Detection of Thin Lines using Low-Quality Video from Low-Altitude Aircraft in Urban Settings

JOSHUA CANDAMO
RANGACHAR KASTURI, Fellow, IEEE
DMITRY GOLDGOF, Fellow, IEEE
SUDEEP SARKAR, Senior Member, IEEE
University of South Florida

A novel thin line detection algorithm for use in low-altitude aerial vehicles is presented. This algorithm is able to detect thin obstacles such as cables, power lines, and wires. The system is intended to be used during urban search and rescue operations, capable of dealing with low-quality images, robust to image clutter, bad weather, and sensor artifacts. The detection process uses motion estimation at the pixel level, combined with edge detection, followed by a windowed Hough transform. The evidence of lines is tracked over time in the resulting parameter spaces using a dynamic line movement model. The algorithm's receiver operating characteristic curve (ROC) is shown, based on a multi-site dataset with 86 videos with 10160 wires spanning in 5576 frames.

Manuscript received October 16, 2006; revised June 13, 2007; released for publication April 5, 2008.

IEEE Log No. T-AES/45/3/933950.

Refereeing of this contribution was handled by L. Kaplan.

This research was supported in part by the Safety Security Rescue-Research Center at the University of South Florida which is a National Science Foundation supported Industry/University Cooperative Research Center.

Authors' address: Dept. of Computer Science and Engineering, University of South Florida, 4202 E. Fowler Ave., ENB 118, Tampa, FL 33620-5399, E-mail: (candamo@cse.usf.edu).

I. INTRODUCTION

The United States Army reports that they have lost more helicopters to power lines than against enemies in combat [1]; most importantly, collisions usually occur in daylight. Helicopters and small aircraft pilots flying at low altitude fail to identify thin objects such as cables, power lines, and wires because they are not easy to perceive over heavily cluttered backgrounds, or when the contrast between the object and the background is low. This work focuses on low-altitude flight in urban search and rescue (USAR) missions, for which often there are no constrained or controlled conditions of weather, lighting effects, sensor noise, and scene complexity.

The thin line detection algorithm and the results presented are novel. The algorithm introduces a tracking model for thin lines that uses the parameter space output of a windowed Hough transform to track line motion through time. This is the first work that uses a large and challenging multi-site dataset, or has dealt with thin objects in heavily cluttered videos in urban settings, or has tried to use a visual complexity cue as a measure to dynamically train and select parameters used in automatic object detection. The high level of difficulty of the dataset (see Fig. 1) was chosen in order to use data that correspond to representative examples of different urban scenarios, which is especially important in the military combat and USAR domain.

In recent years, significant effort has been extended towards the development of visual navigation systems for autonomous aircraft and micro air vehicles [2–5]. Small aircraft have significant limitations in payload and electrical power that can be supported, which impose severe restrictions on the hardware that can be used on-board. Computer vision has been an effective and popular tool due to the low size, weight, and power consumption of small vision-based sensors. The reader is referred to [6] for an overview of the challenges faced in visual navigation of small autonomous aircraft in urban environments. Algorithms for airborne automatic object detection using computer vision techniques are abundant. In broad terms, there are spatial-based and temporal-based methods for automatic object detection. Among the most common spatial techniques are Gabor filters [7], and morphological filtering [8]. Among the temporal-based there are dynamic programming [9–11], and maximum likelihood [12]. In general, previous work in automatic object detection has focused in detection of thick targets. In this paper the focus is automatic target detection of thin objects, which potentially represent harder objects to discern by human pilots, or to detect by automated systems. Kasturi, et al. [13] used a line detection algorithm combined with a Hough transform to detect wires in the path of low-altitude rotorcraft vehicles

Fig. 1. Challenging sample frames from dataset.

with subpixel accuracy in still images. Single-frame image processing and traditional Hough transform are shown in this paper to be unsuitable to cope with heavily cluttered urban scenes. In this work, temporal information is exploited through the use of motion estimation and object tracking. Motion estimation, using optic flow, has been used as the backbone of autonomous helicopter navigation systems for urban environments [14]. Also, temporal tracking of objects is performed in a multi-window Hough space. Hills, et al. [15] demonstrates tracking of rectangular structures in video, using the parameter space created by a Hough Transform. Tracking the peak of the parameter space takes advantage of the Hough transform's robustness against feature occlusion and illumination changes.

