TU Vision Hamburg-Harburg

Technische Universität Hamburg-Harburg
Vision Systems

Prof. Dr.-Ing. R.-R. Grigat

Patch Based 3D Reconstruction of the Liver Surface from Laparoscopic Videos

Master Thesis

Pedro Santos

September 30, 2013

universidade de aveiro theoria poiesis praxis

TUHH
Technische Universität Hamburg-Harburg

Declaration

I hereby assert that this diploma thesis has been compiled by me with only the help of the auxiliary material listed in the text or in the bibliography.

Hamburg, September 30, 2013

Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit von mir selbständig und nur unter Verwendung der aufgeführten Hilfsmittel erstellt wurde.

Hamburg, den 30. September 2013

Abstract

Endoscope is commonly used in Laparoscopic surgery. It allows to the doctor to perform the surgery without opening a cavity. Therefore, it allows the surgeon to perform smaller incisions and consequently diminish the risk of infection. An investigation regarding the reconstruction of the liver surface from a sequence of images obtained from an endoscope video has been done by the Vision Institute of TUHH, in cooperation with the Medical Center Hamburg-Eppendorf. The goal of this paper is to optimize that reconstruction. Previous works in the institute allowed, using matlab functions, to obtain a first reconstruction. Having the matrices of the transformations between cameras and the reconstructed points in 3D, the goal is to optimize that using a patch based 3D reconstruction. The method consists of the construction of a patch for every point and trying to find a good normal for the patch. Some tools are used for the optimization like normal cross correlation that is the similarity function used for obtaining the match of the corresponded 2D windows in different cameras, and a nonlinear optimization method to look for the best normal who gives the best match between the windows. The main objective is to evaluate this method and draw conclusions about its utility and viability: can this method be used for the optimization in the case of the endoscope video of a liver surface.

Acknowledgements

I would like to express my deepest appreciation to all those who provided me the possibility to complete this master thesis. A special thanks to my tutor MSc. Jan Marek Marcinczak for the support during this last 6 months. I would also like to express my gratitude to Prof. Dr.-Ing. R.-R.Grigat for the given support. Furthermore, I would like to thank my parents, my sister and all my family and friends for all the support, not only this last year but through all my life. I will always be grateful for your love. Finally, I would like to thank all the professors in the University of Aveiro for all the required experience that allowed me to finish my studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Foreword

This thesis is included in a continuous research of the PhD of MSc. Jan Marek Marcinczak at Vision Systems Institute of the Hamburg University of Technology. This paper refers continuously to the paper of Yasutaka Furukawa and Jean Ponce [1] and to the book Multi View Geometry by Hartley and Zisserman [2]. A CD is attached with all matlab m.files of the method and all m.files used during the research.

1.2 Motivation

This project is also a part of a study that is being developed in partnership with the University Medical Center Hamburg-Eppendorf. This medical center treats 400 patients a year, through laparoscopic surgery. During each procedure the abdomen, with the goal of having work volume, is insufflated with CO_2. Trough much smaller incisions than those needed for classic procedures, a system of cannulae and trocars are inserted. These trocars seal and fix the abdomen, while the cannulae allow the access of the insufflators and operation devises. This procedure is used for some medical conditions, but in this project in particular the data is refers to liver cirrhosis. It is a very important disease that basically consists of scarring of the liver and poor liver function. It is the final phase of chronic liver disease. It's etiology is various, between bile duct disorders, nonalcoholic fatty liver disease (NAFLD) and nonalcoholic steatohepatitis (NASH), the causes than can lead to a cirrhosis are countless. However, the most common are hepatitis (B or C) and alcohol abuse. Considering this large etiology, the types of cirrhosis also vary accordingly. Pathogenically, what happens is that this disease changes the structure of the liver and the blood vessels that nourish it. Therefore, it reduces the liver's ability to manufacture proteins and process hormones, nutrients, medications, and poisons. As a chronicle disease, cirrhosis gets worse over time, and has serious complications, such as

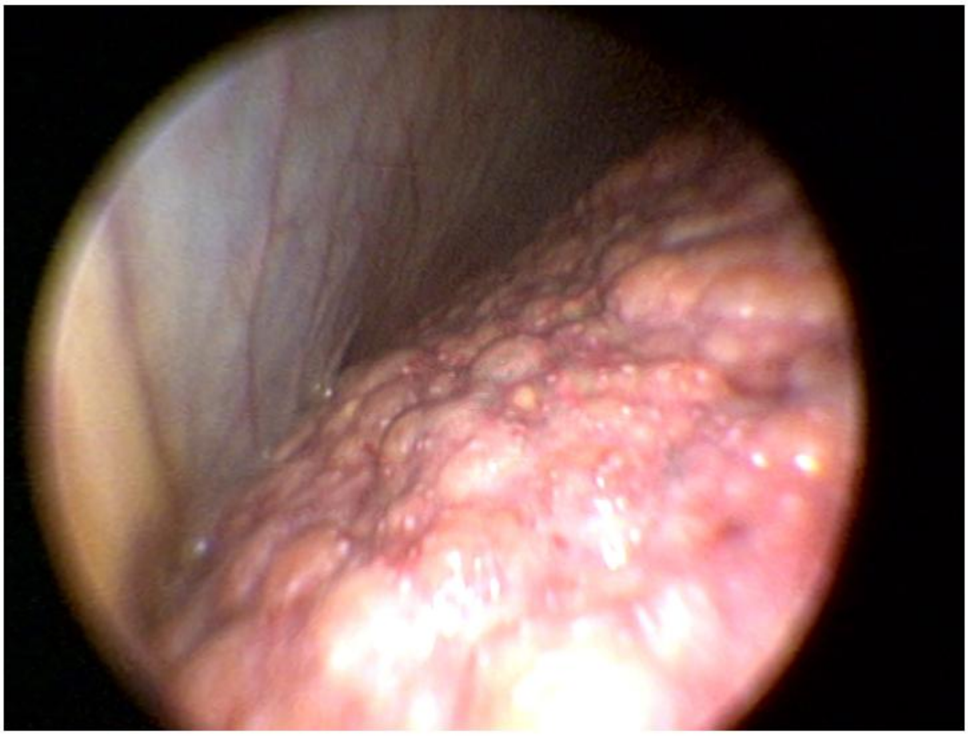


Figure 1.1: Cirrhosis liver obtain by a laparoscopic video in University Medical Center Hamburg-Eppendorf

portal hypertension (and consequently variceal bleeding), kidney failure, increased risk of infections, hepatic encephalopathy among many others. Even though some patients are able to embrace a relatively normal life with this kind of condition, most of them suffer a lot of complications and need the only long term solution that Medicine can provide nowadays: a liver transplant. Considering all of this information, the most important message that needs to be spread is to prevent this kind of pathology: to avoid certain risk conducts (such as unprotected intercourse) and limit the consumption of alcohol. The main goal of this project is to help the doctors to understand the evolution of this disease. With a good 3D reconstruction will be possible to measure if the nodules are growing or not.

1.3 Overview

This thesis is organized in 8 main chapters. The first is a brief introduction to the project containing the foreword a brief motivation and an overview of the thesis. In the second chapter the theoretical background is presented. Therefore, there are briefly explained the followed points , camera models , transformations in 3D , normalized cross correlation and unconstrained optimization. The third chapter contains the state of the art worldwide; in this chapter, other methods of reconstruction will be presented. The problem statement and objectives are presented in chapter four. In chapter five, the method is explained. The initialization of the patch, discrepancy function, optimization of the normal, and the expansion are the topics of this chapter. The sixth chapter introduces the implementation; therefore, it is possible to understand how the method was implemented in Matlab. In the seventh chapter the results and evaluations are presented. Results of the evaluations that had to be made are presented and explained. In chapter 8 the conclusion of the work is presented. Finally in the last chapter some ideas for improving are referred.

Chapter 2

Theoretical Background

2.1 Introduction

In the present chapter, I approach the theoretical background needed for the understanding of the present work, such as the camera models, focusing on the one used in this work, pinhole camera model. However the ideas are presented in this chapter in a resumed version. If the reader wants to learn more the used of the follow sources can be useful [2] [3] .Transformations in 3D is presented with a table and a small explanation. For the last 2 useful tools used in this work are explained, normalized cross correlation and unconstrained optimization, both tools were used as matlab functions that already exist in the Matlab library.

2.2 Laparoscopic Evolution

The great pioneer of laparoscopic was Georg Kelling of Dresden Germany, in fact he performed the first laparoscopic procedure in 1902 on a dog. The first reported la- paroscopic operation performed in humans was in 1910 by Hans Jacobaeus of Sweden, [4]. The computer chip television camera boosted the field. This technology allowed the visualization in a magnified view. Before this, Laparoscopic was a surgical approach with few applications. Laparoscopic had a huge increase during the war in fact comparing to the open cavity surgery this technique allowed smaller incisions , reducing the external contaminants , faster post-surgery recovery and less cost and time for the hospital. In 1924 the swiss Richard Zollikofer [5] recommend the use of CO2 to insufflate the abdomen. This practice allowed an enormous evolution in this field. Then, in the 80s, the invention of CCD (charged couple device image sensor) put the laparoscopic as a gold standard in general surgery.

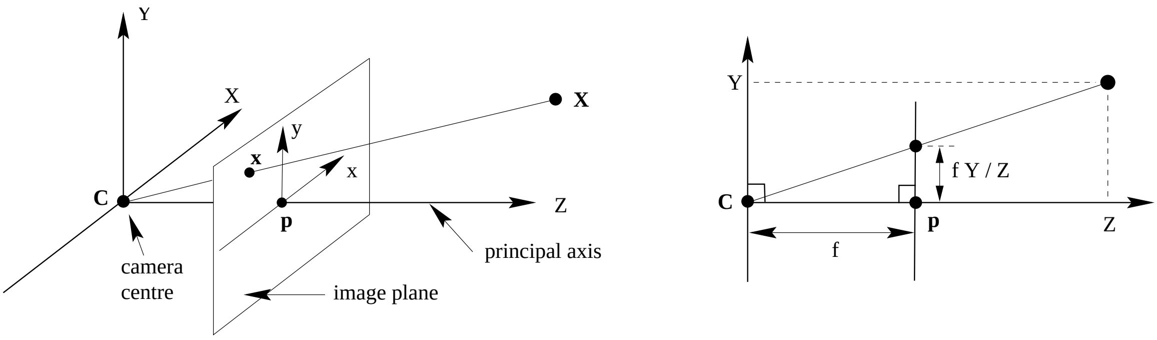


Figure 2.1: pinhole model - example of a projection.Figure taken from [Hartley 04]

2.3 Cameras Models

A camera is the one who maps 3D world points to 2D image. There are quite a few camera models that are basically the matrices that permit the mapping. The models are divided in 2 major classes the ones that the center of projection is finite and the others with the center in infinity. One of the simplest models is the pinhole model that is used in this work. The pinhole model is an example of a finite camera.

2.3.1 Pinhole Camera Model

In figure 2.1 is possible to see a projection of 3D points onto a 2D image points. Considering center of projection to be the center of a Euclidean coordinates system the points in real world are projected in the direction of C (center of the camera) and this projection intercepts the plane in Z or f (focal distance) where the plane (focal plane) is located. So as equation (2.1) shows the 3D point $(X,Y,Z)^T$ is mapped to image point $\left(\frac{fX}{Z},\frac{fY}{Z}\right)^T$.

$$(X,Y,Z)^T \longmapsto \left(\frac{fX}{Z},\frac{fY}{Z}\right)^T \tag{2.1}$$

This transform of equation (2.1) maps points in Euclidean $\mathbb{R}^3$ to Euclidean $\mathbb{R}^2$. The axis Z that is the line perpendicular to the image and passing through camera center is called principal axis or ray of the camera. p that is visible in figure (2.1) that lays in the center of the Euclidean space $\mathbb{R}^2$ is called principal point. Is quite simple to represent the transformation between 3D and 2D as is possible to see in equation (2.2).

