

# **The Development of a 3D Imaging and Analysis Application for Medical Endoscopes**

*A dissertation submitted in partial fulfilment of the requirements  
for the Open University's Master of Science Degree in  
Computing for Commerce and Industry*

by

**Roger Steven Rowland PGDCCI(Open)**

*Personal Identifier - M5827130*

**17<sup>th</sup> August 1999**

*14,545 words*

## **Preface**

This exciting and rewarding project was instigated by Professor Duncan Bell, Consultant Gastroenterologist at Sunderland Royal Hospital. The work would not have been possible without the innovative research performed by Dr. John Bladen, again instigated by Professor Bell, in developing the technique of imaging medical endoscopes in three dimensions using magnetic fields. The system undergoing enhancement in this project uses data captured from Dr. Bladen's imaging system.

I would like to thank John Bladen for revealing and explaining the file formats used by his system and for his assistance with the implementation of the curve fitting algorithm. I would also like to thank Duncan Bell for his continuous enthusiastic support and practical assistance in conducting some of the more detailed testing. Last but not least of course, I would like to thank Dr. Mark Fisher - my supervisor for this project - for his advice, suggestions, corrections and comments which have helped greatly to keep this dissertation on the correct path.

## Table of Contents

<b>CHAPTER 1 INTRODUCTION .....</b>	<b>8</b>
1.1 INTRODUCTION.....	8
1.1.1 Overview .....	8
1.1.2 Structure of the dissertation .....	9
1.2 COLORECTAL CANCER.....	10
1.2.1 Incidence and effects.....	10
1.2.2 Cancer screening .....	13
1.2.3 Discussion.....	16
1.3 COLONOSCOPY .....	16
1.3.1 Problems with colonoscopy .....	16
1.3.2 Addressing the problem of loop-formation .....	18
1.3.3 Discussion.....	19
1.4 AIM AND OBJECTIVES .....	20
1.4.1 Project aim.....	20
1.4.2 Objectives.....	20
<b>CHAPTER 2 MEDICAL IMAGING AND COLONOSCOPY .....</b>	<b>22</b>
2.1 MEDICAL IMAGING.....	22
2.1.1 Imaging in medical applications.....	22
2.1.2 Virtual colonoscopy .....	24
2.1.3 The Bladen imager.....	25
2.1.4 Summary .....	26
2.2 THE COLONOSCOPE.....	26
2.2.1 Colonoscope design .....	26
2.2.2 Aids to colonoscopy .....	28
<b>CHAPTER 3 MODELLING THE COLONOSCOPE .....</b>	<b>31</b>
3.1 3D MODELLING TECHNIQUES.....	31
3.1.1 3D graphics software.....	31
3.1.2 Co-ordinate systems.....	32
3.1.3 3D objects .....	37
3.1.4 Viewing systems .....	40
3.2 BUILDING THE MODEL .....	41
3.2.1 Bladen imager data.....	41
3.2.2 Internal structure .....	43
3.2.3 Rendering and display .....	50
<b>CHAPTER 4 MAKING MEASUREMENTS USING THE MODEL.....</b>	<b>52</b>
4.1 REQUIREMENTS DEFINITION .....	52
4.1.1 Functional requirements .....	52
4.2 OUTLINE DESIGN .....	53
4.2.1 Length measurement strategy .....	53
4.2.2 User interface design .....	56
4.3 DETAILED DESIGN .....	63
4.3.1 Data requirements.....	63
4.3.2 Function requirements .....	64
4.3.3 Measurement module design.....	66
4.3.4 Implementing the picking function .....	69
<b>CHAPTER 5 EVALUATION OF MEASUREMENT ACCURACY .....</b>	<b>72</b>
5.1 THE SYSTEM IN USE .....	72
5.1.1 Initiating measurement mode.....	72
5.1.2 Selecting the first point .....	73
5.1.3 Selecting the second point.....	74
5.1.4 Colonoscope length.....	75

5.2	EVALUATION.....	77
	5.2.1 <i>Methods</i> .....	77
	5.2.2 <i>Results</i> .....	80
5.3	COMMENTS .....	84
	5.3.1 <i>Straight-line distance measurement</i> .....	84
	5.3.2 <i>Measurement of path length</i> .....	85
<b>CHAPTER 6 CONCLUSIONS.....</b>		<b>87</b>
6.1	PROJECT REVIEW .....	87
	6.1.1 <i>Achievement against objectives</i> .....	87
	6.1.2 <i>Discussion</i> .....	90
6.2	FUTURE APPLICATIONS.....	92
	6.2.1 <i>Medical</i> .....	92
	6.2.2 <i>Communication</i> .....	92
	6.2.3 <i>Sport</i> .....	93
<b>REFERENCES.....</b>		<b>94</b>
<b>APPENDIX A - CURVE FITTING ALGORITHM.....</b>		<b>98</b>
<b>INDEX .....</b>		<b>102</b>

## **List of tables**

Table 2-A – Two-dimensional medical imaging techniques .....	22
Table 2-B – Three-dimensional medical imaging techniques.....	23
Table 4-A – Data table for the measurement module .....	64
Table 4-B – Procedure and function specifications .....	65
Table 5-A – Results of inter-marker distance tests .....	80
Table 5-B – Results of path length to sensor position tests .....	82
Table 5-C – Results of path length to tip tests .....	83

## List of figures

Figure 1-A – The colon (large intestine).....	11
Figure 1-B – Colonoscope with wire snare and a polyp inside the colon.....	12
Figure 2-A – An enhanced ‘virtual colonoscopy’ image.....	24
Figure 2-B – The controls of a colonoscope.....	27
Figure 2-C – Effect of using a stiffening overtube during colonoscopy.....	29
Figure 3-A – Object co-ordinate system.....	33
Figure 3-B – World co-ordinates.....	33
Figure 3-C – Eye co-ordinates and view plane.....	34
Figure 3-D – Sensor position and orientation (axes rotated for clarity).....	44
Figure 3-E – Creating mitred joints.....	45
Figure 3-F – Octagonal segmentation.....	47
Figure 3-G – Defining polygonal surfaces.....	47
Figure 3-H – The wireframe model.....	48
Figure 3-I – Generating surface normals.....	49
Figure 3-J – Fully rendered colonoscope image.....	51
Figure 4-A – State transition diagram.....	59
Figure 4-B – Functional decomposition diagram.....	61
Figure 4-C – Data flow diagram.....	62
Figure 4-D – Mapping prism selections to length offsets and vertices.....	70
Figure 5-A – Initiating measurement mode.....	72
Figure 5-B – Selecting the first measurement point.....	74
Figure 5-C – Selecting the second measurement point.....	75
Figure 5-D – Colonoscope length.....	76
Figure 5-E – Inter-marker distances displayed on the statistics screen.....	77
Figure 5-F – Benchmark for straight-line distance evaluation.....	78
Figure 5-G – Correlation of actual and calculated straight-line distances.....	81
Figure 5-H – Correlation of actual and calculated path length – 1.....	82
Figure 5-I – Correlation of actual and calculated path length – 2.....	83

## **Abstract**

Colorectal cancer (cancer of the colon or rectum) is a condition with high morbidity and mortality. Colonoscopy – the examination of the large bowel using a medical endoscope – is widely considered to be the ‘gold standard’ diagnostic and therapeutic technique in the fight against colorectal cancer. However, colonoscopy has a number of problems which are not easy to overcome. One of these is the difficulty of performing a full colonoscopy with the colonoscope tending to form into painful loops and twists during insertion. This project describes enhancements to a computer system which attempts to address this problem by providing quantitative data about the behaviour of the colonoscope during colonoscopy. It is hoped that this information may be of use to instrument manufacturers to design improved colonoscopes or accessories or to provide information to assist in the education of medical staff.

This project shows that it is possible to make reasonably accurate measurements of various sections of the colonoscope while inside the patient’s body. The system developed in this project has since been evaluated further with results suggesting that the technique is sufficiently sensitive to detect differences between male and female patients (Rowland et al., 1999).

# Chapter 1 Introduction

## 1.1 Introduction

### *1.1.1 Overview*

This dissertation describes the development and evaluation of enhancements to a computer system designed to aid the difficult and lengthy process of colonoscopy. Colonoscopy is the medical term describing an examination of the large bowel using a medical endoscope. Colonoscopy is a particularly important technique in the diagnosis and treatment of colorectal cancer.

A medical endoscope is a flexible snake-like instrument with the ability to allow the operator to view an image from a camera or lens at the endoscope tip. It is typically introduced into the patient via an existing body orifice or, in some cases, a small incision. Most endoscopes have facilities for performing minor surgical procedures using a selection of surgical tools passed through a hollow channel within the instrument body (the biopsy channel). Medical endoscopes are available in a wide range of dimensions depending on the intended use and the part of the body to be examined. An endoscope optimised for colonoscopy is often referred to as a *colonoscope*. It is introduced into the patient via the anus.

This document details the development and evaluation of additional functionality for an existing computerised imaging system designed to aid colonoscopy. The imaging system uses data captured from sensors placed in the biopsy channel of a

colonoscope to produce a three-dimensional image of the instrument on a computer screen.

### *1.1.2 Structure of the dissertation*

The remainder of this chapter sets colonoscopy in context as a valuable diagnostic and therapeutic technique in the fight against colorectal cancer. The incidence and effects of the disease are noted and alternative screening methods discussed. There follows a summary of some of the major problems with colonoscopy, focusing in particular on the problem of loop-formation. At the end of this chapter, the imaging system to be enhanced is introduced and the aim and the objectives of the project are stated.

Chapter 2 outlines some alternative medical imaging techniques with respect to both colorectal cancer screening and possible application to the problem of loop-formation during colonoscopy. The chapter concludes with a more detailed description of the colonoscope itself and the imaging system to be enhanced.

Chapter 3 begins with a general description of the tools and methods available for 3D computer modelling. There follows a detailed description of the methods used to produce a 3D image from the colonoscope imager data. This is the starting point for the enhancements to the system which are the main focus of this work.

Chapter 4 concentrates on the implementation of enhancements to the imaging software. The functional requirements are listed and the system design detailed. The methods used to implement the enhancements are also described.

Chapter 5 covers the evaluation of the added functionality using bench test techniques. The methods employed and results obtained are documented and discussed.

Chapter 6 concludes the document with a review of achievement against objectives, a critical appraisal of the project and a discussion of potential future modifications or uses in a wider context.

## **1.2 Colorectal cancer**

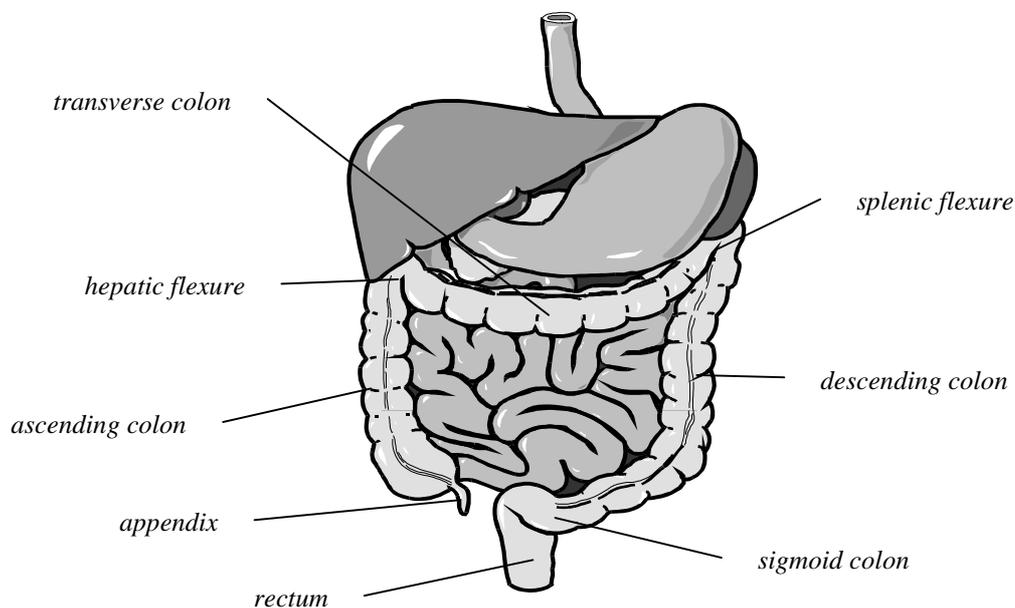
### *1.2.1 Incidence and effects*

Colorectal cancer – cancer of the colon or rectum – is a common condition with high mortality. Bladen (1995) includes the following quotation in his introductory chapter:

*“Colorectal cancer is the second most common cause of death due to malignancy in England and Wales. 25000 new diagnoses are made each year, of which 17000 will die. Colorectal cancer accounts for 10% of all cancer deaths and is responsible for 3% of all deaths in the UK. About one in twenty of the British population will get colorectal cancer and only about one-third will survive five*

*years or more*” Kettner (1989). The latest available mortality statistics for the United Kingdom are for 1997. In that year, colorectal cancer accounted for 11% of all cancer deaths – some 17,300 people (Anon. 1999).

The colon is the section of large bowel extending from its junction with the small intestine (near the appendix) to the anus. The various sections of the colon are named as shown in Figure 1-A. Following the normal course of material through the large bowel, the journey ‘begins’ in the ascending colon, progresses around the hepatic flexure to the transverse colon, passes the splenic flexure to the descending colon and finally around the sigmoid colon to the rectum before reaching the outside world via the anus.

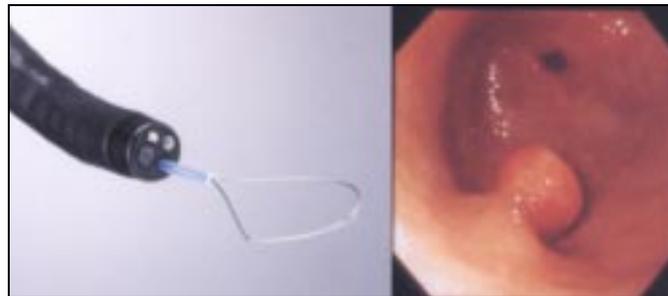


**Figure 1-A – The colon (large intestine)**

Colorectal cancers are thought to develop from *polyps*. These are small, initially benign growths on the inner wall of the colon, some of which may turn cancerous

over a number of years. Khullar and DiSario (1997) have demonstrated that early detection and removal (polypectomy) of these precursors to cancer, as well as removal of early localised cancer itself, can reduce the incidence and mortality of the disease.

Polyps may be removed during colonoscopy by a number of methods. One of these involves the use of a wire 'snare' passed through the biopsy channel of the colonoscope. The snare is looped around the polyp and then electrically heated to effect the removal and cauterise the wound. Figure 1-B shows the snare protruding from the colonoscope tip and an image of a polyp as seen through the optics of the instrument.



**Figure 1-B – Colonoscope with wire snare and a polyp inside the colon**

Polyps are not in themselves dangerous, are very slow to form and during colonoscopy relatively easily removed. A one-off 'harvesting' of polyps may be all that is required to drastically reduce the incidence of colorectal cancer. There is a great deal of evidence that some form of colorectal cancer screening programme would be effective in this respect.

### 1.2.2 Cancer screening

In recent years, there has been an increased recognition of the prevalence and effects of colorectal cancer, particularly in the developed West where the disease is quite widespread. Presently, in the UK, the Government is not only piloting the introduction of a 'home-test' screening program but also providing additional funding "... *to improve the quality and availability of services for bowel cancer ...*" (Anon. 1998).

Polyps are especially common in older people and may take some years to become cancerous. It is because of this that Saunders (n.d.) notes - "*a therapeutic window exists when polyps can be removed to prevent cancer*". Saunders goes on to remark that a simple, safe and cheap screening test would be the ideal method of detecting polyps but that none at present exists. The home-test kit noted above relies on faecal occult blood testing which is described in more detail overleaf.

A number of methods have been proposed, either singly or in combination, as potential candidates for colorectal cancer screening. Among these are the following –

**Digital rectal examination** – *An examination performed by a doctor, with gloved hand, inserting a finger into the rectum and feeling for abnormalities. This is not a serious contender for screening due to the very small proportion of the large bowel examined and inherent difficulty in detecting anomalies. Robinson and Hardcastle (1998) note that digital rectal examination will detect no more than 10% of cancers.*

**Faecal occult blood testing (FOBT)** – *This is a test performed on patients' stools (faeces) to detect hidden (occult) blood. There is an average loss of blood through the gastrointestinal tract of about 1ml per day (Robinson & Hardcastle, 1998). The early development of cancerous tumours increases the rate of blood loss which may then be detected with this test. Kronborg et al. (1996) report a decrease in mortality of 18% by using FOBT screening every two years over a ten year period. The same study estimated colorectal cancer detection rates of up to 50% using this technique. Whilst this form of screening is non-invasive and relatively inexpensive, it is still not ideal that this test will only discover cancers after they have formed.*

**Rigid sigmoidoscopy and Flexible sigmoidoscopy** – *Both of these techniques provide a method of screening the rectum and sigmoid colon only. In the first case, a rigid sigmoidoscope is used to examine the first 16cm or so of the rectum and sigmoid colon. Robinson & Hardcastle (1998) estimate that at most 40% of all colorectal cancers would be detectable in this way. On the other hand, flexible sigmoidoscopy examines some 30 – 60 cm where approximately 70% of cancers are found. While both techniques are invasive and uncomfortable, they are usually carried out with no sedation and are less labour intensive than full colonoscopy. It still remains, however, that 30% of all colorectal cancers would never be detected by this method.*

**Double contrast barium enema (DCBE)** – *This involves coating the inside of the colon with a barium suspension mixture introduced via the anus. Barium is opaque to x-rays and allows the inside surface of the colon to be examined for*

*abnormalities. Although safer and cheaper than colonoscopy, this technique is invasive and unpleasant as it requires mechanical bowel preparation. It is also much less sensitive than colonoscopy and exposure to x-rays is necessary. According to Robinson & Hardcastle (1998), few studies have evaluated it as a screening test.*

**Virtual colonoscopy** – *This is a relatively new technique involving the computer-aided reconstruction of a ‘virtual’ colon using 3D data from a CT (Computer-aided Tomography) or MRI (Magnetic Resonance Imaging) scanner. Sometimes called ‘CT colography’ or ‘CT colonography’, this technique is both accurate and non-invasive, it will identify some pre-cancerous polyps as well as developed tumours. The technique covers the whole of the colon and, according to the Mayo Clinic (1996), is less than half the cost of a full colonoscopy. This obviously has potential as a screening aid.*

**Colonoscopy** – *This involves examination of the whole of the colon using a colonoscope. During the procedure, the patient is partially sedated while the colonoscopist introduces the colonoscope via the anus and manipulates it until fully inserted into the colon (some 1.6 metres). The examination is made using either a fibre-optic view through a lens at the tip of the instrument or a more expensive miniature video camera. This technique has the advantage that any polyps (or other abnormalities) discovered may be treated or removed as part of the same procedure. Colonoscopy is widely recognised as the “gold standard” method for examination of the large intestine.*

### 1.2.3 Discussion

For a colorectal cancer screening programme to be effective, several factors need to be assessed. On one side is the practicality and cost of the chosen method, on the other its effectiveness. In most cases, the less expensive methods (FOBT or DCBE) are also less effective and vice versa. One notable exception is the advent of virtual colonoscopy which, as is the case with many new technologies, will probably become cheaper as it becomes more widely available.

