SURFACE RECONSTRUCTION FROM STRUCTURED-LIGHT IMAGES

SURFACE RECONSTRUCTION FROM STRUCTURED-LIGHT IMAGES FOR RADIATION THERAPY

By

OLESYA PESHKO, B.Sc.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Science

McMaster University

MASTER OF SCIENCE (2005)
(Computing and Software)

McMaster University
Hamilton, Ontario

TITLE: Surface Reconstruction from Structured-Light Images for Radiation Therapy

AUTHOR: Olesya Peshko, B.Sc.

SUPERVISORS: Dr. Christopher Kumar Anand and Dr. Tamás Terlaky

NUMBER OF PAGES: xii, 136.

Abstract

In this thesis the surface reconstruction problem for radiation therapy is studied. The knowledge of the body's surface in the affected area is required to design and deliver proper radiation treatment for cancer patients. We develop an inexpensive non-contact optical 3D surface scanner prototype. It can easily be mounted on an x-ray treatment machine integrating unobtrusively into the current workflow.

We start with preliminary information on radiation therapy, explaining how the patient's body surface is used in planning and treatment, and discussing the currently available methods. Then, we review the commonly used optical 3D surface measurement methods, and discuss the advantages of structured-light techniques for the stated problem. In our method, the projected pattern is the regular rectangular grid of colour-coded circular spots. Images of the light pattern deformed on the surface are photographed by an off-the-shelf inexpensive camera and analyzed by the developed software.

Finally, we present the special purpose image processing and surface reconstruction algorithm, implemented as a MATLAB package, and provide the experimental results. Optimization procedures are used in the pattern recognition code to process low-quality images efficiently. The software package can run on any standard PC and takes a couple of minutes to process one image.

Acknowledgments

The thesis was written under the guidance and with the help of my supervisors, Dr. Christopher Kumar Anand and Dr. Tamás Terlaky. Their valuable advice and extended knowledge of the area helped me to do my best while working on the thesis. My special thanks are to the members of the examination committee: Dr. Jeffery Zucker (Chair), Dr. Skip Poehlman, Dr. Christopher Kumar Anand and Dr. Tamás Terlaky.

I would like to thank to the physicists of the Juravinski Cancer Centre in Hamilton, Ontario, Dr. Tom Farrell and Dr. Erik Griffioen for defining the surface reconstruction problem, explaining the requirements, and assisting with data collection. Also, I appreciate the great aid and support from all the members of the Advanced Optimization Laboratory.

Finally, I am indebted to thank my family and friends for their patience and understanding. Special thanks go to my father, Dr. Igor Peshko, for inspiration, fruitful discussions and continuous support.

Contents

List of Figures

Chapter 1

Introduction

In industry, art, science, and medicine it is often necessary to measure a 3D surface of an object with a good accuracy. In most applications, the surface is acquired either by using special devices, for example, range finders, or by recovering the 3D surface from one or more 2D images of the object. The latter utilizes image processing, pattern recognition, and surface reconstruction techniques that require the design of special purpose algorithms and extensive use of mathematical tools, such as, numerical computation, optimization, computational geometry, linear and vector algebra, modelling, 3D visualization, etc.

In this thesis, the problem of the patient's body 3D surface reconstruction arising in radiation therapy is studied. Although the particular conditions inherent in the given problem define the technical and algorithmic design, the developed 3D surface scanner prototype can be taken as a basis of surface reconstruction devices for other applications.

1.1 Surface Reconstruction in Radiation Therapy

Radiation therapy (RT) is prescribed alone or in a combination with other treatment methods, such as surgery, chemotherapy, or biologic therapy, to about half of all patients with cancer [16]. As high-energy radiation is employed to achieve the treatment effect in RT, it is especially important to plan doses, points and angles of entry of the radiation beams carefully. An exact patient's body surface in the affected area has to be measured. Radiation is absorbed by tissues in the body, consequently, the dose decreases before radiation reaches cancer cells. Therefore, it is important to know the distance between the target area and the body's surface from any angle of rotation of the x-ray machine to be able to provide an efficient treatment plan. Section 2.2.3 explains the RT process in more detail.

Section 2.3 describes Computed Tomography (CT), Magnetic Resonance Imaging (MRI), and mechanical contouring as the most widely used methods for surface reconstruction in RT. All these methods are characterized by slow data acquisition, and errors in repositioning a patient to a treatment machine. Firstly, some displacement can occur because imaging machines are separated from the treatment ones, and, secondly, body positions advantageous to treatment are often not possible in the restricted volume of imaging machines. MRI and CT technologies are expensive, and contouring is a contact method, causing patient's displeasure and deforming soft tissues easily.

Besides planning, the 3D surface can be used to provide the patient's position relative to a radiation source for each treatment procedure instead of using immobilizing devices and tattoo marks (see Section 2.2.3). And, finally, having a way to inexpensively and quickly measure the surface each time will provide data for medical research on anatomical variations in patients during the course of treatment, and help to determine if re-planning or any corrections to the treatment plan are necessary.

1.2 Project Objective

Analyzing the disadvantages of the existing methods described in Section 2.3, the requirements of the radiotherapy team, and hardware constraints (see Section 4.1.1), we propose an inexpensive, robust, rapid, non-contact portable 3D surface scanner, designed to integrate unobtrusively into the current workflow, using optical methods.

The task of developing an industrial 3D surface scanner for RT treatment procedure has two successive steps:

1. Designing a device with a software package that reconstructs the 3D body's surface in the area of treatment.

2. Extending the software package by adding the modules for, but not limited to, maintaining a database with patients' surface scans, and comparing the current scan with the previous data. This will provide possibilities to make corrections to the patient's position, to analyze anatomical changes since the last visit in order to determine if the treatment replanning is necessary, and to conduct some relevant medical research.

In this thesis, the first problem is addressed, the problem of 3D body's surface reconstruction. We develop a prototype 3D surface scanner device, design and implement algorithms for analyzing the acquired data and reconstructing a surface. More specific problem statement can be found in Section 2.4.

1.3 Approach and Methodology

Among the many optical 3D surface measurement methods (see Chapter 3), we use the well known method of structured light with a new specially designed light pattern. A structured light technique is inexpensive and flexible, and can provide fast full field measurement. In structured light the surface is illuminated by a light pattern of some known structure, for example, lines, a grid, or circles, which deforms on the surface. Only a camera and a pattern projector are needed. By choosing an appropriate pattern it is possible to reconstruct the surface from a single image, although multiple images can be used to increase robustness and accuracy. In structured light, the surface is reconstructed by finding the 3D coordinates of the surface points from triangulation between a light source, a camera, and a surface. The correspondence between the projected and the observed in the images patterns has to be known. For more details see Sections 3.10 and 5.4.

Technical design is defined by two main constraints:

1. The 3D surface scanner must not interfere with the usual planning or treatment procedure. Consequently, the examined surface should always be visible to the device, independent of the machine's position.

2. The device should not be expensive.

Taking into consideration these two features, we aimed to construct a lightweight device, which can be easily attached to the rotating machine's head, using an existing carrier tray. We also use an existing light source inside the machine, a pattern mask and a small inexpensive camera to produce structured-light images of the scene. As a result of the operating environment and choice of the camera, the captured images are low-light and noisy (see Chapter 4). In this thesis, we develop theoretically and implement in MATLAB a novel special purpose pattern recognition algorithm (see Chapter 5), which allows obtaining necessary information for surface reconstruction even from low-quality images.

The novelty of the algorithm is in the following features:

1. As the projected pattern, we use colour-coded round spots located on a regular rectangular grid and a black background. To our knowledge, nobody has previously used this kind of pattern. The detailed description and motivation for the chosen pattern is provided in Section 4.2.

2. The incoherent light source is not as powerful and focused as a laser, and human skin diffuses light. Therefore the projected light pattern is dark and diffused in the images. To find the light pattern elements in the image, which is necessary for surface reconstruction, we have to provide a good guess of their location. In addition to rectangularly located spots in the pattern, we also use a special hardware setup described in Section 4.1 to create images with a certain structure, which helps to reveal the location of the pattern elements (see Chapters 4 and 5).

3. Due to diffuse and low-light images, the image processing techniques, which can be applied, are limited. For example, we are unable to utilize commonly used edge detection to find the pattern elements and their shapes. Instead, we apply least-squares optimization procedure to fit the

acquired data to the inverse polynomial function, which we use to model the intensity distribution of the light spot. Related work on estimating spot parameters for rigit target detection was done by Korobkine [21], though in that case a polynomial model was used, which can give more error than the inverted polynomial due to its more free shape.

4. Spots in our pattern are colour-coded to help to establish the correspondence between the projected and the observed pattern. The problem is that with low light and diffuse skin properties, the colours change, and sometimes it is hard to define a colour of a spot. To solve this problem, we use a k-means clustering algorithm to separate the colours, which does not rely on the "ideal" colours but on the proximity information between the data points inside the group (see Section 5.3). However, this is not enough because the k-means algorithm also does not assign all the colours correctly. We use heuristics based on the pattern structure and spatial information to correct the colour errors (see Section 5.3.2).

1.4 Thesis Outline

The thesis describes the theoretical background of the 3D surface scanner for RT, its technical and algorithmic design, and the implementation of the theoretical results as a MATLAB software package. This predetermines the following organization of the thesis.

In the current Chapter 1, we state the surface reconstruction problem in RT, the thesis project objective, and briefly describe chosen approach to solve the problem. Finally, the outline of the thesis is provided.

Chapter 2 contains the background information on cancer treatment and the role of surface reconstruction in the treatment process. We also discuss the ongoing research and existing methods of body surface reconstruction for RT in detail, pros and cons analysis of which leads us to the problem formulation.

Chapter 3 reviews optical 3D surface measurement methods, their principles of operation, advantages and disadvantages, and concludes with the structured light approach, which was chosen to solve the surface reconstruction problem.

Chapter 4 is devoted to the technical design of the 3D surface scanner prototype. Technical and optical constraints determining the features of the scanner design are analyzed. Special attention is paid to the projected pattern design used for creation of the structured-light images. Our pattern design is based on characteristics of the human skin, and on camera and light parameters. An optimization model for the recognition procedure of the pattern elements is formulated.

In Chapter 5 our algorithm for image processing, pattern recognition, and surface reconstruction is described step by step. Experimental results are provided to illustrate the algorithm stages.

Finally, Chapter 6 contains concluding remarks and suggestions for future work.

Appendix A contains the high-level documentation of our MATLAB software package, which implements the algorithms of the 3D surface scanner concept that we developed in this work.

Chapter 2

The Role of Surface Reconstruction in Cancer Treatment

In this chapter we give some general information about cancer treatment and concentrate on the RT equipment and procedure. The use of the patient's body surface is explained. We discuss advantages and disadvantages of the currently employed surface reconstruction methods in RT, related research, and conclude by presenting the surface reconstruction problem.

2.1 Cancer Treatment

We start with giving some general information about cancer, what makes cancer cells so dangerous to human health and life, and a short review of different types of conventional therapy.

2.1.1 Cancer Cells and Their Behaviour

Cancer occurs when a group of cells start to divide and grow in an uncontrolled and disordered way. They form a tumour, which can invade surrounding tissues, start secondary tumours, and interfere with body functions. The causes for some cancers are known while those for others are not. The process of tumour formation is called *carcinogenesis.*

The number of cells in a tissue depends on a balance of cell proliferation and programmed cell death. Sometimes, a group of cells disobeys the control mechanisms of a human body and that is how malignant transformations occur. Potentially, almost any cell in a human body can turn malignant. In order to develop into cancer, the *genetic material* (DNA) of a cell has to be damaged. Changes can be caused by chemical, viral, and physical agents. Cancer cells can be less genetically stable than the normal cells, which causes cell heterogeneity within any given tumour [16]. In most cases, presence of a tumour-promoting agent is needed to complete the process of carcinogenesis.

In fact, there are two types of cancer causes. The nuclei of some cells have much stronger inherent predisposition to become malignant than the others – a cause called a *tendency.* A second reason is the *triggers* that cause the actual genetic damage and push cells into becoming malignant [9].

In conclusion, the activity of cancer cells can be described by five properties [9]:

1. Cancer cells grow and multiply when they are not supposed to.

2. Cancer cells can invade surrounding areas.

3. Cancer cells can get into the bloodstream, or lymph vessels and travel to distant locations in the body.

4. Cancer cells can cause secondary tumours, called *metastases*, at those distant areas.

5. Some cancer cells may produce substances that interfere with the body control functions and impact nerves, muscles, salt regulation, and other systems.

2.1.2 Main Types of Conventional Therapy

Three approaches are commonly used in cancer treatment: *surgery*, *radiotherapy*, and *chemotherapy*. Recently, the new field of *biologic therapy* has been opened. This Section provides a short overview of these four therapy types in the order of invention [9].

Surgery

This method has been utilized for more than two thousand years for cutting out individual tumours. Surgery is the best choice when cancer cells are totally contained in the area, which has to be removed, and there is a low probability of secondary tumours [9]. Sometimes, this procedure is a necessity if a tumour threatens important organs or tissues because of its size.

Radiation Therapy

This form of treatment was invented a hundred years ago, shortly after Curies' discovery of radium, a chemical element with a particular type of emission. This type of radiation, named *gamma rays*, was first used to treat cancer in France shortly after the discovery. In 1895 Wilhelm Conrad Röntgen invented a method of producing *x-rays*, which were used to image the human body [9]. Now, high-dose x-rays, or other similar types of rays, are used to kill tumour cells. The main advantage of the RT is that it can be precisely directed to the cancer area omitting other parts of the body.

Chemotherapy

Chemotherapy began in the late 1940s. During the First World War, mustard gas was used as a weapon because in high doses it caused destruction of the bone marrow. Later, it was found that a derivative of mustard gas – *mustine* – could kill growing cells, including cancer. This was the first chemotherapy drug. Afterwards, other components were added. At the same time, it was noted that certain *hormones* (or substances that resemble hormones) can also be used effectively to control cell growth [9]. The disadvantage of this method is that drugs are usually given orally or intra-venously, and therefore, circulate through the body before getting to the target.

Biologic Therapy

Over the last twenty years, researchers have been studying the ways body responds to cancer and trying to modify them in a certain controlled way to help the body fight the disease. Some substances can change the elements of the immune system so that it works more effectively against cancer. They are called *biologic response modifiers*, or *BRMs*. An example of an agent used in *immunotherapy* is Interferon. Other types of biological agents are being considered, for instance, it is possible to use *antibodies* to carry toxins or chemotherapy drugs or radioactive isotopes to cancer cells [9] but this mechanism still needs improvement. Examples of biological therapy also include *gene therapy*, *vaccines* and *antiangiogenesis therapy* where agents attack specific features of the cancer cells that are not present in normal cells.

2.2 Radiation Therapy

The primary application of the 3D surface scanner developed in this thesis is to assist in RT treatment. In order to understand specifics of the application and motivation for the design choices, it is necessary to discuss some background information in more detail, including principles of RT equipment, and radiotherapy planning and treatment procedures.

2.2.1 Principles of Radiation Therapy

RT is prescribed alone or in a combination with other treatment methods to about a half of cancer patients for *cure*, *palliation* (easing of symptoms), or the *local control* of a malignant disease [20]. Sometimes when a tumour has already been surgically removed, some cancer cells can still be left in the area. In this case, RT is prescribed to prevent a *tumour recurrence*. In some cases, RT is used to treat non-cancerous benign tumours.

The advantages of RT is that, as opposed to chemotherapy, it is free of systemic toxicity because it does not require an absorption at a distant site, transportation via the blood vessels, or diffusing from vessels into the tissues. Also, it does not have anatomical restrictions as surgery does. Besides, patient's medical problems, such as cardiac or pulmonary disease, are usually not an obstacle for radiation treatment. After-treatment complications are reduced by the introduction of high-quality megavoltage equipment, better dosage measurements, more careful, computer-assisted treatment planning, and advances in the understanding of radiation biology. Side effects are common to those of chemotherapy but are usually localized to the body site irradiated. For example, alopecia, i.e., loss of hair, occurs only in the treatment field, and nausea

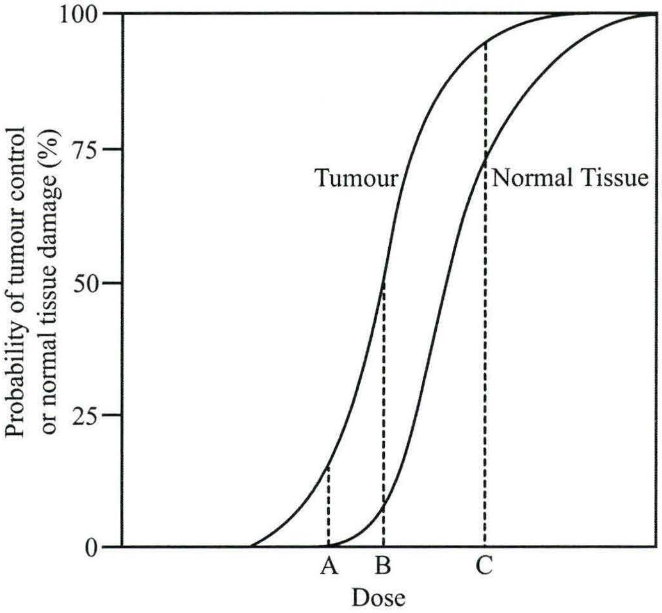


Figure 2.1: The relation between x-ray dose and outcome of radiotherapy

and vomiting can occur after irradiation of the upper abdomen [16]. Also, a disadvantage of the method is that patients usually have to attend procedures on a daily basis, and the whole course of treatment can take up to two and a half months though the procedure itself takes minutes.

Radiation Treatment Effect

The rays used by RT can be created artificially using, for example, *linear accelerators*, or given off by *radioactive substances.* The radiations can be *electromagnetic* (x-rays, gamma rays) or *corpuscular* (electrons, protons, heavy ions, neutrons, alpha particles) but their mechanisms of interaction with tissues are similar [16].

The main effect of radiation treatment is cancer cell death. Most cell death results from misrepair or failure of repair of double-strand breaks in DNA, an injury which can lead to a cell death only when the cell is dividing quickly (cells in division process are much more sensitive to x-rays) [16]. High-energy radiation deposits energy in body tissues through a process of *ionization*, which creates positively and negatively charged particles. The ions cause chemical reactions in cells, and highly reactive *free radicals* are formed, which cause breaks in DNA. Double-strand breaks can occur after a single hit of dense ionization or be formed from two independent neighbouring single-strand breaks. Single-strand breaks are usually repaired efficiently and are rarely lethal. Double-strand breaks can be repaired as well but are more prone to errors, which change the genome. The most common consequence is a cell's loss of reproductive ability [16].

Figure 2.1 represents the relation between an x-ray dose and probability of the tumour or normal tissues damage (see Fig. 7-1 on p. 56 in [16]). As implied by Figure 2.1, at any given dose there is a balance between cancer area control and damage to the normal tissues. An oncologist chooses the dose depending on the patient's condition, the treatment objective, and the characteristics of the normal and tumour tissues in the affected area. For example, choosing dose A does not harm healthy tissues but at the same time is not enough to cure cancer because with this dose only about 15 percent of cancer cells are killed. On the contrary, choosing dose C kills almost all the cancer cells but damage to the normal cells is too high. Dose B results in a lethal outcome for about half of the cancer cells with relatively small damage to the normal cells.

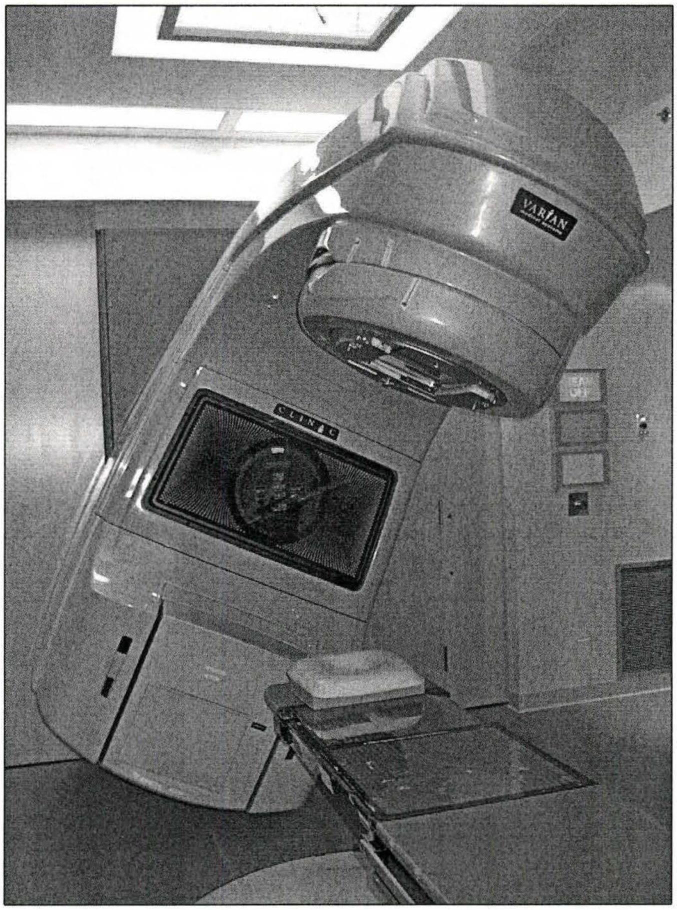


Figure 2.2: X-ray treatment machine

2.2.2 Radiation Therapy Types and Equipment

In this Section the three main types of RT are described, that is *external beams*, *brachytherapy*, and *large-field RT*. They differ by the treatment area and the equipment. External beams is the most frequently used type of radiotherapy.

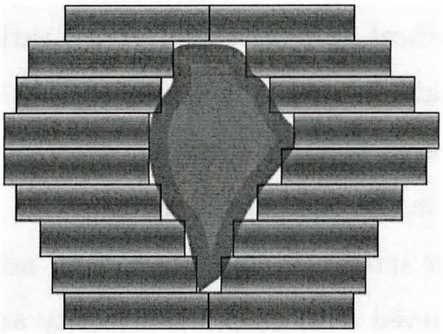


Figure 2.3: Multileaf collimator

External Beams

Modern RT was made possible by equipment capable of precise generation and delivery of very high energy radiation. These energies are expressed as *MeV* (million electron volts potential) Radiations with energies greater than 4 MeV are called supervoltage or megavoltage and used almost exclusively for external-beam therapy The most common source is a *linear accelerator* (LINAC) with the use of high-energy x-rays or high-energy electrons. Many facilities have *dual energy linear accelerators* [20] This means that the LINAC has two levels of high-energy x-rays: at the lower and at the higher ends of high-energy spectrum. Superficial tumours, such as head and neck cancers, are best treated with lower energy beams, while deep tumours of the chest, abdomen, and pelvis are better treated with higher energy x-rays [20] Megavoltage beams have very important characteristics, such as skin sparing, reduced absorption in bone, high dose rates, and reduced lateral scattering into adjacent tissues. The penetration of a beam increases in direct proportion to the energy of the photons [16]

Figure 2.2 shows an example of an x-ray treatment machine, CLINAC produced by *VARIAN Medical Systems* (http://www.varian.com). The machine can rotate around the table where the patient rests, which enables the exposure of the body's affected area to radiation from different angles. A high-energy x-ray source is located in the head of the machine.

The beam is more intense in the center than near the edges. The best treatment effect is achieved with constant intensity across the field of irradiation. This is why a flattening filter is put below the target. At the same time, if it is decided that a beam should be more intense on one side than the other, high atomic number filters, known as *wedges* are placed in the beam [30]. In order to be able to change the field of treatment, special high atomic number *collimators* are installed into the machine. They can vary the size from 4 × 4 cm to 40 × 40 cm at a distance of 100 cm from the target where most treatments are performed [30].

After opening the collimators to the desired radiation field size including the tumour some normal tissues in the field can still be shielded by placing *blocks* (or alloy), constructed of a combination of bismuth, tin, cadmium, and lead, in the path of the beam [30]. Then, it is possible to treat with a higher dose compared to the case where normal tissues were not covered. Some modern machines have *multileaf collimators* that are located in the linear accelerator's head and can adapt to the tumour's shape. Figure 2.3 demonstrates the principle of multileaf collimator's operation. The positions of the individual "leaves" are planned with a computer. To give an example, the *VARIAN Medical Systems* company produces *MillenniumMLC* multileaf collimators that offer $40 \times 40\ cm^2$ field, 0.5 cm resolution and can have 52, 80, and 120 "leaves" (http://www.varian.com).