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \longmapsto \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.2}$$

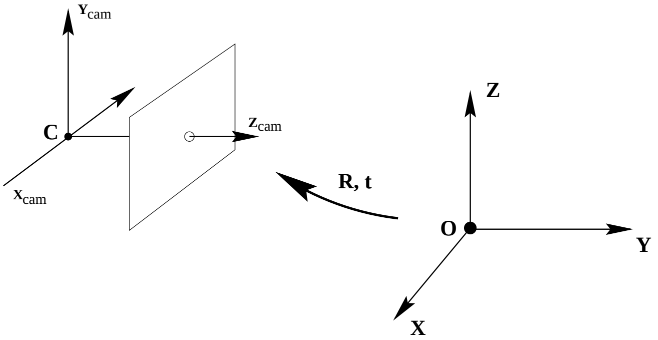


Figure 2.2: Rotation transform.Figure taken from [hartley2000multiple]

We can simplify and say that $x = PX$ being P the camera projection matrix. In equation (2.2) we assume that the origin of the image coordinates lays in the image center which is not always true so another factor as to be considered as is possible to observe in equation (2.3).

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \longmapsto \begin{bmatrix} fX + Zp_x \\ fY + +Zp_y \\ Z \end{bmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.3}$$

Being K the camera calibration matrix.

$$K = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \tag{2.4}$$

$$x = K[I|O]x_{cam} \tag{2.5}$$

x_{cam} is the coordinates in real world but transform to lay in the ray of the camera as is presented in Fig.2.2.

2.4 Transform of 3D

Considering point X in 3D space is possible to represent X in homogeneous coordinates as X=(X_1, X_2, X_3, X_4) with $X_4 \neq 0$, if $X_4 = 0$ the point lays in infinity. The homogeneous coordinates above referred represent the 3D point (X,Y,Z) so X=$\frac{X_1}{X_4}$, Y=$\frac{X_2}{X_4}$, Z=$\frac{X_3}{X_4}$.

$$X' = HX \tag{2.6}$$

Equation (2.6) represents a projective transformation in P^3 the matrix H is the transformation of X to X' and has 16 dof.

Group	Matrix	Additional Invariants of P^3
Projective 15dof	$\begin{bmatrix} A & t \\ v^T & \upsilon \end{bmatrix}$	
Affine 12dof	$\begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix}$	Parallelism of planes, Plane at infinity
Similarity 7dof	$\begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix}$	Relative volume
Euclidean 6dof	$\begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$	Volume

Table 2.1: Transformation of P^3

In this work the only used transformation is the Euclidean that has 6 dof 3 in the rotation matrix and 3 in translation matrix. The transformations between cameras are already obtained in previous works in Vision Systems department of TUHH.

2.5 Normalized Cross Correlation

Template matching that is a technique for finding small parts of an image witch match a template, can be used for feature detection. Assuming one window of points in the reference image as the template and then find that template in other images is possible to detect features and is that principle that will be used in this work as it would be explain in chapter 5. For doing the feature detection normalized cross correlation is very used. The used of this function is based on the distance measure.

$$d^2_{\text{f,t}}(u,v) = \sum_{x,y} [f(x,y) - t(x-u,y-v)]^2 \tag{2.7}$$

Being f the image and t the feature positioned at u,v.

$$d^2_{\text{f,t}}(u,v) = \sum_{x,y} \left[f^2(x,y) - 2f(x,y)t(x-u,y-v) + t^2(x-u,y-v))\right] \tag{2.8}$$

As $t^2(x-u,y-v)$ is constant:

$$C(u,v) = \sum_{u,v} [f(x,y)t(x-u,y-v)] \tag{2.9}$$

Being equation (2.9) the measure of similarity. To overcome some difficulties , like variations of energy $\sum f^2(x,y)$ with position normalization is used so:

$$\gamma(u,v) = \frac{\sum_{x,y}\left[f(x,y) - \overline{f_{\mathrm{u,v}}}\right]\left[t(x-u,y-v) - \bar{t}\right]}{\sqrt{\sum_{x,y}\left[f(x,y) - \overline{f_{\mathrm{u,v}}}\right]^2 \sum_{x,y}\left[t(x-u,y-v) - \bar{t}\right]^2}} \tag{2.10}$$

Being $\gamma(u,v)$ the normalized cross correlation $\bar{t}$ the mean of the feature and $\overline{f_{\mathrm{u,v}}}$ the mean of f(x,y).

2.6 Unconstrained Optimization

2.6.1 Introduction

In this section a few unconstrained algorithms will be presented. For many engineering applications the use of unconstrained optimization is really useful. Multivariable problems are always constrained yet the optimum is sometimes unconstrained. Unconstrained optimization is common used to extremize nonlinear functions of single, multiple variables subjects to no constraints. Considering a scalar real objective function with n variables, f(x):$\mathbb{R}^n \rightarrow \mathbb{R}$.

$$\min_{x\in\mathbb{R}^n} f(x) \tag{2.11}$$

Being equivalent to,

$$\max_{x\in\mathbb{R}^n} -f(x) \tag{2.12}$$

Obtaining the relation of objective function values,

$$-\max_{x\in\mathbb{R}^n} -f(x) = \min_{x\in\mathbb{R}^n} f(x) \tag{2.13}$$

2.6.2 Direct search Methods

The class of optimization methods that does not require derivative information is named direct search methods. The biggest advantage is the easy implementation. There are some strategies of direct search methods like classical direct search methods developed in the 1996s and 1970s, simulated annealing and genetic algorithms. Nelder-Mead Derivate-Free search was proposed by Nelder and Mead [6]. This method starts with a input of a set of ordered points $x^1,...,x^n$. The method is based in the principle of away from the worst. First the best centroid has to be obtained,

$$x^c \triangleq \frac{1}{n}\sum_{i=1}^{n} x^i \tag{2.14}$$

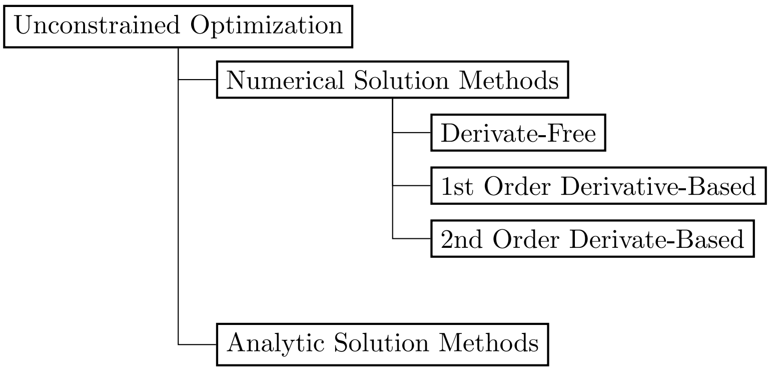

Unconstrained Optimization
Numerical Solution Methods
Derivate-Free
1st Order Derivative-Based
2nd Order Derivate-Based
Analytic Solution Methods

Figure 2.3: Unconstrained optimization methods

Now the direction has to be found for that Δx should move away from the worst,

$$\Delta x \overset{\Delta}{=} x^c - x^{n+1} \tag{2.15}$$

Now is important to defined the step size, for that we need to consider possible solutions, first lets admit a step of 1 so, $x^c + \Delta x$. If this is the new best the step should be increase to 2 , $x^c + 2\Delta x$ and the expansion should be accepted if there were improvements. In case of $x^c + \Delta x$ is not the new best we should attempt contraction so step $\alpha = \frac{1}{2}$. In case of f($x^c + \Delta x$) better than f(x^{n+1}) attempt with $\alpha = -\frac{1}{2}$. Only accept the contraction if $x^c + \pm\frac{1}{2}\Delta x$ if there is improvement. If there were any improvement with expansion and contraction shrink the whole arrays toward current best x^1,

$$x^i_{new} \overset{\Delta}{=} \frac{1}{2}\left[x^1 + x^i\right], \text{for all i=2,...,n+1} \tag{2.16}$$

For the last years the used of more computation power allowed a great improvement of the researches considering the classical direct search approaxes. Dennis and Torczon [7] improved the method of Nelder and Mead [6]. They realize that the method was failing with the increase of variables even in simpler problems.For overcoming they combine the multidimensional pattern search approach with reflection, expansion and contration step. Because this kind of methods are not based on gradients a global minimum is more likely to be found rather then a locally.

2.6.3 1st-order Derivate-based

For unconstrained optimization there are quite a few available derivative based methods. Both 1st and 2nd order methods calculate the direction Δx using derivatives. The steepest descent algorithm starts with a chosen x^0, and a define stopping tolerance $\varepsilon > 0$. Is needed to set $k \leftarrow 0$. Having x^0 compute the gradient $\nabla(x^k)$. If $\nabla(x^k) < \varepsilon$ stop because x^k

is an approximate solution if not continue and calculate the direction, $\Delta x^{k+1} \leftarrow \nabla f(x^k)$, with this proceed to linesearch,

$$\min_{\alpha} l(\alpha) \triangleq f\left(x^k + \alpha \Delta x^{k+1}\right) \tag{2.17}$$

Being α^{k+1} the step. Then update $x^{k+1} \leftarrow x^k + \alpha^{k+1} \Delta x^{k+1}$. Incrementing $k \leftarrow k+1$ compute again the gradient with new k value and repeat all the procedure. There are some pros and cons of this algorithm. This approach is robust and appealingly straightfoward in other hand the approach can converge to a saddle point, the convergence is slow and the approach is sensitive to numerical errors.

2.6.4 2nd- order Derivate-based

In this subsection it will be presented 1 method of second-order, Newton step . This method has the goal of incorporate 2nd order information to enhance the convergence. In Newton-step a quadratic approximation p_2 of f passing through x^k is obtained and then optimize,

$$p_2(x^k + \Delta x) \triangleq f(x^k) + \nabla f(x^k)^T \Delta x + \frac{1}{2} \Delta x^T H(x^k) \Delta x \tag{2.18}$$

For any x^k with $\nabla f(x^k) \neq 0$ (Newton step) Δx^{k+1} is obtain by solving the linear system,

$$H(x^k)\Delta x = -\nabla f(x^k) \tag{2.19}$$

Comparing to 1st order methods this one is less sensitive to numerical errors and better performance close to the optimum. By other side this method is very sinsitive to the starting point x^0 if this point is far from the local optimum the method can fail to converge, and arduous task because the Hessian matrix and the solution of linear system are needed each iteration.

2.6.5 Quasi-Newton

The idea is conjugate 1st order with second order , to guarantee convergence use only 1st partial derivatives like in steepest descent search, then speed-up the converge with some 2nd order information. The question is how to do that. Deflection matrix allows to obtain a modified gradient search direction,

$$\Delta x^{k+1} = -D \nabla f(\nabla x^k) \tag{2.20}$$

Steepest descendant and Newton step are special instances of deflection matrix $D^k \triangleq l$ and $D^k \triangleq H(x^k)^{-1}$.

