamma}^1_k$ and minimize the geometric distance of

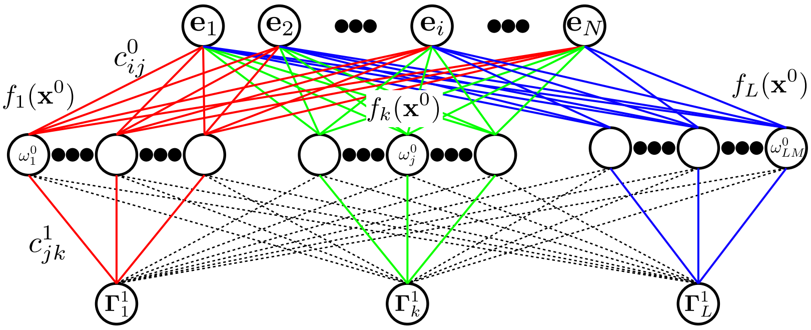


Figure 3.10: HFL formulation for the unsupervised calibration. $\mathbf{\Gamma}^1_k$ are calibration hypothesis and ω^0_j are conics estimated from segments e_i constrained by the associated calibration. The dotted lines indicates infinite costs between nodes.

e_i to ω^0_j (the method described in section 3.2.2), with:

$$j \in \{..., \underbrace{M(k-1)+1, ..., kM}_{j \in \mathbf{j}_k}, ...\}.$$

where $\mathbf{j}_k$ contains the indices of the conics that were generated from a particular calibration $\mathbf{\Gamma}^1_k$. Our HFL formulation retrieves a single calibration by setting the connection costs c^1_{jk} between $\boldsymbol{\omega}^0_j$ and $\mathbf{\Gamma}^1_k$ as

$$c^1_{jk} = \begin{cases} 0 & \text{if } j \in \mathbf{j}_k \\ \infty & \text{otherwise} \end{cases}$$

The high combinatorial nature of calibration models and line segments, along with the fact that we aim at jointly detecting+clustering the contours which are projection of lines and finding the calibration parameters, motivate the use of a HFL approach over other multi-model fitting approaches [57]. Being formulated as a HFL problem, the unsupervised calibration algorithm can be computationally intensive if the number of segments e_i and/or the number of calibration hypothesis $\mathbf{\Gamma}_k$ is high. From our experiments, an HFL formulated as in Figure 3.10, where approximately 100 segments e_i and 50 calibration hypothesis $\mathbf{\Gamma}^1_k$ are used, lead to unpractical computational time of more than 10 minutes. We show that our HFL problem can be efficiently solved as a minimization over a calibration dependent function, as depicted bellow.

Let the proposed formulation of the plumb-line calibration as a HFL problem be:

$$j \in \{..., \underbrace{M(k-1)+1, ..., kM}_{j \in \mathbf{j}_k}, ...\}.$$

where $\mathbf{j}_k$ contains the index of the conics that were generated from a particular calibration $\mathbf{\Gamma}^1_k$. Our HFL formulation retrieves a single calibration by setting the connection costs c^1_{jk} between ω^0_j and $\mathbf{\Gamma}^1_k$ as

$$c^1_{jk} = \begin{cases} 0 & \text{if } j \in \mathbf{j}_k \\ \infty & \text{otherwise} \end{cases} \tag{3.15}$$

Rewriting the initial HFL equation 3.2.5 as

$$\min_{[\mathbf{x}^0,\mathbf{x}^1]} \sum_{k=1}^{L} \left(\underbrace{\sum_{i=1}^{N} \sum_{j \in \mathbf{j}_k} c^0_{ij} x^0_{ij} + \sum_{j \in \mathbf{j}_k} \sum_{k=1}^{L} c^1_{jk} x^1_{jk} + \sum_{j \in \mathbf{j}_k} f^0_j y^0_j + f^1_k y^1_k}_{f_{\Gamma^1_k}} \right) \tag{3.16}$$

we can see that for calibration assignments where more than two hypotheses Γ^1_k are used, the HFL energy of equation 3.16 contains ∞ term (see equation 3.15). Following this, the energy is minimized for labellings containing a single Γ^1_k, and we want to select the $k = k_{min}$ for which $f_{\Gamma^1_k}$ in equation 3.16 is minimized. The indicator variables $x^0_{ij}, x^1_{jk}, y^0_j, y^1_k$ for which $k \neq k_{min}$ and $j \notin \mathbf{j}_{k_{min}}$ are equal to 0, and the second term of equation 3.16 can be discarded. Following this, the objective is to select k and infer $\mathbf{x}^0 = \{x_{ij}\}$ with $j \in \mathbf{j}_k$ for which

$$\min_k f_{\Gamma^1_k}(\mathbf{x}^0) \tag{3.17}$$

and $f_{\Gamma^1_k}$ can be rewritten as

$$f_{\Gamma^1_k}(\mathbf{x}^0) = \min_{\mathbf{x}^0} \sum_{i=1}^{N} \sum_{j \in \mathbf{j}_k} c^0_{ij} x^0_{ij} \; + \sum_{j \in \mathbf{j}_k} f^0_j y^0_j \; + \; f^1_k \tag{3.18}$$

subject to the constraints in equation 3.14.

We have just show that our HFL problem can be efficiently solved as a minimization over a calibration dependent function $f_{\Gamma^1_k}(\mathbf{x}^0)$, which in turn is the result of solving the UFL problem. Following this derivation, we propose a RANSAC-based approach for unsupervised calibration: the **RANSAC-UFL**. As the name suggests,

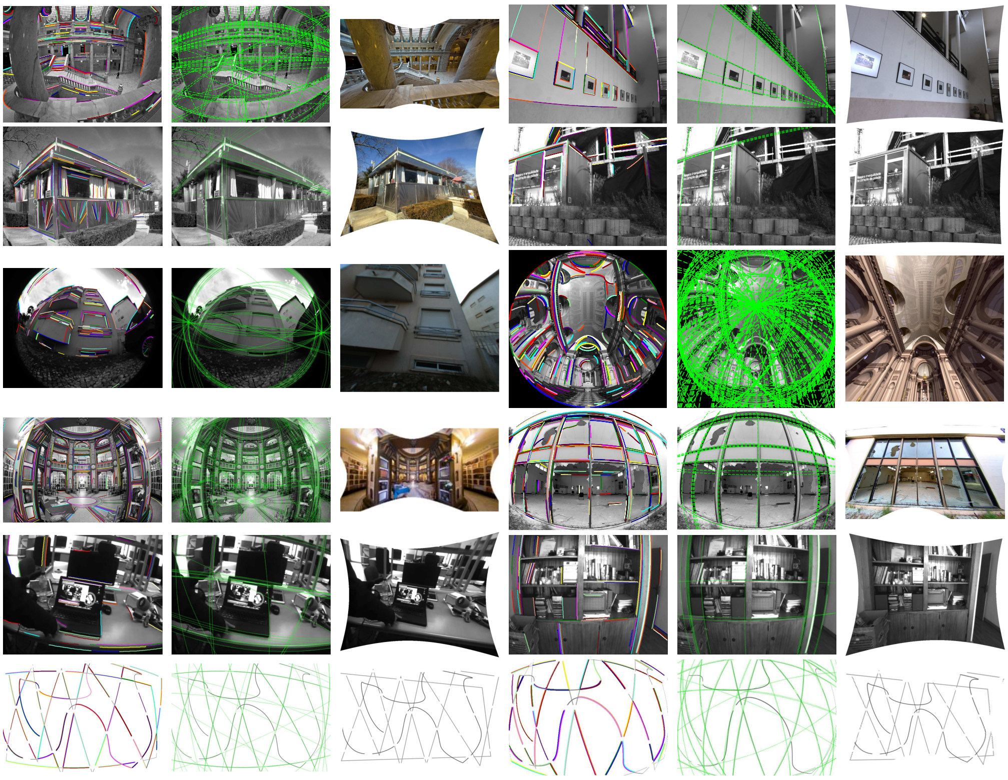


Figure 3.11: Results of the unsupervised calibration using the RANSAC-UFL approach in images mined from the internet. The first column shows the segments e_i highlighted in different colors. The second column shows the detected lines consistent with the calibration. The third column shown the distortion correction. The next 3 columns have the same meaning for different images. The last row shows results in synthetic images.

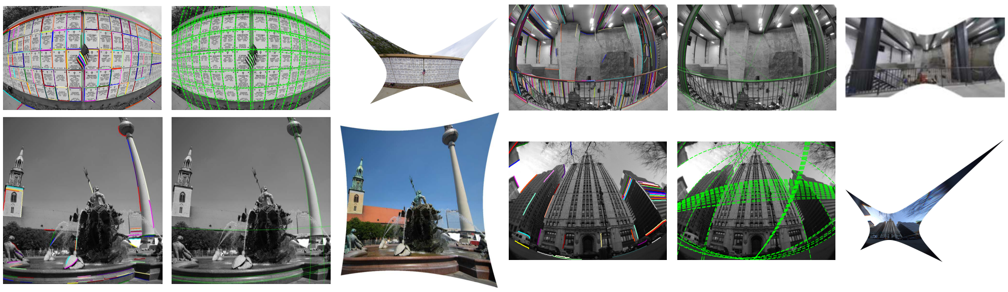


Figure 3.12: Main reasons for failure of the RANSAC-UFL approach. On the top row, the absence of long segments preclude the generation of a valid calibration hypothesis for the RANSAC-UFL framework. The second row shows an example where a calibration hypothesis containing bad segment detections has the minimal energy, leading to an imprecise calibration estimation. In the third row, the curved fountain borders generated the best model although they are not the projection of lines in the image. In last row, noise in the edge detection over separated the segments of the handrail, generating imprecise curves.

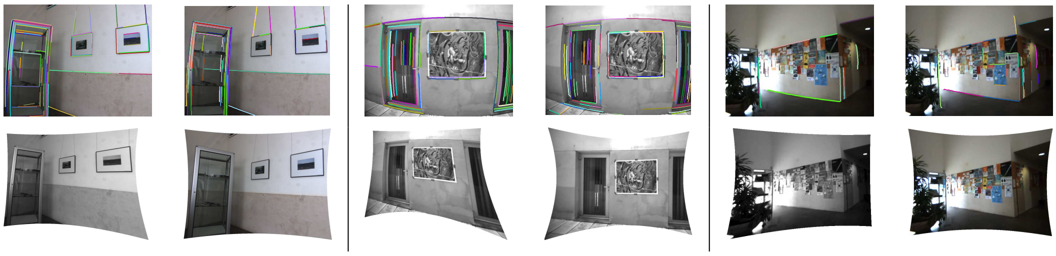


Figure 3.13: Comparison in distortion correction between Bukhari et al. [45] (leftmost image) and our method (rightmost image) in 2 scenes. For each scene we show the segment e_i on the top and the resulting distortion correction in the bottom.

this framework consists in subdividing the general HFL problem into various UFL problems, therefore greatly reducing the computational cost without changing the optimality bounds. With this formulation we reduce by a factor of 10 times the computational time relatively to the standard UFL approach. The RANSAC-UFL randomly samples triplets of connected components, generating calibration hypothesis $\mathbf{\Gamma}_k^1$. Then, each $\mathbf{\Gamma}_k^1$ is evaluated separately using equation 3.18. The calibration with the lowest UFL energy $f_{\Gamma_k^1}(\mathbf{x}^0)$ is the output of the unsupervised calibration.

Figure 3.11 shows the result of the unsupervised calibration using the RANSAC-UFL approach in some challenging environments and synthetic data. Figure 3.12 shows example of failed cases of the proposed method.

3.2.6 Experimental Results

To evaluate the unsupervised calibration accuracy we compared the camera parameters estimation in 8 image of a cluttered environment against ground truth calibration obtained with [24]. In Figure 3.14, we show results of: (i) using 3 manually selected lines per image and calibrating with the minimal solution presented in section 3.2.2; (ii) using the unsupervised calibration with exemplar based clustering and MAPSAC; and (iii) using the unsupervised calibration using RANSAC-UFL. The manually selection of 3 segments e_i to perform the comparison in this experiment provides an evaluation of the minimal solution accuracy derived in section 3.2.2 as well as it shows that edge detection noise greatly influenced the calibration results (high standard deviation). When comparing the performance of the MAPSAC and RANSAC-UFL approaches, we can see that the latter outperforms the former in terms of accuracy and robustness in this heavily cluttered environment. The accuracy of our unsupervised "plumbline" calibration depends on the length and position of the line arcs, as well as on the amount of image clutter. We can notice that the

	Ground Truth	Manual Selection	MAPSAC	RANSAC-UFL
η	$-3.79\text{x}10^{-6}$	$-4.39\text{x}10^{-6} \pm 3.64\text{x}10^{-7}$	$-4.15\text{x}10^{-6} \pm 4.47\text{x}10^{-7}$	$-4.08\text{x}10^{-6} \pm 2.97\text{x}10^{-7}$
c_x	504.67	506.38 ± 33.87	506.73 ± 36.97	518.06 ± 9.12
c_y	397.53	390.06 ± 20.73	414.38 ± 30.76	388.33 ± 20.25
a	0.99	0.99 ± 0.01	1.00 ± 0.01	1.00 ± 0.00
s	$-8.91\text{x}10^{-5}$	$2.48\text{x}10^{-4} \pm 4.59\text{x}10^{-4}$	$1.38\text{x}10^{-10} \pm 2.30\text{x}10^{-10}$	0.02 ± 0.04

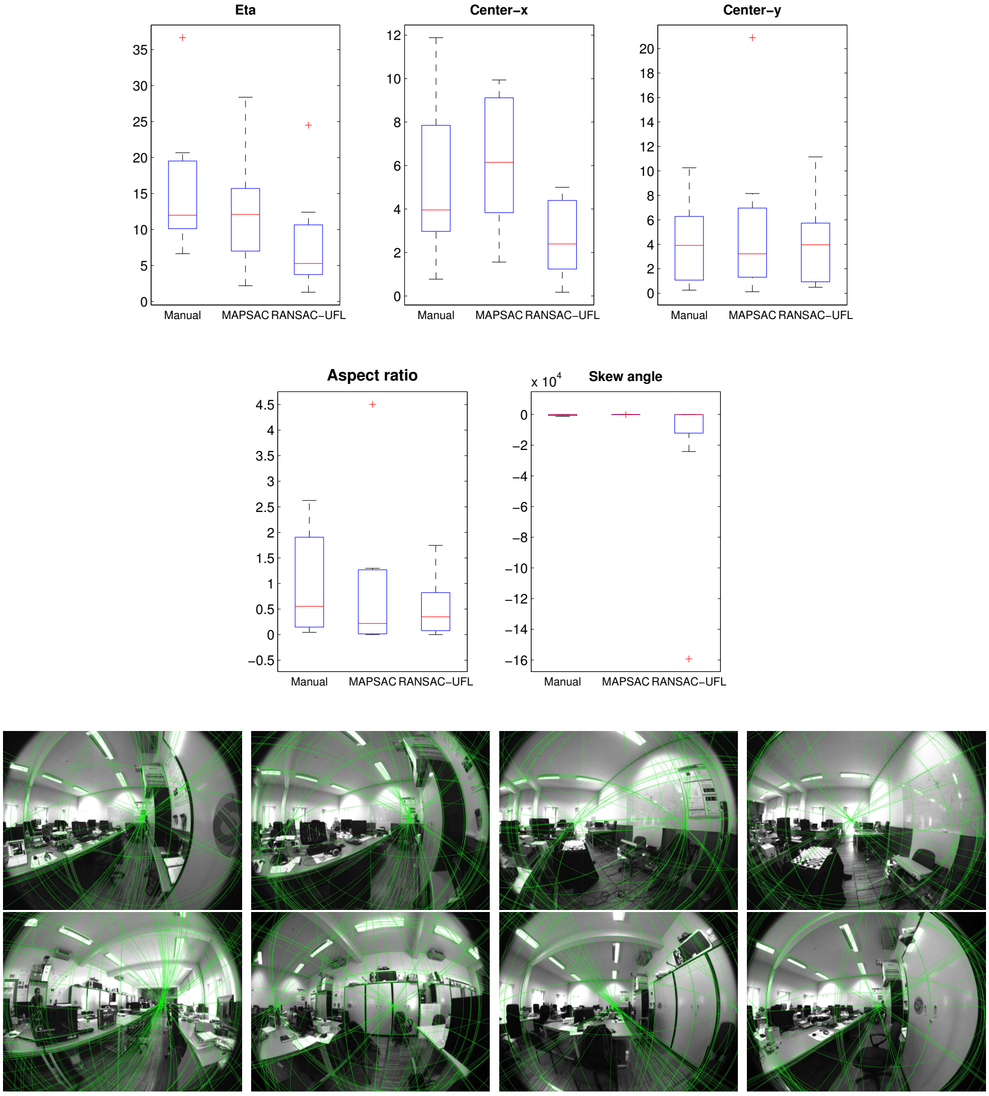


Figure 3.14: Comparison of calibration performance and repeatability. The dataset used has 8 images of a chessboard pattern for ground truth calibration [24] and 8 natural images for comparing the calibration parameters estimation: (i) manually selecting the lines; (ii) using the MAPSAC approach; and (iii) using RANSAC-UFL. The top table shows the mean and standard deviation of the parameters estimation for the 8 images. The middle row show graphics of the percentage deviation of the parameters regarding the from ground truth. The bottom images are the result obtained with RANSAC-UFL in the set of images tested.

fully automatic RANSAC-UFL method outperforms even the manual selection of lines. This is due to the automatic selection of more line segments e_i that contribute to the calibration, and hence lower the pernicious effect of edge noise.

The method in [45] is publicly available and we successfully tested it in some of the examples given by the authors. However, when we ran it in the images used in the experiment of Figure 3.14, the medians of the errors were: 83% for η (std=70%), 59% for c_x (std=104%) and 19% for c_y (std=17%). From our experiences with [45], their approach does not provide plausible estimations in cluttered environments. In Figure 3.13, we compare the RANSAC-UFL approach with [45] by correcting the radial distortion in some images dominated by straight world lines, showing that our approach outperforms [45] for the estimation of the center and amount of distortion in both robustness and accuracy.

We used for our experiments a non-optimized matlab implementation of the RANSAC-UFL that is mathematically equivalent to the HFL formulation. The run-times are dependent on resolution and amount of image clutter. The average times for the experiment of Figure 3.14 (1024x768 resolution) were: 38s for RANSAC-UFL with 50 iterations, which proved to be more than 10x faster than the equivalent HFL, and 15s for the code of [45] that is written in C. The MAPSAC approach has computational times in the same order as the RANSAC-UFL approach.

Note that, in the absence of large segments in the image, the estimation of a and s (Table 3.2, steps 1 to 3) becomes very unstable (see Figure 3.4). To overcome this, we set a and s to reasonable values in images with small segments and proceed with the estimation from steps 4 to 7.

As a final remark on the proposed solution we can emphasize that line detection is a multi-model fitting problem and recent works show that formulating the fitting as an optimal labeling problem with a global energy function is preferable to methods that greedily search for models with most inliers within a threshold [57]. This motivated the use of a global UFL/HFL framework that beats the greedy RANSAC based approach proposed in [45] (as well as our MAPSAC approach) and confirms the conjectures of [57]. For our application the use of a global energy function instead of RANSAC is of particular importance because of the high combinatorial nature and the fact that we aim at jointly detecting and clustering the contours which are projections of lines. This provides an accurate/robust initialization for the subsequent bundle adjustment (equation 3.10).

3.3 Closure

In this chapter we have presented an overview of the calibration approaches used in the literature and proposed a new method for the calibration of a camera using a minimum of 3 natural lines in a single image. Our work is based on a solid geometric interpretation of the line projection under the division model in perspective cameras and is able to estimate the principal point, aspect ratio, skew angle and a coupled parameter of the distortion and focal distance. For unsupervised camera calibration, we devised a framework for the joint line detection and calibration parameters estimation from a single image, that has been tested in challenging situations.

Chapter 4

Radial Distortion Correction System for Endoscopy

This chapter presents the created system for distortion correction in medical endoscopy, which aggregates the main contributions of this thesis into a functional system able to operate in the OR. We propose a complete solution for RD correction in medical endoscopy that comprises the modules and blocks shown in the scheme in Figure 4.1. The medical practitioner starts by acquiring a single image of a checkerboard pattern, the corners in the calibration frame are detected automatically, and both the matrix of intrinsic parameters K_0 and the radial distortion ξ are estimated without further user intervention. After this brief initialization step the processing pipeline on the right of Figure 4.1 is executed for each acquired image. At each frame time instant i we detect the boundary contour $\mathbf{\Omega}_i$, as well as the position of the triangular mark $\mathbf{p}_i$. The detection results are used as input in an Extended Kalman Filter (EKF) which, given the boundary $\mathbf{\Omega}_0$ and marker position $\mathbf{p}_0$ in the calibration frame, then estimates the relative image rotation due to the lens probe rotation with respect to the camera head. This 2D rotation is parametrized by the angle α_i and the fixed point $\mathbf{q}_i$ that serve as input for updating the camera calibration based on the new adaptive projection model presented in chapter 2. Finally, the current geometric calibration K_i, ξ is used for warping the input frame and correct the radial distortion. This processing pipeline runs in real time with computationally intensive tasks, like the image warping and the boundary detection, being efficiently implemented using the parallel execution capabilities of the Graphics Processing Unit (GPU). This chapter introduces the system building blocks and depicts the methods developed to build a robust system, while in the next

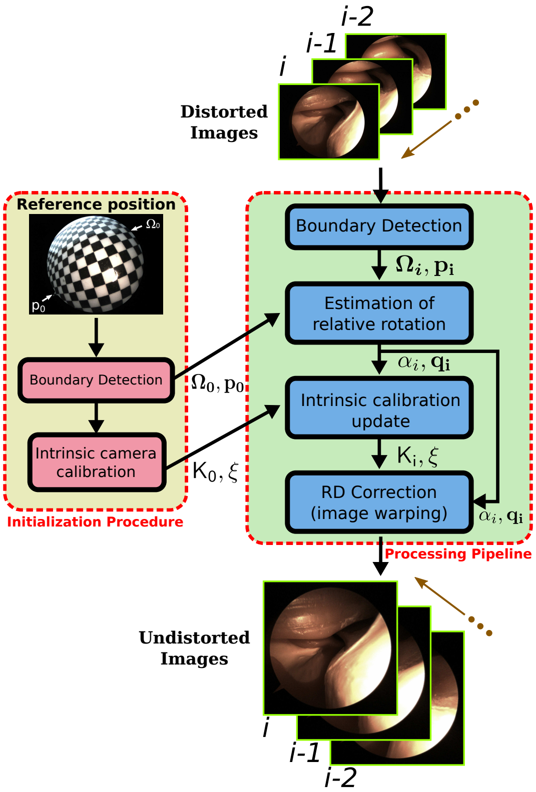


Figure 4.1: Scheme showing the different modules of the system that is proposed for correcting the RD of the image. The left-hand side concerns the *Initialization Procedure* that is performed only once, after assembling the endoscopic lens with the CCD camera. The right-hand side shows the *Processing Pipeline* that is executed at each frame time instant i.

chapter we will give special emphasis on the real-time execution and implementation strategies on a parallel computing hybrid platform.