However, virtual colonoscopy is purely a *diagnostic* tool and would not remove the need for trained colonoscopists to perform polypectomies and other therapeutic procedures. Indeed, if anything, it may possibly *increase* the demand for *therapeutic* colonoscopy as more treatable abnormalities are discovered. It appears that colonoscopy will remain an important procedure in the fight against colorectal cancer for the foreseeable future.

Unfortunately, colonoscopy has its own problems. Some of which are more difficult to address than others.

## **1.3 Colonoscopy**

### 1.3.1 Problems with colonoscopy

Colonoscopy is expensive. A typical colonoscope currently costs in the region of £15,000 to £20,000 and may need to be supported by ancillary equipment for displaying and recording video images from its remote optics. The procedure

required for sterilisation of colonoscopes is complex and the chemicals used are hazardous. This adds to the overall cost.

Colonoscopy is not an easy skill to master. Parry & Williams (1991) have estimated that it takes about 200 supervised procedures for a trainee in gastroenterology to become technically even reasonably proficient. Saunders (n.d.) supports this analysis and adds that the teaching process is “*time-consuming and frustrating for teacher and student alike*”. The effective use of a colonoscope involves a considerable amount of manual dexterity which may only be mastered through practice.

As the size and elasticity of the colon varies widely amongst individuals, there are no set rules for learning how to successfully intubate (insert into a patient) a colonoscope. The average instrument is some 1.6 metres in length and can form painful loops and twists during insertion. As the colonoscopist is usually working ‘blind’, he/she has to rely very much on experience to determine whether and what type of loops have formed and what combination of twisting or pulling may be required to remove them. Even in the hands of an experienced colonoscopist, this only adds to the patient’s discomfort. For this reason, the patient is usually partially sedated and requires a period of recovery before being discharged. This not only complicates and lengthens the procedure but also adds to the cost.

The inability to see exactly the position and configuration of the colonoscope is one of the most important factors affecting the difficulty of a colonoscopy. As Saunders remarks “*Unpredictable, recurrent looping of the colonoscope in a*

*mobile or long colon is the most common cause of difficulty at colonoscopy ...”.*

Saunders also reports that women are typically more difficult to examine than men and that there are significant differences in the complexity of the colon between racial groups.

Although a good quality image can be obtained from the remote lens at the colonoscope tip, the characteristics of the colon are such that it is not always easy to determine the position of the instrument from this image alone. Cotton and Williams (1996) suggest that “ ... *gross errors in localization are possible during the insertion of the colonoscope in up to 30% of cases, even for expert endoscopists.*” Combined with the fact that instrument looping may give an exaggerated impression of insertion depth, it follows that it is often difficult to ensure that the *whole* of the colon has been traversed without some alternative means of visualising the instrument in the patient’s body.

### *1.3.2 Addressing the problem of loop-formation*

In 1993, John Bladen – a PhD student at Sheffield University - and colleagues developed a system to allow computerised three-dimensional imaging of a colonoscope using low-power magnetic fields and a chain of sensor coils inserted into the biopsy channel of the instrument (Bladen et al., 1993).

The system analyses the signals obtained from the coils to determine the 3D coordinates and orientations of each sensor. This data is then used to reconstruct a three-dimensional image of the shape of the colonoscope on a computer screen.

Reference points on the patient's body are marked prior to the examination and are also displayed on screen.

The ability to visualise in real-time the position and configuration of the instrument relative to fixed markers on the patient's body aids greatly in the identification and removal of loops and in assessing the progress of the colonoscope. The system also allows electronic capture of this positional data, which could prove useful for retrospective analysis or teaching purposes.

The 'Bladen system' – described above – is used as a source of data for the imaging software to be enhanced in this project.

### *1.3.3 Discussion*

The obvious point emerging from the preceding notes is that loop-formation during colonoscopy is undesirable for a number of reasons. In order to see how instrument design could be improved to minimise this, we need to be able to extract some detailed measurements of what actually happens to the colonoscope during intubation.

We know that loops tend to form more readily where the patient has a long or complex colon. Saunders (n.d.) has noted correlation with gender and/or racial origin. Women tend to have longer colons than men, and Western colons are generally more complex than Oriental. In such cases, it seems logical to suggest that if the design of the colonoscope used in terms of dimensions and flexibility

were more closely adapted to match the patient, then this could make a difference to the length of the procedure, its effectiveness and the level of discomfort caused.

However, we have few statistics covering the characteristics of the colon in living subjects or, more importantly, the behaviour of the colon and/or colonoscope while inside the body.

In an attempt to capture such statistics, this project covers the enhancement of a recently-developed computer system for the 3D rendering and display of the colonoscope image (Rowland & Bell, 1998). The new computer system (known as the 'RMR' system) uses stored data from Dr. Bladen's imaging system.

## **1.4 Aim and objectives**

### *1.4.1 Project aim*

**The aim of this project is to provide a method of obtaining accurate measurements of sections of a colonoscope using data captured during colonoscopy.**

### *1.4.2 Objectives*

The detailed objectives of this project are as follows:

- *To research the background to the problem area to confirm the need for such an application and to introduce the imaging system to be enhanced.* } **Chapters 1, 2**
- *To describe the tools and techniques used for 3D computer imaging of the colonoscope.* } **Chapter 3**

- *To perform a requirements analysis for the measurement part of the system.*
  - *To produce a system design. The design will focus on the additional functionality required with regard to measurement of lengths.*
  - *To produce a working system based on the above design suitable for evaluation.*
- Chapter 4**
- *To evaluate the performance and accuracy of the new features of the system.*
- Chapter 5**
- *To identify strengths and weaknesses of the new system, produce a comparison with alternative systems, if any, and to propose a plan for future development.*
- Chapter 6**

## Chapter 2 Medical imaging and colonoscopy

### 2.1 Medical imaging

#### 2.1.1 *Imaging in medical applications*

A number of medical imaging systems are in common use. The majority of these systems produce images which are two-dimensional in nature, either as 2D projections of 3D distributions or as tomographic images (i.e. a two-dimensional 'slice' through the body or organ). Table 2-A summarises some of the more common 2D imaging techniques.

<b>Imaging method</b>	<b>Comments</b>
X-ray imaging	<i>Measures transmission of x-rays through the body. Different tissues attenuate the x-rays to different degrees, the transmitted rays form the basis of an image on a photographic plate.</i>
Fluoroscopy	<i>Real-time version of static x-ray above. Instead of a photographic plate, chemicals in an x-ray camera fluoresce when struck by the transmitted rays. The resulting image is displayed as a monochrome TV picture.</i>
Ultrasound	<i>Tomographic images are produced by analysing variations in the velocity and attenuation of ultrasound (high frequency sound waves) resulting from differing tissue densities. Both transmitter and receiver must be in contact with the body.</i>
Electrical impedance tomography (EIT)	<i>Tomographic images are produced by analysing variations in conductivity in a slice of the body. A set of electrodes defines the plane of the 'slice', the number of electrodes used determining the resolution.</i>

**Table 2-A – Two-dimensional medical imaging techniques**

Three-dimensional imaging techniques are generally more complex and expensive. Most of these rely on computer-assisted reconstruction of a series of tomographic images from an existing 2D imaging method.

Most medical imaging methods are primarily designed to provide details of the different characteristics of body tissues. Other techniques (PET and ECT, described below) provide a method of imaging body functions. These techniques are not necessarily as effective when attempting to image a metal instrument inside the body. Table 2-B summarises some of the more common 3D imaging techniques in use.

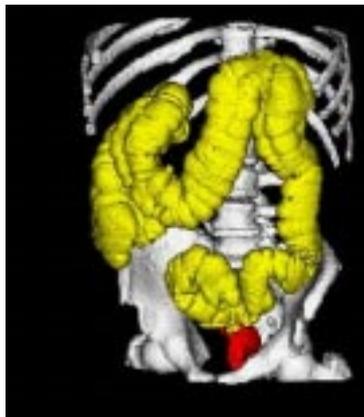
<b>Imaging method</b>	<b>Comments</b>
X-ray computed tomography (CT)	<i>Conventional x-rays are used to produce a series of images as 'slices' through the body. These images are processed by computer to reconstruct a 3D image.</i>
Magnetic resonance imaging (MRI)	<i>The subject is exposed to a large magnetic field which affects the spin of protons in water molecules. Differing water content of body tissue allows tomographic images to be captured from which a 3D reconstruction may be computed.</i>
Positron emission tomography (PET) Emission computed tomography (ECT)	<i>The subject is injected with a substance containing a radioactive isotope which is designed to locate itself in particular organs or tissue types. Radiation cameras detect the emissions from the isotope to produce a density map. The resulting data may then be reconstructed to give a 3D view (Emission computed tomography – ECT).</i>

**Table 2-B – Three-dimensional medical imaging techniques**

### 2.1.2 Virtual colonoscopy

Recent developments in CT and MRI scanners have raised the possibility of a non-invasive method of imaging the colon in fine detail. So-called ‘Helical scanning’ CT and MRI imagers are able to produce much thinner ‘slices’ (of the order of 2mm or better) through the body and so result in a more accurate 3D reconstruction when processed by computer.

In order to produce an image of the inside of the colon, it is first necessary to fill the colon with a medium suitable for imaging. For the helical CT scanner, carbon dioxide gas is used, while for the MRI scanner a single-contrast enema in the form of two litres of water spiked with 20ml of Gd-DTPA (gadolinium diethylenetriaminepentaacetic acid) is employed (Debatin et al., 1999). Figure 2-A shows a computer-enhanced image from such a scan in which a tumour has been identified (coloured red).



**Figure 2-A – An enhanced ‘virtual colonoscopy’ image**

Debatin et al., (1999) report that *‘The diagnostic performance of virtual colonoscopy regarding polyp detection is directly related to the size of the polyps. Polyps larger than 10mm in diameter can be reliably identified ...’*.

As this technique is still in its infancy, this performance is likely to improve in future and this type of imaging will ‘... *open up tremendous diagnostic opportunities in the assessment of the colon*’.

### 2.1.3 *The Bladen imager*

The Bladen imaging system, introduced in section 1.3.2, is specifically designed to provide an image of the colonoscope rather than an image of the patient’s body. It uses low-power magnetic fields to derive the position and orientation of sensor coils placed in the biopsy channel of the instrument. The current system uses 12 sensors, each approximately 2mm wide and 16mm long formed into a ‘chain’ inside a conventional medical catheter. This ‘sensor catheter’ fits easily into the biopsy channel of the colonoscope and is wired directly to the imaging system while the colonoscopy is performed. The sensor catheter may be temporarily removed if the biopsy channel is required for other procedures.

Electric currents are induced in the sensors by the magnetic fields. The magnetic fields are generated by three field-generator assemblies, each of which comprises three orthogonal coils. These generators are arranged in a triangular formation on a flat wooden (non-permeable) board which is placed under the patient’s bed (also made of wood). Electric current is fed to each of the nine coils (three coils in each of three generators) in sequence by the imaging system hardware. Software within the system then analyses the resulting sensor output and derives the position and orientation of each sensor in terms of  $(x, y, z)$  co-ordinates and  $(\theta, \phi)$  rotations. These data, along with the physical characteristics of the sensor catheter and

various calibration data, are fed to a conventional personal computer on which a three-dimensional image of the endoscope is rendered in real time.

In addition to the real-time display, the system stores the positional data from each sensor once every second. This data may be transferred to floppy disk for later input to another computer. It is this data which is used by the so-called 'RMR System' (Rowland and Bell, 1998) to render a very high quality image and provide off-line analysis and review of the entire procedure. The RMR system is the application being enhanced in this project to provide reliable length measurement and analysis from captured data.

#### *2.1.4 Summary*

Whilst there are potentially useful imaging techniques for diagnosis or screening purposes, only the Bladen system is really suitable for providing an image of the colonoscope to assist the colonoscopist in the execution of the procedure itself. In addition, the nature of the data captured by the Bladen system readily lends itself to the type of analysis which is the subject of this dissertation.

## **2.2 The Colonoscope**

### *2.2.1 Colonoscope design*

Colonoscopes are available in a variety of shapes and sizes, largely depending on the section of bowel to be examined. Generally, the shorter instruments tend to be thicker and/or less flexible whilst the longer instruments tend to be thinner and/or

more flexible. There are also paediatric endoscopes which are more suitable for a child's anatomy.

A colonoscope intended to examine the whole length of the adult colon would typically be some 1.6 metres long and 1.5 cm in diameter. Shorter instruments may be used where complete colonoscopy is not necessary. The flexible sigmoidoscope, for example, is used to examine the sigmoid colon (the length of bowel nearest to the rectum) and is typically around 60 cm in length.

The design of the controls of a colonoscope has changed little in the last 25 years (Saunders, n.d.), the flexible tip being manipulated by two thumbwheels connected by wire along the length of the instrument, with the colonoscopist providing the push/pull and torque forces required to advance the tip along the colon. Figure 2-B shows a picture of the colonoscope controls. Access to the biopsy channel is made via the hole shown in the lower right of the picture.



**Figure 2-B – The controls of a colonoscope**

The top 11 cm or so of the colonoscope tip is more flexible than the rest of the instrument and may be ‘steered’ in two planes by using thumbwheels attached by wires within the colonoscope body. This is not only essential during the examination itself but may be used to temporarily ‘anchor’ the colonoscope while a twisting and/or pulling action is used to remove any loops formed during intubation.

In a situation where the colonoscope is formed into loops, not only does it take more effort to manipulate the thumbwheels, but the resolved component of the push/pull forces transmitted to the tip becomes progressively less as the colonoscope becomes more convoluted (Mosse et al., 1997, 1998).

### 2.2.2 *Aids to colonoscopy*

With the above problems hampering colonoscopists for many years, one might imagine a variety of aids would be available to ease the procedure. But, as Cotton and Williams (1996) point out – *‘The only specialized accessory in colonoscopy is the stiffening tube, stiffener or split overtube ... it is still very occasionally invaluable in avoiding recurrent loop formation of the sigmoid colon ...’*.

As the name implies, the stiffening overtube is a relatively rigid ‘splinting device’ which is placed over the colonoscope shaft once the sigmoid colon has been completely straightened and the tip of the instrument is in the descending colon or splenic flexure. With the overtube in place, the possibility of loops reforming in

the sigmoid colon is avoided and the colonoscopist can concentrate on navigating the remainder of the bowel.

There are two basic designs in common use; the newer, split overtube has a slit along its complete length and may be ‘wrapped’ around the colonoscope shaft at a convenient time, the slit being sealed with a type of adhesive tape, whilst the older, sometimes wire-reinforced overtube needs to be ‘muzzle-loaded’ onto the instrument before the procedure commences.



**Figure 2-C – Effect of using a stiffening overtube during colonoscopy**

Figure 2-C shows the effect of using a stiffening overtube. The diagram on the left shows an unrestrained colonoscope snaking through the sigmoid colon with the instrument tip hooked over the splenic flexure. The diagram on the right shows the possible configuration once the loop has been removed and an overtube inserted. It is evidently much easier to transmit more force to the instrument tip once the colonoscope is straightened and splinted.

The discussion at the end of the first chapter mooted the possibility of quantitative information about the behaviour of the colonoscope during colonoscopy being of use to instrument designers. In a similar vein, the possibility exists that such data may be able to provide information useful for the design of splinting overtubes.

## Chapter 3 Modelling the colonoscope

### 3.1 3D modelling techniques

#### *3.1.1 3D graphics software*

Computer display screens are inherently two-dimensional. To display a two-dimensional representation of a three-dimensional object, and to allow that object to be manipulated on-screen necessitates the use of 3D graphics techniques.

As the mathematics involved in 3D graphics processing is not only complex but also processor-intensive, a number of 3D graphics systems are available which contain highly optimised, low-level routines to which processing may be delegated by an applications program. Such systems generally accept definitions of objects and viewing parameters in 3D terms and automate the transformation and display onto the 2D computer screen.

3D graphics library routines are typically accessed via an API (Application Programmer Interface). This is a strict specification of the data structures and function calls which must be used to access the 3D graphics routines. An application program would normally create and populate any necessary data structures before delegating the more complex processing to the graphics library by executing one or more function calls.

Examples of two 3D graphics systems in common use on Windows 95/NT platforms are Microsoft's *Direct3D* (part of the Microsoft DirectX (n.d.) API) and the *OpenGL* (n.d.) API pioneered by Silicon Graphics Inc.

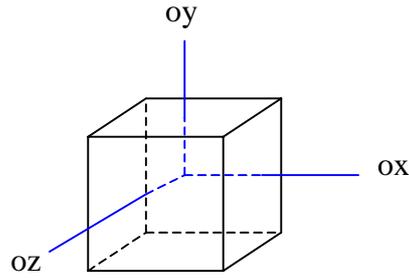
In order to take full advantage of the functionality provided by a 3D graphics library, some basic concepts need to be understood. According to Taylor (1994), there are three areas which underlie any 3D graphics system. These are –

- Co-ordinate systems,
- 3D objects, and
- Viewing systems

### 3.1.2 *Co-ordinate systems*

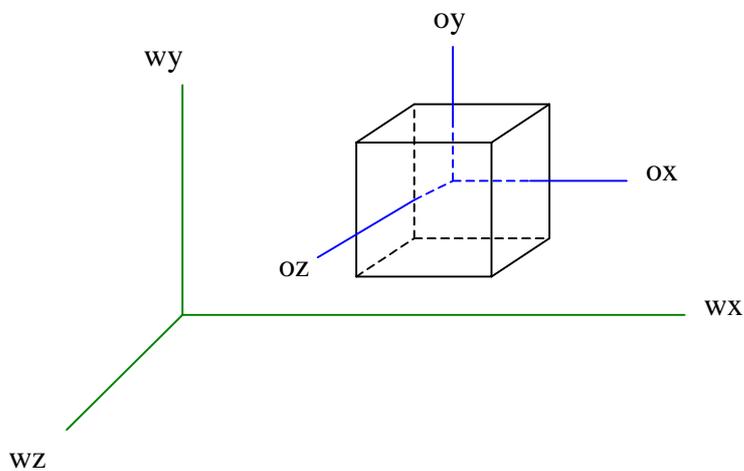
One of the most confusing aspects of 3D graphics systems is the use of multiple co-ordinate spaces. Co-ordinates (individual points) exist within a co-ordinate space. This is a system consisting of a number of dimensions (in this case three), an origin, and a range of values for which the system is defined (Taylor 1994).