Brachytherapy

Brachytherapy is an application of radioactive sources within or adjacent to the tumour [16]. It usually requires an operative procedure. The doses of radiation in tissues containing a tumour are relatively high compared to those received by the surrounding normal tissues. As a source, radioactive isotopes generated in atomic reactors or cyclotrons are used, for example, ^{60}Co, ^{137}Cs, iridium 192, and iodine 125 [16]. The source is usually "afterloaded" into an adaptable, flexible, custom-made applicator by remote control. The advantages of this technique lie in the possibility to check radiographically the position of the applicator. Also, medical personnel are not exposed to radiation unnecessarily, and sources can be withdrawn temporarily to a special safe storage if needed [16].

Large-Field Radiation Therapy

Total-body irradiation from an external source is used in treatment of lymphomas and leukemias and as a method of immunosuppression and eradication of bone marrow prior to the transplantation of donor bone marrow. It is also combined with chemotherapy for patients with high risk of metastases [16].

2.2.3 Radiation Treatment Procedure

In the sequel, by RT we mean *external beam* RT because it is the primary field of application of the 3D surface reconstruction system developed in this thesis.

Measuring Radiation Absorption

The precise measurement of the radiation dose is very important in the treatment of cancer because the dose has to be high enough to kill cancer cells effectively and, at the same time, harm to normal cells should be minimized. Besides, the prescribed dose depends on the depth of the tumour in a body because radiation is absorbed by tissues. Also, as energy increases, the penetrative ability of the radiation beam increases and the skin dose decreases [30].

The dose of radiation absorbed correlates directly with the energy of the beam. The basic unit of radiation is *gray* (*Gy*), which equals the amount of energy (*joules*) absorbed per unit mass (*kg*). In order to measure dose in a patient, one should first measure the ionization produced in air by a radiation beam. Then, the dose distribution can be corrected for the presence of soft tissue [30].

Radiation measurements in a water container make a good approximation for dose distribution in a patient's body. This approach is used not only to plan the radiation dose but also to calibrate x-ray machines weekly.

Pretreatment Imaging

To diagnose, and later to plan the treatment, accurate detailed images of the problematic area are required. Computed Tomography (CT) scans, bone or liver scans, arteriograms, Magnetic Resonance Imaging (MRI) scans can be used. The oncologist also checks what other organs or tissues are likely to be included in the field, for example, bowel, spinal cord, lungs, kidneys, skin, etc [9].

Simulation

After the oncologist analyzes the diagnostic images and the patient's condition, and RT is prescribed, a patient is sent for simulation. For this procedure, a patient is placed on a diagnostic unit that is identical geometrically and mechanically to an actual treatment machine. There is an x-ray source inside the *simulator* as well but of much lower energy as compared to the treatment machine. It is important that the patient's position is the same at simulation and each treatment procedure. After the position is set, *tattoos* or *ink markers* are put on the skin in the points where the red lasers, which go across the room, enter the body. These lasers are located at the same places in both simulation and treatment rooms. Sometimes, it may also be necessary to prepare a plastic mold or mask to immobilize the patient.

The simulator is rotated around the patient and a series of 2D orthogonal x-ray images are made from different positions. Having several images, the oncologist chooses the best angles of radiation entry taking into account organs and tissues which are in the field and the proximity of a target area to the surface of the body. The reason for having many beams (usually two to four but possibly up to seven to nine) is that in this way the dose to a tumour can be maximized while surrounding tissues get a lower dose of radiation (all the beams intersect in the area containing a tumour and this way the tumour gets the maximum dose). *Blocks* that will shield any normal structures are drawn on the films. These fields are sent to the mold room, where blocks are constructed [30]. If the machine possesses a multileaf collimator, the optimal relative position of the collimator "leaves" is planned with a computer.

Within the last decade, some cancer centres have combined CT and simulation into one procedure. During this procedure a patient is placed on the *CT-simulator* unit and undergoes a CT scan. Then, the physician has the capability of outlining the tumour and normal structures on each CT slice. A computer performs 3D transformation of the data and creates a *digitally reconstructed radiograph* (DRR). It can be manipulated to achieve better contrast and detail and viewed from any angle [30].

Treatment Planning

At the planning stage, the radiation dose for each beam has to be computed. The location of the tumour (or the area with cancer cells left after the surgery) is already known from diagnostic or simulation x-ray images. As has already been mentioned, the radiation is absorbed by tissues so the dose in the tumour depends on its depth in the body. The further the tumour from the surface, the less radiation reaches it. A medical physicist or dosimetrist needs the *digitized surface* around the area of treatment to be able to estimate with a computer an accurate dose of radiation for any chosen x-ray machine's position. For example, *Pinnacle* software can be used for dose and angles planning. The surface data can be provided from the MRI or CT results, or acquired by the contouring machine. These methods are discussed in more detail in Section 2.3.

A complete collection of data, including dose distribution in a body and beam profile information, is stored in the computer. Medical physicists can also assist the oncologist in deciding on the number of beams and angles of entry to a body. Several treatment plans can be generated, and the oncologist chooses the best one. Also, the oncologist plans how many fractions the whole radiation dose should be divided into. It depends on the size of the area and the sensitivity of the surrounding tissues. For example, bone is relatively insensitive to radiation and that is why the treatment to the small area near a bone can be given all in one fraction. Another example is breast cancer where treatment requires up to twenty or twenty-five fractions to maximize damage to the tumour while minimizing damage to the skin [9]. Breaks between the treatments are necessary for recovering of healthy tissues.

Treatment Delivery

All the data, such as beam-on time, beam angles, blocks, and wedges, are recorded in the patient's treatment chart and passed to the radiation therapist who will use this information together with skin marks to set up and treat the patient [30].

In many cases, radiation treatment is given in small fractions every working day. The whole course of treatment can take up to two and a half months. The treatment itself usually takes up to several minutes but taking into consideration the setting up of the machine and finding exact position of a patient, it may take from 15 minutes to half an hour.

Weekly *port films* are taken for each beam to ensure the correct placing of a patient, beams and blocks. Port films are images generated by the linear accelerator at energies of 6-20 MeV [30]. They have much poorer quality than the x-ray images from the simulator because of the high energies but are good enough for trained personnel to ensure the quality of setup and treatment.

2.3 Surface Reconstruction Methods for Radiation Therapy

There are several ways to get a digitized surface. To spot the disease, some patients undergo MRI or a CT scan. For these people, the 3D surface of the body can be computed from scan information given that the radiation treatment is planned and delivered right after the scan. On the contrary, if there was a long time interval between the scan and RT, for example because another type of treatment was given first, the anatomy may change and rebuilding of the 3D body's surface is necessary. Sometimes, the positions advantageous for treatment cannot be realized in the restricted volume of an MRI or CT machine. In this case, to avoid repositioning errors, the 3D surface has to be measured in the treatment pose. Also, some patients are diagnosed with 2D images, for example, x-rays. Therefore, constructing a digitized surface is necessary for a number of patients. A short overview of MRI, CT, and mechanical contouring as the most commonly used methods for surface reconstruction is given in Sections 2.3.1 – 2.3.3.

2.3.1 Computed Tomography

Computed Tomography or CT, also known as Computer Assisted Tomography or CAT, was introduced in the early 1970's by Hounsfield and Cormack [4]. It was the first imaging technology to allow 3D inner body images. In CT, *x-ray images* are used and processed on the computer to create virtual slices of the examined parts of the body.

Bones absorb more of the x-rays passing through them than do the surrounding tissues and that is why initially x-rays were used in medicine to image

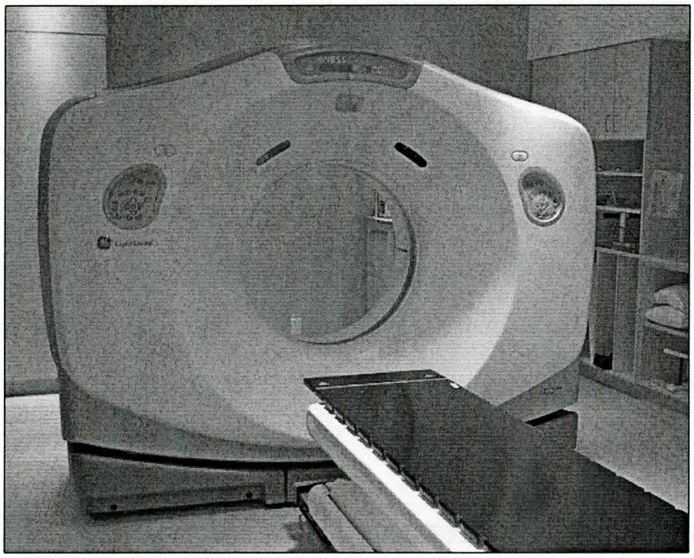


Figure 2.4: CT scanner

bones. Absorption of x-rays by other tissues and organs does not vary too much and they differ only slightly in the x-ray image. However, *contrast material* that can be swallowed or injected enhances the visibility of certain tissues and organs by outlining them clearly on the film. For example, swallowed contrast material helps to outline stomach or bowel while injected contrast can reveal arterial or renal structure [4]

In current machines, a fan beam of radiation sweeps through 360 degrees and detectors read radiation amount and degree [4] From multiple projections, it is possible to reconstruct a sectional display, or a slice. To build a 3D image of the examined body part, a number of slices are required, and this may be time-consuming. Having several slices, 3D image of the body part can be produced, and 3D surface of it can easily be extracted to use in RT procedures.

The recent National Survey of Selected Medical Imaging conducted by the Canadian Institute for Health Information shows that on January 1, 2003 the number of CT scanners in Canada was 326, which is 10.3 units per million population [4]. It is not time and cost efficient to use CT technology when only the surface is needed. The price of a CT scanner is about $1 million CAD [4]. Besides, since during a CT scan x-rays are utilized, it is better to cut down the number of scans to a necessary minimum because there is an increased lifetime cancer risk associated with CT [36]. A typical CT scanner is shown in Figure 2.4.

2.3.2 Magnetic Resonance Imaging

To create detailed images, Magnetic Resonance Imaging, or MRI, uses *hydrogen atoms* in the tissues, *a strong external magnet*, and *intermittent radio waves.* A patient is put in a strong magnetic field created by external magnet, and the magnetic moments of atoms in the body tend to align with the external field. A pulse of radio-frequency radiation is sent and it disturbs the alignment. When atoms return to their previous state they emit the energy that reveals their molecular environment and spatial location. For example, the nucleus of a hydrogen atom in a molecule of fat will emit a different signal than a hydrogen atom in the protein of muscle [4].

MRI can provide detailed images of all tissues except bone where the protons are tightly bound and less susceptible to magnetic influence. Received signals are processed by a computer and images are reconstructed similarly to CT. MRI images can also be enhanced by injecting *agents* analogous to the contrast materials in CT.

MRI does not use ionizing radiation, so in this sense it is less harmful

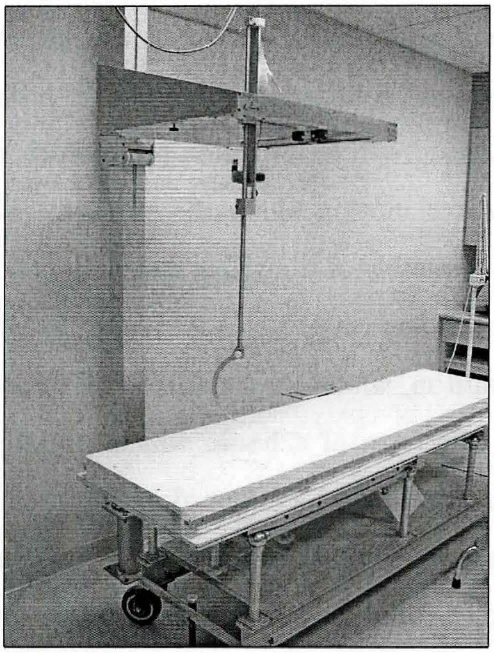


Figure 2.5: Contouring machine

than CT but there are drawbacks for this technology as well. First, the cost is high, the machine is twice as expensive as a CT scanner (over $2 million CAD), and there are fewer MRIs than CTs in Canada. The total number of MRIs in Canada is 147 according to the Recent National Survey of Selected Medical Imaging [4] Another inconvenience is noise produced during the scan. Some patients cannot have MRI, for example, those with pacemakers, those who have difficulty holding still for extended periods of time, and those susceptible to claustrophobia [4]

2.3.3 Mechanical Approach: Contouring Machine

The inconvenience of having MRI or CT scans for surface reconstruction of a patient's body, discussed in Sections 2.3.1 – 2.3.2, led to the utilization of the mechanical contouring approach. Many cancer centres have designed and successfully use mechanical contouring machines on a daily basis. Often, the machine works as a pantograph (see Section 3.1), which makes the mechanical replication of the pointer motion along the surface. The more complicated variant is the machine capable of digitizing the patient's surface directly. One of such contouring machines (see Figure 2.5) was invented at the Juravinski Cancer Centre in Hamilton, where we conducted all of our experiments. It has a metallic pointer in the form of a hook, which is easily rotated and moved in 3D by a technician. It has a local coordinate system and can measure in a $50 \times 50 \times 50$ cm^3 space. The 3D coordinates of a point are reported to the computer if the distance from the current point to the previous one is greater than 2 mm in 3D. Accuracy in an individual point reaches 0.5 mm. So, the accuracy, resolution and robustness of this device are good enough.

There are, however, some disadvantages associated with it. First of all, this is a contact method. It is unpleasant for a patient to feel the pointer on the skin. Besides, it can give some errors because skin is soft and can be easily pushed. Also, a patient can feel pain in the measured area. Another drawback is that the machine is manually operated, which is slow, and prone to errors because a patient can move slightly during the measurement. The contouring machine is separated from the treatment machine, which results in repositioning errors.

All these lead us to conclude that some non-contact, inexpensive, quick, accurate, on-site 3D surface reconstruction method is needed to assist in RT.

2.3.4 Methods Under Development

Recently, some techniques have been developed in research settings to efficiently build the surface for RT planning purposes and to use a reconstructed surface to evaluate and correct a patient's position.

Krempien, et al. [22] use a structured-light surface scanner to obtain a number of surface points. The scanner projects a set of lines varying in width, changing sequentially, and several images are needed from each viewpoint. To evaluate a patient's position, the obtained surface is compared to the previously scanned one and the volume of congruency is computed. An error below 1 mm is considered acceptable. In cases where set-up is not satisfactory a map of the surface comparison is evaluated showing the areas of missing alignment. According to this information a manual repositioning is performed. Authors report that no more than 3 repetitions are necessary to obtain an acceptable result. The whole procedure including registration, calculation and visualization takes about 20 sec for one repetition.

Lilley et al. [26] use a twin-fiber interferometer to produce stripes on the surface. The surface is reconstructed from a single image by applying Fourier fringe analysis techniques. The locations of the interferometer and the camera are fixed, therefore the field of view of the device is limited. Extensive calibration provides good accuracy, which is important in medical applications.

Some methods do not use surface reconstruction to control the patient's position. Johnson et al. [19] use two black-and-white CCD cameras, one mounted on the wall next to the left lateral lasers, and the other one on the ceiling, directly above the foot of the treatment table. Zoom lenses have to be utilized because of the distance between the cameras and a patient. Live subtraction images are obtained by subtracting a reference image (i.e., an image of the patient in the correct position) from real-time video. Subtraction images reveal patient misalignment in multiple views. They provide the therapists with immediate feedback and allow correcting a patient's position before radiation is delivered. Intratreatment movement of patients also is dramatic and can be easily detected. The authors report reducing of positioning error to 1 to 3 mm against 5 to 7 mm error of aligning a body using skin marks and room red lasers.

Berry and Aldrich [8] use the body's surface information to construct the compensators and to verify the patient's position from simulation to treatment and from one treatment to another. A series of narrow fan beams is projected to intersect the patient's surface. At simulation, the reference video image of the lines is stored. At treatment setup, video images are compared with the reference in real time. Operator adjustments to the patient's position are immediately shown on the composite image, and a congruency value is calculated to help in this process of alignment.

2.4 Problem Statement

The initial idea of this thesis project came from the medical physicists of the Juravinski Cancer Centre in Hamilton, Ontario who felt the need for an alternative to the existing methods for surface reconstruction in RT.

The choice of the method originates from the analysis of the existing methods. The cost of MRI and CT gives a constraint for the price of a future 3D surface scanner. The scanner should be inexpensive not only to manufacture but also to operate. The second important issue is to have a non-contact method to avoid patient's displeasure and decrease the errors coming from human interaction. This brings us to choosing *optical methods* discussed in Chapter 3. Since there are errors for all the methods in repositioning a patient to the treatment machine, it is desirable that the new device can be mounted on the machine. The machine rotates around the couch and the patient, and so will the device. Therefore, it should be simple, lightweight, preferably consisting of a minimum of parts. Another drawback inherent in all the earlier mentioned methods is slow data acquisition, thus the 3D surface scanner should do it quickly.

To conclude, the objective of the work presented in this thesis is to choose a method which is optimal taking into account all discussed demands and restrictions, to create a prototype device by using simple inexpensive hardware, and to design and implement algorithms for surface reconstruction. The developed 3D surface scanner has to be inexpensive, non-contact, portable (can be mounted on a treatment machine), quick and accurate.

Chapter 3

Optical Techniques for 3D Surface Measurement

In industry and art, there is a vast variety of applications where measuring the 3D surface of objects with a good accuracy is necessary. Applications include, to name a few, control systems in robotics, stress/strain and vibration measurement in materials, obstacle detection for vehicle guidance, 3D map building, inspection of industrial parts for quality, reverse engineering, computer assistance in animation for rapid development of characters and their behaviour, and finally, the important and currently very active research area of medical applications. Since each application has its own specifics, there is also a large number of methods and their combinations to solve each task. Here, we describe briefly the most popular ones.

3.1 Willème's Invention

The era of creating replicas of an object or a person began with the invention of photosculpture by François Willème (1830–1905) in 1860. He patented his invention and opened photosculpture studios in Paris, London and New York.

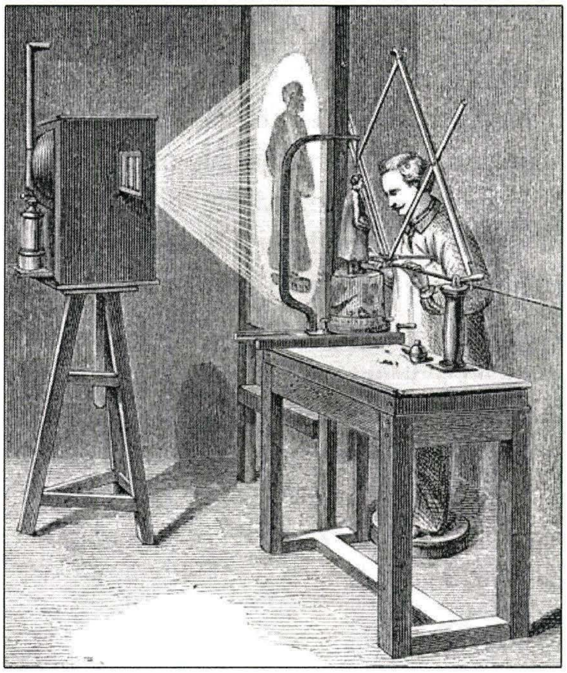


Figure 3.1: Photosculpture process

According to [2], *photosculpture* is a process in which, by means of a number of photographs simultaneously taken from different points of view on the same level, rough models of the figure or bust of a person or animal may be made with great expedition. An object was placed in a circular room and simultaneously photographed by 24 cameras placed equally around [32]. Each photographic plate was projected onto a screen using a magic lantern and an artisan would carve out 1/24 of a cylindrical portion of a clay figure by tracing the projected image with a pantograph. The *pantograph* is an instrument for copying a plane figure to a desired scale, consisting of styluses for tracing and copying mounted on four jointed rods in the form of a parallelogram with extended sides [3]. Figure 3.1 demonstrates an artisan carving out a figure using a projected im-

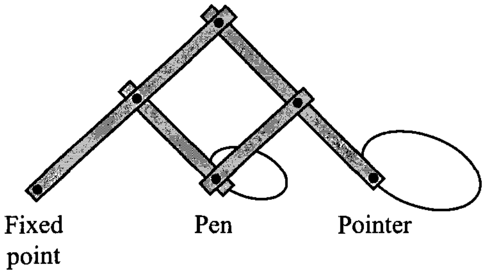


Figure 3.2: Pantograph

age and a pantograph. Figure 3.2 explains the principle of the pantograph's operation. A pointer is used to trace an original image, and a pen, or a cutting tool in the case of sculpture, produces the scaled copy of the drawing. With the pantograph in Figure 3.2 it is possible to draw a smaller copy of the image. The pen and the pointer can be switched to produce a larger copy.

Willème's studios stayed in operation until 1867 [5], when it was realized that the photosculpture process was not more economical than traditional sculpture. Since only 24 profiles were supplied, a professional sculptor was needed to finish all the fine details. Besides, the process required quite an investment in terms of cameras, projection and reproduction systems, and skilled labour to operate them. The problem of automatic 3D surface reconstruction regained interest only with the invention of the computer.

With time, many new devices, techniques and algorithms have been invented, nevertheless 3D surface reconstruction remains important and needs further improvement and investigation. Various methods differ by their application area, resolution, accuracy, restrictions, level of sophistication, equipment, size of examined objects, sensitivity to errors, ability to measure dynamic scenes, etc.

3.2 Time-of-Flight Method

Originally, this distance measuring technique was used in military and surveying applications. The *pulsed time-of-flight* method is based on the direct measurement of the time of flight of a laser or other light source pulse. The pulse is reflected back from an examined surface to a sensor. A reference pulse is transmitted at the same time as the main one. It is passed through an optical fiber with known parameters and then received by the sensor, which allows us to compute the starting time easily. The time difference between the two pulses is converted to a distance. A typical resolution for this method is around a millimeter. To obtain 1 mm accuracy, the accuracy of the time interval measurement should be 6.7 ps [5]. Diode lasers with subpicosecond pulses and high-resolution electronics provide submillimeter resolution. Similar techniques can give up to several μm resolution [10].

For short-range applications, this technique has the disadvantage of requiring the measurement of extremely short times for good range accuracy. Another variant of the method is the *continuous wave time-of-flight* method, which uses a continuously modulated laser beam to indirectly measure time of flight. The advantage of the amplitude modulation approach is the possibility of tailoring the modulation wavelength to the range to be measured. With different wavelengths, long range and high resolution measurements can be achieved. Systems with frequency modulation can even resolve multiple targets, which is important in scenes where the main target is obscured by objects in front of it [15].

In addition to the need to use lasers, detectors and optics, the main disadvantage of the time-of-flight methods is that they provide one surface point measurement at a time. Therefore, to cover the whole scene, measurements on two axes have to be performed, which increases the cost of the system and reduces the pool of its commercial applications.

3.3 Laser Scanning

Laser scanning is used for ultra-precise measurements to create 3D models and drawings, for complex geometries and free-form surfaces. Laser scanned images are widely used in virtual reality simulators and in reverse engineering applications. Equipment based on this method gained a broad recognition in many famous museums and art galleries because it gave the possibility of producing precise full-colour models of valuable antiquities and pieces of art, many of which may demand special storing conditions. With the 3D models, many people can study historical data and see the presentations of collections without risk of damage to the originals.