The *Initialization Procedure*, depicted in Figure 4.1, aims at determining the intrinsic calibration matrix K_0 and the radial distortion ξ when the lens probe is at a reference position $\mathbf{\Omega}_0$, $\mathbf{p}_0$. Planar regular patterns are widely used as a calibration object because they are readily available and simplify the problem of establishing point correspondences. Bouguets toolbox [27] is a popular software that implements Zhang's method [19] for calibrating a generic camera from a minimum of 3 images of a checkerboard. Unfortunately, the Bouguet toolbox does not meet the usability requirements of our application. In practice the number of input images for achieving accurate results is way above 3, and the detection of grid corners in images with RD needs substantial user intervention. Several authors addressed the specific problem of intrinsic calibration or RD correction in medical endoscopes [32–37]. However, these methods are either impractical for use in the OR, or they employ circular dot patterns to enable the automatic detection of calibration points, undermining the results accuracy [38].

Our system reconciles estimation accuracy with usability requirements by using the recent algorithm of [24] that fully calibrates a camera with lens distortion from a single image of a known checkerboard pattern. The intervention of the medical practitioner is limited to the acquisition of a calibration frame from an arbitrary position.

4.1 Robust Detection of the Boundary Contour and Triangular Mark

Rigid endoscopes with exchangeable optics allow rotation of the lens scope with respect to the camera-head, which enables observing the walls of narrow cavities without having to displace the camera. The problem is that the relative motion between lens and camera head causes changes in the calibration parameters that prevent the use of a constant model for correcting the distortion [15]. There is a handful of works proposing solutions for this problem [12–14, 58, 59], but most of them have the drawback of requiring additional instrumentation for determining the lens rotation [12,13,58]. The few methods relying only on image information for inferring the relative motion either lack robustness [14] or are unable to update the full set of camera parameters [59].

Finding the image contour that separates circular and frame regions is important not only to delimit meaningful visual contents, but also to infer the rotation of the lens probe with respect to the camera head. It might seem in a first analysis that the circular region can be easily segmented using a naive pixel intensity approach. Unfortunately there are several issues that must be tackled in order to ensure robustness and accuracy: (i) the light often spreads to the frame region, which undermines the assumption of constant low intensity in the pixels of this region; (ii) the circular region can have dark areas, depending on the imaged scene and lighting conditions, and (iii) highlights, specularity, and saturation often occur, and they affect the performance of many standard segmentation methods. Figure 4.2 shows that the detection of the boundary contour is far from being a trivial problem. We try three active contour approaches [60–62] that are widely used in medical image segmentation. It can be seen that none of them leads to satisfactory results, neither

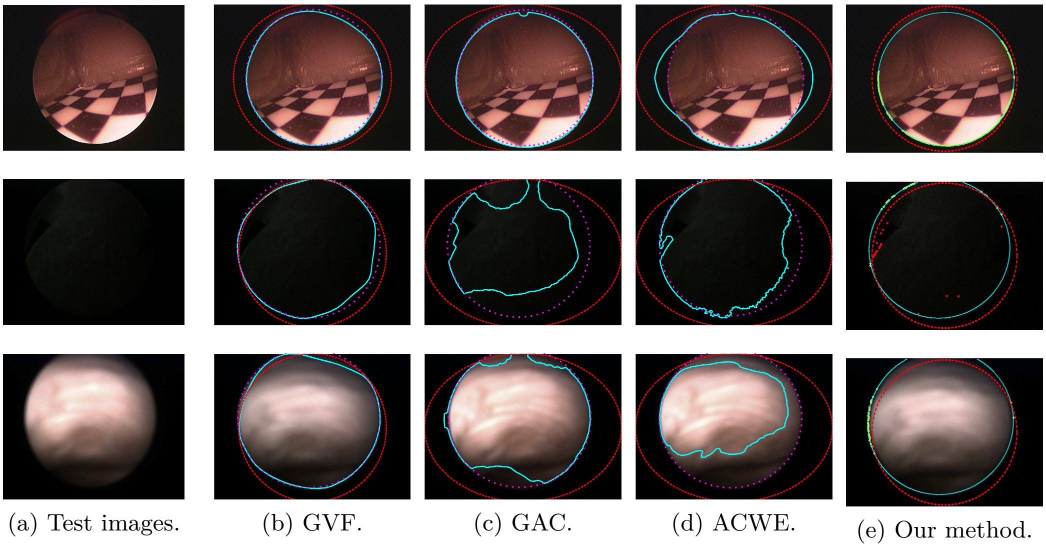


Figure 4.2: Comparison of different approaches for detecting the boundary contour. The first column shows the test images considered in the experiment. From top to bottom we have the situations of light spreading to the frame region, very low illumination, and image saturation. Columns (b) to (d) show the segmentation results obtained using different variants of active contours: (b) *Gradient Vector Flow* [60], (c) *Geodesic Active Contours* [62], and (d) *Active Contours Without Edges* [61]. The purple dashed contour is an overlay of the true boundary that was manually determined, the red curve is the used initialization, and the cyan contour is the final output. The three algorithms took more than 400 iterations to converge. In each case we tried several possible initializations and chose the one leading to the best results. The last column shows the results achieved with our algorithm, using the same initialization as GVF. The method converges to the correct contour in 3 iterations. The same happens when using other initialization hypotheses.

in terms of accuracy, nor in terms of execution time.

An active contour approach has the advantage of avoiding prior assumptions about the shape of the curve to be detected. Since the problem is formulated as the minimization of an energy functional, finding the correct solution for each frame requires iterative optimization. The number of iterations till convergence depends on several factors, such as the initialization and quality of input data, which makes it difficult to ensure real-time performance.

Our proposed approach is related to work by Fukuda et al. [14] and Stehle et al. [59]. The first infers the lens rotation in oblique viewing endoscopes by extracting the triangular mark using conventional image processing techniques. The method assumes that the position of the circular region does not change during operation, which is in general not true, and it is unclear if it can run in real time. Stehle et al. proposes tracking the boundary contour across frames by fitting a conic curve to edge points detected in radial directions. The main difference from our algorithm is that we render a polar image and carry out the search along horizontal lines, which is a very optimized operation in our hybrid platform, allowing the detection on a frame-basis and reconciling robustness and accuracy with low computational cost.

In chapter 2 we presented a new endoscopic camera model that can accommodates the lens rotation into the image formation. In this section we present a new way of robustly inferring the parameters of the lens rotation that are required to update the camera model. Such parameters are estimated by a robust EKF that receives image information about the boundary contour and triangular mark as input. Our dynamic calibration scheme has two important advantages with respect to [59]: (i) the entire projection model is updated as a function of the lens rotation, and not only the RD profile curve; and (ii) the rotation of the lens can still be estimated in the absence of the triangular mark. The approach is validated by convincing experimental results in reprojecting 3D points in the scene.

4.1.1 Tracking the Boundary Contour Across Frames

Yamaguchi et al. [12] present the first adaptive model for oblique-viewing endoscopes designed to project 3D virtual objects in augmented reality applications. The world coordinate system is attached to the camera-head, the intrinsics are assumed to be constant, and the effect of the relative lens motion is modeled by a varying rigid transformation between world and camera reference frames. This transformation comprises two successive rotations by an angle that is measured by a rotary encoder. The approach uses few assumptions, but the model is unnecessarily complicated and the calibration is not practical as it requires the use of opto-tracking. In [58] the endoscope model is interpolated using tens of calibration images acquired for different angular displacements of the lens probe. The calibration process is very troublesome and the determination of the lens rotation angle again requires opto-tracking. Wu et al. [13] simplify the formulation of [12] by considering that it is the camera head that rotates with respect to the lens cylinder, and not the other way around. The effects of the relative motion are modeled by a 2D rotation, on the assumption that the rotation axis coincides with the optical axis. Nevertheless, the angular displacement is determined by opto-tracking and, according to our experiments, their assumption does not hold, in general. Like us, Stehle *et al.* [59], infer the lens rotation directly from the image, without requiring additional sensor modalities. However, our approach has two important comparative advantages: (i) the entire camera projection model is updated as a function of the lens rotation, and not only the RD profile curve. Thus, our endoscopic modeling and calibration can be useful in many applications besides improving the visualization by distortion correction; (ii) unlike in [59], the rotation of the lens can still be estimated in the absence of the triangular mark.

This section discusses the localization of the image contour that separates the circular and frame regions. Since the lens probe moves with respect to the camera head the contour position changes across frames, which prevents using an initial off-line estimation. The boundary detection must be performed at each frame time instant, which imposes constraints on the computational complexity of the chosen algorithm. Several issues preclude the use of naive approaches for segmenting the circular region (Figure 4.2): the light often spreads to the frame region; the circular region can have dark areas, depending on the imaged scene and lighting conditions; and there are often highlights, specularity, and saturation that affect the segmentation performance.

It is reasonable to assume that the curve to be determined is always an ellipse $\boldsymbol{\Omega}$ with 5 degrees of freedom (DOF) [16]. Thus, we propose to track the boundary across frames using this shape prior to achieve robustness and quasi-deterministic runtime. Let $\boldsymbol{\Omega}_{i-1}$ be the curve estimate at the frame time instant $i-1$ as shown in Figure 4.3(a). The boundary contour for the current frame i is updated as follows:

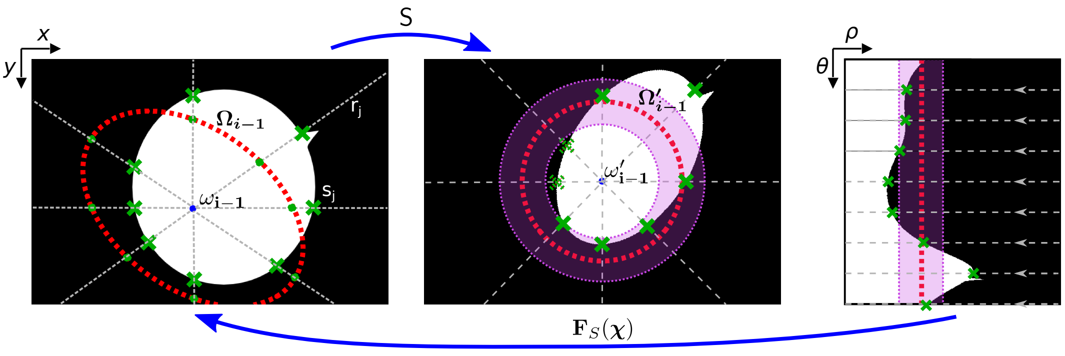
1. Consider a set of equally spaced lines $\mathbf{r}_j$, with $j = 1, 2 \dots N$, that intersect in the conic center $\mathbf{w}_{i-1}$ (this center can be easily computed by selecting the third column of the adjoint of the 3×3 matrix $\boldsymbol{\Omega}_{i-1}$ [16]).

2. For each $\mathbf{r}_j$, interpolate the image signal and compute the 1-D directional derivative at every point location.

3. For each $\mathbf{r}_j$ choose the first maximum of the 1-D derivative when moving from the periphery towards the center. The detected point $\mathbf{s}_j$ is the probable location where the scan line $\mathbf{r}_j$ intersects the boundary contour.

4. Compute $\boldsymbol{\Omega}_i$ by fitting an ellipse to the N points $\mathbf{s}_j$ using a standard RANSAC procedure [50, 63].

In brief, for each frame the tracking algorithm detects edge points along N radial directions of the previous estimate $\boldsymbol{\Omega}_{i-1}$, and updates the boundary contour



(a) (b) (c)

Figure 4.3: Tracking of the boundary contour. The schemes from left to the right relate to the original frame acquired at time instant i, the warped image using the affine transformation S, and the polar image obtained by changing from Cartesian to spherical coordinates. The red dashed overlay relates to the previous boundary estimate, and the crosses are the detected points in the current contour. S maps $\mathbf{\Omega}_{i-1}$ into a unit circle. The purple shaded area is the image region that suffers the polar warp and is where the search is performed.

by performing robust ellipse fitting. Despite being faster than active contours, the runtime is not fully deterministic because the RANSAC is an iterative procedure with variable time till convergence. This time tends to increase with the number of radial lines $\mathbf{r}_j$ and the ratio of outliers in points $\mathbf{s}_j$. In theory, a conic curve can be uniquely determined from a minimum of $N = 5$ points. However, the number of scan lines must be much larger ($N > 180$) in order to ensure robustness with respect to incorrect point contour detection. In the experiments of this chapter, 360 scan lines were used to robustly detect the boundary contour.

To implement this strategy we must derive the mapping function that transforms the current frame i, shown in Figure 4.3(a), into the polar image of Figure 4.3(c). Let $\mathbf{\Omega}_{i-1}$ be the estimation of boundary contour in the frame $i-1$. It is well known that there is always an affine transformation that maps an arbitrary ellipse into a unitary circle whose center is in the origin [16]. Such transformation S is given by

$$\mathsf{S} \sim \begin{pmatrix} r\,\cos(\phi) & r\,\sin(\phi) & -r\,(w_{i-1,x}\,\cos(\phi)+w_{i-1,y}\,\sin(\phi)) \\ -\sin(\phi) & \cos(\phi) & w_{i-1,x}\,\sin(\phi)-w_{i-1,y}\,\cos(\phi) \\ 0 & 0 & 1 \end{pmatrix},$$

where r is the ratio between the minor and major axis of $\mathbf{\Omega}_{i-1}$, ϕ is the angle between the major axis and the horizontal direction, and $(w_{i-1,x}, w_{i-1,y})$ are the non-homogeneous coordinates of the conic center $\mathbf{w}_{i-1}$. The transformation S is used to generate the intermediate result of Figure 4.3(b), and the polar image is

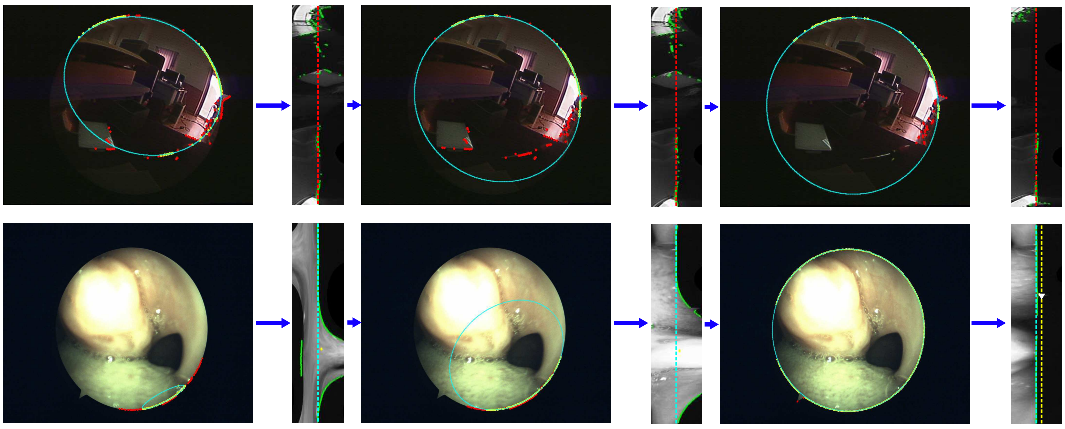


Figure 4.4: Convergence of the boundary contour estimation after poor initialization in two image sequences. We show from left to right a sequence of 3 successively acquired frames with the rendered polar images in between. The initial contour estimate is considerably off but the tracking algorithm quickly converges to the correct solution. The green dots in the polar images are the point detections that are mapped back into the original image using $\mathbf{F}_S$. The RANSAC procedure divides the points into inliers (yellow dots) and outliers (red dots), and estimates the boundary contour at the current frame time instant. Note that after convergence the boundary is mapped into the middle vertical line of the polar image (red dashed overlay).

obtained by applying a change from Cartesian to spherical coordinates.

The edge points are detected by scanning the horizontal lines of Figure 4.3(c) from the right to the left. These points, which are expressed in spherical coordinates $\boldsymbol{\chi} = (\rho, \theta)$, are mapped back into the original image points by the function $\mathbf{F}_S$ of equation 4.1.

$$\mathbf{x} \sim \mathbf{F}_S(\boldsymbol{\chi}) \sim \mathsf{S}^{-1} \begin{pmatrix} \rho \cos(\theta) \\ \rho \sin(\theta) \\ 1 \end{pmatrix} , \tag{4.1}$$

The current conic boundary $\boldsymbol{\Omega}_i$ is finally estimated using the robust conic fitting that avoids the pernicious effects of possible outliers. Figure 4.4 top shows the tracking behavior in a sequence of 3 frames when the initial boundary estimate is significantly off and Figure 4.4 bottom presents results in a live tissue endoscopy.

4.1.2 Tracking the Lens Mark Angle

Locating the position of the triangular mark gives important information about the rotation of the lens relatively to the reference position, as we will see later on. The

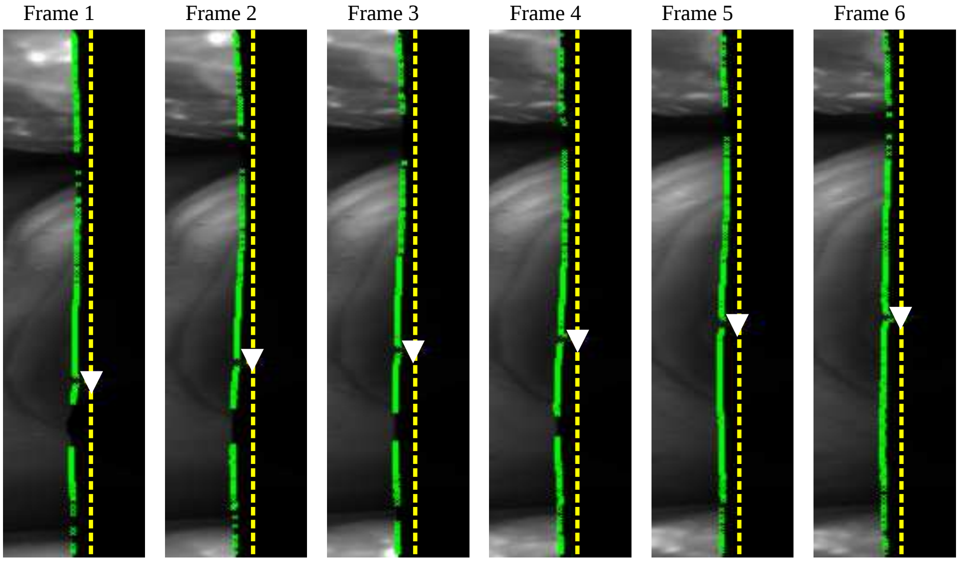


Figure 4.5: Sequence of polar images when the lens is undergoing rotation motion. The vertical yellow line is deviated to the right of the middle vertical line where $\boldsymbol{\Omega}_{i-1}$ is mapped after convergence. The lens mark (white triangle) is robustly detected by selecting the pixel on the yellow line with maximum intensity.

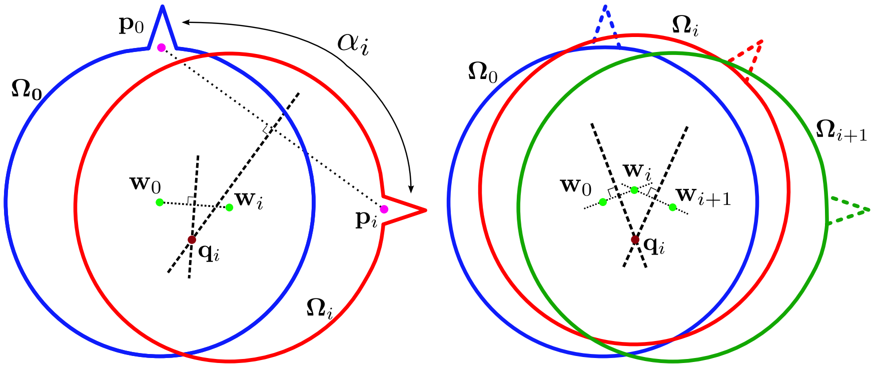
method presented in the previous section to detect the boundary contour across frames guarantees that, upon convergence, the boundary contour gets mapped into a straight vertical line in the polar image. As shown in Figure 4.5, we can scan an auxiliary vertical line that is slightly deviated to the right, and select the pixel location that has maximum intensity.

It is also important to note that the search for contour points is limited to a ring region around the previous boundary estimation (Figure 4.3(b)), saving computational time both in the polar warping and in the horizontal line scanning for lens mark detection.

4.2 Lens Rotation Estimation

The update of the matrix of intrinsic parameters at each frame time instant requires knowing the relative angular displacement α_i and the image rotation center $\mathbf{q}_i$. We now describe how these parameters can be inferred from the position of the boundary contour $\boldsymbol{\Omega}$ and the triangular mark $\mathbf{p}$.

Let $\mathbf{w}_i$ and $\mathbf{w}_0$ be respectively the center of the boundary contours $\boldsymbol{\Omega}_i$ and $\boldsymbol{\Omega}_0$ in the current and reference frames. Likewise, $\mathbf{p}_i$ and $\mathbf{p}_0$ are the positions of the



(a) (b)

Figure 4.6: Computing the image rotation center $\mathbf{q}_i$ when the triangular mark is correctly detected (a), and when there is no mark information (b).

triangular markers in the two images. We assume that both $\mathbf{w}_i$, $\mathbf{w}_0$ and $\mathbf{p}_i$, $\mathbf{p}_0$ are related by the plane rotation $\mathsf{R}_{\alpha_i,\mathbf{q}_i}$ whose parameters we aim to estimate. This situation is illustrated in Figure 4.6a where it can be easily seen that the rotation center $\mathbf{q}_i$ must be the intersection of the bisectors of the line segments defined by $\mathbf{w}_i$, $\mathbf{w}_0$ and $\mathbf{p}_i$, $\mathbf{p}_0$. Once $\mathbf{q}_i$ is known the estimation of the rotation angle α_i is trivial.

Let us now assume that there is no available information about the triangular mark, either because it does not exist in a particular endoscope model, or because it cannot be reliably detected in the image (e.g. it moves outside the frame, is over a dark area and is undistinguishable from the frame region, etc). In this case the rotation center $\mathbf{q}_i$ can be anywhere along the bisector of $\mathbf{w}_i$, $\mathbf{w}_0$. A possibility for disambiguating is to consider other parameters of the boundary contour (e.g. major axis). However, the conic $\boldsymbol{\Omega}$ is usually very nearly a circle and such cues tend to be unstable. Thus, whenever the triangular mark is unknown, the estimation of $\mathbf{q}_i$ requires a minimum of three distinct boundary contours (Figure 4.6b).