3D objects are defined in their own local co-ordinate system, '*object co-ordinates*'. These co-ordinates define the shape, size and default orientation of each object itself. Objects are commonly centred on their own local origin to ease the application of rotation and/or scaling transformations. Figure 3-A (Taylor 1994) shows a simple object – a cube – together with the axes of its object co-ordinate system.



**Figure 3-A – Object co-ordinate system**

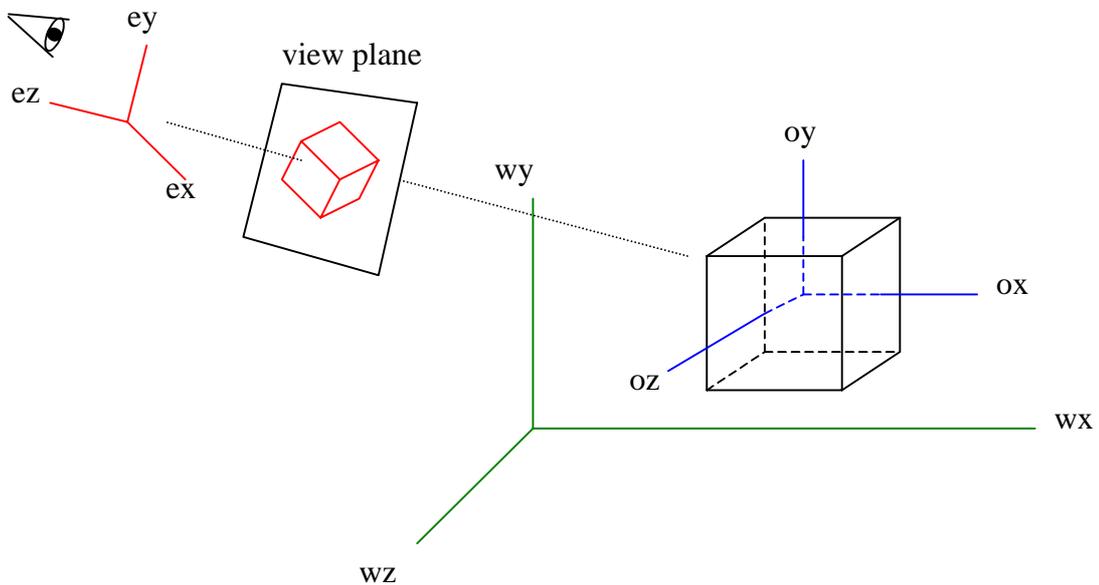
The object co-ordinates of all objects in a scene may be transformed into ‘*world co-ordinates*’, which position, orient and/or scale each object in ‘world’ space. This is the conceptual space where all 3D objects are placed to form the basic scene. Figure 3-B (Taylor 1994) shows an object which has been positioned in world space by a simple translation.



**Figure 3-B – World co-ordinates**

One object of particular interest in a 3D scene is the ‘eye’ or ‘camera’. This is the point from which the observer is deemed to be viewing the scene and, together with other viewing transformations, determines the *viewing volume*. Like any

other object in the scene, the eye has its own local co-ordinate system allowing it to be placed into world space in any orientation. The transformations applied to the ‘*eye co-ordinates*’ determine, among other things, in which direction the observer is looking and which way is ‘up’. Other viewing parameters, like a field of view angle, serve to define both the viewing volume and the projection plane, or ‘view plane’, for the eventual conversion of 3D to 2D co-ordinates. Figure 3-C (Taylor 1994) shows the position and orientation of the observer in eye co-ordinates.



**Figure 3-C – Eye co-ordinates and view plane**

The co-ordinates of objects in world space may be converted from 3D to 2D by a projection transformation onto the viewing plane. The nature of this transformation is controlled by various viewing parameters but essentially involves dividing the  $x$  and  $y$  co-ordinates by some function of the  $z$  co-ordinate.

This projection from 3D to 2D will result in each object being mapped to a two-dimensional view plane as shown in Figure 3-C.

Although this completes the 3D transformation of co-ordinates, one final mapping, the ‘viewport transformation’, is used to map the rectangular view plane to the actual display screen. This may involve linear scaling of the image in two dimensions to accommodate it on the physical device (i.e. enlargement or compression).

To summarise, we have four different co-ordinate systems –

- *Object co-ordinates* Describing the shape and default orientation of each object in the scene
- *World co-ordinates* Used to arrange and size *all* of the objects in the scene
- *Eye co-ordinates* Used to describe the position and orientation of the camera (effectively, another object)
- *Screen co-ordinates* 2D co-ordinates either to a viewport (logical co-ordinates) or a display device (physical co-ordinates) after projection

The creation of a two-dimensional image from a three-dimensional scene involves four transformations –

- *Modelling transformation* Arranging objects in the scene by converting object co-ordinates to world co-ordinates.
- *Viewing transformation* Positioning and orienting the camera by converting eye co-ordinates to world co-ordinates.
- *Projection transformation* Setting the shape of the viewing volume by adjusting field-of-view parameters.

- *Viewport transformation*                      Determining the size of final rendered image by converting logical screen co-ordinates to physical device co-ordinates (pixels).

These transformations are not always defined to the graphics API in the above order. This in no way constrains the order in which the operations are performed within the graphics library itself, as it is possible to perform many optimisations to minimise computation time. Woo et al. (1997) give a useful analogy for illustrating the above transformations which compares the sequence of steps required in an OpenGL application program with the actions required when taking a photograph.

The steps are given as –

1. Set up your tripod and point your camera at the scene (*viewing transformation*).
2. Arrange the scene to be photographed into the desired composition (*modelling transformation*).
3. Choose a camera lens or adjust the zoom (*projection transformation*).
4. Determine how large you want the final photograph to be – for example, you might want it enlarged (*viewport transformation*).

The above methods provide mechanisms for creating a 3D scene and give the ability to view it from different angles and/or magnifications simply by manipulating the position of the ‘camera’ in world space. It is also relatively

straightforward to leave the ‘camera’ in a fixed position but translate and/or rotate individual objects to provide an all-round view. As these objects could conceivably be of any shape or size, there needs to be some mechanism for defining the characteristics of each object to the graphics API.

### 3.1.3 3D objects

A simple, three-dimensional object could be classified by just two characteristics – its geometry (i.e. position and shape), and its surface properties (i.e. colour and texture).

#### **Geometry**

Where an object’s geometry is concerned, the position can be manipulated by applying transformations to the object co-ordinates of the basic shape. For example, a cube standing on one of its corners is still the same shape as a cube standing on one of its flat faces - it has just undergone a rotation transformation. This ability to divorce the position and orientation elements from an object’s geometry leaves one with the task of defining basic shapes to the graphics API.

The simplest way of defining ‘primitive’ shapes is in terms of vertices (‘corners’) and polygons (‘faces’). To describe a cube, one would need eight vertices defining each of the eight corners of the cube and some way of describing six four-sided polygons joining these vertices to describe the six faces. More complex objects can be constructed by using more vertices and polygon definitions. By far the most popular method used to model *curved* surfaces is to define the shape in terms

of a 'mesh' of triangular polygons. The quality of the surface may be controlled by adjusting how finely it is divided into triangles. The triangle is used as the most basic polygon as the three points comprising it, unless co-linear, always define a plane. This makes lighting and shading calculations much easier (and therefore faster) when rendering a solid surface. The performance of hardware-accelerated 3D graphics cards available for personal computers is often rated in terms of millions of triangles per second.

Defining shapes this way to a graphics API is usually achieved by storing the coordinates of each vertex in an array. A separate list then enumerates the vertices involved in polygonal faces by holding indices into the vertex array. The order in which the vertices comprising a polygon are listed is important. The graphics API analyses the positions of the corresponding vertices once the conversion to 2D has been performed and makes an assumption about whether the front or back face of the polygon is showing depending on whether the 2D vertices appear in clockwise or counterclockwise order. Significant performance improvements are possible if the graphics system can safely avoid performing lighting / shading calculations for polygon surfaces which will later be obscured by others. This is known as 'back face culling'.

Primitive shapes defined as above may also be combined in various ways to produce composite objects. This technique is known as Constructive Solid Geometry (CSG) and is provided in some graphics API's. As an example, a CSG 'difference' operation could be applied to a coincident sphere and cylinder to produce an object resembling a ball with a hole drilled through it. A little

forethought can result in some quite complex objects being formed from primitive shapes.

Another method, more computationally intensive, is to define a complex surface in terms of a mathematical equation of a line (often, but not always, curved), then to ‘sweep’ this line around an axis in a manner conceptually similar to carving a shape using a lathe. Because of its greater complexity, this technique is not often used for 3D scenes which are animated in real time. Other curved surfaces may be defined mathematically by using NURBS (Non-uniform Rational B-Splines), details of which are beyond the scope of this document.

### **Surface properties**

The surface properties of an object affect how the graphics API will apply lighting and shading calculations in order to decide the colour of each individual pixel in the final rendered image.

Although not strictly a property of the surface, for curved objects defined in terms of vertices and polygons, one of the most effective options is the ability to define ‘surface normals’ for each vertex in the object. As the name implies, a surface normal is a vector considered to be perpendicular to the surface at that point. By judicious definition of surface normals, fairly rough, blocky shapes can be rendered as smooth surfaces by the graphics system. Surface normals are used during lighting calculations to modify the colour of individual pixels due to reflected light. The surface normal defines the direction in which the maximum amount of incident light will be reflected.

Colour and texture are the most basic properties of any surface. As with surface normals, these properties are often defined per vertex. The colour information for each vertex can be used to interpolate colours for all parts of surfaces between vertices to produce smooth transitions between colours. Most graphics API's provide the ability to do this automatically ('Gouraud shading'). Where this facility is not used, surfaces are said to be 'flat shaded'. Whilst this is much faster to implement, it can make curved surfaces appear unrealistic.

The texture of a surface determines how light is absorbed and/or reflected. Again, often defined per vertex, manipulation of texture properties can make objects appear smooth, 'wet', shiny, metallic etc. A related feature often provided by graphics API's (especially those used for games programming) is the ability to define a 'texture map' for a surface. In this situation, a conventional two-dimensional image – in the form of a bitmap – is conceptually 'wrapped' around the 3D object and used to modulate the lighting / shading calculations. Although not particularly relevant to this project, texture maps can be used to great effect to simulate brickwork, paving or even foliage.

#### *3.1.4 Viewing systems*

Viewing systems manage the set of transformations allowing object definitions, in terms of co-ordinates and surface properties, to become visible pixels. Usually known as the 'viewing pipeline', this functionality is provided by the graphics system and gives the application programmer convenient tools to apply

translation, scaling and rotation transformations without the need to implement complex linear algebra routines in the applications program.

Together, the tools provided by a graphics API such as Direct3D or OpenGL serve to facilitate the production of realistic 3D images in an efficient manner with a minimum of complexity. The RMR system employs the OpenGL API to produce an image of the colonoscope using data from the Bladen imaging system. Within the RMR system, the 3D model is built using a collection of vertices and polygons. Surface normals are used to give a smooth surface in the final rendered image. The methods used to build the model from the source data are detailed in the following section.

## **3.2 Building the model**

### *3.2.1 Bladen imager data*

The data produced by the version of the Bladen imager used are contained in three proprietary files. The files all have a common filename, with the file extension differing. Each 'set' of three files records one colonoscopy. The content and layout of each file may be abstracted follows.

\*.IMG files contain information relating mainly to the sensor configuration :

<b>Field</b>	<b>Type</b>	<b>Description</b>
<i>NumSensors</i>	int	Number of sensors used
<i>SensorSpacing</i>	array[0..17] of real	Inter-sensor spacing

\*.MAR files contain information relating to the marker positions taken to identify fixed points on the patient's body. Each record denotes one marker position :

<b>Field</b>	<b>Type</b>	<b>Description</b>
<i>x</i>	real	x – co-ordinate
<i>y</i>	real	y – co-ordinate
<i>z</i>	real	z – co-ordinate
<i>theta</i>	real	$\theta$ - rotation
<i>phi</i>	real	$\phi$ - rotation

The above file may contain multiple 'sets' of a number of markers. This allows marker positions to be reset during colonoscopy if the patient has moved.

\*.END files contain information relating to the sensor positions as the colonoscopy proceeds. Each record denotes the position and orientation of all sensors at a point in time :

<b>Field</b>	<b>Type</b>	<b>Description</b>
<i>datetime</i>	long	Date/time stamp
<i>points</i>	array[0..15] as follows	Position / orientation of each sensor
<i>x</i>	real	x – co-ordinate
<i>y</i>	real	y – co-ordinate
<i>z</i>	real	z – co-ordinate
<i>theta</i>	real	$\theta$ - rotation
<i>phi</i>	real	$\phi$ - rotation

Data from all three of the above files is processed by the RMR system to generate a 3D model of the endoscope and/or markers for display on a computer screen.

The following section describes how this data is used to build a set of vertices and polygon definitions representing the colonoscope.

### 3.2.2 *Internal structure*

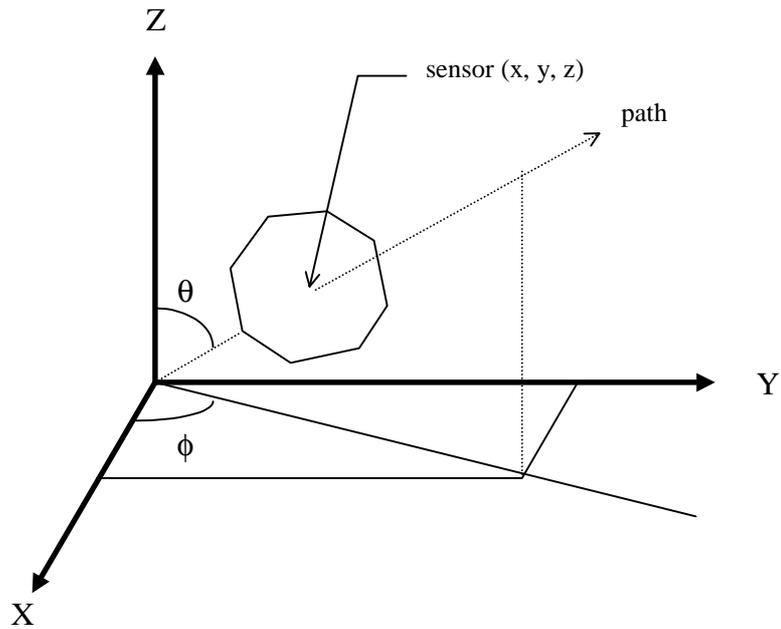
The body of an endoscope is cylindrical. To render a realistic 3D image, this cylindrical shape is approximated as a ‘stack’ of octagonal prisms following the curve of the endoscope. Surface normals are calculated for each of the prism vertices such that the octagonal edges of the image appear smooth when viewed obliquely. This gives a satisfactory compromise between image quality and response time, the octagonal nature of the image only being apparent when viewing the endoscope image end-on.

The steps involved in calculating all image vertices are as follows :

- Build a representation of the path of the endoscope as a set of  $(x, y, z)$  co-ordinates approximating a 3D curve.
- Generate octagonal points around each of the above co-ordinates and use them to define a set of polygons making up the endoscope surface.
- Calculate surface normals for each image vertex to ‘smooth’ the final rendered image.

#### **Building a 3D curve**

Assuming there are  $s$  sensors providing data from the Bladen imager, the input to this process is a set of  $s$  data records  $r_1, r_2, \dots, r_s$  each providing  $(x, y, z)$  and  $(\theta, \phi)$  for the corresponding sensor and a set of  $s$  length measurements  $l_1, l_2, \dots, l_s$  giving,



**Figure 3-D – Sensor position and orientation (axes rotated for clarity)**

for each sensor, the distance from the previous sensor (or from the instrument tip for the first sensor). Figure 3-D shows an example of a sensor in a particular position and orientation indicating the path of the curve at that point.

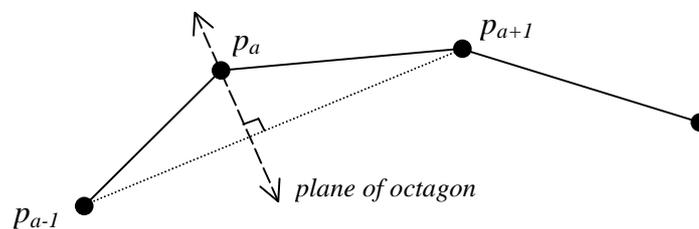
For each pair of adjacent sensors  $a$  and  $a+1$ , a curve-fitting algorithm is used to provide a set of  $(x, y, z)$  co-ordinates which closely approximate the path of the complete endoscope. This algorithm is implemented in the manner described by Bladen (1995), details of which are reproduced in Appendix A. The algorithm uses length value  $l_{a+1}$  with data records  $r_a$  and  $r_{a+1}$  to interpolate a given number  $m$  of  $(x, y, z)$  points on the curve joining the two sensors. The process is repeated for all sensor pairs. The output from this stage of calculation is set of  $n = m(s-1)$  co-ordinates approximating the 3D curve describing the path of the endoscope, where  $s$  is the number of sensors.

## Generating octagonal prisms

The input to this stage is the set of  $n$   $(x, y, z)$  points describing the 3D curve generated above. Call these points  $p_1 .. p_n$ . Given that each of these points is to be taken as the centre of an octagon of diameter  $d$  (configurable from the application), it is a straightforward matter to generate the eight vertices of each octagon once the plane in which it should lay is determined. Figure 3-D shows one such octagon constructed around one of the sensor points. Each of the eight vertices of neighbouring octagons may be joined to form a series of connected octagonal prisms along the path of the endoscope.

To create a smooth image from the collection of octagonal prisms so formed, each one must be ‘mitred’ to its neighbour to avoid unsightly kinks in the rendered image. This is achieved as follows.

Taking three adjacent points on the 3D curve, say  $p_{a-1}$ ,  $p_a$  and  $p_{a+1}$ , the required plane for the octagon surrounding  $p_a$  is that which is normal to the vector joining  $p_{a-1}$  and  $p_{a+1}$  (see Figure 3-E).



