Real-Time Quantitative Bronchoscopy

Simon Wilson, Brian Lovell
Intelligent Real-Time Imaging and Sensing Group
School of Information Technology and Electrical Engineering
University of Queensland
{sbwilson,lovell}@itee.uq.edu.au

Anne Chang, Brent Masters
Department of Respiratory Medicine
Royal Children's Hospital, Brisbane
{Brent_Masters,Anne_B_Chang}@health.qld.gov.au

Abstract

The determination of motion within a sequence of images remains one of the fundamental problems in computer vision after more than 30 years of research. Despite this work, there have been relatively few application of these techniques to practical problems outside the fields of robotics and video encoding.

In this paper, we present the continuing work to apply optical flow and egomotion recovery to the problem of measuring and navigating through the airway using a bronchoscope during a standard procedure, without the need for any additional data, localization systems or other external components. The current implementation uses a number of techniques to provide a range of numerical measurements and estimations to physicians in real time, using standard computer hardware.

1. Introduction

The field of medicine has undergone significant change in recent years, with advances in technology and understanding able to provide information that is now vital to the diagnosis and treatment of patients and their conditions. Physicians continue to demand higher-quality imaging systems and new techniques to visualize the human body and its conditions. Modalities such as Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) have been able to meet these demands, and now form part of the diagnostic process for many conditions.

There are a number of other imaging modalities which, although they cannot produce the same accurate and repeatable results as MRI or CT can, are just as important. This is particularly true in "visible light" imagery, which rely on direct observation by the operator. The various types of endoscopy (bronchoscopy, laparoscopy, gastroscopy and colonoscopy) are all incredibly useful tools for the minimally-invasive diagnosis and treatment of a number of conditions.

However, there is no way in which the size, location or seriousness of the an injury or disease can be accurately measured or monitored. Obtaining even a rough measurement can be a lengthy and error-prone process. The scale or extent of an injury or illness relies heavily on the interpretation of the operating physician, which may differ greatly from other interpretations depending on a large number of factors. And, should a procedure be recorded, it is impossible to accurately compare two studies, whether they be of the same patient over time to monitor the progress of a particular treatment, or a comparison between patients. Results obtained by "eyeballing" studies can have little meaning or validity depending on the conditions with which these studies were taken.

The aim of this work is to develop a means in which much of this guesswork can be eliminated, and to provide a range of measurement values during an endoscopy procedure, by providing a fast, accurate and repeatable method in which to obtain this information. This will then enable a more numerical approach to the comparison of studies, and allow tracking of a patient's progress over time. In many conditions, there is little or no measurement data available with which to gauge the extent, and the ability to acquire these measurements will allow better diagnosis and selection of treatment options.

The initial work focuses on bronchoscopy, the visualization of the larger regions of the lower respiratory tract. The

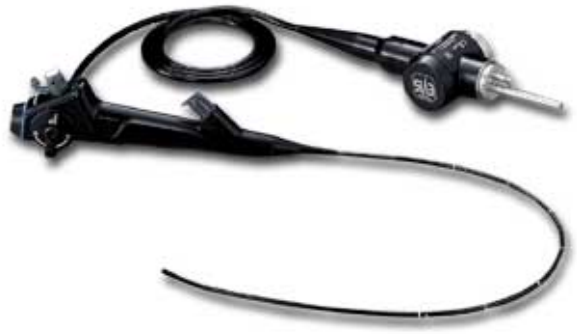

Figure 1. An Olympus flexible videobronchoscope.

goal is to develop a real-time implementation of a system that can aid in the navigation and measurement of the airway, without the need for external means of location, or the modification of existing bronchoscopes.

This paper is organized as follows: An overview of the problems and challenges that must be overcome are presented in section 2. Section 3 describes the principles behind optical flow, the chosen algorithm, and its implementation. Section 4 describes the robust motion estimation algorithm use to convert the recovered optical flow result into an approximation of the motion of the bronchoscope tip as it travels through the airway. Details of the airway area measurement system are presented in section 5. The process of removing the bronchoscope's non-linear image distortion is detailed in section 6. Section 7 provides details on the implementation of this work, section 7 details the future work for this project, and section 8 concludes.

