Visual Odometry for Quantitative Bronchoscopy Using Optical Flow

Simon Wilson[1], Brian Lovell[1], Anne Chang[2] and Brent Masters[2]
[1]School of Information Technology and Electircal Engineering
Intelligent Real-Time Imaging and Sensing (IRIS) Group
University of Queensland
[2]Department of Respiratory Medicine
Royal Children's Hospital, Brisbane

***Abstract*— Optical Flow, the extraction of motion from a sequence of images or a video stream, has been extensively researched since the late 1970s, but has been applied to the solution of few practical problems. To date, the main applications have been within fields such as robotics, motion compensation in video, and 3D reconstruction.**

In this paper we present the initial stages of a project to extract valuable information on the size and structure of the lungs using only the visual information provided by a bronchoscope during a typical procedure. The initial implementation provides a real-time estimation of the motion of the bronchoscope through the patients airway, as well as a simple means for the estimation of the cross sectional area of the airway.

I. Introduction

THE ability to produce accurate, repeatable measurements of the human body is becoming increasingly important in many fields. Some systems, such as modern Magnetic Resonance Imaging (MRI) and Computed Tomography (CT), can provide this information to the operator, without any additional requirements.

However, for a number of other imaging systems, and particularly those that rely on direct visualization by an operator, such as endoscopy (e.g., bronchoscopy, gastroscopy, colonoscopy, laproscopy), obtaining even rough measurement estimates can be a lengthy or complicated process. Making a rough guess of the extent of an injury or a patient's progress over time may in fact be of little use, due to inter- and intra-observer variations. And, even if a procedure is archived by some means for future reference, there is often no other way to accurately compare two procedures over time or between patients than by eye. Simple image manipulation and comparison tools may give a numerical answer, but the vast number of variables in procedures such as these could make any results obtained using such methods invalid.

Our objective in this work is to develop a system that takes the guesswork out of obtaining measurements from any of the endoscopy procedures, and providing a fast, accurate and repeatable method to obtain and compare this information. The initial focus with this work is bronchoscopy, the visualization of the larger regions of the lower respiratory tract. The goal of this work is to provide real-time information during a procedure to physicians, giving the distance traveled by the bronchoscope within the airway, an estimation of the size of the trachea or bronchi, and a rotation guide to help with the positioning and operation of the bronchoscope itself.

The primary measurement principle behind the majority of this system is known as Optical Flow, which is one of several method for extracting the apparent motion in a sequence of images. There are in turn many different implementations of optical flow, with different strengths and trade-offs. The flow field is then provided to a second algorithm, which is used to estimate the three-dimensional motion of the camera relative to the scene, known as Egomotion. The output of this algorithm provides not only a measure of how far the bronchoscope has traveled, but also provides the 3D rotation of the bronchoscope's camera relative to a specified starting location. Estimating the area of the airway is a relatively simple procedure involving basic ellipse fitting, but more advanced methods can give far more accurate results without significant overhead.

Many of these principles and algorithms have already been presented by a number of authors, particularly in the field of robotics, where the recovery of motion from video data can produce results that are far more accurate than more traditional methods such as wheel odometers, due to factors such as wheel slippage from loose or slippery terrain. However, many of these practical applications optimize these techniques and algorithms used to suit the typical conditions that the robot may face. Because of this, much of the findings of other researchers may not be directly applicable to this particular application.

The tools this system provides can all be accomplished using the tremendous processing power available in today's personal computers. By harnessing existing media frameworks and signal processing libraries provided by the operating system of choice and third party developers, such as Microsoft DirectShow and Intel's OpenCV, and the advanced processing features of modern CPUs or video hardware, an efficient and accurate algorithm can be implemented to provide relevant information in real-time, in an easily understandable format, without compromising the safety of the patient, and still provide the original image data for the operator.