**Figure 3-E – Creating mitred joints**

The points  $p_1$  and  $p_n$  are obviously special cases as no preceding/succeeding vertex is available to determine the plane in the manner described above. To simplify the coding of the algorithm, the vertices of the octagon around  $p_1$  are calculated by applying the  $\theta$  and  $\phi$  rotations for the first sensor to a ‘default’ octagon defined in the plane  $z=0$ , and an additional point is programmatically generated at the other end of the curve simply by continuing the existing line segment in the same direction. This effectively gives a range of points  $p_1 \dots p_{n+1}$  which are used to generate a set of  $n$  octagons centred on points  $p_1 \dots p_n$ . In practice, each octagon other than the first is ‘projected’ from the vertices of the previous octagon in the following manner :

Referring to Figure 3-E above, assume we have defined the eight vertices  $q_1 \dots q_8$  of the octagon around  $p_{a-1}$  and we wish to find the corresponding vertices  $r_1 \dots r_8$  of the octagon around  $p_a$ . We first find the normal  $n$  to the required plane for the octagon as :

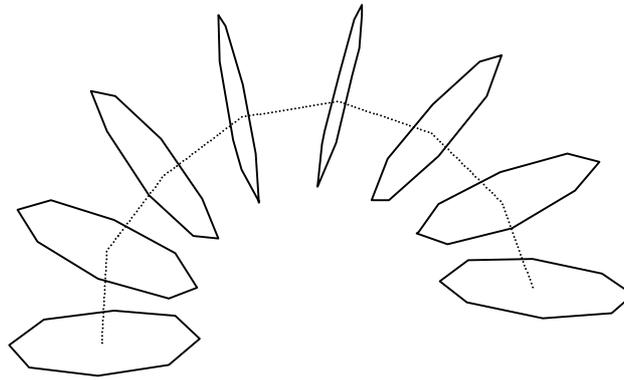
$$n = v1 + v2$$

where  $v1$  is a unit vector in the direction of  $p_{a-1}$  and  $v2$  is a unit vector in the direction of  $p_{a+1}$ .

Each point  $q_1 \dots q_8$  is then projected to the corresponding  $r_1 \dots r_8$  by :

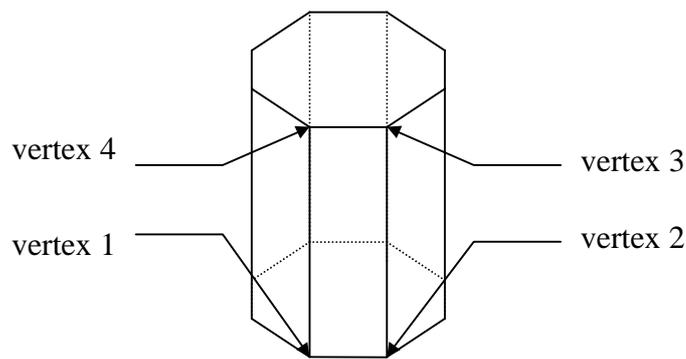
$$r_x = q_x - (n \bullet q_x) / (n \bullet v1) v1$$

This process is repeated to define a complete set of smoothly mitred octagons around each point  $p_1$  to  $p_n$  (see Figure 3-F).



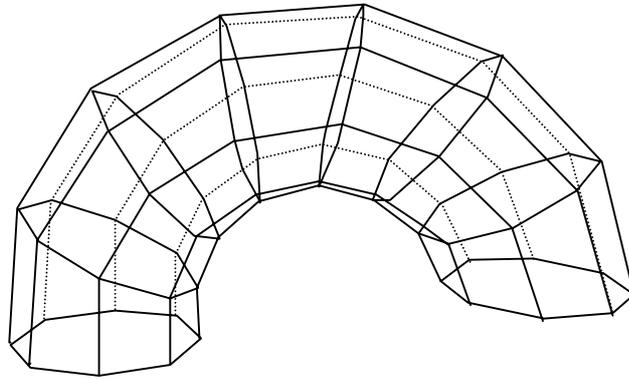
**Figure 3-F – Octagonal segmentation**

Having created all the vertices required for the endoscope image, these are used to describe a set of four-sided polygons defining the endoscope surface. Each pair of adjacent vertices in an octagon is coupled with the corresponding pair of vertices in the neighbouring octagon (see Figure 3-G). The set of four vertices is defined to OpenGL in counterclockwise order to allow the graphics system to differentiate between front-facing and back-facing polygons during rendering.



**Figure 3-G – Defining polygonal surfaces**

Once this process has been repeated for all octagons surrounding all points on the curve, the result is a collection of four-sided polygons describing a wire-frame representation of the endoscope (see Figure 3-H).



**Figure 3-H – The wireframe model**

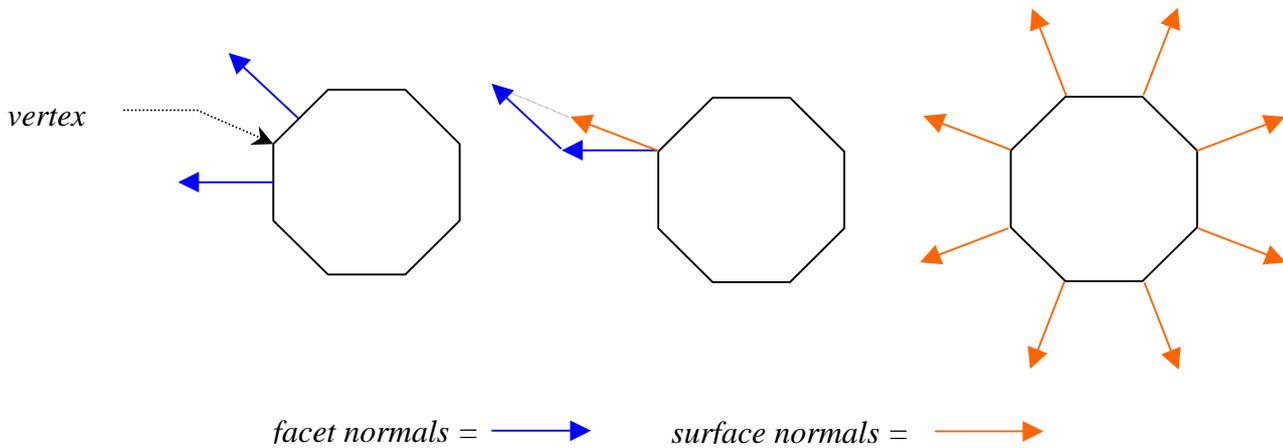
The next stage is to generate surface normals for each of the model vertices to allow the image to be rendered as a realistic solid object.

### **Calculating surface normals**

Surface normals are supplied to the rendering system in the form of vectors associated with each vertex of the model. The surface normal indicates the curvature of the surface at a particular point and is used in lighting / shading calculations to generate the appearance of a smooth surface.

Each vertex in the model is shared by up to four polygons, each of which may have different orientations. The surface normal at such a vertex is formed by first calculating normals to each of the four polygons (facets) in which the vertex

participates. These ‘facet normals’ are then added and re-normalised to provide the overall surface normal at the vertex concerned.



**Figure 3-I – Generating surface normals**

This process creates surface normals which imply a smooth curve both around the octagonal body and along the curved path of the model. Figure 3-I shows in plan view how surface normals are constructed to imply a cylindrical surface from an octagonal approximation. The figure on the left shows two facet normals calculated for the facets adjacent to the marked vertex. The centre figure shows the surface normal resulting from the addition and re-normalisation of the facet normals. The figure on the far right shows surface normals calculated similarly for all vertices. Given this configuration, the OpenGL system will calculate surface characteristics for all points between the vertices, giving an overall circular (cylindrical) appearance when viewed from the side.

### 3.2.3 *Rendering and display*

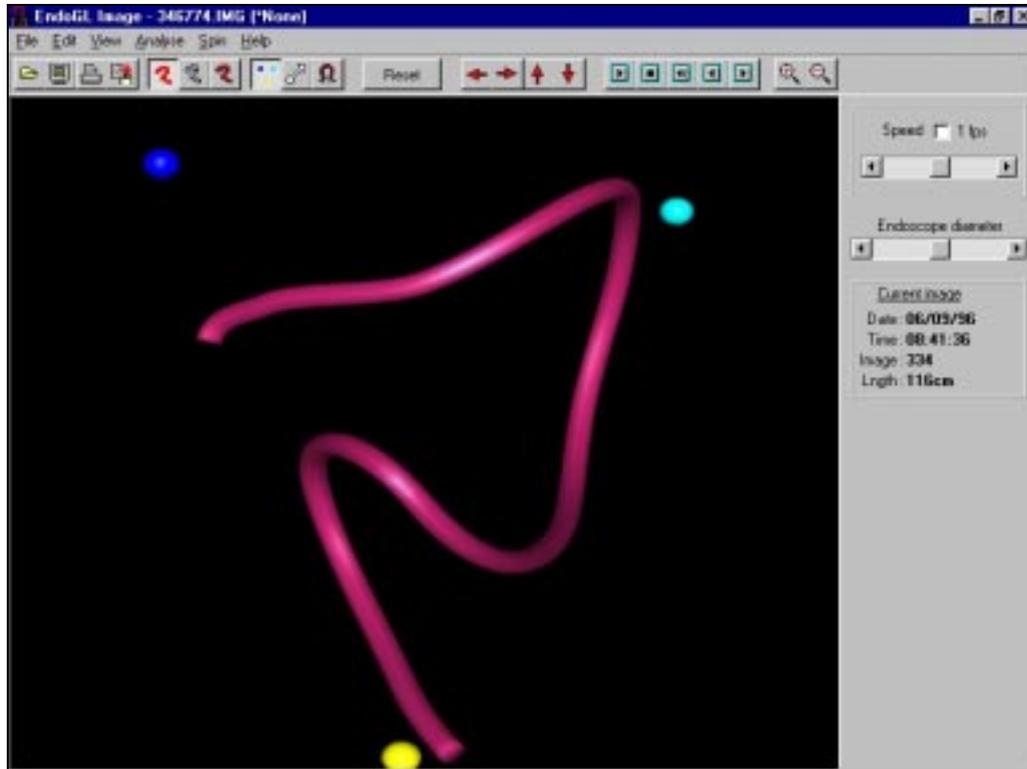
As a result of the previous calculations, the following data is available which relates to the position and configuration of the colonoscope at a fixed point in time.

- An array of vertices defining the shape of the colonoscope
- A collection of polygons defining the outer surface of the shape
- An array of surface normals to each vertex

At this point, an additional array is defined giving colour information for each vertex. This would not strictly be necessary if the colonoscope were all one colour (it usually is), but is used to allow certain features – for example, the positions of the sensors – to be made distinguishable in the final rendered image. This information is supplied to the OpenGL API along with lighting parameters and any transformations affecting the ‘camera’ or model – these are driven by the user interface to the system where controls allow zoom, pan, and spin functions. In addition, a facility to allow the user to rotate the image using a click=drag action with the computer mouse has been implemented. This is known as a ‘virtual trackball’ and provides an effective and intuitive method of viewing the colonoscope image from various angles.

The three marker positions set before the colonoscopy commences are defined to OpenGL as different coloured spheres. These represent the rough positions of the hepatic flexure, splenic flexure and anus.

Figure 3-J shows a sample screen shot from the RMR system which shows a colonoscope almost fully intubated.



**Figure 3-J – Fully rendered colonoscope image**

This screen shot demonstrates how effective the use of surface normals is in producing a convincing cylindrical image from an octagonal approximation.

## Chapter 4 Making measurements using the model

### 4.1 Requirements definition

#### *4.1.1 Functional requirements*

The functional requirements for this enhancement to the system were relatively simple to establish. Consultation with one of the principle users of the Bladen imaging system revealed the following needs.

- The measurement function should be fully integrated into the existing software. The length of colonoscope inside the patient should always be shown.
- The measurement process should be interactive with instant feedback of results. Existing zoom, pan and rotation controls should be unaffected.
- It should be possible to select any two points along the colonoscope image and to show not only the straight-line distance between these points but also the length of the section of colonoscope so defined.
- For each of the two points selected, the offset distance of each from the colonoscope tip should be reported.
- It should be possible to change either one of the selected points without the need to redefine the other.

## **4.2 Outline design**

### *4.2.1 Length measurement strategy*

Measurement of distances between two points on the colonoscope image is one major part of the requirement. *Selecting* those points is another major part. The latter is covered in the next section.

When attempting to make measurements between two points on the image, the first problem is how to define such a ‘point’. As detailed in the previous chapter, the 3D model of the colonoscope consists of a set of related vertices. Initially, the ‘obvious’ technique was to use these vertices as ‘available’ measurement points and allow measurement between any two of them. On further analysis, however, this method presented some problems.

- The accuracy of measurements based on the co-ordinates of vertices chosen on ‘opposite’ sides of the colonoscope image would be compromised by a factor related to the displayed diameter of the image. This complication would be compounded by the fact that the displayed diameter is variable and may be adjusted during measurement.
- It would be theoretically possible to pick vertices from the same ‘octagon’, in effect measuring ‘around’ the shaft of the colonoscope image. This would need to be detected and dealt with in some suitable fashion.
- The resolution of the measured length would be dependent on the relative spacing between octagonal segments. This would vary depending on whether the segments concerned lay on the outside or inside of a bend.

Considering the above it became clear that the most accurate measurements should be - a) made as though through the centre of the colonoscope image, and b) unaffected by any approximations used to optimise display speed. The following design was thus established.

- Measurement points must relate to points directly on the 3D curve describing the colonoscope path, not those on the subsequently ‘inflated’ octagonal image.
- For measurement purposes only, the number of points interpolated between sensor positions should be temporarily increased to give acceptable precision.
- For measurement of colonoscope section length, the inter-sensor spacing contained in the data files should be used.

The last item above is to avoid any cumulative errors arising when adding many small vector magnitudes of the line segments approximating the curved path. As we know the distance between adjacent sensors and we control the number of line segments fitting the curve between them, it is a simple matter to derive an ‘offset’ length for each line segment from the instrument tip. The method is as follows.

For each sensor, the image data includes the relative spacing (in centimetres) from the previous sensor (or from the instrument tip for the distal sensor). From this data, the absolute distance of each sensor from the instrument tip may be determined and stored in an array. Suppose, for sensors  $1..n$  that these absolute distances are  $d_1 .. d_n$ . The curve fitting algorithm described in section 3.2.2 produces an arbitrary number (say  $m$ ) of straight-line segments approximating the

curve following the instrument path between two adjacent sensors, say  $a$  and  $b$ . The approximate length,  $l$ , in centimetres of each straight line segment between sensors  $a$  and  $b$  is then given by

$$l = (d_b - d_a) / m$$

This information is used to produce a refined distance array  $rd_1 \dots rd_m$  such that an absolute distance is held for one vertex of each intermediate straight-line segment rather than just for each sensor.

Once the two points to be measured are identified, it is possible to retrieve the distance of each point from the instrument tip using the refined distance array and to calculate the length of the section of colonoscope between these points by subtraction.

Finding the distance ‘as the crow flies’ between the same two points is more straightforward. The Bladen imager reports  $(x, y, z)$  co-ordinates in terms of centimetres, so the straight-line distance between two points is simply the magnitude of the vector joining them. For points  $p1$  and  $p2$ , the distance  $d$  would be given by

$$d = \sqrt{[(x_{p2} - x_{p1})^2 + (y_{p2} - y_{p1})^2 + (z_{p2} - z_{p1})^2]}$$

Having developed a method for implementing the distance calculation functions given the two points concerned, a method is required to identify and select those points.

#### *4.2.2 User interface design*

The requirements specified that the selection of measurement points should be interactive and should not interfere with the existing image manipulation functions of the application. This implied that there also needed to be some mechanism which made it clear to the user which measurement points he/she had selected. For the design of the user interface, two options were considered.

##### **Design one**

The first option considered was to annotate the displayed image by superimposing numbered labels along its length and prompting the user to key in the numbers corresponding to those points required. As it was evident that this could lead to a very cluttered display, the idea was refined to provide a ‘drill-down’ technique whereby the broad area was first selected and a magnified section of the image then redisplayed containing further labelled sub-divisions. Whilst early tests proved the technique to be technically feasible, a more detailed design of the user interface appeared unavoidably complicated and uncomfortable. A very early prototype revealed further shortcomings with this technique –

- When ‘drilling-down’ to a magnified image, it was more difficult to select measurements points with the magnified section displayed out of context.

- The need to manipulate both mouse and keyboard to make selections appeared cumbersome and it would be difficult to subsequently change one or other measurement point without repeating the ‘drill-down’ process.

As the ‘look and feel’ of this type of interface was unsatisfactory almost from the outset and further study served only to reveal new complications, this design was abandoned at an early stage.

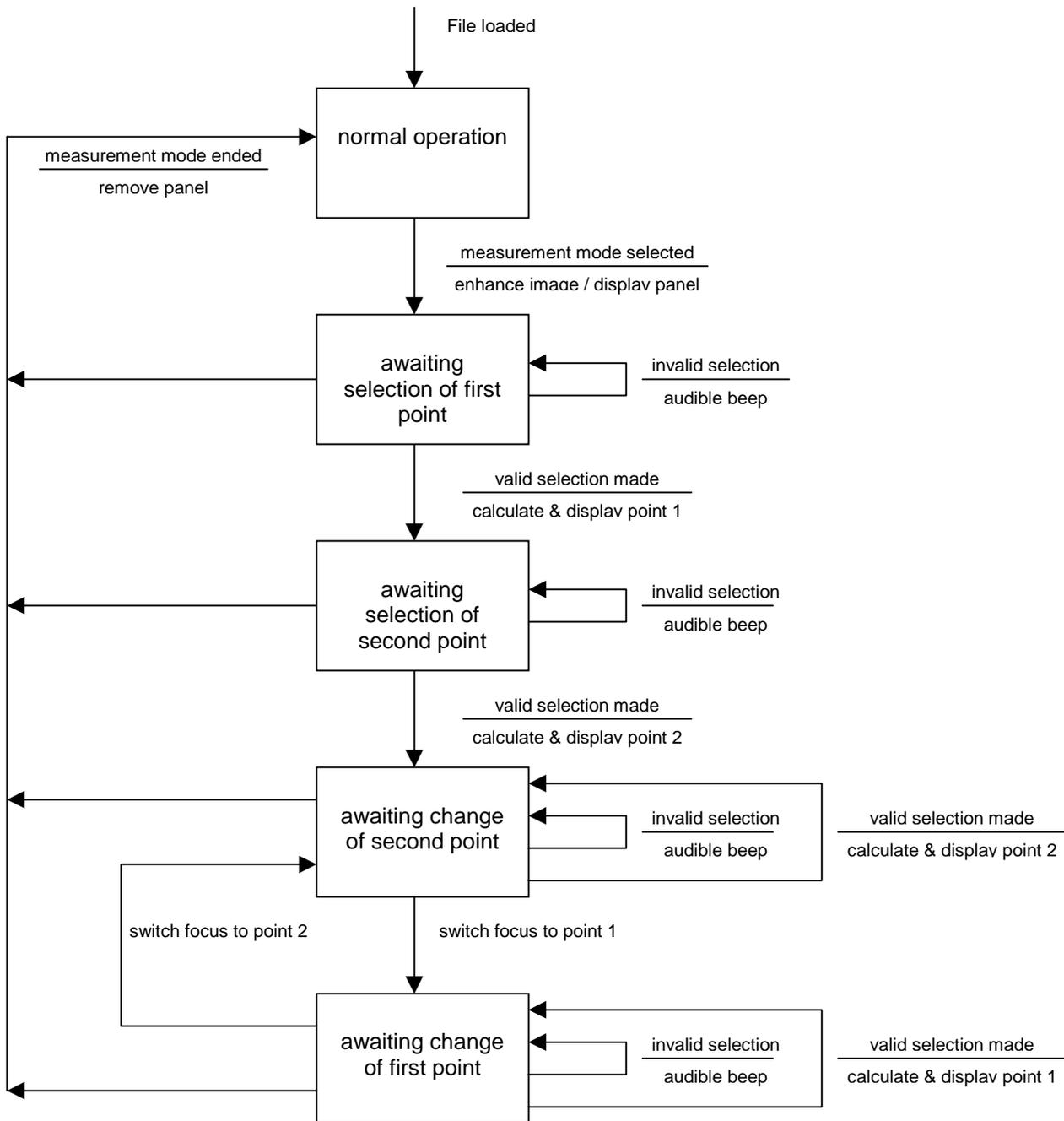
## **Design two**

The second design was based primarily on the use of the computer mouse for selection of measurement points. As the user was already accustomed to manipulating the colonoscope image using the left mouse button to operate the virtual trackball, the ‘point and click’ type of interaction could logically be provided with either a double-click or by use of the right mouse button. This would allow the user to select the required view and to identify a measurement point almost in one motion. Based on this idea, the following sequence of events was envisaged –

- After using the normal features of the system to navigate to the required image, the user would select an option from the menu to place the system into a ‘measurement’ mode.
- At this point, the display would be redrawn in higher resolution and a display panel presented for feedback of measurements.

