Automated Instrument Tracking in Robotically-Assisted Laparoscopic Surgery

Darrin R. Uecker[†], Cheolwhan Lee[‡], Y. F. Wang[‡], Yulun Wang[†]

[†] Computer Motion Inc.
University Business Center
130D Cremona Drive
Goleta, CA 93117

[‡] Department of Computer Science
University of California
Santa Barbara, CA 93106

Abstract

This paper describes a practical and reliable image analysis and tracking algorithm to achieve automated instrument localization and scope maneuvering in robotically-assisted laparoscopic surgery. Laparoscopy is a minimally invasive surgical procedure which utilizes multiple small incisions on the patient's body through which the surgeon inserts tools and a video scope for conducting an operation. The scope relays images of internal organs to a camera and the images are displayed on a video screen. The surgeon performs the operation by viewing the scope images, as opposed to the traditional "open" procedure where a large incision is made on the patient's body for direct viewing.

The current mode of the laparoscopy has an assistant holding the scope and positioning it in response to the surgeon's verbal commands. However, this results in suboptimal visual feedback because the scope is often aimed incorrectly and vibrates due to hand trembling. Computer Motion Inc. has developed a robotic laparoscope positioner to replace the assistant. The surgeon commands the robotic positioner through a hand/foot controller interface (see Figure 1.b in Introduction). To further simplify the man-machine interface in controlling the robotic scope positioner, we report here a novel scope positioning scheme using automated image analysis and robotic visual servoing. The scheme enables the surgeon to control his visual feedback and be more efficient in performing surgery *without requiring additional use of the hands*.

Key Words: Laparoscopy, Endoscopy, Image Analysis, Robotics, Tracking

¹ This project was supported in part by a State of California MICRO grant, 93-158.

Point of Contact: Professor Y. F. Wang, Department of Computer Science, University of California, Santa Barbara, CA, 93106. Phone: (805) 893-3866. Fax: (805) 893-8553. E-mail: yfwang@cs.ucsb.edu.

1 Introduction

The objective of this research is to develop *practical and reliable image analysis and tracking algorithms* to achieve automated instrument localization and scope maneuvering in robot-assisted laparoscopic surgery. There has been a revolution in medical surgery in recent years toward "minimally invasive surgery" [6]. Minimally invasive surgery reduces the trauma inflicted on the patient during surgery, significantly shortens the time for the patient to recuperate, and lowers the cost of the treatment. Because of the benefit gained over traditional surgical procedures, minimally invasive surgery is fast gaining popularity.

A key technological advance which has fueled the minimally invasive revolution is video-laparoscopy [5]. Laparoscopic procedures are minimally invasive surgical procedures where several small incisions are made on the patient to accommodate surgical instruments such as scalpels, scissors, and staple guns. A video laparoscope is inserted through the navel to acquire video images of the body cavity which are displayed in real time on a

Figure 1: (a) Traditional laparoscopy performed by a surgeon and a scope assistant, and (b) robotically-assisted laparoscopy where a robot replaces the scope assistant.

monitor to provide visual feedback to the surgeon. This setup enables the surgeon to operate instruments through the small incisions, as opposed to a large incision for direct viewing and operation.

This change to a video-based operating paradigm opens the door for tremendous innovation opportunities. Since the surgeon has become comfortable operating through a video interface, it is now possible to provide computer assistance with added functionality. This project is thus aimed at bringing state-of-the-art in imaging and robotic technologies to enhance this emerging robot-assisted surgical procedure.

The current mode of laparoscopic surgery is that an assistant holds the laparoscope for the surgeon and positions the scope in response to the verbal directions from the surgeon (Figure 1.a). The method of operation is inefficient and frustrating for the surgeon because commands are often interpreted and executed erroneously by the assistant. The views are suboptimal and unstable because the scope is aimed incorrectly and vibrates due to hand trembling. Consequently, a waste of man power and a high risk to the patient result.

To improve on the current mode of laparoscopic surgery, Computer Motion Inc. has recently developed an automated scope positioning system called AESOP, for Automated Endoscope System for Optimal Positioning [9]. The AESOP idea is to have a robot holding the scope and responding to the positioning commands issued by the surgeon using a hand/foot controller (see Figure 1.b). This mode of operation improves the visual feedback to the surgeon by giving the surgeon direct control of his visual feedback and eliminating the assistant from the loop. The procedure can thus be performed faster and with greater ease.

In this project, we use AESOP as the testbed for developing useful image analysis and tracking modalities to further facilitate laparoscopic surgery. More precisely, this project is aimed at developing novel “handless” paradigms for improving the ease of the surgeon to control his visual feedback. This is accomplished by automating the scope positioning and aiming mechanism and thus freeing the surgeon from using the hand/foot controller. The first steps in achieving automated scope positioning, we believe, are

1. to implement practical, real-time image analysis algorithms for locating and tracking surgical instruments in laparoscopic image sequences, and
 2. to use the output from image analysis to automate the process of maneuvering and aiming the laparoscope.
- The AESOP robot can servo to compensate for the movements of an instrument and has the instrument

centered under the field of view of the scope.