The remainder of this paper is organized as follows: The challenges this project must overcome are discussed in section

2. Section 3 describes the principle of optical flow, and how the current algorithm is implemented. Section 4 shows the robust motion recovery used to extract the distance traveled and rotation of the bronchoscope's head as it travels through the body. Section 5 details the method used for the estimation of the cross-sectional area of the currently visible section of airway. Section 6 details some of the future work to be derived from this, and section 7 concludes.

II. Challenges

Most procedures today make use of the flexible bronchoscope, developed in the 1960's by Professor Shigeto Ikeda [1], a Japanese bronchologist. Most modern systems now use videobronchoscopes, which incorporate a CCD sensor at the distal tip of the bronchoscope, replacing the fragile fibreoptic system used in earlier devices. A video processing unit provides high resolution colour images for the physician and other staff through a monitor, which can also be archived to tape with a VCR or video camera. A typical videobronchoscope is shown in figure 1.

Fig. 1. An Olympus flexible videobronchoscope [2].

The respiratory system is not easily accessible due to its anatomy, and that of the surrounding structures. The trachea and bronchi require the use of a narrow, flexible bronchoscope, which limits the size of the CCD image sensor, and hence the image quality and light sensitivity. Motion of the bronchoscope is further hindered by the upper respiratory system such as the pharynx, which contains structures designed to protect the lungs from damage.

Since the final goal of this application is to be of use during clinical procedures, it would be beneficial to use systems that can be easily used by a respiratory physician or assistant during a procedure. To keep costs low, it should be able to run on commodity hardware, so that upgrades and replacement parts are easily available.

III. Optical Flow

Optical Flow is one of a number of methods which have been proposed to extract the apparent motion within an image sequence, but is one of the most extensively studied. Recovering image motion has many other important applications, in fields such as video compression, where it is an essential component of the MPEG encoding process [3].

The origins of optical flow have been attributed to the work of Fennema and Thompson [4], though the term was first defined by Horn and Schunck [5] 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

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

To recover the optical flow from a sequence of images, the vector field of this motion, $\overrightarrow{o}(x, y)$ must be recovered from the intensity field $I(x, y, t)$. Since the equation has only one constraint, a second constraint 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.

The latter two approaches are the most common. The smoothness constraint assumes that neighboring groups of pixels will all have the same motion, except when one region within an image is occluded by another object in the scene, causing a discontinuity within the flow field. Applying a velocity constraint is used to simplify the calculation, by reducing the search space to the motion range specified by the model.

Three classes of algorithms have been developed, depending on the method used to recover the optical flow from an image sequence. Block matching methods divide the images into a grid of smaller blocks, then attempt to compare these blocks in two frames using some form of matching metric, such as cross-correlation. While this is the simplest approach, it can break down in low-contrast and smooth images. Phase Correlation methods make use of the 2D spatial Fourier domain to directly estimate pixel motion, and it is used in a number of video encoding systems. Gradient methods use a multidimensional image gradient operator to generate image gradient maps, which are used to directly evaluate the optical flow. However, this method works for small displacements in the image.

The method chosen initially was the Pyramidal Lucas-Kanade method [6], an extension of the gradient approach, which uses a multi-resolution approach to give a sparse optical flow for a series of feature points detected within the image, and effectively overcomes the displacement issues with traditional gradient approach. A series of images of different resolutions is generated from the original image, each time decreasing the resolution in both the x and y coordinates by a factor of two. This process effectively anti-aliases the image using a filter kernel of 5×5 pixels.

The next phase of the algorithm is to track the motion between consecutive frames within the image, I and J. The results of the optical flow calculations of the lowest resolution images, I_m and J_m, are used as estimates for the calculation of the optical flow within the next images in the pyramid, I_{m-1} and J_{m-1}. This process continues until the optical flow has been calculated for the original image sequences. This algorithm is greatly beneficial in many applications, since it allows large feature movements to be tracked through

the image sequence, but still retains sub-pixel accuracy for each feature's coordinates. By using a pyramid depth of 4, the maximum length of a motion vector can be 31 times larger than is possible to detect with a standard Lucas-Kanade implementation. Unfortunately, due to the filtering of the image, smaller or less prominent features may not be easily detected, since the lowest resolution image, used for the initial feature detection, may simply not include enough detail of the original image.