- After using the pan, zoom or virtual trackball controls to position the image, the user would click with the right mouse button over a point on the colonoscope shaft.
- The position of the click would be established by the system and audible feedback given if this did not coincide with a point on the image. Immediate visual feedback of a successful ‘hit’ would be given by displaying a coloured highlight at that point on the image.
- The position of this first point, in relation to the colonoscope tip would be calculated and shown in the measurement display panel. The system would now ‘assume’ that a second click represented the second point.
- After capturing and highlighting two measurement points in this manner, all remaining distances specified in the requirements would be calculated and displayed.
- The system would allow selection of one or other of the points by clicking an option button. The associated point could then be ‘overwritten’ by further mouse clicks.
- The ‘measurement’ mode would be terminated and the system returned to conventional display on selection of a menu option or command button.

This part of the design could be represented by the following state transition diagram.

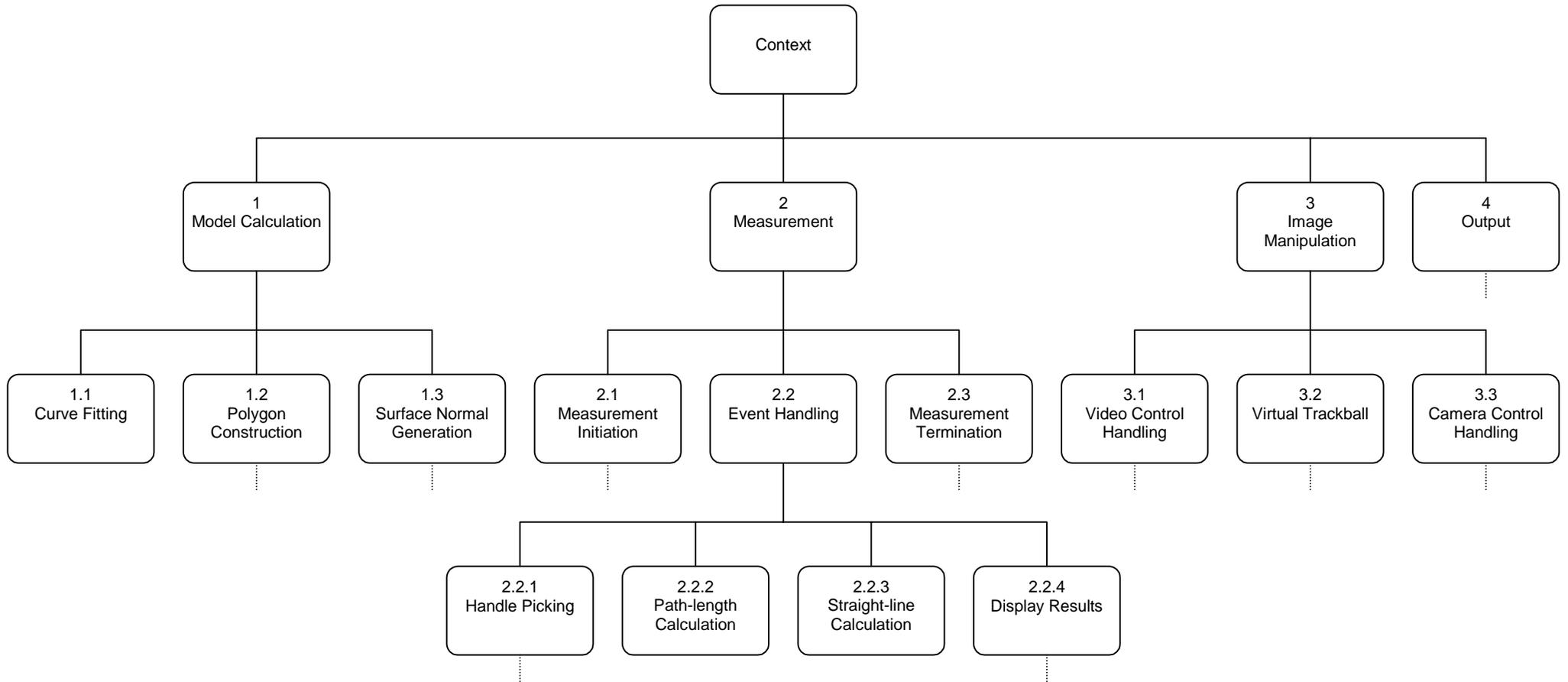


**Figure 4-A – State transition diagram**

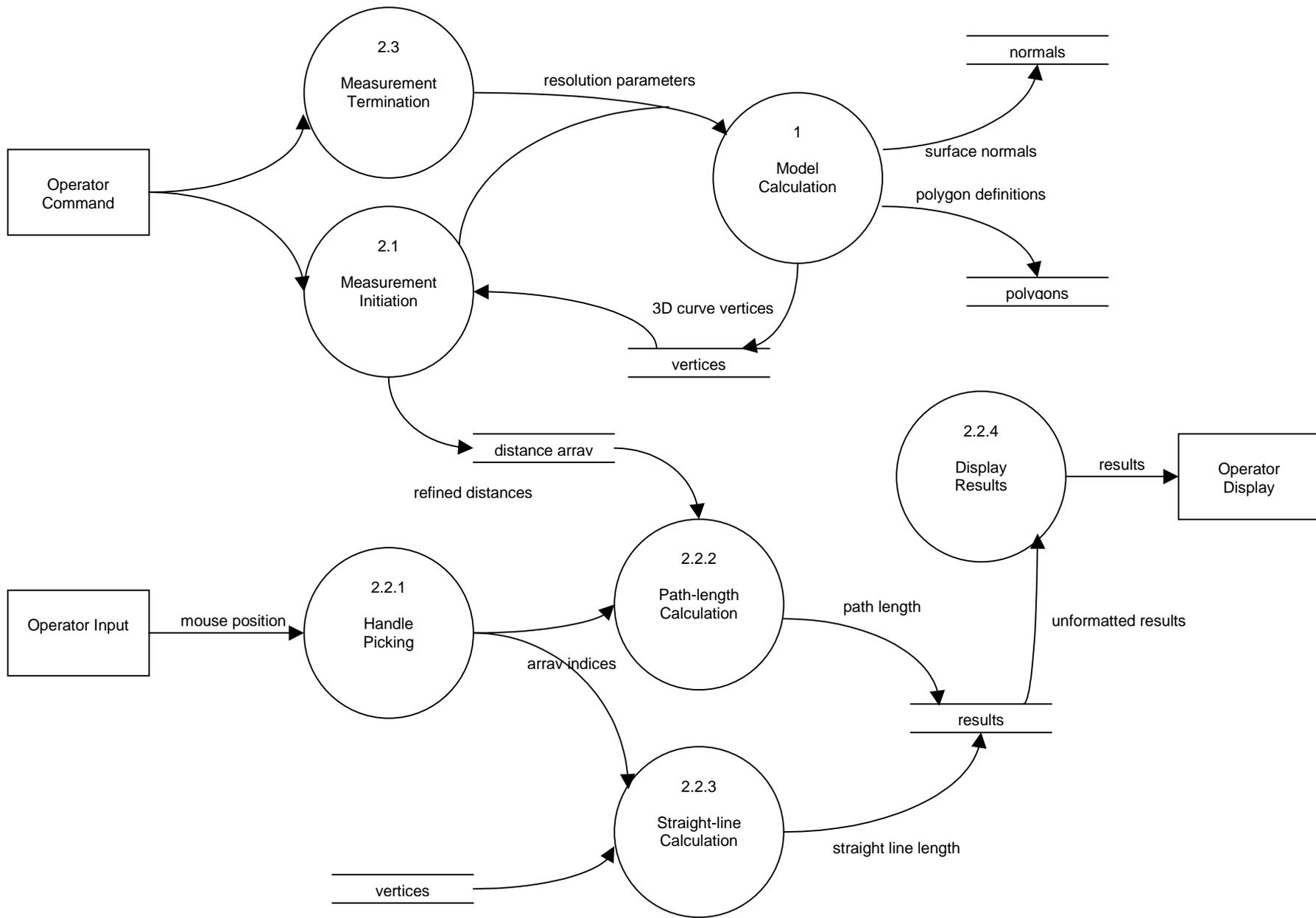
This design has the advantage that all measurements are made using the mouse. The user would be able to zoom, pan and rotate the image, select measurement

points and receive instant feedback of the selected points and calculated distances without needing to look away from the screen. As the intention is to highlight the selection on the image in a contrasting colour, further manipulation of the image would retain this visibility and allow modification of one or both points when viewed in a different context. This design was chosen as the most suitable to allow effective measurements to be made in an intuitive manner.

The measurement module needs to be integrated into the system to take advantage of existing data structures and functions. Figure 4-B overleaf shows a partial functional decomposition diagram illustrating the broad structure of the measurement function along with other key functions in the existing system. The data flow diagram in Figure 4-C shows the interaction between these additional functions and key functions in the existing system.



**Figure 4-B – Functional decomposition diagram**



**Figure 4-C – Data flow diagram**

### **4.3 Detailed design**

**The design language for this section follows the guidelines of Sargent (1991).**

#### *4.3.1 Data requirements*

In order to implement the chosen design, the following *additional* data items are required.

- An array of values for each point on the 3D curve holding the distance of each point from the instrument tip. (*distanceArray*)
- A structure to store the details of each of the two measurement points. This would include indices into the above array and measurement results. (*measureData*)
- A structure to record the current state during measurement mode. (*currentState*)

Note also that the existing sensor spacings from the \*.IMG file and the array of 3D points defining the colonoscope path will be needed.

Table 4-A shows the data table for this design. Two constant definitions are included; one to define the maximum possible number of sensors that could be used, and the other to define the maximum number of line segments *between* sensors used to approximate the 3D curve. Together, these constants define the maximum allowable size of the refined distance array.

Identifier	Description	Type
<i>maxSensors</i>	Constant definition	Integer constant
<i>maxSegments</i>	Constant definition	Integer constant
<i>stateType</i>	Enumerated type definition	( <i>inactive, wait1, wait2, change1, change2</i> )
<i>eventType</i>	Enumerated type definition	( <i>initiate, terminate, switch1, switch2, mouseclick</i> )
<i>measureType</i>	Record type definition	
	<i>mIndex1</i> – Index to point 1	Integer
	<i>mIndex2</i> – Index to point 2	Integer
	<i>mPathLen</i> – Path length	Real
	<i>mShortLen</i> – Shortest length	Real
	recordend	
<i>vertexArray</i>	An array of (x,y,z) points	array [0.. <i>maxSegments</i> * ( <i>maxSensors</i> -1)]
<i>distanceArray</i>	An array of reals	array [0.. <i>maxSegments</i> * ( <i>maxSensors</i> -1)]
<i>measureData</i>	Measurement data record	<i>measureType</i> variable
<i>currentState</i>	Current measurement state	<i>stateType</i> variable
<i>SensorSpacing</i>	An array of reals	array [0..17]
<i>EventHandler</i>	( <i>event, mouseX, mouseY</i> )	procedure ( <i>eventType</i> , integer, integer)
<i>BeginMeasurement</i>	( <i>currentState</i> )	procedure ( <i>stateType</i> )
<i>EndMeasurement</i>	( <i>currentState</i> )	procedure ( <i>stateType</i> )
<i>PopulateDistances</i>	( <i>SensorSpacing, distanceArray</i> )	procedure (array, array)
<i>ProcessClick</i>	( <i>mouseX, mouseY</i> )	Integer function (integer, integer)
<i>GetPathLength</i>	( <i>measureData, distanceArray</i> )	procedure ( <i>measureType</i> , array)
<i>GetShortestLength</i>	( <i>measureData, vertexArray</i> )	procedure ( <i>measureType</i> , array)

**Table 4-A – Data table for the measurement module**

#### 4.3.2 Function requirements

The following procedures and functions will be required:

- A controlling procedure. (*EventHandler*)
- A procedure to initiate measurement mode. (*BeginMeasurement*)
- A procedure to terminate measurement mode. (*EndMeasurement*)
- A procedure to populate the distance array. (*PopulateDistances*)
- A function to capture and evaluate a mouse-click over the image area.  
(*ProcessClick*)
- A procedure to update the path length between two points. (*GetPathLength*)

- A procedure to update the shortest distance between two points.

(*GetShortestLength*)

Table 4-B shows the specifications for the above routines.

***EventHandler***

<i>event</i>	Event to process	<i>eventType</i> parameter
<i>mouseX</i>	Mouse X co-ordinate	Integer parameter
<i>mouseY</i>	Mouse Y co-ordinate	Integer parameter
{ Control the measurement process, using helper functions where necessary }		

***BeginMeasurement***

<i>state</i>	Current measurement state	<i>stateType</i> parameter
{ Recalculate and display image in high resolution, initialise measurement display panel, return <i>wait1</i> measurement state }		
<i>state</i>	Updated state	<i>stateType</i> parameter

***EndMeasurement***

<i>state</i>	Current measurement state	<i>stateType</i> parameter
{ Recalculate and display image in normal resolution, remove measurement display panel, return <i>inactive</i> measurement state }		
<i>state</i>	Updated state	<i>stateType</i> parameter

***PopulateDistances***

<i>SensorSpacing</i>	Sensor spacing values	array[0..17] parameter
{ Calculate refined distances using sensor spacings and populate <i>distanceArray</i> }		
<i>distanceArray</i>	Returned distances	array parameter

***ProcessClick***

<i>mouseX</i>	Mouse X co-ordinate	Integer parameter
<i>mouseY</i>	Mouse Y co-ordinate	Integer parameter
{ Determine and return index to <i>distanceArray</i> for the vertex under the mouse position, return -1 if no vertex found under the mouse }		
<i>ProcessClick</i>	Returned index or -1	Integer parameter

***GetPathLength***

<i>measureData</i>	Measurement data record	<i>measureType</i> parameter
<i>distanceArray</i>	Array of distances	array parameter
{ Calculate the path length and update the measurement data record }		
<i>measureData</i>	Updated record	<i>measureType</i> parameter

***GetShortestLength***

<i>measureData</i>	Measurement data record	<i>measureType</i> parameter
<i>vertexArray</i>	Array of (x,y,z) points	array parameter
{ Calculate the shortest length and update the measurement data record }		
<i>measureData</i>	Updated record	<i>measureType</i> parameter

**Table 4-B – Procedure and function specifications**

Note that the *EventHandler* procedure is included above as a separate logical entity although, in practice, part of the design for this procedure will be implemented within existing event handling routines for the system.

#### 4.3.3 Measurement module design

The detailed designs for the measurement control module and associated procedures and functions are shown in the following figures.

##### ***BeginMeasurement***

---

<i>state</i>	
1.1	Recalculate high resolution model vertices
1.2	Display model
1.3	Initialise measurement panel
1.4	Display measurement panel
1.5	<i>state := wait1</i>

---

<i>state</i>	
--------------	--

---

##### ***EndMeasurement***

---

<i>state</i>	
1.1	Recalculate normal resolution model vertices
1.2	Display model
1.3	Hide measurement panel
1.4	<i>state := inactive</i>

---

<i>state</i>	
--------------	--

---

##### ***EventHandler***

---

<i>event, mouseX, mouseY</i>	
1.1	select case depending on <i>event</i>
2.1	<i>initiate</i> : <i>BeginMeasurement(currentState)</i>
2.2	<i>PopulateDistances(SensorSpacing, distanceArray)</i>
3.1	<i>terminate</i> : <i>EndMeasurement(currentState)</i>
4.1	<i>switch1</i> : <i>currentState := change1</i>
5.1	<i>switch2</i> : <i>currentState := change2</i>
6.1	<i>mouseclick: tmp := ProcessClick(mouseX, mouseY)</i>
6.2	if <i>tmp &lt; 0</i> then
6.3	audible beep
6.4	else
6.5	select case depending on <i>currentState</i>

```

6.6      wait1 : measureData.mIndex1 := tmp
6.7      display distanceArray[tmp]
6.8      currentState := wait2
6.9      wait2 : measureData.mIndex2 := tmp
6.10     display distanceArray[tmp]
6.11     GetPathLength(measureData)
6.12     GetShortestLength(measureData)
6.13     display measurements
6.14     currentState := change2
6.15     change1: measureData.mIndex1 := tmp
6.16     display distanceArray[tmp]
6.17     GetPathLength(measureData)
6.18     GetShortestLength(measureData)
6.19     display measurements
6.20     change2: measureData.mIndex2 := tmp
6.21     display distanceArray[tmp]
6.22     GetPathLength(measureData)
6.23     GetShortestLength(measureData)
6.24     display measurements
6.25     selectend
6.26     ifend
6.27     selectend

```

---



---

### ***PopulateDistances***

#### ***SensorSpacing***

```

1.1     distanceArray[0] := SensorSpacing[0]
1.2     loop for s := 0 to maxSensors - 2
1.3       dx := SensorSpacing[s + 1] / maxSegments
1.4       loop for d := 1 to maxSegments
1.5         x := (maxSegments * s) + d
1.6         distanceArray[x] := distanceArray[x - 1] + dx
1.7       loopend
1.8     loopend

```

---

#### ***distanceArray***

---

### ***GetPathLength***

#### ***measureData, distanceArray***

```

1.1     with measureData do
1.2       wrk := distanceArray[mIndex1] - distanceArray[mIndex2]
1.3       mPathLen := abs(wrk)
1.4     withend

```

---

#### ***measureData***

---

The *GetPathLength* function is not only called by the measurement module whilst in measurement mode but, as noted in the requirements definition, is also called from the main application to determine and display the length of colonoscope inside the patient. When called in this way, the *distanceArray* is less precise due to the lower resolution image but is deemed sufficiently accurate given the inherent difficulty in identifying the exact position of the anus from within the software

---

### ***GetShortestLength***

---

*measureData, vertexArray*

---

- 1.1 with *measureData* do
  - 1.2      $vx := vertexArray[mIndex1].x - vertexArray[mIndex2].x$
  - 1.3      $vy := vertexArray[mIndex1].y - vertexArray[mIndex2].y$
  - 1.4      $vz := vertexArray[mIndex1].z - vertexArray[mIndex2].z$
  - 1.5      $mShortLen := \sqrt{vx^2 + vy^2 + vz^2}$
  - 1.6 withend
- 

*measureData*

---



---

### ***ProcessClick***

---

*mouseX, mouseY*

---

- 1 if mouse position corresponds to a model vertex
  - 2     return index to associated point in vertex and distance arrays
  - 3 else
  - 4     return -1
  - 5 ifend
- 

*ProcessClick*

---

The outline design for the above function shows this to be one of the key components in the measurement module. The method used to evaluate the position of a mouse-click in relation to the 3D model displayed at the time relies heavily on functionality provided by the OpenGL API. Rather than attempting to decompose the above design further, section 4.3.4 describes the implementation of this function in detail.