In order to avoid under-constrained situations and increase the robustness to errors in measuring $\mathbf{w}$ and $\mathbf{p}$, we decided to use a stochastic EKF [64] for estimating the rotation parameters. The state transition assumes a constant velocity model for the motion and stationary rotation center. The equation is linear on the state variables

$$\begin{pmatrix} \alpha_{i+1} \\ \dot{\alpha}_{i+1} \\ \mathbf{q}_{i+1} \end{pmatrix} = \begin{pmatrix} \mathsf{T} & 0_{2\times 3} \\ 0_{3\times 2} & \mathsf{I}_3 \end{pmatrix} \begin{pmatrix} \alpha_i \\ \dot{\alpha}_i \\ \mathbf{q}_i \end{pmatrix}$$

with T depending on the frame acquisition interval δt

$$\mathsf{T} = \begin{pmatrix} 1 & \delta t \\ 0 & 1 \end{pmatrix} .$$

The measurement equation is nonlinear in α_i and $\mathbf{q}_i$

$$\begin{pmatrix} \mathbf{w}_i \\ \mathbf{p}_i \end{pmatrix} = \begin{pmatrix} \mathsf{R}_{\alpha_i,\mathbf{q}_i} & \mathsf{0}_{3\times 3} \\ \mathsf{0}_{3\times 3} & \mathsf{R}_{\alpha_i,\mathbf{q}_i} \end{pmatrix} \begin{pmatrix} \mathbf{w}_0 \\ \mathbf{p}_0 \end{pmatrix} ,$$

with the two last equations being discarded whenever the detection of the triangular mark fails.

4.3 Radial Distortion Correction

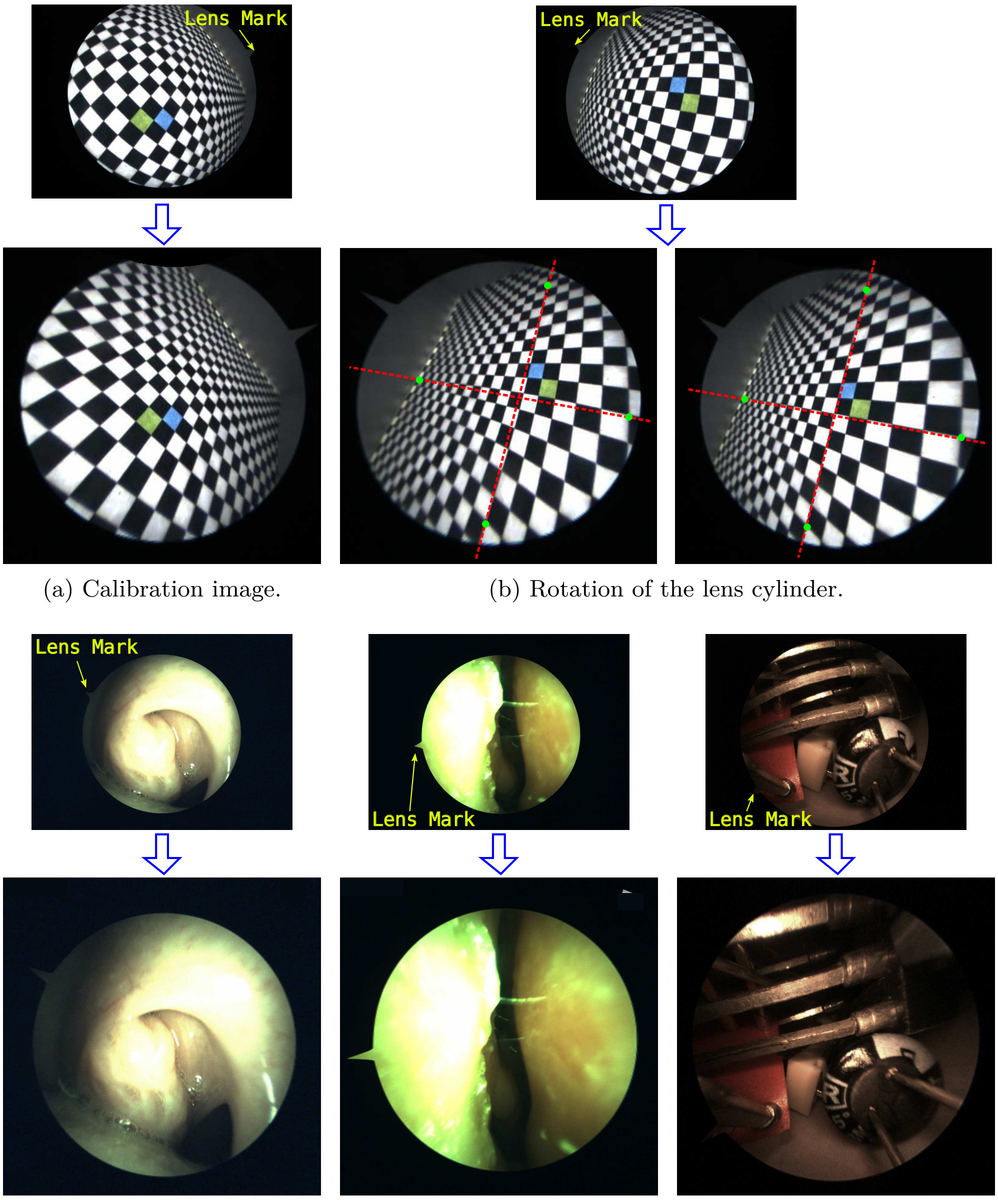
This section discusses the rendering of the correct perspective images that are the final output of the visualization system. Transforming and rendering images is usually done via image warping. Warping filters are topological filters designed to change the shape of the objects of an image. In the continuous domain, given an image $f : U \subset \mathbb{R}^2 \to C$, a warping filter is defined by a map $h : U \to V \subset \mathbb{R}^2$ that should be injective and continuous. From the mathematical point of view, if h is bijective and its inverse h^{-1} is continuous, the map is called a *homeomorphism*. If h and h^{-1} have continuous partial derivatives and the determinant of the Jacobian is not zero, then the map is called a *diffeomorphism* [65, 66].

In computer vision, a warp can also be regarded as a coordinate change. If the coordinates of a feature in the original image are (x, y), the coordinates of the same feature on the warped image would be:

$$h(x, y) = (u, v), \qquad \textit{with } u = f_1(x, y) \textit{ and } v = f_2(x, y) \tag{4.2}$$

For the case of digital images, the domain is discretized (pixels) and brightness is quantized. As pointed out in [66], the efficient warping of an image by a particular transformation should be performed using the inverse mapping method. Thus, we must derive the function $\mathbf{F}$ that maps points $\mathbf{y}$ in the desired undistorted image into points $\mathbf{x}$ in the original distorted frame. From equation 2.4, it follows that

$$\mathbf{F}(\mathbf{y}) \sim \mathsf{K}_i \, \mathbf{\Gamma}_\xi(\mathsf{R}_{-\alpha_i,\mathbf{q}_i''} \, \mathsf{K}_y^{-1} \, \mathbf{y}) \, . \tag{4.3}$$



(a) Calibration image.

(b) Rotation of the lens cylinder.

(c) Oral cavity.

(d) Nasal cavity.

(e) Artificial scene.

Figure 4.7: Radial distortion correction of endoscopic video sequences with lens probe rotation. The original and warped frames are presented in the top and bottom rows, respectively. (a) shows the reference position ($\alpha = 0$) for which the initial calibration is performed. (b) compares the distortion correction results without (left) and with (right) compensation of the lens rotation. (c), (d) and (e) present distortion correction results in various environments. All the results were obtained from a video sequence of 36 minutes with no system recalibration or reinitialization.

K_y specifies certain characteristics of the undistorted image (e.g. center, resolution), $\mathsf{R}_{-\alpha_i,\mathbf{q}''_i}$ rotates the warping result back to the original orientation, and $\mathbf{q}''_i$ is the back-projection of the rotation center $\mathbf{q}_i$

$$\mathbf{q}''_i \sim \begin{pmatrix} q''_{i,x} & q''_{i,y} & 1 \end{pmatrix}^{\mathsf{T}} \sim \Gamma_{\xi}^{-1}(\mathsf{K}_i^{-1}\,\mathbf{q}_i)\,. \tag{4.4}$$

According to feedback from our medical partners, it is very important to preserve object's scale in the center region otherwise the practitioner may be reluctant to adopt the proposed visualization solution. So instead of correcting the RD and rescaling the result to the resolution of the original frame we decided to expand the image periphery and keep the size of the undistorted center region, avoiding loss of information in the image center. This was done by computing the size u of the warped image from the radius of the boundary contour. Let r_d be the distance between the origin and the point ${\mathsf{K}_i}^{-1}\,\mathbf{p}_0$ (the distorted radius). The desired image size u is given by $u = f\,r_u$, where f is the camera focal length, and r_u is the undistorted radius determined using equation 2.9. Accordingly, the matrix K_y must be

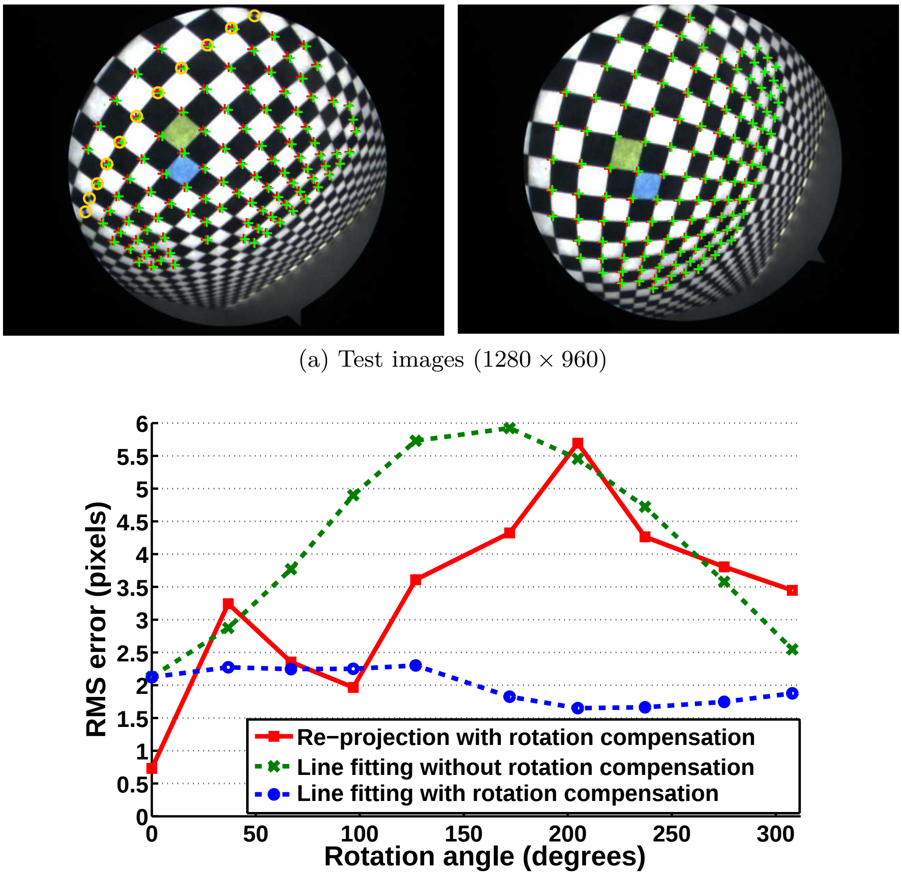
$$\mathsf{K}_y \sim \begin{pmatrix} f & 0 & -f\,q''_{i,x} \\ 0 & f & -f\,q''_{i,y} \\ 0 & 0 & 1 \end{pmatrix} \tag{4.5}$$

with the center of the warped image being the locus where the image rotation center $\mathbf{q}_i$ is mapped. Note that a conjugation of extreme RD with a small original image resolution can produce pixelation in the periphery of the image. This effect is not noticeable when using high resolution input images. Anyway, in older systems this problem can be tackled by either improving the interpolation technique (currently we are using bilinear interpolation) or reducing the overall resolution of the target image.

Figure 4.7 shows the RD correction results for some frames of a video sequence. The examples clearly show the improvements in the scene's perception, and the importance of considering the lens rotation during the correction of the image (Figure 4.7b).

4.4 Experimental Results

The proposed model for calibration update according to the lens relative rotation was validated by re-projecting grid corners onto images of the checkerboard pattern



(a) Test images (1280 × 960)

(b) Root Mean Square (RMS) Errors relatively to different lens rotation angles α (x-axis)

Figure 4.8: Experimental validation of the model for updating the camera calibration. The left-hand side shows the image of the planar grid for a lens rotation angle of $\alpha = 97°$ $\alpha = 127°$ relatively to the reference position. The red crosses are the actual position of the image corners, the green crosses refer to the re-projected grid points using the derived model. There is an average of 130 evaluation points used for the error analysis, many of which are located in the image periphery where the distortion is more pronounced. The red curve in the graphic of (b) shows the RMS value of the re-projection error for different angular displacements α of the lens probe. The two other curves refer to the error of fitting a line to a set of collinear points (the yellow circles in Fig 9(a) - right) after correcting the image distortion with and without taking into account the lens rotation.

acquired for different angles α (Figure 4.8). The SIC was performed for the reference position ($\alpha = 0$), enabling the determination of the matrix K_0, the RD parameter ξ, and the 3D coordinates of the grid points. Then, the camera head was carefully rotated without moving the lens probe in order to keep the relative pose with respect to the calibration pattern. The rotation center $\mathbf{q}_i$ and the angular displacement α_i were estimated for each frame using the geometry in Figure 4.6. Finally, the 3D grid points were projected onto the frame using equation 4.3, and the error distance to the actual image corner locations was measured. The red curve of Figure 4.8b shows the Root Mean Square (RMS) of the re-projection error for different angular displacements α. The values vary between 2 and 5.6 pixels, but no systematic behavior can be observed. We believe that the reason for the errors is more in the experimental conditions than in the camera modeling. Since the images are in a very close range ($\approx$ 10 mm), a slight touch on the lens while rotating the camera head is enough to cause a re-projection error of several pixels. The results of Figure 4.8a not only validate the accuracy of the calibration, but show that having a model that accounts for the lens rotation is of key importance for achieving a correct compensation of the lens distortion.

The presented experiment only focused on the intrinsic parameters, while [12] and [13] consider both intrinsic and extrinsic calibration and employ additional instrumentation. Although no direct comparison can be made, it is worth mentioning that our reprojection error is smaller than [13] and equivalent to [12], where only points close to the image center were considered. From the above, and despite all conjectures, the experimental results clearly validate the proposed model.

4.5 Closure

In this chapter we have presented the building block of the system to calibrate a camera in the operating room and correct the RD. The development of the system led to new methods and models for the following problems: (i) intrinsic camera calibration in the OR with minimum user intervention; (ii) robust segmentation of the circular region; (iii) inference of the relative rotation between lens probe and camera-head using uniquely image information; and (iv) on-line updating of the camera calibration during the clinical procedure. The generation of correct perspectives of the scene in real-time, to be addressed in the next chapter, is likely to improve the depth perception of the surgeon and chapter 6 details an experimental study to quantify the influence of the RD in the success rate of surgeries. With the system presented in this chapter we are able to keep an accurate calibration at all time during the surgical procedure, meaning that this work represents a cornerstone for the future deployment of complex computer vision algorithms in medical endoscopy.

Chapter 5

Accelerating Image Processing for Visualization in Medical Imaging

While the processing of medical images has been mainly dominated by dedicated hardware solutions, general purpose computing platforms have become the choice for many systems due to its less complex implementation. Nevertheless, traditional CPU-based implementations are too slow to be suitable for many real-time clinical applications. In order to overcome this limitation, current medical image computing is being implemented in modern GPU platforms, where this general purpose parallel processor plays an important role in medical diagnosis and analysis [4, 67]. For example, medical images from similar or different modalities often need to be aligned with the reference image as a preprocessing scheme for many further procedures, for instance, atlas-based segmentation, automatic tissue identification and visualization tasks [67].

GPUs have emerged as powerful processors in the last few years. The introduction of the Compute Unified Device Architecture (CUDA) interface [68] has enabled the scientific community to parallelize computationally intensive algorithms and to achieve faster execution times [69]. The Nvidia GF100 and subsequent architectures (also known as Fermi) introduced significant improvements in memory accesses and a drastic increase in compute capability when compared with the previous G80 and GT200 families [70].

In recent years, GPU's have been used in a variety of image processing applications due to the increasing demand in real-time computation and low cost requirements. Yang et al. [71] ported a variety of general image processing algorithms into the GPU using CUDA, such as histogram equalization, edge detection and Discrete Cosine Transform (DCT) encoding/decoding. They achieved significant speed-ups, only possible due to the computational capabilities of the GPU and the high parallelism nature of the algorithms used[1].

In the video compression field, Colic et al. [72] evaluate the performance of motion estimation for video coding in the GPU, presenting a comparison on several motion estimation parallel implementations using memory accesses optimization. The results show speed-ups up to 100× relatively the baseline code for this algorithm.

In the medical image processing field, several applications have seen the power of heterogeneous computing turn heavy algorithms tractable for practical uses. Boyer et al. [2], achieved a speedup of 27.5× in the detection and tracking of white blood cells in video microscopy using the GPU comparatively to the best OpenMP [73] implementation, enabling near real-time analysis. Hartley et al. [3] used a cooperative cluster of CPU-GPU to parallelize the analysis of large biomedical datasets. In their work, the authors present a thorough analysis of optimization techniques for image processing kernels such as color conversion, feature extraction, feature description and histogram computation. The design trade-offs presented and the performance evaluation demonstrated the advantages of using heterogeneous platforms (CPU-GPU) to decrease execution time in classification of stromal development in neuroblastoma images.

Generally, visualization in medical imaging is regarded as the visualization of large collected medical data, which originates from numerical simulations of sensor measurements such as CT and MRI [4]. The visualization of these 2D/3D imaging datasets is indispensable for understanding and making full use of this medical data, being separated into two distinctive groups: surface rendering and volume rendering [74]. The surface rendering is usually implemented via extracting the corresponding isosurface as polygonal mesh from a 3D scalar field (Marching Cubes algorithm [75]). Unlike surface rendering methodologies, volume rendering (also called direct volume rendering) is a technique used to visualize 3D discretely sampled data set by computing 2D projections of a colored semitransparent volume, which can show the whole information of the 3D scalar fields [76]. Volume rendering techniques in the frequency domain have become increasingly popular [4]. After the volume data is transformed into frequency domain, the inverse Fast Fourier or Fast Hartley

[1] Note that the speedups presented in [71] do not take into account the transfer times between host and device - one of the major bottlenecks of General-Purpose Computing on Graphics Processing Units (GPGPU) [69]

Transforms are used to generate the final image by transforming the extracted slice back into the spatial domain [77], which is an operation prone to parallelization on the GPU.

The previous chapter presented the building blocks of a system designed to calibrate and correct the RD of endoscopic images even under lens rotation. This chapter presents the chosen architecture for the distortion correction system and details the implementation design to achieve the high processing frame rates required for use in the operating room. We detail the implementation of the system previously presented in our heterogeneous CPU+GPU platform and present the computational performance using different hardware and image resolution.

5.1 Image Warping for Medical Endoscopy in Heterogeneous Platforms

The techniques for visualization in medical imaging described above are not the only methods that take advantage of parallel implementations. The rendering of 2D images in medical endoscopy has been an active research topic, specially when the objective is to achieve real-time performance in medical application. One particularly relevant application is the correction of RD in small cameras, such as endoscopes, in order to enhance the depth perception and improve the robustness of subsequent image processing algorithms. In recent years, hardware based solutions have appeared to cope with the parallel processing of high resolution images at high frame rates. Advances in very-large-scale integration (VLSI) systems using the distortion model estimation proposed in [32] showed promising results in the correction of the RD using dedicated hardware. Asari presented in [78] an efficient Very-large-scale Integration (VLSI) architecture to correct the RD in wide-angle camera images by mapping the algorithmic steps onto a linear array. Later, in [79], a pipelined architecture was presented that was able to process images at a rate of 30 Mpixels/s. In [80] the authors proposed a VLSI implementation for RD correction that reduced in 61% the number of cells compared to [79] and achieved a throughput of 40Mpixels/s. The recent work presented in [81] reduced at least 69% hardware cost and 75% memory requirement compared to previous works. In [82], the authors presented a comparison of RD correction implementations on a homogeneous multi-core processor, a heterogeneous cell broadband engine, and a Field-programmable Gate Array (FPGA). They concluded that only an FPGA and a fully optimized version of the code running on the Cell processor could provide real-time processing speed (30fps for input images of 2592x1944, which translates into a throughput of 150Mpixels/s).

While previous software-based implementations fail to process large amounts of data in real-time or do not fully model the endoscopic camera, hardware-based solutions lack the versatility to adapt to different devices or lenses (and therefore changes in the projection model in real-time) and involve additional costs and effort to implement.

In this section we describe the parallelization of our algorithms for correcting the RD on our heterogeneous CPU+GPU platform. The natural parallelism underlying warping operations, where all data elements are computed by interpolating a common mapping function, and the efficient memory access procedure herein developed and detailed later on, allow significant speedups. These turn out to be decisive in accomplishing the real-time requirements of the visualization application, especially when the input frames are HD.

We started by proposing the first software approaches to correct radial distortion and update the projection model in images with high resolution, keeping a high frame-rate [5]. This initial approach leveraged the computational power of the GPU to parallelize the most intensive part of the algorithm - the image warping required to apply the non-linear mapping for distortion correction. Figure 5.1 details the implementation presented in [5]. This approach achieved a performance of roughly 30*fps* for HD input images. Since most of the current endoscopic systems are capable of outputting full HD video at 60*fps* (1080p@60Hz) this system is not viable for a functional setup in the OR.

Since the solution we proposed in [5] failed to achieve the required performance, we proposed in [6] a system for acquiring and processing the HD video feed from an endoscope in real-time using a conventional Personal Computer (PC) equipped with an acquisition board and a GPU. This solution is based on the work in [5], that updates the endoscopic camera projection model according to the possible lens rotation at each frame time instant. The system acts like a plug-and-play module that captures the video feed, processes each frame on a regular PC and then outputs the result back into the existing visualization system (Figure 5.2). We verified that an homogeneous multi-core Central Processing Unit (CPU) is not capable of supporting HD real-time video distortion correction, as observed in [82], and the GPU-based implementation in [5] also fails to deliver the necessary frame rates for the latest endoscopic devices. The approach we follow for correcting the radial distortion of a HD video stream is based in a heterogeneous implementation that

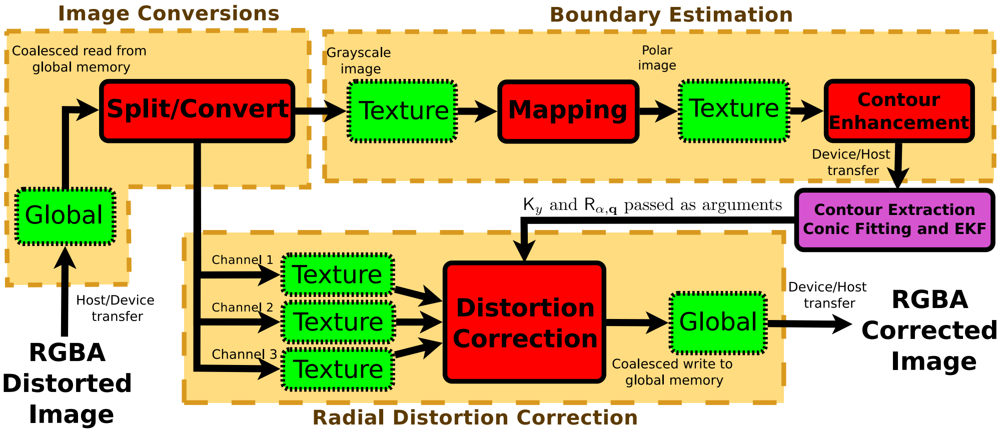


Figure 5.1: Radial distortion correction algorithm steps proposed in [5]. The red blocks represent CUDA kernels calls. The green dotted blocks represent allocated device memory. Except for the purple block, which is computed on the CPU, all processing is performed on the GPU.

uses both CPU and GPU concurrently. This system demonstrates that an hybrid solution, where the computational workload is distributed across the CPU and the GPU in parallel, enables the processing of the video feed (1920x1080 pixels/frame) at frame rates up to 250fps (500Mpixels/s throughput) when implementing efficient memory access patterns on the GPU side of the heterogeneous parallel system. Figure 5.3 presents a schematic view of the algorithm implemented in both CPU and GPU.

