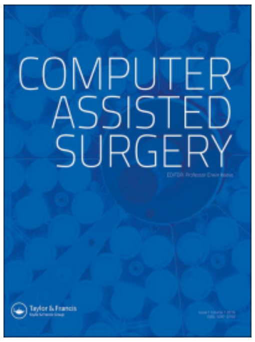


Taylor & Francis
Taylor & Francis Group

Journal of Image Guided Surgery

ISSN: 1078-7844 (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/icsu19

Laboratory Investigation:Automated Instrument Tracking in Robotically Assisted Laparoscopic Surgery

Darrin R. Uecker, Y. F. Wang, Cheolwhan Lee & Yulun Wang

To cite this article: Darrin R. Uecker, Y. F. Wang, Cheolwhan Lee & Yulun Wang (1995) Laboratory Investigation:Automated Instrument Tracking in Robotically Assisted Laparoscopic Surgery, Journal of Image Guided Surgery, 1:6, 308-325, DOI: 10.3109/10929089509106338

To link to this article: https://doi.org/10.3109/10929089509106338

Laboratory Investigation

Automated Instrument Tracking in Robotically Assisted Laparoscopic Surgery

Darrin R. Uecker, Cheolwhan Lee, Y.F. Wang, and Yulun Wang
Computer Motion, Inc., University Business Center, Goleta, California (D.R.U., Y.W.); Department of Computer Science, University of California, Santa Barbara, California (C.L., Y.F.W.)

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 that utilizes multiple small incisions on the patient's body through which the surgeon inserts tools and a videoscope in order to conduct 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 rather than performing the traditional "open" procedure, where a large incision is made on the patient's body for direct viewing.

The current mode of laparoscopy employs an assistant to hold the scope and position 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. We have developed a robotic laparoscope positioner to replace the assistant. The surgeon commands the robotic positioner through a hand/foot controller interface. To further simplify the human-machine interface that controls 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 visual feedback and to perform surgery more efficiently without requiring additional use of the hands. *J Image Guid Surg 1:308–325 (1995).* ©1996 Wiley-Liss, Inc.

Key words: laparoscopy, endoscopy, image analysis, robotics, tracking

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 *may* reduce the trauma inflicted on the patient during surgery, significantly shorten the time for the patient to recuperate, and lower the cost of the treatment. Because of the benefits gained over traditional surgical procedures, minimally invasive surgery is fast gaining popularity.

A key technological advance that has fueled the minimally invasive revolution is video laparoscopy.[5] Laparoscopic procedures are minimally invasive surgical procedures in which several small incisions are made on the patient to accommodate surgical instruments, such as scalpels, scissors, and staple guns. A video laparoscope is

Received original January 10, 1996; accepted June 7, 1996.

Address correspondence/reprint requests to Professor Y.F. Wang, Department of Computer Science, University of California, Santa Barbara, CA 93106.

inserted through the navel to acquire video images of the body cavity that are displayed in real time on a monitor, providing visual feedback to the surgeon. This setup enables the surgeon to operate instruments through the small incisions rather than using the traditional large incision for direct viewing and operation.

This change to a video-based operating paradigm permits the introduction of innovative techniques to assist the surgeon. Because the surgeon has become comfortable operating through a video interface, it is now possible to provide computer assistance with added functionality.

Usually, an assistant holds the laparoscope for the surgeon and positions the scope in response to the verbal directions from the surgeon (Fig. 1a). This 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. This represents a waste of personnel and can pose a risk to the patient.

To improve on the current mode of laparoscopic surgery, an automated scope-positioning system, Automated Endoscope System for Optimal Positioning (AESOP); Computer Motion Inc., Goleta, CA[9] has been developed. The robot holds the scope and responds to positioning commands issued by the surgeon, who uses a hand/foot controller (Fig. 1b). This mode of operation improves the visual feedback to the surgeon by giving the surgeon direct control of visual feedback and by eliminating the assistant from the loop. Thus, the procedure can be performed faster and with greater ease.

We have employed this system as a test bed for developing image analysis and tracking modalities to facilitate laparoscopic surgery. A goal of the project is to develop novel "handless" paradigms for improving the ease with which the surgeon can control visual feedback by automating the scope positioning and aiming mechanism and by eliminating the hand/foot controller.

The first steps in achieving automated scope positioning are 1) implementing practical, real-time image analysis algorithms for locating and tracking surgical instruments in laparoscopic image sequences, and 2) using 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 large numbers of image sequences that were taken during real laparoscopic surgery to serve as our test data. To accomplish the first step, we have obtained a large sample of color signatures from over 100,000 instrument and organ pixels in these sequences and 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 step, we have used the two-dimensional 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 that 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 measurements (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 automatically for the deviation.

The remainder of this paper is organized as follows: First, we discuss the image analysis algorithm for instrument localization and tracking from an laparoscopic image sequence. Next, we present the mathematic framework for the AESOP robot to maneuver and aim the scope based on two-dimensional image measurements. Experimental results based on both real image sequences and computer simulations are then described.

METHODS

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, a unique label is assigned to each instrument region, and new labels are generated for instruments just ap-