The focus of this work, in terms of the data domain, is on urban scenery using low-quality video. In general, published algorithms work well when the background is fairly simple, or uniform. However, in USAR missions, reliable methods should be insensitive to data uncertainty due to sensor artifacts, weather conditions, and cluttered backgrounds. Gandhi, et al. [16] and Mau-Tsuen, et al. [17] used cluttered backgrounds in images from high-altitude video; nevertheless, the fact that low-altitude flight is close to the ground places more severe requirements on the algorithms to be used; especially a major constrain is to deal with urban settings, where buildings, trees, power lines, and thin wires are common objects in the flight path of the aircraft.

Vision technology is not designed to be a replacement of radar and other sensors typically used in object detection for aircraft navigation, but as an additional mean to improve the detection reliability under stressed conditions. Among the sensor-based wire detection strategies are infrared technology and laser systems [18]. High-resolution infrared cameras have been used [19] as image sensor systems to aid pilots, but often the target-to-background contrast is not enough for automatic target detection, and results are too dependant on image processing algorithms that could potentially cause thin lines to disappear. Laser systems are highly sensitive to atmospheric attenuation effects [20], limiting their reliability to fairly good weather conditions. Due to the size and power constraints of small aerial vehicles, nonvision-based sensor technology is currently not viable as a standalone system. Recent years have shown that adapting micro-technology, such as miniature radars [21, 22], although limited, opens new and promising possibilities for the future of USAR aircraft navigation. It is important to recognize that disaster areas are likely to have power outages, smoke, and fire, which affects considerably infrared-based techniques; consequently, even for multi-modal systems, it is essential to have a reliable computer vision algorithm capable of functioning under uncertain conditions of weather and scene complexity, and affected by sensor noise.

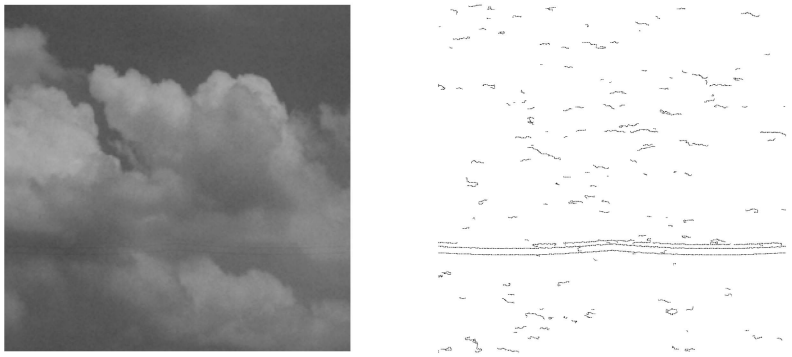
II. BACKGROUND

A. The Clutter Measure

In recognition or identification systems, high visual complexity becomes an especially difficult constraint when dealing with poorly specified targets, i.e., not salient and not predicted by its context. A common approach in computer vision has been to solve problems using targets of high contrast, unoccluded, and in typical settings. The clutter measure proposed by Schmieder and Weathersby [23] is sufficient to effectively characterize low-altitude aircraft images as low or high cluttered images [24]. In this work, an adaptive algorithm based on the clutter measure of different windows of an image is presented. The clutter measure is used to train the algorithm parameters. The clutter measure only characterizes the image in terms of visual complexity and not in terms of how difficult the image is to process in terms of detection, i.e., an image could be low cluttered but blurred, or have poor contrast wires

(a) (b) (c)

Fig. 2. Examples of different cluttered scenes. (a) Low cluttered scene. (b) Medium cluttered scene. (c) Highly cluttered scene.