As opposed to previous works [32,36,78–83], where only the problem of RD correction using a static projection model is solved, we address the RD correction under projection model changes due to the possible endoscopic probe rotation. The update of the projection model requires additional computation for determining the boundary contour of the meaningful region of the image and the relative lens rotation [5]. In order to achieve higher processing performance we execute both GPU and CPU code concurrently. As shown in Figures 5.3 and 5.4, the processing is split in two POSIX threads: (i) *Pthread 1* is responsible for acquiring the image and performing the CUDA API calls for converting the image colorspace, extracting the boundary contour points and correcting the RD; (ii) *Pthread 2* is responsible for performing the serial part of the boundary contour estimation. This includes a RANSAC, low pass and EKF operations that are detailed in [5]. The high processing frame rate of the system allows the RD correction of the current frame (at time t) based on the

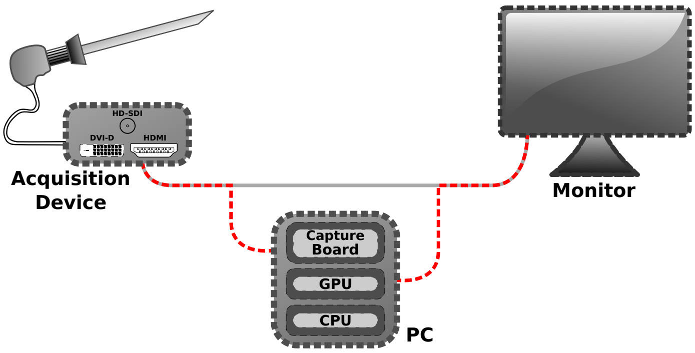


Figure 5.2: Proposed system implementation scheme [8]. The video feed is captured directly from the video output of the acquisition device, processed in our heterogeneous system, and then sent back into the existing visualization system.

boundary parameters of the previous frame (at time $t-1$) without compromising the accuracy of the correction. By using both the CPU and GPU concurrently, we are able to hide the serialized CPU processing workload and therefore substantially increase the system's performance.

The system is divided into 4 main processing blocks:

- Colorspace Conversion - After transferring the image into the GPU (in YUV422 format), a colorspace and grayscale conversion is performed. Each RGB channel plus the grayscale value are written to the global memory and later bound to the texture memory space of the GPU.

- Boundary Detection - Using the grayscale image and the previous boundary estimation parameters from the CPU thread, the boundary contour points are extracted using the procedure described in [5] and the result is passed to the CPU in order to compute the boundary for the next iteration.

- RD Correction - Using the R, G and B channel textures and the previous boundary estimation parameters, the RD is corrected on the GPU and the result is written to the OpenGL global memory buffer for visualization.

- CPU thread - The CPU thread is responsible for robustly estimating the boundary contour from a set of contour points extracted on the GPU. This procedure involves a RANSAC [50], low pass and EKF [64] to robustly fit an ellipse to the boundary contour and estimate the lens rotation parameters.

In this system the video feed is captured directly from the endoscope's control unit video output using the YUV422 transmission format. Taking human percep-

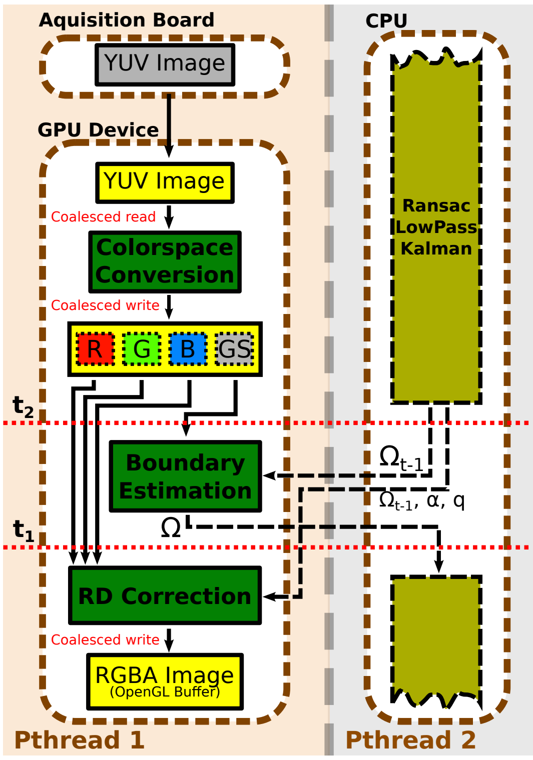


Figure 5.3: Processing stages of the RD correction system proposed in [8]. The system runs in two POSIX threads. *Pthread 1* is responsible for the acquisition and processing of the acquired frame on the GPU. *Pthread 2* is responsible for the serial parts of the algorithm running on the CPU. The threads are synchronized trough conditional variables placed at the red horizontal dashed lines. *Pthread 2* is launched at t_1 and delivers the previous boundary estimation Ω_{t-1} as well as the rotation parameters (α, q) to *Pthread 1*, that is waiting at t_2. In this way, the system is processing the current frame based on the previous boundary estimation.

tion into account for chrominance components, the YUV422 format encodes 2 RGB pixels into a single YUV quadruple. This is of great importance when implementing real-time systems since this video format significantly reduces the necessary bandwidth for transmission and, consequently, the latency of the video stream without compromising the image quality. Moreover, as shown in Figure 5.6, the memory alignment of the YUV422 image is perfectly suited to fulfill the GPU's optimized memory access patterns, as shown later. The RGB values of each pixel are recovered using the following formula:

```
R = clamp(Y + 1.402 x (Cr − 128))
G = clamp(Y − 0.344 x (Cb − 128) − 0.714 x (Cr − 128))
```

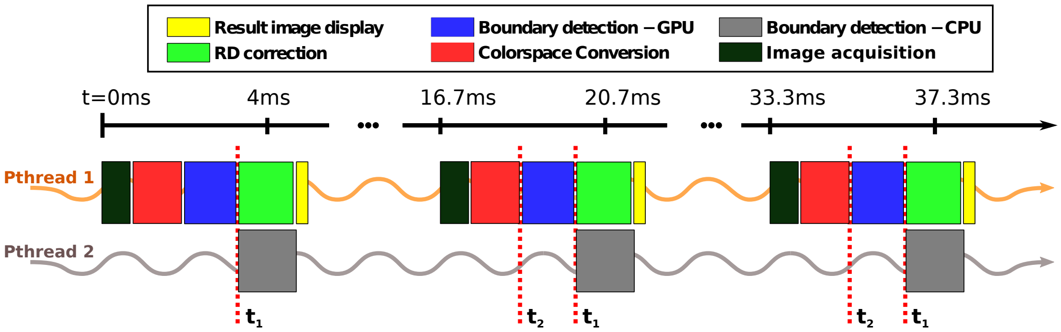


Figure 5.4: Image processing time-line sequence for a generic video stream. Note that the *Pthread2* execute code exclusively on the CPU.

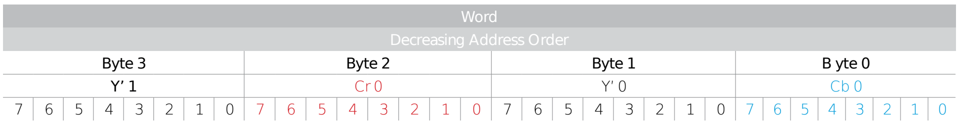

<table>
<tr><td colspan="32">Word</td></tr>
<tr><td colspan="32">Decreasing Address Order</td></tr>
<tr><td colspan="8">Byte 3</td><td colspan="8">Byte 2</td><td colspan="8">Byte 1</td><td colspan="8">B yte 0</td></tr>
<tr><td colspan="8">Y' 1</td><td colspan="8">Cr 0</td><td colspan="8">Y' 0</td><td colspan="8">Cb 0</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table>

Figure 5.5: YUV422 pixel format. Using this format a 1080p image has $1920 \times 1080 \times 2$ *byte*.

```
    B = clamp(Y + 1.772 x (Cb − 128))
```

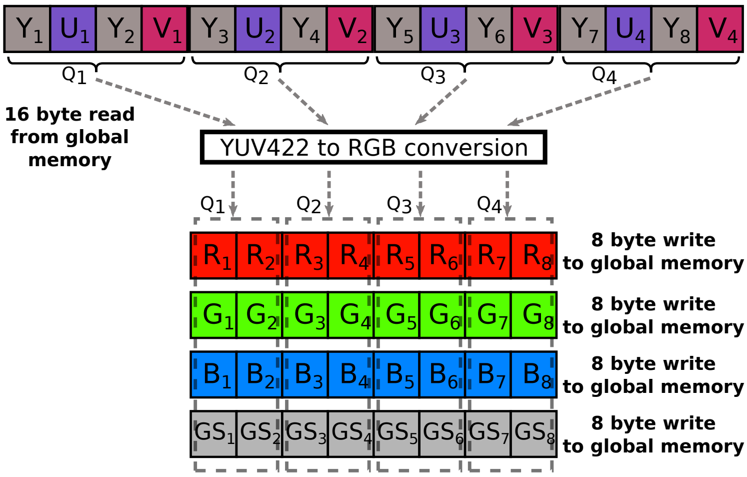
Where the `clamp()` function clamps the output to the `[0,255]` interval.

Note that the acquisition board introduce a minor latency in the video stream (around $0.7ms$) that is not considered in the processing times of the experimental results.

5.2 Optimizing Data Parallelism

The natural parallelism underlying warping operations, where all data elements are computed by interpolating a common mapping function, makes the algorithm for distortion correction specially suited for execution on the GPU.

Since the RD correction problem is mainly memory bound, we devised efficient memory accesses to/from the slow GPU's global memory in order to hide data accesses' latency. The optimization of the GPU's global memory accesses is based on a specific memory alignment procedure, known as *coalescence*, that allows reducing the global number of memory accesses. In this way, threads that are processed simultaneously in batches of 16 (known as half warps) by one multiprocessor can perform the corresponding memory accesses during the same clock cycle. Although



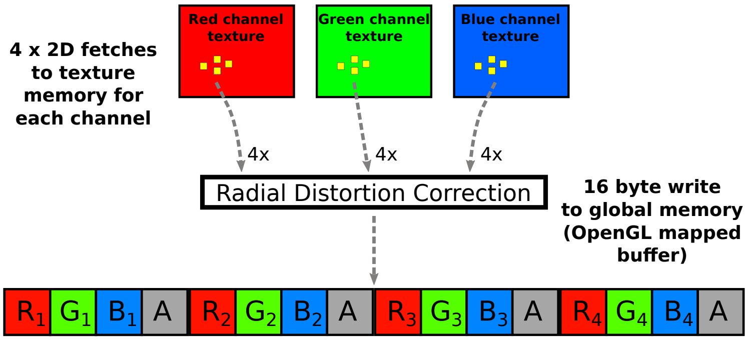
Y1 U1 Y2 V1 Y3 U2 Y4 V2 Y5 U3 Y6 V3 Y7 U4 Y8 V4
Q1
Q2
Q3
Q4
16 byte read from global memory
YUV422 to RGB conversion
Q1
Q2
Q3
Q4
R1 R2 R3 R4 R5 R6 R7 R8
8 byte write to global memory
G1 G2 G3 G4 G5 G6 G7 G8
8 byte write to global memory
B1 B2 B3 B4 B5 B6 B7 B8
8 byte write to global memory
GS1 GS2 GS3 GS4 GS5 GS6 GS7 GS8
8 byte write to global memory

Figure 5.6: Memory access pattern per thread for the *Colorspace Conversion* kernel. Each thread in a half-warp accesses a 16 *byte* word from the global memory (four YUV422 quadruples). For each YUV422 quadruple the thread computes the two corresponding R, G, B and grayscale values and packs them into 8 *byte* words that are written to the corresponding global memory location. Since the data is aligned, the 16 threads of an half-warp read a total amount of $16 \times 16 = 256$ *byte* and write $16 \times 8 = 128$ *byte* for each image channel plus the grayscale using single memory load/store instructions.

more recent GPU architectures, such as the *Fermi*, already incorporate hardware mechanisms to help coalescing memory accesses when data is misaligned, having a careful software design that is aware of the memory alignment performance issues is helpful to improve the computational time in older and even in newer architectures, as shown by the experiments later on.

In the *Colorspace Conversion* kernel of Fig. 5.3, each thread of a half warp accesses the global memory data as a 16 *byte* aligned array corresponding to four YUV422 quadruplets (Fig. 5.6). Each quadruplet is decomposed into two RGB pixels and the data is packed into 8 *byte* words for writing in global memory (each channel and the grayscale value are stored into different memory locations). In this way, the 16 threads of a half-warp read a total amount of $16 \times 16 = 256$ *byte* data and write $16 \times 8 = 128$ *byte* for each image channel plus the grayscale image into global memory. Since data is perfectly aligned, the global memory read/writes are totally coalesced into single memory load/store accesses.

In the *RD Correction* kernel, each thread of an half-warp fetches 4 texture values form the texture memory of the GPU and interpolates the result using the built-in bilinear interpolation hardware. The retrieved values are interlaced into 4 RGBA



Red channel texture
Green channel texture
Blue channel texture
4 x 2D fetches to texture memory for each channel
4x
4x
4x
Radial Distortion Correction
16 byte write to global memory (OpenGL mapped buffer)
R1 G1 B1 A R2 G2 B2 A R3 G3 B3 A R4 G4 B4 A

Figure 5.7: Memory access pattern per thread for the *RD correction* kernel. For each group of 4 RGBA pixels, the radial distortion kernel thread computes the corresponding locations in the distorted space. As the resulting coordinates do not necessarily fall into the regular lattice of the input image, the data is retrieved through 2D texture memory fetches that, through the built-in interpolation hardware, perform the bilinear interpolation of the value. Each thread in a half warp fetches 4 elements of each channel texture and the result is packed into a 16byte word (consisting of 4 RGBA pixels) and written to the global memory (that is mapped to an OpenGL buffer). The data to be written into the global memory by the 16 threads of an half warp is perfectly aligned and therefore the operation is coalesced into a single memory write instruction.

quadruplets and therefore the write operations requested from the 16 threads of an half-warp are coalesced into a single 256 *byte* memory transaction (Fig. 5.7).

One question that can arise is why to convert the YUV422 image into RGB, since the use of the image in YUV422 colorspace would remove the need for a colorspace conversion kernel on the GPU (see Figure 5.3 and 5.6). Keeping the image with this colorspace on the GPU would require the explicit interpolation of the pixel values by the *RD correction* kernel (see Figure 5.7), which would in its turn require the access to 4 YUV422 pixels to write each RGBA output pixel. Moreover, our current system requires the frame buffer to be laid out in RGBA format to be displayed in the screen, and the generation of a gray-scale image for boundary detection, which naturally involves a colorspace conversion. By converting the image into the RGB colorpace we are able to use the dedicated GPU hardware units for bi-linear interpolation, taking the most of the GPU hardware. Obviously, if another interpolation method is requiree (bicubic interpolation for example), this choice presents no advantage other than converting the image for the display format.

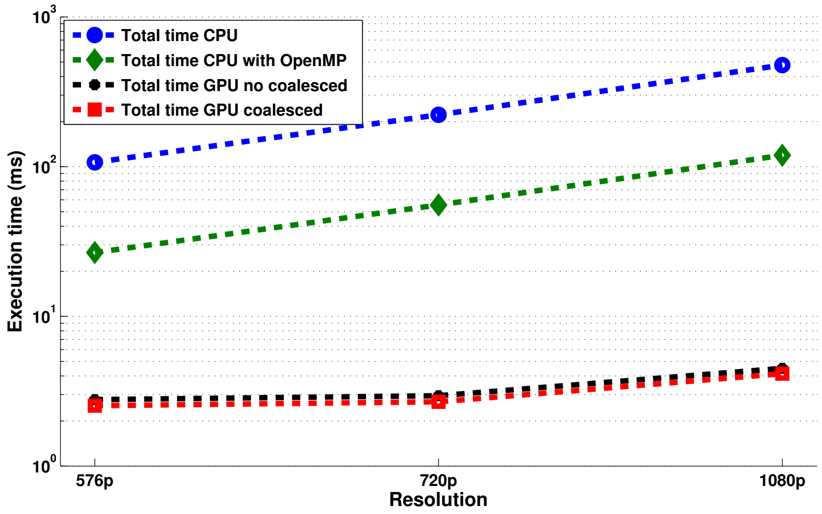


Figure 5.8: Comparing CPU and GPU execution times for correcting the radial distortion of HD images. The times represent the mean time needed to correct each frame of the video stream at different resolutions.

5.3 Experimental Results

Fig. 5.8 compares the processing time of 4 different implementations of our distortion correction algorithm: (i) a naive purely CPU based solution; (ii) a hypothetical CPU version using OpenMP[2] directives [73]; (iii) our heterogeneous approach using a GTX580 GPU without efficient memory accesses; and (iv) our heterogeneous approach using a GTX580 GPU and efficient coalesced memory access patterns. The CPU used in the experiment is an Intel® Core™2 Quad CPU running at 2.40 GHz. The comparison given in Fig. 5.8 shows that the CPU is not able to handle the distortion correction in HD images even when parallelizing the code throughout the multiple CPU cores.

Since the heaviest workload is distributed on the GPU we conducted a series of experiments using different GPUs and different HD resolution inputs. We performed experimental tests on 4 Nvidia GPUs belonging to 3 distinct architectures: (i) a GTX580 (Fermi architecture) with 16 multiprocessors and a total of 512 CUDA

[2] OpenMP can be used for shared-memory architectures, such as conventional commercially available off-the-shelf many-core CPUs of the x86 family. As the workstation CPU has 4 processor cores, OpenMP can be used to parallelize the serial code into all the cores [73]. Doing so would allow achieving a theoretical maximum speedup of 4x relatively to the current CPU code (although the speedup achieved with this kind of parallelization usually does not reach those theoretical maximum values).

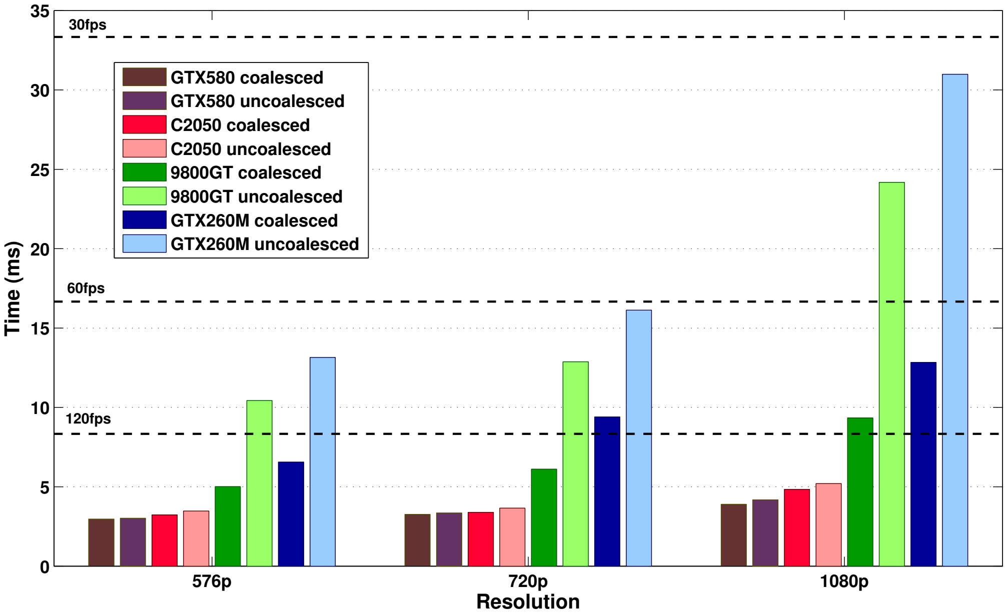


Figure 5.9: Mean total time per frame of the system for different GPUs at different resolutions. Both implementations using coalesced and uncoalesced memory accesses are compared. The resulting output frame size of the system is equal to the input resolution. All four devices can process 1080p video resolution at 60fps when using coalesced accesses to global memory.

cores running at a clock speed of 1544Mhz; (ii) a high-end C2050 (Fermi architecture) with 14 multiprocessors and a total of 448 CUDA cores running at a clock speed of 1150Mhz; (iii) a 9800GT (G80 architecture) with 14 multiprocessors and 112 CUDA cores running at 1500Mhz; and (iv) a GTX260M (GT200 architecture) with 14 multiprocessors and 112 CUDA cores at 1375Mhz. For each different hardware we tested the code using the uncoalesced accesses to the GPU's global memory implementation and also the optimized coalesced version on a sequence of 450 frames. The code is written in C++ using CUDA 4.0.

Fig. 5.9 and Table 5.2 show the mean time needed to process each frame of the input video stream at different resolutions using the 4 GPUs mentioned above. The times were computed by correcting a sequence of 450 endoscopic video frames and computing the mean time per frame for each resolution used. It can be seen that the system can handle full HD resolution at frame rates above 60Hz when using the efficient global memory access patterns. The best processing time for a 1920x1080

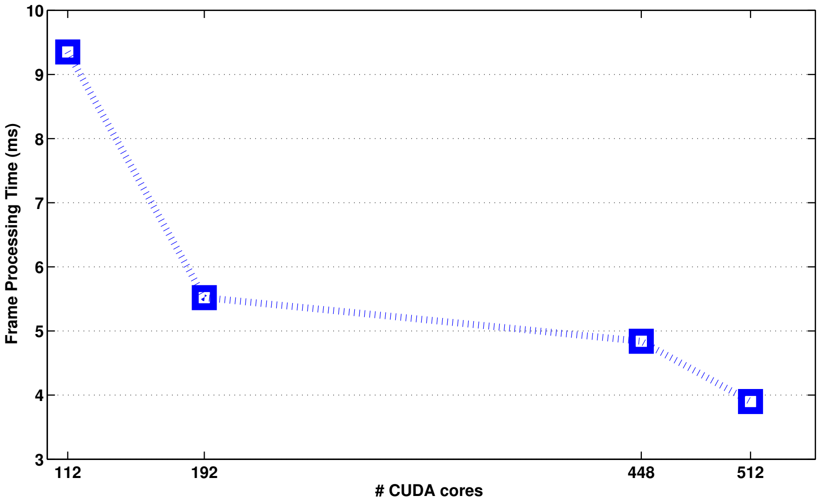


Figure 5.10: Execution time per frame of our system processing a 1080p video stream as a function of the number of cores available on the GPU.

video resolution is achieved with the coalesced implementation in the GTX580 GPU. With this setup, the system is capable of correcting the RD of endoscopic images at a frame rate of approximately 250fps (500Mpixels/sec throughput).