2. Challenges

The majority of today's bronchoscopy procedures are performed using the flexible bronchoscope, developed in the 1960's by the Japanese bronchologist Professor Shigeto Ikeda[10]. His original model used a bundle of optical fibers to transmit images of the lungs from the distal (far) tip of the bronchoscope to an eyepiece so that the physician can view the airway. While these are still in use in some areas, the majority of procedures today use the videobronchoscope, which replaces the fragile bundle with a small CCD camera embedded at the distal tip, which transmits video data to an external processing unit, which adds patient and procedure information to the video, and displays the final image onto a TV monitor.

The very nature of bronchoscopy makes it a challenging method for collecting the desired information. Unlike procedures like CT and MRI, which acquire an image over a relatively short time, a typical bronchoscopy procedure can take a significant period of time. During this time, the lungs continue to expand and contract, causing the trachea and bronchi to deform and move as well. This will further complicate the acquisition and processing of data retrieved from this procedure.

Navigating the bronchoscope into position is a difficult task. The tip can be flexed in one axis by the operator by controls mounted on the handle, and the whole device can be rotated, but this still only gives minimal control over the device, and a great deal of training and practice is required in order to be able to position the device into the desired location. Even gaining access to the airway is difficult, as the anatomy is structured in order to prevent foreign bodies from being inspired.

The size of the airway places further constraints on the quality of results obtained by a bronchoscope. Decreasing the size of the bronchoscope allows it to access narrower regions of the airway, but also limits the size of the CCD device, light sources and instrument port, reducing the resolution and quality of the resulting image. This lower resolution reduces the already limited texturing of the airway, causing the surface to appear blurred, which further impacts the ability to extract usable data.

3. Optical Flow

There have been several techniques proposed for the recovery of image motion within a sequence of images. This is important in a number of fields, such as video encoding, where it is an essential part of a number of compression algorithms. It is also a major area of research in robotics and navigation systems, where visual sensors can be used to obtain motion information.

Optical Flow is one of the most extensively studied of the motion recovery approaches. The origins of optical flow have been attributed to the work of Fennema and Thompson [4], though the term was introduced by Horn and Shunck [7] as the distribution of apparent velocities of movement of brightness patterns within an image, based upon the apparent motion of regions of similar intensity over an image sequence. In its simplest form, this can be expressed as

$$\frac{\partial I}{\partial t} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} \tag{1}$$

To recover the optical flow from a sequence of images, the vector field of this motion, $\vec{o}(x, y)$, must be recovered from the intensity field $I(x, y, t)$. The equation has only one constraint, so a second must be used to obtain a solution. This is typically one of:

- Use a higher-order derivative using additional assumptions about the motion

- Impose a global smoothness constraint to the velocity field, or
- Impose a parametric model to the local velocity field, such as constant or linear variation.

Of the three approaches, the latter two are the most common. Smoothness constraints assume that neighboring groups of pixels will all have the same motion, unless it is occluded by another object within the scene, causing a discontinuity within the flow field. Velocity constraints simplify calculation of the flow field by reducing the search space of the motion range that can be identified.

Three different classes of algorithms have been identified, each using a different method of extracting the flow field from an image sequence. Block matching methods divide the image into a grid of smaller blocks, then, using some form of matching metric such as cross-correlation, attempt to match blocks in two frames from the sequence. This is the simplest of the three methods, but it can perform poorly in situations where there is poor contrast or the images have insufficient texture to track. Phase Correlation methods use the 2D spatial Fourier domain to directly estimate the pixel motion. This is used in a number of video encoding systems, where motion information is used to determine various aspects of the compression. Gradient approaches use a multidimensional image gradient operator to generate image gradient maps, which are used to directly evaluate the optical flow. Within these three classes, there exist a wide range of algorithms, each tailored to a particular application.