Initialization	Choose initial x^0 and a stopping tolerance $\varepsilon > 0$
Derivatives	Compute gradient $\nabla f(x^k)$ and Hessien matrix $H(x^k)$
Stopping	if $\left\|\nabla f(x^k)\right\| < \varepsilon$ stop x^k is an approximated solution
Newton step	Calculate x^{k+1} direction by solving the linear system $H(x^k)\Delta x = -\nabla f(x^k)$, to compute the step α^{k+1} solve 1-D line-search problem $\min_\alpha l(\alpha) \overset{\Delta}{=} f\left(x^k + \alpha\Delta x^{k+1}\right)$. update $x^{k+1} \leftarrow x^k + \alpha^{k+1}\Delta x^{k+1}$ and increment $k \leftarrow k+1$ after return to the step of derivatives

Table 2.2: Resume of the Newton step algorithm

Initialization	Choose initial x^0, a initial deflection $D^0 = l$ and a stopping tolerance $\varepsilon > 0$
Stopping	if $\left\|\nabla f(x^k)\right\| < \varepsilon$ stop x^k is an approximated solution
Quasi-Newton step	Obtain move direction $\Delta x^{k+1} \leftarrow -D^k\nabla f(x^k)$ solve 1-D line-search problem $\min_\alpha l(\alpha) \overset{\Delta}{=} f\left(x^k + \alpha\Delta x^{k+1}\right)$, obtain deflection matrix $D^{k+1} \leftarrow D^k + \left(1 + \frac{g^T D^k g}{d^T g}\right)\frac{dd^T}{d^T g} - \frac{D^k g d^T + d g^T d^k}{d^T g}$, with $d = x^{k+1} - x^k$ and $g = \nabla f(x^{k+1}) - \nabla f(x^k)$, update $x^{k+1} \leftarrow x^k + \alpha^{k+1}\Delta x^{k+1}$ and increment $k \leftarrow k+1$ after return to the step of stopping

Table 2.3: Quasi-Newton method

In table 2.3 the Quasi-Newton method is presented.

Chapter 3

State of the Art

This chapter describes some methods of 3D Reconstruction of surfaces. Some Scientific papers where used to provide information to the state of art in this particular area of studies. The 3D reconstruction from calibrated images is an important problem and has been extensively studied by people in Computer Vision Science. The 3D reconstruction of some object can be obtained by 2 ways, Acquisition of high quality 3D models and Image-based modeling. The last one is actually a very low cost option of the system, basically the only input is a set of images possibly calibrated. Off course with this we will get a less quality 3D model. There is a really huge amount of literature about methods of 3D reconstruction. The 3 major techniques are Pure image-based rendering techniques and Hybrid methods,having this 2 methods the goal of getting a coherent view of the real world not to get real world measurements. With the last major technique is possible to try to recovery full 3D and for that there are 2 ways Active(control of the source of light) lasers are used to get all surface , and Passive- method that only use information present in 2D images of the object. Depending on the information used is possible to obtain different results , Silhouettes- Obtain visual Hull , but the result is only good if the object is very simple , Shading it needs a really control of light environment and for the last Color info that allows to get a really nice model if there is no light source variations.

3.1 Related Work

3.1.1 Silhouette and Stereo Fusion for 3D Object Modeling

One interesting method was purposed in [8]. Silhouettes, Shadings and features are some of the most important information present in an image. To use shading a very controlled calibration of light is needed. For silhouettes is important to extract the image from the background to get only silhouettes of the object. For last the features that are texture, points, contours). As the use of shading required a very controlled calibration of light in [8] only features and silhouettes are used. The question is how to use both information together. Deformation models allow that to happen. There are two techniques to do it, Classical Snake and Level set Methods. In the classical snake approach is possible to obtain a optimal surface with the minimization of global Energy.

$$E(S) = E_{texture}(S) + E_{silhouettes} + E_{regularization}(S) \tag{3.1}$$

- $E_{\text{texture}}(S) \rightarrow$ Energy term related to texture of the object.
- $E_{\text{silhouettes}}(S) \rightarrow$ Energy term related to silhouettes.
- $E_{\text{regularization}}(S) \rightarrow$ Surface model regularization term.

To minimize the Energy, the sum of the gradients of the energy of optimize surface should be 0 and each gradient represents one force that drives the snake. Therefore, the forces that drive the snake of the optimize surface should cancel each other.

Snake initialization: First step is finding the initial surface, that should be close enough to the real surface to guarantee the conversion of the algorithm. The lower the distance of the surface to the object surface, the lower is the number of iterations. A really good initialization is the visual hull that intercept the cones containing the object.

Texture driven force: This force uses the texture of the object to help reconstruct the 3D object shape. This force has to maximize the image coherence of all cameras for that the following projective geometry property is used: If a surface is visible in 2 different cameras then the 2 images are related by a transformation depending only in the 3D geometry of the object. This last property is only valid in perfect projective cameras, perfect lambertian surface and same lightning conditions. The chosen method for measure the image coherence was the normalized cross-correlation. For the optimization in [8] were proposed 2 methods, Proposed Voting Approach and Octree-based Gradient Vector flow.

Silhouette Driven Force: With this force is possible to match the snake with the initial silhouettes. When used alone the force makes the model to approach to the visual hull, but the force depend in the self-occlusion of the snake. If the snake matches already a particular silhouette , that one will be discarded by the rest of the snake because that silhouette has already been matched. Comparing the visual hull with the real object is possible to see that the object matches the silhouettes but not all points. The big problem is to obtain and differentiate the points that have to obey or not the silhouette. For that the force is decomposed in 2 parameters one that measure the fitting and other calculates how strong the force should be applied.

3.1.2 Accurate Multi-view Reconstruction

Accurate Multi-view Reconstruction , [9] is other reconstruction method. Having as an input a set of calibrated images and being the segmentation of the object from the background provided, that allows visual hull to be represented by silhouettes images, reconstruction can be obtained in 2 steps. Binocular Stereo on images pairs and for last surface reconstruction. The binocular stereo creates depth maps. Stereo matching is used to obtain the depth maps. Having a primary and a reference image first the images are rectified for the scanlines of both images lay in the epipolar lines. Then assuming brightness to be constant normalized cross correlation is used for obtaining the best match. The bad matches are ignored by the thresholding of the NCC metric. To compensate distortions, the use of a scaled-window matching technique is needed. This technique in the present of high curvature regions allows a large improvement in that regions. The binocular stereo pair is then converted to a single cloud of 3D points. After this first step, the goal is to obtain the triangular mesh from the point cloud. This reconstruction of a triangular mesh consists of 3 different steps: dowsampling , cleaning and meshing. In downsampling, and because generally the cloud is more dense then the actual density of points needed , hierarchical vertex clustering is used. In cleaning the remain noise is removed and for the last in meshing the triangular mesh is generated.

3.1.3 Poisson Surface Reconstruction

This method proposed in [10] by Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. A set of samples $s \in S$ containing the points s.P and the inward facing normal s.$\vec{N}$ as the input data and assuming that the normal lay in that is the model. With s the goal is to obtain a reconstruction of a waterlight, for that the approximation of the indicator function to the model is needed. The indicator function is the interception of the plane trough the object with the object s.$\vec{N}$. The question is how to compute this indicator function out from the samples. In this work [10] a relation between gradient of indicator function and integral of the surface normal field is obtained. With a summation of the integral of the surface over the sample points the integral of the surface is approximated and for last from the gradient field created , is possible to reconstruct the indicator function. The indicator function is a piecewise constant function and from the explicit computation of the gradient field results a vector field having , in the surface boundary, unbounded values. A way to change it is to convolve the indicator function with a smooth filter. Because the surface geometry is not known the surface integral cannot be evaluated. However with the input of orientated points is possible to approximate the integral. For that discrete summation is used. For the implementation the assumption of that all points are uniformly distributed has to be done.

Chapter 4

Problem Description

4.1 Problem Statement

The goal of this work is to optimize the 3D reconstruction of a liver surface. The input is a set of calibrated images obtained by the endoscope. The environment of the records is not the best for obtaining good quality images, and this brings a huge challenge regarding the reconstruction. All illumination comes from the endoscope and with that shadows and artificial features appear. The noise present in the set of images depends of the quality of the sensor. Other important factor is the movement of fluid and organs during the recording, which increases substantially the possibility of error. Consequently, the obtained reconstruction loses quality. In this work and , because features are used to optimize the reconstruction, the artificial features and shadows created by the poor illumination that comes from the endoscope have a huge impact in the results. The used of normalized cross correlation, that is explained in chapter 5, needs an uniform intensity of the colors in the record so shadows and artificial features are a big obstacle. This can appear in different places in different images resulting from this bad matches of NCC. With all that was stated above, the big challenge is to recover as much tracked points as possible to use our method. As will be possible to observe in chapter five not all the points will lead to a patch because some could result after the optimization in a bad match of NCC. All bad matches will be ignore in other words the patches with bad matches will be not saved.

4.2 Objectives

The principal objective of this work is to optimize the reconstruction of a liver surface. For that it is needed to check the method in other sequences like the book sequence that will be possible to observe in chapter 7. The main point would be having the obtained patches look for the true normal. Chapter 5 will explain how this search is done. The goal in the end is to obtain the best possible normal of parts of the surface and if a good cloud of points is obtained a robust model of the surface can be obtained.

Chapter 5

Patch Based 3D Reconstruction

5.1 Initialization of the Patch

Having a group of tracked points in all frames and choosing one starting frame and one tracked point initialize the patch is poissble. The first step is ,having a point in a frame , obtain the plane with a initial normal represented by the normal calculated with the result of the point in real world and the camera center. Having this, construct a rectangle $R(x,y)$ with $n \times n$ point in the neighborhood of p(image point). After this and being K the intrinsic matrix:

$$L = K^{-1} R(x,y) \tag{5.1}$$

L is the matrix of vectors representing the lines passing through the camera center and the points of R(x,y). Having L , and having $\vec{n}$ as the normal to initialize $\vec{p}_0$ as the chosen 3d point and $\vec{r}_0$ as camera center, is possible to obtain the equation of a plain:

$$\vec{n}(\vec{X} - \vec{p}_0) = 0 \tag{5.2}$$

And with λ as an unknown:

$$\vec{X} = \lambda \vec{L} + \vec{r}_0 \tag{5.3}$$

Then obtain the equation to calculate the variable lambda to each point of $R(x,y)$ that intercept the plane replace 2.3 in 2.2:

$$\lambda = -\frac{\langle \vec{n}, \vec{r}_0 - \vec{p}_0 \rangle}{\langle \vec{n}, \vec{L} \rangle} \tag{5.4}$$

5.2 Discrepancy Function

After obtain the 3d patch the goal is optimize the normal to do that we first calculate the discrepancy function $d(p)$:

$$d(p) = \frac{1}{I(p)\backslash I(0)} \sum d(p, I_1, I_n) \tag{5.5}$$

Being I(p) a set of images with the chosen point visible and I(0) the image of the initialization of the patch and $d(p, I_1, I_n)$ one minus the cross correlation between the interpolated values of I_1 and I_2. I(p) in this work is a set of images where the point was visible and that the rotation was bigger than some threshold value. To get the grey interpolated values of an image the projection is needed so we have to used the inverse transformation T that is the transformation between the camera and the world coordinates. So the pixels values of a frame are obtain by this follow equations:

$$x = K \times T^{-1} X \tag{5.6}$$

Being x the pixels coordinates K the intrinsic matrix and X the patch in world coordinates.

5.3 Optimization of the Normal

The discrepancy function will then be used to optimize the normal of the patch. In the initialization is assumed that the normal is the direction to the camera , but after to get good results , a minimization of the discrepancy function will provide a more accured normal of the liver surface in that specific patch. In this work and because Matlab is the used tool is possible to use a function of the Matlab library “Fminunc”. This function looks for a x that minimize F(x) so for that to happen F(x) should be created and in this case x should be the normal and F(x) should be the resulting discrepancy function. Basically the function will by a gradient look for a normal that minimize the discrepancy function. To simplify computation in this work the normal was parameterized to Euler angles (yaw and pitch) so x will have 2 parameters only instead of 3.