The optical flow algorithm used here utilizes the original Lucas and Kanade method [7], which was originally defined as the image matching error function

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

where x is an n-dimensional row vector, such as 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 for contrast and brightness adjustment. This can be simplified by simply constraining α and β.

To further enhance the algorithm, the standard Lucas-Kanade method has been implemented iteratively, which is used to obtain successive approximations of the pixel displacement $\mathbf{d}$, with each approximation effectively translating the second image J by the initial guess determined in the previous stage of the algorithm, such that

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

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

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

This can also be presented in the matrix form

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

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

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

the matrix G is given by

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

with I_x and I_y as 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 a pixel $\mathbf{p}$ as

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

Since the two image derivatives can be precalculated at the start of each iteration, the matrix G remains constant throughout the entire operation, and only $\bar{b}^k$ need be calculated at each stage. However, this will only hold true if G is an invertible matrix, which occurs only when the image has gradients in both the x and y directions.

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

$$\mathbf{d}^k = \mathbf{d}^{k-1} + \bar{\eta}^k$$

This process will continue until either $\bar{\eta}^k$ is less than a specified threshold, or the maximum number of iterations has taken place. The final solution for the optical flow vector is then given as

$$\mathbf{d}^L = \sum_{k=1}^{K} \bar{\eta}^k$$

Feature detection is an important aspect of this optical flow system, and the speed, accuracy and robustness of a chosen algorithm can greatly affect the final results. Since feature detection and tracking has been implemented as part of the optical flow method presented here, it is utilized to initially select feature points as well.

The G matrix is first calculated for each pixel within the image, and the smallest eigenvalue λ_m for each pixel is stored. The maximum eigenvalue λ_{max} is found, and all λ_m within a threshold (normally 5 or 10%) of λ_{max} are retained. Of this subset of pixels, those which are the local maximum of a 3×3 window are said to be "good to track", and form the set of features detected by the algorithm. Unlike the optical flow algorithm, which must track specific points through the image, a 3×3 window is sufficient for the initial location of good features. Once the initial features have been located, a sub-pixel corner detector is used to further refine these coordinates.

The algorithm presented here produces a real-time estimation of the optical flow occurring in images, and on its own, runs with only minimal delay on a reasonably modern machine. The implementation has not been hand-optimized, but compiler optimizations do make some use of available vector processing units on the underlying hardware. Further use of this hardware, as well as additional code optimizations, will no doubt improve the performance of this algorithm. However, the validity of using point features within this specific application remains questionable, and the low-contrast environment of the airway further compounds the problem, which can be seen in figure 2.

IV. Motion Recovery

Once the optical flow has been recovered from a pair of images, we would like to know how the camera has moved relative to the scene. Just as with optical flow, there are numerous methods for this given a set of point correspondences, with different benefits and weaknesses. Optical flow gives a 2D motion field, so some method must be used to determine what kind of motion the vector field represents, in order to extract the 3D motion and rotation of the camera relative to the scene [8].

The use of projective geometry, which considers 2D points as a 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 [9],