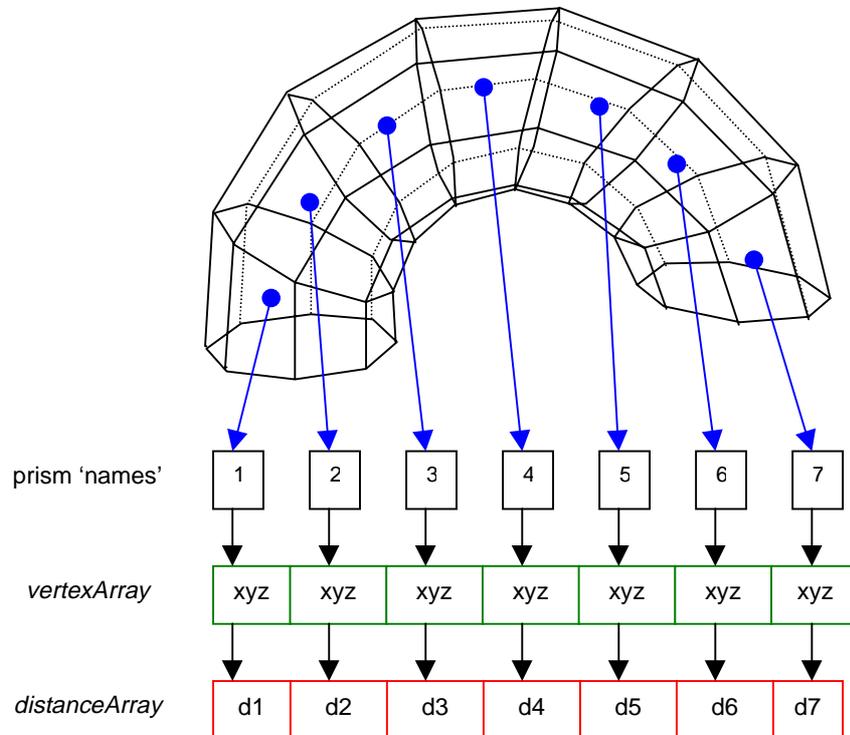
#### 4.3.4 *Implementing the picking function*

OpenGL provides a powerful feature called *selection*. This process involves defining a small area of the computer screen (in window co-ordinates) and then obtaining from the API a list of the 3D objects in the model that would intersect with this area when rendered with the current viewing parameters. The act of selecting a specific object displayed on the screen in this way is called *picking*.

Whilst it is technically possible to implement picking for each individual graphic primitive (point, line or polygon) defined in a 3D model, it is more usual to group primitives together to allow selection of larger objects. For the model of the colonoscope described in section 3.2.2, each octagonal prism is defined as a separate object and associated with the point on the 3D curve around which one end of the prism was built. Selection of any point on the surface of the colonoscope image will then allow this point to be identified.

When using the OpenGL selection functions, each graphic primitive is allocated a 'name' at the time it is defined to the API. It is these 'names' that are returned by the selection function to indicate the 3D objects intersecting the reduced viewing volume. In practice, the 'names' are unsigned 32-bit integers and would normally be referenced in the application code via constant definitions. In this case, however, it makes sense to use the index into the array of 3D points defining the colonoscope curve to 'name' each prism. As each of the points on this curve will also have an associated value in the *distanceArray*, any point on the colonoscope image surface can then be directly related to both an  $(x, y, z)$  co-ordinate and an

offset length from the instrument tip – both of which are then used in the measurement functions (see figure 4-D).



**Figure 4-D – Mapping prism selections to length offsets and vertices**

As noted in the outline design of the picking function (*ProcessClick*), the selection procedure is initiated following a click with the right mouse button at some place on the 2D display of the colonoscope image. The *ProcessClick* function is supplied with the *x* and *y* window co-ordinates of the mouse pointer as parameter values. The next major steps in the function are as follows :

- A buffer to capture selection 'hits' is defined using the API function *glSelectBuffer*.

- OpenGL is placed into selection mode using the API function *glRenderMode(GL\_SELECT)*.
- The API function *gluPickMatrix* is used to define a reduced viewing volume with a two-pixel radius about the given mouse coordinates.
- The colonoscope image is ‘redrawn’ as a series of quadstrips each defining an octagonal prism and named with the appropriate index value. No rendering to the display device occurs while in selection mode.
- OpenGL is placed back into render mode using the API function *glRenderMode(GL\_RENDER)*. This collects any hit records into the selection buffer.
- If no hits were recorded then the function returns -1 otherwise it returns the ‘name’ of the first ‘hit’ record in the selection buffer.

The selection buffer may contain a number of ‘hit’ records if more than one graphic object appears under the mouse pointer. This may occur when the colonoscope image is looped and the selection is made at a point where one part of the image crosses another. In such a situation, the ‘hit’ records in the selection buffer are arranged in order of their  $z$  (depth) co-ordinates. The policy of returning the first record in the buffer ensures that the graphic object closest to the camera is always selected.

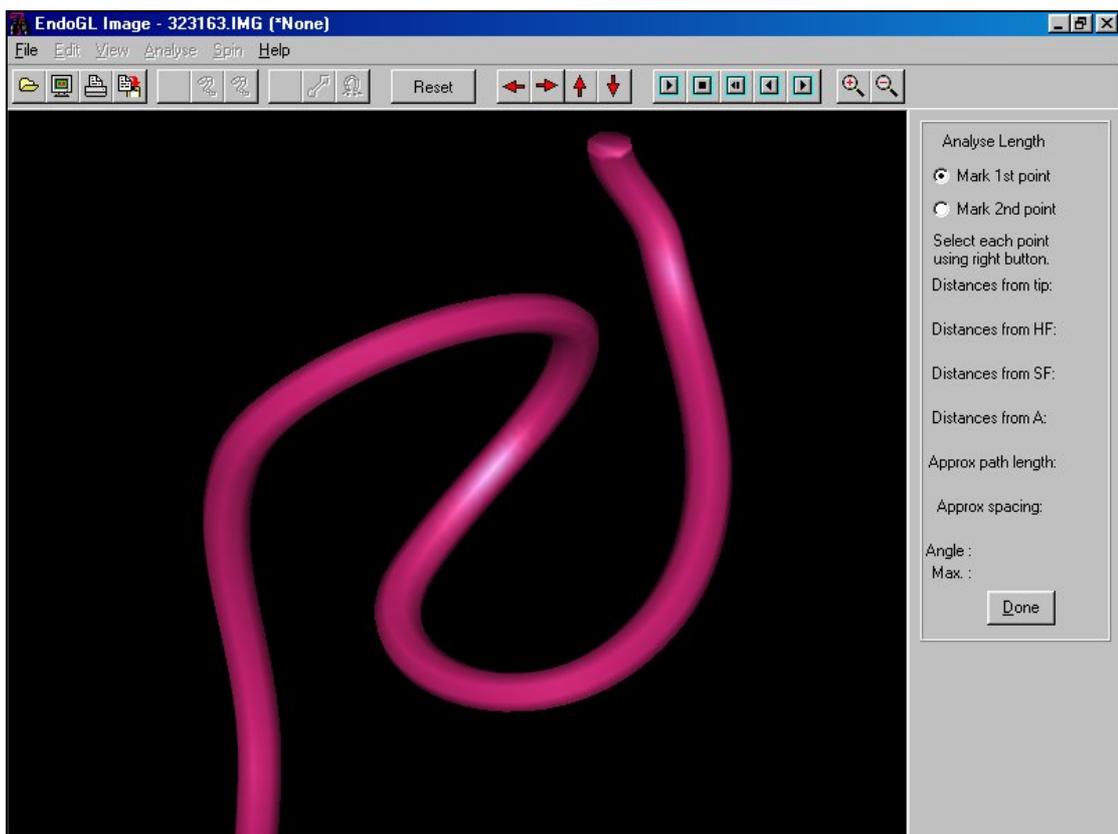
## Chapter 5 Evaluation of measurement accuracy

### 5.1 The system in use

The following figures show the steps involved in making measurements using the enhanced system.

#### 5.1.1 Initiating measurement mode

Figure 5-A shows an example of the application display immediately after measurement mode has been selected from the menu (option *Analyse / Lengths*).



**Figure 5-A – Initiating measurement mode**

The measurement panel has been initialised and displayed on the right of the screen and irrelevant menu options and toolbar buttons have been disabled. The option buttons in the measurement panel used to switch focus between the two possible measurement points are set to point one and all output data has been cleared.

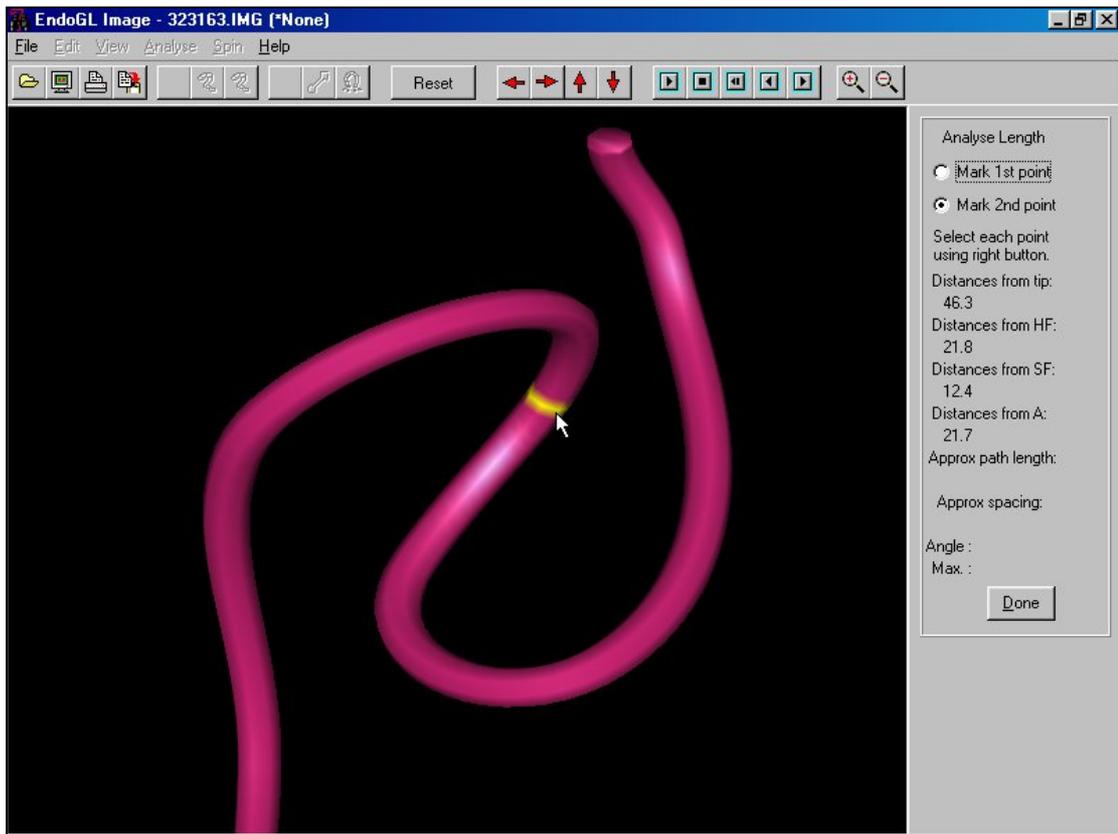
In addition to the distance of each point from the instrument tip, the path length between the points and the straight-line distance, this display also reports the straight-line distance of each point from the three markers. Marker 1 indicates the hepatic flexure (HF), marker 2 the splenic flexure (SF) and marker 3 the anus (A). These measurements are intended for later enhancements.

Note that the system will still allow the displayed image to be rotated using the virtual trackball. It is also possible to pan / zoom using the toolbar buttons or to navigate to an alternative image using the video controls.

### *5.1.2 Selecting the first point*

Figure 5-B shows the display immediately after clicking with the right mouse button on the colonoscopy image at the point shown.

The length of colonoscopy from the tip to the indicated point has been calculated as have the straight-line distances to all three markers. The results are displayed in the measurement panel and the system now automatically selects the option button to allow the second measurement point to be chosen.



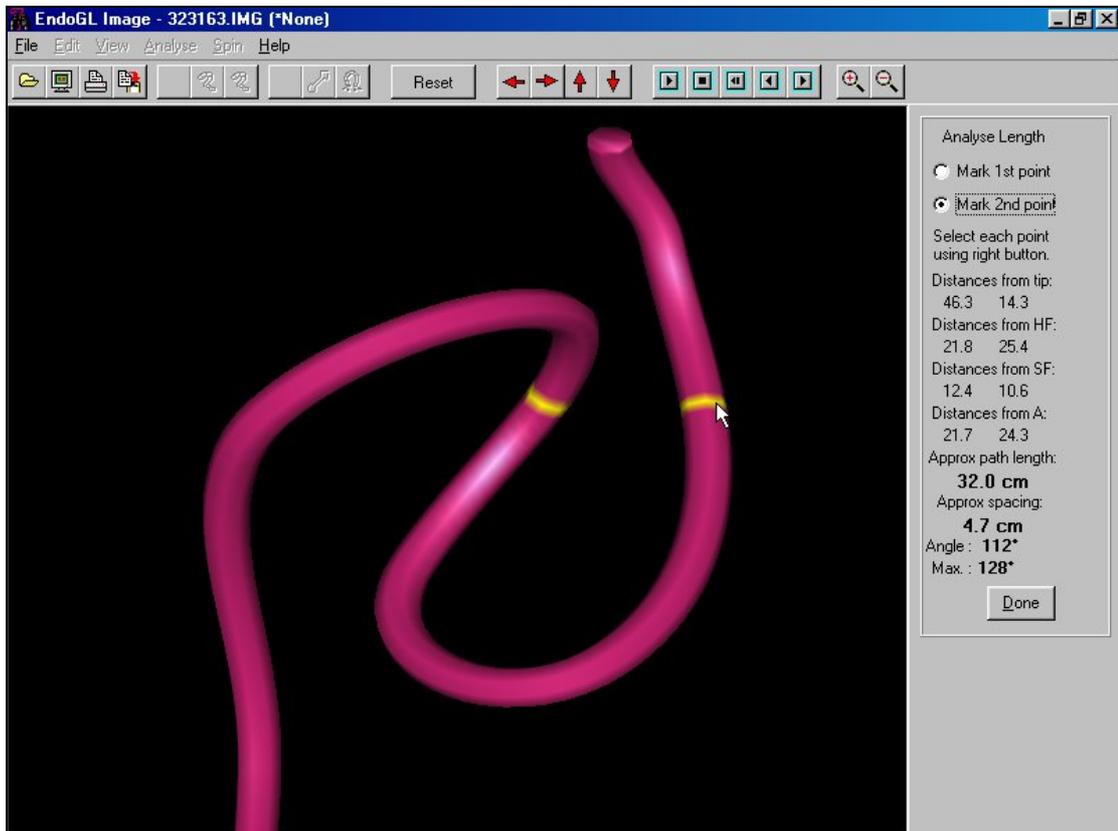
**Figure 5-B – Selecting the first measurement point**

Note that the selected measurement point appears as a bright yellow band on the colonoscope image.

### *5.1.3 Selecting the second point*

Figure 5-C shows the display after the second point has been selected at the position indicated by the mouse pointer. As with the first point, the path length from the instrument tip and straight-line distances to all three markers have been calculated. In addition, the path length and straight-line distance between points one and two have been calculated and displayed.

The system does not make any automatic change to the option selection. This will mean that subsequent right-clicks on the image will redefine measurement point two.



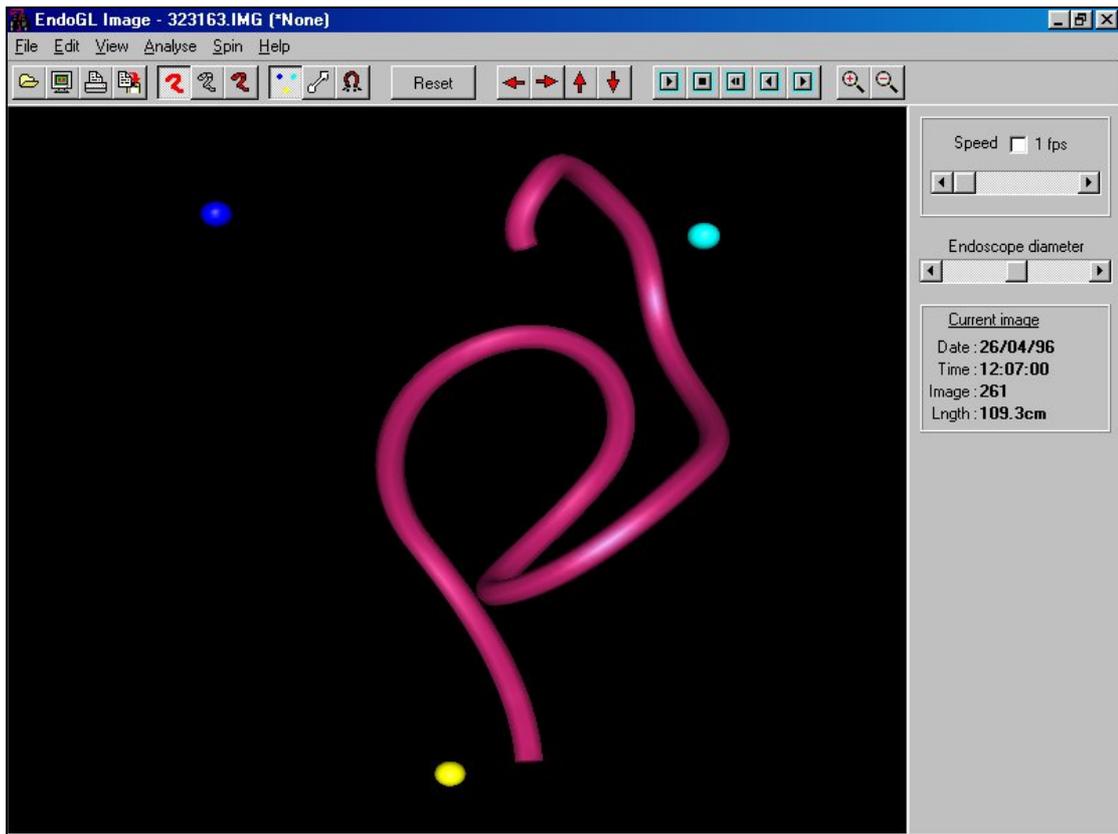
**Figure 5-C – Selecting the second measurement point**

Either of the measurement points may be re-selected by first clicking the required option button in the measurement panel.