5.3.1 F(x)

This function will rotate the patch and project to the images(I_p) and will calculate again the discrepancy function. For doing the rotation the follow matrix is used: As is possible

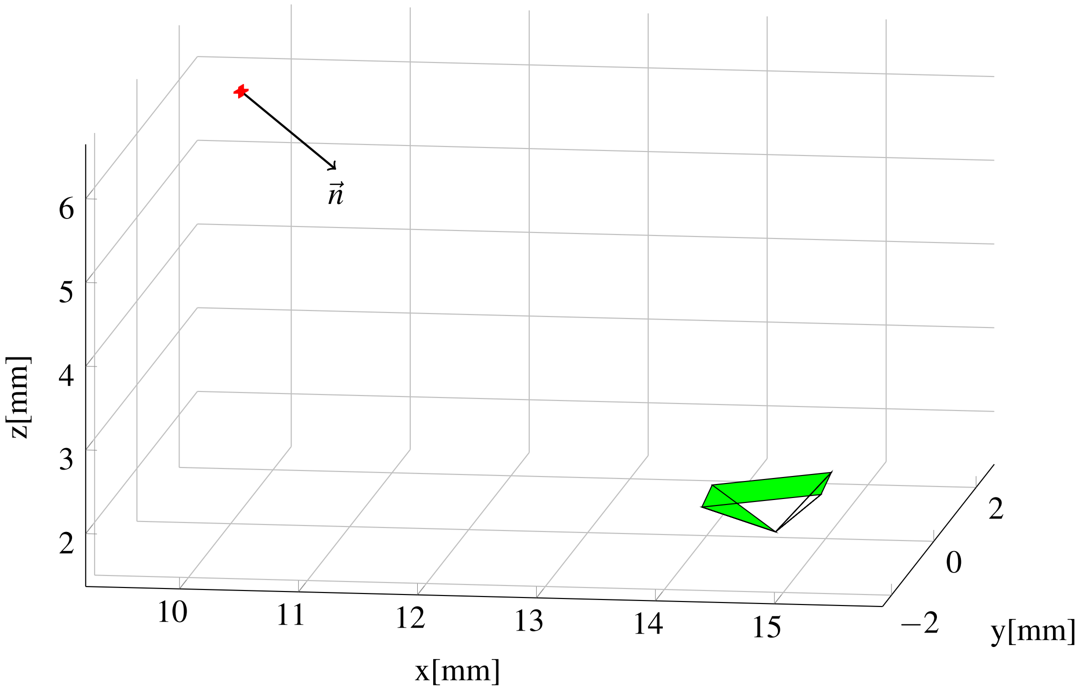


Figure 5.1: normal of the initial patch

to see in figure 5.1 and 5.2 the patch rotates to the optimum solution.

$$R(\alpha,\beta)=\begin{bmatrix}\cos(\alpha)\cos(\beta) & -\sin(\alpha) & \cos(\alpha)\sin(\beta)\\ \sin(\alpha)\cos(\beta) & \cos(\alpha) & \sin(\alpha)\sin(\beta)\\ -\sin(\beta) & 0 & \cos(\beta)\end{bmatrix} \tag{5.7}$$

Being α the yaw and β the pitch. After each rotation this function will compute the equations(5.6) and (5.5).

5.3.2 Minimization function

Having F(x) defined the optimization function is needed to obtain the minimum discrepancy function possible as we can see figure 5.4 that represents a cost function , “fminunc” will change x by a gradient of F(x) to obtain the minimum of F(x). Figure 5.4 show a contour cost function for the optimization. Is possible to observe that in this case that between point [0,0] and the dark circle, that represents the minimum, the cost function is decreasing so no local minimums. In this case the global minimum will be obtained. In Figure 5.5 is poissble to understand the behavior of “fminunc”, the discrepancy function is decreasing until getting almost flat and that is a local minimum so in that case that minimum should represent the correct normal of the surface.

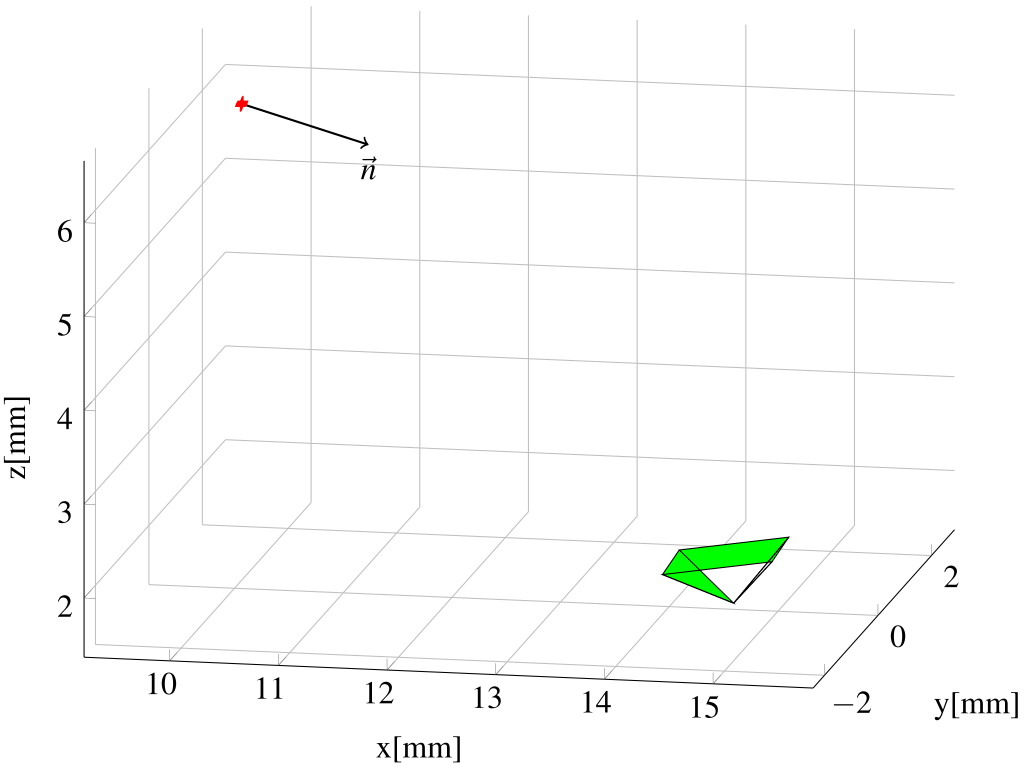


Figure 5.2: normal of the optimize patch

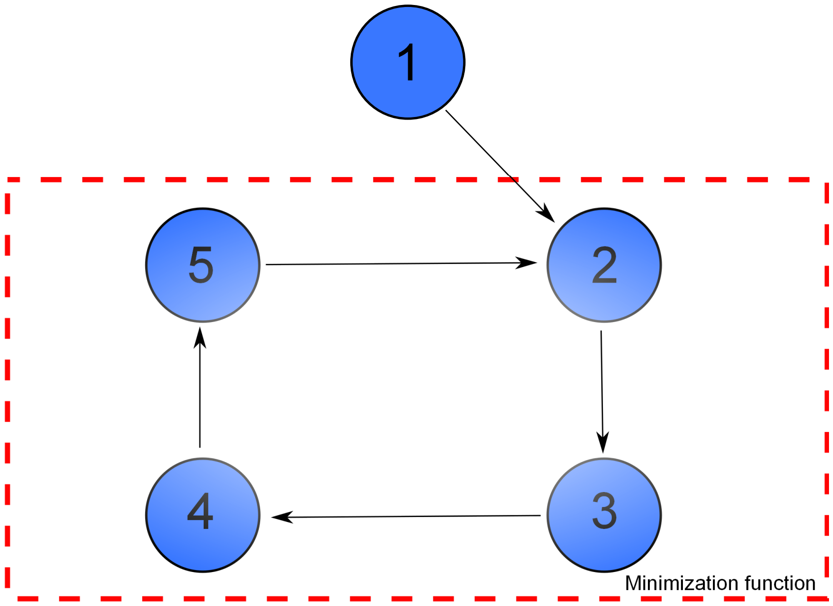
Only the patches with discrepancy bellow a certain threshold will be considered:

$$P(X) = \left\{ X \in I, \frac{1}{I(p) \backslash I(0)} \sum d(p, I_1, I_n) < \alpha \right\} \tag{5.8}$$

As is possible to see only when d(p) is bellow α ,being d(p) the last discrepancy function obtain in the optimization and α the threshold value, the patch will considered. Figure 5.3 shows a diagram of this minimization function this function will be called by the non linear optimization function “fminunc”.

5.4 Expansion

After having a Cell containing all patches we need to expand the patches meaning that we need to extend our patches with the goal of obtaining a better smooth surface. For that we need to get in count 2 essential rules if a cell has already a patch that cell can not be expanded by a neighborhood cell. The second rule is if after expanding the discrepancy function of the new create patch is too big the patch should not be expanded.



1- Initialize new patch
2- Rotate the patch
3- Compute the corresponded 2D points in images
4- Interpolate the grey levels of the windows obtained in 3
5- Compute the discrepancy function for the new patch

Figure 5.3: Diagram of minimization Function

In the figure 5.6 is possible to see how the expansion works in the neighborhood of:

$$\left\{x, y \in Image | \left\langle \Delta x^2 + \Delta y^2 = 1 \right\rangle \right\} \tag{5.9}$$