We have digitized a large number of image sequences taken during real laparoscopic surgery to serve as our test data. To accomplish the first goal above, we have obtained a large sample of color signatures from over a hundred thousand instrument and organ pixels in these sequences, and have computed the color statistics of various instruments and organs. These color statistics have allowed us to classify, group, and label instrument pixels. Furthermore, we can compute the shape and motion parameters of each instrument region detected in an image sequence.

To accomplish the second goal above, we have used the 2D shape and motion parameters of an instrument computed from image analysis to derive the control signal for AESOP to maneuver and aim the scope automatically. We have derived a mathematic formulation which relates the change of appearance (i.e., shape and location) of an instrument to the scope's degree of freedom in motion. When the tracked instrument's position and/or shape measurements deviate from the desired, canonical ones (e.g., the instrument is too far from the center of the image or becomes too small), an error signal is generated. AESOP then uses the error signal to compute and execute a movement that compensates for the deviation automatically.

The remainder of the paper is organized as follows: Section 2 discusses the image analysis algorithm for instrument localization and tracking from an laparoscopic image sequence. Section 3 presents the mathematic framework for the AESOP robot to maneuver and aim the scope based on 2D image measurements. Experimental results based on both real image sequences and computer simulation are described in Section 4. Finally, Section 5 contains a concluding remark.

2 Automated Instrument Localization and Tracking

The algorithm for instrument localization and tracking in laparoscopic sequences iterates through four steps: (1) *classification*: pixels are classified based on their color signatures and spurious noise points are removed through directional median filtering; (2) *grouping and labeling*: instrument pixels are grouped together and a unique label is assigned to each instrument region. Furthermore, new labels are generated for instruments just appearing in view; (3) *shape analysis*: useful information, such as the centroid and orientation, of an instrument region is computed and a bounding box is estimated; and (4) *temporal update*: movement of an instrument's bounding box between successive frames is estimated and used to predict its location in the next frame to propagate the instrument labels over time. We describe these steps in more details below.

2.1 Classification

After consulting with practitioners in laparoscopy and studying many video tapes of laparoscopy, we concluded that when there is sufficient lighting in an image (the laparoscope is equipped with its own light source for illumination), instruments and organs possess distinct color signatures. Furthermore, instruments have more elongated shapes, and the shaft of an instrument is always of a circular cylindrical shape (so that an instrument can pass through the cannular opening on the body). This domain-specific knowledge was utilized in designing the image analysis algorithm.

The first stage in our algorithm is to utilize different color signatures of organs and instruments to classify individual pixels. This is accomplished by training a classifier on a large sample (over 100,000) of pixels from over ten typical laparoscopic sequences, and using the color statistics thus obtained for classification.

We employ a standard Bayesian classifier which maximizes the *a posteriori* probability of the class assignment [2]. Mathematically speaking, a test sample is characterized by a d -dimensional measure vector, $\mathbf{x}$, which should be representative of the sample. Each possible class assignment ω_i has an *a priori* probability $p(\omega_i)$, which represents the class' relative frequency of occurrence. Furthermore, a conditional probability $p(\mathbf{x} | \omega_i)$ expresses the likelihood of a class sample assuming a particular measure vector $\mathbf{x}$. A Bayesian pattern classifier selects a

best label ω_i for a test sample of measure vector $\mathbf{x}$ from a set of N possible class assignments by maximizing the *a posteriori* probability using the Bayes rule. The *a posteriori* probability of a class ω_i is defined as:

$$p(\omega_i | \mathbf{x}) = \frac{p(\mathbf{x} | \omega_i)p(\omega_i)}{p(\mathbf{x})},$$

where

$$p(\mathbf{x}) = \sum_{j=1}^N p(\mathbf{x} | \omega_j)p(\omega_j).$$

The class assignment which attains the largest $p(\omega_i | \mathbf{x})$ is chosen.

In our case, there are two classes: organ (ω_1) and instrument (ω_2), hence, $N = 2$. We estimated the *a priori* probabilities $p(\omega_1)$ as 0.7, and $p(\omega_2)$ as 0.3. This corresponds to the typical scenarios where the surgeon operates two instruments simultaneously and each instrument occupies roughly 15% of the image space. Each sample (i.e., a pixel) provides a 3-dimensional measure vector $\mathbf{x} = (R, G, B)$, where the components represent the red, green, and blue color intensities. Furthermore, we assumed $p(\cdot | \omega_i), i = 1, 2$ to be a multi-variate normal distribution with mean μ_i and variance Σ_i , and estimated $\mu_i = \sum_{j=1}^{N_i} \mathbf{x}_{ij} / N_i$, $\Sigma_i = \sum_{j=1}^{N_i} (\mathbf{x}_{ij} - \mu_i)(\mathbf{x}_{ij} - \mu_i)^T / (N_i - 1)$, where $\mathbf{x}_{ij}$ represents the measure vector of samples in class i , and N_i the sample size of class i , $i = 1, 2$. Then a pixel, with a measure vector $\mathbf{x}$, is assigned to class i if $\log p_i(\mathbf{x}) > \log p_j(\mathbf{x})$, where

$$\log p_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \mu_i)^T \Sigma_i^{-1}(\mathbf{x} - \mu_i) - \frac{3}{2} \log 2\pi - \frac{1}{2} \log |\Sigma_i| + \log p(\omega_i).$$

Some sample classified images are shown in Figures 7.b and 8.b in Section 4. To remove spurious noise in these images, we use four directional (vertical 5×1 , horizontal 1×5 , $45^\circ 5 \times 5$, and $-45^\circ 5 \times 5$) median filters to filter the images. Some typical outputs from median filtering are shown in Figures 7.c and 8.c.