The technique involves projecting a point or a line of laser light onto a surface and viewing it from an offset camera. Advantages of using laser light instead of a conventional incoherent light source are brightness and spatial coherence that enables the laser beam to "stay in focus" when projected on the scene [5]. Alterations of the line shape or point position correspond to the topography of the object. Triangulation is used to compute the 3D position of an individual point. *Triangulation* is a geometrical technique to determine a position of a point, which is viewed from two well-separated points, where their location and the angles between the line joining them and the lines to the distant point are known. The drawback of the laser scanning is similar to that of the time-of-flight method: it is slow and costly due to the necessity of

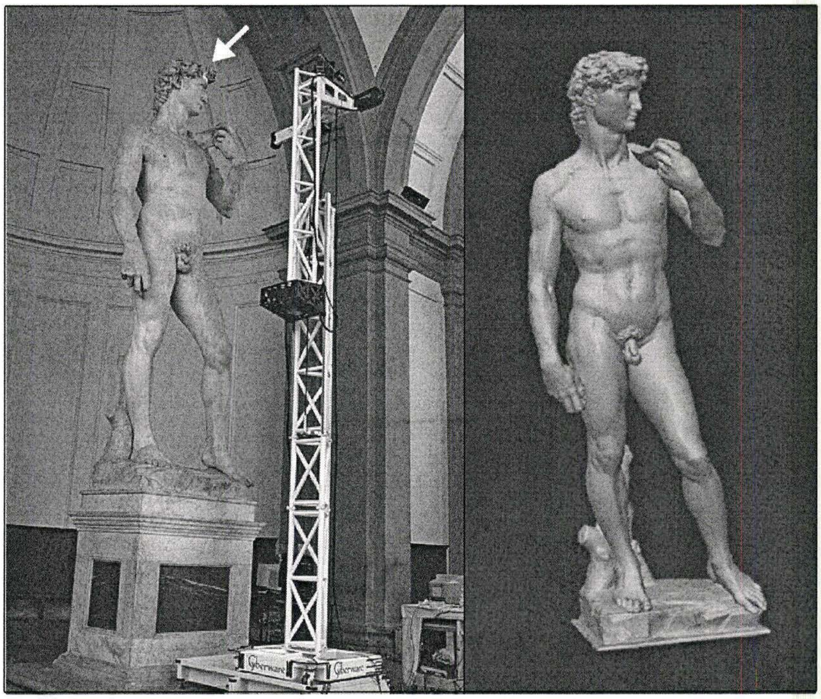


Figure 3.3: Laser scanning

equipment, which moves the light to cover the whole object scanned, or the device has to be moved manually. The typical measurement range is ±5 to ±250 mm with the accuracy of about 0.01% of the distance to the measured object.

Figure 3.3 (left) demonstrates the laser scanner gantry positioned in front of Michelangelo's David. The white arrow shows a laser line on the forehead of the statue. Figure 3.3 (right) shows the reconstructed model of David. It was acquired over a period of 4 weeks by a crew of 22 people scanning 16 hours per day 7 days a week [24].

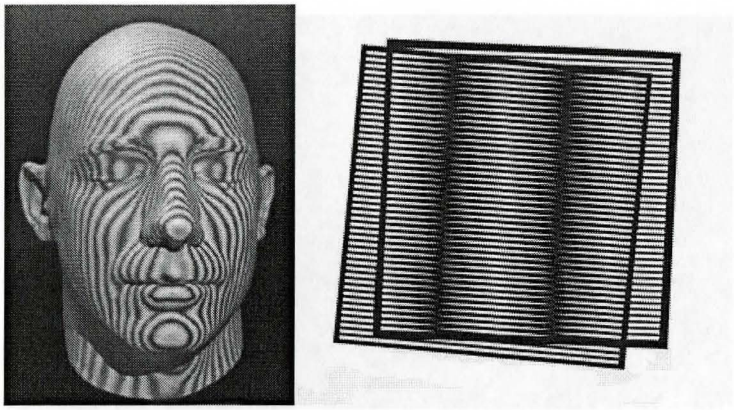


Figure 3.4: Moiré method

3.4 Moiré

The word *moiré* comes from the name of silk fabric with a wavy surface pattern. The first paper describing moiré interference was published in 1874 by Lord Rayleigh who predicted that the effect can be used as the test of irregularity of gratings or deflection of one grating to second. Now it is also applied to inspection systems for locating missing or misaligned parts in a multicomponent assemblies, analysis of dynamic processes (vibrations), and optical contour extraction of surfaces [6] In moiré method two gratings are used, one is a master grating and the other is a reference grating, from superposing of which contour fringes can be generated, photographed by a camera and analyzed. For surface reconstruction two initially identical gratings are often utilized. Master grating follows the deformation of the object, and a reference grating remains unchanged. Increased resolution is realized since the gratings themselves do not need to be resolved by a CCD camera, except for the *logic-moiré* method, where the reference grating is computer generated. The price of high resolu-

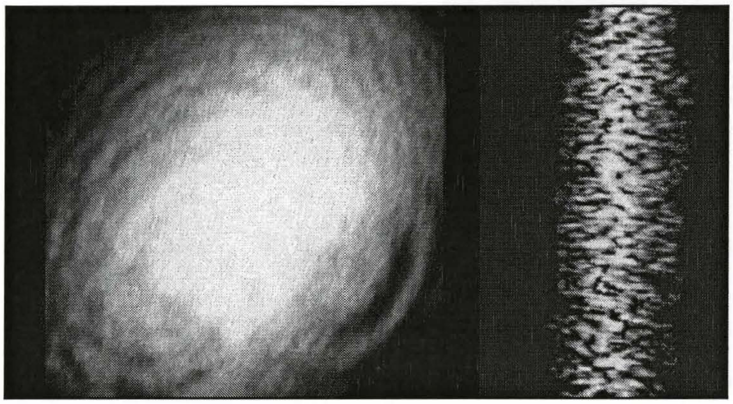


Figure 3.5: Laser speckles

tion is implementation complexity and the need for a high power light source as compared to a structured light technique (described in the sequel) To increase the accuracy of reconstruction, one might want to use multiple images and several moiré fringe patterns with different phase shifts. The typical measurement range of the phase shifting moiré method is from 1 mm to 0.5 m with the resolution at 1/10 to 1/100 of a fringe [10]

Moiré methods are commonly applied to 3D surface reconstruction tasks. Figure 3.4 shows moiré fringes superimposed on model head (left) and superimposed gratings (right) [1]

3.5 Laser Speckle Pattern Sectioning

Laser light is very coherent in nature. *Speckle* (see Figure 3.5) is a mottled pattern which arises when the light reaches a non-specular reflecting surface. It scatters, and the waves collide. They can cancel each other out if their phases are opposite, and then a dark space occurs, or a light space otherwise.

For some kinds of lasers this effect is easily visible. Laser speckle pattern can also be called an interference pattern.

The 3D Fourier transform relationship between optical wavelength (frequency) space and the distance (range) space is used to measure the shape of an object. This approach utilizes the principle that the optical field in the detection plane corresponds to a 2D slice of the object's 3D Fourier transform. To get information for other 2D slices, the wavelength of the laser has to be changed. Images with different wavelengths are photographed by a digital camera, and then added up to form a 3D array, to which 3D Fourier transform is applied in order to get the object's shape [10].

The measurement range can be from a micrometer to a few meters, which is very flexible. This method can give very precise results, for instance, in the measurement range of 10 mm the resolution of 1 to 10 μm is achievable. At the same time, if a relatively large object has to be measured, it takes more time to acquire the images with the different wavelengths [10], and changing wavelength demands some special equipment.

3.6 Interferometry

A surface with an interferometric pattern looks similar to one with a moiré pattern but the nature of the phenomenon and a measurement range are different. *Interference* can be observed when two (or more) waves come together. The effect can be used for measuring purposes. The basic principle of an interferometer lies in splitting off the light coming from one source into two parts by placing a system of mirrors to provide two coherent sources. Half of the light is directed to a measured object, and then, after being reflected, combined with the other half. The waves interfere. At some places they add up (constructive interference), at other places cancel each other (destructive interference), which produces an interference pattern – alternation of light and dark stripes – in a photographed image. The shape and thickness of stripes change follows the topology of the object.

Interferometric methods can provide very precise measurement of the surface change, the resolution for most of them depending on the light wavelength. One of the methods, belonging to this group, is double heterodyne interferometry, which uses frequency shift. It achieves 0.1 mm resolution with 100 m range. Combined with phase shifting analysis, interferometric methods and heterodyne techniques can have accuracies of 1/100 and 1/1000 of a fringe, respectively. With certain optical configuration design, accuracy can reach 1/10,000 of a fringe [10].

The disadvantage of the method comes from its sensitivity – a highly stable environment is a necessity, which is usually provided by expensive equipment. The other complex issue is providing coherent sources. Interferometry can be used for very precise measurements, for example, it can measure the structure of skin.

3.7 Shape-From Techniques

Shape-from techniques are a group of monocular approaches that recover relative depth of the surface from *texture*, *shading*, *contours*, *motion*, etc. To measure a surface, several images are acquired. Since the reconstruction error depends on the number and choice of viewpoints, some techniques has been developed to automatically control the choice of viewpoints [28].

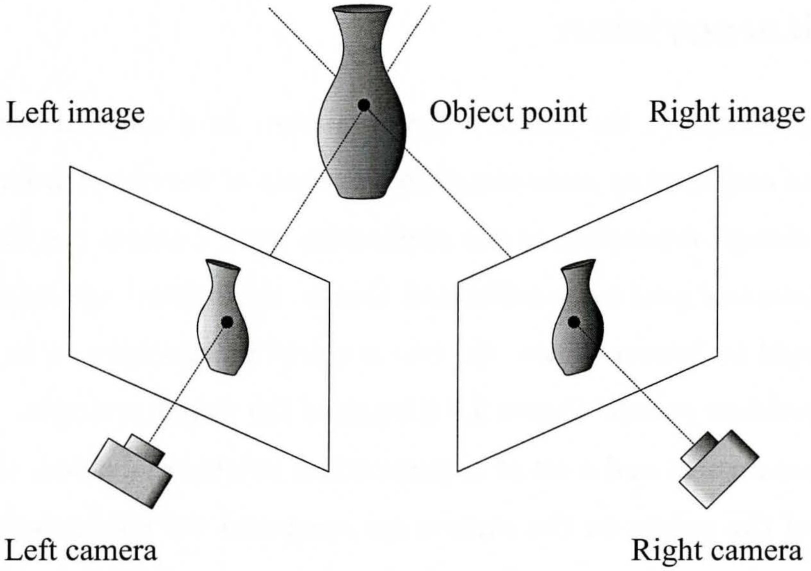


Figure 3.6: Stereo principle

For example, to get a shape from *silhouettes*, one should first separate the object from the background on all the images. In controlled environments, this can be done by using a background of uniform colour Then, a bounding pyramid is constructed using the focal point of the camera and the silhouette points. Finally, all the pyramids, built from different points of view, are intersected and form an approximation of the bounding volume [28] Accuracy and resolution depend mostly on the number and position of viewpoints. This method is not very practical because of the necessity of the special background and a large number of viewpoints for a good resolution.

3.8 Stereovision

Stereovision simulates the human's vision system. In a classical version, two cameras are employed to make simultaneous shots of the object from different positions, though depending on the application more cameras can be utilized. First, the cameras need to be calibrated, that is, the exterior and interior parameters should be known. Then, the two acquired images have to be analyzed for correspondent points. Figure 3.6 illustrates the stereo principle. After the camera's parameters and a set of corresponding points are defined, the 3D coordinates of the points on the surface are computed via triangulation as the intersection of the lines going through the optical centres of the cameras and the corresponding points on the image planes. The term *triangulation* actually refers to the fact that there is a triangle created by the three vectors: the vector between the cameras' optical centres, and the two vectors connecting the cameras' optical centres and the surface point. The 3D coordinates of the surface point can be found from the relation between the three vectors. The necessary equations are provided in Section 5.4.

Computation of the 3D coordinates of the surface points is easy but the difficulty of the stereovision technique is in finding corresponding points in the two images. Some points are visible only in one picture. 3D coordinates for those cannot be found precisely, only approximate coordinates relying on the neighbouring points and shading can be computed. The accuracy of the measurement depends on the quality and accuracy of the camera and calibration techniques.

However, the problem of finding the corresponding point in the second image for a certain point from the first image may be reduced to the problem of

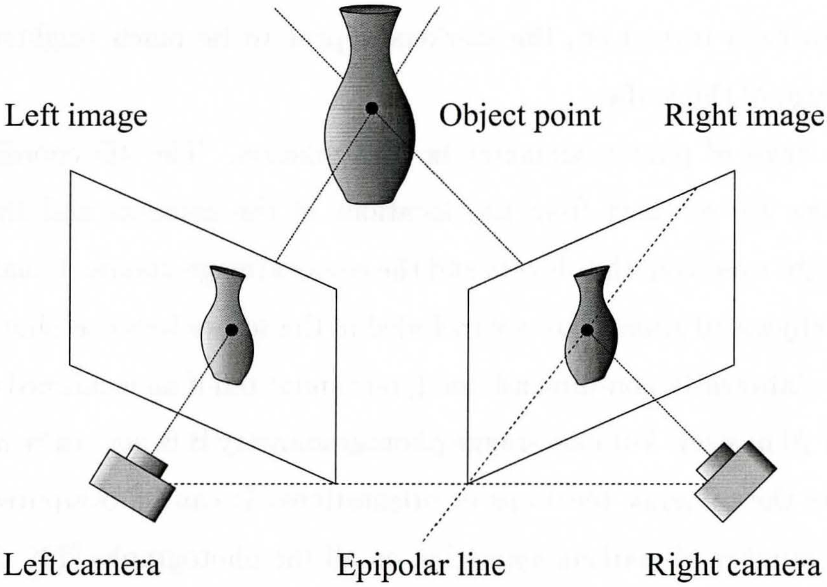


Figure 3.7 Epipolar line

finding the corresponding point on a line. With the precise setup of equipment, the corresponding point can only be located on the *epipolar line*, which is the intersection line of the second image plane with the plane going through the two cameras' optical centres and the point from the first image (see Figure 3.7)

3.9 Photogrammetry

Originally, the photogrammetry approach was applied to make aerial surveys and maps. More recently, it has been adopted for close-range measurements. Photogrammetry can employ either stereo technique (most frequently), or defocus, shading and scaling [10] This method is good for feature type 3D measurement and determining the orientation of the object because it uses some markers put on the surface such as retroreflective painted dots or LEDs. With the camera flash turned on, the markers appear to be much brighter than a diffuse image of the surface.

The basis of photogrammetry is triangulation. The 3D coordinates of the markers are revealed from the locations of the cameras and the angles between light rays from the objects and the camera image planes. Usually, some reference objects of known size are included in the scenes because photographic images are inherently non-dimensional (you cannot tell if an examined object is of 10 m or 20 m size). For close-range photogrammetry it is not really necessary to measure the cameras' locations or orientations. It can be computed from a minimum number of markers appearing on all the photographs [29]. After the correspondence between the markers from different images is established, the *bundle adjustment algorithm* based on a least-squares procedure is applied to compute space locations. The accuracy achieved by this method can be as high as 0.001% or even 0.001% of a distance to the object [10].

In [11], human faces are measured by means of multi-image photogrammetry. Five CCD cameras are used. To facilitate the establishment of correspondences in the images, texture in the form of random patterns is projected from two directions onto the face. The multi-image matching process, based on a geometrical constrained least squares matching algorithm, produces a dense set of corresponding points in the five images. An accuracy of 0.1 – 0.2 mm can be achieved. The described method is quick and accurate but it demands resources in the form of random pattern projectors and sensitive cameras.

3.10 Structured Light

Stereovision is an attractive method for 3D surface measurement but it has a serious drawback: the correspondence problem. To improve the situation, a method based on the structured light concept, or *active triangulation*, can be used. For structured light, a second stereo camera is replaced by a light source, which projects a known pattern of light on the surface. A single camera photographs the illuminated scene. The required 3D information can be obtained by analyzing the deformation of the observed pattern with respect to the projected one [7]. The structured-light method includes both *projected coded light* and *sinusoidal fringe techniques.* Salvi at al. [35] have classified the known patterns for structured light.

Figure 3.8 demonstrates the geometry of the structured-light technique. The angle γ between projected (line a) and reflected (line b) light rays should not be too small, otherwise the deformations of the projected pattern on the surface are negligible. The camera's optical centre and the image plane are computed from the parameters of the camera. If, for a point in the image, a corresponding point from the pattern plane among all the projected points can be identified, the point on the surface is the intersection of the two lines: one going through the light source and the pattern plane point, and the second going through the image plane point and the camera's optical centre (lines a and b in Figure 3.8 respectively). When the spatial coordinates for surface points are computed, the surface can be reconstructed. With many patterns, a single image can be enough to reconstruct the whole examined surface [35]. However, for surfaces with a lot of curvature, some parts of the surface can shade others from either the camera or the projector. In this case, several

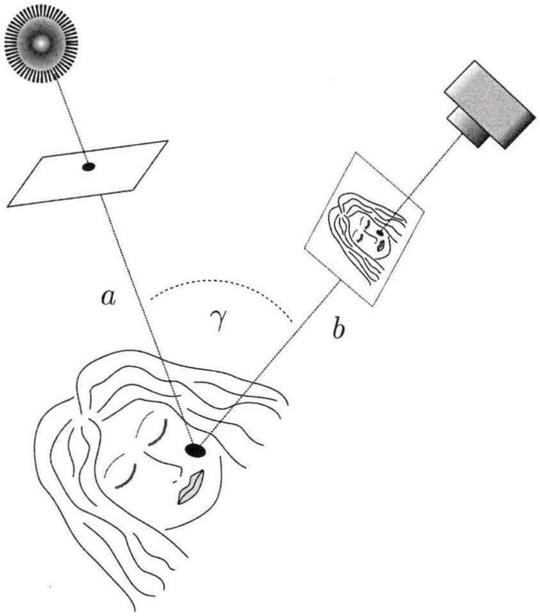


Figure 3.8: Structured light

images from different viewpoints are needed, and the reconstructed surfaces have to be combined into one final surface. Multiple images can be used to increase accuracy and resolution.

Fringe techniques are very popular in 3D surface reconstruction. For example, Lilley et al. [26] use an interferometer to produce fringe pattern and an extensive calibration to conduct good accuracy measurement of a patient's body for controlling the position in the x-ray treatment machine. The alternative is to use a number of distinct shapes as a light pattern, such as a set of stripes, grids, rectangles, concentric circles, circular spots, etc. More details are provided in Section 4.2.

3.11 Conclusion and Motivation of Choice

In this chapter, a brief review for the most popular optical 3D surface measurement methods was provided. We decided to use a structured-light technique for the 3D reconstruction of the patient's body for the following reasons:

1. All the light rays are in the visible spectrum range, which is not harmful to human health.

2. Structured light does not demand expensive equipment, only a camera and a projector have to be provided.

3. Because of the minimal number of parts and simple implementation, the 3D surface scanner for application in radiation planning and treatment based on a structured-light technique is light-weight and can be mounted on the x-ray machine's head. Therefore, the necessary surface part can be observed without intrusion into the current workflow.

4. Structured-light techniques provide fast full field measurement. A single image can be enough to reconstruct the surface though multiple images can be used to improve accuracy and resolution, and also to cover the surface not visible from one viewpoint.

5. Due to the vast variety of possible patterns, the method is very flexible. The choice of the existing pattern or the design of the new one depends on the application, available equipment and software, and properties of the surface. After locating pattern elements, 3D coordinates for the surface points can be computed. Also, a pattern can be constructed so as to

provide some other useful information about the examined surface, such as the slope of the surface in the number of surface points.

Chapter 4

Experimental Design

In this chapter technical and optical aspects of the 3D surface scanner are discussed in Section 4.1. Then, Section 4.2 provides description of the projected pattern and motivation for this choice. Finally, the mathematical model for a pattern element is presented.

4.1 Technical Setup

This section describes design constraints, equipment and its disposition.

4.1.1 Challenges and Design Constraints

Figure 4.1 shows an x-ray machine, which rotates around the patient couch. The couch can be moved up and down, back and forth, and rotate so that the centre of rotation of the machine appears inside the target area in the patient's body. As mentioned in Chapters 1 and 2, we wanted to construct such a device that a patient's surface is measured in the same position, place and time as treatment planning and delivery. It brings a number of constraints to the 3D surface scanner design:

1. Because of the treatment machine's rotation and couch movement, it is not possible to put the equipment in a fixed location close to the patient. With some additional parts, such as zoom lenses, we could mount it at some distance but the price is one of the important constraints too. Also, in the case of distant location, for some positions of the x-ray machine the surface can be shaded from either the camera or the projector by the machine's head. Taking all these into account, we decided to mount the device onto the machine's head (between the head and the patient couch).

2. Mounting the 3D surface scanner on the x-ray machine puts constraints on its size, weight, and number of parts. To satisfy the demands, we have chosen small and lightweight camera, and used a light source located inside the x-ray machine's head to project the pattern through the pattern mask mounted on the machine's head as well.

3. In order to use triangulation equations to compute the spatial positions of the surface points (see Section 5.4), the camera's and the projector's axes have to intersect at the centre of rotation of the x-ray machine.

4. Since all the equipment is mounted on the machine, the current angle of rotation should be known. Shape of the surface can be inferred from the relative position and orientation of the light, the camera and the pattern but surface orientation cannot be obtained without the angle of the machine.

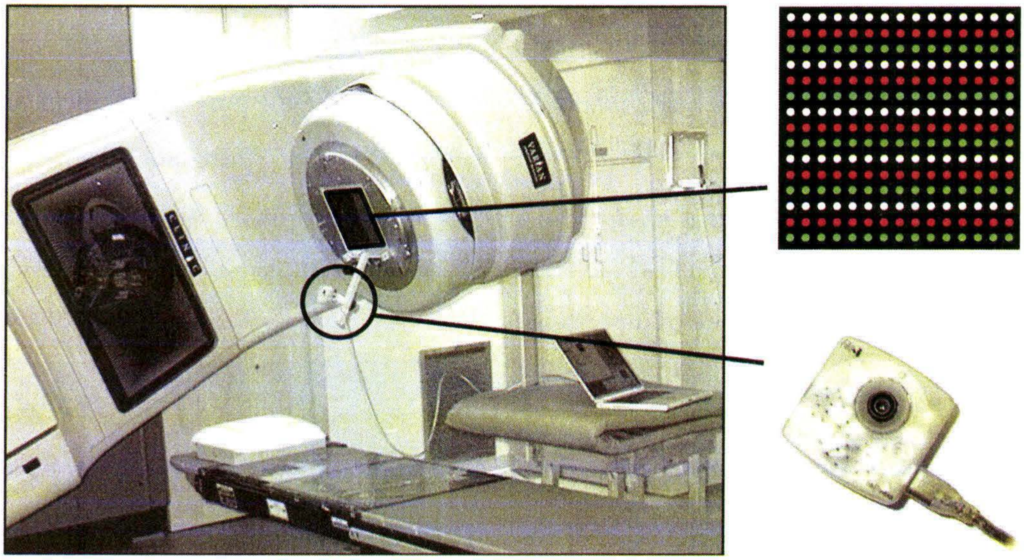


Figure 4.1: Technical setup

4.1.2 Surface Reconstruction Equipment

For the structured light technique, it is important to know the position and orientation of the projector and the camera. We use a light source inside the machine's head, and mount the camera and the pattern firmly to the head, so that the position of the light source and the camera can always be computed from the current x-ray machine's angle. There is no way to control the speed of the rotation of the treatment machine, it is not stable because it depends on how much the button on the remote is pressed. Therefore, the position of the projector has to be inferred either from the calibration of the system (fiducial marks in the scene, an inclinometer attached to the machine's head, etc.) or from the initial setup. For the experiments we chose the second option, namely, taking the images from predefined angles of rotation because the focus of this thesis is not in the calibration techniques but rather in the development of the algorithm capable of analyzing deformed pattern in the low-quality images. More details on low quality resulted from the low-light environment and the camera's optical parameters are provided in Section 4.2.

Equipment Specifications

The 3D surface scanner prototype was mounted on the VARIAN's CLINAC 2100 EX x-ray treatment machine (see Figure 4.1). All the experiments were run in the MATLAB 7.0.1 environment on a PC with Pentium 4 CPU 2.8 GHz, 1 GB RAM, Microsoft Windows XP. The projector was represented by a light source and a pattern mask. We used inexpensive equipment:

1. *Light source:* incoherent light source (as opposed to a laser source) built into the x-ray machine's head. It is more focused than an ordinary light bulb but less focused and less powerful than a laser. The light field is of the same size as the radiation beam, so the radiation therapist can see the area covered by x-rays without turning the radiation source on.

2. *Pattern mask:* a transparency with rows of white, red, and green transparent circular spots and an opaque black background. Spots are located on an invisible regular rectangular grid. The motivation for this choice is given in Section 4.2.

3. *Camera:* Unibrain Fire-i digital video camera (about $120 CAD) with the following characteristics:

Interface	FireWire 400 Mbps, 2 ports
Sensor Type	Sony^{TM} Wfine* 1/4 CCD Color, progressive, square pixels
Resolution	VGA 640 × 480
Optics	Built-in f 4.65 mm lens, anti-reflective coating
Video Modes	YUV, RGB, Monochrome
Frame rates	30, 15, 7.5 and 3.75 frames per second
Dimensions (W×H×D)	62 × 62 × 35 mm
Housing and Weight	Semi-transparent polymer, 60 g

We take images in RGB mode with 24-bit colour coding.