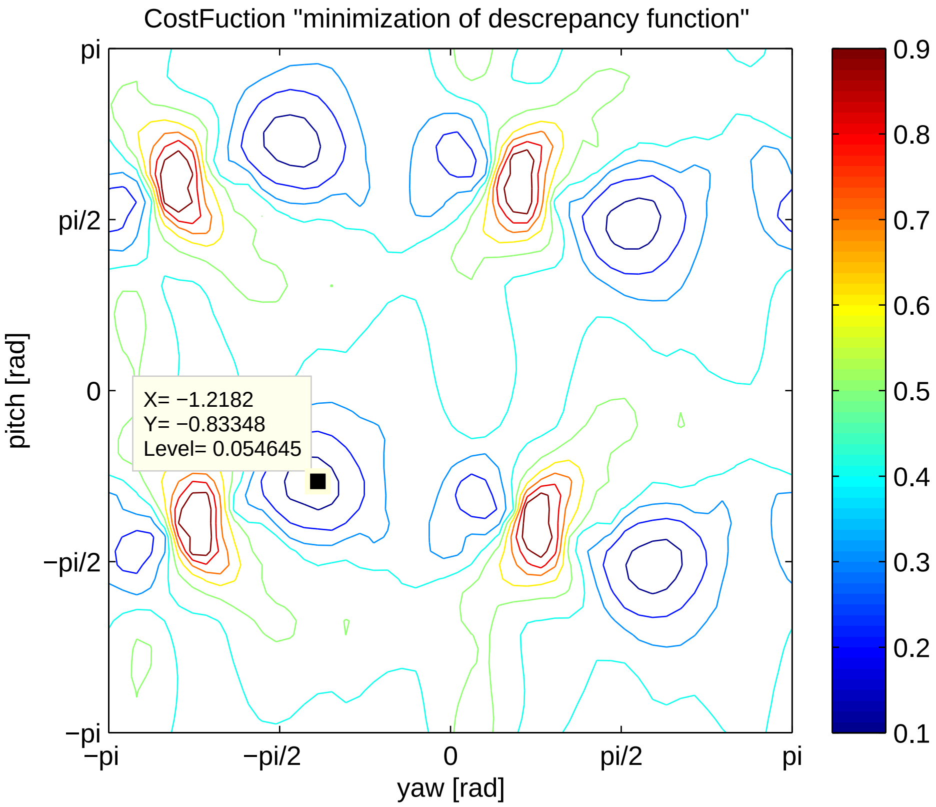


Figure 5.4: Cost Function

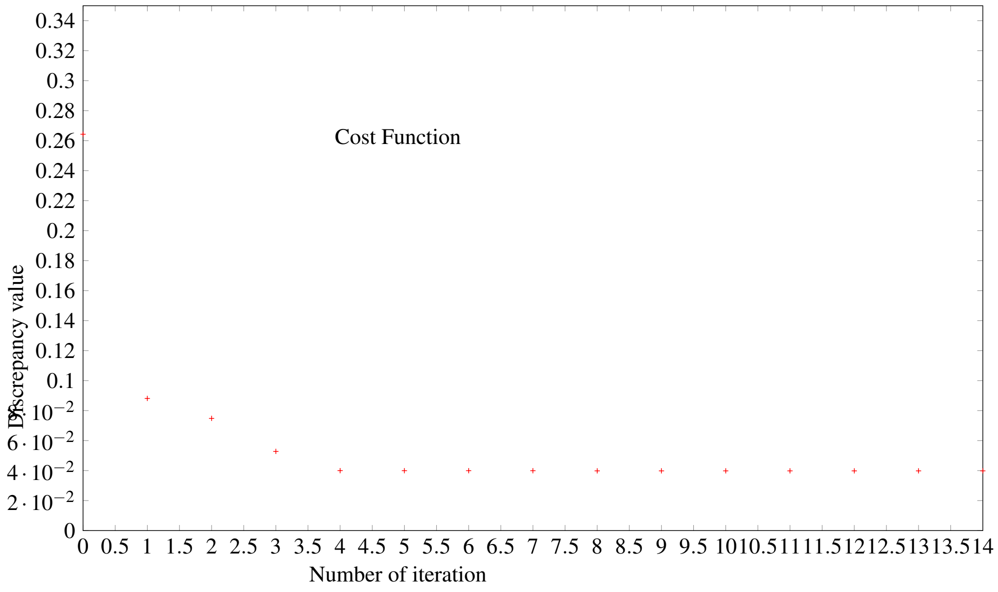


Figure 5.5: Cost Function

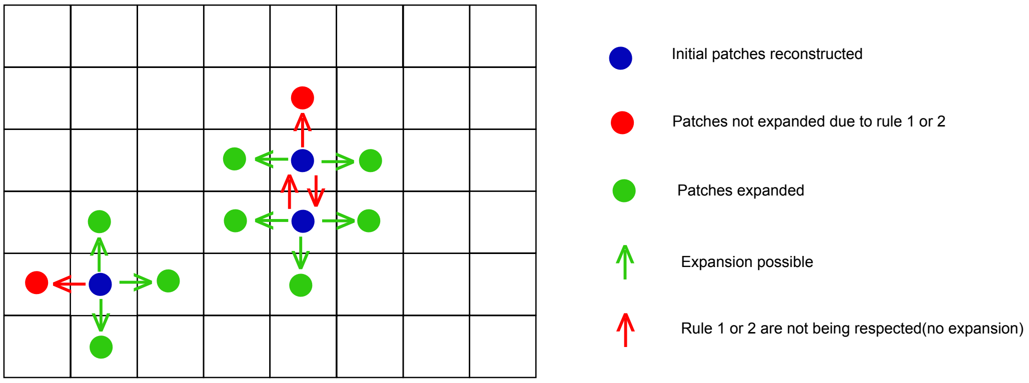

Initial patches reconstructed
Patches not expanded due to rule 1 or 2
Patches expanded
Expansion possible
Rule 1 or 2 are not being respected(no expansion)

Figure 5.6: Rules for expansion

Chapter 6

Implementation

6.1 Experimental Setup

The endoscope used in the experiment is “Adrolook USB Scope”. In table 6.1 the technical characteristics are presented. In figure (6.1) and (6.2) is possible to see the control unit and the rod of the endoscope. This endoscope works as a normal camera with the usb connection is possible to record the videos an easily transfer them to the computer. The calibration matrix was estimated by the Caltechzs Camera Calibration Toolbox for Matlab [11], and the result matrix is presented in equation (6.1).

$$\begin{bmatrix} 716.6 & 0 & 328.2 \\ 0 & 639.2 & 226.5 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.1}$$

The experiment of the book was recorded with an outside light source. The videos of the liver were given to the institute by the Medical Center Hamburg-Eppendorf. Another sequence tested was the model liver and the record was obtain inside a box with no outside light source using only the one of the endoscope. The box is totally black to avoid the light to enter inside where the liver model was positioned.

6.2 Implementation of the method

In table 6.2 is possible to see a quick resume of the method in this chapter the code will be explained and the m.file function will be presented. All code was obtain is Matlab and some functions were used that were created in previews works in the department.

Figure 6.1: unitcontrol. Figure taken from [http://www.adrolook.de]

Figure 6.2: Endoscope Rod. Figure taken from [http://www.adrolook.de]

In function create_3dpatch.m the patch will be created. The first function called is Matrix_initlzpxl.m, this function, having a chosen 3D point transform this point to 2D first with TCW(transformation camera world) and then with the intrinsic matrix. Having the corresponded 2D point the function Points_R_2D.m creates the window of points with a neighborhood defined in patch_paramfunct.m. After the use of function isInsideImage.m allows to check if this window is inside reference image. The final step comes with function project_patch.m. In this step the initial normal will be obtain and calling function project_P2_P3.m the window is projected until the plane defined by the normal having then the initial patch.

Diameter cabel	2.7/3.9/4.3/5.5/5.8/8/20/25 mm
Cable protection	IP57 waterproof/Dustproof
Light source 10mm cable	6 white LED(850lux/8cm)
Light source 5mm cable	6white LED(350lux/1.5-4cm)
Image sensor	CMOS
Lenght of cable	1/2/3/5/10/20/30 meters
White balance	tight
Resolution	720 x 480
Storage video format	AVI-Format
Storage image format	JPG-FOrmat
USB connection	Mini USB 1.1/2.0
Frame rate	30fps
Line of sight	0°
Field of view/angle	67°
Depth of field	1cm/8cm
Side view adapter	yes/accessory
Bending radius	10mm
Waterproof/oil resistance	yes/yes
Software	XP/Vista/Windows7

Table 6.1: Technical caracteristics of the endoscope obtained in http://www.adrolook.de/

In function optimization_patch.m the patch created before will be optimize. The first step is using Cameras_2project.m obtain the cameras to be used in the optimization. This function looks for cameras and check , with the functions isInsideImage.m and getcamerarotationangle.m, if the patch is visible in the camera and the angle with the reference camera. The first camera to be chosen will be one with bigger angle with the reference camera and one that patch is visible. Having the first one it will look for other one that the patch is visible too and the angle with reference and with the first chosen camera bigger than a threshold value in our case 4°. It will stop trying to find more cameras when is not possible to obtain other camera with angle of 4° in relation to the others. Having the cameras The next step is obtain the discrepancy function using the m.file Discrepancy_func. This function will rotate the patch using Rotation_Y_P.m with pitch and yaw. After it will call function optm_funt.m to calculate the discrepancy function that is 1 minus the cross correlation of the grey levels of the chosen cameras. For doing that is needed to obtain the corresponded 2D windows of the patch in the chosen cameras using project_secondC.m. Having the corresponded windows interpolation of the grey levels is needed allowing then to calculate the normal cross correlation between the windows. Now in the final step the use of fminunc.m allow

1. Pick a 3d point p(x)
2. Project to 2D to a chosen frame and Create a R(x,y)
3. Project points of R(x,y) to space with the direction represented by c(P),R(x,y) central of camera and point , to the plane defined by the normal $\vec{n}$ represented by c(P),R(x,y) central of camera and point , to the plane defined by the normal $\vec{n}$
4. Calculate frames that p(x) is visible and that have a bigger angle than some α value
5. Rotate the patch P(X) with R(yaw,pitch)
6. Having the frames [F] project all the points to each frame
7. Interpolate the Grey values for the project points of each frame
8. Calculate then the one minus normal cross correlation between the initial frame and F
9. Repeat points 5 , 6 , 7 and 8 until the Matlab function "Fminunc" finds a minimum to the discrepancy function d(f)
10. if d(f)<α → Patch is saved in P(cell)
11. Repeat all last points for all tracked points
12. Having all saved patches expand to the neighborhood obeying some rules
13. Having the expand patch repeat points 6 , 7 and 8 to get d(f) of the new expanded patch
14. if d(f)<α and P(cell) isempty →new patch is saved
15. Repeat points 12 , 13 and 14 until no more chances of expansion

Table 6.2: Implementation of the method

patch_paramfunct.m	This m.file produce the input of initial reference image neighborhood size spacing size.
create_3dpatch.m	This m.file produce the patch of a point
optimization_patch	This m.file looks for the best match patch
expand_patch.m	This m.file expands the optimize patches obtained

Table 6.3: first class m.files used

Matrix_initlzpxl.m	This m.file transforms a 3D point in 2D and call function Points_R_2D.m
Points_R_2D.m	This m.file produce a window of 2D points
project_patch.m	This m.file calculates the initial normal and call function project_P2_P3.m
project_P2_P3.m	This m.file projects the window obtaining the initial patch
project_P2_P3.m	This m.file compute with the optimize yaw and pith the optimize patch

Table 6.4: Functions called by create_3dpatch.m

Cameras_2project.m	This m.file Computes the cameras to be used in the optimization
Discrepancy_func	This m.file calls function Rotation_Y_P.m and optm_funt.m
Rotation_Y_P.m	This m.file computes the rotation of the patch with yaw and pitch
optm_funt.m	This m.file calls function project_secondC.m and computes the normalized cross correlation
project_secondC.m	This m.file calculates the corresponded window of the patch in the cameras of Cameras_2project.m

Table 6.5: Functions called by optimization_patch.m

the method to find the best yaw and pitch for getting the best match of the windows. Only if the discrepancy function is lower that a certain threshold that this optimization will matter or in other words that this patch will be saved. Assuming that the discrepancy was lower than the threshold value and having the optimize yaw and pitch is possible using function opt_patch_3d.m to get the optimize patch.

In each iteration if the patch is optimize in a way that the discrepancy function is lower than the defined threshold the patch is saved in a cell that is a $n \times m$. Being n and m the size of the camera divided by the neighborhood size. If a patch is optimize but there is already a patch in the cell the patch is neglected. In the function expand_patch.m the patches are expanded. This function try to expand in the 4 direction horizontal and vertical concerning the cells. It starts calling for the m.file expansion_patch_create to create a window of points to be reconstructed in the neighbor of the window corresponded to the patch and this action is repleted for the 4 direction. Then is needed to check if the patch exists already in the cell or if the patch has a good match or if will be excluded for having a big discrepancy function. The discrepancy function is obtain after obtaining

expansion_patch_create	This m.file Computes the cameras to be used in the optimization
checkpatches.m	This m.file calls function Rotation_Y_P.m and optm_funt.m
Cameras_2_project_expansion	This m.file computes the rotation of the patch with yaw and pitch

Table 6.6: Functions called by expand_patch.m

the new corresponded 3D patch of the expand window. Having the expand patch and computing the cameras to project using Cameras_2_project_expansion the discrepancy function can be calculated. For that function checkpatches.m is called. This expansion will only stop when no more expansion is possible.

Chapter 7

Results and Evaluation

7.1 Introduction

In this chapter some evaluations and results will be presented. For this evaluations some sequences of data were used. A surface of a book was recorded to make an error evaluation of the normal obtain with the method of optimization. After that a real video of a liver surface was used to test the method. Some m.files were used only for the evaluation of the method as is possible to see in table 7.1.