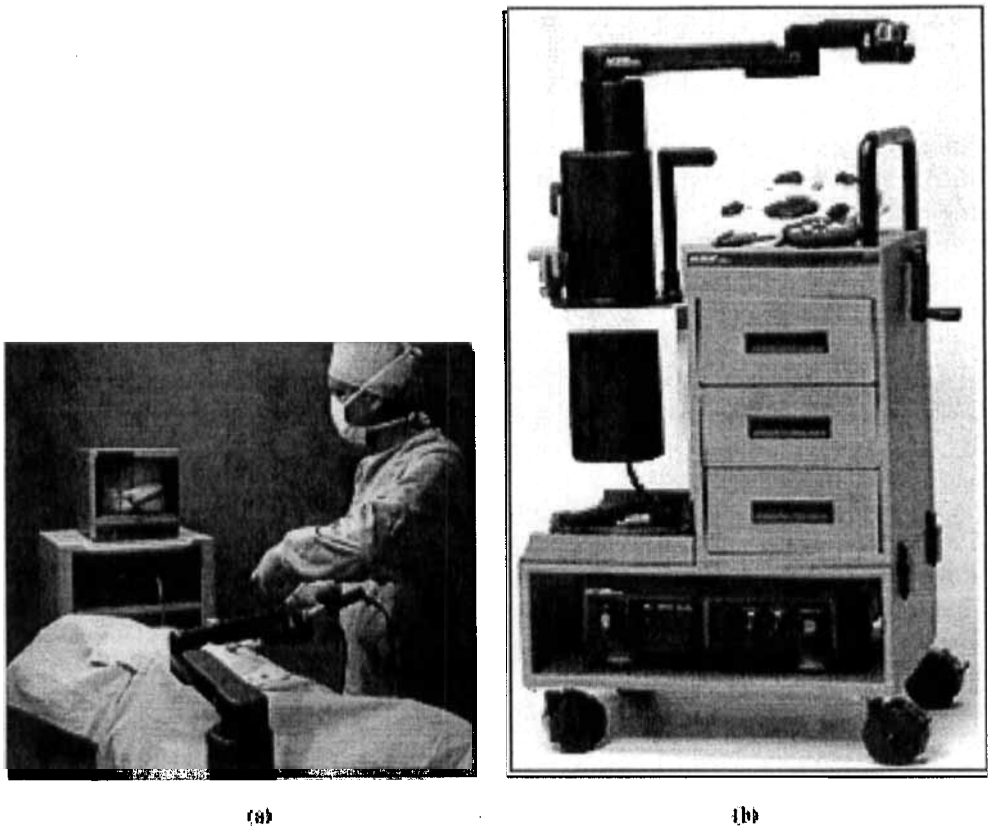


Fig. 1. **a:** Traditional laparoscopy performed by a surgeon and a scope assistant. **b:** Robotically assisted laparoscopy in which a robot replaces the scope assistant.

pearing 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 is used to predict its location in the next frame in order to propagate the instrument labels over time). We describe these four steps in more detail below.

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 in order 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 by 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] 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's relative frequency of occurrence. Furthermore, a conditional probability, $p(\mathbf{x} \mid \omega_i)$, 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:

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

where

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

The class assignment which attains the largest $p(\omega_i \mid \mathbf{x})$ 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 three-dimensional measure vector $\mathbf{x} = (R, G, B)$, where the components represent the red, green, and blue color intensities. Furthermore, we assumed that $p(\bullet \mid \omega_i)$, $i = 1,2$ was a multivariate normal distribution with mean μ_i and variance Σ_i, and we estimated that

$$\mu_i = \sum_{j=1}^{N_i} \mathbf{x}_{ij} \Big/ N_i,$$

$$\sum_i = \sum_{j=1}^{N_i} (\mathbf{x}_{ij} - \mu_i)(\mathbf{x}_{ij} - \mu_i)^T \Big/ (N_i - 1),$$

where $\mathbf{x}_{ij}$ represents the measure vector of samples in class i, and N_i represents 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

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

Some sample classified images are shown in Figures 7b and 8b in section 4. To remove spurious noise in these images, we use four directional (vertical: 5 × 1; horizontal: 1 × 5, 45° 5 × 5, and −45° 5 × 5) median filters to filter the images. Some typical outputs from median filtering are shown in Figures 7c and 8c.

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.

We predict the locations of instrument regions in the current frame by using their positions in the previous frame and by using the anticipated movement (discussed below). The anticipated motion of an instrument is computed as a weighted average of the instrument motion over a few previous frames. Instrument pixels that are detected in the current frame and whose positions confirm the predicted instrument locations assume the labels of the instrument pixels of the previous frame.

Because we anticipate error in this simple motion-prediction scheme, we go through another iteration in which 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 ten pixels per frame. Thus, 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.

Shape Analysis

Each localized instrument region is fitted with a bounding box. The needs for a bounding box approximation are as follows: 1) Due to uneven lighting and specular reflection, a localized instrument region may assume a somewhat irregular shape (see, e.g., Fig. 8c, the instrument region at the lower left corner). However, it is known that the shaft of an instrument is cylindrical in 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. 3) Goodness in the bounding box approximation can be used to identify instruments that are just entering the field of view, and it can also be used to discard large dark areas in images that have a color signature similar to that of an instrument region but that 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, then the region is classified as an instrument region, where M_{20} and M_{02} denote the 2nd-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 $2\sqrt{M_{02}/(M_{00} - 1)}$ (Fig. 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. If w and h denote the lengths of the base and height, respectively, of such a trapezoid, then one can easily show that

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

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 $M_{00}^{3/2}/(3\sqrt{M_{20}})$ (Fig. 2).

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

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

where K is the number of image frames used in predicting the instrument movement, and $\Sigma_{k=0}^{K-1} w(k) = 1$. 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.

Specular Reflection

One implementation detail that is worth mentioning is how specular reflection is handled. At times, laparoscopic images 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 (see Classification) to classify specular spots. That is, pixels that assume 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 they are spatially adjacent to that region.

Time Complexity

We give an estimate of the rate of image analysis in terms of the floating-point and integer operations needed. Denote T_I as the time to execute an integer operation, and denote T_F as 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.

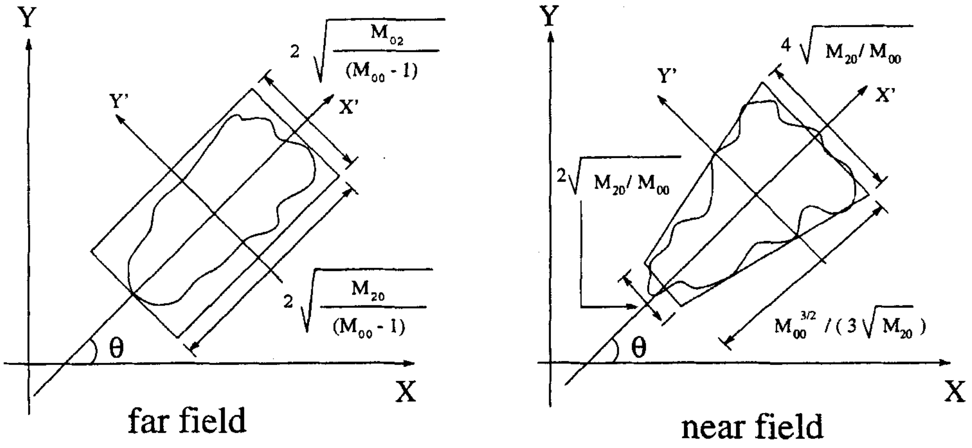

Y
2 $\sqrt{\frac{M_{02}}{(M_{00}-1)}}$
Y'
X'
θ
2 $\sqrt{\frac{M_{20}}{(M_{00}-1)}}$
X
far field
Y
4 $\sqrt{M_{20}/M_{00}}$
Y'
X'
2$\sqrt{M_{20}/M_{00}}$
θ
$M_{00}^{3/2}/(3\sqrt{M_{20}})$
X
near field

Fig. 2. Computation of the bounding box.