#### *5.1.4 Colonoscope length*

The length of colonoscope inside the patient is displayed on the 'normal' display.

Figure 5-D shows an example.



**Figure 5-D – Colonoscope length**

The system attempts to identify the portion of colonoscope inside the patient's body by monitoring the distance of each point on the generated 3D curve from marker three (shown in yellow above). As marker three approximates the position of the anus, the system calculates the length of colonoscope from the instrument tip to the vertex closest to marker three. Although this is performed on the lower resolution image, the accuracy is sufficient to provide a rough guide to the user when reviewing a recorded colonoscopy. Greater accuracy may be obtained by switching to measurement mode and making manual selection of the required points.

## 5.2 Evaluation

### 5.2.1 Methods

Two distinct length measurements need to be evaluated: the straight-line distance between two points and the length of colonoscope between two points.

#### Straight-line distance evaluation

As the straight-line distance between two points is difficult to verify accurately using points on a flexible colonoscope, the system was modified to provide a display of the calculated distances between the three markers.

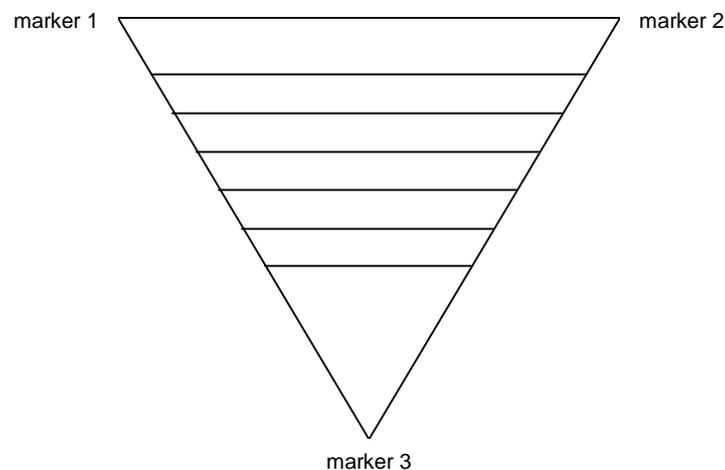
The screenshot shows a software window titled "EndoGL - Statistics". At the top, it displays three inter-marker distances: "Marker 1 to marker 2 = 23.1 cm", "Marker 2 to marker 3 = 34.0 cm", and "Marker 3 to marker 1 = 34.7 cm". Below this, there are two columns of data entry fields. The left column includes fields for SF (14), HF (3564), Time (SF-HF) (785), A-SF2 (87587), A-SF3 (7464), A-HF1, SF-HFcm, Sex, Age, Overtube, and Scope. The right column includes fields for Ulcerative Colitis, Dist SF1, SF1ACF, DescLen, A-Junct 1, A-junct 2, Loop S, Loop trans, Where, Rectum(cm), and Remarks. At the bottom of the window are "Save" and "Cancel" buttons.

**Figure 5-E – Inter-marker distances displayed on the statistics screen**

The 'roving' sensor in the hand-held box used with the Bladen system lent itself more easily to being accurately positioned to provide a stable benchmark. As this

calculation within the system relies on the  $(x, y, z)$  co-ordinates reported by the Bladen imager, the measurement of distances between co-ordinates reported by a single sensor was considered a suitable evaluation method. Figure 5-E shows the statistics screen used to retrospectively record various measurements from the captured colonoscopy session. The inter-marker distances have been added near the top of this display.

To evaluate the accuracy of straight-line measurement, a series of seven isosceles triangles, each representing a set of marker positions were drawn accurately on the back of a perspex model colon. With the point representing marker three constant, the distance between marker three and markers one and two was varied in 5cm decrements from 40cm to 10cm (see figure 5-F).



**Figure 5-F – Benchmark for straight-line distance evaluation**

The hand-held ‘marker’ box attached to the Bladen imager was used to record each set of points, from the largest triangle down to the smallest then back again, with the perspex model positioned in various orientations. The files produced by

the Bladen imager were then loaded into the RMR system and the statistics screen (figure 5-E) displayed for each set to reveal the calculated inter-marker distances. The calculated distances were then recorded and compared with the true distances from the perspex model.

### **Colonoscope length evaluation**

The calculation of path length between two points is affected by the relative spacing between the sensors in the sensor catheter. For this reason, the test was performed twice using sensor catheters with slightly different sensor spacing.

Using the same perspex model colon as above, with the three marker positions recorded as accurately as possible (particularly marker three), a sensor catheter was passed slowly into the model colon while recording the resulting data with the Bladen imager. As the mid-point of each sensor within the catheter passed marker three, the movement was paused for a few seconds. The same procedure was followed as the fully inserted catheter was slowly withdrawn.

The resulting files were loaded into the RMR system and, at each point where the movement was paused, measurement mode was entered and the path length from the tip of the image to the vertex judged to be nearest to marker three was calculated and recorded. These results were compared with the known lengths of the sensor catheters to each of the sensor positions.

At this point, an additional test was performed to assess the accuracy of the calculation of length of colonoscope inside the patient. With the Bladen imager recording, the sensor catheter was advanced into the perspex colon in 10cm increments up to 100cm. At each increment, movement was paused for a few seconds as before. The captured data was again analysed using the RMR system and this time the ‘normal’ display (see figure 5-D) was used to record the calculated length of sensor catheter inside the model colon.

### 5.2.2 Results

#### **Straight-line distance results**

Table 5-A shows the results from the tests for each pair of markers.

<b>Marker 1 - 2</b>	<b>Distance (cm)</b>											
<i>Actual</i>	26.00	26.00	22.60	22.60	19.60	19.60	16.30	16.30	13.10	13.10	9.70	6.70
<i>Calculated</i>	26.70	27.50	24.20	24.60	21.30	21.20	18.10	18.30	14.80	14.80	11.60	8.50

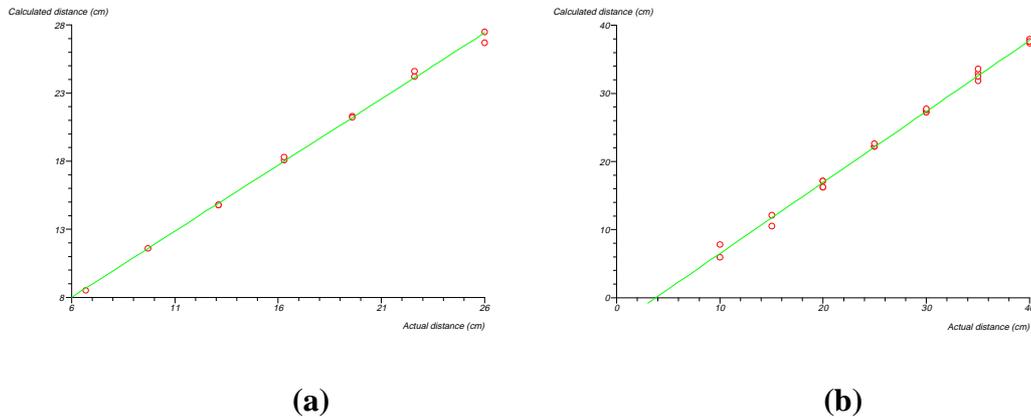
<b>Marker 2 - 3</b>	<b>Distance (cm)</b>											
<i>Actual</i>	40.00	40.00	35.00	35.00	30.00	30.00	25.00	25.00	20.00	20.00	15.00	10.00
<i>Calculated</i>	37.40	37.60	31.90	32.90	27.20	27.20	22.20	22.20	16.20	16.30	10.50	5.90

<b>Marker 3 - 1</b>	<b>Distance (cm)</b>											
<i>Actual</i>	40.00	40.00	35.00	35.00	30.00	30.00	25.00	25.00	20.00	20.00	15.00	10.00
<i>Calculated</i>	37.90	37.60	32.50	33.60	27.60	27.70	22.30	22.60	17.10	17.10	12.10	7.80

**Table 5-A – Results of inter-marker distance tests**

As, in an isosceles triangle, the distances between markers two/three and markers three/one should be the same, this data has been combined to compare the overall result. The correlation charts for the measurements of straight-line distances

between markers are shown in figures 5-Ga (markers one/two) and 5-Gb (markers two/three and three/one).



**Figure 5-G – Correlation of actual and calculated straight-line distances**

The above charts were produced with the statistics package Arcus Quickstat. Using this tool, a simple linear regression analysis gave the correlation coefficient ( $r$ ) for figure 5-Ga = 0.998851, and for figure 5-Gb = 0.998438 both with two sided  $P < 0.0001$ .

### **Path length calculation results**

Table 5-B shows the results of the path length tests when measuring distances from the sensor catheter tip to the estimated mid-point of each sensor position.

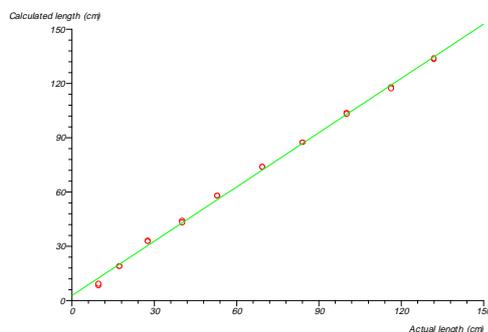
Results are shown separately for both the sensor catheters used. The table is split between measurements made when inserting and withdrawing the catheter. All measurements are in centimetres (cm).

	Sensor	Sensor catheter A		Sensor catheter B	
		Actual	Calculated	Actual	Calculated
Inwards	2	9.70	8.00	9.10	10.30
	3	17.30	18.80	16.50	20.00
	4	27.70	32.80	26.50	31.90
	5	40.20	43.80	39.00	45.00
	6	52.80	58.00	51.50	61.00
	7	69.40	73.80	66.30	75.30
	8	84.20	87.30	81.30	89.50
	9	100.20	103.80	96.30	104.50
	10	116.20	118.00	111.30	119.50
	11	132.00	133.00	126.30	134.50

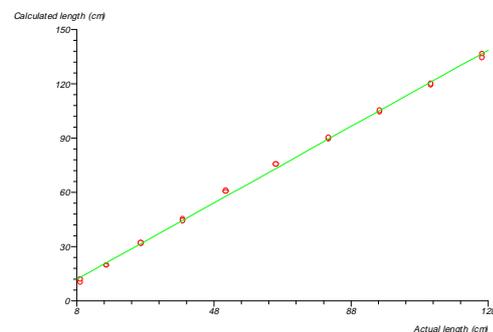
Outwards	2	9.70	9.50	9.10	12.10
	3	17.30	18.50	16.50	19.50
	4	27.70	32.50	26.50	31.90
	5	40.20	43.10	39.00	44.40
	6	52.80	58.00	51.50	60.30
	7	69.40	73.80	66.30	75.30
	8	84.20	87.30	81.30	90.30
	9	100.20	103.00	96.30	105.30
	10	116.20	117.30	111.30	120.30
	11	132.00	133.80	126.30	136.80

**Table 5-B – Results of path length to sensor position tests**

Figures 5-Ha (catheter A) and 5-Hb (catheter B) show the above data in chart form.



**(a)**



**(b)**

**Figure 5-H – Correlation of actual and calculated path length - 1**

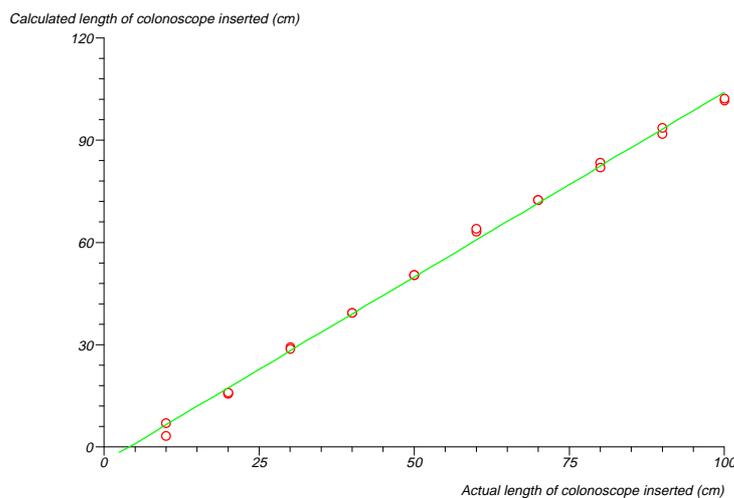
Using Arcus Quickstat, a simple linear regression analysis gave the correlation coefficient (r) for figure 5-Ha = 0.998943, and for figure 5-Hb = 0.999335 both with two sided  $P < 0.0001$ .

Table 5-C shows the results of the path length tests when measuring distances in 10cm increments from the sensor catheter tip to the point nearest marker three. Results are shown separately for measurements made when inserting and withdrawing the sensor catheter.

	Distance (cm)									
<i>Actual</i>	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
<i>Calculated (in)</i>	6.90	15.50	29.40	39.40	50.50	63.30	72.30	83.30	91.80	101.50
<i>Calculated (out)</i>	3.30	16.00	28.80	39.40	50.50	64.00	72.30	82.00	93.50	102.20

**Table 5-C – Results of path length to tip tests**

Figure 5-I shows this data in chart form.



**Figure 5-I - Correlation of actual and calculated path length - 2**

Again using Arcus Quickstat, a simple linear regression analysis gave the correlation coefficient (r) for figure 5-I = 0.998701 with two sided  $P < 0.0001$ .

### **5.3 Comments**

#### *5.3.1 Straight-line distance measurement*

From Bladen (1995) one would expect the positional accuracy of the imaging system – when used with a ‘naked’ sensor catheter – to be correct to within a few millimetres. As the RMR system uses a very simple method to calculate straight-line distances with negligible effect from rounding or precision errors, the accuracy of this measurement method was expected to be very high.

The small discrepancies between the actual and calculated inter-marker distances could possibly be explained by the characteristics of the hand-held calibrating box used to make the measurements. The single 1.6cm positional sensor in the hand-held box (external measurements 7cm by 11cm by 2cm) was centrally placed near its tip but set back by about 0.6cm from the part which was in contact with the model. This made it more difficult to precisely position the sensor on the model. A more accurate placement could possibly have been made if the single sensor were mounted on the end of a ‘wand’ – this would have revealed any inherent inaccuracy in the measurement algorithm or the Bladen imager.

No constraint for the accuracy of this type of measurement was specified in the requirements definition, although it is most unlikely that millimetric precision would be needed – or appropriate – for this application. In such an ‘elastic’

environment undergoing some unknowable degree of deformation at the hands of the colonoscopist, measurements of straight-line distances between sections of colonoscope may only need to be accurate to within a few centimetres to give useful feedback to instrument designers.

### 5.3.2 *Measurement of path length*

The accuracy of path length measurement is affected primarily by the inter-sensor spacing and the number of sub-divisions made when fitting the 3D curve between sensor pairs. During testing, the system was configured to generate 20 such sub-divisions.

As the method of fitting the 3D curve to the sensor positions (described in Appendix A) *guarantees* that the curve length will be equal to the actual length of sensor catheter between sensor pairs, the accuracy of the Bladen imager – in positional terms – is largely irrelevant. As long as the inter-sensor gaps have been measured accurately, the 3D model will reliably sub-divide these lengths to provide a precise measurement to an interpolated point. Using the RMR system, it is possible to highlight the vertices on the model corresponding to the actual sensor positions and to use this information to more accurately select measurement points. When tested in this manner against the known sensor catheter length, the correlation is exact.

The discrepancies in calculated against actual length when using the system to measure the path length to each sensor are possibly due to the difficulty of

visually identifying the vertex nearest to marker three. Of course, inaccuracy in the Bladen imager in reporting the positions of the sensors and markers may also have had an effect but a more likely explanation would be inadvertent movement of the model colon in mid-test. Any movement of the sensor catheter would be reflected in the 3D model whilst the position of marker three would be ‘frozen’ at the same position as at the beginning of each test. This may have resulted in an inaccurate visual identification of the vertex nearest marker three. The exact correlation obtained when using other features of the system to aid identification of the ‘correct’ vertex supports this analysis.

As each inter-sensor gap is composed of 20 sub-divisions, the measurement resolution available depends on the section of colonoscope being measured. The five ‘lower’ sensors in the catheter are spaced some 15 – 16cm apart so the accuracy would potentially be less than with the ‘higher’ sensors which are 7 – 8cm apart. It would be possible to amend the system to generate a variable number of sub-divisions (depending on the inter-sensor gap) to give similar resolution throughout although, in practice, the more flexible instrument tip is the focus of more precise measurement. Future versions of the software will address this in a more flexible manner.

## Chapter 6 Conclusions

### 6.1 Project review

#### *6.1.1 Achievement against objectives*

The objectives stated in section 1.4.2 are repeated below. Whilst all objectives were achieved as stated, section 6.1.2 gives a broader analysis of the success of the project.

- 1. To research the background to the problem area to confirm the need for such an application and to introduce the imaging system to be enhanced*

Chapters 1 and 2 gave a broad background to the problem of colorectal cancer and the importance of colonoscopy as a diagnostic and therapeutic technique. Methods proposed for colorectal cancer screening were compared and a brief outline of existing medical imaging methods given. At this point, the Bladen imaging system and the software to be enhanced (the RMR System) were described.