7.2 Results and Evaluation

7.2.1 Evaluation of the Minimization Function

The first evaluation needed was the minimization function. After obtaining the patch of a tracked point the patch should be projective multiple times two the reference and the other images until a minimum is obtained.

Rnasac.m	This m.file calculates with a set of points of a book sequence the real normal of the surface
costfunction.m	This m.file plots the contour graphic of the discrepancy function with variation of yaw and pitch
distance_point2plane_byline.m	This m.file calculates the distance of a point until intercept a plane

Table 7.1: Functions used only for the evaluation

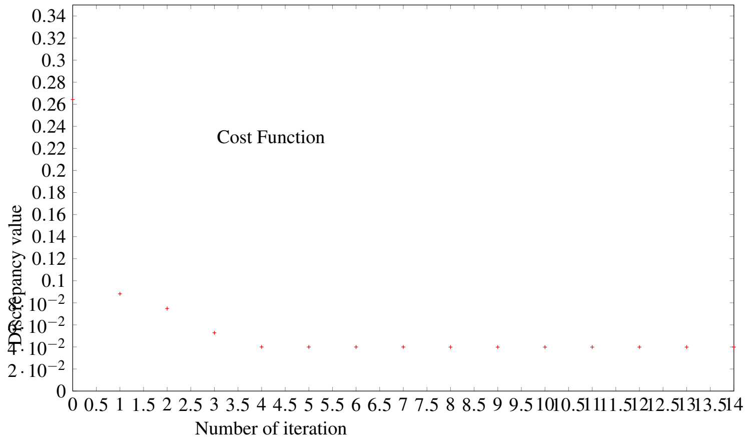


Figure 7.1: Cost Function iterations

In figure 7.1 is possible to conclude that for the optimization 6 iterations are enough because as is possible to observe in figure 7.1 after 5 the value does not decrease anymore. For checking if fminunc is working a function that plots the cost function is needed. costfunct.m is a function that get the plot of a cost function. As is possible to see in figure 7.3 the minimum is possible to achieve. The figure 7.3 is a counter plot showing the variation of the discrepancy function with yaw and pitch. There are two minimums and that is because there are 4 ways to obtain the optimize normal. Was possible to see that the minimization works better with a first neighborhood of 21 pixels because the cost function with small neighborhood is a flat so it finds local minimums before the optimize value.

7.2.2 Evaluation of the Discrepancy Function

As is possible to see in figure 7.2 the error increases with the decrease of the angle between cameras. That results happen because of the cost function associated if the angle between cameras is low the cost function is flat and is unable to find the optimize normal of the surface.

In figure 7.3 is possible to see a cost function of the discrepancy function with variation of pitch and yaw, in this case the global minimum can be found due to the continuous minimization until the global minimum. In figure 7.4 is not possible to find the global minimum the optimization will give us a local one due to the flat cost function. Until the optimization achieves the global minimum a local minimum will be found so with low angles between cameras is not possible to get a good match.

As is possible to observe in figure 7.4 the method was succeeded in optimizing the patch with 21 pixels of neighborhood but failed with 5 this can be explained by the lack of information with only 5 pixels. In the case of 5 pixels the beginning discrepancy

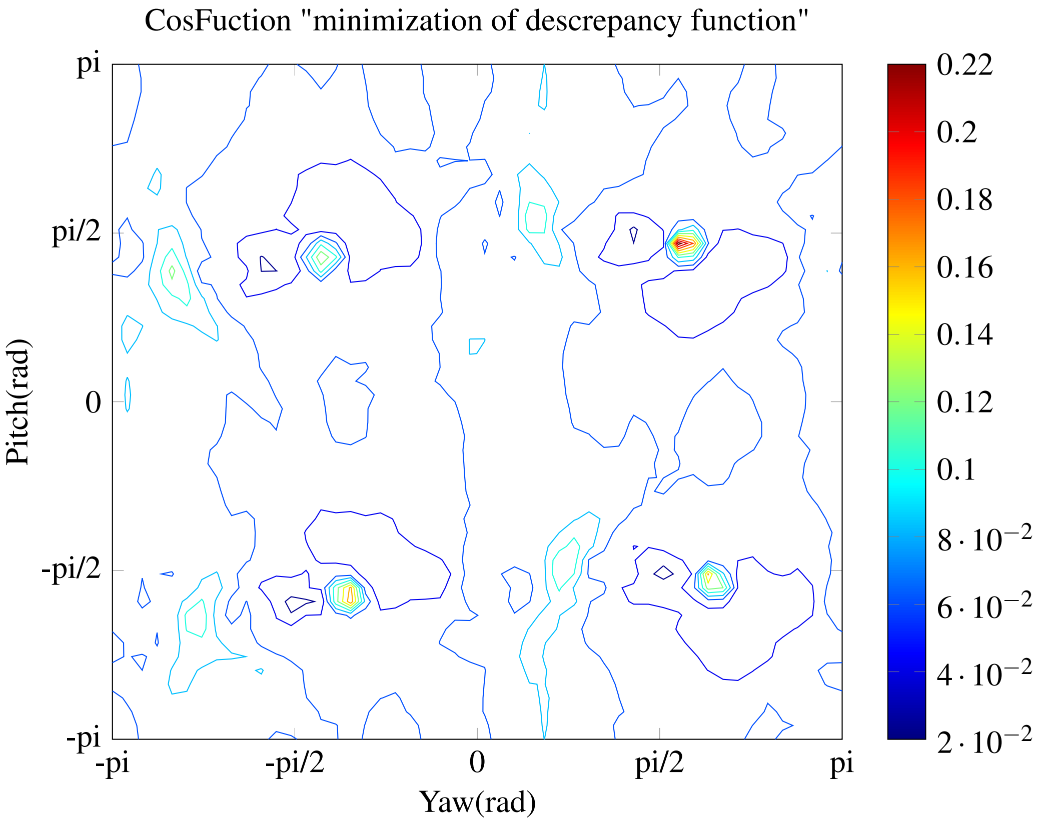


Figure 7.2: Cost function with 0,87° between the 2 cameras

is bigger and the cost function is flat not allowing the method to do the optimization. This does not happen in every cases in some cases in fact the method works better with 5 pixels for the liver. Due to the non flat surface of the liver this method can lead us to obtain windows with different information for example one window can have some points that in the other are not visible so the method in that cases will not work. Logically as more points of neighborhood used more chances of that to happen. So is not possible to state what is the optimum value but in my evaluation i concluded that using 21 pixels the results were quite better.

7.2.3 Evaluation of the Book Sequence with Ransac Method

To get a good evaluation of the results was needed to calculate the true normal of the book sequence for then compare the result values with the real normal values. For doing that ransac function was used “Ransac_plane.m''. Ransac (Random Sample consensus iterative method) estimates from data containing outliers, parameters. In our case we need to estimate the true normal of the book plane from a set of data with some large

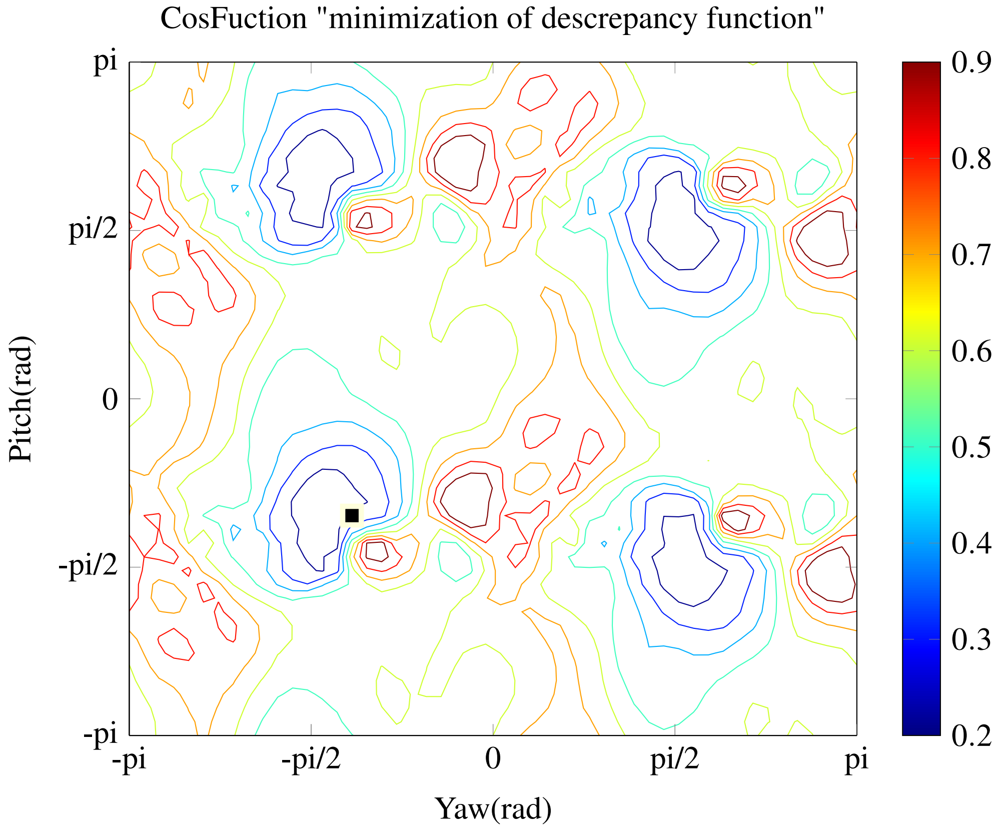
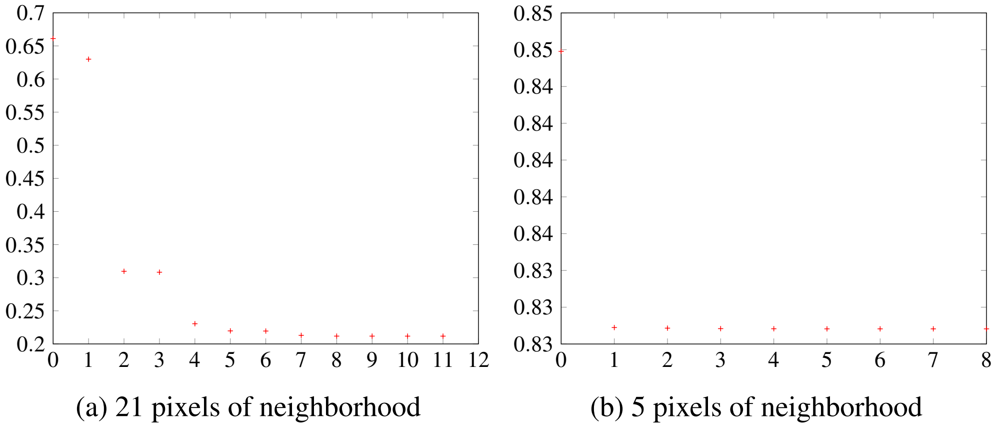


Figure 7.3: Cost function with 19,14° between the 2 cameras



(a) 21 pixels of neighborhood

(b) 5 pixels of neighborhood

Figure 7.4: Discrepancy function along the iterations with 21 and 5 pixels of neighborhood