In the classification stage, the pattern classifier is trained off line, and results are stored in a table. Thus, classification based on color signature uses a table look-up process, and no floating-point operation is needed. Processing time for pixel classification, then, is $2N_pT_I$ or 1 indexing and 1 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, 5 fetch and add operations ($10N_pT_I$), 1 comparison operation (N_pT_I), and 1 assignment operation (N_pT_I) are needed. For spatial relaxation, a maximum of 9 operations ($9KN_pT_I$) are needed, where K is the number of iterations in the spatial relaxation process. The constant 9 comes from 1 comparison operation for each of the 8 neighbors plus 1 more class assignment. Another 3 operations ($3N_pT_I$) 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 2 comparisons to assert that pixels in the current frame and in the temporal template are both of the instrument type plus 1 possible assignment operation.

To process instrument pixels that are left unlabeled after temporal and spatial propagation, 18 operations ($18N_pT_I$) are needed, where 1 comparison is for locating an unlabeled pixel, 1 comparison operation and 1 assignment operation per neighbor are for the 8 neighbors to find a minimum label, and 1 assignment is for storing the chosen label. Computing the moments of an instrument region has a complexity of $4N_pT_I$, where 1 logic operation is used to ascertain the identity of a pixel, and 3 additions are for computing the area and centroid. Furthermore, 10 operations ($10N_pT_I$) are used to compute the principal axis directions. One notes that these operations require scanning the images pixel by pixel, and they can be time consuming. However, one also notes that, because of the regularity of these algorithms, they can be parallelized easily to achieve a significant increase in speed.

Computing the bounding box dimension and location and computing the instrument motion are constant time operations that are independent of the size of the processing window. We estimate the complexity to be $47N_pT_F$. Finally, a maximum of $85N_pT_I + 47N_pT_F + 16N_iN_sT_I$ 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 ($10N_pT_I + 47N_pT_F$), and merging into a nearby instrument region ($16N_iN_sT_I$).

Therefore, the total computation time is $(170 + 9K)N_pT_I + 94N_pT_F + 16N_iN_sT_I$. If $N_i = N_s = 1$ and $K = 10$, then approximately 12,000 pixels can be processed by our algorithm every 1/30 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

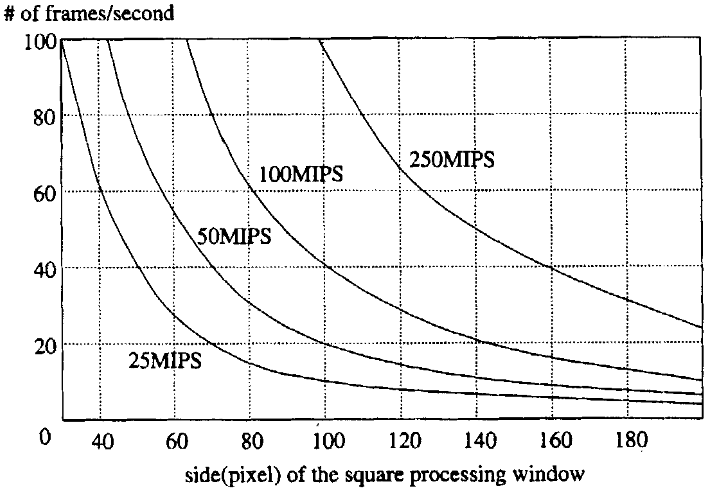


Fig. 3. Expected performance of the image-analysis algorithm.

allows us to select the processing window size that maintains the desired processing speed.

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 human-machine interface between the surgeon and the AESOP system.

Visual servoing of a robot manipulator has been studied by a number of researchers.[1,3,7,10] Visual servoing, as applied to robotic control of a 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 three-dimensional space, i.e., any position and orientation (pose).[7] However, the physical constraint imposed on the scope and, hence, on the camera position presents new problems for this application. In laparoscopic surgery, the camera is typically mounted on 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. Therefore, we represent the motion of the laparoscope and, thus, 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. This particular configuration is for 0° scopes. The pivot point allows three degrees of freedom (θ,ϕ,ρ) for manipulating the camera. This coordinate system makes sense from the point of view of application, because it defines the typical camera mo-

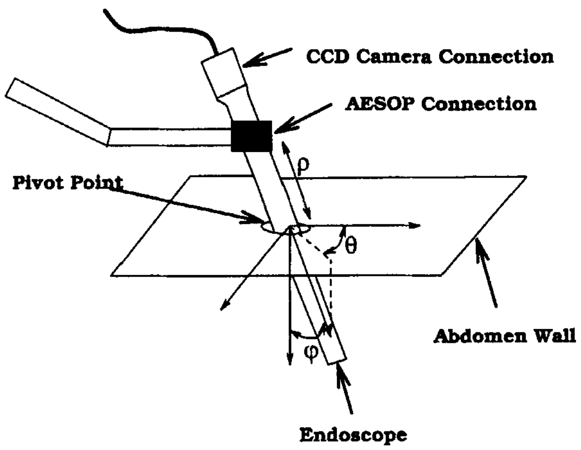


Fig. 4. Sliding constraint imposed by the pivot point on the laparoscope.

tions: 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, **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 **J** is as follows: We consider a target that is defined by an image feature located at a point, **P**, in three-dimensional space. Our formulation uses a pinhole camera model that 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 **P** with coordinates $[X_c, Y_c, Z_c]^T$ in reference frame $\{\mathbf{R}_c\}$ projects onto a point **p** in the image plane with image coordinates $[x, y]^T$ given by

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

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 **P** can be described as

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

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:

$$\mathbf{v}_i = \frac{d\mathbf{p}}{dt} = [\dot{x}, \dot{y}]^T = \mathbf{J}_c\mathbf{V}_c$$

$$\mathbf{J}_c = \begin{bmatrix} -\frac{1}{Z_c} & 0 & \frac{x}{Z_c} & xy & -1-x^2 & y \\ 0 & -\frac{1}{Z_c} & \frac{y}{Z_c} & 1+y^2 & -xy & -x \end{bmatrix}$$

$$\mathbf{V}_c = \begin{bmatrix} \mathbf{T} \\ \mathbf{W} \end{bmatrix},$$

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 $\{\mathbf{R}_o\}$:

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

and

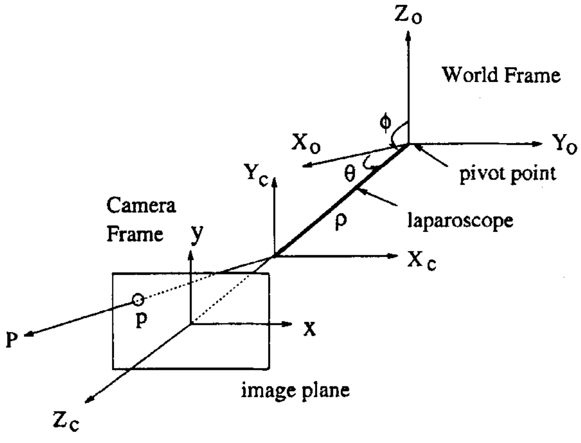


Fig. 5. The relations among various coordinate frames.