(a) (b)

Fig. 3. Sample images used in previous work. (a) Natural background combined with synthetic wires. (b) Steger's algorithm output.

due to weather effects or video quality (see Fig. 2). The measure is based on the mean and standard deviation (μ,σ) of the intensity distribution by color at the pixel level

$$\mu = \frac{1}{N}\sum_{i=1}^{N}(X_i) \tag{1}$$

where μ is the intensity mean, X_i is the individual pixel intensity, and N the total number of pixels to be measured in a given window

$$\sigma = \sqrt{\frac{1}{N(N-1)}\sum_{i=1}^{N}(X_i-\mu)^2}. \tag{2}$$

The clutter measure of an image is given by average of the intensity variance of contiguous windows in the scene:

$$\text{clutter(image)} = \sqrt{\frac{1}{K}\sum_{i=1}^{K}\sigma_i^2} \tag{3}$$

where K represents the number of windows the image was divided into and σ is the intensity standard deviation described above in (2). In this work, the window size used is 25% of the original image, i.e., $K = 16$ (refer to Section IIIE for parameter selection details). An image is considered low cluttered if the clutter measure is less than 30 units, otherwise it is considered high cluttered. The threshold for low and high clutter was chosen using the clutter mean per video of our dataset $\mu = 30$. The clutter measure per image is used as a high level description of the dataset used, in terms of visual complexity. The clutter measure is also used in the parameter training process.

B. Baseline Algorithm

Kasturi, et al. in [13] proposed the use of a line detection algorithm, Steger's algorithm [25], followed by a Hough transform as a viable wire detection strategy for low-altitude rotorcraft images. Kasturi's approach is used as the baseline comparison for this work. The strategy was tested on synthetically generated images combining natural backgrounds; containing clouds and sky, with artificial wires corrupted with noise (see Fig. 3 for a sample grey-level image used in this work). The two kinds of synthetic wires used are hanging bridges (parabolas) and power lines (catenary curves). The noise appears in the image as breaks of the wire structure. A uniform distribution was assumed for the location of the breaks, the number of breaks is Poisson distributed, and the size of the breaks is Rayleigh distributed. The main difference among datasets is that the images used in our work contain urban characteristics (buildings, cables, trees, etc.), which make the detection process much more complex, due to the many linear patterns that are formed by the clutter and objects in the background.

Noise and clutter effects are reduced by eliminating short lines computed by the line detection algorithm. The Hough transform threshold for peaks

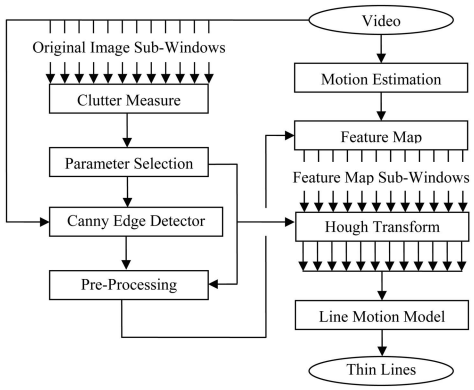


Fig. 4. Algorithm flowchart.

in the parameter space is given by

$$\text{Threshold} = \text{mean(votes)} + \tfrac{1}{2}[\max(\text{votes}) - \min(\text{votes})] \tag{4}$$

where mean, max, and min are the mean, maximum, and minimum vote counts in the parameter space. As the gap between the maximum and minimum vote count gets larger, the threshold becomes too lenient, resulting in a large false-alarm rate. The illumination model used to generate wires assumes ambient light reaches the entire wire equally, from all directions, which is true for visually simple images; thus, the test images were overly simple compared with real data. The performance of this strategy is summarized in the results section, and compared against the method proposed in this work. Steger's implementation was provided by the author. The problems that arise from the artificial data and the overall limitations of the approach, give valuable information that leads to the development of the algorithm presented in this paper.