2.2 Grouping and Labeling

The classification result is stored as a binary image. We then assign a unique label to pixels in each localized instrument region. The pixel identity is determined in two phases: a temporal propagation phase where the instrument labels from the previous frame are propagated to the current frame, and a second spatial relaxation phase where the identities of still unlabeled instrument pixels are determined by their spatial adjacency to already-labeled instrument pixels.

As will be discussed in Section 2.4, we predict the locations of instrument regions in the current frame using their positions in the previous frame and the anticipated movement. The anticipated motion of an instrument is computed as a weighted average of the instrument motion over a few previous frames. Instrument pixels, detected in the current frame and whose positions confirm to the predicted instrument locations, assume the labels of the instrument pixels of the previous frame.

As we anticipate error in this simple motion prediction scheme, we go through another iteration where still unlabeled instrument pixels copy the labels of nearby instrument pixels. This process is performed in the spatial domain through relaxation. At each iteration, an unlabeled instrument pixel assumes the label of its neighboring instrument pixel, if such a pixel exists and is labeled. The number of iterations depends on how much an instrument can move between successive frames. More iterations are needed to propagate instrument labels over a long distance if large instrument motion is expected. Our experience indicates that at half the video rate (the top sampling rate of our hardware), movement of an instrument in any direction in the image plane is usually less than 10 pixels per frame. So the relaxation step is limited to ten.

Finally, if there are still unlabeled instrument pixels left after the temporal and spatial propagation steps, these pixels are either discarded as spurious noise points, or registered as a new instrument in the next shape analysis stage.

Figure 2: Computation of the Bounding box.

2.3 Shape Analysis

Each localized instrument region is fitted with a bounding box. The needs for a bounding box approximation are: (1) Due to uneven lighting and specular reflection, a localized instrument region may assume a somewhat irregular shape (For example, see the instrument region at the lower left corner in Figure 8.c). However, it is known that the shaft of an instrument is of a cylindrical shape, which dictates its appearance in the image plane. For example, when an instrument is far away from the scope (the far field case), ideally its projection assumes a rectangular shape. When one end of an instrument is much closer to the scope than the other (the near field case), its projection resembles a trapezoid. This domain knowledge can be used to regularize image processing; (2) The bounding box representation is more efficient in storage. Shape characteristics, such as the centroid and tip locations, are useful parameters in controlling the aiming of the scope; and (3) Goodness in the bounding box approximation can be used to identify instruments just entering the field of view. It can also be used to discard large dark areas in images which have a color signature similar to that of an instrument region, but assume a much more irregular shape.

We compute the 0th, 1st, and 2nd-order moments for each instrument region. If the ratio of M_{20} to M_{02} exceeds a certain threshold—which indicates an elongated shape—the region is classified as an instrument region, where M_{20} and M_{02} denote the second-order moments computed in the object-centered coordinate system [4].

In the far field case, we compute the bounding box as a rectangle of width $2\sqrt{M_{20}/(M_{00}-1)}$ and height $2\sqrt{M_{02}/(M_{00}-1)}$ (Figure 2), where $\sqrt{M_{20}/(M_{00}-1)}$ and $\sqrt{M_{02}/(M_{00}-1)}$ are used to approximate the spread of instrument pixels along the two principal axes. Two times the spread cover almost 100% of an instrument region. In the near field case, the bounding box is assumed to be a trapezoid whose base is twice as wide as its top. Let w and h denote the lengths of the base and height of such a trapezoid, one can easily show that

$$M_{00} = \frac{3}{4}wh, \quad M_{20} = \frac{3}{64}w^3h.$$

Based on the above equations, we compute w and h as functions of M_{00} and M_{20} , or the length of the base is $4\sqrt{M_{20}/M_{00}}$ and that of the height is $M_{00}^{3/2}/(3\sqrt{M_{20}})$ (Figure 2).

2.4 Temporal Update

Finally, positions of the bounding boxes of the instrument regions in the next frame are predicted for propagating instrument labels to the next frame. Denote the positions of the four corner points of a bounding box at the current

frame as $\mathbf{cp}_i(t) = (cp_{x_i}(t), cp_{y_i}(t))$, then $\mathbf{cp}_i(t+1)$ is estimated using a simple formula

$$\mathbf{cp}_i(t+1) = \mathbf{cp}_i(t) + \sum_{k=0}^{K-1} w(k)(\mathbf{cp}_i(t-k) - \mathbf{cp}_i(t-k-1)),$$

where K is the number of image frames used in predicting the instrument movement, and $\sum_{k=0}^{K-1} w(k) = 1$. In our experiment, $k = 4$, and $w(0) = 8/15, w(1) = 4/15, w(2) = 2/15, w(3) = 1/15$. The predicted positions of the corner points are used to estimate the location of an instrument region in the next frame. A template image with updated instrument regions is generated. Pixels retain their labels in this template image and this image is used to aid in the grouping and labeling process described above.