Geometry

To solve the stated surface reconstruction problem, we need a special setup. The light source moves in the plane of rotation of the x-ray source. We mount the camera offset from the light source in the same plane (see Figure 4.1). The pattern mask is aligned so that the rows of spots are parallel to the plane of rotation. In this way the deformed pattern in the images has an exploitable structure that will be clear from the discussion in Section 4.2 and Chapter 5. A single image is enough together with calibration parameters to reconstruct the surface of the visible part of a body. Here "visible" means that the part where the pattern is projected on the body is visible for the camera. However, to reconstruct the surface not visible from one point of view, and also to improve the accuracy and resolution, several images from different viewpoints may be taken.

4.2 Choice of the Projected Pattern

In this section we provide an explanation of the pattern designed for our 3D scanner prototype and describe the mathematical model for the pattern elements.

4.2.1 Low-Light Environment

With the structured light technique for body's surface reconstruction, the best choice is to use a light pattern on a dark background. In this way, body features, such as difference in skin colour within the examined area, freckles, scars, eye-brows, etc. are not visible. Only the light pattern elements should be distinguishable, which makes the image easier to analyze. Therefore, all the light except for projector light should be off during measurements. The x-ray machine's light source is neither very strong nor very focused, therefore the light pattern on the surface is dull and the images have low contrast.

At the same time, the inexpensive camera that we use (see Section 4.1.2) is considered to be the best home and office FireWire camera available in the market, and it is, indeed, relatively good for ordinary use, but it is not designed to serve in low-light conditions: its sensitivity is low. This is why *the level of noise is considerable.* Sometimes, it is hard to say if the observed bright pixels belong to the pattern element or it is noise.

4.2.2 Previous Work

Salvi at al. [35] have classified the known patterns for structured light. The choice of pattern depends on the camera and light characteristics, expected surface properties, and the level of image processing software available. It is important in RT procedures to acquire the data quickly, therefore we cannot use patterns that change over time or move and imply taking *multiple images*, though this technology can provide good accuracy and resolution even with consumer hardware [33]. Coding the pattern elements with different *colours* simplifies the correspondence problem. High precision can be reached with pseudo-randomly colour-encoded patterns [23]. However, because of the camera's characteristics, low light and skin scattering properties, we cannot use many colours. Dark colours disappear in the image and similar hues are not distinguishable well, and they can be misclassified easily.

In many cases, to reconstruct the surface, a *fringe pattern* projected by an interferometer is used [26]. Many variations to increase robustness in a particular setting have been developed. The technique for converting the fringe frequency information into spatial coordinates is well studied. With this kind of pattern, problems of indexing the lines on curved surfaces and of phase unwrapping arise. *Phase unwrapping* is the reconstruction of a function on a grid given its values modulo 2π. The alternative is to use a number of distinct shapes as a light pattern, such as a set of *stripes* [33], *grids* [23], *rectangles* [17], *concentric circles* [14], *circular spots* [12, 25], etc.

Davies and Nixon [12] project yellow, cyan, and magenta hexagonally tessellated (tiled) circular spots onto a face painted white using theatrical makeup. Each spot's position in the array is defined by its colour and its immediate neighbours. From the shape and position of the observed ellipses, the location and orientation of the 3D surface at video-rate is reconstructed with the use of geometric constraints.

Lewin et al. [25] have constructed a camera capable of recording 3D images of objects. It does this by projecting thousands of spots onto an object and then measuring the range to each spot by determining the parallax from a single image. The spots are organized in a tilted periodic fashion. The camera reconstructs the surface with an accuracy of less than 1 mm at 1 m distance.

4.2.3 Modelling Spots

With a low-sensitivity camera and an incoherent low-power light source we get noisy images, and the fact that human skin diffuses and absorbs light further reduces contrast. To locate pattern elements in such images imposes restrictions on the method:

1. complicated shapes cannot be used as pattern elements, and
2. the searching procedure has to be provided with a good estimate of where the pattern elements are to be robust and efficient.

Under low-light conditions the image processing techniques that can be applied to the data are limited, for example, edge detection will fail on such noisy and diffuse images. The shapes of the projected pattern elements should take into account these problems and be robust enough for this environment.

We chose *circular spots arranged on a regular rectangular grid* as a pattern (see Figures 4.1 and 4.2). One of the advantages of this pattern is the *discreteness* of pattern elements in both directions (as opposed to line elements). It allows partitioning the problem of recognizing pattern elements and assigning them to the grid into discrete small subproblems. It simplifies the problem, and in this way it is easier to differentiate between good and erroneous data. Elliptical spot shape also gives information about the surface orientation in the area

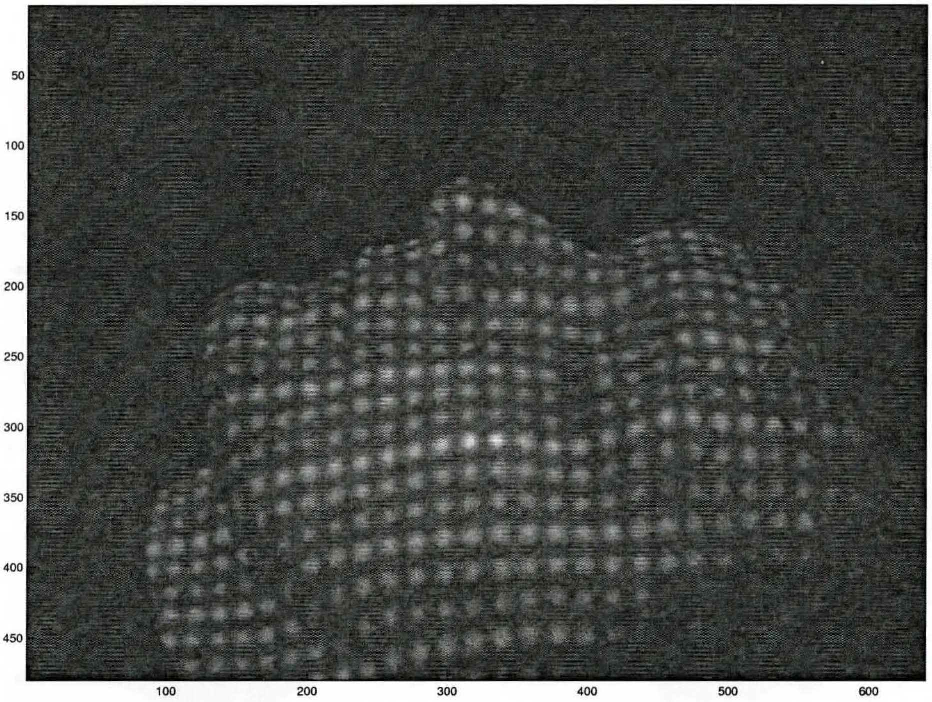
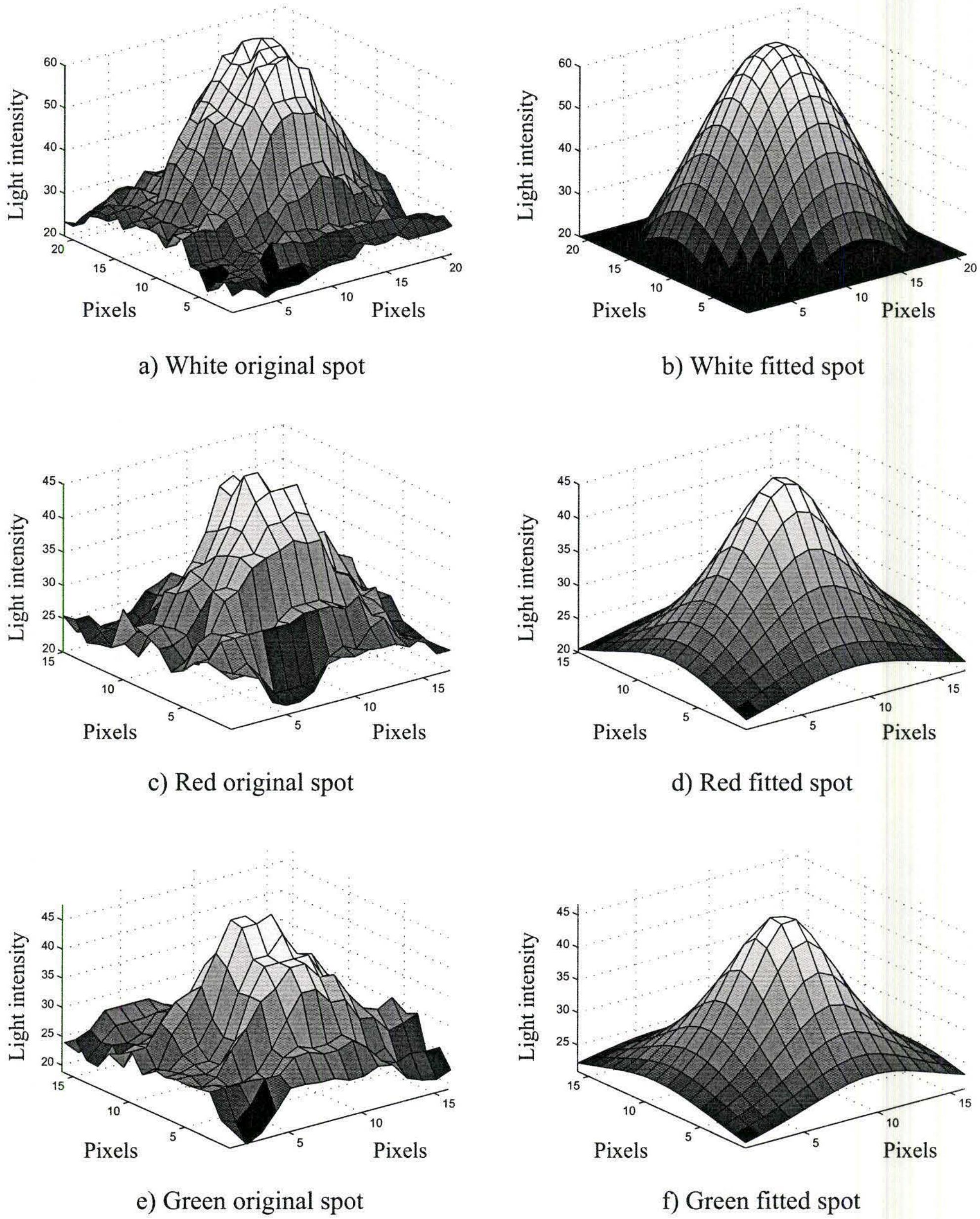


Figure 4.2: Circular spots projected on the patient's face (in pixels)

containing the spot [12] This information is useful at the surface interpolation stage.

When the circular spot strikes a small surface area, which is approximated by a plane, it becomes an *ellipse.* The spot does not have a sharp edge, because the incident light is not parallel, the skin diffuses light, and the charge-coupled device (CCD) introduces noise.

The light intensity of a sample of white, red, and green spots is shown in Figures 4.3a, 4.3c, and 4.3e. We model the spot's cross-sectional intensity with



a) White original spot

b) White fitted spot

c) Red original spot

d) Red fitted spot

e) Green original spot

f) Green fitted spot

Figure 4.3: Original light intensity data and fitted models

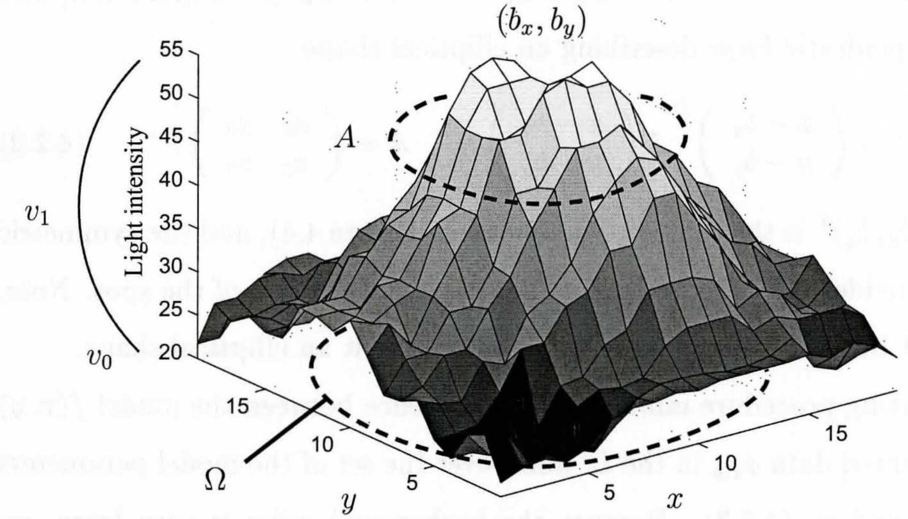


Figure 4.4: Parameters of the spot model

an *inverse polynomial function.* It has only seven parameters (see (4.2.1) and (4.2.2)) and was found empirically to fit the spot's shape well. To estimate the model parameters, we iteratively minimize the difference between the model and the data in a least-squares sense. Then, we discard spots whose estimated parameters don't fit a predetermined profile, e.g., non-positive-definiteness, as explained below

To model the light intensity of the spot observed in the image (see Figure 4.4) we use the inverse polynomial model

$$f(x, y) = v_0 + \frac{v_1}{1 + s(x, y)} , \tag{4.2.1}$$

where (x, y) are the coordinates of a pixel in the image, v_0 is the asymptote of the function (from below), v_1 is the height of the bump (see Figure 4.4), and $s(x, y)$ is a quadratic form describing an elliptical shape

$$s(x,y) = \begin{pmatrix} x - b_x \\ y - b_y \end{pmatrix}^T A \begin{pmatrix} x - b_x \\ y - b_y \end{pmatrix}, \qquad A = \begin{pmatrix} a_1 & a_2 \\ a_2 & a_3 \end{pmatrix}, \tag{4.2.2}$$

where $b = (b_x, b_y)^T$ is the centre of the spot (see Figure 4.4), and the symmetric matrix A provides the parameters for the shape and the size of the spot. Note, that A must be *positive definite* in order to represent an elliptical shape.

The fitting procedure minimizes the difference between the model $f(x, y)$ and the acquired data $\phi_{x,y}$ in the L^2 norm over the set of the model parameters - A, b, v_0, and v_1 (4.2.3). Because the background noise is very large, we found the quality of the estimates were significantly degraded by including background pixels. Consequently, minimization runs only for the pixels inside a circular area Ω of the size of an average spot:

$$\min_{A,b,v_0,v_1} \sum_{\Omega} (\phi_{x,y} - f(x,y))^2. \tag{4.2.3}$$

Figure 4.3 illustrates the original light intensity data (a, c, e) and the the maximum-likelihood inverse polynomial model (b, d, f) for a white, red, and green spots.

4.2.4 Colour coding

Note, that with the geometry described earlier (light and camera are in the same plane of rotation, with rows of the spots parallel to this plane) independent of the angle of the machine's head the vertical columns of the illuminated elliptical spots in the image will always appear as almost straight and vertical columns

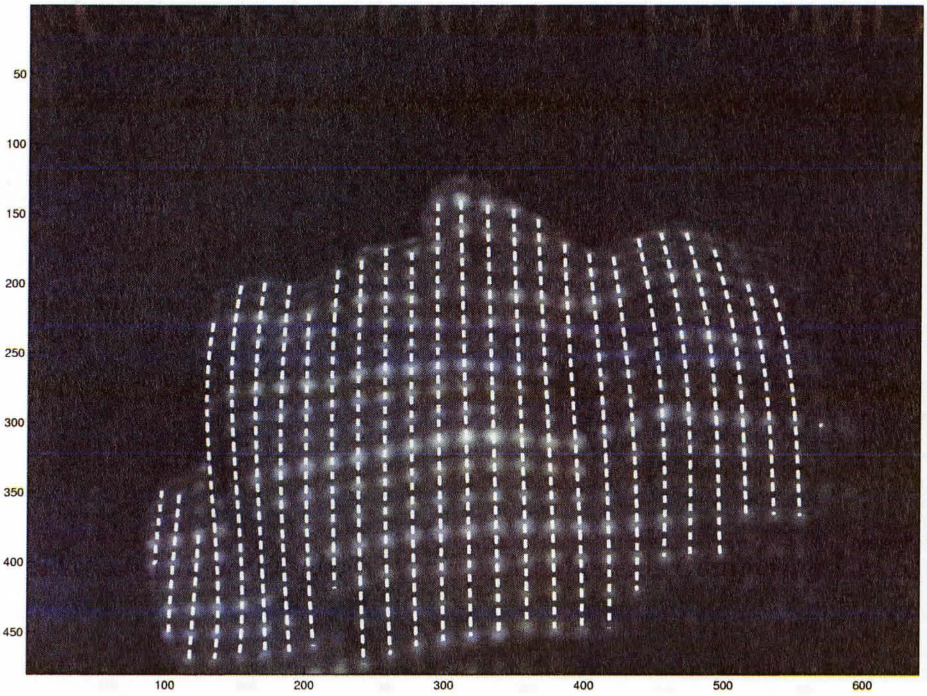


Figure 4.5: Columns of spots (in pixels)

(see Figure 4.5), especially in the central part of the image. This can be used as a good guess for the location of the spots because if we find columns the problem of locating the spots in the image reduces to a simpler problem of locating them along the columns. This not only helps to recognize the spots more easily but also solves half of the problem of assigning identified spots to the grid by arranging them in the columns.

However, closer to the sides of the image, the period changes slightly and the columns incline more to follow the surface and because of the perspective.

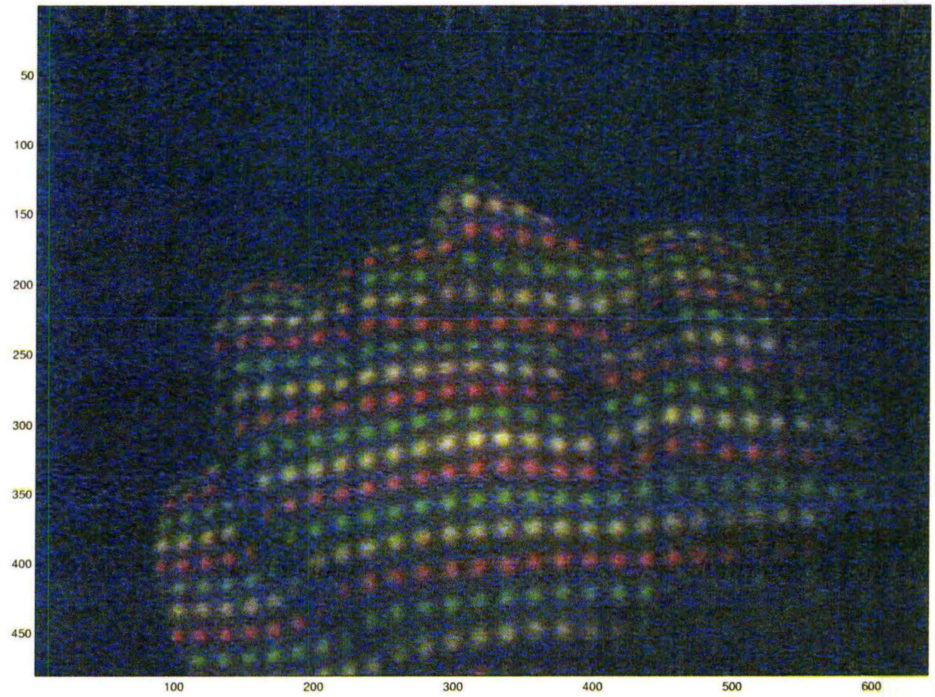


Figure 4.6: Colour coding (in pixels)

We start from the centre assuming that the central most illuminated column is straight, and move towards the sides column by column, adapting to the column's profile deviation from vertical by fitting a cubic polynomial to the identified spots' centres, using the estimate from the previous column as a starting point for the fit of the next column. This explanation gives an idea of the pattern recognition algorithm, which is described in detail in Chapter 5.

It is easy to notice that the horizontal rows of the spots follow the surface (see Figure 4.2). The shape of the row can change rapidly, and it is not always possible to decide correctly what row a spot belongs to. We use colour coding of the rows to make the assignment procedure simpler and more robust. Rows are coded periodically with white, red, and green colours (see Figure 4.6). These colours were chosen empirically. They are reflected well from the skin, and at the same time they are different enough to be separated even with our low-intensity light source. For example, as experiments showed, blue and cyan spots cannot be used because they are not visible, especially on people with darker skin. Projecting similar colours is also not a good idea because they are not distinguishable well in the image, and colour classification errors can lead to serious surface reconstruction errors.

Chapter 5

Algorithm and Experimental Results

In this chapter the algorithm for 3D surface reconstruction of the human body in the area of radiation treatment from 2D images is explained in detail. More information about the MATLAB implementation can be found in Appendix A.

Our algorithm consists of the two consecutive special-purpose parts:

1. Processing of the acquired images in order to find elliptical spots, and identifying the correspondence between the projected and the deformed patterns.

2. Reconstructing the surface using triangulation.

The goal of this thesis is to design an algorithm capable of processing images taken under low-light conditions. See Section 4.2 for more information about low-light environment. We do not concentrate on the second part of the algorithm too much. The technique of triangulation is well-known. The result of the triangulation depends heavily on the calibration of the equipment and at this time the system was not calibrated and set up precisely enough. In the end, we present the surface reconstructed from a single image as the final stage of the algorithm, though we cannot say how good or bad it is because the calibration parameters were approximate and, at the moment, we do not have the digitized surface, measured using an approved method, to compare.

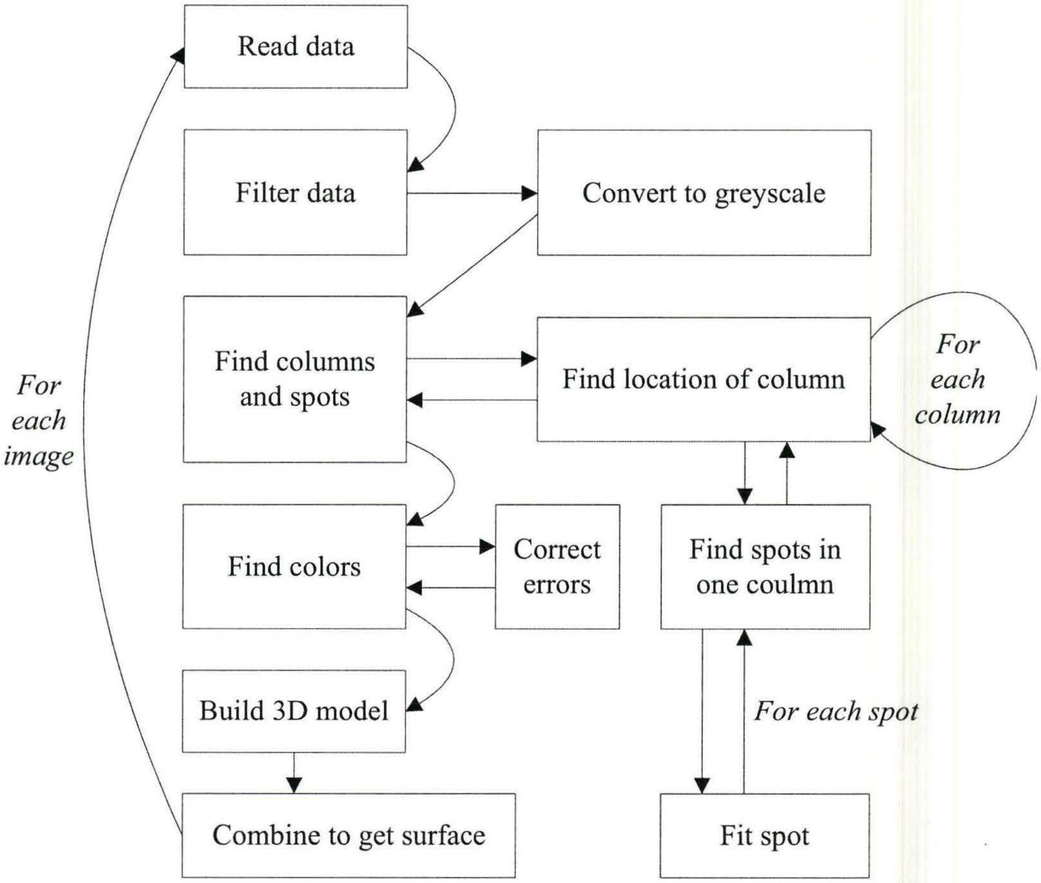


Figure 5.1: Algorithm Scheme

The algorithm scheme is depicted in Figure 5.1. Arrows between the blocks show a control flow. According to the algorithm, first we do noise reduction, or data filtering. In the spot finding procedure colour information is not used for the reasons explained later in this chapter. Therefore, we convert the initial image to greyscale. The resulting image is processed and the spots are found in an iterative procedure utilizing optimization techniques. After that, the colours of the spots are determined by a clustering procedure. The correct colour assignment is very important because it is required to identify the position of the spots on the pattern grid. Due to the low-light conditions and skin scattering properties, colour assignment errors occur. We designed a three-step procedure, which corrects colour errors by using the sequence of the colours in the known projected pattern and the position of the spots relative to their neighbours. After correction, spots are assigned to the grid and the correspondence between the projected and the observed patterns is found. 3D coordinates for surface points are computed and the surface is reconstructed.