The choice of optical flow algorithm is particularly important, as it can greatly affect the results produced. For this project, the Pyramidal Lucas-Kanade method[1], an extension of Lucas and Kanade's original gradient method[8], which uses a multi-resolution approach to improve the performance and overcomes the displacement issues present in the traditional gradient approach. This method provides a flow field over a sparse set of features detected on the surface of the airway. This multi-resolution technique starts by generating a pyramid of images of successively lower resolutions from the original input image, generated by halving the resolution of the image in both *x* and *y* directions by 2, equivalent to using a 5×5 anti-aliasing filter kernel.

The next phase of the algorithm is to track the motion of the selected features between consecutive frames in the image sequence, *I* and *J*. The results of this calculation on the lowest resolution images, I_m and J_m, are used as initial estimates for the next images in the pyramid, I_{m-1} and J_{m-1}. The results of each stage are propagated down the pyramid up until the optical flow has been calculated for the original image sequence. This algorithm performs substantially better than traditional gradient methods, since it allows large feature movements to be tracked through the image sequence, yet is able to provide sub-pixel accuracy in the results, with a reduced computational load. This pyramidal implementation, using a pyramid depth of 4, is able to detect motion that is 31 times larger than would be possible with the traditional non-pyramidal implementation. Unfortunately, due to the filtering inherent in the generation of the filter, smaller details may be lost, and will not be identified as features within the image. As a result, these parts of the image may not be tracked by the application, even if they will make suitable features for tracking.

The determination of the optical flow is performed by the original Lucas and Kanade method, which was originally defined as the image matching error function

$$\epsilon = \sum_{x \in \Re} (F(xA + h) - \alpha G(x) + \beta)^2 \tag{2}$$

where x is an n-dimensioonal row vector, which in the case of an image sequence are the pixel coordinates (x, y); F and G correspond to the functions of the two images $I(x, y)$ and $J(x, y)$; the parameters A and h give the linear transformations of the first image, such as scaling, rotation or shearing; and α and β are the parameters controlling contrast and brightness adjustment. By constraining both α and β, the problem can be further simplified.

An additional enhancement to the standard Lucas-Kanade technique is to implement it in an iterative form, so that each iteration gives successive approximations of the pixel distance $\mathbf{d}$, with each approximation effectively translating the second image J by the initial guess determined in the previous stage of the algorithm, such that

$$J_k(x, y) = J(x + \mathbf{d}_x^{k-1}, \mathbf{d}_y^{k-1}) \tag{3}$$

The residual pixel motion vector $\overline{\eta}^k = [\overline{\eta}_x^k, \overline{\eta}_y^k]$ is then given by

$$\epsilon^k(\overline{\eta}^k) = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} (I(x, y) - J_k(x+\mathbf{d}_x^{k-1}, y+\mathbf{d}_y^{k-1})) \tag{4}$$

This can be presented in the matrix form

$$\overline{\eta}^k = G^{-1}\overline{b}^k \tag{5}$$

where $\overline{b}^k$ is a 2×1 vector known as the image mismatch vector, which is defined as

$$\overline{b}^k = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} \begin{bmatrix} \delta I_k(x, y) I_x(x, y) \\ \delta I_k(x, y) I_y(x, y) \end{bmatrix} \tag{6}$$

and the matrix G is given by

$$G = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \quad (7)$$

where I_x and I_y represent the image derivatives in the x and y directions, and the k^{th} image derivative δI_k is defined for all points within the search window surrounding each pixel $\mathbf{p}$ as

$$\delta I_k(x, y) = I(x, y) - J_k(x, y) \quad (8)$$

The two image derivatives can be precalculated at the beginning of each iteration, so the matrix G will remain constant throughout the entire operation, and only $\overline{b}^k$ needs to be re-evaluated at each stage. This is only true if G is invertible, which occurs only when the image has gradients in both the x and y directions.