2.5 Specular Reflection

One implementation detail worth mentioning is how specular reflection is handled. Laparoscopic images at times show strong specular reflection on both organ and instrument surfaces because a strong, focused light source is used for illumination, and bodily fluid tends to accumulate in the abdominal cavity. Because the color of specular reflection is that of the light source, the color classification scheme does not work well and it can be difficult to tell whether a specular spot is part of an instrument or an organ based on the color information alone. Here, we rely mostly on temporal and spatial propagation processes described in Section 2.2 to classify specular spots. That is, pixels assuming specular color are classified as instrument pixels if these pixels confirm to the instrument locations predicted through temporal propagation. Specular reflection spots can also be merged into an instrument region if it is spatially adjacent to that region.

2.6 Time Complexity

We give an estimate of the rate of image analysis in terms of floating-point and integer operations needed. Denote T_I the time to execute an integer operation, and T_F the time to execute a floating-point operation. We assume that there are N_i instrument clusters and N_s specular spots in an image, and the size of the image is N_p pixels.

In the classification stage, the pattern classifier is trained off line and results stored in a table. Classification based on color signature thus uses a table look-up process and no floating-point operation is needed. Processing time for pixel classification is thus $2N_pT_I$, or one indexing and one assignment operations per pixel.

Most of the processing time is spent on median filtering, and on grouping and labeling. Filtering an image with four directional median filters takes $48N_pT_I$ operations per image frame, because, for each median filter, five fetch and add ($10N_pT_I$), one comparison (N_pT_I), and one assignment operations (N_pT_I) are needed. For spatial relaxation, a maximum of $9KN_pT_I$ operations are needed, where K is the number of iterations in the spatial relaxation process. The constant 9 comes from one comparison operation for each of the eight neighbors plus one more class assignment. Another $3N_pT_I$ operations are needed for the temporal propagation step where instrument labels are propagated from the previous frame to the current frame by a simple logic operation, which involves two comparisons to assert that pixels in the current frame and in the temporal template are both of the instrument type, plus one possible assignment operation.

To process instrument pixels left unlabeled after temporal and spatial propagation, $18N_pT_I$ operations are needed, where one comparison is for locating an unlabeled pixel, one comparison and one assignment operations per neighbor for the 8 neighbors to find a minimum label, and one assignment for storing the chosen label. Computing the moments of an instrument region has a complexity of $4N_pT_I$, where one logic operation is used to ascertain the identity of a pixel and 3 additions are for computing the area and centroid. Furthermore, $10N_pT_I$ operations are used to compute the principal axis directions. One notes that these operations require scanning images pixel by pixel and can be time consuming. However, one also notes that because of the regularity of these algorithms, they can be easily parallelized to achieve a significant speedup.

Figure 3: Expected performance of the image analysis algorithm

Computing the bounding box dimension and location, and computing the instrument motion are constant time operations independent of the size of the processing window. We estimate the complexity to be $47N_iT_F$. Finally, a maximum of $85N_pT_I + 47N_iT_F + 16N_iN_sT_I$ operations are needed to process specular reflection spots in the worst case: classification ($2N_pT_I$), filtering ($48N_pT_I$), labeling from temporal propagation ($3N_pT_I$), assigning a new label ($18N_pT_I$), computing the shape parameters ($4N_pT_I$), estimating the bounding box ($10N_pT_I + 47N_iT_F$), and merging into a nearby instrument region ($16N_iN_sT_I$).

Therefore, the total computation time is $(170 + 9K)N_pT_I + 94N_iT_F + 16N_iN_sT_I$. If $N_i = N_s = 1$ and $K = 10$, approximately 12,000 pixels can be processed by our algorithm every 30th of a second on a computer rated at 20 MFLOPS and 100 MIPS. Figure 3 shows the estimated response rate as a function of the size of the processing window and the speed of the computer. Each curve in Figure 3 in fact represents a family of curves where N_i can range anywhere from 1 to 10. That is, because most of the operations are repeated over the whole image frame, the number of instrument regions does not significantly affect the processing speed. This plot allows us to select the processing window size that maintains the desired processing speed.

3 Visual Servoing using Image Analysis Feedback

This section describes the mathematical framework for utilizing the image analysis algorithm described in the previous section as a feedback mechanism for controlling AESOP. The goal is to enable the surgeon to command AESOP to position an instrument feature, such as the tip of the instrument, at a desired location in the image and then track the feature as it moves. This capability is a crucial step towards creating a seamless man-machine interface between the surgeon and the AESOP system.