- 2. To describe the tools and techniques used for 3D computer imaging of the colonoscope.*

Chapter 3 gave a brief introduction to the subject of 3D graphics programming. Whilst this section covered only those areas necessary for understanding the methods used to build the computer model from which the measurements are made, many references are available giving a more detailed view. One such is the excellent textbook by Foley et al. (1995). The second

part of chapter 3 explained in some detail the methods used to generate the colonoscope model within the RMR software. This description was needed to provide a base for the enhancements.

*3. To perform a requirements analysis for the measurement part of the system.*

A brief analysis of requirements was all that was possible for this project. The requirements were simple to state and had no conflicting constraints. The list of requirements was given in section 4.1.

*4. To produce a system design. The design will focus on the additional functionality required with regard to measurement of lengths.*

As the project covered the enhancement of an existing system rather than a new development, the detailed design given in section 4.3 relied heavily on the way the existing system was implemented. This design was based on the computer model described in section 3.2.

*5. To produce a working system based on the above design suitable for evaluation.*

The system was enhanced as designed and implemented as a Microsoft Windows 32-bit application, written in Pascal, using the Borland Delphi 2.0 compiler and the OpenGL API.

The system was supplied to Professor Bell at Sunderland Royal Hospital to be used in conjunction with data captured from the Bladen imaging system available in the Endoscopy Suite at the hospital.

6. *To evaluate the performance and accuracy of the new features of the system.*

With the assistance of Professor Bell, the accuracy of measurements was tested as described in chapter 5. Explanations for anomalous results were proposed and a discussion made of the need for absolute precision in measurement.

7. *To identify strengths and weaknesses of the new system, produce a comparison with alternative systems, if any, and to propose a plan for future development.*

Given that the accuracy of the system – in theory – depends solely on the accuracy of the Bladen imaging system, the major strength of the system is the ability to interpolate measurements along the colonoscope to any required precision although the value of doing so is questionable in the context of colonoscopy.

The accuracy in measurement of colonoscope insertion depth also depends on how closely reference marker three is placed to the patient's anus, and the assumption that the patient has not moved significantly during the procedure. If the placement of marker three is inaccurate or if movement *has* occurred, it will not be evident from such a retrospective analysis of the colonoscopy data except in extreme circumstances. Similarly, the positions of the other reference markers are currently only approximations and are not anatomically correct. One possible solution would be to physically locate sensor coils at fixed points on the patient during the whole procedure and have these points updated in real time along with the colonoscope image.

At the time of writing, no colonoscope imaging system with an accuracy comparable to the Bladen imager is known to exist. The Japanese company Olympus, a major manufacturer of medical endoscopes, have an imaging system which is understood to be similar in nature to the Bladen imager although no details of the features or accuracy of this system are yet available. Certainly, during the course of this project, we have been unable to find any published work covering the measurement of colonoscope behaviour *in vivo*.

Whilst there are many potential applications for a positional imaging system, applications of *measurements* from such a system are more limited. The remainder of this chapter suggests possible enhancements to the current system and some alternative applications.

### 6.1.2 Discussion

Given the importance of colonoscopy as a technique in the diagnosis and treatment of colorectal cancer and its potential as a cancer screening tool to help lower the number of cancers developing in the first place, it is evident that the problems affecting colonoscopists in the execution of their work are similarly important. In the opening chapters of this dissertation several problems affecting the ease or effectiveness of colonoscopy were discussed and the problem of loop formation was highlighted as one of the targets of this project. As much research has already been done in this area, one result being Dr. Bladen's magnetic positioning image system (Bladen et al., 1993, Bladen 1995), there can be little

doubt of the importance of overcoming the current difficulties with colonoscopy. What is less clear is that the end result of *this* project – the ability to retrospectively measure sections of the colonoscope during colonoscopy – will ultimately be of benefit. Certainly there are no such measurements currently available but more research is in progress to investigate the possibilities further.

With the assistance of Professor Bell at Sunderland Royal Hospital, we have completed a study which indicates that this technique is sufficiently sensitive to detect differences between men and women during colonoscopy (Rowland et al., 1999) – a fact known to experienced colonoscopists (Saunders, n.d.). We have also gathered statistics to compare the performance of new, thinner, prototype flexible sigmoidoscopes produced by Olympus – one of the leading manufacturers of medical endoscopes. The results of this study (Bell et al., 1999a) show that it is indeed useful to have *objective* measurements of the behaviour of the instrument during colonoscopy to assist in the comparison of performance. Further studies are currently being planned which may help to decide how useful this work could be to instrument designers, or may at least direct us to alternative methods of analysis.

The RMR system is at present undergoing additional enhancements to provide data relating to the angle of bend of an instrument at certain times and over selected sections of its length. Although the benefit of this data to endoscope designers has yet to be put to the test, one other possibility is that this may lead to more efficient designs of overtube. Again using the RMR system, Professor Bell and colleagues have analysed the effect on colonoscopy of the use of an overtube

(Bell et al., 1999b). Initial results imply that the routine use of a stiffening overtube may improve the speed of colonoscopy. Our intention is to analyse the shape of the combined colonoscope and overtube to determine whether an optimised shape of overtube might be possible.

## **6.2 Future applications**

### *6.2.1 Medical*

An avenue of investigation that has recently been tentatively explored is the use of the Bladen imaging system together with suitable measurement software to aid in the assessment of the rehabilitation of stroke victims or accident victims with similar symptoms.

Where loss of or reduction in limb movement has occurred, it is possible that accurate measurement could be made using similar techniques to those employed in this project to give an objective evaluation as the patient's recovery progresses. Given the potential accuracy of the system, quite small changes could be detected and recorded.

### *6.2.2 Communication*

One area currently the subject of active research concerns sign language. The relevance to this project is perhaps more to the 3D modelling of objects than the measurement of 3D distances.

The system under development consists of a Bladen imaging system connected to sensors incorporated in a glove, rather like a conventional VR 'data-glove' but perhaps more accurate. The data from the imaging system is fed in real time to a PC-based application which 'builds' a 3D model of a hand on a computer screen. This 'virtual' hand then mirrors the gestures made by the person wearing the glove (Dogramadzi et al., 1999).

The possible applications for the capture, transmission and analysis of sign language – or any gesture-based communication are manifold. For example, by transmitting only the sensor positional data remotely to 'drive' an avatar on a connected computer, only minimal communications bandwidth would be required. Other applications may include computer-aided teaching of sign language – large 'dictionaries' or tutorials could be coded and stored in a space-efficient manner by using a discrete number of sensor positions, obviating the need for storing video clips. There are many more possibilities, perhaps beyond the scope of this section, which need not be detailed here.

### 6.2.3 *Sport*

One final possibility where accurate measurement and analysis of 3D movement may be useful relates to improving sporting performance. Analysis of golf swings, gait analysis of humans or horses (on treadmills), tennis strokes, bowling action, even darts players are all possibilities worth at least cursory investigation.

## References

Anon. (1998) '*Home-test kits to be issued for 'taboo' cancer*', Sunday Times issue 1199, 6<sup>th</sup> September 1998.

Anon. (1999) '*CRC CancerStats: Mortality – UK*', Cancer Research Campaign web site <http://www.crc.org.uk> July 1999.

Bell GD, Painter J, Atkin WS, Rowland RS, Dogramadzi S, Allen C. (1999a) *Preliminary experience comparing two thinner prototype Olympus endoscopes with a standard 60cm flexible sigmoidoscope*. Gut 1999; 44: (Suppl 1) A22 (T86).

Bell GD, Rowland RS, Rutter M, Abu-Sada M, Dogramadzi S, Allen C. (1999b) *Colonoscopy aided by magnetic 3-D imaging - Would the routine use of a stiffening sigmoid overtube speed up the procedure ?* Med. Biol. Eng. Comput. 1999 (*In press*).

Bladen JS, Anderson AP, Bell GD, Rameh B and Evans B. (1993) *Non-radiological technique for three dimensional imaging of intestinal endoscopes*. Lancet 341:719-22.

Bladen JS. (1995) *Imaging Medical Endoscopes in three dimensions using magnetic fields*. PhD thesis University of Sheffield.

Cotton P and Williams C. (1996) *Practical Gastrointestinal Endoscopy. Fourth Edition*. Blackwell Science Ltd., Oxford. ISBN 0-86542-851-4.

Debatin JF, Luboldt W and Bauerfeind P. (1999) *Virtual Colonoscopy in 1999: Computed Tomography or Magnetic Resonance Imaging?* Endoscopy 1999; 31 (2): 174-179.

Dogramadzi S, Allen C, Bell GD, Rowland RS. (1999) *An electromagnetic imaging system for remote sign language communication.* IMTC/99 Proceedings of the 16<sup>th</sup> IEEE Instrumentation and Measurement Technology Conference. Venice, Italy. Vol 3 May 1999. IEEE Catalog: 99CH36309 – ISBN: 0-7803-5276-9. 1443-1446.

Foley JD, van Dam A, Feiner SK and Hughes JF. (1995) *Computer Graphics – Principles and Practice – 2<sup>nd</sup> Ed.* In C. Addison-Wesley Publishing Inc. ISBN 0-201-84840-6.

Kettner JD. (1989) *Colorectal Cancer: Epidemiological and Clinical Issues.* Symposium at Royal Society of Medicine, London. Duphar Medical Relations.

Khullar SK, DiSario JA. (1997) *Colon cancer screening. Sigmoidoscopy or Colonoscopy?* In Gastrointestinal Endoscopy Clinics of North America. Evolving Issues in Colon Endoscopy ed. Rex DK. Volume 7, Number 3 , July 1997 pages 365-386.

Kronburg O, Fenger C, Olsen J, Dan Jorgensen O, Sondergaard O. (1996) *Randomised study of screening for colorectal cancer with faecal-occult-blood test.* Lancet 1996; 348: 1467-71.

Mayo Clinic (1996). *New technique detects pre-cancerous polyps of the colon*. URL <http://www.mayohealth.org/mayo/9601/ctcologr.htm>

Microsoft DirectX (n.d.). URL <http://www.microsoft.com/directx/default.asp>

Mosse CA, Mills TN, Bell GD, Swain CP. (1997) *Forces exerted by the endoscopist on the colonoscope during colonoscopy*. Gut 1997; 40 (suppl 1) A40 (abstract).

Mosse CA, Mills TN, Bell GD, Swain CP. (1998) *Device for measuring the forces exerted on the shaft of an endoscope during colonoscopy*. Med. Biol. Eng. Comput., 1998 36, pp. 186-190.

OpenGL (n.d.) URL <http://www.opengl.org>

Robinson MHE, Hardcastle JD. (1998) *Should we be screening for colorectal cancer?* British Medical Bulletin 1998; 54 (No. 4): 807-821.

Rowland RS & Bell GD. (1998) *Non-radiological technique for 3D imaging of intestinal endoscopes: computerised graphical 3D representation of endoscope and skeleton*. Med. Biol. Eng. Comput., 1998, 36, pp. 285-290.

Rowland RS, Bell GD, Dogramadzi S, Allen C. (1999) *Colonoscopy aided by 3D magnetic imaging – Is the technique sufficiently sensitive to detect differences between men and women?* Med. Biol. Eng. Comput. 1999 (*In press*).

Sargent D. (1991) *An Introduction To Program Design*. McGraw-Hill (Europe). ISBN 0-07-707246-4.

Saunders BP. (n.d.) *Making Colonoscopy Easier : Advances In Procedure And Practice*. MD thesis University of London.

Taylor PH. (1994) *3D Graphics Programming in Windows*. Addison-Wesly Inc. ISBN 0-201-60882-0.

Woo M, Neider J, Davis T. (1997) *OpenGL Programming Guide 2<sup>nd</sup> Edition*. Addison-Wesly Inc. ISBN 0-201-46138-2.

## Appendix A - Curve fitting algorithm

The following extract is from John Bladen's Ph.D. thesis entitled 'Imaging Medical Endoscopes In Three Dimensions Using Magnetic Fields' (Bladen 1995). This extract describes the curve fitting algorithm used to generate a set of 3D co-ordinates approximating the path of the endoscope.

*(Bladen, 1995 pp. 7-4 to 7-6) – begin quote –*

### 7.2.2 Developing A Curve Fitting Technique

The output from the position location algorithm is a series of points corresponding to the sensors inside the colonoscope, where each point is described by its position (x, y, z) and orientation ( $\theta$ ,  $\phi$ ). To create an image of the colonoscope configuration, a curve must be fitted through these points. The simplest way to do this is to fit a best fit line through the points using B splines or other curve fitting technique. This method was tried initially and it produces a visually convincing result. However, it is not too accurate since the curve is not guaranteed to go through all the measured points, and corners tend to get 'short-cut'. The solution to this problem was to use the sensor orientation information in addition to the position information and to fit a cubic equation between each sensor point. Three cubic equations are fitted, one for each of the three dimensions. The equations were solved so that the path length of the curve between consecutive points is equal to the physical path length on the colonoscope. This technique enables a curve to be fitted between any given pair of points, without requiring a knowledge of the position of any adjacent points. This is different to the B spline approach, where a knowledge of several points either side is necessary.

Consider the general parametric cubic equation, representing one axis  $x$ ,  $y$  or  $z$ , and with parameter  $p$  representing the distance along the cubic (equivalent to the distance along the colonoscope).

$$f(p) = ap^3 + bp^2 + cp + d \quad (7.1)$$

$$\frac{df}{dp} = 3ap^2 + 2bp + c \quad (7.2)$$

Let the measured positions be at positions  $p_1$  and  $p_2$ . Without loss of generality,  $p$  can be assumed to take the interval 0 to 1 ('normalised' length of colonoscope between points).

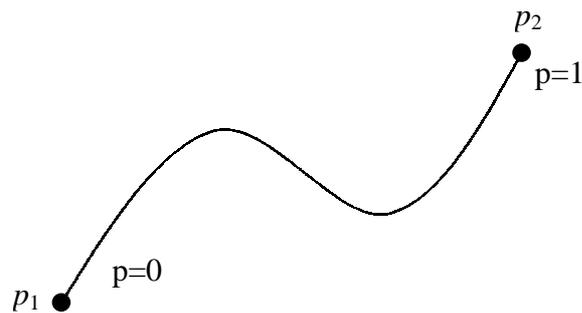


Fig 7.2: Parametric cubic between adjacent points

For the  $x$  axis,

$$f_x(0) = d \quad (7.3)$$

$$f_x(1) = a + b + c + d \quad (7.4)$$

$$\frac{df_x(0)}{dp_1} = c \quad (7.5)$$

$$\frac{df_x(1)}{dp_2} = 3a + 2b + c \quad (7.6)$$

These equations can be combined to produce the coefficients  $a$ ,  $b$ ,  $c$  and  $d$ :

$$a = \frac{df_x(0)}{dp} + \frac{df_x(1)}{dp} + 2(f_x(0) - f_x(1)) \quad (7.7)$$

$$b = 3(f_x(1) - f_x(0)) - 2\frac{df_x(0)}{dp} - \frac{df_x(1)}{dp} \quad (7.8)$$

$$c = \frac{df_x(0)}{dp} \quad (7.9)$$

$$d = f_x(0) \quad (7.10)$$

Substituting the coefficients into the general cubic equation, gives a cubic which describes the path in the  $x$  axis. The function values  $f_x$  are given by the measured positions and its derivatives are calculated from the measured orientation of the sensors at the two points. The equations are similarly derived for the  $y$  and  $z$  axes. By substituting the values of  $p$  between 0 and 1 into the three cubic equations, values of  $x$ ,  $y$  and  $z$  are obtained. In practice around ten points are obtained and straight lines are drawn between them.

This method of fitting cubic equations between measured points is only an approximation. It works reasonably well because the measured points are sufficiently close together that any curves that occur between the points cannot have an order higher than three, hence the use of a cubic equation. This is guaranteed by the mechanical construction of the colonoscope. A better solution would be to use a knowledge of the mechanical behaviour of the colonoscope to calculate the path that the colonoscope must take between two points. This could be done, for example, by

simulating the behaviour of an articulated chain with springs at each junction, fitted between the two known points. The solution could be found by minimising the strain energy of the chain. Although this would be more accurate, it was felt to be unnecessarily complicated for this application, due to the computational overhead and the small advantage gained over the simpler cubic method.

*(Bladen, 1995 pp. 7-4 to 7-6) – end quote –*

# Index

3D graphics .....	31	loop formation .....	17
aim .....	20	measurement	
anus .....	50	design strategy .....	53
position of .....	76	errors.....	84
back face culling .....	38	evaluation .....	77
biopsy channel .....	8, 12, 25	examples .....	72
Bladen imager, the .....	25	results.....	80
Bladen, John .....	18	medical imaging	
camera co-ordinates .....	<i>See</i> eye co-ordinates	three-dimensional .....	23
colon		two-dimensional .....	22
ascending.....	11	object co-ordinates .....	32
descending.....	11	objectives.....	20
description.....	11	review .....	87
gender differences.....	19, 91	octagonal prisms	
racial differences .....	19	generating .....	45
sigmoid.....	11	mitred.....	45
transverse .....	11	Olympus .....	90
colonoscope		OpenGL.....	32, 36, 41
cost.....	16	overtube.....	28
definition.....	8	improving design.....	92
description.....	27	picking function	
design .....	26	implementation .....	69
measurement .....	73	polygon	
colonoscopy		defining.....	38
diagnostic .....	16	polygons	
difficulty.....	17	generating .....	47
procedure .....	15	polypectomy .....	12, 16
therapeutic.....	16	polyps .....	11, 13
training .....	17	removing.....	12
virtual.....	15, 24	primitive	
colorectal cancer		defining.....	37
incidence .....	10	rigid sigmoidoscopy .....	14
screening .....	13	RMR system .....	20, 26, 41
Constructive Solid Geometry.....	38	sensor catheter .....	25
co-ordinate systems.....	35	sigmoidoscopy	
CSG..... <i>See</i> Constructive Solid Geometry		flexible.....	14
curve fitting.....	44	rigid .....	14
data table.....	63	sign language.....	92
DCBE..... <i>See</i> double contrast barium enema		splenic flexure .....	11, 50
digital rectal examination.....	13	sporting performance.....	93
Direct3D .....	32, 41	stroke victims	
double contrast barium enema .....	14	rehabilitation.....	92
endoscope		surface normals	
definition.....	8	calculating.....	48
errors .....	84	description .....	39
eye co-ordinates .....	34	transformations.....	35
facet normals.....	49	user interface	
faecal occult blood testing .....	14	design.....	56
flexible sigmoidoscopy .....	14	point and click .....	57
FOBT .....	<i>See</i> faecal occult blood testing	view plane.....	34
function specifications .....	65	viewing pipeline .....	40
functional requirements .....	52	viewport.....	35
Gouraud shading .....	40	virtual colonoscopy .....	15, 24
hepatic flexure.....	11, 50	virtual trackball.....	50
insertion depth.....	80	wireframe view.....	48
large bowel..... <i>See</i> colon		world co-ordinates .....	33

## Notes