III. THIN LINE DETECTION ALGORITHM

A. Algorithm Overview

The algorithm (see Fig. 4) detects thin lines by exploiting pixel motion, morphological properties, and linear patterns found in consecutive frames in low-quality video. Consecutive frames are used to estimate relative motion of individual pixels in the scene, and then the motion information is merged with an edge map produced by an edge detector. The feature map used consists of the detected edges with strength equal to their estimated motion. Next, the feature map is preprocessed in order to minimize the number of pixels to be further analyzed. Morphological filtering of 8-connected components found in the feature map is performed. The feature map is subdivided into windows and a Hough transform is performed in each of them. The resulting parameter spaces are tracked over time using a line motion model, which predicts the next location of previously tracked thin lines even when they are not fully visible. Section IIIE describes all the parameters used in the algorithm.

B. Feature Detection and Preprocessing

An edge map is computed using the Canny edge detector [26] with parameters (refer to Section IIIE for details on the parameters summary and training) chosen depending on the clutter measure of the entire image. Then, preprocessing uses two filtering techniques based on the morphology of the 8-connected components found in the edge map. Connected components with a pixel count less than $T_s = 30$ are eliminated. Also, components are filtered based on the eccentricity of the ellipse that has the same second-moments as the connected component. Components with eccentricity less than $T_e = 0.98$ are discarded. High motion values are used to increase the number of true line points to be fed into the Hough transform. Motion estimation at the pixel level is given by (5). Pixels that have motion estimation greater or equal than the 90% of the highest estimated pixel motion value in the image are kept.

The spatial and temporal derivatives $\partial I/\partial x$, $\partial I/\partial y$, $\partial I/\partial t$ of the image intensity function are used as image motion descriptors. The derivatives represent the normal flow at every pixel in the image, i.e., the projection of the optic flow in the direction of the image gradient (I_x, I_y). By definition, the normal flow $\overline{u_n}$ at any pixel $I(x, y)$ at time t is given by

$$\overline{u_n} = \frac{1}{\sqrt{I_x^2 + I_y^2}} \left(I_x \frac{\partial_x}{\partial_t} + I_y \frac{\partial_y}{\partial_t} \right). \tag{5}$$

Motion is only considered at those pixels present in the edge map. The normal flow is used as an estimated relative motion value. Given that the magnitude of the intensity gradient is large at the edge pixels, the projection of the optic flow on the direction of the local intensity gradient is equal to the normal flow [27]. Using normal flow as an image motion descriptor allows avoiding the computation of the more computationally expensive optic flow. The feature map is given by the combination of the edge map and the estimated motion.

C. Line Fitting using Windowed Hough Transform

Different regions of natural images will have significant variations in the amounts of clutter, thus causing a large number of incidental alignments that will correspond to strong peaks in the parameter space, i.e., false alarms (FA). The windowed Hough transform has been successfully used [28] to avoid incidental edge alignment produced by noisy edges or edges belonging to patterns not being searched