Visual servoing of a robot manipulator has been studied by a number of researchers [7, 3, 10, 1]. Visual servoing as applied to robotic control of an laparoscope can be defined as: “Given the current location of the tracked feature (e.g., the tip of an instrument) in the image plane, how do we manipulate the scope so that the feature appears at the desired image location (e.g., center of the image)?” The general form of the visual servoing algorithm drives the camera motion based on error between the desired and current feature locations in the image plane. Hence, the control law used by the servoing algorithm must relate the error in feature position to an appropriate camera command which is then implemented by the robot.

Previous research on visual servoing has assumed the camera is capable of general motion in 3-space, i.e., any position and orientation (pose) [7]. However, the physical constraint imposed on the scope and hence, the camera position, presents new problems for this application. In laparoscopic surgery the camera is typically mounted on

Figure 4: Sliding constraint imposed by the pivot point on the laparoscope.

the back end of the scope, outside of the abdomen, and the image is projected onto the camera CCD imaging chip through the scope optics. Because the laparoscopic optics simply translate the image as seen at the tip of the scope inside the abdomen, we can assume that the camera reference frame, denoted as $\{\mathbf{R}_c\}$, is centered at the tip of the laparoscope. The constraint imposed by the abdomen entry point, or pivot point, prevents direct positioning of $\{\mathbf{R}_c\}$ inside the abdomen because AESOP controls the position of the opposite end of the laparoscope which lies outside of the abdomen. The sliding constraint imposed on the scope position by the pivot point is demonstrated in Figure 4.

Because of the constraint imposed on the camera by the pivot point, the Cartesian coordinate system (x, y, z) is not an appropriate reference frame for manipulating the laparoscope. Thus, we represent the motion of the laparoscope and therefore the camera in spherical coordinates relative to the world coordinate frame, $\{\mathbf{R}_o\}$, located at the pivot point. A representation of the coordinate frames and their relationship to the image plane is shown in Figure 5. The pivot point allows 3 degrees-of-freedom, (θ, ϕ, ρ) , for manipulating the camera. This coordinate system makes sense from an application point-of-view as it defines the typical camera motions: zooming in/out is a change in ρ , panning left/right is a change in θ , and panning up/down is a change in ϕ .

A relationship between errors in feature location and commands to position the laparoscope in spherical coordinates has been derived. This relationship is the Jacobian, $\mathbf{J}$, which translates velocities of an image feature, $\mathbf{v}_i = [\dot{x}, \dot{y}]^T$, to velocities of the laparoscope in spherical coordinates of the world frame, or $\mathbf{V}_s = [\dot{\theta}, \dot{\phi}, \dot{\rho}]^T$. The derivation of $\mathbf{J}$ is as follows: We consider a target defined by an image feature located at a point $\mathbf{P}$ in 3-space. Our formulation uses a pinhole camera model which has the camera reference frame $\{\mathbf{R}_c\}$ attached to it. We also assume a perspective projection onto the image plane with focal length f . A point $\mathbf{P}$ with coordinates $[X_c, Y_c, Z_c]^T$ in reference frame $\{\mathbf{R}_c\}$ projects onto a point $\mathbf{p}$ in the image plane with image coordinates $[x, y]^T$ given by,

$$x = f \frac{X_c}{Z_c} \quad y = f \frac{Y_c}{Z_c} .$$

This is the ideal equation for this projection. We can include factors to account for image sampling and image coordinate translation; however, we will use the above equation to simplify the notation without loss of generality.

We can also describe the motion of the object in the image plane, assuming it is rigid, with respect to the motion of the camera. Assuming the motion of the camera given by a translational velocity, $\mathbf{T} = [T_x, T_y, T_z]^T$, and an angular velocity $\mathbf{W} = [W_x, W_y, W_z]^T$, the velocity of point $\mathbf{P}$ can be described as,

$$\frac{d\mathbf{P}}{dt} = -\mathbf{T} - \mathbf{W} \times \mathbf{P} .$$

Figure 5: The relations among various coordinate frames.

Taking the time derivatives of the above expression for x and y gives the motion of the point p in the image plane which is induced by the motion of the camera

$$\begin{aligned} \mathbf{v}_i &= \frac{d\mathbf{p}}{dt} = [\dot{x}, \dot{y}]^T = \mathbf{J}_c \mathbf{V}_c \\ \mathbf{J}_c &= \begin{bmatrix} -\frac{1}{Z_c} & 0 & \frac{x}{Z_c} & xy & -1-x^2 & y \\ 0 & -\frac{1}{Z_c} & \frac{y}{Z_c} & 1+y^2 & -xy & -x \end{bmatrix} \\ \mathbf{V}_c &= \begin{bmatrix} \mathbf{T} \\ \mathbf{W} \end{bmatrix}, \end{aligned}$$

where $\mathbf{J}_c$ is the Jacobian relating the camera velocities to the feature velocities in the image plane.