Once $\overline{\eta}^k$ has been calculated, the new pixel displacement guess is given by

$$\mathbf{d}^k = \mathbf{d}^{k-1} + \overline{\eta}^k \quad (9)$$

The process will continue until either $\overline{\eta}^k$ falls below a specified threshold, or a maximum number of iterations have occurred. The final solution for the optical flow vector is then given as

$$\mathbf{d}^L = \sum_{k=1}^{K} \overline{\eta}^k \quad (10)$$

The initial selection of features within the image forms a critical part of this algorithm, and the speed, accuracy and robustness of the chosen algorithm can greatly affect the results obtained. Since it is an integral part of the optical flow calculation, the feature detection algorithm can be reused to establish the initial features for tracking, as well as for tracking these features as they move during the image sequence.

To establish the initial feature set, the G matrix is first calculated for each pixel within the image, and the smallest eigenvalue λ_m for each pixel is retained. The maximum eigenvalue λ_{max} is then found within this list, and all λ_m below a threshold (usually 5 or 10%) are discarded. Of the remaining pixels, those which are the local maximum of a 3×3 window are said to be "good to track", and form the set of tracked features. Unlike the optical flow, this small window size is sufficient here to establish the features suitable for tracking. Once the initial features have been identified, a sub-pixel corner detector is used to further refine the position of each feature.

The entire optical flow extraction algorithm provides a sparse flow field at real-time rates. Depending on the hardware and the frame size of the image sequence provided, it can run with minimal delay to the display of the resulting image. However, the accuracy of the optical flow estimation is largely dependent on the ability to robustly track features within this image. Finding a method that robustly track features and matching them with the corresponding features in subsequent image frames will be one of the major foci of this project.

4. Motion Estimation

Once the optical flow has been recovered from a pair of images, the motion that the camera (or equivalently, the scene) has made between the two can be estimated. Since the optical flow is given as a 2D vector field, so some method is required in order to extract the 3D motion and rotation of the camera relative to the scene [2].

Just as with optical flow, there are a number of different algorithms which, given a set of point correspondences between two images, can estimate the motion taken between the two images.

To simplify the mathematics into matrix form, projective geometry is used. This considers a 2D point as the triplet $\mathbf{x} = (x_1, x_2, x_3)$, and a 3D point as $\mathbf{x} = (x_1, x_2, x_3, x_4)$, the so-called homogeneous coordinates [3]. A 2D image can then be considered as a projection of a 3D space. Transforming between the image and world coordinates is performed using the camera's projection matrix P, which contains information on the camera's intrinsic parameters (the focal length, aspect ratio and principal axis projection point), and its extrinsic parameters (its orientation and position in world coordinates). A point in world coordinates X can then be transformed to image coordinates x by

$$x = PX \quad (11)$$

The goal of motion recovery is to estimate the Essential Matrix, E, which contains all the geometric information about the camera's orientation and position between two frames of the image sequence. By determining the camera's calibration matrix C (which is part of the projection matrix P), the essential matrix can be "calibrated" to give the Fundamental Matrix F, which will give results in the units of calibration. It can be determined by

$$F = C^{-T}EC^{-1} \quad (12)$$

Motion recovery algorithms are classified into two general categories: robust, and non-robust, based upon how they treat data points which do not match the motion model[11]. Non-robust models assume that any error in the point correspondences is small, and is spread throughout the entire data set. This approach works well for synthetic data sets, where there are no outliers present, and can produce fast and accurate results. However, they can break down in the presence of gross outliers, as in the case in all real-world

situations. If these outliers can be identified before they are incorporated into the model, the algorithm can compensate or discard them in order to improve the solution.