Fig. 5.11 shows the temporal profile of each part of the system individually. It can be seen that, as expected, the use of efficient memory access patterns significantly decreases the processing time of the colorspace conversion and RD correction kernels. Note that the boundary detection on the CPU (textured bar) is overlapped because it runs concurrently with the GPU code (see Fig. 5.4).

Concerning the computation on the GPU presented in Fig. 5.11 and table 5.1, we expect a lower gain in performance when applying our efficient memory access patterns on the C2050 and GTX580 GPUs since Fermi architectures perform intrinsic memory access optimizations when accessing misaligned data from the global memory space. By coalescing data accesses to global memory we obtain gains of 6.6% and 3.7% in the kernel execution time for the C2050 and GTX580 GPUs respectively, and approximately 25% for the older GPUs. This represents a 7% and 63% reduction in the total computation time for the Fermi and G80/GT200 architectures, respectively.

The graphic of Fig. 5.10 shows the performance of our solution as a function of the number of processing cores available on the GPU. It can be seen that, by using a GTX580 GPU with 512 CUDA cores, we achieved a processing time 19.5% inferior

Table 5.1: GPU ocupation percentage for host-device transfers (H-D), device-device transfers (D-D) and kernel executions for the *Colorspace Conversion*, *Boundary Detection* and *RD Correction* (Kernels) in the different GPUs.

	H-D	D-D	Kernels	Total GPU
GTX580 u.	20.8%	27.6%	26.3%	3.37ms
GTX580 c.	22.5%	30.8%	**22.6%**	3.07ms
C2050 u.	21.2%	29.8%	40.3%	4.2ms
C2050 c.	24.7%	32.6%	**33.7%**	3.8ms
9800GT u.	5.0%	9.0%	85.3%	23.2ms
9800GT c.	13.8%	22.2%	**59.6%**	8.3ms
GTX260M u.	4.7%	10.0%	78.9%	29.8ms
GTX260M c.	12.1%	21.4%	**51.1%**	11.6ms

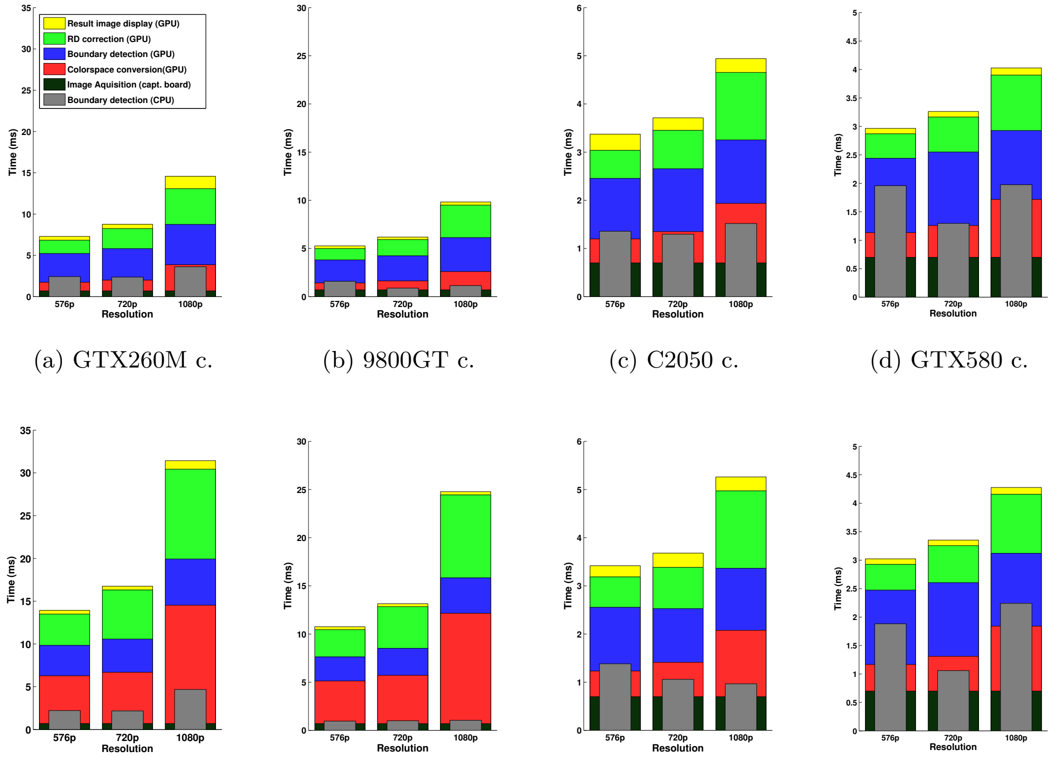
Table 5.2: Mean total time per frame in milliseconds for the different hardware tested at the resolutions used in Fig. 5.9.

	CPU	GTX260M		9800GT		C2050		GTX580	
		u.	c.	u.	c.	u.	c.	u.	c.
576p	106.75	13.15	6.56	10.44	5.01	3.48	3.23	3.02	2.97
720p	221.72	16.12	9.40	12.87	6.11	3.65	3.39	3.35	3.26
1080p	**476.72** (2fps)	30.98	**12.84**(77 fps)	24.18	**9.35**(106 fps)	5.20	**4.84**(206 fps)	4.18	**3.90**(256 fps)

to the time of the system equipped with a C2050 GPU (448 CUDA cores). Fig. 5.10 also shows that the proposed solution is scalable and that it should suit future requirements of this type of medical imaging systems, that expectedly will consist of higher HD image resolutions and frame rates.

Table 5.1 also depicts the difference in GPU occupancy while using the efficient memory access patterns proposed. We can observe that, since the coalesced accesses to the memory significantly reduce the transfer times, the overall time is decreased and the GPU occupancy is more balanced across data transfers and kernel executions.

As shown in Fig. 5.4, as long as the boundary detection on the CPU (running on Pthread 2) does not exceed the sum of the *Image acquisition*, *Colorspace conversion*, *RD correction*, and image display processing times (running on Pthread 1), the



(a) GTX260M c. (b) 9800GT c. (c) C2050 c. (d) GTX580 c.

(e) GTX260M u. (f) 9800GT u. (g) C2050 u. (h) GTX580 u.

Figure 5.11: Time profile of the processing stages for the system using coalesced (c.) and uncoalesced (u.) memory accesses to the GPU's global memory. The *Colorspace conversion* time also includes the transfer of the input image from the host to the device (GPU).

CPU computation of the boundary is entirely hidden by the GPU processing. For example, observing Fig. 5.11(d), at 576p resolution for a GTX580 GPU, we can see that the CPU time (grey bar) is higher than the concurrent GPU stages execution time (*Image acquisition* + *Colorspace conversion* + *RD correction* + *Image display*). In this case the CPU is the bottleneck of the proposed system performance. In the other hand, in most of the remaining setups, the GPU execution time is always higher than the CPU execution and the implementation of a heterogeneous system significantly increased the overall performance of the system, truly balancing the workload distribution between CPU and GPU.

5.4 Closure

In this chapter we detailed the architecture and implementation of the system for correcting the RD in endoscopic images that is capable of correcting 1080p HD images at 250fps. The proposed solution is based on a heterogeneous parallel computing architecture, that uses both the CPU and the GPU concurrently to process the HD video feed, and not only corrects the RD but also adapts the projection model according to the endoscopic lens rotation. All the memory access optimizations on the GPU mentioned turn out to be fundamental for achieving higher processing frame rates and real-time execution on both new and older GPU architectures. This work showed that a careful and efficient usage of conventional hardware outperforms current software-based solutions and competes with dedicated hardware-based and heterogeneous Cell implementations of the RD correction in wide angle lens. With this work we proved that a careful and efficient usage of conventional hardware outperforms current software-based solutions and competes with dedicated hardware-based and heterogeneous Cell implementations of the RD correction in wide angle lens. Our solution is scalable and will support GPUs with even more processing cores, reducing the video processing times and potentially supporting upcoming video systems, such as 4kUHD (3840x2160) or 8kUHD (7680x4320). The developed HD image processing pipeline can be extended for purposes other than RD correction, such as stereo reconstruction or visual SLAM for computer assisted surgery.

Chapter 6

Qualitative Performance Analysis in Arthroscopy

In the last years, MIS has become increasingly popular in many medical fields and is now used not only for diagnosis but also in surgical treatments. MIS requires the use of a camera with a miniature lens that is inserted into the human body through a small incision or a natural orifice, as in the recently developed Natural Orifice Translumenal Endoscopic Surgery (NOTES) technique [84–86]. The images acquired by this camera allow the surgeon to observe the interior of anatomical cavities and control the action of instruments that are manipulated from the outside. Compared with the equivalent open surgery, MIS has the advantage of minimizing trauma, which results in lower complication rates [87], faster patient recovery [88], and shorter hospital stays [89]. However, minimally invasive procedures, also referred to as keyhole surgeries, have the disadvantage of being technically more demanding [90].

The use of long surgical instruments with reduced haptic feedback and limited degrees of freedom, as well as the fact that the entire procedure is executed using endoscopic video as the only guidance, make hand-eye coordination difficult to accomplish. Vakil et. al [91] showed that the correct perception of the anatomical structures and relative depths may reduce complications and contribute to the clinical success of the surgeries. Way et al [92], in a performance analysis of 252 laparoscopic bile duct injuries, identified inaccurate visual perception as the cause of 97% of all such error cases. Therefore, it has often been concluded that MIS requires a longer learning curve than conventional surgery.

Visual perception in general, and depth perception in particular, has always been an issue in MIS. Given that procedures are executed using the endoscopic images as the only guidance, the surgeon has to undergo a long training period to learn how to infer the 3D anatomical structures from the 2D video and ultimately master the surgical techniques [93].

This chapters presents a real-life application of the distortion correction system described in chapters 4 and 5, and presents the results of a preliminary clinical study on the benefits of the RD correction in arthroscopy. This study, to be published [9], presents an evaluation of surgeons' performance while executing a simple arthroscopic procedure using original vs. corrected images. While we did not integrate the author list of this study to avoid partiality of the results and conclusions, we actively contributed to the design and implementation of the study and consider it very relevant for the context of the work herein presented.

6.1 RD Correction for Enhanced Depth Perception in Monocular Endoscopy

Several studies indicate that the lack of depth perception is indeed a source of clinical errors and contributes for a slower learning curve in MIS. It is known that the correct perception of the anatomical structure and relative depths may reduce complications and contribute to the clinical success of surgeries [91]. A study on skill-based errors occurring in endoscopic dacrocystorhinostomy (DCR) surgery revealed that 67% of the errors where due to grasping inefficiency, which resulted in mucosal trauma [94]. In a performance analysis of 252 laparoscopic bile duct injuries, another study identified inaccurate visual perception as the cause in 97% of error cases [92]. A study in the open-biopsy forceps measurement technique compared the error frequency when using original vs. corrected endoscopic images (where the RD was removed). The mean error of the forceps technique in vivo using the original image was 26.5% +/- 5.7% (under-estimation of size), which improved significantly to an error of 2.8% +/- 3.2% with the distortion correction [91]. In laparoscopic bariatric surgery, studies show that it takes a least 50 procedures to master the technique [93], emphasizing the general hard learning curve of MIS. The technical skills can be acquired more quickly if the image gives a better depth perception to the surgeon, as reported in [95], where novices with lower depth perception aptitude in laparoscopy took longer to acquire the required skill to perform the surgery.

One of the ways to improve the overall quality of the procedures is to correct the RD of the endoscopic imaging system. Previous studies have addressed the issue of RD in medical endoscopy and reported that distortion correction improved depth

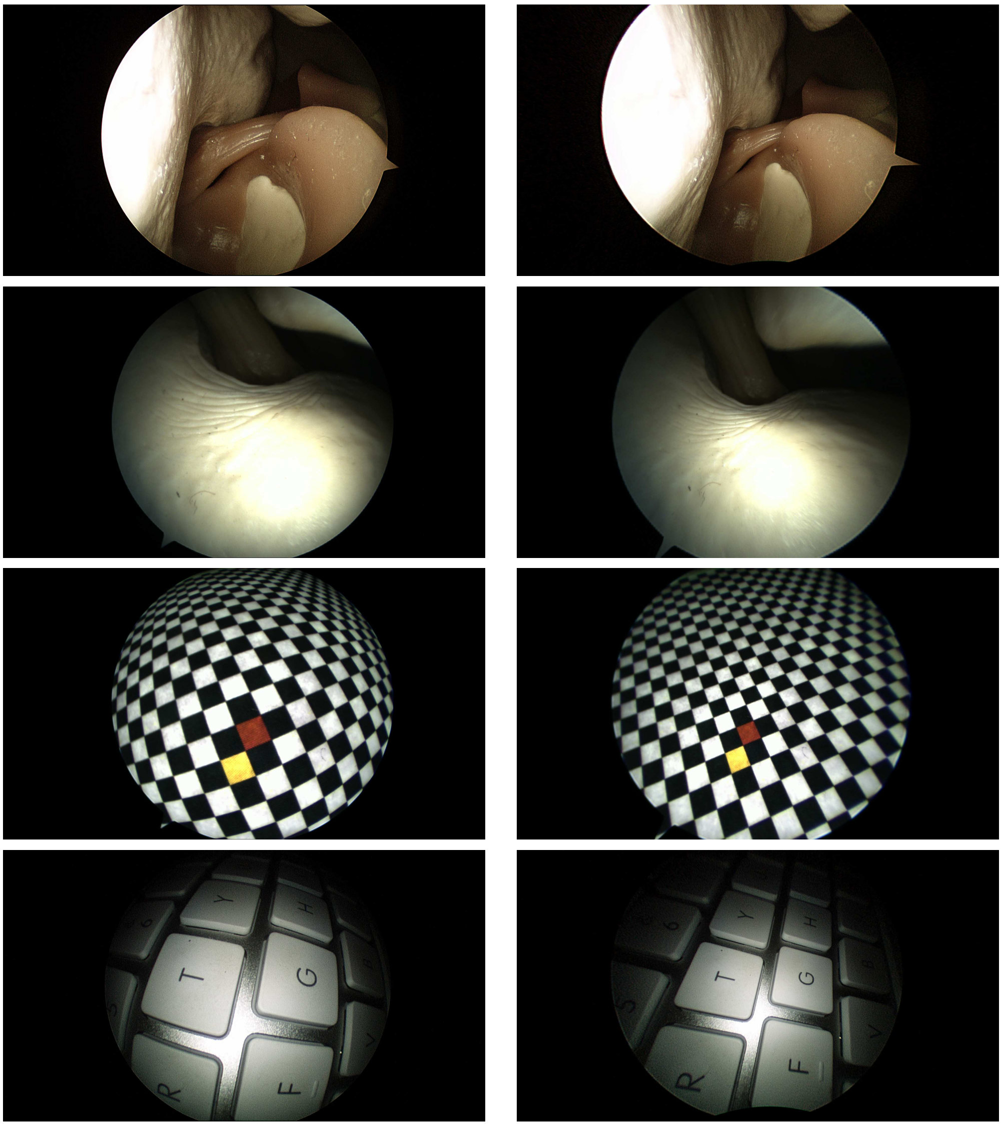


Figure 6.1: Distortion correction in different scenarios. The left column shows the original image and the right column the image after distortion correction.

perception, in both 2D endoscopy and the new 3D endoscopy [96]. RD may also influence the performance of feature descriptors in endoscopic images. In [97] the authors conclude that, from among the different distortion models and calibration methods presented in the literature, the performance of classification result in celiac diseases diagnosis is likely to improve when using the endoscopic camera calibration and distortion correction method of Melo et al. [5].

Figure 6.1 shows some examples of the RD correction resulting from the system presented in the previous chapters. The images were acquired by a 4mm, 30 degree cut arthroscopic probe with a resolution of 1920×1080 pixels. The Appendix B provides more examples of distortion correction in an arthroscopic surgery, where the differences in perception in a real application can be observed.

In order to access the surgical impact of the RD correction we evaluated the performance of a population of twelve arthroscopic surgeons under training, with different levels of expertise. The surgeons were recruited to take a blind test consisting of the execution of a common arthroscopic task in a dry knee model. During this blind test, the surgeon had no information regarding the existence of any manipulation of the image and/or the visualization system used.

In the experiment described in this chapter we used and arthroscopy knee model to perform a common orthopedic arthroscopic procedure: loose body removal from the lateral compartment. For the experimental setup, two endoscopy towers were prepared and equipped with two identical 30 degree cut arthroscopic lenses. **Tower A** was prepared to display the original view, and **Tower B** has the distortion correction system installed. Figure 6.2 presents the setup mounted for the skill test. The resolution of both images, with or without correction, was 1920x1080 pixels displayed at 60Hz. The tests were conducted in a dedicated room and procedure was supervised and scored by two experienced orthopedic surgeons (supervisors). From the surgeons' and supervisors' point of view, the two arthroscopy setups were completely identical. Residents were asked to enter the dedicated room individually and in a random order, and they were invited to perform the removal of a loose body using each of the visualization systems in a sequential manner. The balance in the order of using **Tower A** or **Tower B** was guaranteed following a subject randomization system. After each procedure, both supervisors scored the clinical performance of the surgeon, who then completed a survey about the visualization experience. The time occupied with the survey was also useful as a washout period between the two procedures.

6.2 Experimental Results

In order to evaluate surgeons' performance under different visualization conditions we used an adaptation of the Global Rating Scale (GRS) [98]. This scale measures how well tasks are completed on a 1 to 5 point Likert-type scale. In this study, the scale was adapted for the model used and included four performance parameters:

1. **Instrument Maneuver (IM)**, where 1 is given when the surgeon "*repeatedly*

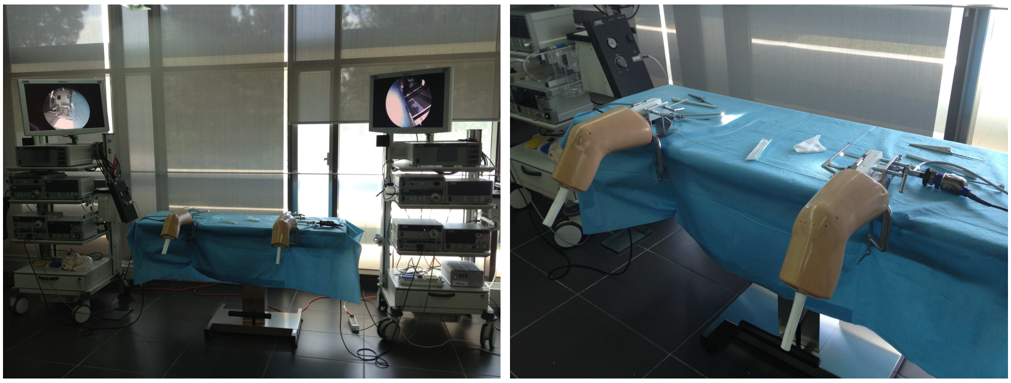


Figure 6.2: Experimental setup for the arthroscopy skill test. Two independent towers presented the original image or the corrected image to each subject as they performed the loose body removal procedure[1].

makes tentative or awkward movements with instruments”, 3 is given when the surgeon presents a“*competent use of instruments, although occasionally appearing stiff or awkward*” and 5 is given when the surgeon makes “*fluid moves with instruments and no awkwardness is observed*”;

2. **Depth Perception (DP)**, where 1 is given when the surgeon “*constantly overshoots target, slow to correct*”, 3 is given when the surgeon experience “*some overshooting or missing of the target*” and 5 when the surgeon “*accurately directs instruments in the correct plane to target*”;

3. **Efficiency (E)**, where 1 is given when the surgeon performs “*many unnecessary, inefficient movements and is constantly changing focus or persisting without progress*”, 3 is given when the surgeon is “*slow but planned movements are reasonably organized with few unnecessary or repetitive movements*” and 5 is given when the surgeon is “*confident, with clear economy of movement and maximum efficiency*”;

4. **Quality of Procedures (Q)**, where 1 is given when the overall quality of the procedure is “*very poor*”, 3 is given when the surgeon appears “*competent*” and 5 is given when the quality is “*clearly superior*”.

Each surgeon that performed the procedure also answered a survey where three dichotomous questions were addressed:

1. What was your perception about the use of a correction method?

[1] Courtesy of Life and Health Sciences Research Institute.

Table 6.1: Answers to surgeons' subjective survey after performing the arthrosopic task with and without distortion correction. The experience column is expressed in number of solo surgeries performed.

Demographic data						Surgeons' answer to Subjective Survey					
						Perception of image manipulation		Wider field of view		Better depth perception	
Subj.	Gen.	Age	Year	Hand	Exp.	Orig.	Corr.	Orig.	Corr	Orig.	Corr
1	M	30	2^{nd}	Right	< 5	No	Yes	No	Yes	No	No
2	M	27	2^{nd}	Right	< 5	No	No	No	Yes	Yes	No
3	M	25	1^{st}	Right	< 5	No	No	Yes	Yes	No	Yes
4	M	32	2^{nd}	Right	< 5	Yes	Yes	Yes	Yes	No	Yes
5	M	28	3^{nd}	Right	$[5-20]$	Yes	No	No	Yes	No	No
6	M	31	2^{nd}	Right	< 5	No	No	No	No	No	Yes
7	M	28	3^{nd}	Right	< 5	No	Yes	No	Yes	No	Yes
8	M	29	5^{th}	Right	$[5-20]$	No	Yes	No	No	No	Yes
9	M	30	4^{th}	Right	< 5	No	Yes	No	Yes	No	Yes
10	M	34	6^{th}	Right	< 5	No	No	No	No	No	Yes
11	M	31	2^{nd}	Right	< 5	Yes	Yes	No	No	Yes	Yes
12	M	28	4^{th}	Right	< 5	No	No	No	No	No	No

2. Did you perceive the existence of a wider field of view?

3. What was your perception about the existence of an improved depth of the field of view?

The answers to this subjective survey, along with the demographic data of the subjects is presented in table 6.1. The population of twelve arthroscopic surgeons included ten males and two females, with ages ranging from 25 to 34 years old (average age of 29.4). All surgeons were right-hand dominant. Regarding year of residency, 50% were in the first or second year, 33% in the third and fourth, and 17% in the last two years. Despite this distribution, only two members of the population had some previous experience in arthroscopic surgery (5 or more procedures).

Objective analysis of the surgeons' performance was obtained through the evaluation form, where the response scale from 1 to 5 was taken as ordinal. Because comparisons involved the same surgeons, the Marginal Homogeneity Test was used. Non-parametric significance tests for two dependent samples are used when we want to study correlated, or matched, samples such as the before-after effect and matched paired studies. Testing marginal homogeneity is often useful in analyzing rater agreement [99]. Since the superiority of the results in the condition with the corrected image (hypothesis-driven) was being tested, the significance level used was unilateral. Because the number of subjects was large enough, the level of significance was obtained using exact methods (not asymptotic).

Figure 6.3 shows the result of the experiment conducted with the surgeons. The unilateral significance level was used and obtained through exact methods, and a 0.05 criteria value was established. As with other significance tests, we consider a measure statistically significant at the 95% level (where p=0.05). Lower values of p mean a greater significance in the results. We can see in Figure 6.3 that the results indicate a statistically significant improvement in all four evaluated parameters when RD correction is in place: instrument maneuver (p=0.008), depth perception (p=0.001), and quality of procedures (p=0.003), with a special emphasis on improved procedure efficiency (p<0.001).