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

where $\mathbf{V}_0$ is the velocity vector in the world frame, and $\mathbf{R}_{\mathrm{o}}^{\mathrm{c}}$ is the rotational matrix from the camera frame to the world frame. Next, we relate $\mathbf{V}_{\mathrm{o}}$ to $\mathbf{V}_{\mathrm{s}}$ by differentiating the equations that relate spherical coordinates to Cartesian coordinates, or

$$X_o = \rho\cos\theta\sin\phi$$

$$Y_o = \rho\sin\theta\sin\phi$$

$$Z_o = \rho\cos\phi.$$

Hence

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

and

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

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

$$\mathbf{V}_{\mathrm{o}} = \begin{bmatrix} \mathbf{T}_{\mathrm{o}} \\ \mathbf{W}_{\mathrm{o}} \end{bmatrix} = \mathbf{J}_{\mathrm{s}}\mathbf{V}_{\mathrm{s}}$$

and

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

The final relationship is then

$$\mathbf{v}_{\mathrm{i}} = \mathbf{J}_{\mathrm{c}}\mathbf{V}_{\mathrm{c}} = \mathbf{J}_{\mathrm{c}}\mathbf{V}_{\mathrm{o}}^{\mathrm{c}}\mathbf{V}_{\mathrm{o}} = \mathbf{J}_{\mathrm{c}}\mathbf{V}_{\mathrm{o}}^{\mathrm{c}}\mathbf{J}_{\mathrm{s}}\mathbf{V}_{\mathrm{s}} = \mathbf{J}\mathbf{V}_{\mathrm{s}},$$

where

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

To implement this in the visual servoing framework we must invert the **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 joy stick for the left/right and up/down motions and use separate buttons for the in/out motions. For this reason, we decided to limit the tracking motion to 2 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 **J** matrix that is defined by

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

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) denotes the feature location reported from the image processing algorithm, and $(\delta x, \delta y)$ denotes the error signal that 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.

RESULTS

Two-Dimensional 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 sec, which covered about 6 sec at one-half the frame rate or 12 sec at one-quarter the frame rate.

Two typical results showing various stages

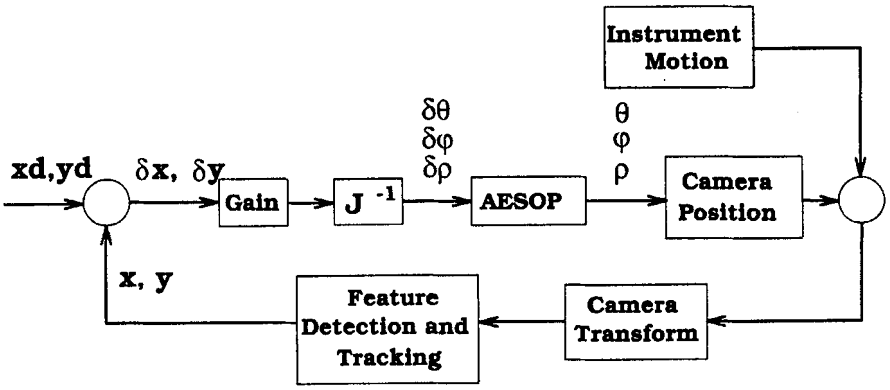
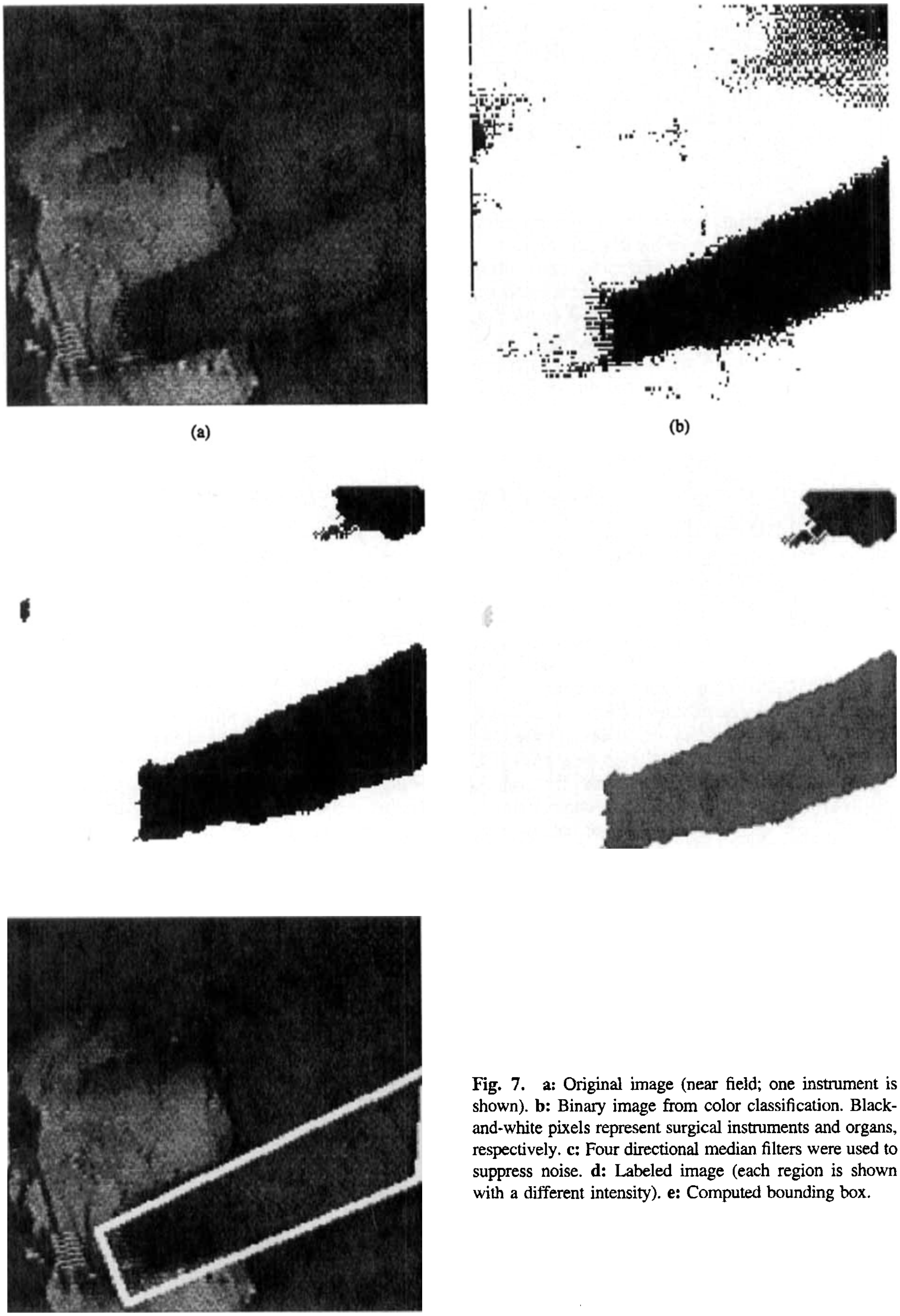