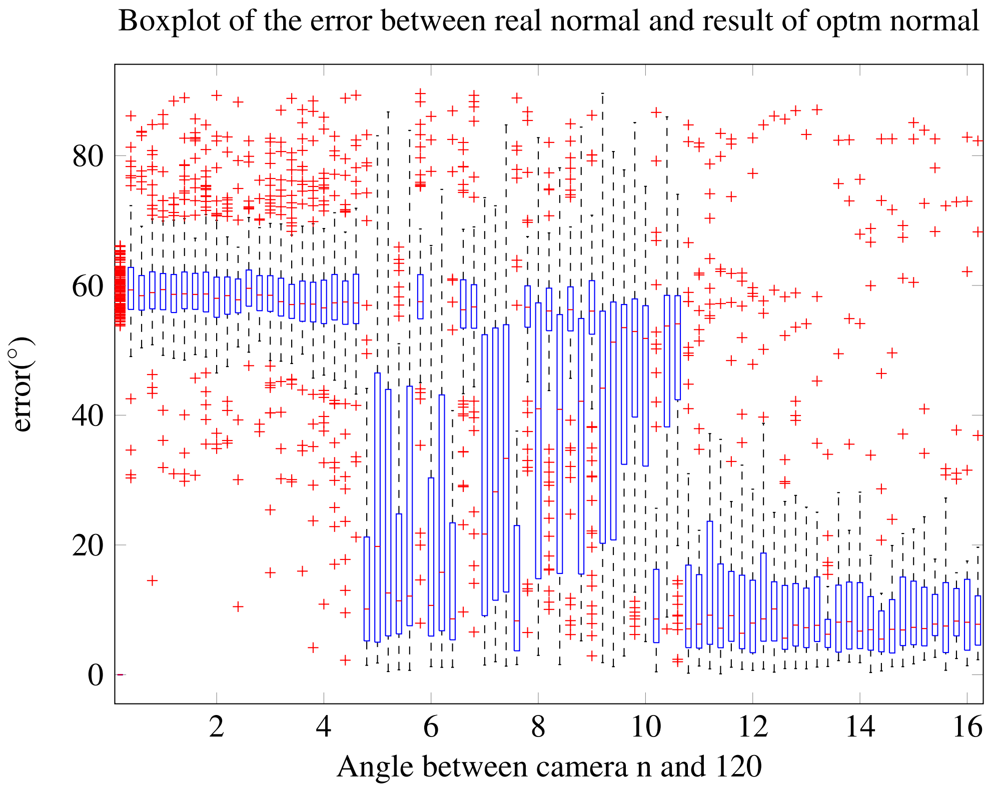


Figure 7.5: Boxplot normal angle error with different angles of the 2 used cameras for the optimization

error(outliers). So “Ransac_plane.m'' , iteratively calculates with 3 different random points each iteration one normal. After that calculates the number of points that are close to the plane at some chosen threshold distance. After some iterations the normal that will be chosen as the true one will be the one with less percent of outliers, being the outliers the ones further then the threshold distance. Having the true normal the next step is to obtain for each point the result optimize normal obtain with our method. The reference image is the same in all evaluations of the angle but the second project image varies and the angles between the reference and principal images should be obtain. With this we can plot the variation of the error between true and result normal with variations of angle between cameras In figure (7.5) is possible to see a box plot of the angle between calculated normal and real normal.

In figure 7.6 the box plot description is presented.

7.2.4 Book Sequence Results

The evaluation of the method in the book sequence was important to understand if the method was giving the right normal. As figure 7.7 shows, the normal of the reconstructed patches is always approximately the same as it should be due to the flat surface of a book. As the surface is totally flat the expansion works really nice. The problems of this evaluation were already explained in previews sections. In figure 7.7 the reconstructed patches are presented.

7.2.5 Liver Surface Results

The liver surface optimization is quite more challenging than the book sequence the big amount of shadows and bright spots are a big obstacle to the optimization other problem is presented in figure 7.9. As the cameras for calculating the discrepancy function are computed as the ones that the window in the reference camera is visible and the ones with larger angle with reference camera, the window on the other cameras can lay in a dark zone like in figure 7.9. With this the method, it can not obtain a good normal of the patch so in this cases the method fail.

In figure 7.11 7.12 and 7.13 are presented 3 different reconstructed patches with the corresponded 2D point in the image. In the first there is a slight deviation from the camera direction, figure 7.12 shows an example of a patch that was already with the correct normal when initiated. Is possible to see that the normal is the direction to the center of the camera. The last example is the one with bigger deviation from the initial normal and is quite easy to understand why looking at the 2D point in the image. In figure 7.14 is possible to observe the reconstructed patches with the use of 5 different reference cameras.