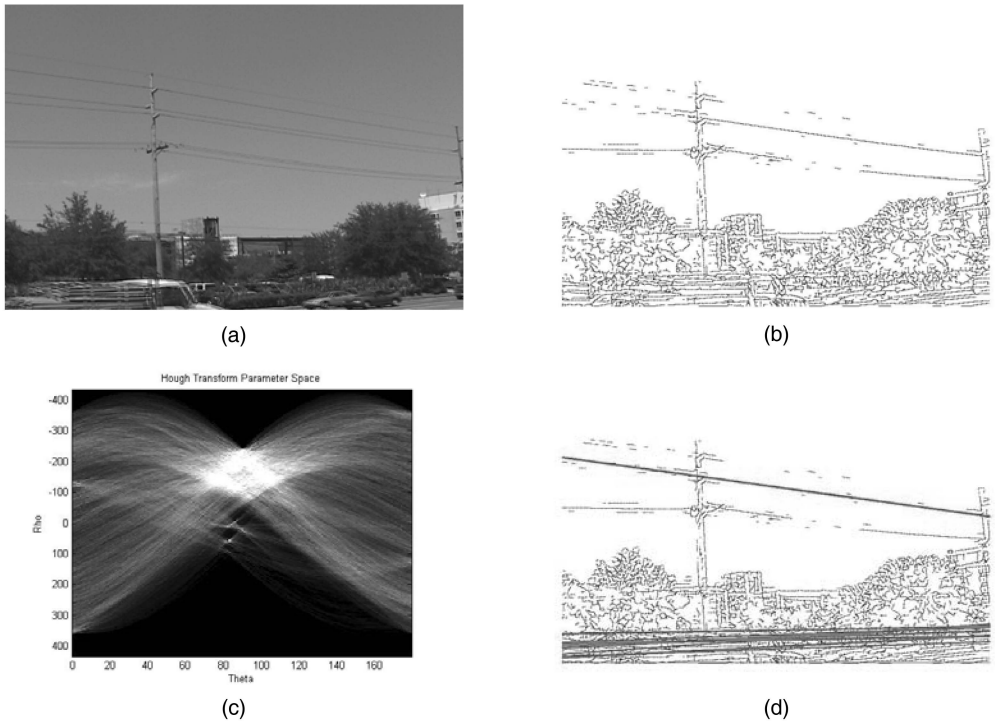


Fig. 5. Clutter effects on traditional Hough transform. (a) Original image. (b) Feature map produced by Steger's algorithm. (c) Parameter space. (d) Hough transform results.

for. A traditional Hough transform would often fail, in cluttered images, since the clutter peaks will be much stronger than the peaks generated by the thin obstacles (see Fig. 5). Typically, the size of the window depends on the size of the pattern being searched for, but we use large-sized windows in order to search for regional trends [29] rather than local trends. In this work the windows used are 25% of the size of the entire image, and a Hough transform is performed in each of them. Using a window-based approach also reduces the effects of clutter in the detection performance: it gives the algorithm various opportunities to detect a line if it appears in different windows in the image.

The local maxima, in each parameter space at time t, represent the evidence of thin lines within a window in that frame. In each window, our line candidates in the parameter space are given by pairs of the form (ρ,θ). The vote count threshold is learned based on the clutter measure of the window being processed. Due to the thickness and the catenary shape of wires and cables, some of these thin objects might produce multiple responses in the parameter space. In the parameter space, thin objects are being approximated piecewise by linear segments by using multiple (ρ,θ) pairs; therefore, within a window it would be redundant to further process those lines that are very close in terms of ρ and θ (ρ within 40 px and θ within 40°), in which case only the line with the highest vote count is kept. A weighted average based on the vote count of each line was also considered, but did not show any improvement in the algorithm's performance.

D. Parameter Space Tracking: the Line Motion Model

It is necessary to describe how lines are moving within a scene, and also what their relative motion is with respect to all other lines in the image. It is clear that small variations of clutter and video artifacts could affect potentially the feature map; for example, Fig. 6(a)–(b) show a small variation of the number of wire pixels found that leads to the still-frame misdetection shown in Fig. 6(c)–(d); consequently, there is an evident need to use persistence across frames of wires that have been previously found.

It is possible to keep the cumulative sum of the motion magnitude estimated for each feature pixel in a line. Every time the vote count on a bin in the parameter space is increased the mean motion of the entire line can be recalculated. Pixel-based motion values are highly sensitive to errors due to clutter and sensor artifacts (see Fig. 7). In order to minimize the effects of the clutter the motion vector at time t $\vec{v}_t = (|\vec{v}_t|, \alpha_t)$ is estimated after a line has been successfully tracked from a previous frame at time $t-1$. The motion magnitude is given by the Euclidean distance between the output of the tracking model at frames Z_{t-1} and Z_t. The direction is given by

$$\alpha_t = \tan^{-1}\left(\frac{Z_{t\rho} - \rho_{t-1}}{Z_{t\theta} - \theta_{t-1}}\right). \tag{6}$$