Least-squares optimization is the most common approach used due to its speed, stability and relative simplicity. However, outliers can cause so much distortion that the final outcome can become an arbitrary fit of the data. In order to remove these erroneous data points from the set, a method to identify them is required. The most commonly used approach is RANSAC, the Random Sample Consensus[5]. Unlike other approaches, which attempt to determine outliers with the entire set of data, RANSAC uses the minimum number of points to test its hypothesis. This minimum set is chosen at random, and is repeated a large number of times using different randomly-selected subsets until the probability that one of these sets contains only inliers is sufficiently high. The best solution is then the estimation that maximises the number of points whose residuals are below a specified threshold. RANSAC assigns a penalty to the outliers, and makes no changes to the weighting of the points identified as inliers. Other algorithms, such as MLESAC and MAPSAC [12] overcome some of the problems with RANSAC's weighting approach by changing the score attributed to inliers as well.

Initially, RANSAC has been chosen for its relative simplicity, and its widespread use in other vision applications. As with the other parts of this system, the effectiveness of RANSAC can be evaluated independently of the other components, and if necessary, it can be replaced by an alternative method.

RANSAC is a general purpose algorithm, which is now finding its way into field other than computer science. In order to use it for a specific problem, it needs to be coupled with a specific hypothesis algorithm to test the data points. There are a range of algorithms that can be used to solve this "relative pose" problem in order to estimate the camera's motion. The number of point correspondences required to solve this vary depending on the particular algorithm chosen, using as few as three points, but most typically use between 5 and 9 points for increased accuracy. These algorithms require the construction of a 1×9 constraint matrix $\tilde{q}$ such that

$$\tilde{q} = [q_1 q_1' \; q_2 q_1' \; q_3 q_1' \; q_1 q_2' \; q_2 q_2' \; q_3 q_2' \; q_1 q_3' \; q_2 q_3' \; q_3 q_3'] \quad (13)$$

where q and q' represent the homogeneous coordinates (q_1, q_2, q_3) from a single feature within two consecutive frames in an image sequence. The constraint matrices for each point are concatenated together to form an $n \times 9$ matrix $\hat{q}$, such that $\hat{q}^T \tilde{E} = 0$. From this, the single-value decomposition is used to extract the fundamental matrix from the column of the right singular matrix that corresponds to the smallest singlular value

$$\begin{aligned} [U, D, V] &= svd(\hat{q}) \\ F &= V[:, 0] \end{aligned}$$

Once the estimate has been obtained, a second single value decomposition is taken of the estimate to ensure the result has a rank of 2, one of the requirements of the fundamental matrix.

$$\begin{aligned} [U, D, V] &= svd(F) \\ F &= U \begin{bmatrix} D_{11} & 0 & 0 \\ 0 & D_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T \end{aligned}$$

This resulting matrix F is the Fundamental Matrix, and contains both the translational and rotational information for the camera motion between the two frames in the sequence. The translation vector t can be extracted by

$$t \sim t_u = \begin{bmatrix} u_{13} & u_{23} & u_{33} \end{bmatrix}^T \quad (14)$$

and the rotation matrix R by either

$$\begin{aligned} R_a &= UDV^T \\ R_b &= UD^TV^T \end{aligned}$$

Since any combination of R and t form a solution, additional constraints are needed in order to produce the correct result. If it is assumed that the first camera projection matrix F_0 is $[I|0]$, and that t is of unit length, then only four possible combinations are valid:

$$P_a = [R_a|t_u], P_b = [R_a|-t_u], P_c = [R_b|t_u], P_d = [R_b, -t_u] \quad (15)$$

Only one of these represents the actual camera motion between the two frames. One of the other solutions is the twisted pair, which can be obtained by rotating one of the images 180 degrees around the baseline, the line joining the center of the camera in the two frames. The other two solutions are reflections of the true solution and the twisted pair. Transforming between the twisted pair and the correct rotation can be performed with the transform

$$H_t = \begin{bmatrix} I & 0 \\ -2v_{13} - 2v_{23} - 2v33 & 1 \end{bmatrix} \quad (16)$$

The reflected views can also be transformed using

$$H_r = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad (17)$$

Selecting the correct orientation is performed by performing a triangulation on a single point, assuming that the scene lies in front of the camera.