Fig. 6. Block diagram of the Automated Endoscope System for Optimal Positioning (AESOP) visual servoing algorithm.



(a)

(b)

Fig. 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.

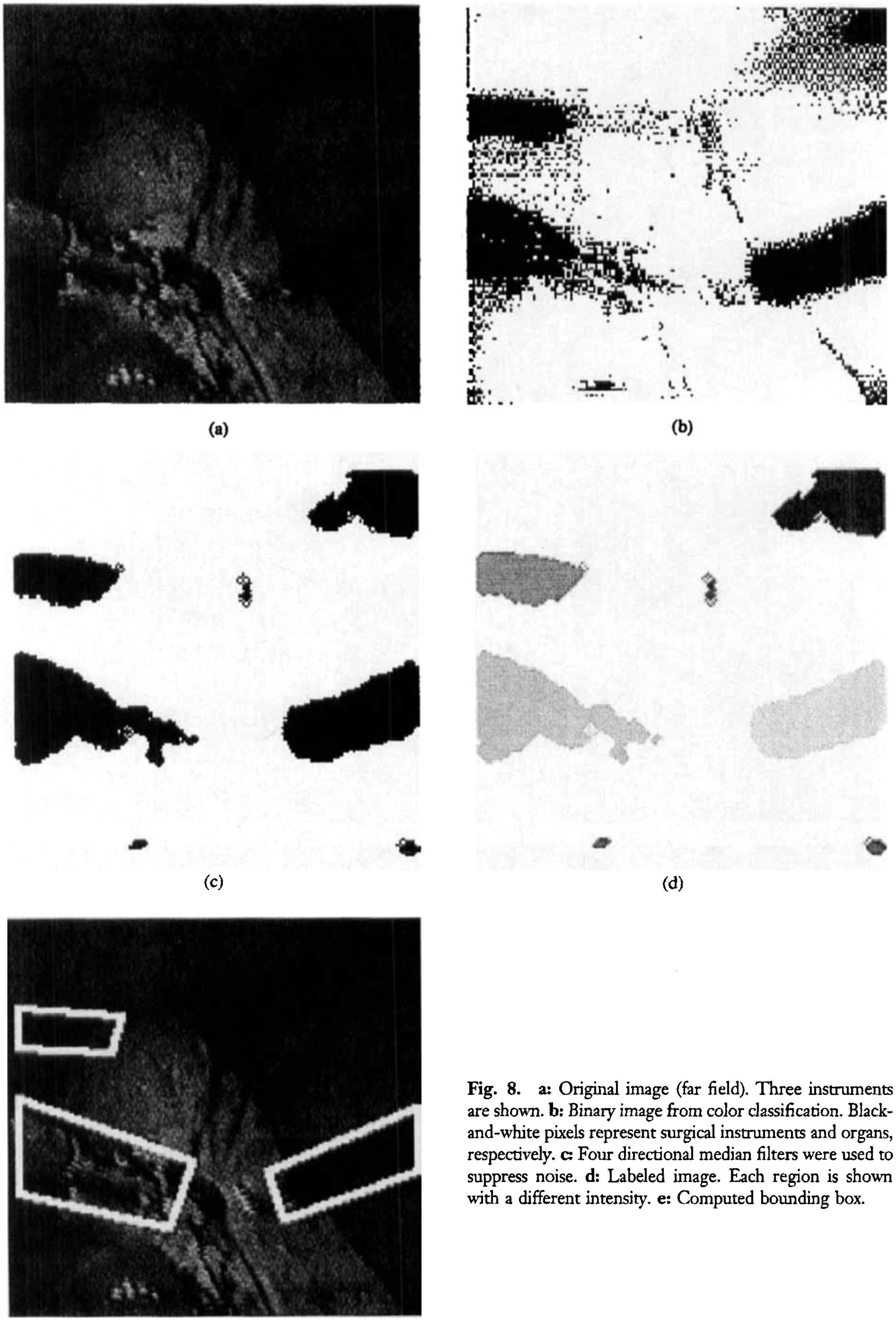


Fig. 8. **a:** Original image (far field). Three instruments are shown. **b:** Binary image from color classification. Black-and-white pixels represent surgical instruments and organs, respectively. **c:** Four directional median filters were used to suppress noise. **d:** Labeled image. Each region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