5.1 Preprocessing

The image file (see Figure 4.6) of $m \times n$ pixels is opened in MATLAB as an $m \times n$ matrix, each element of which contains a light intensity value for one pixel of a greyscale image, or as an $m \times n \times 3$ matrix with red, green, and blue channel intensities for one pixel of a colour image. Since we use an RGB [34] 24-bit colour camera mode to acquire images with the 640×480 resolution, we initially get a $640 \times 480 \times 3$ matrix with integer values ranging from 0 to 255 (8 bits for each colour). In what follows, we use the terms *light intensity of a pixel* and *matrix element* interchangeably if it does not introduce confusion.

5.1.1 Noise Reduction

In the low-light conditions, when the threshold of the camera's sensitivity is approached, the CCD introduces significant noise. To reduce the noise slightly, so that denoising does not change the original data too much, we tried several filters, such as mean, median and Gauss-like filters.

A *mean* filter sets the light intensity value for a pixel as an average of the neighbouring pixels' intensities. Correspondingly, a *median* filter assigns the median value. But the best results in the experiments with respect to the number and quality of the found spots were reached after applying a *Gauss-like* filter, as we call it here. Our Gauss-like filter finds the intensity of the pixel as a weighted average of the neighbouring pixels' intensities, where the weights are computed from some Gauss-form function. We say that the neighbouring pixels belong to the *weighted average area.* We choose an inverted polynomial function to describe the weights:

$$filter(x,y) = \frac{1}{1 + k \cdot ((x - c_x)^2 + (y - c_y)^2)}, \tag{5.1.1}$$

where $(x, y) \in \Psi$ are the pixels from the weighted average area Ψ, and k is the coefficient controlling the width of the bump. The current pixel, for which the weighted average is done, is located in the centre (c_x, c_y) of the $filter(x, y)$ bump in order to have the most weight. After the weights are computed, they are normalized so as their sum equals one. In the experiments, we filtered each colour channel (red, green, and blue intensity images) separately with 5×5 pixels Gauss-like filter. The weights computed from 5.1.1 are shown in Figure 5.2 by black dots.

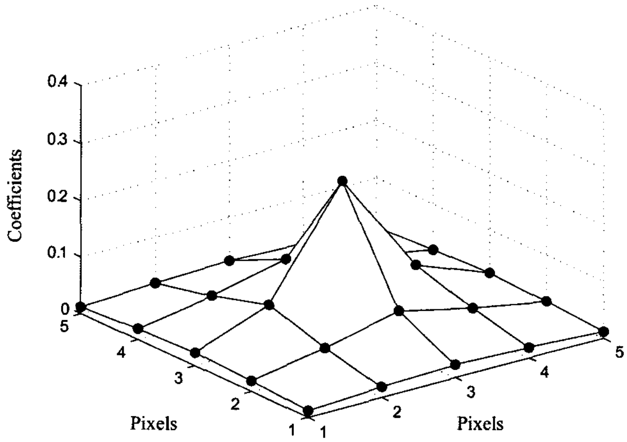


Figure 5.2: Filter weights

5.1.2 Conversion to Greyscale

According to the algorithm, we must find the spots, and assign them to the grid in order to find the correspondence between the projected and the observed spots on the surface. In the discussion in Section 4.2 we explained our choice of the inverse polynomial function to model light intensity data of a spot. But the problem is that there are three colour channels, and the intensities of the white, red, and green spots are fairly different in each of them. In addition, the colour channels themselves have different significance. For example, blue is considered to be a noisy channel and usually provides less useful information than red and green channels. If the spot fitting procedure runs separately for each colour channel, the three sets of found spots will not match. We have to decide how to combine the results. For example, if the spot is found in one channel but is absent in the two others, will we consider it to be a spot?

We conclude that the three channels must be analyzed simultaneously. The non-convex minimization problem of the least-squares difference between the model and the observed data (4.2.3) has to be upgraded to be able to fit a spot in the three channels. Then, the number of the problem variables increases and the objective function gets more complicated. It is not desirable because the problem fits every potential spot, and the number of spots in the image can be about 350–400. So, complicating the model increases the processing time and provokes difficulties in providing the effective first guess for the problem variables, which is crucial because the problem is non-convex.

In fact, there is another option consisting of the two steps. First, converting the image to the greyscale with the chosen coefficients for the red, green, and blue channels so that spots have approximately equal intensity, as much as it is possible. And second, fitting spots on an $m \times n$ matrix by the initial (4.2.3) minimization model. An additional advantage of this approach is that a greyscale image has better Signal to Noise Ratio (SNR) because it averages the three colour channels. In the following discussion the term image means the greyscale image unless the other is specified.

The conversion to greyscale is done by multiplying every colour channel matrix by the chosen coefficient (5.1.1). If the initial image is saved in the matrix D of the size $m \times n \times 3$, then $D_{greyscale}$ corresponds to the greyscale image of $m \times n$ size, and D_{red}, D_{green}, and D_{blue} are the matrices of the red, green, and blue channels of the initial image respectively:

$$D_{greyscale} = r \cdot D_{red} + g \cdot D_{green} + b \cdot D_{blue}. \tag{5.1.1}$$

Now, the question is how to choose the coefficients r, g, and b so as to provide good quality of the greyscale image. By good quality we mean that:

1. spots of different colours have more or less equal intensity in a greyscale image;
2. spots are distinguishable from the background.

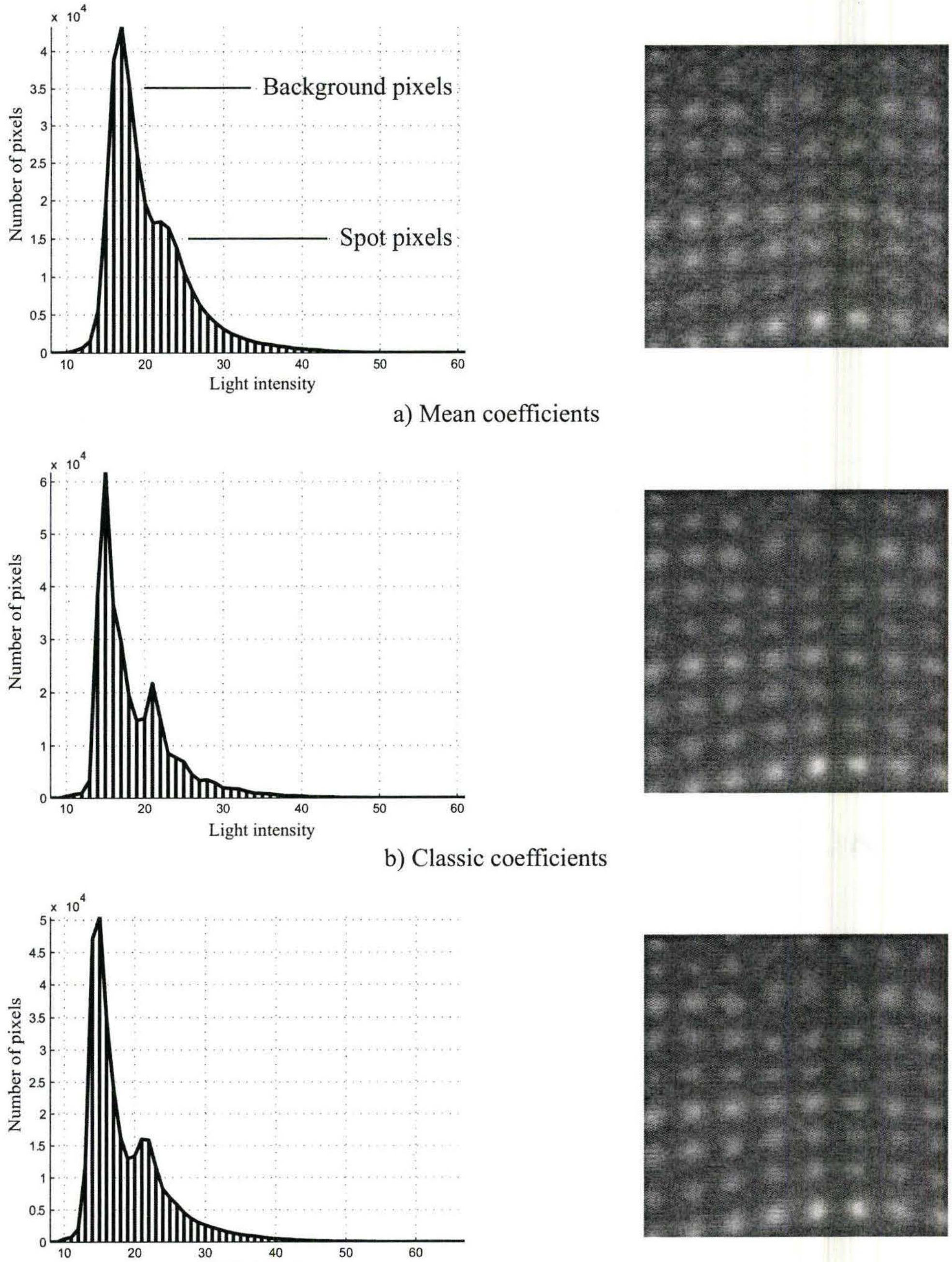
Note, that if the pattern consists of white, red and green spots, as in our case, white spots will most probably be the brightest with any chosen conversion coefficients compared to the green and red spots because the white spots are visible in every colour channel. Therefore, we aim to choose the coefficients so that green and red spots have approximately equal intensity.

As a first example, let us choose the coefficients equal to the *mean* values of red, green, and blue components of all pixels from the initial colour image. Let us analyze the *histogram* of the image showed in Figure 5.3a. The histogram represents the relative frequency of occurrence of the various grey levels in the image [18], that is, for every grey level, from 0 to 255 in our case, the histogram value is the number of pixels, which have this grey level. Hence, the image with bright light elements over dark background should have a two-peak histogram. It is easy to see that the histogram in Figure 5.3a does not have two well-defined peaks. The fragment of the greyscale image next to it demonstrates that some of the spots are too dark and many of them will not be accepted by the spot finding procedure.

Let us choose the most-used *classic* coefficients for colour to greyscale conversion (p. 40 in [34]):

$$D_{greyscale} = 0.299 \cdot D_{red} + 0.587 \cdot D_{green} + 0.114 \cdot D_{blue}. \tag{5.1.2}$$

Figure 5.3b demonstrates the results of the greyscale conversion with the classic coefficients. The image fragment shows much better results compared



a) Mean coefficients

b) Classic coefficients

c) Reciprocal coefficients

Figure 5.3: Histograms

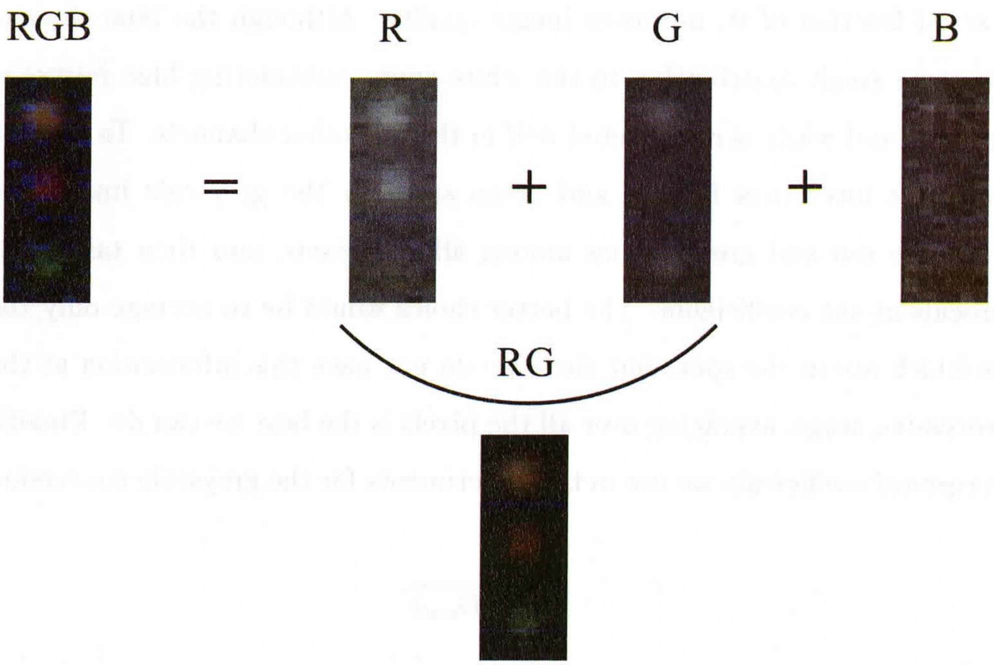


Figure 5.4: Colour channels of white, red, and green spots

to the one in Figure 5.3a, spots are easily visible and have more regular smooth shapes. The histogram shows two well-defined peaks.

The obtained greyscale image (Figure 5.3b) is good enough for processing, it satisfies the two characteristics of the good image stated earlier in this section, however, it is possible to improve the quality even more. Figure 5.4 shows the fragment of the original image (RGB) with white, red, and green spots arranged in a column. R, G, and B are, respectively, red, green, and blue channels of the full-colour RGB image. A closer look on the channels reveals white and red spots in the red channel, white and green spots in the green channel, and almost nothing except for noise can be found in the blue channel (see Figure 5.4) For our images, subtracting the blue channel from the colour image (RG), or taking very small fraction of it, improves image quality. Although the blue channel makes some small contribution to the white spots, subtracting blue improves image SNR and white is represented well in the two other channels. To provide similar light intensities for red and green spots in the greyscale image, we find average red and green values among all the pixels, and then take their reciprocals as the coefficients. The better choice would be to average only the pixels which are in the spots but since we do not have this information at the preprocessing stage, averaging over all the pixels is the best we can do. Finally, the *reciprocal* coefficients we use in the experiments for the greyscale conversion are:

$$\begin{aligned} r &= \frac{1}{\mathrm{mean}(D_{red})}, \\ g &= \frac{1}{\mathrm{mean}(D_{green})}, \\ b &= 0. \end{aligned} \tag{5.1.3}$$

Note, that the coefficients are normalized so that their sum equals one. Figure 5.3c demonstrates the results of the application of the coefficients defined in (5.1.3). The image fragment looks similar to the image in Figure 5.3b but from the histogram it is clear that with the reciprocal coefficients the greyscale image has more dark background pixels than with classic coefficients, which causes the spots to stand out better from the background. Experiments confirm out expectations: the number and quality of found spots is greatest using the reciprocal coefficients.

5.2 Finding Columns and Fitting Spots

Because of the low-quality images, we must provide a good guess for the spots' locations. This is possible, thanks to the structure of the images, as discussed in Section 4.2. In a few words, the spots in the images are always organized in columns independently of the angle of the x-ray machine. The observed columns are especially straight and vertical in the central part of the image, and if we find columns' positions, the task of finding spots in the whole image reduces to finding them along columns, which is much easier. The shape of the processed column and the period of columns is used to predict the shape and position of the next column. This section provides algorithmic details on finding the columns and fitting spots.

5.2.1 Finding Columns

Since the columns of the spots in the image are almost vertical (see Figure 5.6), we sum the rows of the $m \times n$ greyscale light intensity matrix, which results in the $1 \times n$ integrated curve. The central part of the curve shows well-defined periodic peaks (see Figure 5.5) in the positions where columns of spots are in the image. Closer to the sides of the image the peaks become blurry and low. By applying the Fast Fourier Transform (FFT) algorithm [31] to the central part of the integrated curve, we can find the frequency and phase of the peaks. Choosing the frequency which is contained in a reasonable range of values and inverting it gives the period, and the phase gives the location of the columns.

We assume that the most reliable information will be contained in the centre of the image as it is the most visible part, and the perspective does not significantly affect the central columns, so they remain straight and vertical.

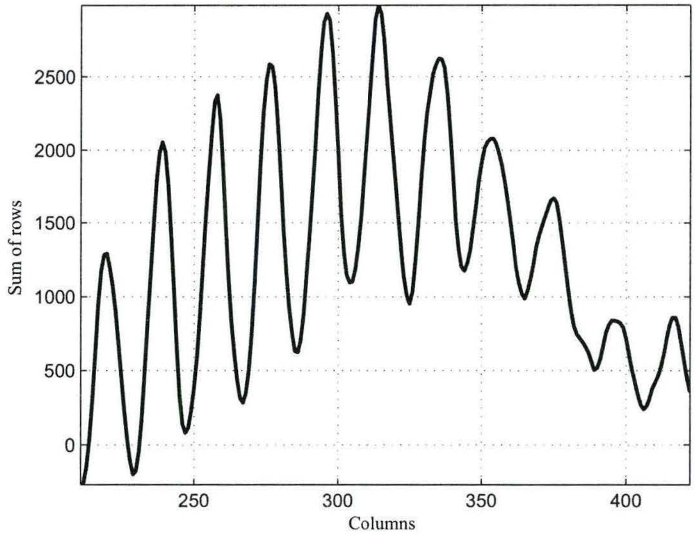


Figure 5.5: Sum of the light intensity matrix rows

Since the best data is in the central part of the image, it makes sense to start the spot finding procedure from the central column, and then proceed right and left to the image sides. The guess for the next column location and shape takes into account the current column shape (deviation from the straight line), and the slight variations of the period across the image. Figure 5.6 shows the straight line guess for the central column, which corresponds to the brightest peak from the integrated curve. It is computed using FFT results.

5.2.2 Finding and Fitting Spots

To find the spots along a column, we analyze that part of the intensity matrix (stripe), whose elements are in a neighbourhood of the estimated column line (plus or minus a column half-period). The brightest areas in the stripe are

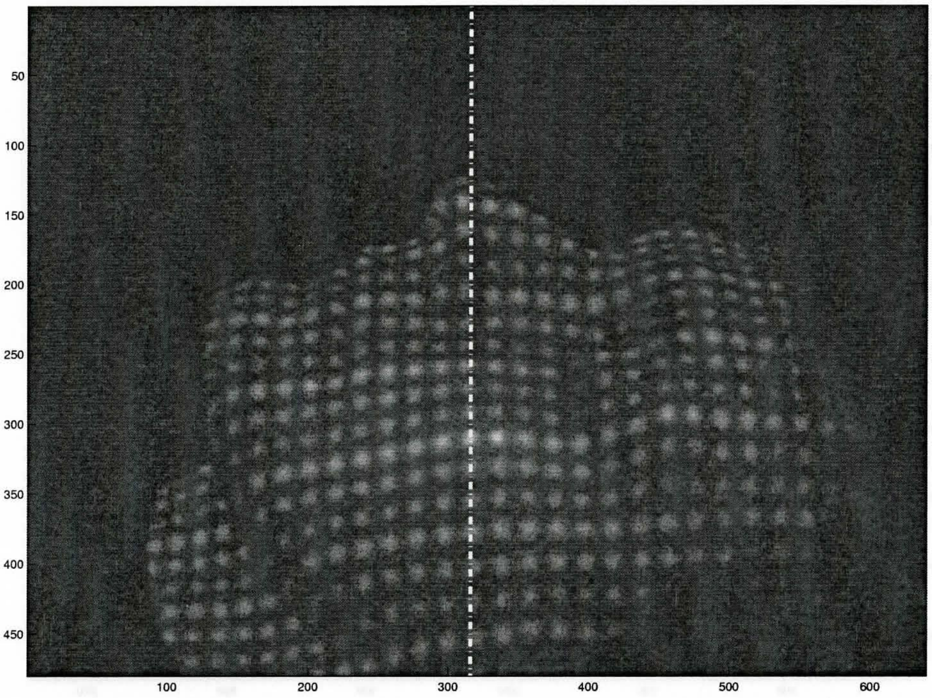


Figure 5.6: Central column (in pixels)

found and the minimization procedure runs to determine if good spots are located in the guessed places. The formulas for the spot model and the least-squares minimization procedure are provided in Section 4.2 (4.2.1, 4.2.2, and 4.2.3) An unconstrained 7-variable minimization procedure runs separately for every spot. After optimization, the parameters of the fitted spots are analyzed. If they are not reasonable, for example, if matrix A (see (4.2.2)) is not positive definite, and hence does not describe an ellipse, or if the centre of the spot is not contained in the analyzed area, etc. the spot is discarded. It may be noise

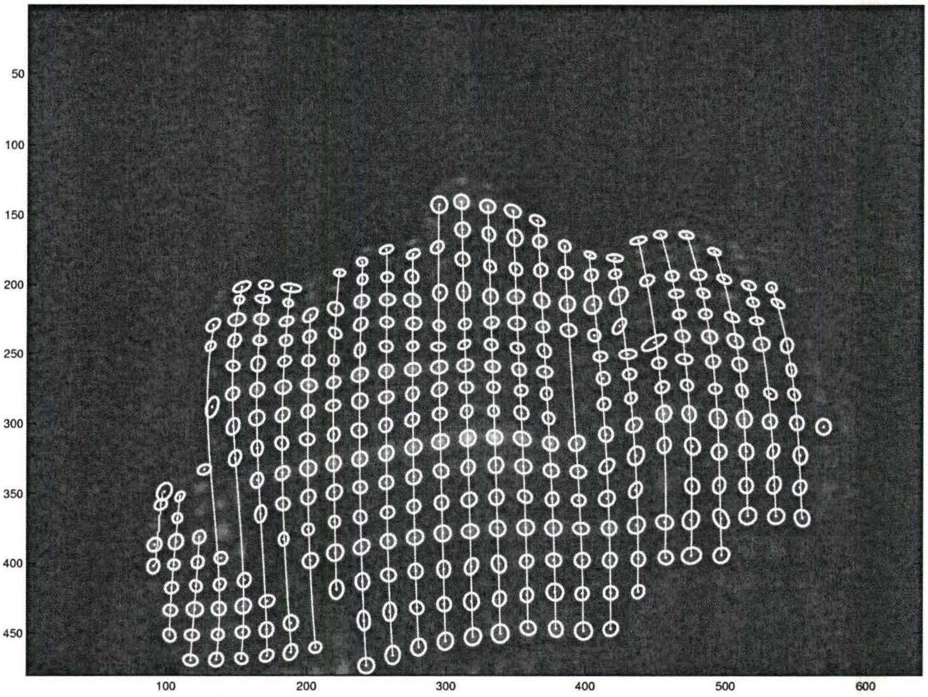


Figure 5.7: Fitted spots along the columns (in pixels)

peaks, or a spot which is partially obscured from either the light source or the camera, and can be covered from another viewpoint.

After spots from the central column are found, they are sorted according to the vertical position and a *cubic curve* is fitted to their centres. Adding a period to this newly created line gives the estimated line for the next column to the right, and subtracting, correspondingly, the column to the left. Using straight estimated lines soon leads to errors in assigning spots to columns because of perspective and surface curvature, so using curves is necessary. Higher degree polynomials could give a little better fit to the centres of the spots in the column, but experiments showed that there is no significant gain after the third degree. The period value is also corrected through the image to adapt for minor changes. The minimal number of spots to build the cubic curve through their centres is four, however, the shape of the curve can become too twisted sometimes and intersect other lines. To avoid this, we demand at least five spots to fit the curve, otherwise the current line estimate is shifted by one period to obtain the guess for the next column.

Figure 5.7 demonstrates the results of the column finding and spot estimation procedure. Note, that fitting spots along the columns not only simplifies and makes the procedure of finding spots in a low-quality images more robust but also does half the work of assigning spots to the grid by assigning them to columns. Putting spots on the grid is necessary to find the correspondence between the projected and the photographed light patterns used in the final surface reconstruction procedure.