Conversely, the responses to the subjective survey were dichotomous (yes/no), and comparisons involved the same surgeon (paired). Therefore, a McNemar test, or symmetry McNemar chi-square test, was performed to assess statistical significance [100]. The significance level was unilateral for the reasons indicated above and calculated by exact methods. As observed in Table 6.1, despite the surgeons' lack of perception about the use of a correction method to manipulate the image (p=0.188), it was clear that there was a perception of a wider field of view when using RD correction (p=0.031) and improved depth perception during the procedure (p=0.035).

Upon data analysis, it was observed that RD correction seems to have had an effect on the majority of the surgeons' performance, given that a significant improvement in their tasks was obtained with this image correction. Given that only two of the surgeons were slightly more experienced, it was not possible to determine the influence of this variable on the outcomes. No significant correlations (Spearman Test) were obtained between the years of residency and improved ratings. Answers to the subjective survey provided valuable outcomes. Most surgeons did not perceive that there was any image manipulation, compared with their previous experience; this was a positive observation, suggesting that the use of the RD correction did not negatively impact any of the imaging outputs, such as image delays and/or mismatches between instrument movement and image. Additionally, most surgeons perceived a wider field of view and better depth perception when the image was effectively corrected. When analyzing Table 6.1, it is apparent that some surgeons did not perceive an image improvement with RD correction. However, this response was not consistent throughout all questions on the subjective survey. This lack of consistency might be due to the surgeons' reduced experience and therefore low comparability capacity.

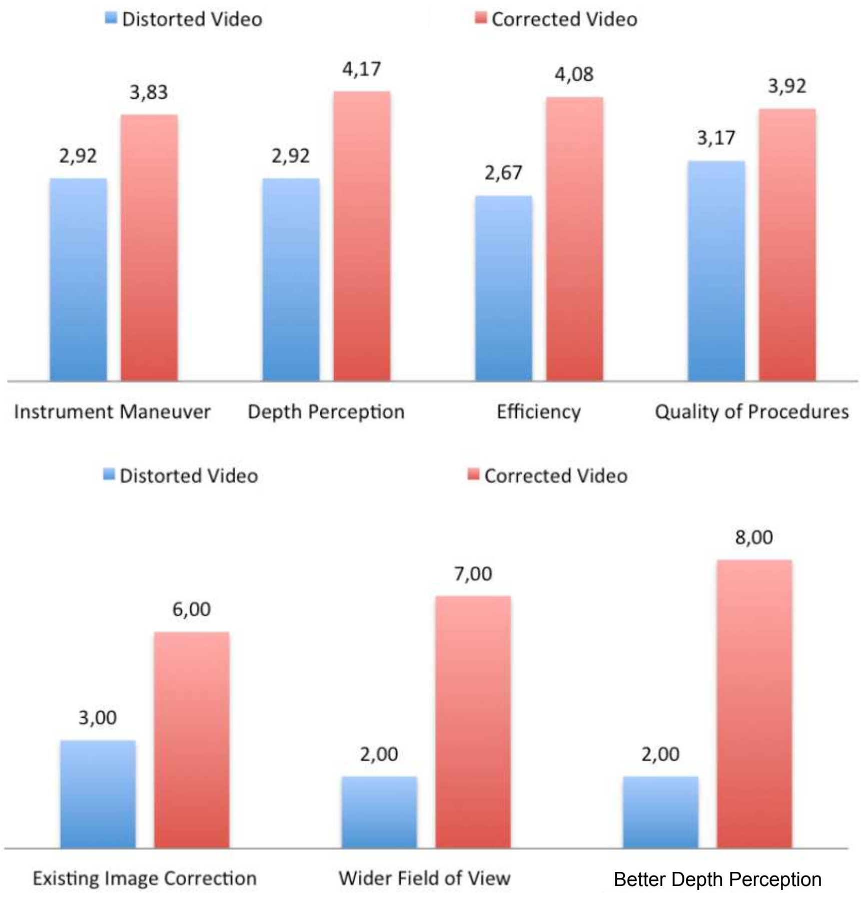


Figure 6.3: Mean score of the subjects' rating given by the evaluators (top), and responses to the subjective survey by the subjects (bottom). For the subjects score (top), a scale from 1 to 5 was used for scoring the different metrics used while using corrected vs. original video. For the subjective survey (bottom), the number of positive answers is plotted for the questions asked to the subjects after completion of each task.

6.3 Closure

The data generated in this pilot study confirmed the hypothesis that the distortion correction provides an improved visualization experience that a surgeon can notice clearly, which influences surgical performance and the learning curve. This pilot study was designed to be performed by orthopedic surgeons who work with small-sized lenses with a wide field-of-view, where the distortion effect is particularly strong. It is reasonable to assume that the observed improvements generalize to other surgical fields that use similar optics such as urology, neurosurgery and pediatric surgery.

Despite the positive trend, the surgical assessment to other procedures within orthopedics and to other surgical fields would provide more conclusive data and evidences of transversality across different procedures. A broader population in both number and experience is also desirable, as are more quantitative metrics (such as execution time and number of surgical movements), to correlate the data with shorter learning curves and improved technical skills, which are of major interest to the surgical community.

Chapter 7

Conclusions

This thesis addressed the general problem of calibration and real-time image processing in medical endoscopy. Camera calibration has been a widely studied topic in the literature. Nevertheless, few works have taken into account the difficulties of modeling and calibrating an endoscope in the OR and the potential of having an accurate calibration procedure for future image processing algorithm to come. Such algorithms will ultimately assist the surgeon during surgery, bringing great benefits to the patients and medical community. We presented the current challenges in this field, and developed a new camera model to cope with the specific properties of the endoscopic lens. A new plumb-line calibration method was also created, that is able to accurately calibrate the camera intrinsics up to a global scale factor using information from the projection of a minimum of 3 straight world lines. This method works in a completely unsupervised manner and can be used in any perspective camera/lens system that presents noticeable RD.

The work underlying this thesis led to the development of a versatile and low cost system for superior visualization in clinical endoscopy by rendering geometrically correct perspective images through RD correction. The solution runs in real time in an heterogeneous hardware and takes into account usability constraints specific to medical procedures. Moreover, it is designed to be used with any type of endoscopic technology, including oblique-viewing borescopes and HD video acquisition. The development of the system led to the creation of new methods and models for the following problems: (i) robust segmentation of the circular region; (ii) inference of the relative rotation between lens probe and camera-head using uniquely image information; and (iii) on-line updating of the camera calibration during the clinical procedure.

The need for real-time performance in HD video feeds image processing was also addressed, which lead to the use of an hybrid architectures to cope with the huge amount of data to be processed. Aiming for real-time execution is essential either at the development stage in order to allow more/faster essays, or at a final stage where minimal time to process a frame is required in order to guarantee a certain image diagnosis capability/quality to the practitioner. The presented system is capable of correcting 1080p HD images at 250fps and is based on a heterogeneous parallel computing architecture, that uses both the CPU and the GPU concurrently to process the HD video feed. The system not only corrects the RD but also adapts the projection model according to the endoscopic lens rotation. Moreover, we perform memory access optimizations on the GPU that turn out to be fundamental for achieving higher processing frame rates and real-time execution on both new and older GPU architectures.

The system was tested in real environment and used to perform an independent arthroscopic skill test that aimed at accessing the pernicious effects of RD in basic arthroscopic tasks. The outcome was positive and suggested that systems for improving visualization in medical endoscopy are of major importance and will eventually become a common feature in the future of MIS.

The contributions of this thesis to endoscopic camera modeling and calibration can be of major importance to many other tasks and applications in computer aided surgery and image guided intervention. Examples include 3D modeling and registration from endoscopic video [101], augmented reality and overlay of pre-operative information [102], and visual SLAM [103] for surgical navigation. Regarding the distortion correction system proposed, as future work we will port the code to many-core systems (multiple CPUs/GPUs for example) in order to increase the computational capabilities of the system and support more complex image processing algorithms in the pipeline with the aim of supporting entire computer assisted navigation pipelines.

Bibliography

[1] R. J. Radke, “A survey of distributed computer vision algorithms,” in *Handbook of Ambient Intelligence and Smart Environments*, pp. 35–55, Springer, 2010.

[2] M. Boyer, D. Tarjan, S. T. Acton, and K. Skadron, “Accelerating leukocyte tracking using cuda: A case study in leveraging manycore coprocessors,” in *Proceedings of the 2009 IEEE International Symposium on Parallel&Distributed Processing*, IPDPS ’09, (Washington, DC, USA), pp. 1–12, IEEE Computer Society, 2009.

[3] T. D. Hartley, U. Catalyurek, A. Ruiz, F. Igual, R. Mayo, and M. Ujaldon, “Biomedical image analysis on a cooperative cluster of gpus and multicores,” in *Proceedings of the 22nd annual international conference on Supercomputing*, ICS ’08, (New York, NY, USA), pp. 15–25, ACM, 2008.

[4] L. Shi, W. Liu, H. Zhang, Y. Xie, and D. Wang, “A survey of gpu-based medical image computing techniques,” *Quantitative Imaging in Medicine and Surgery*, vol. 2, no. 3, 2012.

[5] R. Melo, J. P. Barreto, and G. Falcao, “A new solution for camera calibration and real-time image distortion correction in medical endoscopy - initial technical evaluation,” *Biomedical Engineering, IEEE Transactions on*, vol. 59, pp. 634 –644, march 2012.

[6] R. Melo, M. Antunes, J. Barreto, G. Falcao, and N. Goncalves, “Unsupervised intrinsic calibration from a single frame using a plumb-line approach,” *Computer Vision, IEEE International Conference on*, vol. 0, pp. 537–544, 2013.

[7] K. Hoeller, *Novel Techniques for Spatial Orientation in Natural Orifice Translumenal Endoscopic Surgery (NOTES).* Shaker Verlag GmbH, January 2011.

[8] R. Melo, G. Falcao, and J. P. Barreto, "Real-time hd image distortion correction in heterogeneous parallel computing systems using efficient memory access patterns," *Journal of Real-Time Image Processing*, pp. 1–9, 2013.

[9] R. Duarte, N. Ferreira, A. Oliveira, F. Fonseca, M. Vieira-Silva, and J. Correia-Pinto, "Benefits of radial distortion correction in arthroscopic surgery: a first experimental study on a knee model," *MICCAI*, 2014 (to appear).

[10] J. P. Barreto, "A unifying geometric representation for central projection systems," *Comput. Vis. Image Underst.*, vol. 103, pp. 208–217, 2006.

[11] C. Y. Liu, M. Y. Wang, and M. L. J. Apuzzo, "The physics of image formation in the neuroendoscope," *Child's Nervous System*, vol. 20, no. 11–12, pp. 1433–0350, 2004.

[12] T. Yamaguchi *et al.*, "Camera Model and Calibration Procedure for Oblique-Viewing Endoscope," in *MICCAI*, pp. 373–381, 2003.

[13] C. Wu, B. Jaramaz, and S. G. Narasimhan, "A Full Geometric and Photometric Calibration Method for Oblique-viewing Endoscope," *Int Journal of Computer Aided Surgery*, vol. 15, pp. 19–31, 2010.

[14] N. Fukuda, Y. Chen, M. Nakamoto, and T, "A scope cylinder rotation tracking method for oblique-viewing endoscopes without attached sensing device," *Software Eng and Data Mining*, pp. 684–687, 2010.

[15] J. Barreto, J. Santos, P. Menezes, and F. Fonseca, "Ray-based Calibration of Rigid Medical Endoscopes," in *OMNIVIS*, 2008.

[16] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge University Press, ISBN: 0521540518, second ed., 2004.

[17] C.-H. Chen, T.-K. Yao, and C.-M. Kuo, "Wide-angle camera distortion correction using neural back mapping," *2013 IEEE International Symposium on Consumer Electronics (ISCE)*, pp. 171–172, June 2013.

[18] M. Gadermayr, M. Liedlgruber, A. Uhl, and A. Vécsei, "Evaluation of different distortion correction methods and interpolation techniques for an automated classification of celiac disease," *Computer methods and programs in biomedicine*, 2013.

[19] Z. Zhang, “Flexible camera calibration by viewing a plane from unknown orientations,” in *ICCV*, pp. 666–673, 1999.

[20] L. Alvarez, L. Gómez, and J. R. Sendra, “An algebraic approach to lens distortion by line rectification,” *Journal of Mathematical Imaging and Vision*, vol. 35, no. 1, pp. 36–50, 2009.

[21] R. Hartley and S. B. Kang, “Parameter-free radial distortion correction with center of distortion estimation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 8, pp. 1309–1321, 2007.

[22] A. Fitzgibbon, “Simultaneous linear estimation of multiple view geometry and lens distortion,” in *CVPR*, vol. 1, pp. I–125–I–132 vol.1, 2001.

[23] C. Brauer-Burchardt and K. Voss, “A new algorithm to correct fish-eye and strong wide-angle-lens-distortion from single images,” in *ICIP*, 2001.

[24] J. Barreto, J. Roquette, P. Sturm, and F. Fonseca, “Automatic Camera Calibration Applied to Medical Endoscopy,” in *BMVC*, 2009.

[25] M. Lourenco, J. Barreto, F. Fonseca, H. Ferreira, R. Duarte, and J. Correia-Pinto, “Continuous zoom calibration by tracking salient points in endoscopic video,” *The International Journal of Medical Robotics and Computer Assisted Surgery*, 2014 (to appear).

[26] P. Sturm, S. Ramalingam, J.-P. Tardif, S. Gasparini, and J. a. Barreto, “Camera models and fundamental concepts used in geometric computer vision,” *Found. Trends. Comput. Graph. Vis.*, 2011.

[27] J.-Y. Bouguet, “Camera Calibration Toolbox for Matlab.”

[28] D. C. Brown, “Decentering distortion of lenses,” *Photometric Engineering*, vol. 32, no. 3, pp. 444–462, 1966.

[29] J. P. Barreto and K. Daniilidis, “Fundamental matrix for cameras with radial distortion,” in *ICCV*, 2005.

[30] Z. Kukelova and T. Pajdla, “A minimal solution to radial distortion autocalibration,” *PAMI*, 2011.

[31] Z. Zhang, Y. Matsushita, and Y. Ma, “Camera calibration with lens distortion from low-rank textures,” in *CVPR*, 2011.

[32] K. Asari, S. Kumar, and D. Radhakrishnan, "A new approach for nonlinear distortion correction in endoscopic images based on least squares estimation," *IEEE Trans. Med. Imag.*, vol. 18, pp. 345–354, 1999.

[33] J. Helferty, C. Zhang, G. McLennan, and W. Higgins, "Videoendoscopic distortion correction and its application to virtual guidance of endoscopy," *IEEE Trans. Med. Imag.*, vol. 20, pp. 605–617, 2001.

[34] R. Shahidi *et al.*, "Implementation, calibration and accuracy testing of an image-enhanced endoscopy system," *IEEE Trans. Med. Imag.*, vol. 21, pp. 1524–1535, Jan 2002.

[35] C. Wengert, M. Reeff, P. Cattin, and G. Székely, "Fully automatic endoscope calibration for intraoperative use," in *Bildverarbeitung für die Medizin 2006*, pp. 419–423, 2006.

[36] F. Vogt, S. Krüger, H. Niemann, and C. H. Schick, "A System for Real-Time Endoscopic Image Enhancement," in *MICCAI*, pp. 356–363, 2003.

[37] S. Krüger, F. Vogt, W. Hohenberger, D. Paulus, H. Niemann, and C. H. Schick, "Evaluation of computer-assisted image enhancement in minimal invasive endoscopic surgery.," *Methods Inf Med*, vol. 43, pp. 362–366, 2004.

[38] J. Mallon and P. F. Whelan, "Which pattern? Biasing aspects of planar calibration patterns and detection methods," *Pattern Recognition Letters*, vol. 28, pp. 921–930, Jan 2007.

[39] B. Atcheson, F. Heide, and W. Heidrich, "CALTag: High precision fiducial markers for camera calibration," in 15th *International Workshop on Vision, Modeling and Visualization*, Nov. 2010.

[40] M. Lourenco, V. Pedro, and J. P. Barreto, "Localization in large-scale indoor environments by querying omnidirectional visual maps using perspective images," *IEEE International Conference on Robotics and Automation*, 2012.

[41] J. Bazin, Y. Seo, C. Demonceaux, P. Vasseur, K. Ikeuchi, I. Kweon, and M. Pollefeys, "Globally optimal line clustering and vanishing point estimation in manhattan world," in *CVPR*, 2012.

[42] J. P. Barreto and K. Daniilidis, "Unifying image plane liftings for central catadioptric and dioptric cameras," *OMNIVIS*, 2004.

[43] D. C. Brown, “Close-range camera calibration,” *Photogrammetric Engineering*, 1971.

[44] A. Wang, T. Qiu, and L. Shao, “A simple method of radial distortion correction with centre of distortion estimation,” *J. of Math. Imag. and Vis.*, 2009.

[45] F. Bukhari and M. Dailey, “Automatic radial distortion estimation from a single image,” *Journal of Mathematical Imaging and Vision*, 2012.

[46] J. Semple and G. Kneebone, “Algebraic Projective Geometry,” 1952.

[47] J. P. Barreto, “General central projection systems, modeling, calibration and visual servoing,” *PhD thesis*, 2003.

[48] G. H. Golub and C. F. Van Loan, *Matrix computations*, vol. 3. JHUP, 2012.

[49] J. Canny, “A computational approach to edge detection,” *PAMI*, 1986.

[50] M. Fischler and R. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[51] G. Taubin, “Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation,” *IEEE TPAMI*, vol. 13, no. 11, pp. 1115–1138, 1991.

[52] N. Lazic, B. J. Frey, and P. Aarabi, “Solving the uncapacitated facility location problem using message passing algorithms.,” *Journal of Machine Learning Research*, 2010.

[53] I. Givoni, C. Chung, and B. J. Frey, “Hierarchical affinity propagation,” *CoRR*, 2012.

[54] B. J. Frey and D. Dueck, “Clustering by passing messages between data points,” *Science*, 2007.

[55] P. Torr, “Bayesian model estimation and selection for epipolar geometry and generic manifold fitting,” *IJCV*, 2002.

[56] N. Lazic, I. E. Givoni, B. J. Frey, and P. Aarabi, “Floss: Facility location for subspace segmentation,” in *ICCV*, 2009.

[57] H. Isack and Y. Boykov, “Energy-based geometric multi-model fitting,” *IJCV*, vol. 97, no. 2, pp. 123–147, 2012.

[58] S. D. Buck, F. Maes, A. D’Hoore, and P. Suetens, “Evaluation of a novel calibration technique for optically tracked oblique laparoscopes,” *MICCAI*, pp. 467–474, 2007.

[59] T. Stehle *et al.*, “Dynamic Distortion Correction for Endoscopy Systems with Exchangeable Optics,” in *Bildverarbeitung für die Medizin*, (Berlin), pp. 142–146, 2009.

[60] C. Xu and J. L. Prince, “Snakes, Shapes, and Gradient Vector Flow,” *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, 1998.

[61] T. F. Chan and L. A. Vese, “Active Contours without Edges,” *IEEE Transactions on Image Processing*, vol. 10, no. 2, 2001.

[62] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic Active Contours,” *IJCV*, vol. 22, no. 1, pp. 61–79, 1997.

[63] A. Fitzgibbon, M. Pilu, and R. Fisher, “Direct least square fitting of ellipses,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, pp. 476–480, May 1999.

[64] S. M. Bozic, *Digital and Kalman Filtering.* Edward Arnold, 1979.

[65] Melo, R. and Barreto, João P. and Menezes, P., “Interfaces and Visualization in Clinical Endoscopy,” *Master Thesis*, July 2009.

[66] L. Velho, A. C. Frery, and J. Gomes, *Image Processing for Computer Graphics and Vision.* Springer, 2008.

[67] A. Eklund, P. Dufort, D. Forsberg, and S. LaConte, “Medical image processing on the GPU - past, present and future,” *Medical Image Analysis*, vol. 17, pp. 1073–1094, 2013.

[68] NVIDIA, *CUDA Programming Guide 3.0*, Febuary 2010. Rev 1.

[69] D. Kirk and W.-m. Hwu, *Programming Massively Parallel Processors: A Hands-on Approach.* Morgan Kaufmann, 2010.

[70] C. M. Wittenbrink, E. Kilgariff, and A. Prabhu, “Fermi gf100 gpu architecture.,” *IEEE Micro*, vol. 31, no. 2, pp. 50–59, 2011.

[71] Z. Yang, Y. Zhu, and Y. Pu, "Parallel image processing based on cuda," in *Computer Science and Software Engineering, 2008 International Conference on*, vol. 3, pp. 198–201, 2008.

[72] A. Colic, H. Kalva, and B. Furht, "Exploring nvidia-cuda for video coding," in *Proceedings of the first annual ACM SIGMM conference on Multimedia systems*, MMSys '10, (New York, NY, USA), pp. 13–22, ACM, 2010.

[73] B. Chapman, G. Jost, and R. Van Der Pass, *Using OpenMP: Portable Shared Memory Parallel Programming (Scientific Computation and Engineering Series).* The MIT Press, 2008.

[74] J. Liu, W. Ma, F. Liu, Y. Hu, J. Yang, and X. Xu, "Study and application of medical image visualization technology," in *Digital Human Modeling*, pp. 668–677, Springer, 2007.

[75] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *ACM Siggraph Computer Graphics*, vol. 21, ACM, 1987.

[76] R. Shen and P. Boulanger, "Hardware-accelerated volume rendering for real-time medical data visualization," in *Advances in Visual Computing*, pp. 801–810, Springer, 2007.

[77] I. Viola, A. Kanitsar, and M. E. Gröller, "Gpu-based frequency domain volume rendering," in *Proceedings of the 20th spring conference on Computer graphics*, pp. 55–64, ACM, 2004.

[78] K. V. Asari, "Design of an efficient VLSI architecture for non-linear spatial warping of wide-angle camera images," *Journal of Systems Architecture*, vol. 50, no. 12, pp. 743–755, 2004.

[79] H. T. Ngo and V. K. Asari, "A pipelined architecture for real-time correction of barrel distortion in wide-angle camera images," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 15, no. 3, pp. 436–444, 2005.

[80] P.-Y. Chen, C.-C. Huang, Y.-H. Shiau, and Y.-T. Chen, "A VLSI implementation of barrel distortion correction for wide-angle camera images," *Trans. Cir. Sys.*, vol. 56, pp. 51–55, January 2009.

[81] S.-L. Chen, H.-Y. Huang, and C.-H. Luo, “Time Multiplexed VLSI Architecture for Real-Time Barrel Distortion Correction in Video-Endoscopic Images,” *IEEE Trans. Circuits Syst. Video Techn.*, vol. 21, no. 11, pp. 1612–1621, 2011.

[82] K. Daloukas, C. Antonopoulos, N. Bellas, and S. Chai, “Fisheye lens distortion correction on multicore and hardware accelerator platforms,” in *Parallel Distributed Processing (IPDPS), 2010 IEEE International Symposium on*, pp. 1 –10, april 2010.