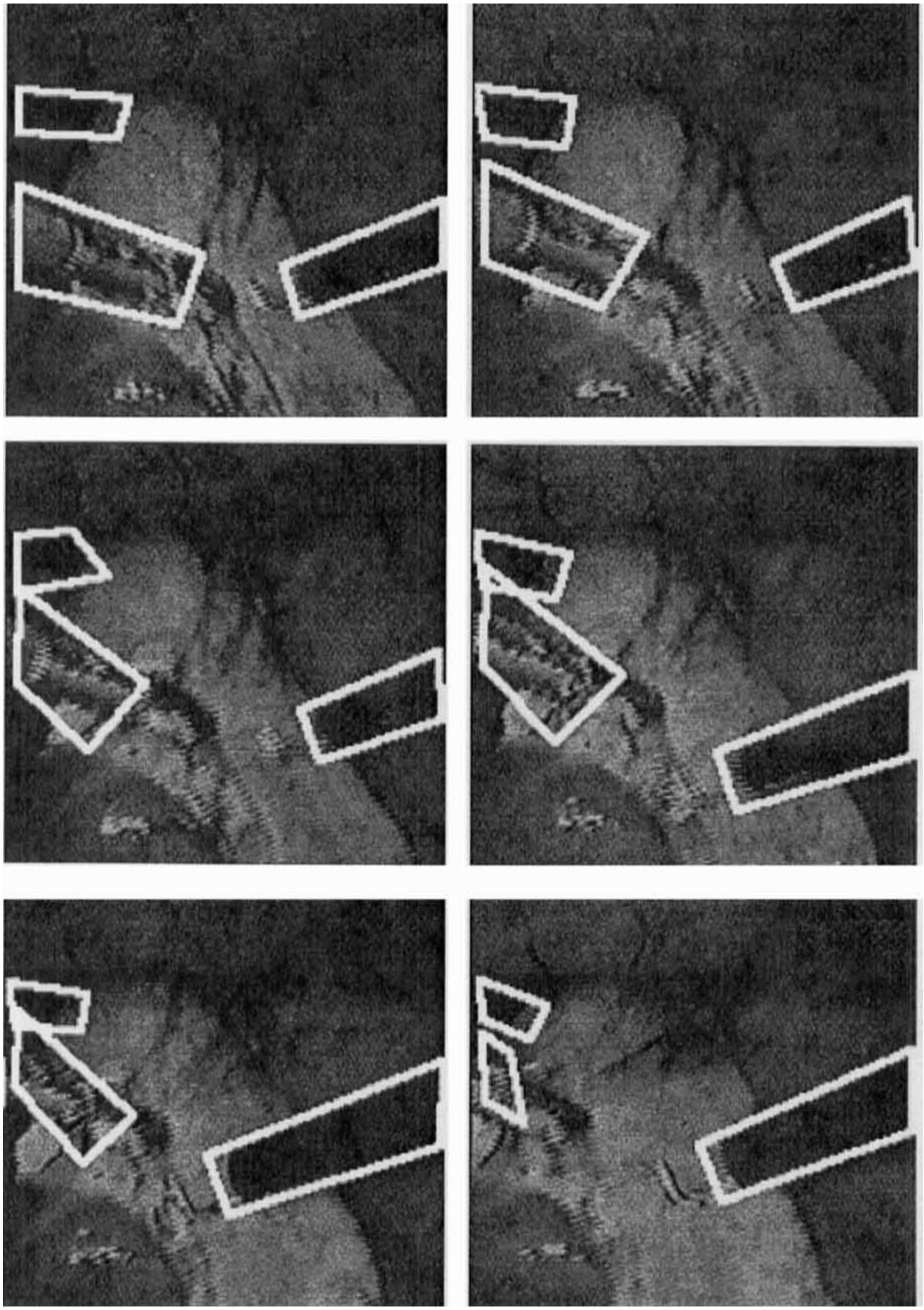


Fig. 9. A sequence of ten image frames showing the movement of a bounding box. These frames were sampled from 30 continuous images.

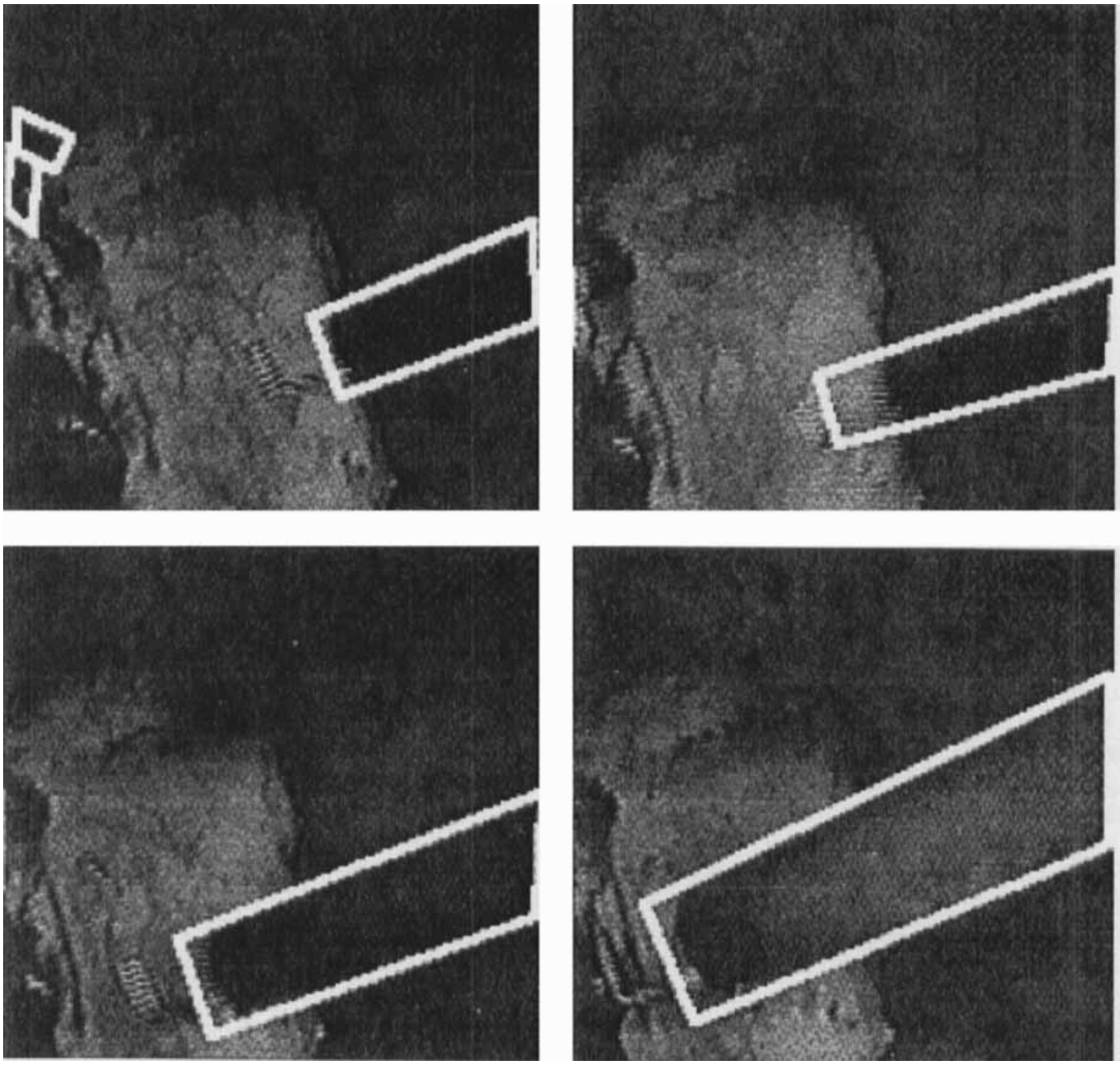


Fig. 9. *(Continued)*

in image processing are shown in Figures 7 and 8. Figure 7a shows a frame in which a single instrument was visible. The instrument was fairly close to the scope, and its tip was much narrower than its shaft. Figure 8a shows another frame in which three instruments were within sight. Figures 7b and 8b 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) by using the pattern classifier discussed above.