5.3 Separating Colours and Assigning Spots to the Grid

When all the spots are identified in the greyscale image, and assigned to the columns, their colours are analyzed in order to locate the spots onto the horizontal rows of the grid.

To separate spots by colours, we have to define concept of a spot's colour first. It is not straightforward because a spot contains many pixels, and each of them has an individual hue and brightness. The simplest way is to use the colour of the central-most pixel. It does not seem robust though because it relies

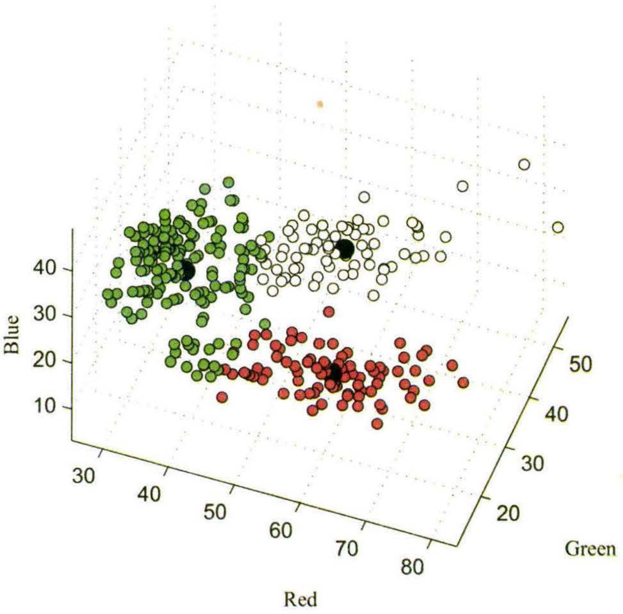


Figure 5.8: Result of k-means colour clustering

on a single pixel. Another approach is to average colours of the pixels inside the spot. This method has the disadvantage of the decreased light intensity of the spot's colour compared to the first approach because the brightest pixels are located in the centre of the spot. In the experiments, we use a weighted average colour of the central pixels as a colour of a spot. The weights are computed from the Gauss-like function in the same way as the weights for filtering were computed in Section 5.1.1. With this approach, the central-most bright pixel has the most influence, and the colours of the neighbouring pixels are taken into consideration.

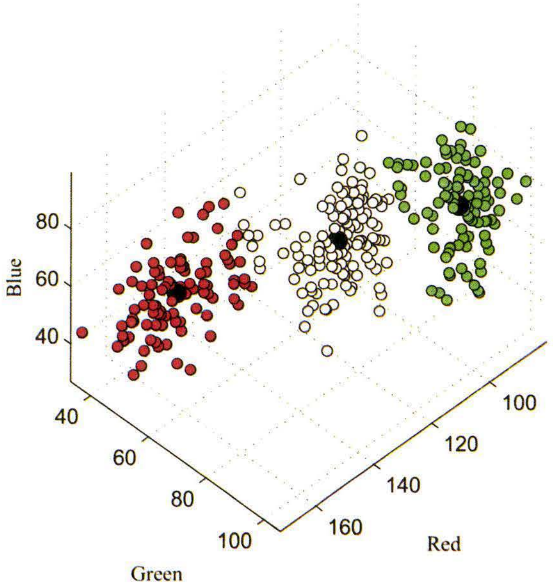


Figure 5.9: Improved k-means colour clustering

5.3.1 Extracting colours

A reliable procedure of extracting colours is not straightforward because human skin absorbs and scatters light variously depending on the light colour, and skin structure and colour. Imaged colours are modified versions of the projected ones. We know the number of the projected colours, therefore the *k-means clustering algorithm* [13] can be applied to make the separation. Figure 5.8 shows colours of the fitted spots in the RGB coordinate system [34], where axes correspond to light intensities of red, green, and blue colour channels. The k-means is an iterative algorithm, which divides the objects into k disjoint subsets so as to minimize sum of squares relative to the geometric centres. It

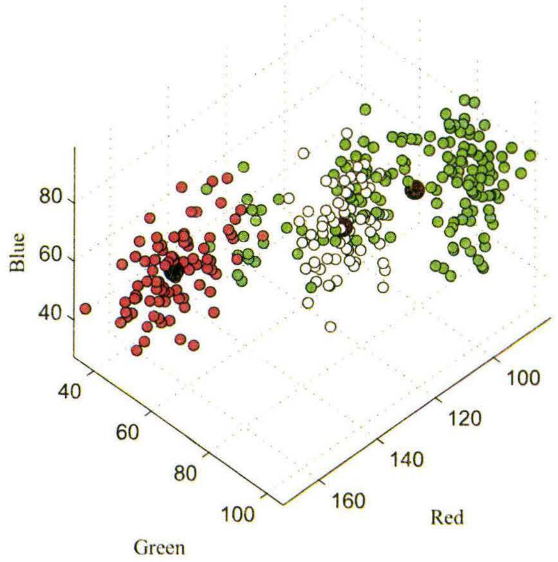


Figure 5.10: Initial k-means colour clustering

has to be supplied with the number of groups, and the initial group centres. Final separation group centres are shown by black dots in Figure 5.8.

A closer look at the colour separation results reveals the separation problem: some of the spots are misclassified. The prevailing problem is that white and red spots are classified as green. It happened because k-means does not account for the shapes of the groups. It basically separates the space by hyperplanes, so that the elements of one group are located closer to their group centre than to other group centres. It is easy to see from Figure 5.8 that, for example, red group seems to be too long, and at some point its spots are located closer in the RGB colour space to the green group centre, and those are classified as green spots by k-means.

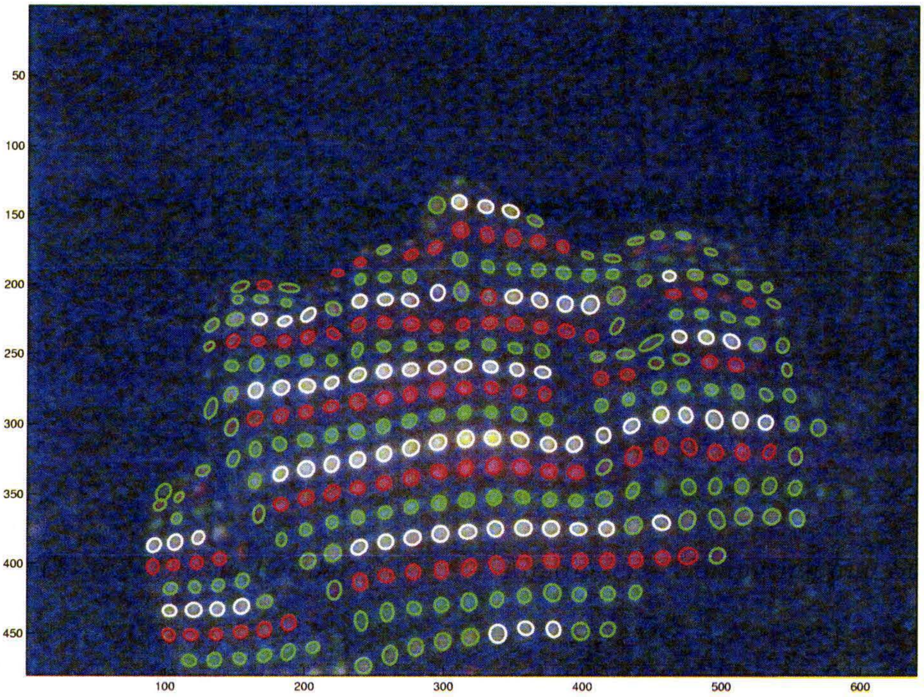


Figure 5.11: Coloured spots with initial k-means clustering (in pixels)

If this result from k-means clustering is used we can have the misclassification error of over 20%. There are two ways to improve the situation: either use another more sophisticated algorithm, or modify the existing data to improve the classification result.

Let us have a look on Figure 5.8 again. The colours of the spots have similar hues but different light intensities. That is why they are located along the three lines connecting the black colour with white, red, and green. We project all the colours of the spots onto the plane M, which is perpendicular to the

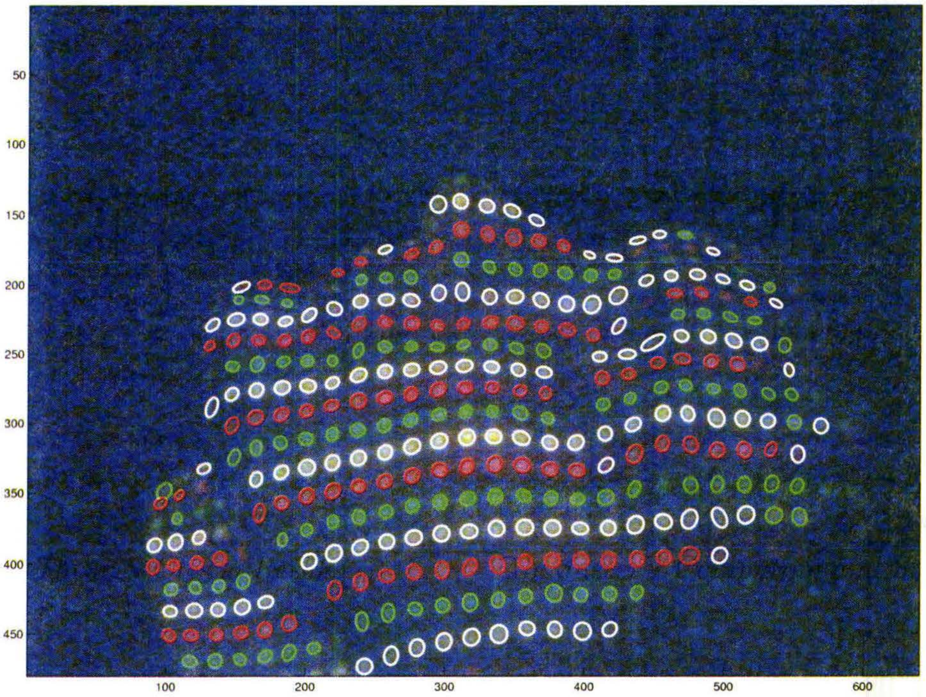


Figure 5.12: Coloured spots with improved k-means clustering (in pixels)

$(1, 1, 1)$ vector, with the centre of projection in $(0, 0, 0)$, that is, the projection of a spot is the intersection of the plane M with the line going through the spot's colour and the black colour $(0, 0, 0)$. Figure 5.9 shows the projected colours and the results of the k-means algorithm applied to them. This separation is much better than the one in Figure 5.8. For an easier comparison, the colours from Figure 5.8 after separation are also projected on M plane and presented in Figure 5.10. It is easy to notice dramatic improvement. Improved k-means has the classification error of about 6%. Figure 5.11 demonstrates the colour separation result of the initial k-means, and Figure 5.12 shows the separation by the improved version of the clustering algorithm.

5.3.2 Correcting Colour Errors

With the improvement to the colour extraction algorithm the number of correctly recognized colours of the spots increases but there are still incorrectly recognized spots, which need to be detected and corrected. Incorrect colour assignment complicates the task of locating the spots on the grid, and can introduce serious spatial errors in the reconstructed surface. We develop a three-step procedure for correcting colour errors.

Step1. Marking Correctly Recognized Coloured Spots

During this procedure colours for no spots are changed. We only mark automatically the spots, whose colours can are *correctly recognized.* The decision of the colour correctness is based on the close location to the neighbouring spots from the same column (no missing spots in between), and the sequence of pattern colours. If the spot has the top and bottom neighbours with the colours correct according to the colour sequence, the spot is marked as correctly recognized. The results are demonstrated in Figure 5.13, where the spots with definitely correctly recognized colours have coloured circles inside, while all the others, unknown, are empty.

Step 2. Analyzing Colour Sequence in Columns

In the second step, the spots marked as unknown are analyzed. If such spot has a correctly recognized close neighbour in a column, its colour is determined from the pattern colour sequence and the spot is marked as correct. The

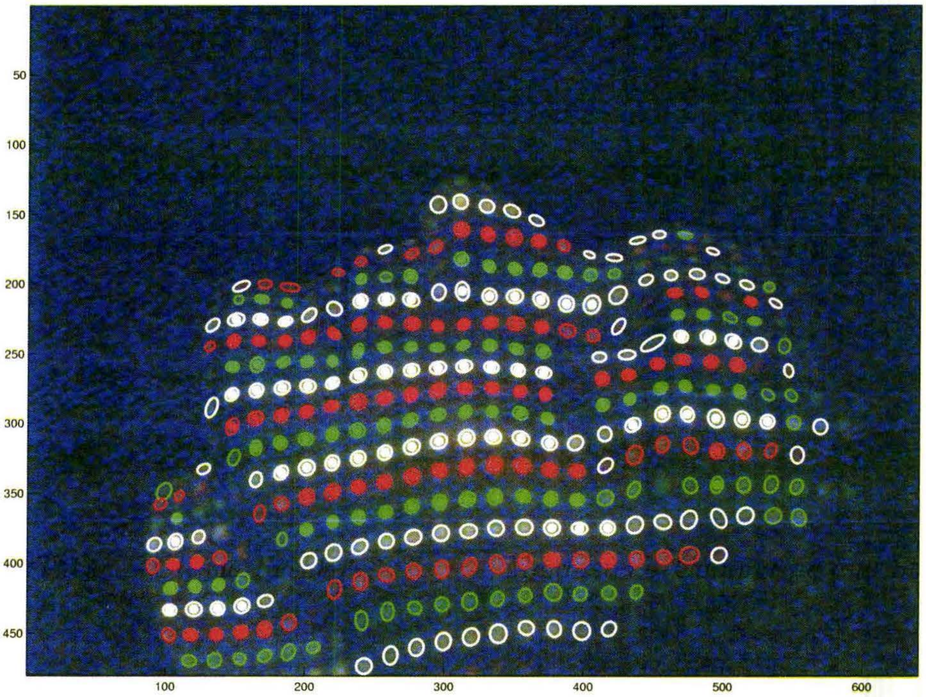


Figure 5.13: Colour correction, step 1: Correctly recognized spots (in pixels)

assigned colour can be different from the initial colour obtained from the k-means clustering. The result of this procedure is shown in Figure 5.14. Again, the correct coloured spots are marked with the same colour circles.

Step 3. Weighted Colour Assignment

Finally, at the last stage the colour should be decided for all the spots that are not marked as correct yet. For each analyzed spot, there are three weights, one for each colour. They depend on:

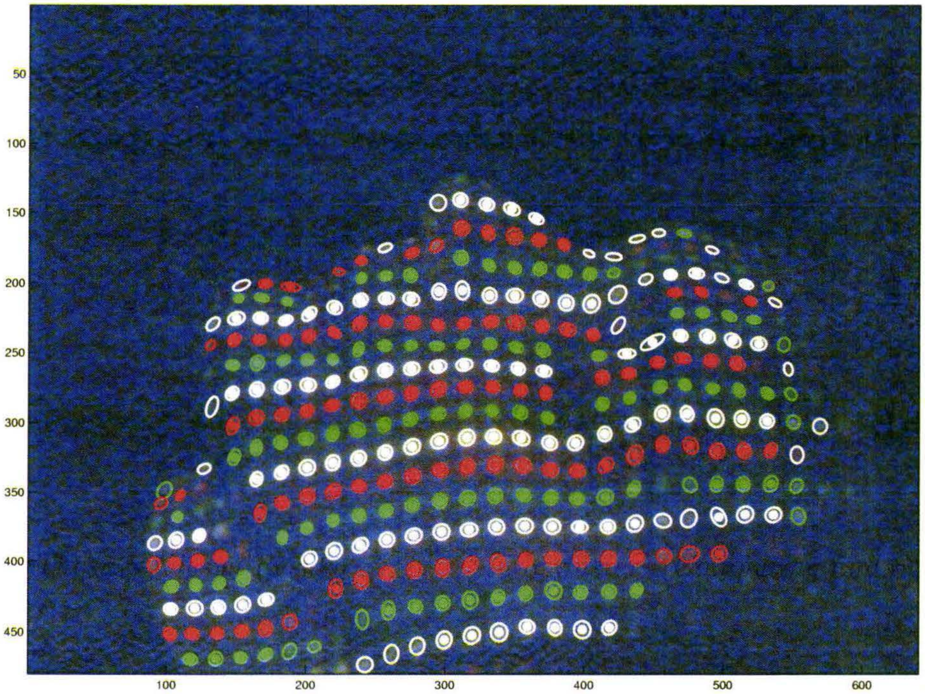


Figure 5.14: Colour correction, step 2: Corrected spots in columns (in pixels)

1. The spot's colour assigned by the k-means clustering procedure.

2. Correctly recognized colours of the column neighbours.

3. Correctly recognized colours of the closest horizontal neighbours.

After analysis, the colour with the greatest weight is assigned to the current spot, and it is marked as correctly recognized. The result is shown in Figure 5.15.

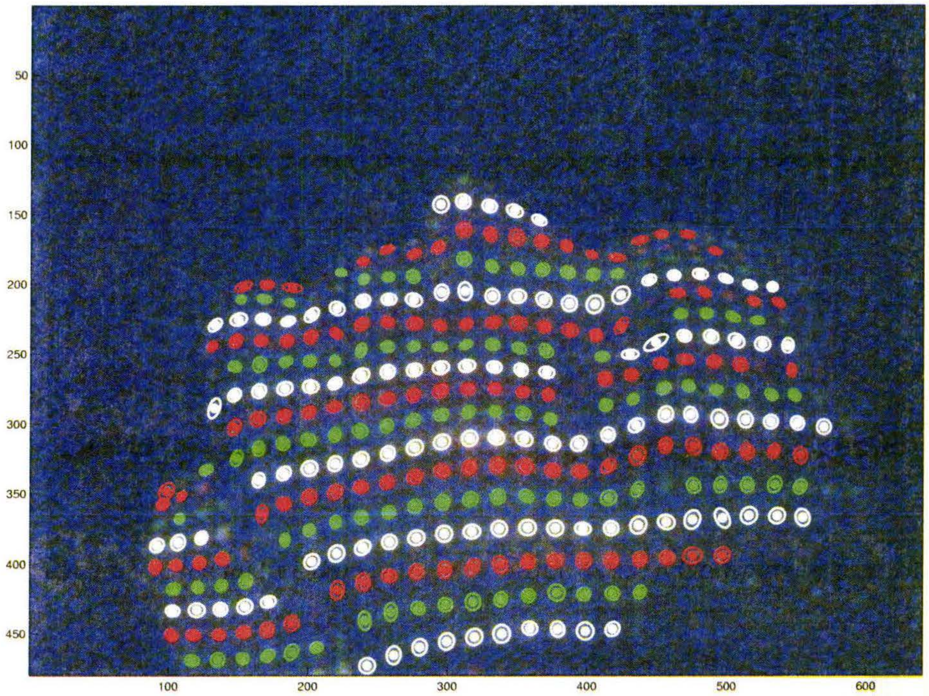


Figure 5.15: Colour correction, step 3: All corrected coloured spots (in pixels)

Taking into account not only the initial k-means colour separation but also the structure of the pattern in the image helps to correct the colour assignment errors and improve the results of the surface reconstruction significantly.

5.3.3 Assigning Spots to the Grid

After the colours are assigned to the fitted spots and the errors are corrected, we can locate the spots on the grid to establish the correspondence between the projected and the photographed light patterns.

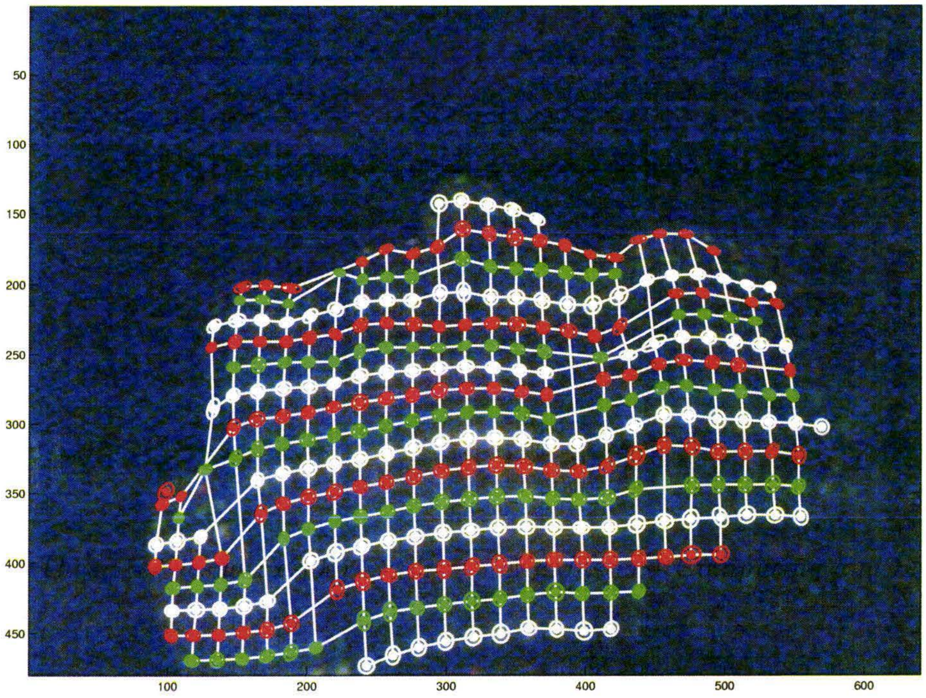


Figure 5.16: Spots on the grid (in pixels)

The assignment to the horizontal rows is made by starting from the central column and moving towards the sides of the image. First, spots from the central column are located on the grid with the consideration of their colours, that is, if there are missing spots in the column, this fact will be interpreted correctly, and there will be corresponding empty nodes on the grid. Then spots from the next column are assigned to the grid depending on their colours and the proximity to the previous column spots of the same colours, and so on until all the spots are placed on the grid (see Figure 5.16).

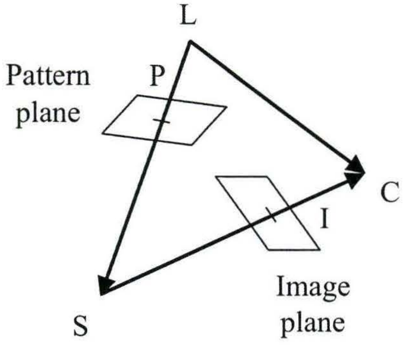


Figure 5.17: Triangulation

5.4 3D Model

After the correspondence between the patterns is known together with the position and orientation of the light source and the camera, spatial coordinates for all identified spots can be computed using *triangulation*. Let L be the light source, C – the camera optical centre, I and P – corresponding points in the image and pattern planes respectively, and S is the point on the examined surface that we want to compute (see Figure 5.17). The surface point S is the intersection of the projected and reflected rays, and in vector algebra this fact means that the vector from the light source to the camera $\overline{LC}$ equals to the sum of the vectors from light to the surface $\overline{LS}$, and from surface to the camera $\overline{SC}$:

$$\overline{LC} = \overline{LS} + \overline{SC}. \tag{5.4.1}$$

Since we do not know vectors $\overline{LS}$ and $\overline{SC}$, they are expressed as

$$\begin{aligned} \overline{LS} &= \overline{LP} \cdot u, \\ \overline{SC} &= \overline{IC} \cdot v. \end{aligned} \tag{5.4.2}$$

Hence, we have a system of three equations (three coordinates for each vector) and two unknowns, u and v:

$$\overline{LC} = \overline{LP} \cdot u + \overline{IC} \cdot v. \tag{5.4.3}$$

After u and v are estimated in the least-squares sense, 3D coordinates of the point S are easily obtained from

$$S = L + \overline{LP} \cdot u. \tag{5.4.4}$$

From the 3D coordinates of the computed surface points, we can reconstruct the surface. Figure 5.18 shows the surface grid reconstructed from the image in Figure 4.6. The obtained surface, indeed, looks like a human face but we cannot estimate how accurate it is. As it was already mentioned, the equipment lacked precise calibration, so we have used approximate values for the calibration and orientation parameters, and we do not have the surface reconstructed by some approved method to compare with.

5.5 3D Surface

This is the final stage of the algorithm when reconstructed surfaces from several images have to be combined together into one surface. Principally, for the structured light technique, one image is enough. However, if some parts of the examined surface cannot be seen by either the camera or the projector from one viewpoint, and also to improve the accuracy and resolution, several images from different viewpoints are necessary.