5. Area Estimation

Knowing the circumference of the airway is of obvious benefit to respiratory physicians and surgeons, who need to be able to assess a particular condition, and to diagnose and quantify the extent of disorders of the airway. As there is no direct way of acquiring this measurement, the existing technique [9] requires recording the procedure to a video tape on a standard digital video camera. This is then reviewed after the procedure using a firewire-enabled computer, and the desired frames are extracted from the image sequence and imported into ImageJ, a popular Java-based image processing program. Here, a simple manual threshold based upon histogram equalization is used to segment the image to obtain an approximate measure of the airway from the image. The number of pixels within this segmented region are then counted using a flood-fill algorithm provided by ImageJ, and this value is multiplied by a scaling factor determined in previous work.

However, differences in lighting and other factors can cause this segmentation to fail, requiring additional manual adjustment of the threshold values to prevent the threshold from breaking down. Additionally, in order to avoid the effects of the substantial non-linear distortion introduced by the bronchoscope's wide- angle lens, the region of interest must be centered within the image so that the distortion is uniform over the whole region. This whole process can take over an hour to complete for a single image. Since these tasks can be easily automated by a computer, this becomes a simple problem to solve.

The Colour Mode Histogram Technique (CHMT) is implemented directly, and a binary threshold of this is taken. An ellipse fitting algorithm can then be applied to the binary image, and the largest ellipse from the set of detected ellipses gives the area corresponding to the airway. This area can then be flood-filled to remove any holes within the region, and the number of pixels can be counted and multiplied by the scaling factor determined for the particular bronchoscope. This gives a robust method for estimating the

6. Distortion Correction

A crucial part of all aspects of this project is removing the distortion from the image. While the field of view provided by the wide-angle lens provides clinicians with the ability to see a larger area of the lungs, it poses substantial difficulties for extracting useful information from the image sequence.

The wide-angle lens placed in front of the CCD at the distal end of the bronchoscope produces a great deal of "barrel" distortion. The lenses are designed such that the greatest detail is provided by the center of the lens, while still allowing a wider field of view.

There have been a great number of techniques proposed to remove the distortion from a lens. The particular method chosen here is an implementation of the work of Helferty *et al.*[6], who calculate a series of distortion parameters based on a previous calibration exercise based on a grid of points. This method was designed specifically for real-time correction of endoscopy video.

In order to further improve performance, the implementation of this algorithm will be moved onto the video hardware, using a fragment shader to perform this operation.

7. Implementation

This project was initially started as a Microsoft DirectShow filter on the Intel x86 platform.

The decision to move to the PowerPC platform was based on a number of factors. Apple's newest release of OS X, "Tiger", provides a number of new features which have made development significantly faster: CoreData provides a simple integrated SQL database system for storing important data; CoreVideo, CoreImage and the new QuickTime implementation allow for high-performance video and image processing, providing a high-performance processing pipeline that utilizes the system's graphics processing unit (GPU) to off-load work from the CPU; and XCode 2.1 allows software to be written and debugged significantly faster than with Microsoft Visual Studio or other Windows-based compilers.

The application is written using Cocoa, a collection of libraries written in Objective-C, an object-oriented extension to C that is closer to Java than C++. A screenshot from this application can be seen in figure 2. Xcode uses the GCC compiler to allow developers to mix C, C++ and Objective-C in the same source file, allowing legacy code and C APIs such as OpenGL and the legacy SequenceGrabber APIs to be used freely.

The majority of work to date has been in implementing this application so as to allow multiple processing modules to be implemented, and the results of these compared between the modules. Unfortunately, due to the relative infancy of these technologies and the lack of documentation and expertise from other developers, some non-optimal paths have been taken in order to obtain the desired data.

8. Future Work

There is still a great deal that needs to be done before this system can be used in a clinical setting.