[83] R. Hartley and S. B. Kang, “Parameter-free radial distortion correction with centre of distortion estimation,” in *ICCV*, pp. 1834–1841, 2005.

[84] M.-P. J., C.-P. J., and R. C., “Natural orifice transluminal endoscopy surgery: A review,” *World J Gastroenterol*, vol. 17, no. 33, pp. 3795–3801, 2011.

[85] L. E., R. C., P. J.M., H.-C. T., S. D., and C. J.L., “Third-generation cholecystectomy by natural orifices: transgastric and transvesical combined approach (with video),” *Gastrointest Endoscopy*, vol. 65, no. 1, pp. 111–117, 2006.

[86] R. C., L. E, P. J.M., H.-C. T., S. D., and M. I., “Third-generation cholecystectomy by natural orifices: transgastric and transvesical combined approach (with video),” *Gastrointest Endoscopy*, vol. 65, no. 1, pp. 111–117, 2007.

[87] F. K.H., “Minimally invasive surgery,” *Endoscopy*, vol. 34, no. 2, 2002.

[88] H. A.G. and H. T., “Am J Surg,” *Minimally invasive abdominal surgery: lux et veritas past, present, and future*, vol. 190, pp. 239–43, 2005.

[89] T. T., F. A., K. S., C. L., S. D., C. J., and R. P., “Minimally invasive abdominal surgery: lux et veritas past, present, and future,” *J Urol*, vol. 191, no. 5, pp. 1225–30, 2014.

[90] D. S., Z. P.L., R. S.T., G. Z.L., and S. D.M., “Minimally invasive spine surgery for adult degenerative lumbar scoliosis,” *Neurosurg Focus*, vol. 36, no. 5, 2014.

[91] N. Vakil, W. Smith, K. Bourgeois, E. Everbach, and K. Knyrim, “Endoscopic measurement of lesion size: improved accuracy with image processing,” *Gastrointest Endosc.*, vol. 40, pp. 178–83, 1994.

[92] L. Way, L. Stewart, W. Gantert, K. Liu, C. Lee, K. Whang, and J. Hunter, “Causes and prevention of laparoscopic bile duct injuries: analysis of 252 cases

from a human factors and cognitive psychology perspective," *Ann Surg.*, vol. 9, 2003.

[93] R. Shin, "Evaluation of the learning curve for laparoscopic roux-en-y gastric bypass surgery," *SurgObesRelat Dis.*, vol. 4, 2005.

[94] R. Malik, P. White, and C. Macewen, "Using human reliability analysis to detect surgical error in endoscopic DCR surgery," *Clinical Otolaryngology & Allied Sciences*, vol. 28, no. 5, pp. 456–460, 2003.

[95] R. Suleman, T. Yang, J. Paige, S. Chauvin, J. Alleyn, M. Brewer, S. Johnson, and R. Hoxsey, "Hand-eye dominance and depth perception effects in performance on a basic laparoscopic skills set," *JSLS*, vol. 14, 2010.

[96] L. Pockett, M. Salmimaa, M. Pölönen, and J. Häkkinen, "The impact of barrel distortion on perception of stereoscopic scenes," *SID Symposium Digest of Technical Papers*, vol. 41, no. 1, pp. 526–529, 2010.

[97] M. Gschwandtner, M. Liedlgruber, A. Uhl, and A. Vécsei, "Experimental study on the impact of endoscope distortion correction on computer-assisted celiac disease diagnosis," in *Information Technology and Applications in Biomedicine (ITAB), 2010 10th IEEE International Conference on*, pp. 1–6, IEEE, 2010.

[98] A. Insel, B. Carofino, R. Leger, R. Arciero, and A. Mazzocca, "The development of an objective model to assess arthroscopic performance," *J Bone Joint Surg Am*, vol. 91, no. 9, p. 2287–2295, 2009.

[99] W. Barlow, "Modeling of categorical agreement," *The encyclopedia of biostatistics*, pp. 541–545, 1998.

[100] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.

[101] C. Wengert, P. C. Cattin, J. M. Duff, C. Baur, and G. Székely, "Markerless endoscopic registration and referencing," in *MICCAI*, 2006.

[102] O. Grasa, J. Civera, A. Güemes, V. Munoz, and J. Montiel, "EKF Monocular SLAM 3D Modeling, Measuring and Augmented Reality from Endoscope Image Sequences," in *MICCAI*, 2009.

[103] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Real time localization and 3D reconstruction," in *CVPR*, pp. 363–370, 2006.

Appendices

Appendix A

Unsupervised "plumb-line" calibration results

This appendix displays the calibration results of the proposed unsupervised intrinsic camera calibration using a “Plumb-line” approach. The first column shows the segments highlighted in colors, the middle column shows the detected lines consistent with the calibration and the last column shows the radial distortion correction using the estimated parameters.