Next, by a coordinate transformation we can relate the camera velocities in $\{\mathbf{R}_c\}$ to $\{\mathbf{R}_o\}$,

$$\mathbf{V}_c = \mathbf{V}_o^c \mathbf{V}_o = \begin{bmatrix} \mathbf{R}_o^c & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_o^c \end{bmatrix} \mathbf{V}_o,$$

and

$$\mathbf{R}_o^c = \begin{bmatrix} \cos\theta\cos\phi & \sin\theta\cos\phi & -\sin\phi \\ -\sin\theta & \cos\theta & 0 \\ \cos\theta\sin\phi & \sin\theta\sin\phi & \cos\phi \end{bmatrix},$$

where $\mathbf{V}_o$ is the velocity vector in the world frame and $\mathbf{R}_o^c$ is the rotational matrix from the camera frame to the world frame. Next we relate $\mathbf{V}_o$ to $\mathbf{V}_s$ by differentiating the equations which relate spherical coordinates to Cartesian coordinates, or

$$\begin{aligned} X_o &= \rho\cos\theta\sin\phi \\ Y_o &= \rho\sin\theta\sin\phi \\ Z_o &= \rho\cos\phi, \end{aligned}$$

hence

$$\mathbf{T}_o = \begin{bmatrix} T_{x_o} \\ T_{y_o} \\ T_{z_o} \end{bmatrix} = \begin{bmatrix} \dot{X}_o \\ \dot{Y}_o \\ \dot{Z}_o \end{bmatrix} = \begin{bmatrix} -\rho\sin\theta\sin\phi\dot{\theta} + \rho\cos\theta\cos\phi\dot{\phi} + \cos\theta\sin\phi\dot{\rho} \\ \rho\cos\theta\sin\phi\dot{\theta} + \rho\sin\theta\cos\phi\dot{\phi} + \sin\theta\sin\phi\dot{\rho} \\ \rho\sin\phi\dot{\phi} + \cos\phi\dot{\rho} \end{bmatrix},$$

and

$$\mathbf{W}_o = \begin{bmatrix} W_{X_o} \\ W_{Y_o} \\ W_{Z_o} \end{bmatrix} = V_1 \dot{\theta} + V_2 \dot{\phi},$$

where $V_1 = [0, 0, 1]^T$ and $V_2 = [-\sin\theta, \cos\theta, 0]^T$ are vectors in the axis of rotation. So we have

$$\mathbf{V}_o = \begin{bmatrix} \mathbf{T}_o \\ \mathbf{W}_o \end{bmatrix} = \mathbf{J}_s \mathbf{V}_s,$$

and

$$\mathbf{J}_s = \begin{bmatrix} -\rho \sin\theta \sin\phi & \rho \cos\theta \cos\phi & \cos\theta \sin\phi \\ \rho \cos\theta \sin\phi & \rho \sin\theta \cos\phi & \sin\theta \sin\phi \\ 0 & -\rho \sin\phi & \cos\phi \\ 0 & -\sin\theta & 0 \\ 0 & \cos\theta & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

The final relationship is then,

$$\mathbf{v}_i = \mathbf{J}_c \mathbf{V}_c = \mathbf{J}_c \mathbf{V}_o^c \mathbf{V}_o = \mathbf{J}_c \mathbf{V}_o^c \mathbf{J}_s \mathbf{V}_s = \mathbf{J} \mathbf{V}_s,$$

where

$$\mathbf{J} = \mathbf{J}_c \mathbf{V}_o^c \mathbf{J}_s = \begin{bmatrix} -x y \sin\phi + y \cos\phi & -\frac{\rho}{Z_c} - (1 + x^2) & \frac{x}{Z_c} \\ -x \cos\phi - \sin\phi(1 + y^2) - \frac{\rho \sin\phi}{Z_c} & -xy & \frac{y}{Z_c} \end{bmatrix}.$$

To implement this in the visual servoing framework we must invert the $\mathbf{J}$ matrix. This matrix is of dimension 2×3 and therefore requires a pseudoinverse. However, the current interface to AESOP is through a foot or hand controller. These controllers use a joystick for the left/right and up/down motions and separate buttons for the in/out motions. For this reason we decided to limit the tracking motion to two degrees-of-freedom by eliminating the in/out motion and leaving that to the surgeon. The instrument tracking will allow the surgeon to guide the left/right and up/down motions of AESOP in response to the instrument motion while maintaining the zoom factor manually. The modification creates a $\mathbf{J}$ matrix defined by

$$\mathbf{J} = \begin{bmatrix} -x y \sin\phi + y \cos\phi & -\frac{\rho}{Z_c} - (1 + x^2) \\ -x \cos\phi - \sin\phi(1 + y^2) - \frac{\rho \sin\phi}{Z_c} & -xy \end{bmatrix},$$

which is a 2×2 matrix and can be easily inverted. The final block diagram for the visual servoing algorithm is shown in Figure 6, where (x_d, y_d) denotes the canonical feature location (e.g., at the center of an image frame), (x, y) the feature location reported from the image processing algorithm, and $(\delta x, \delta y)$ the error signal which is used to compute the robot control signal $(\delta\theta, \delta\phi, \delta\rho)$. The gain in this algorithm is used for robustness. Simulation results are presented in the next section.

4 Experimental Results

4.1 2D Image Analysis

We digitized several real laparoscopic sequences from video tapes. Because of the hardware limitation, we were able to digitize only 95 frames per sequence continuously at a top rate of 15 frames per second, which covered about 6 seconds at half the frame rate or 12 seconds at quarter the frame rate.