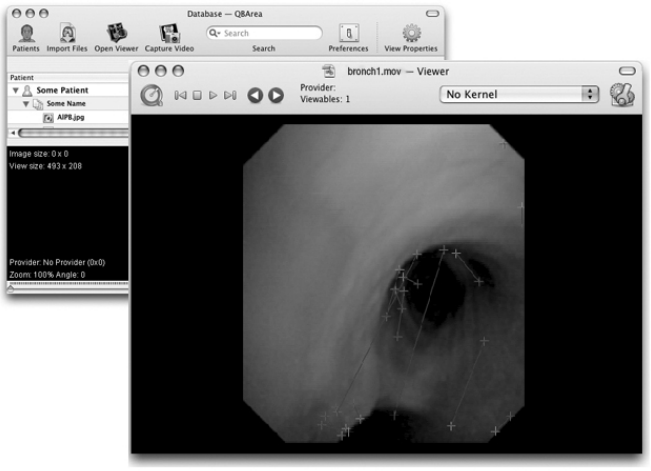

Figure 2. Screenshot of Cocoa application for real-time image and video sequence processing

The first implementation of a distortion correction front-end has been prototyped, and will be added to the application shortly. This will be implemented as a fragment shader for increased performance, since it moves the computation away from the CPU. This stage is crucial in order to obtain accurate, calibrated results.

A high-performance framework application has been developed in order to allow a number of different algorithms and approaches to be quickly written and compared has been developed. This will allow numerous techniques and variations to be explored, allowing us to choose a method that most suits the particular constraints of this particular problem. Challenges such as tracking rapid, jerky movements, occlusion of the lens by fluid and tissue, and working in a low contrast, low texture environment all need to be overcome. A number of areas within this application need additional optimization to improve performance.

The ability to accurately and robustly detect and track features will also play a major role on the performance of this system, and will be the major focus on the remainder of this project. Placing additional constraints on the selection of trackable feature points, and using clusters of features rather than single features will also be explored.

9. Summary

The continuing work into a method for determining distance, rotation and airway size has been presented. By using optical flow and motion estimation on commonly available PC hardware, there is no need for the modification of medical equipment, nor any need for the use of external or internal markers or other measuring equipment. While this project is still in its early stages, the currently available results suggest that this approach will provide a suitable solution to this problem. With additional work, this system may become a valuable tool in the diagnosis and treatment of patients.

References

[1] J.-Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker – description of the algorithm. Technical report, Microprocessor Research Labs, Intel Corporation, 2000.

[2] W. Burger and B. Bhanu. Estimating 3-d egomotion from perspective image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(11):1040–1058, 1990.

[3] T. Davic. Homogeneous coordinates and computer graphics. http://www.geometer.org/mathcircles/cghomogen.pdf.

[4] C. Fennema and W. Thompson. Velocity determination in scenes containing several moving objects. *Computer Graphics and Image Processing*, 9:301–315, 1979.

[5] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[6] J. P. Helferty, C. Zhang, G. McLennan, and W. E. Higgins. Videoendoscopic distortion correction and its application to virtual guidance of endoscopy. *IEEE Transactions on Medical Imaging*, 20(7):605–617, July 2001.

[7] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[8] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *7th International Joint Conference on Artifical Intelligence*, pages 674–679, 1981.

[9] I. B. Masters, M. M. Eastburn, R. Wootton, R. S. Ware, P. W. Francis, P. V. Zimmerman6, and A. B. Chanh. A new method for objective identification and measurement of airway lumen in paediatric flexible videobronchoscopy. *Thorax*, 60(8):652–658, August 2005.

[10] B. K. Reilly, D. Stool, X. Chen, G. Rider, S. E. Stool, and J. S. Reilly. Foreign body injury in children in the 20th century: a modern comparison to the jackson collection. In *8th International Congress of Pediatric Otorhinolaryngology*. British ASsociation for Paediatric Otorhinolaryngology (BAPO), 2002.

[11] P. H. S. Torr. *Outlier Detection and Motion Segmentation*. PhD thesis, Dept. of Engineering Science, University of Oxford, 1995.

[12] P. H. S. Torr. A structure and motion toolkit in matlab. Technical report, Microsoft Research, June 2002.