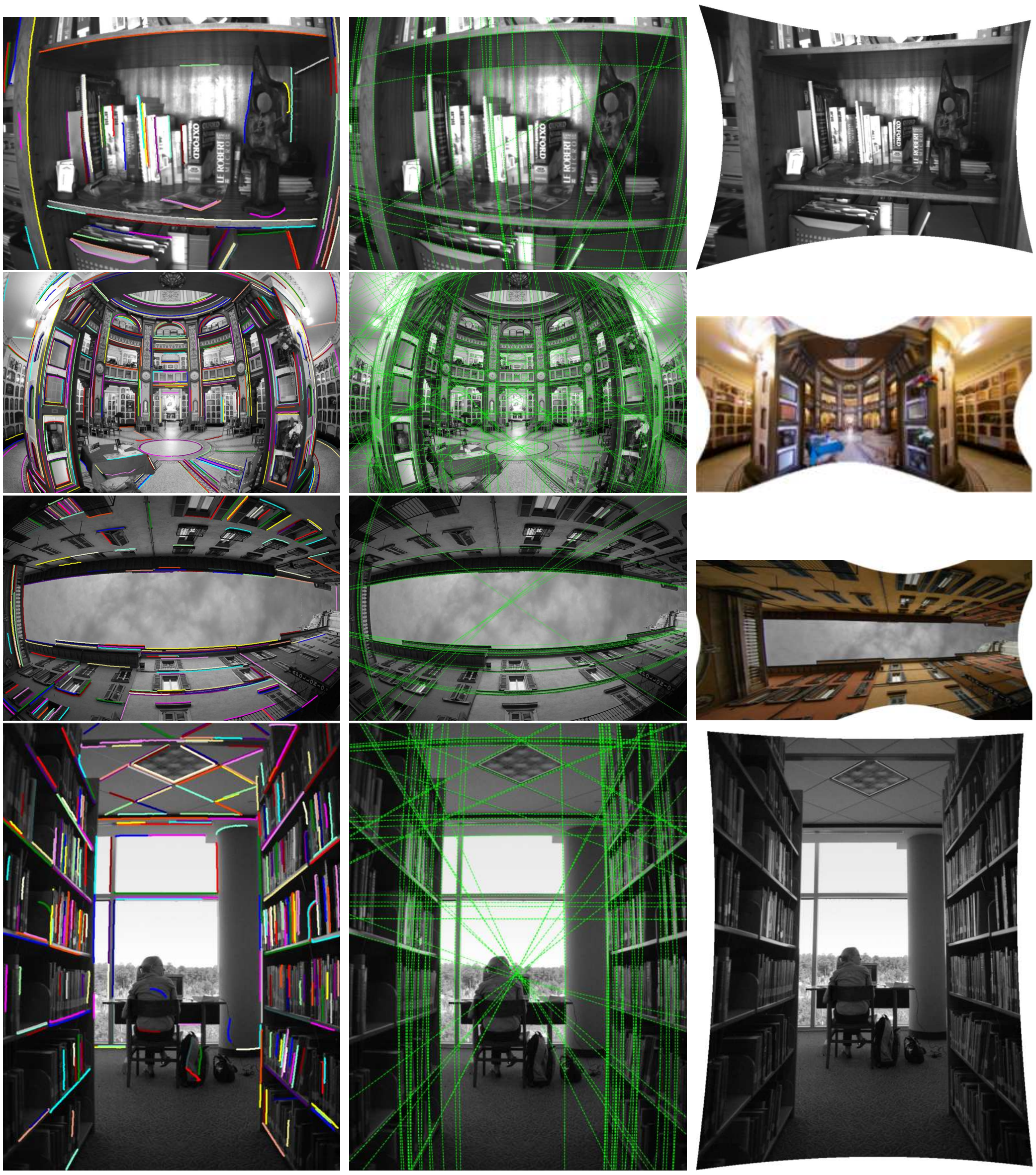


Figure A.1: Results of the unsupervised plumb-line calibration. Data set # 1

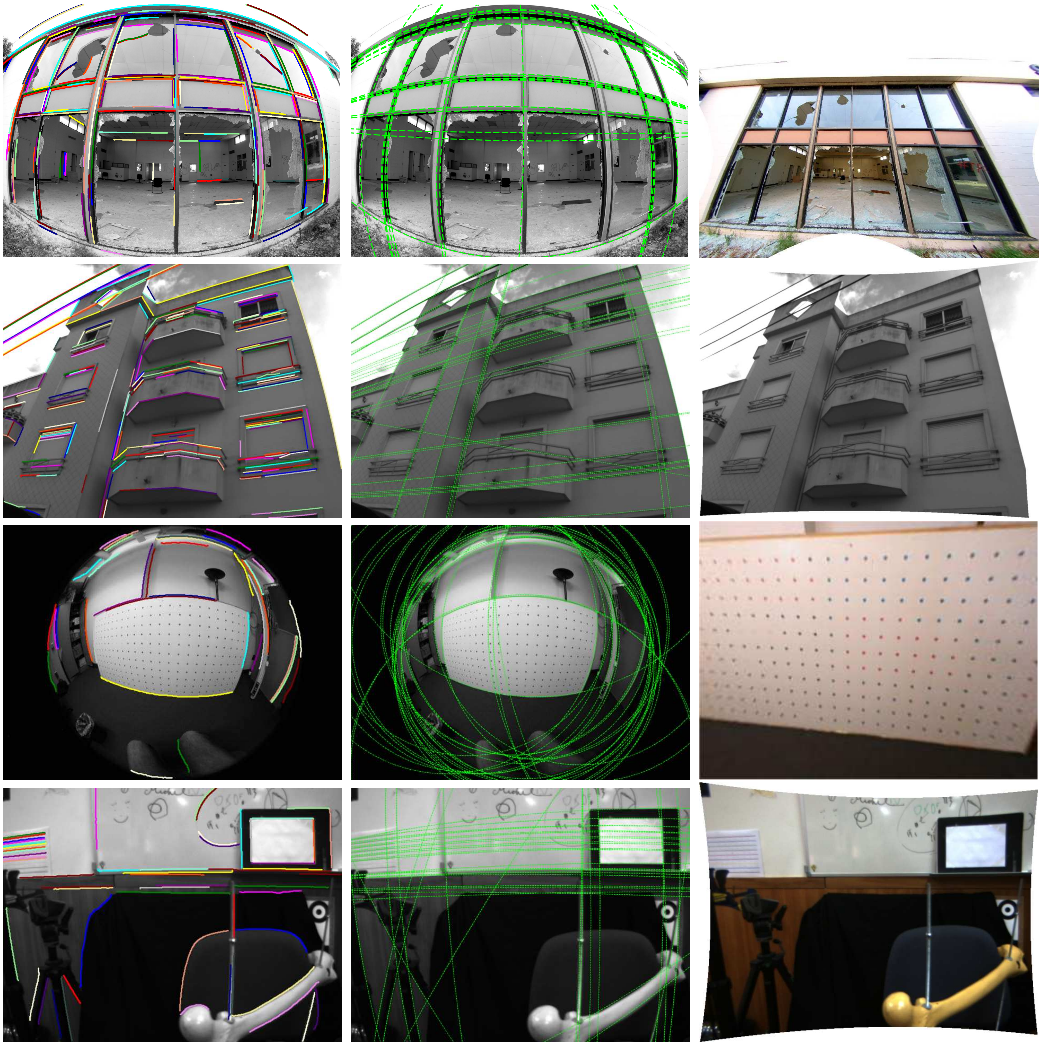


Figure A.2: Results of the unsupervised plumb-line calibration. Data set # 2

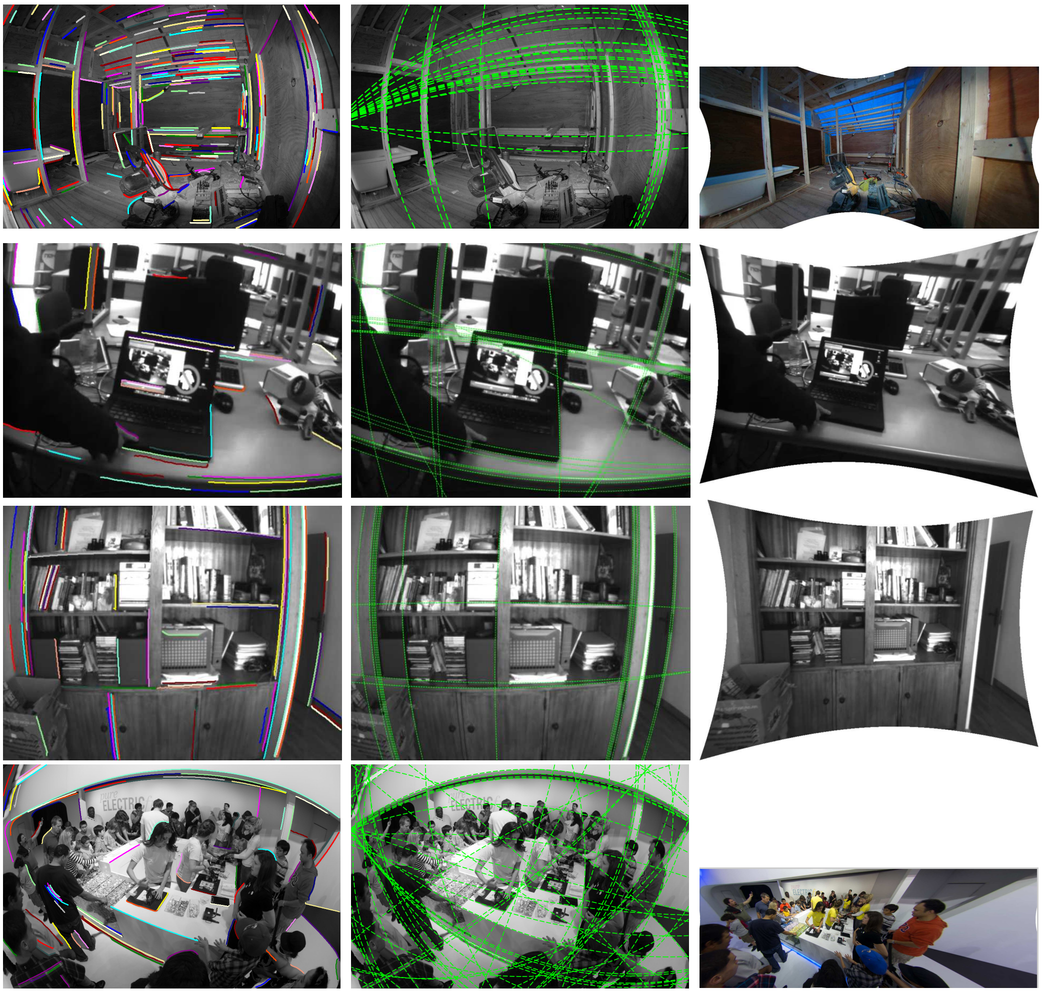


Figure A.3: Results of the unsupervised plumb-line calibration. Data set # 3

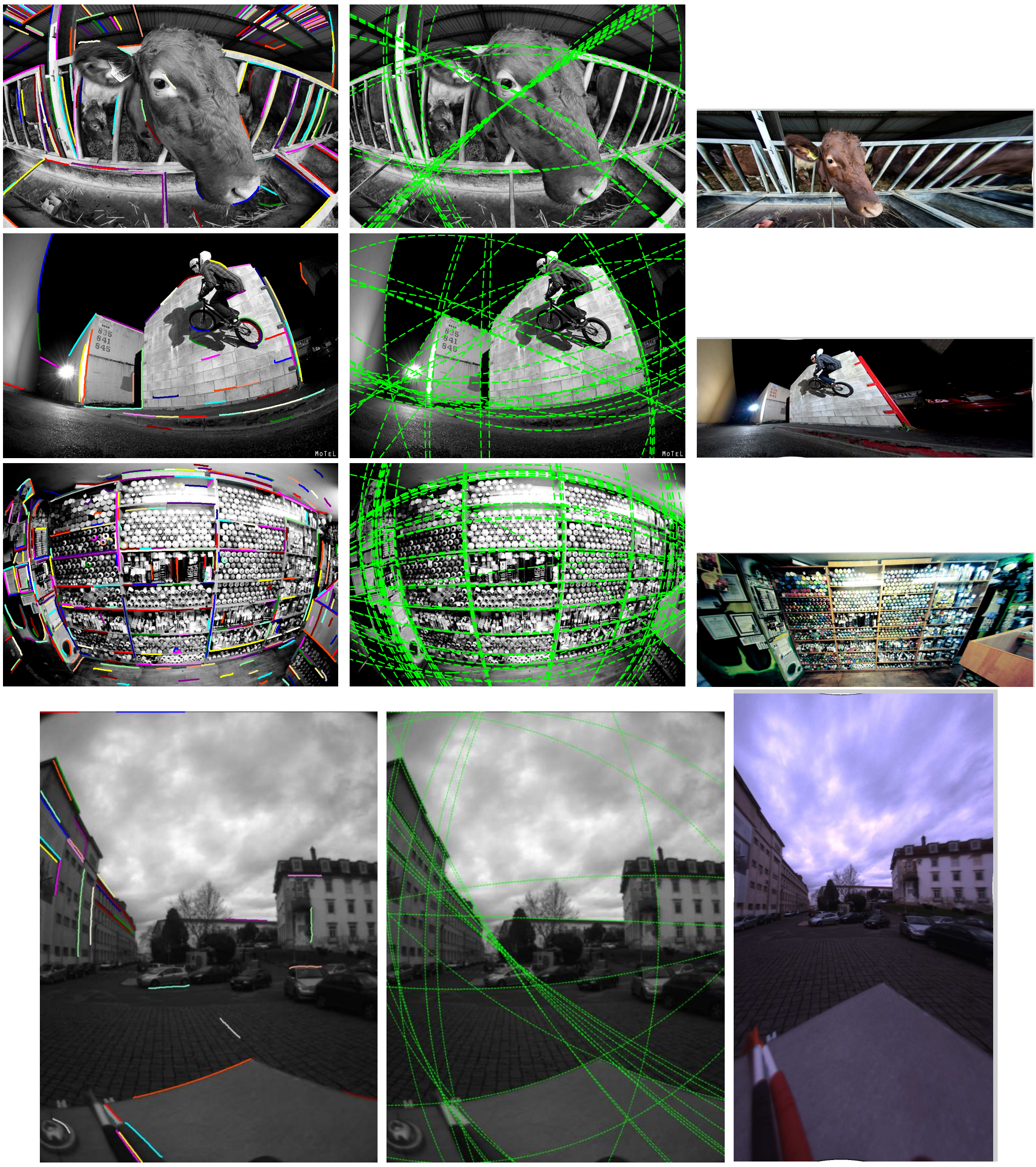


Figure A.4: Results of the unsupervised plumb-line calibration. Data set # 4

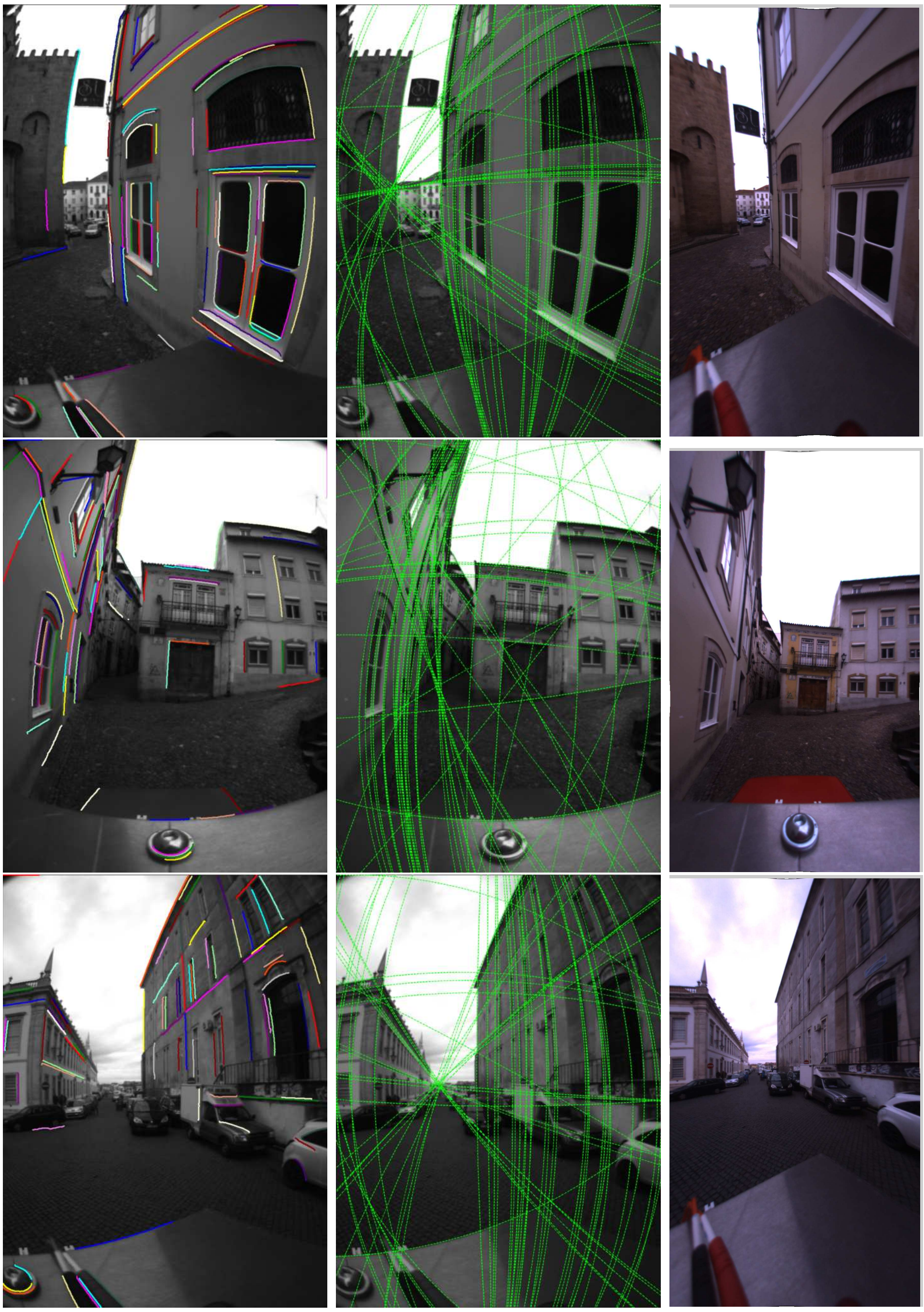


Figure A.5: Results of the unsupervised plumb-line calibration. Data set # 5

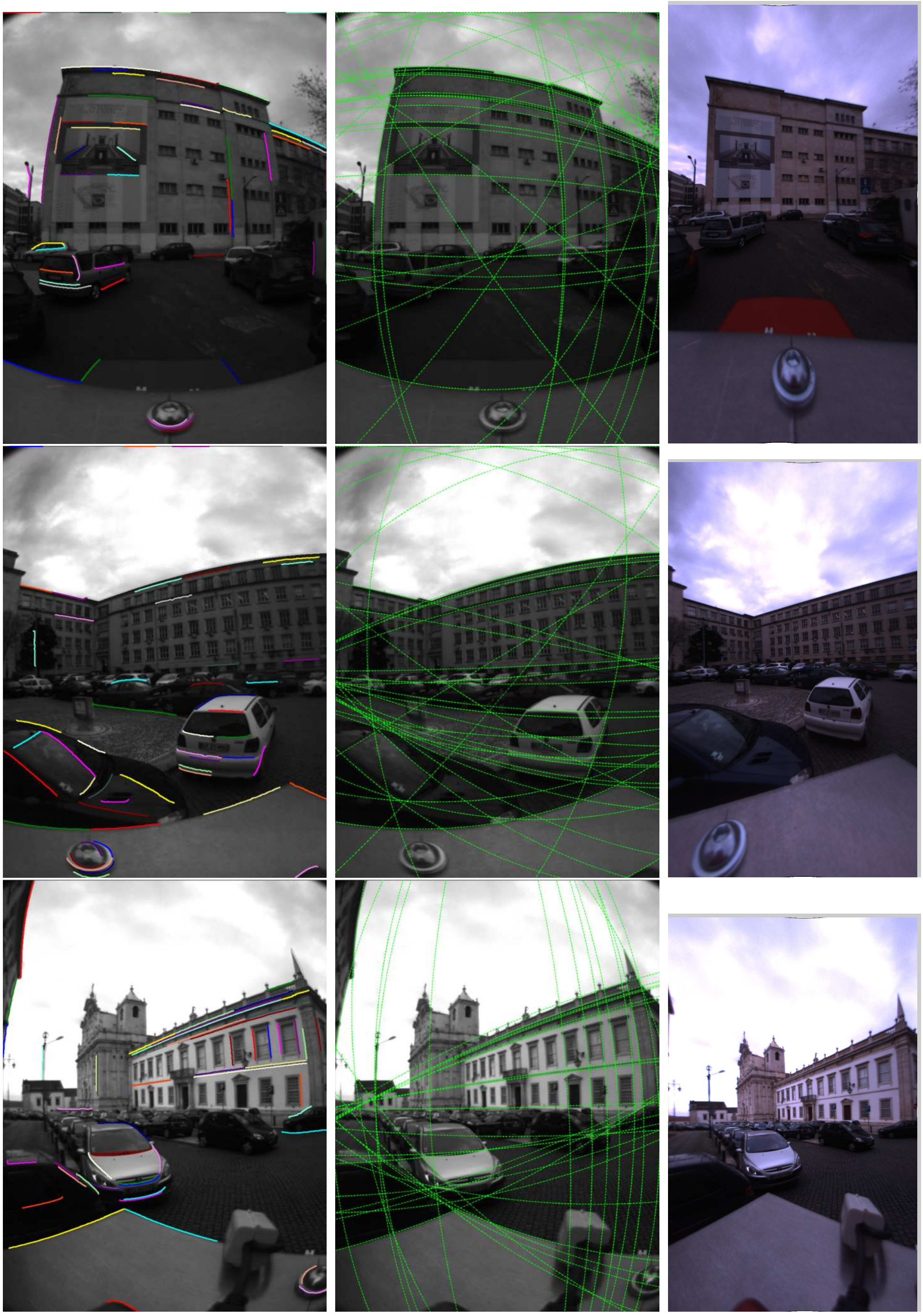


Figure A.6: Results of the unsupervised plumb-line calibration. Data set # 6

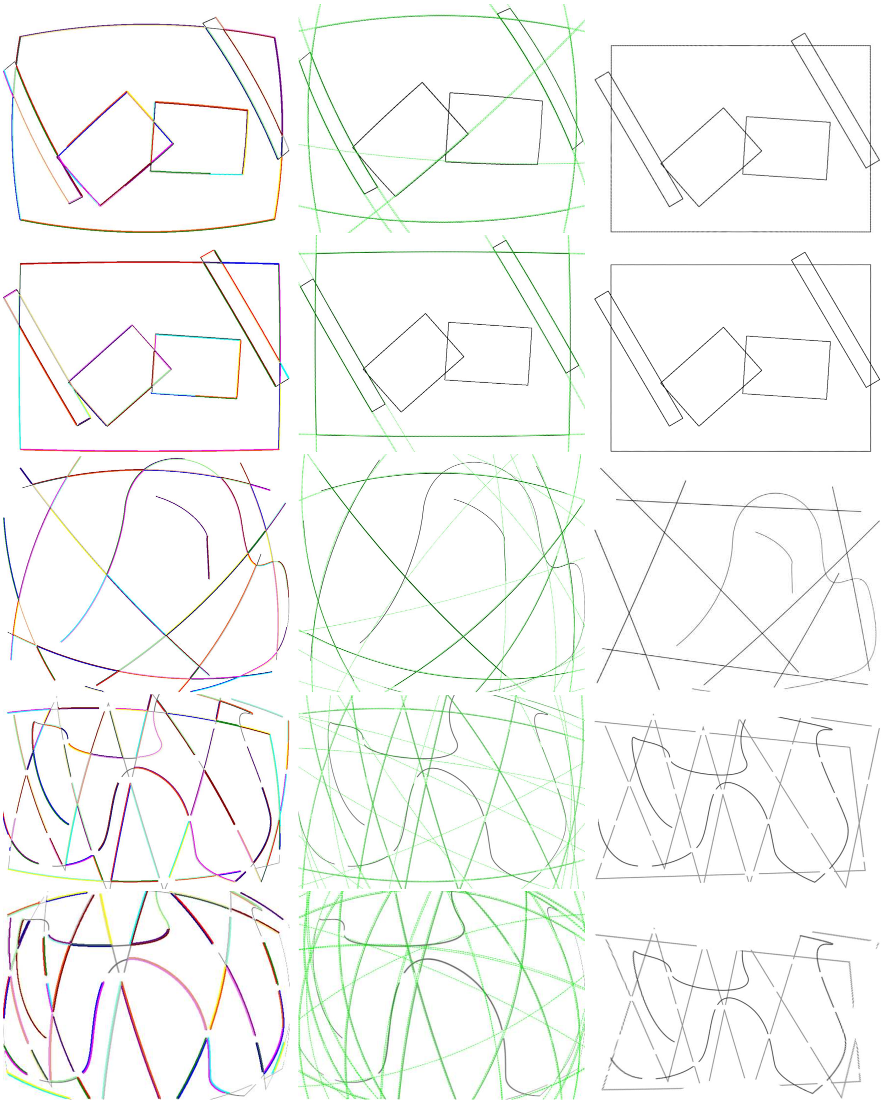


Figure A.7: Results of the unsupervised plumb-line calibration. Data set of synthetic images

Appendix B

RD correction in live surgery

This appendix presents some images corrected with the RD correction system depicted throughout this thesis. The video was acquired by a ConMed® at 1080p60Hz endoscope during a knee ligamentoplasty. This procedure is used to repair the anterior cruciate ligament (ACL) and/or the posterior cruciate ligament (PCL) and is generally needed in young, active patients who have had an accident usually while doing sport. The procedure is performed in a saline solution that is constantly flowing inside the knee joint.

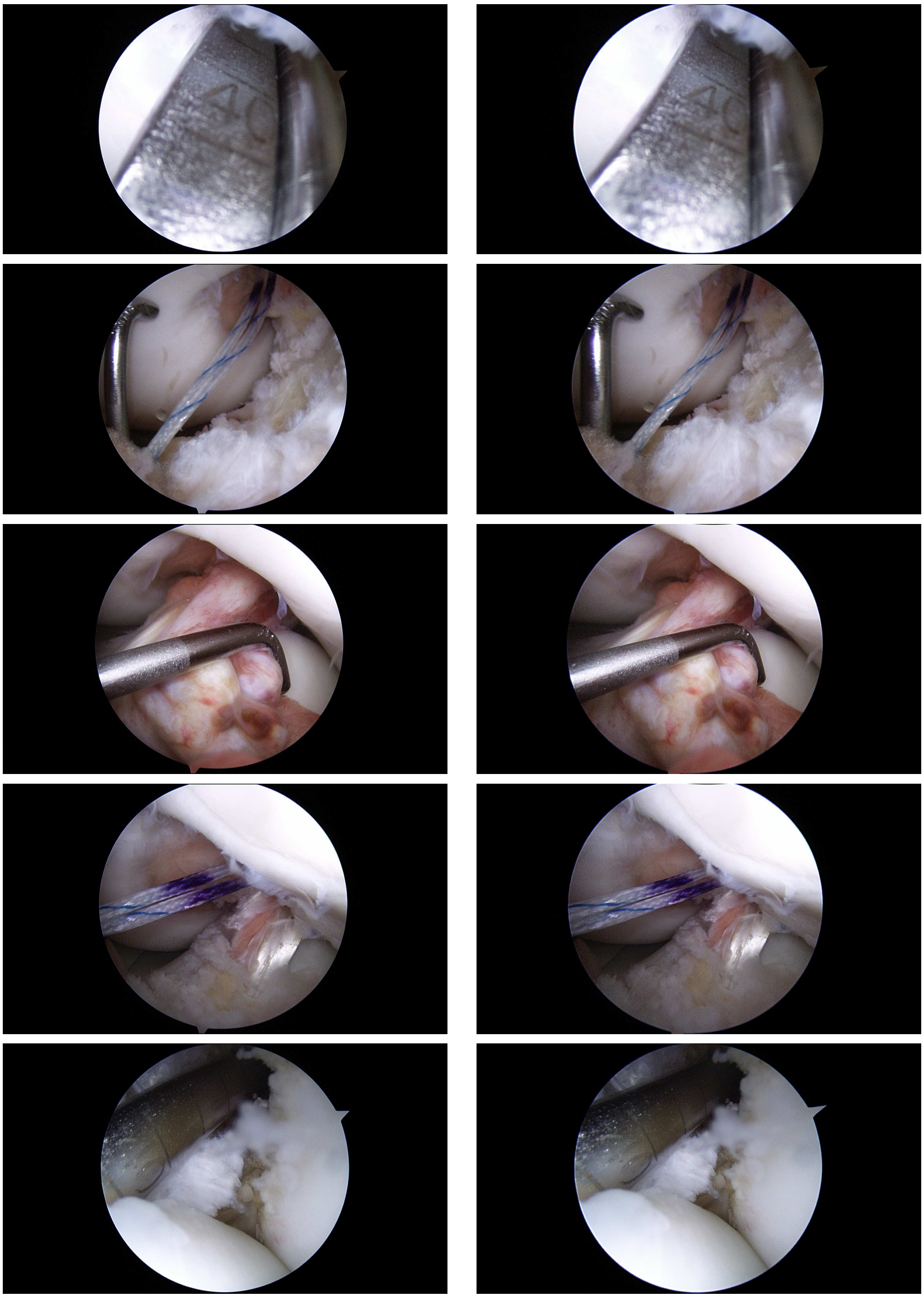


Figure B.1: Result set #1 of the RD correction in a knee ligamentoplasty. The left column shows the original image and the right column the image after distortion correction.

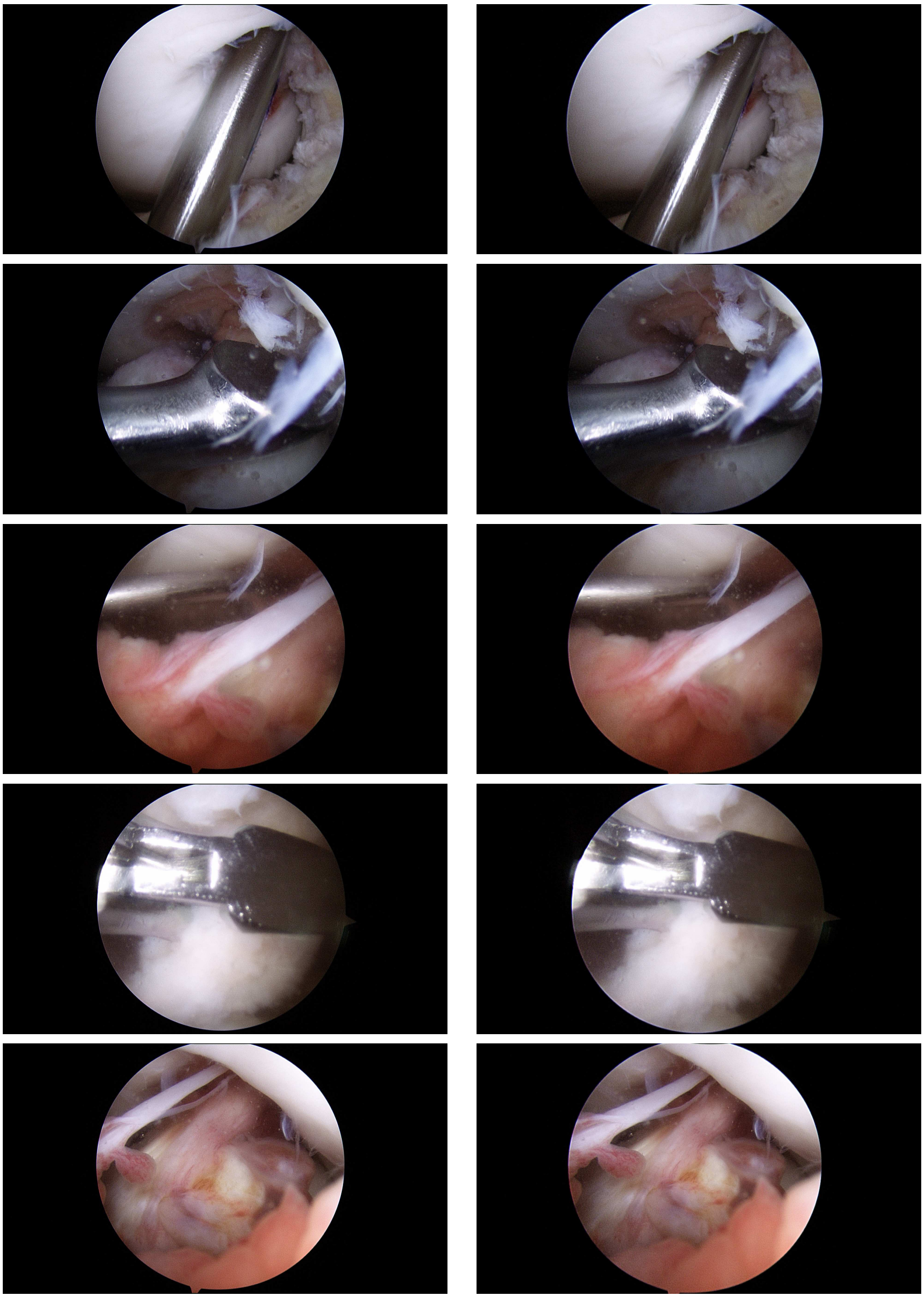


Figure B.2: Result set #2 of the RD correction in a knee ligamentoplasty. The left column shows the original image and the right column the image after distortion correction.

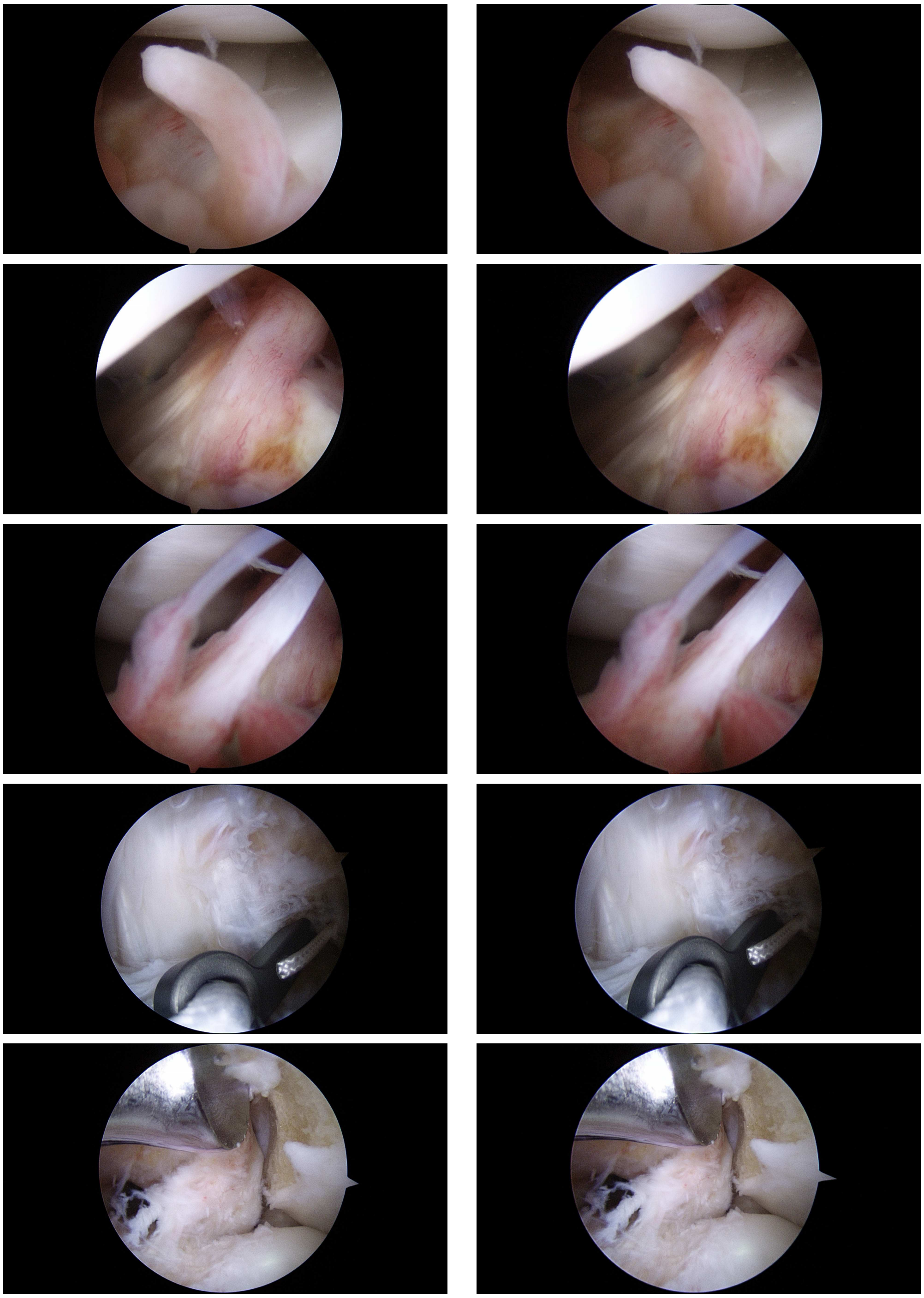


Figure B.3: Result set #3 of the RD correction in a knee ligamentoplasty. The left column shows the original image and the right column the image after distortion correction.

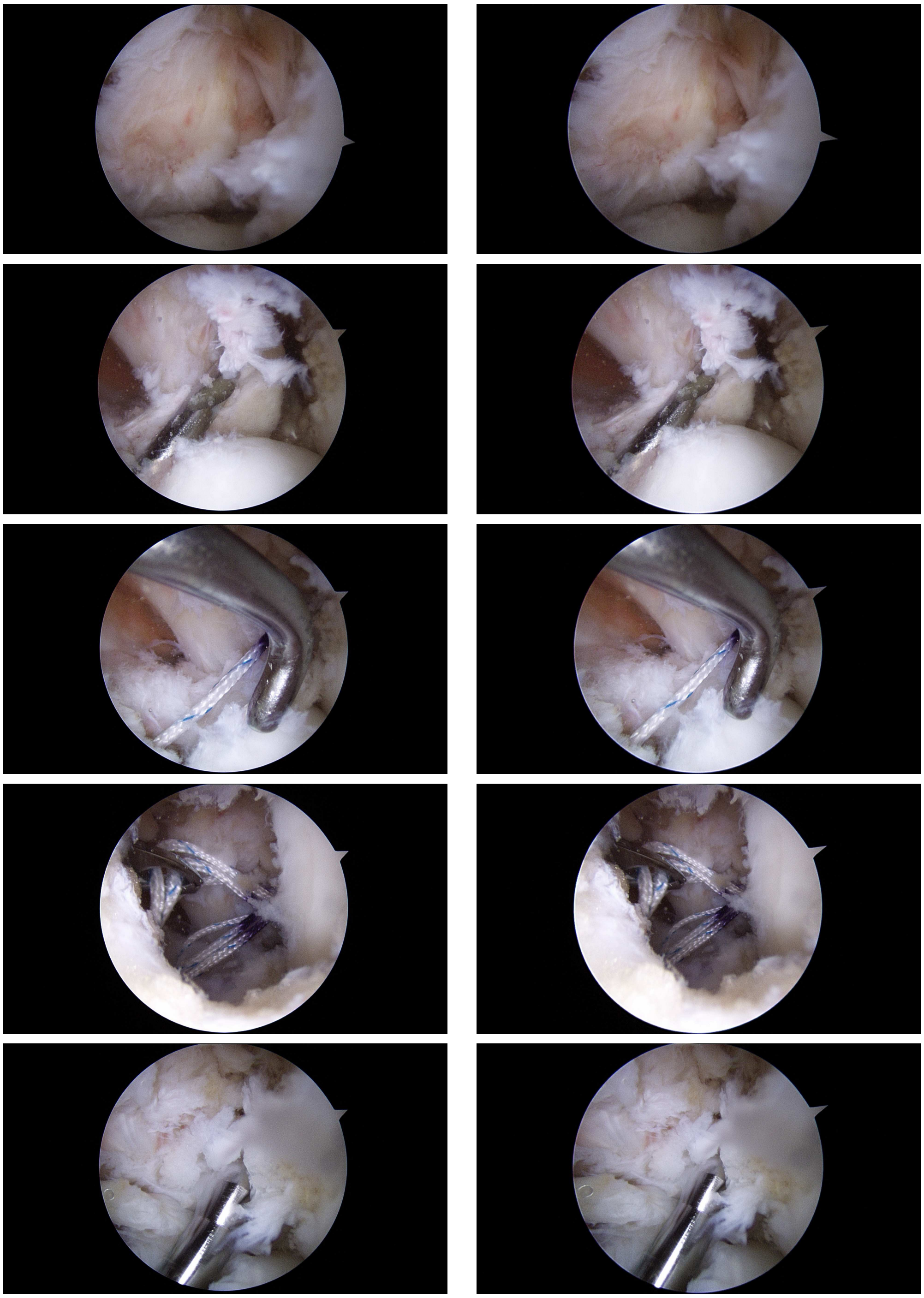


Figure B.4: Result set #4 of the RD correction in a knee ligamentoplasty. The left column shows the original image and the right column the image after distortion correction.