Two typical results showing various stages in image processing are shown in Figures 7 and 8. Figure 7.a shows a frame where a single instrument was visible. The instrument was fairly close to the scope with its tip much narrower than its shaft. Figure 8.a shows another frame where three instruments were within sight. Figures

Figure 6: Block diagram of the AESOP visual servoing algorithm.

Figure 7: (a) Original image (near field). One instrument is shown. (b) Binary image from color classification. Black and white pixels represent surgical instruments and organs, respectively. (c) Four directional median filters were used to suppress noise. (d) Labeled image. Each region is shown with a different intensity. (e) Computed bounding box.

Figure 8: (a) Original image (far field). Three instruments are shown. (b) Binary image from color classification. Black and white pixels represent surgical instruments and organs, respectively. (c) Four directional median filters were used to suppress noise. (d) Labeled image. Each region is shown with a different intensity. (e) Computed bounding box.

Figure 9: A sequence of 10 image frames showing the movement of a bounding box, these frames were sampled from 30 continuous images.

Figure 10: Instrument tip motion deduced from the bounding box over 30 consecutive images.

Table 1: Average error in instrument localization and tracking

sequence No.	number of frames	Mean Localization Error (pixel)	Mean Tracking Error (pixel)
1	29	2.5	3.8
2	30	4.5	6.3
3	25	10.6	6.3
4	22	5.0	4.1

7.b and 8.b show the results of pixel-by-pixel classification. Each pixel in the original images was classified as either an organ pixel (white) or an instrument pixel (black) using the pattern classifier discussed in Section 2.1.

Currently, the hardware used limited the total color resolution to eight bits per pixel, which were distributed evenly to red (3 bits), green (3 bits) and blue (2 bits) color channels. The classifier was trained to assign a unique class label (instrument or organ) to each one of the 256 color patterns. We estimated the misclassification rate to be less than 4%, which was arrived at by adding the number of instrument pixels misclassifying as organ and that of organ pixels misclassifying as instrument as a percentage of the total training samples. We then used directional median filters to clean the images and removed spurious noise points. The results are shown in Figures 7.c and 8.c.

Each localized instrument region then inherited a unique identifier through the temporal and spatial propagation processes described in Section 2.4. Different labels are represented as different gray levels in Figures 7.d and 8.d. Shape parameters were then computed to estimate the bounding box. Figures 7.e shows the computed bounding box in the near field case where a trapezoid was used. Figure 8.e shows several bounding boxes in the far field case where a rectangular shape was assumed.

Next step is to update the bounding boxes' locations over time to propagate the instrument labels. Figure 9 shows 10 sample images in a 30-image sequence. The bounding boxes are superimposed on the video images. Figure 10 displays the trajectories of the three bounding boxes in Figure 9 over time. Tips of the tracked instruments were computed from the bounding box approximation and displayed in Figure 10.

To estimate the error in automated localization and tracking, we manually located the tip of the tracked instrument from an image sequence and compared it with the tip location indicated by the bounding box. Furthermore, movement of the tip was computed using two different methods: a manual process in which we interactively picked the tip of the tracked instrument from each and every frame in a sequence and computed the movement, and an automated process where the image processing routine located the bounding box in each frame and tracked the bounding box movement over time. Using the localization and tracking results obtained manually as the ground truth, we computed the error in automated processing.

We defined the mean localization error as the average distance over a whole sequence between the instrument tips reported from the manual and automated operations. Similarly, mean tracking error was defined as the average difference over a whole sequence between the reported tip movements using the manual and automated operations. We computed the mean localization and tracking errors on four typical sequences and the results are summarized in Table 1. The four sequences were of 29, 30, 25, and 22 frames. The size of the image frames in these sequences was 160×150 , and the error was measured in terms of pixels. Sequences 1 and 2 depicted instruments in the near field, whereas sequences 3 and 4 depicted instruments in the far field. The error in localization and tracking is generally less than 5% of the image size, which is quite satisfactory in this type of applications.

Table 2: Camera and optics model parameters

focal length	3.37mm
field of view	80 degrees
image chip size	8mm (diagonal)
image size	400 × 400 pixels

4.2 Visual Servoing

A simulation was conducted to test the visual servoing algorithm with respect to model uncertainties and noise in the feature tracking feedback. The simulation was done using the block diagram shown in Figure 6. The robot block was modeled as a first order lag filter with a time constant of 0.0318 second. This was based on our knowledge of the current AESOP robot and its capabilities. The camera and optics model parameters are shown in Table 2 and are typical of those used with certain laparoscopes. The image feedback rate was assumed to be at 0.1 second which we believe is achievable with standard DSP hardware.

In our simulation, we defined a feature point in 3-space, calculated the projection onto the image plane, added random noise to this value, and then used this as the feedback for the tracking algorithm. All of these experimental results assumed that Z_s , or the depth of the object, was known.