The hardware we used limited the total color resolution to eight bits per pixel, which were distributed evenly to red (three bits), green (three bits), and blue (two 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 that the misclassification rate was less than 4%, which was arrived at by adding the number of instrument pixels that were misclassified as organ and the number of organ pixels that were misclassified as instrument as a percentage of the total training samples. We then used directional median filters to clean the images, and we removed spurious noise points. The results are shown in Figures 7c and 8c.

Each localized instrument region then inherited a unique identifier through the temporal and spatial propagation processes described above. Different labels are represented as different gray levels in Figures 7d and 8d. Shape parameters were then computed to estimate the bounding

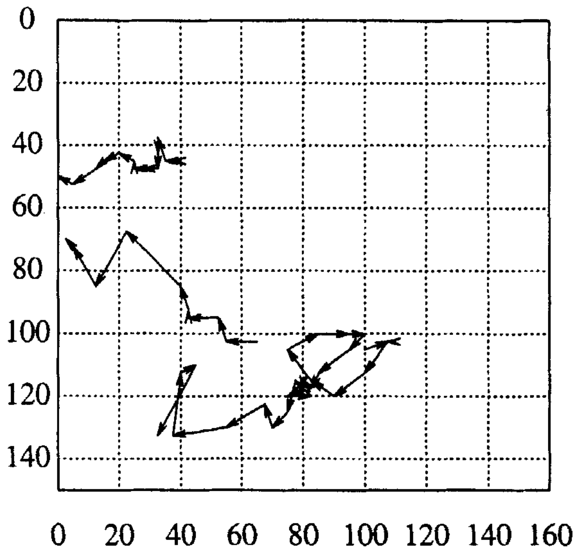


Fig. 10. Instrument tip motion deduced from the bounding box over 30 consecutive images.

box. Figure 7e shows the computed bounding box in the near-field case where a trapezoid was used. Figure 8e shows several bounding boxes in the far-field case where a rectangular shape was assumed.

The next step is to update the location of the bounding boxes over time to propagate the instrument labels. Figure 9 shows ten 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 shown in Figure 9 over time. Tips of the tracked instruments were computed from the bounding box approximation and are displayed in Figure 10.

To estimate the error in automated localization and tracking, we located the tip of the tracked instrument manually from an image sequence and compared it with the tip location that was 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 in which the image processing routine located the bounding box in each frame and tracked the bounding box movement over time. By 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 application.

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 by 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 sec. 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 sec, which we believe is achievable with standard DSP hardware.

In our simulation, we defined a feature point in three-dimensional 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, or the depth of the object, was known.

In Figure 11, the results of centering a stationary feature point is shown. In Figure 11a, the path of the feature point in the image plane is

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	20	4.5	6.3
3	25	10.6	6.3
4	22	5.0	4.1

shown along 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 eight pixels. This was consistent with the experimental image analysis results.

Figure 11b 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 11c 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 20 mm/sec. The feature moved first in the y direction and then diagonally across the image plane. Figure 12a shows the path of the instrument without any tracking. In Figure 12b, 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 it finally came to rest. Figure 12c,d shows the change in the command angles with time and the error of the feature position, respectively. This figure shows clearly the ability of the algorithm to track an instrument in motion with noise in the image analysis.

CONCLUSIONS

The work 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 multiappendage robotic system that is controlled through a seamless and intuitive human-machine interface by the surgeon in order to enhance the overall safety and efficacy of the laparoscopic surgical procedure. The human-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. Understanding the automated image facilitates scope positioning and frees the surgeon from the tedious task of manually controlling the visual feedback. It is a powerful mechanism because of its ability to allow surgeons to control the AESOP robot seamlessly and intuitively.

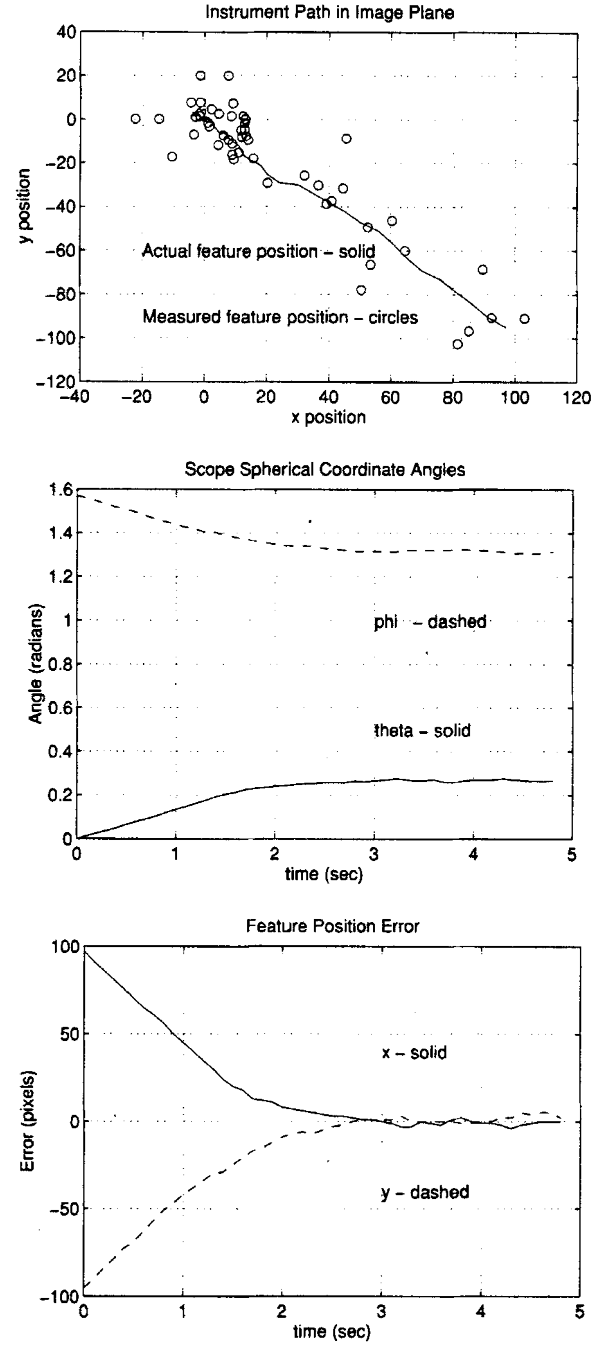


Fig. 11. **a:** Path of the feature point in the image plane. **b:** Change in spherical coordinates of the laparoscope vs. time. **c:** Error in feature location vs. time.