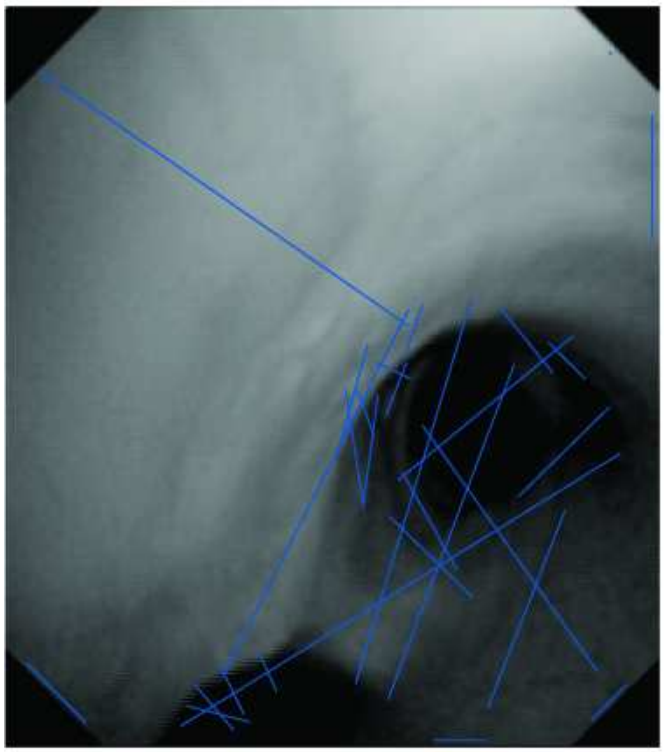

Fig. 2. A sample frame showing matches and displacement vectors between consecutive frames from a Bronchoscopy procedure. The poor contrast within the image highlights the difficulties with tracking features in this environment.

helps to simplify much of the mathematics for this process into matrix forms. An image is then considered as a 2D projection of the 3D scene. Transforming between image and world coordinates is performed using a camera's projection matrix P, which contains information on the camera's intrinsic parameters (focal length, aspect ratio and principal axis projection point) and its extrinsic parameters (orientation and location in world coordinates). A point in space X can then be transformed to image coordinates x by

$$x = PX$$

The goal of motion recovery is to estimate the Essential Matrix, E, which encapsulates all of the geometric information about the camera's position and orientation between two frames of the image sequence. By using a camera's calibration matrix C (a component of the projection matrix P), it is possible to obtain the Fundamental Matrix F, which should yield accurate measurements in the units specified for the calibration, and can be determined by

$$F = C^{-T} E C^{-1}$$

Motion recovery algorithms are classified into two general categories [10], robust and non-robust, based on the treatment of data which does not fit the model. In non-robust methods, these incorrect correspondences, known as outliers, the error is assumed to be small enough to be averaged over the entire data set. These algorithms deal well with synthetic data with no outliers, and can produce fast and accurate results. However, these break down in the presence of gross outliers, as is the case in all real-world situations, and if outliers can be identified before being incorporated into the model, they can be discarded or compensated for in order to obtain a more accurate answer.

Least-squares optimization is the most commonly used due to it's speed and stability, but outliers can cause distortion in the final outcome so much that it becomes an arbitrary fit of the data. In order to discard outliers from their calculations, an algorithm must first identify these outliers. One of the most common algorithms used for this process is known as RANSAC, the Random Sample Consensus [11]. Unlike other methods, which use all available data points to try and determine outliers within the data, RANSAC uses the smallest possible set of data needed to solve the given hypothesis, using points chosen at random. This estimation is repeated on a variety of sets of data, until the probability that one of these sets contains data with only inliers. The best solution to the problem is then the estimation that maximizes the number of points whose residuals are below a given threshold. RANSAC then assigns a penalty to outliers, and no change to inliers. Other algorithms, such as Torr's MLESAC and MAPSAC [12] overcome some of the issues associated with this scoring system. Despite this, RANSAC was chosen for this implementation due to its relative simplicity and widespread use in other vision applications, and it can easily be replaced with another method at a later stage, if required.