In Figure 11, the results of centering a stationary feature point is shown. In Figure 11.a the path of the feature point in the image plane is shown with measured values of the feature point. The figure shows the feature point being positioned in the image plane at (0,0). The random noise in this simulation was generated from a Gaussian distribution of zero mean and a variance of 2% of the image plane size, or approximately 8 pixels. This was consistent with the image analysis experimental results.

Figure 11.b shows the change in the two spherical angles under control as the feature point was tracked. The convergence of these angles is clear. Finally, Figure 11.c shows the error between the desired feature position and the actual feature position. Again, the convergence of the error in the presence of noisy feedback is clear.

The ability to track the instrument while the instrument is moving is shown in Figure 12. In this experiment the instrument was moving in space at a rate of 20mm/sec. The feature moved first in the y-direction and then diagonally across the image plane. Figure 12.a shows the path of the instrument without any tracking. In Figure 12.b the path of the instrument during tracking is shown. In this figure the tracking error can be seen as the feature moved in one direction and then the next before finally coming to rest. Figures 12.c and 12.d show the change in the command angles with time and the error of the feature position respectively. These figures clearly show the ability of the algorithm to track an instrument in motion with noise in the image analysis.

5 The Concluding Remarks

The work we described in this paper is an integral part of a more ambitious project to develop robotic-enhanced technology (RET) [8]. The goal of RET is to create a multi-appendage robotic system which is controlled through a seamless and intuitive man-machine interface by the surgeon in order to enhance the overall safety and efficacy of the laparoscopic surgical procedure. The man-machine interface based on the RET concept is aimed at enabling the surgeon to simultaneously control and maneuver multiple units of AESOP *with ease and without additional use of the hands*. Automated image understanding facilitates scope positioning and frees the surgeon from the tedious task of manually controlling the visual feedback. It is a powerful mechanism in its ability to allow surgeons to seamlessly and intuitively control the AESOP robot.

Though currently we have tested the algorithm only with 0° scopes, the algorithm can be trivially generalized to work with scopes of different viewing angles. An additional rotation of the camera coordinate in Figure 5 will be

Figure 11: (a) Path of the feature point in the image plane. (b) Change in spherical coordinates of the laparoscope vs. time. (c) Error in feature location vs. time.

Figure 12: **(a)** Instrument motion in image plane without tracking. **(b)** Path of feature point being tracked in the image plane. **(c)** Change in spherical coordinates of laparoscope vs. time. **(d)** Error in feature location vs. time.

required. It is also possible to utilize this technique with other sensors, such as ultrasonic sensors. However, new image analysis algorithms and servoing goals need be formulated. Our future research plan includes integrating image processing with other interface modules, such as the voice recognition and feedback system currently under development. Even though our algorithm seems to be adequate for localizing and tracking instruments in laparoscopic images, the potential of automating image analysis to assist surgeons performing surgery is not fully realized. Hence, we will continue to enhance the image understanding module. We believe that, with a more sophisticated analysis, a great wealth of information can be extracted from video images to aid the laparoscopic surgery. Of particular interest to us is the wavelet multi-resolution image decomposition, which allows us to analyze laparoscopic images at multiple scales and orientations.

Acknowledgement: This project was supported in part by a State of California MICRO grant, 93-158.

References

- [1] Francois Chaumette, Patrick Rives, and Bernard Espiau, *Positioning of A Robot With Respect to An Object, Tracking It and Estimating Its Velocity by Visual Servoing*, IEEE International Conf. on Robotics and Automation, Sacramento, CA, Apr. 1991.
- [2] Richard O. Duda and Peter E. Hart, *Pattern Classification and Scene Analysis*, Wiley-Interscience, 1973.
- [3] John T. Feddema and C. S. George Lee, *Adaptive Image Feature Prediction and Control for Visual Tracking with A Hand-eye Coordinated Camera*, IEEE Transactions on Systems, man, and Cybernetics, vol. 20, no. 5, pp. 1172-1183, 1990.
- [4] B. K. P. Horn, *Robot Vision*, McGraw Hill, New York, 1986.
- [5] Jaroslav F. Hulka and Harry Reich, *Textbook of Laparoscopy, 2nd Ed.*, W. B. Saunders Company, Philadelphia, PA, 1994.
- [6] John G. Hunter and Jonathan M. Sackier, *Minimally Invasive Surgery*, McGraw Hill, New York, 1993.
- [7] N. N. Papanikolopoulos, P. K. Khosla, and T. Kanade, *Visual Tracking of A Moving Target by a Camera Mounted on a Robot: A Combination of Control and Vision*, IEEE Transactions on Robotics and Automation, vol.9, no.1, Feb. 1993.
- [8] Yulun Wang, *Introducing Robotic Enhancement's New Applications*, Technical Note #1, Computer Motion Inc. 1993.
- [9] Yulun Wang, *AESOP: Automated Endoscope for Optimal Positioning*, Technical Note #2, Computer Motion Inc., 1993
- [10] Lee E. Weiss, Arthur C. Sanderson, and Charles P. Neuman, *Dynamic Sensor Based Control of Robots with Visual Feedback*, IEEE Journal of Robotics and Automation, vol. RA-3, no. 5, pp. 404-417, 1987.