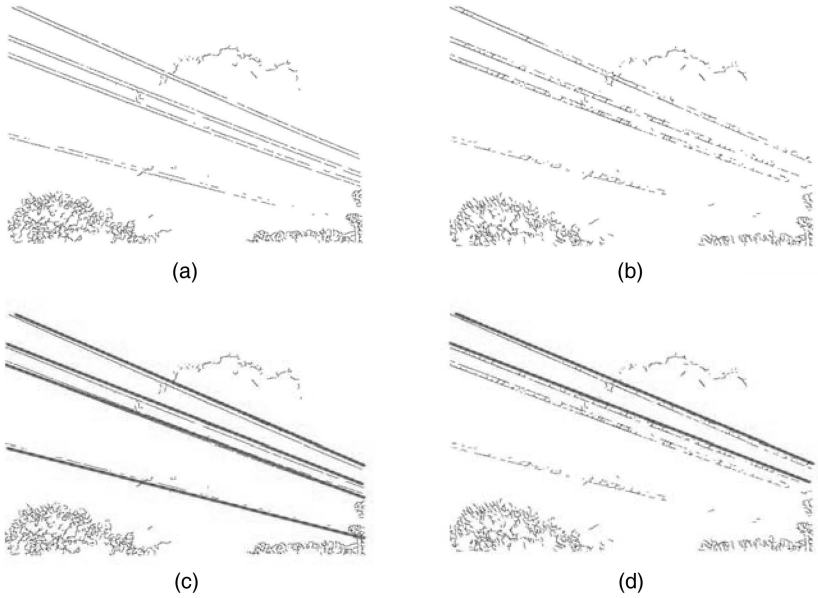


Fig. 6. Misdetection due to feature map inconsistencies in low-quality video. (a) Image frame. (b) Following frame. (c) Detection on (a). (d) Detection on (b).

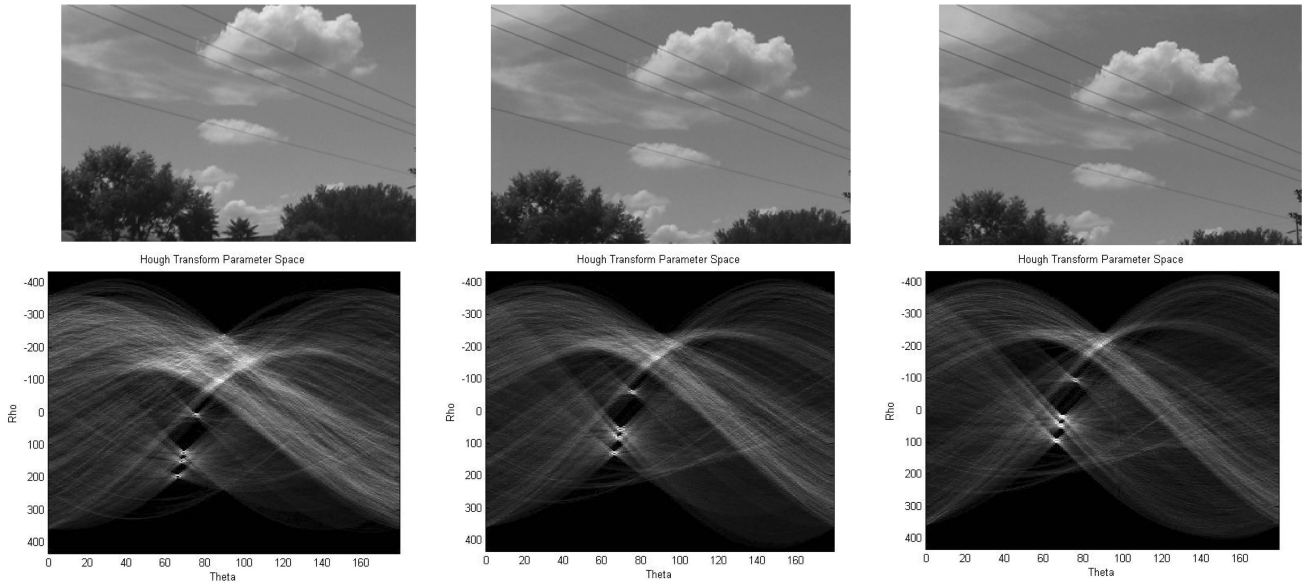


Fig. 7. Peak motion in parameter space.