RANSAC is a general purpose algorithm, which can be used on a number of problems. In order to use it for a particular application, a specific hypothesis test algorithm must be chosen. For egomotion estimation, a range of equations exist which solve this "relative pose" problem, which can estimate the position of the camera from as few as 3 point correspondances, though they typically use between 5 and 9 points for more accuracy [13]. These algorithms require the construction of a 1×9 constraint matrix $\tilde{q}$, such that

$$\tilde{q} = [q_1 q_1' \; q_2 q_1' \; q_3 q_1' \; q_1 q_2' \; q_2 q_2' \; q_3 q_2' \; q_3 q_1' \; q_3 q_2' \; q_3 q_3']$$

where q and q' represent the homogeneous coordinates (q_1, q_2, q_3) from of a single feature in both images. 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 singular 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

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

The resulting matrix F is then the resultant Fundamental Matrix, and contains both the translation and rotation information for the camera motion between the two frames. We can extract the translation vector t as

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

and the rotation matrix R by either

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

where D is given by

$$D = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Since any combination of R and t are a solution for the problem, due to the epipolar constraints, additional constraints are needed in order to produce the correct result. If we assume that the first camera projection matrix F_0 is $[I|0]$, and that t is of unit length, then only four possible solutions to this problem exist

$$P_a = [R_a|t_u], P_b = [R_a|-t_u], P_c = [R_b|t_u], P_d = [R_b|-t_u]$$

Only one of these combinations represents the true camera motion between the two consecutive frames. Of the remaining 3 options, one represents the twisted pair, obtained by rotating on the views 180 degrees around the baseline, the line joining the center of the camera in the two frames. The other two are reflections of the true configuration and twisted pair. Transforming between the twisted pair and the correction solution can be obtained using the transform

$$H_t = \begin{bmatrix} I & 0 \\ -2v_{13} - 2v_{23} - 2v_{33} & 1 \end{bmatrix}$$

The reflected views can also be transformed using

$$H_r = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

To choose the correct orientation, it is first assumed that the scene lies in front of the camera, then the correct orientation is selected based upon the triangulation of a single point.

V. Area Estimation

Knowing the circumference or area of the airway is of obvious benefit for respiratory physicians and surgeons, who need to be able to gauge the effectiveness of treatment, and the extent of disorders within the airway. The current procedure requires the procedure to be recorded to a miniDV tape using a standard digital video camera. This is then reviewed after a procedure using a firewire-enabled computer, and the desired frames are selected from the tape and imported into ImageJ. Here, a simple manual threshold operation is used to segment an approximate region of the airway from the image, which is then flood-filled, and the number of pixels within this region are counted. However, differences in lighting and other factors can cause this operation to fail, requiring additional tweaking in order to obtain a suitable answer. Additionally, the region of interest must lie within the center of the bronchoscope's field of view, otherwise the substantial non-linear distortion will interfere with the simple scaling factor used to translate pixel count into an approximate area. In all, the process of selecting, segmenting and measuring the size of the airway can take over an hour per image, and cannot guarantee accurate or repeatable results.

All these tasks can be completed in real-time by a computer, with no impact on performance of the visual odometer whatsoever. As with the manual method, a simple binary threshold is used to obtain an approximation for the airway directly ahead of the CCD sensor on the bronchoscope. However, rather than attempting to count the number of pixels directly, each region isolated by the threshold is fit to an ellipse, which should give a good approximation for a healthy airway. Then, by simply selecting the largest ellipse within the image and calculating its area, the result can be achieved in real time during a procedure. This can easily be extended to use an alternative method the segmentation of the airway, and multiple areas could be calculated simultaneously, for cases such as when both the trachea and one or more bronchi are visible in a single image.

In tests with just a standard digital video camera and a simulated airway, such as in figure 3, the system can easily identify the airway, and by adjusting the threshold, the distance down the airway from the camera can be increased or decreased accordingly. Tests with real footage from a bronchoscopy produce show that the system can detect and measure the airway when the image is suitable, but fails under certain conditions. A more robust method is still required in order to overcome some issues such as contrast variations and unusually shaped airways or views. In cases where the camera is not orthogonal to the cross section, some means for adjusting the area may be required. This may not be possible with the currently calculated data, and will be the focus of future work in this area.

VI. Future Work

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

Currently, the distortion produced by the wide-angled lens of the bronchoscope is not accounted for, and all calculations are based on the raw, distorted images. A means of correcting this distortion, using additional pre-processing by the computer, will be needed in order to obtain accurate measurement data from the system.