As stated earlier, the focus of this work was to design an algorithm for image processing and pattern recognition in low-light conditions. The final

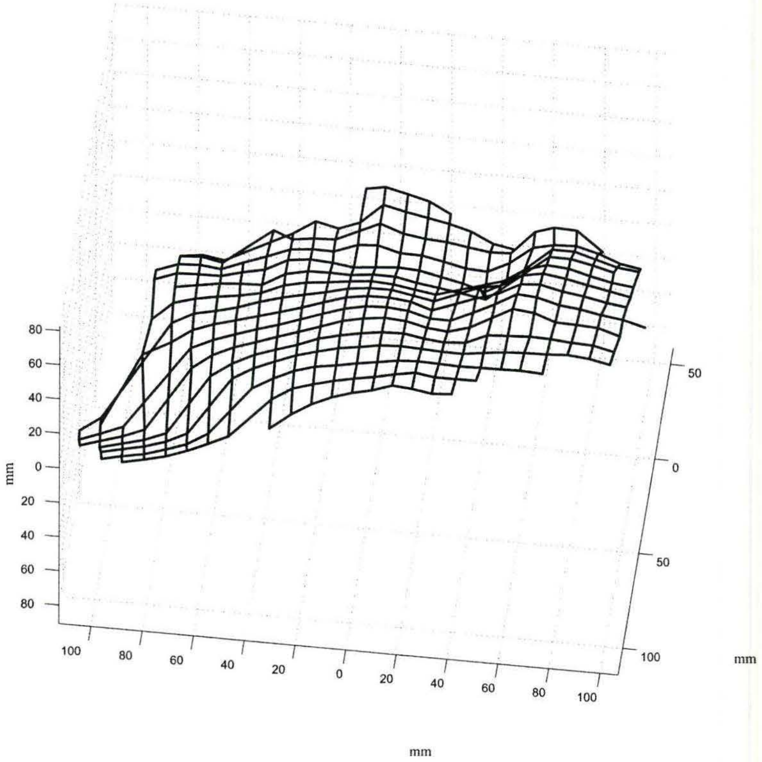


Figure 5.18: Reconstructed surface

3D surface reconstruction procedure has not yet been developed because the equipment was not sufficiently set up and calibrated, largely due to restrictions on access to the equipment in a busy clinical environment. After proper setup, the final stage of the algorithm can be completed to allow the reconstruction of complex surfaces not visible from a single viewpoint.

Chapter 6

Conclusions and Future Work

In this thesis we considered the problem of a patient's body surface reconstruction for radiation therapy planning and treatment, and developed an image processing and surface reconstruction algorithm. To demonstrate its feasibility we built an inexpensive prototype of a 3D surface scanner and showed that it works even under difficult low-light conditions. Procedures described in this thesis are implemented in MATLAB. The processing time of one image on a standard PC[1] is 2–3 minutes. The same algorithm written in C would be able to perform surface reconstruction quickly and unobtrusively within the current clinical workflow. The current version has been implemented in MATLAB because it provides a quick way of testing the algorithm without getting into too many implementation programming details.

We used the 24-bit RGB colour setting in the camera to acquire images. According to the algorithm, the first stage is noise reduction. We tried several filters and chose a Gauss-like filter because it provided the best quantitative and qualitative characteristics for the set of the found spots. The initial image

[1] Pentium 4 CPU 2.8 GHz, 1 GB RAM, Microsoft Windows XP, MATLAB 7.0.1.

is converted to greyscale with the coefficients, which ensure good separation of spots from the background and similar light intensity between the spots of different colours. The acquired images have a certain structure: the columns of spots in the image are almost straight and vertical, especially in the central part of the image. The Fast Fourier Transform is applied to extract the location of the columns, and the spots are found along the columns. This is done, firstly, to reduce the task of finding spots in the image to the simpler procedure of fitting spots along the column, and secondly, to assign spots to the grid in the vertical image direction. After the optimization procedure determines good spots, their colours are found by the k-means clustering algorithm. The correct assignment of colours is very important because it defines the location of a spot on the grid, and later the 3D coordinates of the surface points. We improved the result of the k-means by projecting the colours of the found spots onto the plane in RGB space, where they are separated more easily. We also designed a three-step procedure to correct the colour assignment errors, which uses spatial and colour sequence information to define the spots with correctly recognized colours, and to correct the incorrectly assigned ones. The spots are placed on the grid, and the correspondence between the projected and the observed patterns is established. 3D coordinates of the surface points are computed with triangulation between the light source, the camera, and the surface. The software package is developed in MATLAB and is designed so that it can easily be extended with new program modules.

Future work. There are many possible extensions and future work on the optical hardware, the algorithm, and software design. We would like to mention some of them.

- To make the 3D surface scanner commercially available, it is necessary to review the technical setup in order to improve accuracy, resolution and robustness. The project would benefit from a more sensitive camera with greater resolution, and more powerful and focused light source. For other applications, the system would benefit from multiple cameras and projectors, so that it would not be necessary to move them to different angles to cover the examined surface completely.

- The procedure of combining the surfaces from single images into one surface has to be developed after the proper setup and calibration of equipment to allow reconstructing complex surfaces not visible completely from one viewpoint.

- After combining the surfaces from different views, testing and error estimation is required. We plan to compare our results with the surface reconstructed by either MRI or CT.

- From the software side, the major improvement direction is writing the C code based on the MATLAB implementation described in this thesis. The speedup gain should allow close to real-time surface reconstruction desired for the clinical applications. The C implementation will allow performing the computations before the x-ray machine's head stops in the next position, or viewpoint, to collect data. If we choose to capture a movie instead of separate images then the complete image processing procedure, as described in Chapter 5, has to be executed every n frames. The reason is that spots' shapes and positions are very similar for the neighbouring frames. Therefore, we can use the spots found from the current frame as a good guess for the next frame.

Appendix A

Software Documentation

This appendix contains a high-level documentation for the software package written in MATLAB for 3D surface reconstruction from 2D structured-light images. All the experiments were run in the MATLAB 7.0.1 environment on a PC with Pentium 4 CPU 2.8 GHz, 1 GB RAM, Microsoft Windows XP.

A.1 General Structure of the Software Package

The main package directory contains the following items:

- Main functions and scripts, including the starting script `Contours.m`, which provide the implementation of the algorithm.

- **Service Functions** directory with the auxiliary functions that lighten the work.

- **Drawing Functions** directory, which has functions for outputting the graphical results of the program.

- **Netlab** directory with several functions from the Netlab toolbox of MATLAB developed by Nabney [27]. The Netlab toolbox is designed to provide the central tools necessary for the simulation of theoretically well founded neural network algorithms and related models. We use Netlab's implementation of k-means clustering algorithm. For more details see Netlab's documentation [27].

- **Data** directory where examined images are stored. If this directory is empty or does not exist, the program produces error message and terminates.

- **MAT Files** directory with `.mat` files, where the variables are stored. Every function, called from `Contours.m`, checks if its `.mat` file for the current image is available. If so, the program execution time is reduced by reading the variables from the file instead of running the computations again. This feature is useful if program output has to be generated without making any changes to the functions.

- **Results** directory for saving generated graphs as graphical files for further use.

The package **Contours** contains 39 functions and 4 scripts: 20 main functions and scripts, 10 drawing functions, and 13 service functions. In addition, 4 functions from the Netlab toolbox by Nabney [27] are used for the k-means algorithm. Altogether, our software makes about 3600 lines of code.

A.2 Data Structures

There are several important variables used throughout the **Contours** software package, which have to be explained before the description of the functions is provided:

origColour is a global $m \times n \times 3$ matrix variable, which stores the red, green and blue colour channels of the original $m \times n$ image.

origColourFiltered is a global $m \times n \times 3$ matrix variable, which stores a filtered colour original image, resulted from filtering *origColour* matrix.

orig is a global $m \times n$ matrix variable, representing a greyscale conversion of *origColourFiltered*'s image.

spotParameters is a cell array, every element of which contains an $n_s \times n_p$ matrix, where n_s is the number of spots in one column, and n_p is the number of parameters of a spot. The *spotParameters* array contains the parameters for all the found spots, which are organized in vertical columns.

LIST_OF_FILES is the global array, containing the list of files (names and descriptions) read by ReadDataFileNames function in the Data directory.

centralLine is the number of column in the *orig* matrix, which is the guess line for the central column of the spots. Variable *centralLine* is computed in the **FFTLines** function and passed to the **FindLinesAndSpots** function.

HALF_PERIOD is a global variable, which stores the half of the period in pixels between the spot columns in the image, computed by the **FFTLines** function.

gridMatrix is an $n_r \times n_c \times 3$ matrix, where n_r is the number of spot rows, and n_c is the number of spot columns found. Every element of an $n_r \times n_c$ matrix has 3 values: the number of the spot in the current column, and two coordinates of the centre of the spot in *orig* matrix. Missing spots are expressed by zeros. In the matrix *gridMatrix* spots are arranged not only in the vertical columns, as in *spotParameters* cell array, but also in the horizontal rows, so *gridMatrix* represents the reconstructed grid.

A.3 Connections Between the Main Units

The scheme in Figure A.1 represents the control flow between the mostly significant functions and scripts. The starting script is **Contours.m**, which calls all the program blocks.

A.4 List of Functions

A.4.1 Main Functions

```
spotParameters = AssignAllSpotColours(spotParameters);
gridMatrix = BuildGrid(spotParameters, output, mode);
BuildSurface(gridMatrix, output);
ConvertToGray(output, mode);
spotParameters = CorrectColours(spotParameters, output, mode);
centralLine = FFTLines(output, mode);
```

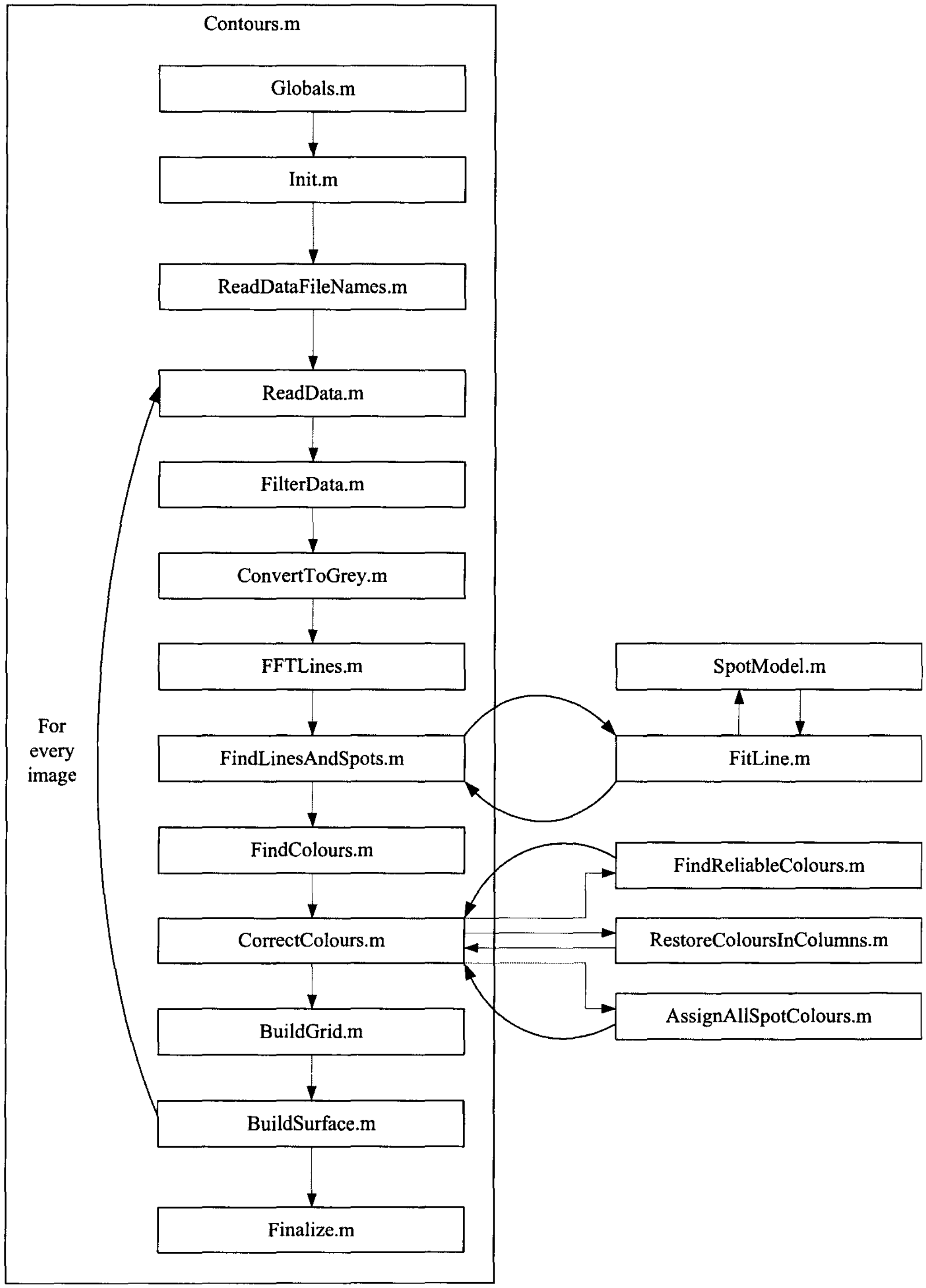


Figure A.1: Program Scheme

```
FilterData(filterName, filterSize, output, mode);
spotParameters = FindColours(spotParameters, output, mode);
spotParameters = FindLinesAndSpots(centralLine, output, mode);
spotParameters = FindReliableColours(spotParameters);
spotParametersOnOneLine = FitLine(approxLine);
ReadData(output, mode);
ReadDataFileNames(output);
spotParameters = RestoreColoursInColumns(spotParameters);
spotParams = SpotModel(row, col);
F = SpotModelFunc(p, n1, n2, spot);
```

A.4.2 Main Scripts

```
Contours;
FinalizeProgram;
Globals;
Init;
```

A.4.3 Service Functions

```
spotParameters = AddColoursToSpotParams(spotParameters,
    pixelsByClusters, colourOrder);
[minIdx, minDist] = ClosestSameColourPrevLineSpot(b1, colour,
    colourVector);
aEdge = ComputeSpotEdge(spotParams);
sumaLineFiltered = DeleteFittedSpot(sumaLineFiltered,
    spotParams);
```

```
exists = ExistsMATFile(imageFileName, programUnit);
colourVector = FillColourVectorZeros(colourVector);
nextColour = NextColour(curColour);
num = NumMissingSpots(colour, gridIdx);
prevColour = PrevColour(curColour);
sumaLine = RefineSumaLine(sumaLine);
spotParams = RemoveDoubleSpots(spotParams, prevSpotParams);
resultLine = SplineThroughSpots(spotCentres);
[resultRow, resultCol] = WeightedCentre(top, bottom, left,
    right);
```

A.4.4 Drawing Functions

```
DrawColouredSpots(spotParameters, len, lineWidth, colour1,
    colour2, colour3, lineStyle);
DrawCorrectSpots(spotParameters, len, lineWidth, colour1,
    colour2, colour3, lineStyle);
DrawHorizontalLines(gridMatrix, lineWidth, lineColour,
    lineStyle);
DrawOrig(mode, graphTitle);
DrawOrigColour(mode, graphTitle);
DrawOrigColourFiltered(mode, graphTitle);
DrawSplines(splineLinesAll, spotParameters, len, lineWidth,
    lineColour, lineStyle);
DrawSpotCentres(spotParameters, len, lineWidth, lineColour,
    lineStyle);
```

```
DrawSpots(spotParameters, len, lineWidth, lineColour, lineStyle);
DrawVerticalLines(gridMatrix, lineWidth, lineColour, lineStyle);
```

A.5 Function Synopses

Function name: AddColoursToSpotParams

Location: *Contours\Service Functions\AddColoursToSpotParams.m*

Prototype: `spotParameters = AddColoursToSpotParams(spotParameters, pixelsByClusters, colourOrder);`

Description: AddColoursToSpotParams assignes colour codes to the found spots according to the results of k-means separation.

This function calls: Globals.

This function is called by: FindColours.

Function name: AssignAllSpotColours

Location: *Contours\AssignAllSpotColours.m*

Prototype: `[spotParameters, numCorrected] = AssignAllSpotColours(spotParameters);`

Description: AssignAllSpotColours is a part of CorrectColours function (step 3). It assigns a colour for any spot with an unconfirmed colour. It does so by looking at the spot's closest confirmed horizontal and vertical neighbours, and the spot's initial colour from k-means. The colour, which gets the most weight, is assigned to the spot.

This function calls: Globals, NextColour, PrevColour.

This function is called by: FindColours.

Function name: BuildGrid

Location: *Contours\BuildGrid.m*

Prototype: `gridMatrix = BuildGrid(spotParameters, output, mode);`

```
    output = {'text' 'timing' 'grid' 'spots on grid'} or {'all'};
    mode = 'real' or 'improved';
```

Description: BuildGrid assignes the coloured spots to a regular rectangular grid, which is necessary for the surface reconstruction procedure. Spots are already assigned to the vertical columns, BuildGrid processes colour information to assign the spots to the horizontal rows based on the proximity information.

This function calls: Globals, DrawColouredSpots, DrawHorizontalLines, DrawOrigColourFiltered, DrawVerticalLines, ClosestSameColourPrevLineSpot, ExistsMATFile, FillColourVectorZeros, NumMissingSpots.

This function is called by: Contours.

Function name: BuildSurface

Location: *Contours\BuildSurface.m*

Prototype: `BuildSurface(gridMatrix, output);`

```
    output = {'text' 'timing' 'surface grid'} or {'all'};
```

Description: BuildSurface computes 3D coordinates of the surface points via triangulation using the grid provided by the BuildGrid procedure.

This function calls: Globals, ExistsMATFile.

This function is called by: Contours.

Function name: ClosestSameColourPrevLineSpot

Location: *Contours\Service Functions\ClosestSameColourPrevLineSpot.m*

Prototype: `[minIdx, minDist] = ClosestSameColourPrevLineSpot(b1, colour, colourVector);`

Description: For a given spot, ClosestSameColourPrevLineSpot finds the closest spot from the previous column of the same colour.

This function calls: Globals.

This function is called by: BuildGrid.

Function name: ComputeSpotEdge

Location: *Contours\Service Functions\ComputeSpotEdge.m*

Prototype: `aEdge = ComputeSpotEdge(spotParams);`

Description: For a given spot, ComputeSpotEdge finds an ellipse representing the spot's "edge". It is chosen between the background and the maximal spot value using some predefined ratio.

This function calls: Globals.

This function is called by: SpotModel, DrawColouredSpots, DrawCorrectSpots, DrawSpots, DeleteFittedSpot.

Function name: Contours

Location: *Contours\Contours.m*

Prototype: `Contours;`

Description: Contours reads images of the surface illuminated with a regular colour-coded pattern and reconstructs the surface in 3D. This version works with the pattern of white, red, green rows of spots, and uses an inverse polynomial model to fit the light spot but can be adapted to read other patterns and operate with other models.

This function calls: BuildGrid, BuildSurface, ConvertToGray, CorrectColours, FFTLines, FilterData, FinalizeProgram, FindColours, FindLinesAndSpots, Globals, Init, ReadData, ReadDataFileNames.

This function is called by:

Function name: ConvertToGrey

Location: *Contours\ConvertToGrey.m*

Prototype: `ConvertToGrey(output, mode);`

```
output = {'text' 'timing' 'greyscale image' 'histogram'} or
{'all'};
mode = 'real' or 'improved';
```

Description: ConvertToGrey coverts a 3D *origColour* matrix (original image) to a 2D *orig* matrix ("greyscale" image) using (r,g,b) coefficients.

This function calls: Globals, DrawOrig, ExistsMATFile.

This function is called by: Contours.

Function name: CorrectColours

Location: *Contours\CorrectColours.m*

Prototype: `spotParameters = CorrectColours(spotParameters, output, mode);`

```
output = {'text' 'timing' 'correct spots 1' 'correct spots 2'
'correct spots 3'} or {'all'};
mode = 'real' or 'improved';
```

Description: CorrectColours analyzes the colours of the found spots and corrects them based on spot spatial proximity and colour alternation information.

On the first step (FindReliableColours function), a set of "reliable" coloured spots is found. On the second step (RestoreColoursInColumns function), the spots in columns are analyzed to infer the colours for the "non-reliable" spots where possible from colour alternation and their proximity to the "reliable" spots. On the third step (AssignAllSpotColours function), colours for all the spots not confirmed yet are estimated in relation to the top-bottom-left-right neighbours' colours and their initial colour.

This function calls: AssignAllSpotColours, FindReliableColours, Globals, RestoreColoursInColumns, DrawCorrectSpots, DrawOrigColourFiltered, ExistsMATFile.

This function is called by: Contours.

Function name: DeleteFittedSpot

Location: *Contours\Service Functions\DeleteFittedSpot.m*

Prototype: `sumaLineFiltered = DeleteFittedSpot(sumaLineFiltered, spotParams);`

Description: DeleteFittedSpot deletes accepted spot bump using its semimajor from the column profile array.

This function calls: Globals, ComputeSpotEdge.

This function is called by: FitLine.

Function name: DrawColouredSpots

Location: *Contours\Drawing Functions\DrawColouredSpots.m*

Prototype: `DrawColouredSpots(spotParameters, len, lineWidth, colour1, colour2, colour3, lineStyle);`

Description: DrawColouredSpots draws the coloured spots in the current figure. Parameter *len* is the number of columns to draw. One is set to draw the central line spots only, and length(spotParameters) draws all the spots.
This function calls: Globals, ComputeSpotEdge.
This function is called by: BuildGrid, FindColours.

Function name: DrawCorrectSpots
Location: *Contours\Drawing Functions\DrawCorrectSpots.m*
Prototype: `DrawCorrectSpots(spotParameters, len, lineWidth, colour1, colour2, colour3, lineStyle);`
Description: DrawCorrectSpots draws the coloured spots in the current figure, placing the same colour circles inside the spots with confirmed colour. Parameter *len* is the number of columns to draw. One is set to draw the central line spots only, and length(spotParameters) draws all the spots.
This function calls: Globals, ComputeSpotEdge.
This function is called by: CorrectColours.

Function name: DrawHorizontalLines
Location: *Contours\Drawing Functions\DrawHorizontalLines.m*
Prototype: `DrawHorizontalLines(gridMatrix, lineWidth, lineColour, lineStyle);`
Description: DrawHorizontalLines connects the spots in the rows with lines.
This function calls: Globals.
This function is called by: BuildGrid.

Function name: DrawOrig

Location: *Contours\Drawing Functions\DrawOrig.m*

Prototype: `DrawOrig(mode, graphTitle);`

`mode = 'real' or 'improved';`

Description: DrawOrig draws the greyscale image of the examined surface contained in matrix *orig*, or an improved version of it.

This function calls: Globals.

This function is called by: ConvertToGray, FFTLines, FindLinesAndSpots.

Function name: DrawOrigColour

Location: *Contours\Drawing Functions\DrawOrigColour.m*

Prototype: `DrawOrigColour(mode, graphTitle);`

`mode = 'real' or 'improved';`

Description: DrawOrigColour draws the RGB image of the examined surface, contained in matrix *origColour*, or an improved version of it.

This function calls: Globals.

This function is called by: ReadData.

Function name: DrawOrigColourFiltered

Location: *Contours\Drawing Functions\DrawOrigColourFiltered.m*

Prototype: `DrawOrigColourFiltered(mode, graphTitle);`

`mode = 'real' or 'improved';`

Description: DrawOrigColourFiltered draws the RGB image of the examined surface, contained in matrix *origColourFiltered*, or an improved version of it.

This function calls: Globals.

This function is called by: BuildGrid, CorrectColours, FilterData, FindColours.

Function name: DrawSplines

Location: *Contours\Drawing Functions\DrawSplines.m*

Prototype: `DrawSplines(splineLinesAll, spotParameters, len, lineWidth, lineColour, lineStyle);`

Description: DrawSplines draws the cubic splines, fitted to the centres of the found spots, for every column of spots, from first spot's centre to the last one's in the column. Parameter *len* is the number of splines to draw. One is set to draw the central column spline only, and length(spotParameters) draws all the splines.

This function calls: Globals.

This function is called by: FindLinesAndSpots.