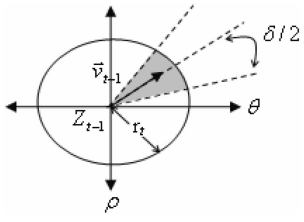


Fig. 8. Parameter space line movement model.

Given a previously tracked line $\vec{Z}_{t-1}$ we set a first estimate $\hat{x}_t^-$ to be the maxima within the shaded region (shown in Fig. 8) of the circle C of center ($Z_{(t-1)\rho}$, $Z_{(t-1)\theta}$) and radius r. δ is the tolerance for the area to be searched to compute the first estimate, as depicted in Fig. 8. Next, if a maxima $\hat{x}_t^+$, greater than $\hat{x}_t^-$, is found in C outside the shaded region we define our model output as

$$\vec{Z}_t = \frac{w_t^- \hat{X}_t^- + w_t^+ \hat{X}_t^+}{w_t^- + w_t^+} \tag{7}$$

where $w_t = e^{\text{votes}}$, votes is the vote count for $\hat{X}_t^-$ and $\hat{X}_t^+$. If $\vec{v}_{t-1}$ is not available we search C fully to get $\vec{Z}_t$. If no maximum is found in C, set $\vec{Z}_t$ to be a point at a distance $r/2$ from $\vec{Z}_{t-1}$ in the direction of $\vec{v}_{t-1}$. After each frame is processed, for each line we update the

model by using

$$r_{t+1} = \begin{cases} r_t + |\vec{v}_t| & \text{if} \quad |\vec{v}_t| > r_t \\ r_t - r_t - \dfrac{|\vec{v}_t|}{b} & \text{otherwise} \end{cases} \tag{8}$$

$$\delta_{t+1} = \begin{cases} \left(\dfrac{\delta_t}{2} + Z_{t\theta} - \alpha_t\right)/c & \text{if} \quad Z_{t\theta} < \alpha_t \\ \left(\dfrac{\delta_t}{2} + Z_{t\theta} + \alpha_t\right)/c & \text{otherwise} \end{cases}. \tag{9}$$

Assuming there is a constant rate of change of the motion vector in the parameter space, the model supports line movement in six degrees of freedom (up, down, left, right, perpendicular up, and perpendicular down). This assumption holds true for small aircraft and rotorcraft vehicles used in USAR missions. Given the low speed of USAR aircraft the following coefficients are suitable to be used in this domain: $a = 0.8$, $b = 2$, $c = 2$. The coefficient values a, b, c, and the initial values r_0, δ_0 depend on the motion characteristics, average speed, and maximum speed of the vehicles that the model approximates.

E. Parameters Summary and Training Strategies

All training was performed using the training dataset only. For training involving the creation of the feature map, the maximization criterion used is shown in (10). The maximization criterion was chosen with 2 main purposes: increase the vote count of true wires produced by the Hough transform (decreasing the possibility of miss detection and increasing the accuracy of the tracking process), and also, decrease incidental alignment in the windowed Hough transform produced by components generated by clutter (decreasing the FA rate). The connected component analysis uses 8-connectivity

$$\text{Maximization Criterion} = \frac{\text{No. True Edge Pixels}}{\text{No. Connected Components}}. \tag{10}$$

The Canny edge detector uses 3 parameters: sigma, and hysteresis thresholds parameters. The edge detection parameters are learned optimizing the maximization criteria with 1000 different combinations. Sigma is chosen for every image depending on the clutter measure of the entire image, while the thresholds remain constant for all images.

The preprocessing filtering uses 3 parameters. The size filtering threshold T_s is learned minimizing the least squared error between output and ground truth lines. The eccentricity threshold T_e was chosen in order to keep linear-shaped components. A motion threshold T_m was chosen manually, and it is used to ensure high-motion pixels are kept in the preprocessing step. In this work $T_s = 30$, $T_e = 0.98$, and $T_m = 0.9$.