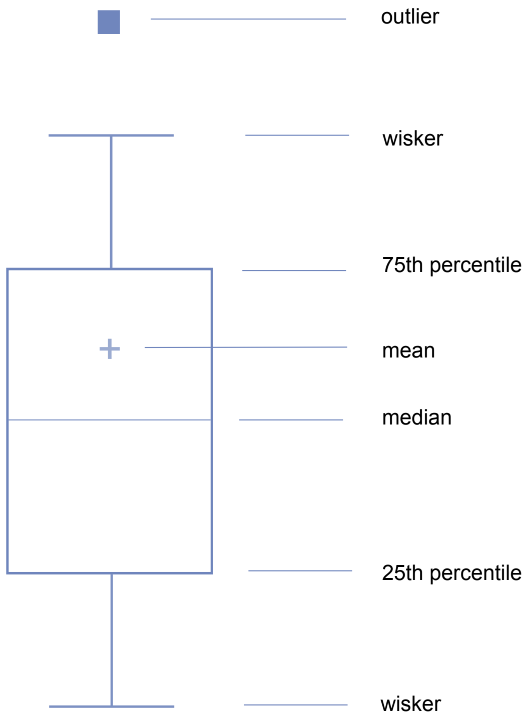


Figure 7.6: Box plot description

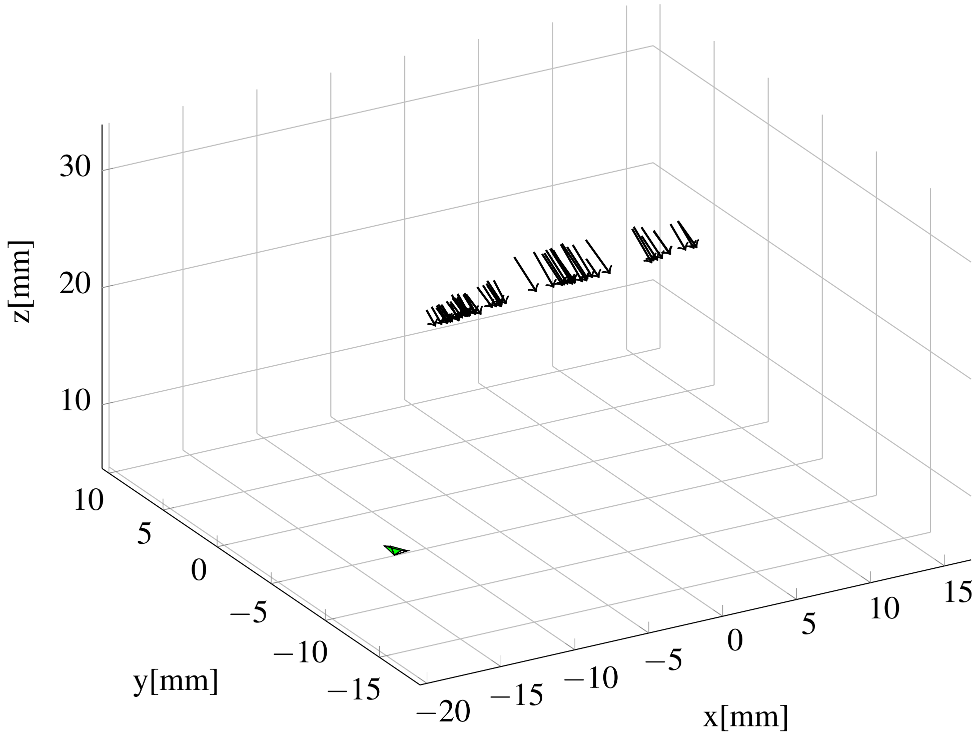


Figure 7.7: Normal of reconstructed patches of a book data sequence

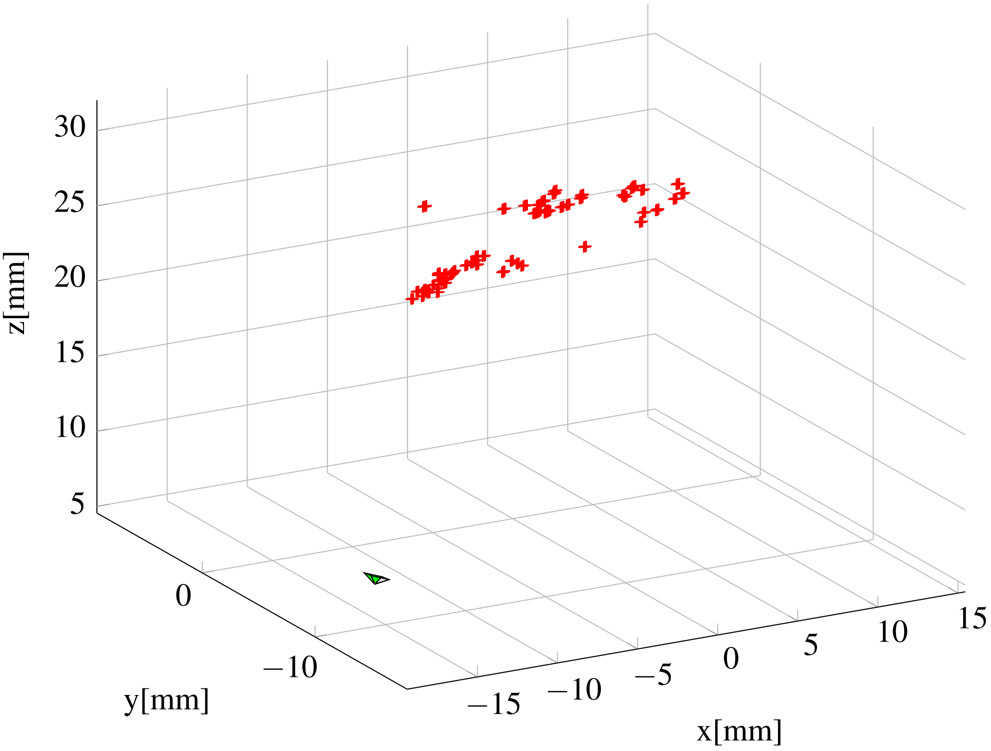


Figure 7.8: Reconstructed patches in book data sequence

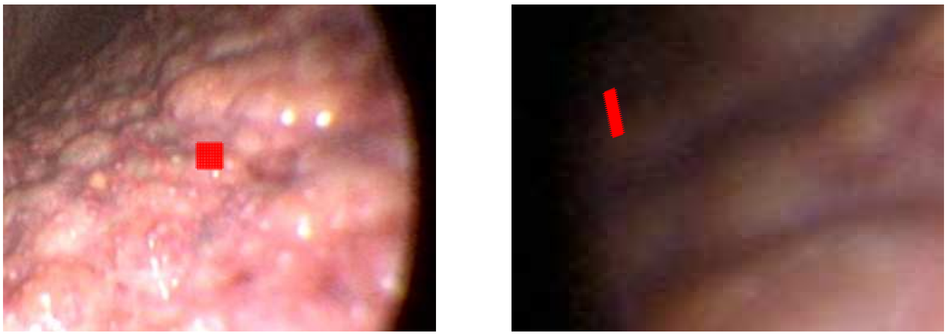


Figure 7.9: Correspondent window in camera 1 and 2

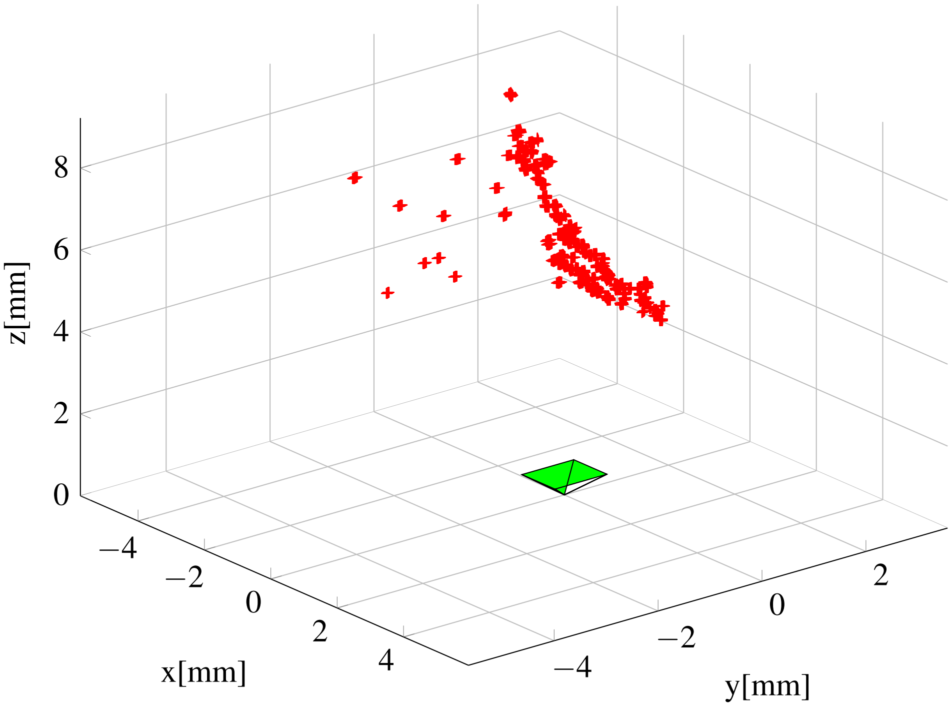
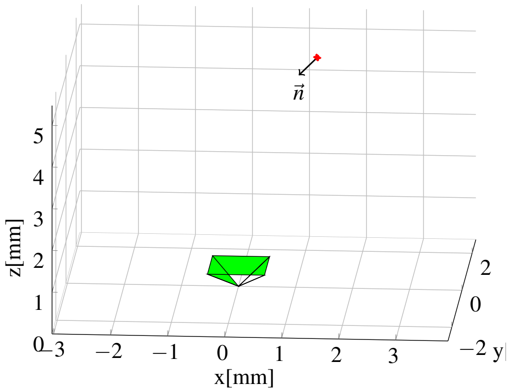
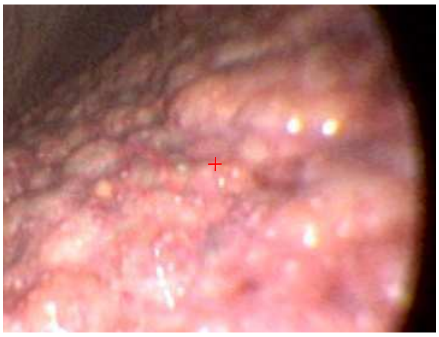


Figure 7.10: Patches Reconstructed using 1 reference camera

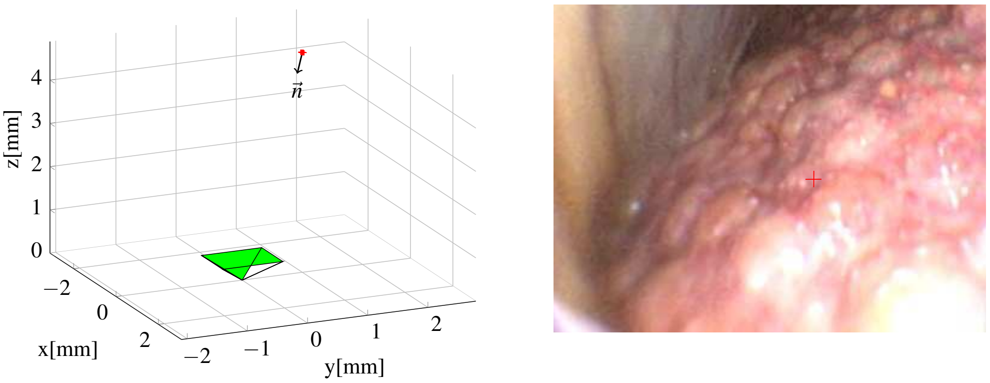


(a) optimized normal of the patch



(b) Corresponded 2D point of the center of the patch

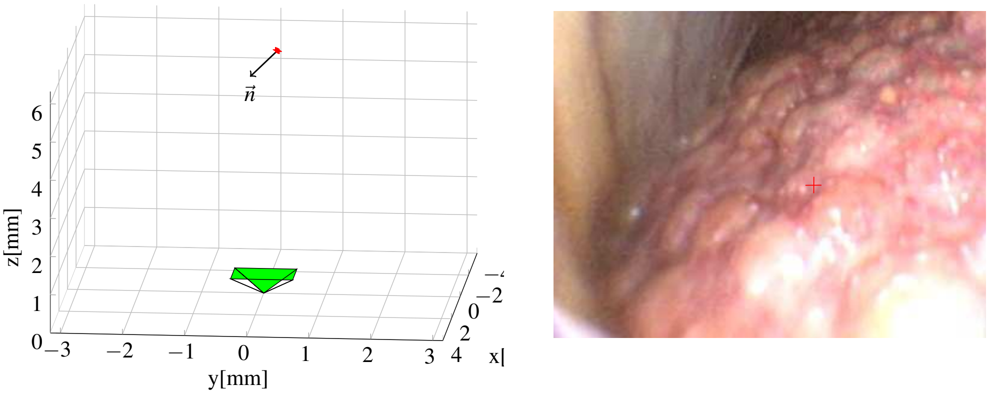
Figure 7.11: Normal obtained and corresponded 2D point in the image



(a) optimized normal of the patch

(b) Corresponded 2D point of the center of the patch

Figure 7.12: Normal obtained and corresponded 2D point in the image



(a) optimized normal of the patch

(b) Corresponded 2D point of the center of the patch

Figure 7.13: Normal obtained and corresponded 2D point in the image

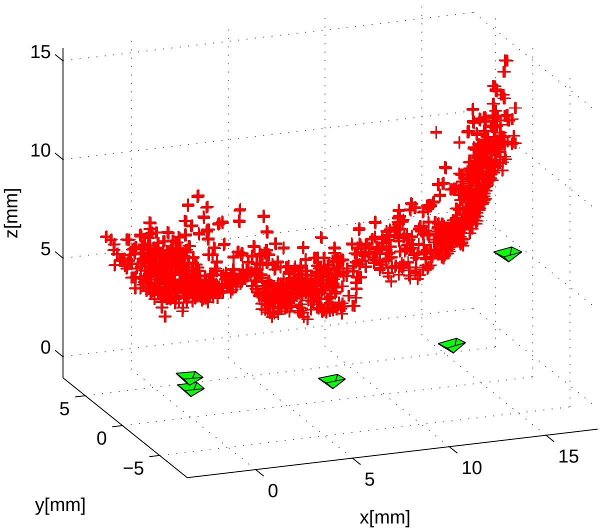


Figure 7.14: Patches Reconstructed with 4 different initial cameras

Chapter 8

Conclusions

8.1 Summary

As was mention in chapter 1 this project is a part of an ongoing research of the Vision Institute of TUHH with the University Medical Center Hamburg-Eppendorf (UKE). The goal of this entire project is to optimize the surface reconstruction of a liver having as a input a set of images from a endoscope video sequence. For implementing this optimization a set of matlab functions were conceived. This function for the method were presented in chapter 6. As was possible to see a set of evaluations had to be performed to insure the robustness of the method. After all the evaluations made during the research is possible to state that the matlab function “fminunc”, that is a nonlinear optimization method, is succeeded in finding the right normal of a 3D patch. Unfortunately that depends of the noise, in noise images the results start to lose quality and the errors start to increase. Other point is that this method only works if there is some texture, if the surface in constant in color and has no texture this method will find good matches (low value of discrepancy function) but wrong normal of the surface.

8.2 Conclusions

8.2.1 Minimization of Discrepancy Function

After all the evaluations made during the research is possible to state that the matlab function “fminunc”, that is a non linear optimization method, is succeeded in finding the right normal of a a 3D patch. Unfortunately that depends of the noise, in noise images the results start to lose quality and the errors start to increase. Other point is that this method only works if there is some texture, if the surface in constant in color and has no texture this method will find good matches (low value of discrepancy function) but wrong normal of the surface.

8.2.2 Results in the Book Sequence

Regarding the book sequence evaluation the method presented in this thesis showed to be able to obtain the normal of the book surface and after to be able to correctly expand the patches in order to obtain a good cloud of points. Some difficulties were also notice regarding the zones in the book with low or without texture. As was presented above the method of optimization does not work without texture, so the resulting normals in that zones were not correct.

8.2.3 Results in the Liver Sequence

Regarding the liver sequence evaluation the results were not so good. One evaluation with a 3D real model of a liver would be interesting to visualize the real error between the reconstructed surface and the real model. Without this evaluation was not easy to say for sure if the final cloud and corresponded normals were correct. The noise images of the liver made the optimization much more complicated. In other hand as was possible to check in figures 7.11 7.12 and 7.13 the optimization found a good normal for the surface because in that cases the condition were good and corresponded points of the first camera did not lay in a dark zone of the image. I conclude that if is possible to prevent that to happen the method will present us a good optimization of the 3D reconstruction.

Chapter 9

Future Work

In this chapter is presented some ideas for future work development, that can optimize the use of this method or even optimize the method itself.

9.1 VirtualGUI

To allow the user to have a better interaction with the reconstruction it will be nice to develop a VirtualGUI. It will allow for the user to plot some data and chose in a more pleasant way the input for the reconstruction. With this the user could also have a view of the state of the reconstruction and obtained graphic data of the procedures.

9.2 Video Sequence

As was referred in the last chapters the noise is really an obstacle to the use of the method. Would be really interesting to evaluate the method using a set of images with noise treatment. Of course this brings some downs due to the possible lose of information. The best option to improve would be to obtain better video sequences with less shadows and bright points, having for example a more stable light source. Would be really important to obtain the liver image sequences with no variation of brightness. This technology for now is not available but in the future maybe would be a possibility.

9.3 Method

As i previously mentioned in the liver sequence the images are really dark in the boarders due to the only endoscope source of light. Would be quite nice to exclude all images containing the corresponded window of the template when they are in the shadow zone of the boarders. Other improvement could be changing the neighborhood in each case depending in the best initial discrepancy. The method could first try to compute the initial discrepancy function for some defined neighborhoods and then compute the optimization with the value with lowest initial discrepancy.

Bibliography

[1] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(8):1362–1376, 2010.

[2] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge Univ Press, 2000.

[3] Lorenz T Biegler. *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. SIAM, 2010.

[4] Martin Hatzinger, ST Kwon, S Langbein, S Kamp, Axel Häcker, and Peter Alken. Hans christian jacobaeus: Inventor of human laparoscopy and thoracoscopy. *Journal of endourology*, 20(11):848–850, 2006.

[5] Daniel B Jones, Justin S Wu, Nathaniel J Soper, and Robert V Rege. Laparoscopic surgery: principles and procedures. *Journal of the American College of Surgeons*, 186(3):377, 1998.

[6] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

[7] John E Dennis, Jr and Virginia Torczon. Direct search methods on parallel machines. *SIAM Journal on Optimization*, 1(4):448–474, 1991.

[8] Carlos Hernández Esteban and Francis Schmitt. Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding*, 96(3):367–392, 2004.

[9] Derek Bradley, Tamy Boubekeur, and Wolfgang Heidrich. Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[10] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006.

[11] J.Y. Bouguet. *Camera calibration toolbox for matlab.*@http://www.vision.caltech.edu/bouguetj/calib_doc/, 2008.