Function name: DrawSpotCentres

Location: *Contours\Drawing Functions\DrawSpotCentres.m*

Prototype: `DrawSpotCentres(spotParameters, len, lineWidth, lineColour, lineStyle);`

Description: DrawSpotCentres draws the centres of the found spots. Parameter *len* is the number of spot columns, for which spot centres are drawn. One is set to draw the central column spot centres only, and length(spotParameters) draws all the spot centres.

This function calls: Globals.

This function is called by: FindLinesAndSpots.

Function name: DrawSpots

Location: *Contours\Drawing Functions\DrawSpots.m*

Prototype: `DrawSpots(spotParameters, len, lineWidth, lineColour, lineStyle);`

Description: DrawSpots draws the found spots in the current figure. Parameter *len* is the number of columns to draw. One is set to draw the central line spots only, and length(spotParameters) draws all the spots.

This function calls: Globals, ComputeSpotEdge.

This function is called by: FindLinesAndSpots.

Function name: DrawVerticalLines

Location: *Contours\Drawing Functions\DrawVerticalLines.m*

Prototype: `DrawVerticalLines(gridMatrix, lineWidth, lineColour, lineStyle);`

Description: DrawVerticalLines connects the spots in the columns with lines.

This function calls: Globals.

This function is called by: BuildGrid.

Function name: ExistsMATFile

Location: *Contours\Service Functions\ExistsMATFile.m*

Prototype: `exists = ExistsMATFile(imageFileName, programUnit);`

Description: ExistsMATFile checks if there is a .mat file with the name <*imageFileName*>_<*programUnit*>.mat. If so, Contours reads the workspace variables and does not run the program unit. Also it sets the global variable *MAT_FILE*.

This function calls: Globals.

This function is called by: BuildGrid, BuildSurface, ConvertToGray, CorrectColours, FFTLines, FilterData, FindColours, FindLinesAndSpots, ReadData.

Function name: FFTLines

Location: *Contours\Drawing Functions\FFTLines.m*

Prototype: `centralLine = FFTLines(output);`

```
    output = {'text' 'timing' 'suma' 'power' 'cosine'
    'central line'} or {'all'};
```

Description: FFTLines uses Fast Fourier Transform (FFT) to find the period and the location of the vertical columns of spots. It returns the number of columns in matrix *orig*, which corresponds to the maximal peak (the brightest column of spots).

This function calls: Globals, DrawOrig, ExistsMATFile.

This function is called by: Contours.

Function name: FillColourVectorZeros

Location: *Contours\Service Functions\FillColourVectorZeros.m*

Prototype: `colourVector = FillColourVectorZeros(colourVector);`

Description: FillColourVectorZeros estimates the vertical location of the missing spots in a column based on their neighbours' image coordinates. The missing spots' positions are initially set as zeros in *colourVector* array.

This function calls: Globals.

This function is called by: BuildGrid.

Function name: FilterData

Location: *Contours\FilterData.m*

Prototype: `FilterData(filterName, filterSize, output, mode);`

```
    filterName = 'mean' or 'median' or 'gauss';
    output = {'text' 'timing' 'filtered image' 'filter'} or
    {'all'};
    mode = 'real' or 'improved';
```

Description: FilterData applies filter *filterName* of the size *filterSize* to the original image stored in *origColour* matrix, which results in a filtered image in *origColourFiltered* matrix.

This function calls: Globals, DrawOrigColourFiltered, ExistsMATFile.

This function is called by: Contours.

Function name: FinalizeProgram

Location: *Contours\FinalizeProgram.m*

Prototype: `FinalizeProgram;`

Description: FinalizeProgram removes Contours' working directories from the MATLAB path and outputs final statistics.

This function calls:

This function is called by: Contours.

Function name: FindColours

Location: *Contours\FindColours.m*

Prototype: `spotParameters = FindColours(spotParameters, output,`
`    mode);`

```
    output = {'text' 'timing' 'coloured spots' 'kmeans'
    'kmeans_grey'} or {'all'};
    mode = 'real' or 'improved';
```

Description: FindColours uses k-means clustering algorithm to divide the fitted spots by three groups depending on their colour. This is necessary to find the correspondence between the projected and the observed pattern.

This function calls: Globals, DrawColouredSpots, DrawOrigColourFiltered, foptions (Netlab), kmeans (Netlab), AddColoursToSpotParams, ExistsMATFile.

This function is called by: Contours.

Function name: FindLinesAndSpots

Location: *Contours\FindLinesAndSpots.m*

Prototype: `spotParameters = FindLinesAndSpots(centralLine, output, mode);`

```
    output = {'text' 'timing' 'spots and splines'
    'approxlines and splines' 'central spots' 'splines'} or
    {'all'};
    mode = 'real' or 'improved';
```

Description: FindLinesAndSpots starts from the central column with spots, computed in FFTLines, fits spots along it, and then moves right till the side of the image, finds next columns (using column period), fits the spots. The same procedure is repeated from the centre to the left side.

The guess line for the next column is computed from the cubic curve, which approximates the centres of the fitted spots in the current column, shifted by one period.

This function calls: FitLine, Globals, DrawOrig, DrawSplines, DrawSpotCentres, DrawSpots, ExistsMATFile, RemoveDoubleSpots, SplineThroughSpots.

This function is called by: Contours.

Function name: FindReliableColours

Location: *Contours\FindReliableColours.m*

Prototype: `spotParameters = FindReliableColours(spotParameters);`

Description: FindReliableColours is a part of CorrectColours function (step 1). It analyzes every spot together with its neighbours in vertical columns. If the distance between spots is small enough, which means there are no missing spots in between, and the colour sequence is correct, the spots are marked as correctly recognized.

This function calls: Globals, NextColour, PrevColour.

This function is called by: CorrectColours.

Function name: FitLine

Location: *Contours\FitLine.m*

Prototype: `spotParametersOnOneLine = FitLine(approxLine);`

Description: FitLine takes the guess line for a column and analyzes if there are any good spots along this line in the image.

This function calls: Globals, SpotModel, DeleteFittedSpot, RefineSumaLine.

This function is called by: FindLinesAndSpots.

Function name: Globals

Location: *Contours\Globals.m*

Prototype: `Globals;`

Description: Globals declares constants and global variables.

This function calls:

This function is called by: AssignAllSpotColours, BuildGrid, BuildSurface, Contours, ConvertToGray, CorrectColours, FFTLines, FilterData, FindColours, FindLinesAndSpots, FindReliableColours, FitLine, ReadData, ReadDataFileNames, RestoreColoursInColumns, SpotModel, DrawColouredSpots, DrawCorrectSpots, DrawHorizontalLines, DrawOrig, DrawOrigColour, DrawOrigColourFiltered, DrawSplines, DrawSpotCentres, DrawSpots, DrawVerticalLines, AddColoursToSpotParams, ClosestSameColourPrevLineSpot, ComputeSpotEdge, DeleteFittedSpot, ExistsMATFile, FillColourVectorZeros, NextColour, NumMissingSpots, PrevColour, RefineSumaLine, RemoveDoubleSpots, SplineThroughSpots, WeightedCentre.

Function name: Init

Location: *Contours\Init.m*

Prototype: `Init;`

Description: Init initializes global variables and adds necessary directories to the MATLAB path.

This function calls:

This function is called by: Contours.

Function name: NextColour

Location: *Contours\Service Functions\NextColour.m*

Prototype: `nextColour = NextColour(curColour);`

Description: NextColour finds the next colour code in the pattern sequence.

This function calls: Globals.

This function is called by: AssignAllSpotColours, FindReliableColours, RestoreColoursInColumns.

Function name: NumMissingSpots

Location: *Contours\Service Functions\NumMissingSpots.m*

Prototype: `num = NumMissingSpots(colour, gridIdx);`

Description: NumMissingSpots finds the number of missing spots before the current one in the column based on the colour sequence in the pattern.

This function calls: Globals.

This function is called by: BuildGrid.

Function name: PrevColour

Location: *Contours\Service Functions\PrevColour.m*

Prototype: `prevColour = PrevColour(curColour);`

Description: PrevColour finds the previous colour code in the pattern sequence.

This function calls: Globals.

This function is called by: AssignAllSpotColours, FindReliableColours, RestoreColoursInColumns.

Function name: ReadData

Location: *Contours\ReadData.m*

Prototype: `ReadData(output, mode);`

```
    output = {'text' 'timing' 'original image'} or {'all'};
    mode = 'real' or 'improved';
```

Description: ReadData reads an $m \times n$ pixels image from a current file to an $m \times n \times 3$ colour light intensity matrix.

This function calls: Globals, DrawOrigColour, ExistsMATFile.

This function is called by: Contours.

Function name: ReadDataFileNames

Location: *Contours\ReadDataFileNames.m*

Prototype: `ReadDataFileNames(output);`

```
    output = {'text' 'timing'} or {'all'};
```

Description: ReadDataFileNames reads the filenames from the *Data* directory.

This function calls: Globals.

This function is called by: Contours.

Function name: RefineSumaLine

Location: *Contours\Service Functions\RefineSumaLine.m*

Prototype: `sumaLine = RefineSumaLine(sumaLine);`

Description: After the spot is found, the pixels it takes are set to zero. It often happens that the small non-zero intervals are left between the fitted spots in the column, which are too small to contain a spot. RefineSumaLine "cleans up" the column, which saves time and resources by omitting the unnecessary optimization procedure runs for such places.

This function calls: Globals.

This function is called by: FitLine.

Function name: RemoveDoubleSpots

Location: *Contours\Service Functions\RemoveDoubleSpots.m*

Prototype: `spotParams = RemoveDoubleSpots(spotParams, prevSpotParams);`

Description: RemoveDoubleSpots has two steps. First, it checks the fitted spots in the current line, and removes the spots which are too close in the case the fitting procedure converged to the one spot from the two initially different neighbouring locations.

At the second step, RemoveDoubleSpots compares the current column of the fitted spots to the previous column and deletes the spots, which are extremely close to one of the previous column's spots, so that it is, in fact, the same spot. This can happen if the columns are curved. The spot is deleted from the current column and not from the previous one because of the assumption that the data, closer to the image centre, is generally better, and the algorithm runs so that the previous column is always closer to the centre than the current one.

This function calls: Globals.

This function is called by: FindLinesAndSpots.

Function name: RestoreColoursInColumns

Location: *Contours\RestoreColoursInColumns.m*

Prototype: `spotParameters =`

`RestoreColoursInColumns(spotParameters);`

Description: RestoreColoursInColumns is a part of CorrectColours function (step 2). It passes column by column, and sets colours by using pattern colour sequence for those spots, whose close vertical neighbours have confirmed colour.

This function calls: Globals, NextColour, PrevColour.

This function is called by: CorrectColours.

Function name: SplineThroughSpots

Location: *Contours\Service Functions\SplineThroughSpots.m*

Prototype: `resultLine = SplineThroughSpots(spotCentres);`

Description: SplineThroughSpots computes a cubic approximation through the fitted spots' centres from one column. Shifted on a period, it will serve as a guess for the next column. It demands at least five fitted spots for good results, otherwise it returns empty array and the current column's guess line is shifted by a period and used for the next column guess.

This function calls: Globals.

This function is called by: FindLinesAndSpots.

Function name: SpotModel

Location: *Contours\SpotModel.m*

Prototype: `spotParams = SpotModel(row, col);`

Description: SpotModel takes the guess of the spot location (*row, col*) in the image matrix, and tries to fit the spot model *f* to the data. Least-squares fitting is used (*lsqnonlin* MATLAB function) to retrieve the spot's parameters

p. If the values of the variables in p are not reasonable, the spot is discarded. Spot model currently used:

$$f(x,y) = v_0 + \frac{v_1}{1+s(x,y)},$$

$$s(x,y) = \left(\begin{array}{c} x-b_x \\ y-b_y \end{array} \right)^T A \left(\begin{array}{c} x-b_x \\ y-b_y \end{array} \right), \qquad A = \left(\begin{array}{cc} a_1 & a_2 \\ a_2 & a_3 \end{array} \right),$$

$$p = (a_1,\ a_2,\ a_3,\ b_1,\ b_2,\ v_0,\ v_1)\,,$$

where A is a symmetric matrix, which has to be positive definite to describe an elliptical spot shape.

This function calls: Globals, SpotModelFunc, ComputeSpotEdge, WeightedCentre.

This function is called by: FitLine.

Function name: SpotModelFunc

Location: *Contours\SpotModelFunc.m*

Prototype: `F = SpotModelFunc(p, n1, n2, spot);`

Description: SpotModelFunc provides the function for the least-squares fitting procedure, which equals to the squared difference between the data and an inverse polynomial model

$$F = (\phi_{x,y} - f(x,y))^2.$$

Fits over circular area (where *spot* is not equal to zero). Parameters *n1* and *n2* represent the size of the *spot* matrix.

This function calls:

This function is called by: SpotModel.

Function name: WeightedCentre

Location: *Contours\Service Functions\WeightedCentre.m*

Prototype: `[resultRow, resultCol] = WeightedCentre(top, bottom, left, right);`

Description: WeightedCentre finds weighted centre over the fitting square described by *top*, *bottom*, *left*, *right* parameters.

This function calls: Globals.

This function is called by: SpotModel.

A.6 Output Example

Here, we provide an example of the program output, when the output options for all the functions were set to {`'all'`}, which means that timing, statistical information and all available graphs were generated. The image file used in the example is the same as in Chapter 5.

If there are no `.mat` files provided for in the MATFiles directory, then the program does all the computation, which usually takes 2 to 3 minutes. The program output is provided below. Every program unit prints its name in the square brackets, and provides some statistics about its execution.

```
======================== CONTOURS ========================
 Surface reconstruction from noisy structured-light images
==========================================================

[ReadDataFileNames]:
  -- Data directory: D:\_Thesis_\Contours\_Data_
  -- 1 file(s) read: "face.tif"
  -- Elapsed time: 0.016 sec

Image #1   :   face.tif
---------------------------------------------------------------

[ReadData]:
  -- Resolution: 480 x 640 pixels
  -- Elapsed time: 0.891 sec

[FilterData]:
  -- Filter type: gauss
  -- Filter size: 5 x 5 pixels
  -- Elapsed time: 1.797 sec

[ConvertToGrey]:
  -- Coefficient vector (r,g,b) = (0.40485, 0.59515, 0)
  -- Elapsed time: 0.656 sec

[FFTLines]:
  -- Frequency = 0.050943
  -- Phase = -69.7365
  -- Central line is at 316 out of 640 pixel columns
  -- Period of lines = 19.6296
  -- Elapsed time: 0.172 sec

[FindLinesAndSpots]:
  -- Number of columns = 27
  -- Number of accepted spots = 311
  -- Elapsed time: 161.469 sec

[FindColours]:
  -- Cluster centres (R,G,B):
        cluster 1: (57.5196, 35.1665, 30.2921)
        cluster 2: (61.4102, 19.435, 21.7517)
        cluster 3: (35.8445, 28.313, 24.4159)
  -- Elapsed time: 0.234 sec
```

```
[CorrectColours]:
  -- Number of corrected spots = 61 out of 311
  -- Elapsed time: 0.578 sec

[BuildGrid]:
  -- Number of horizontal rows of spots = 18
  -- Elapsed time: 0.359 sec

[BuildSurface]:
  -- X-ray machine's angle of rotation = -60 degrees
  -- Elapsed time: 0.063 sec

Final Statistics:
-----------------------------------------------------------
Total CONTOURS execution time: 166.235 sec
```

In this example we had only one graphical file to process. If there are more files, the information about each of them is printed during program execution. During the execution, the files with computed variables were written into the MATFiles directory. Next time, the program is able to read them. If you run the code for the second time, you will notice that the execution time reduced significantly, as in the output displayed below.

```
======================== CONTOURS =========================
 Surface reconstruction from noisy structured-light images
===========================================================

[ReadDataFileNames]:
  -- Data directory: D:\_Thesis_\Contours\_Data_
  -- 1 file(s) read: "face.tif"
  -- Elapsed time: 0.016 sec

Image #1   :   face.tif
```

```
----------------------------------------------------------

[ReadData]:
  -- Resolution: 480 x 640 pixels
  -- Elapsed time: 0.203 sec, MAT file is used!

[FilterData]:
  -- Filter type: gauss
  -- Filter size: 5 x 5 pixels
  -- Elapsed time: 0.484 sec, MAT file is used!

[ConvertToGrey]:
  -- Coefficient vector (r,g,b) = (0.40485, 0.59515, 0)
  -- Elapsed time: 0.609 sec

[FFTLines]:
  -- Frequency = 0.050943
  -- Phase = -69.7365
  -- Central line is at 316 out of 640 pixel columns
  -- Period of lines = 19.6296
  -- Elapsed time: 0.031 sec, MAT file is used!

[FindLinesAndSpots]:
  -- Number of columns = 27
  -- Number of accepted spots = 311
  -- Elapsed time: 0.078 sec, MAT file is used!

[FindColours]:
  -- Cluster centres (R,G,B):
         cluster 1: (57.5196, 35.1665, 30.2921)
         cluster 2: (61.4102, 19.435, 21.7517)
         cluster 3: (35.8445, 28.313, 24.4159)
  -- Elapsed time: 0.046 sec, MAT file is used!

[CorrectColours]:
  -- Number of corrected spots = 61 out of 311
  -- Elapsed time: 0.062 sec, MAT file is used!
```

```
[BuildGrid]:
  -- Number of horizontal rows of spots = 18
  -- Elapsed time: 0.062 sec, MAT file is used!

[BuildSurface]:
  -- X-ray machine's angle of rotation = -60 degrees
  -- Elapsed time: 0.031 sec, MAT file is used!

Final Statistics:
-----------------------------------------------------------
Total CONTOURS execution time: 1.622 sec
```

Besides the text output, the program is also capable of outputting over 20 graphs.

Bibliography

[1] *Moire fringe contouring.* 3D-Matic Research Laboratory, University of Glasgow, `http://www.faraday.gla.ac.uk/moire.htm` (9 January 2005).

[2] *Webster's Revised Unabridged Dictionary,* MICRA, Inc., 1996, 1998. `http://www.dictionary.com` (29 December 2004).

[3] *The American Heritage Dictionary of the English Language*, Houghton Mifflin Company, 4th ed., 2000. `http://www.dictionary.com` (29 December 2004).

[4] *Medical Imaging in Canada*, Canadian Institute for Health Information, Ottawa, 2003.

[5] M.-C. Amann, T. Bosch, M. Lescure, R. Myllylä, and M. Rioux, *Laser ranging: a critical review of usual techniques for distance measurement*, Optical Engineering, 40 (2001), pp. 10–19.

[6] J. Bartl, R. Fíra, and M. Hain, *Inspection of surface by the moiré method*, Measurement Science Review, 1 (2001), pp. 29–32.

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

[8] J. A. Berry and J. E. Aldrich, *Surface topography for patient repositioning*, Medical Dosimetry, 16 (1991), pp. 71–77.

[9] R. Buckman, *What You Really Need to Know about Cancer: a Comprehensive Guide for Patients and Their Families*, Johns Hopkins University Press, Baltimore, 1997.

[10] F. Chen, G. M. Brown, and M. Song, *Overview of three-dimensional shape measurement using optical methods*, Optical Engineering, 39 (2000), pp. 10–22.

[11] N. D'Apuzzo, *Modeling human faces with multi-image photogrammetry*, in Three-Dimensional Image Capture and Applications, B. D. Corner, R. P. Pargas, and J. H. Nurre, eds., vol. 4661 of Proceedings of SPIE, 2002, pp. 191–197.

[12] C. J. Davies and M. S. Nixon, *A Hough transform for detecting the location and orientation of three-dimensional surfaces via color encoded spots*, IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, 28 (1998), pp. 90–95.

[13] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley and Sons, Inc., New York, 2001. Chapter 4.

[14] M. Fujita, T. Fujii, T. Kimoto, and M. Tanimoto, *A proposal of 3-dimensional measurement by projecting concentric circle pattern*, in

Videometrics and Optical Methods for 3D Shape Measurement, Proceedings of SPIE, 2001, pp. 251–258.

[15] E. García and H. Lamela, *Low-cost three-dimensional vision system based on a low-power semiconductor laser rangefinder and a single scanning mirror*, Optical Engineering, 40 (2001), pp. 61–66.

[16] C. M. Haskell, ed., *Cancer Treatment*, W. B. Saunders, Philadelphia, 5^{th} ed., 2001.

[17] M. Ito and A. Ishii, *A three-level checkerboard pattern (TCP) projection method for curved surface measurement*, Pattern Recognition, 28 (1995), pp. 27–40.

[18] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall Information and SYstem Sciences Series, Prentice Hall, 1989. Chapter 7.

[19] L. S. Johnson, B. D. Milliken, S. W. Hadley, C. A. Pelizzari, D. J. Haraf, and G. T. Y. Chen, *Initial clinical experience with a video-based patient positioning system*, International Journal of Radiation Oncology, Biology, Physics, 45 (1999), pp. 205–213.

[20] C. L. Kornmehl, *Best News about Radiation Therapy: How to Cope and Survive*, Academic Radiation Oncology Press, Howell, N.J, 2003.

[21] A. O. Korobkine, *Model-based visual tracking via Maple code generation*, master's thesis, Department of Computing and Software, McMaster University, December 2004.

[22] R. Krempien, *Daily patient set-up control in radiation therapy by coded light projection*, Studies in Health Technology and Informatics, 85 (2002), pp. 237–239.

[23] P. Lavoie, D. Ionescu, and E. Petriu, *A high precision 3D object reconstruction method using a color coded grid and NURBS*, in 10th International Conference on Image Analysis and Processing (ICIAP'99), 1999, pp. 370–375.

[24] M. Levoy, *The digital Michelangelo project*, in Proceedings of the Second International Conference on 3D Digital Imaging and Modeling, October 1999. `http://graphics.stanford.edu/papers/digmich-3dimaging99/` (8 January 2005).

[25] A. Lewin, D. Orchard, A. Scott, N. Walton, and J. Austin, *A 3D camera for improved facial recognition*, in Optics and Photonics for Counterterrorism and Crime Fighting, T. P. Donaldson and C. Lewis, eds., vol. 5616 of Proceedings of SPIE, 2004, pp. 77–84.

[26] F. Lilley, M. J. Lalor, and D. R. Burton, *Robust fringe analysis system for human body shape measurement*, Optical Engineering, 39 (2000), pp. 187–195.

[27] I. T. Nabney, *Netlab: Algorithms for Pattern Recognition*, Advances in Pattern Recognition, Springer, 3^{st} ed., 2004.
`http://www.ncrg.aston.ac.uk/netlab/index.php`.

[28] W. Niem and M. Steinmetz, *Camera viewpoint control for the automatic reconstruction of 3D objects*, in Proceedings of ICIP 96, 1996.

[29] R.-S. Pappa, L.-R. Giersch, and J.M.Quagliaroli, *Photogrammetry of a 5m inflatable space antenna with consumer digital cameras*, NASA STI Report Series, NASA/TM-2000-210627, 2000.

[30] R. Pazdur, L. R. Coia, W. J. Hoskins, and L. D. Wagman, eds., *Cancer Management: A Multidisciplinary Approach: Medical, Surgical, and Radiation Oncology*, Oncology Group, New York, 7^{th} ed., 2003.

[31] W. H. Press, *Numerical Recipies in C*, Cambridge University Press, Cambridge, 1999. Chapter 12.

[32] F. B. Prinz, C. L. Atwood, R. F. Aubin, J. J. Beaman, R. L. Brown, P. S. Fussell, A. J. Lightman, E. Sachs, L. E. Weiss, and M. J. Wozny, *Rapid prototyping in Europe and Japan*, panel report, JTEC/WTEC, March 1997. `http://www.wtec.org/loyola/rp/toc.htm` (29 December 2004).

[33] C. Rocchini, P. Cignoni, C. Montani, P. Pingi, and R. Scorigno, *A low cost 3d scanner based on structured light*, in EUROGRAPHICS 2001 Proceedings, A. Chalmers and T. M. Rhyne, eds., vol. 20, Blackwell Publishing, Oxford, 2001, pp. 299–308.

[34] J. C. Russ, *The Image Processing Handbook*, CRC Press, 4^{th} ed., 2002. Chapter 1.

[35] J. Salvi, J. Pages, and J. Batlle, *Pattern codification strategies in structured light systems*, Pattern Recognition, 37 (2004), pp. 827–849.

[36] R. C. SEMELKA, *Radiation risk from CT scans: A call for patient-focused imaging*, Medscape Radiology, 6 (2005). `http://www.medscape.com/viewarticle/496297` (29 June 2005).