A specific comparison of a number of the various techniques proposed for both optical flow and camera motion estimation is needed, in order to identify which methods are better suited for this particular environment. Challenges such as the low contrast environment, rapid and jerky movements, and the obstruction of the lens by fluids, tissue or other objects, will all impact the performance of the system. The identification of features to track within the image may also require additional

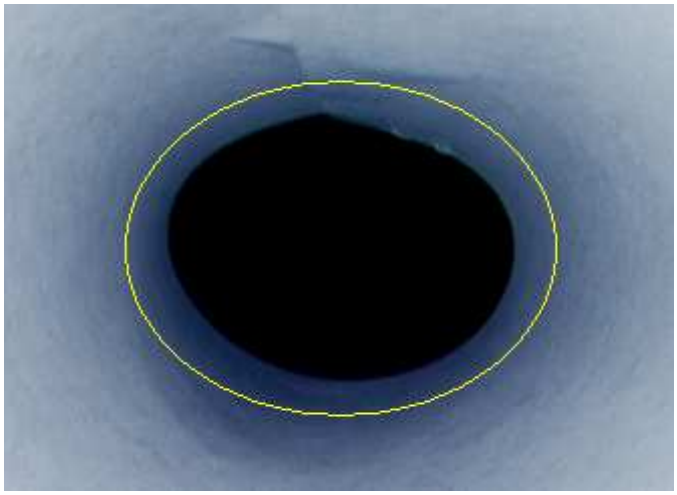

Fig. 3. A sample frame showing the real-time airway area measurement within a simulated airway. In this case, the threshold has been set to find the circumference a short distance in front of the camera's lens.

work, as features lying on contours are not easily tracked by the system, since the identified feature points tend to float along these contours as they move through the image sequence. Additional constraints applied to the regions of the image used to extract motion from, such as the edges of the image, since the wide angle lens shows more the walls of the trachea and bronchi than would otherwise be visible, may assist with the tracking of features within the image sequence. There is also a number of areas where hardware optimizations can take place, utilizing both CPU and video card hardware to increase the performance of this system.

The airway area measurement will also need to be improved. While the current system provides a fast approximation that may be correct in normal circumstances, it will perform poorly in cases where there are deformities or other abnormalities within the airway. By applying a fast, robust contour-finding system, a more accurate representation of the airway's true shape can be obtained, and allow them to be compared between procedures. It also relies on image correction provided by the calibration system in order to produce accurate measurements.

VII. Summary

A system for the measurement of distance, rotation and airway size was presented. By using optical flow, there is no need for the modification of medical equipment, nor the need for external markers or other measuring equipment. While still in early stages of development, the work to date suggests that this method is a valid approach to the problem, and with further work, we believe that the system will be of great value in a number of different procedures.

References

[1] B K Reilly et al, "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.

[2] Olympus America Inc., "Olympus bf-3c160 bronchovideoscope", http://www.olympusamerica.com/, 2005.

[3] Axel Stegner and Reinhard Klette, "Evaluation of mpeg motion compensation algorithms", Tech. Rep., The University of Auckland, October 1997.

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

[5] B. Horn and B. Schunck, "Determining optical flow", *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[6] Jean-Yves Bouguet, "Pyramidal implementation of the lucas kanade feature tracker – description of the algorithm", Tech. Rep., Microprocessor Research Labs, Intel Corporation.

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

[8] Wilhelm Burger and Bir Bhanu, "Estimating 3-d egomotion from perspective image sequences", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 11, pp. 1040–1058, November 1990.

[9] Tom Davis, "Homogeneous coordinates and computer graphics", http://www.geometer.org/mathcircles/cghomogen.pdf, November 2001.

[10] P. H. S. Torr, "Outlier detection and motion segmentation", 1995.

[11] Robert C. Bolles Martin A. Fischler, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

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

[13] David Nistér, "An efficient solution to the five-point relative pose problem", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, June 2004.