Table 2. Camera and Optics Model Parameters

Focal length	3.37 mm
Field of view	80 degrees
Image chip size	8 mm (diagonal)
Image size	400 × 400 pixels

Currently, although we have tested the algo-

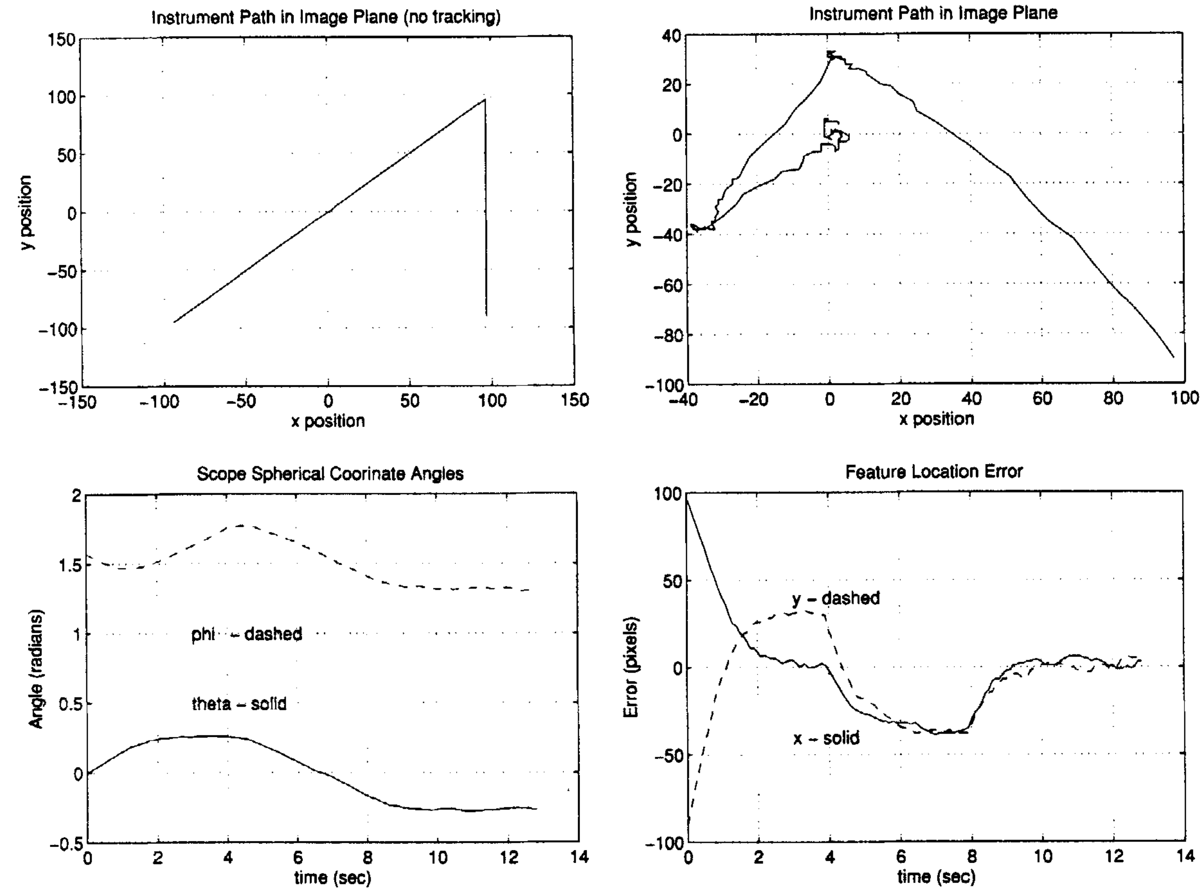


Fig. 12. **a**: Instrument motion in image plane without tracking. **b**: Path of feature point being tracked in the image plane. **c**: Change in spherical coordinates of laparoscope vs. time. **d**: Error in feature location vs. time.

rithm only with 0° scopes, the algorithm can be trivially generalized to work with scopes of different viewing angles (with an additional rotation of the camera coordinate in Fig. 5). 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 automated 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 in laparoscopic surgery. Of particular interest to us is the wavelet multiresolution image decomposition, which allows us to analyze laparoscopic images at multiple scales and orientations.

ACKNOWLEDGMENTS

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

REFERENCES

1. Chaumette F, Rives P, Espiau B (1991) Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing. Paper presented at the IEEE International Conference on Robotics and Automation, Sacramento, CA, April, 1991.
2. Duda RO, Hart PE (1973) "Pattern Classification and Scene Analysis." New York: Wiley-Interscience.
3. Feddema JT, Lee CSG (1990) Adaptive image feature prediction and control for visual tracking with a hand-eye coordinated camera. IEEE Transact Syst Man Cybernet 20(5):1172–1183.
4. Horn BKP (1986) "Robot Vision." New York: McGraw Hill.

5. Hulka JF, Reich H (1994) "Textbook of Laparoscopy, 2nd ed." Philadelphia: W.B. Saunders Co.
6. Hunter JG, Sackier JM (1993) "Minimally Invasive Surgery." New York: McGraw Hill.
7. Papanikolopoulos NN, Khosla PK, Kanade T (1993) Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. IEEE Transact Robot Automat, Vol 9:14–35.
8. Wang Y (1993) Introducing robotic enhancement's new applications, technical note no. 1. Computer Motion, Inc.
9. Wang Y (1993) AESOP: Automated endoscope for optimal positioning, technical note no. 2. Computer Motion, Inc.
10. Weiss LE, Sanderson AC, Neuman CP (1987) Dynamic sensor based control of robots with visual feedback. IEEE J Robot Automat 3:404–417.