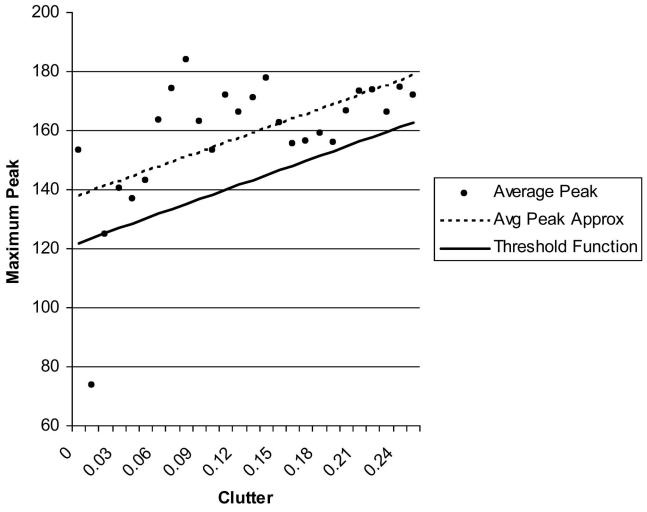


Fig. 9. Line candidate's threshold function based on window's clutter measure.

The windowed Hough transform uses 2 parameters: the window size and the vote threshold function. Large-sized windows are used in order to search for regional trends [29] rather than local trends. The window size to be processed was chosen manually and it is 25% of the original image (4×4 windowed Hough transform). By using the training data subset with no obstacles, the maximum vote count found in the parameter space for small clutter ranges in the average is shown in Fig. 9. Using a least squares algorithm the best first degree polynomial which best approximates the data is used as a baseline for the threshold of the parameter space. Quadratic and cubic models were also tested, but did not improve the algorithm performance. The Hough transform threshold parameter in each window is based on the clutter measure of the given window. Line candidates are those elements in the parameter space which vote count is greater than the threshold shown in (11), which represents a shifted approximating function. Equation (11) is shown in Fig. 9 labeled as threshold function

$$\text{Threshold}_{\text{win}} = X_1 + X_2 \text{clutter(win)} \tag{11}$$

where clutter(win) represents the clutter measure of the window being processed. Using the training dataset the threshold coefficients are as follows: $X_1 = 122$, $X_2 = 163$.

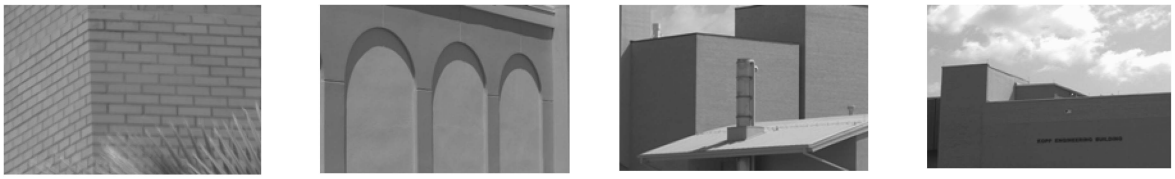
The tracking process uses 3 parameters. There are 3 line model coefficients that are learned using the maximum motion tracked with the training data, and they depend directly on the vehicle motion characteristics and speed. A redundant line criterion is defined as ρ within 40 px and θ within 40°. The redundancy criterion was chosen by inspection to represent the human perspective, i.e., two lines are redundant when a human would consider them as corresponding to the same ground truth.

TABLE I
Dataset Details

	Total		With Obstacles		No Obstacles		Low Clutter		High Clutter	
	Videos	Frames	Videos	Frames	Videos	Frames	Videos	Frames	Videos	Frames
Entire Dataset	86	5576 (100%)	55	3561 (64%)	31	2015 (36%)	44	2774 (50%)	42	2802 (50%)
Training	44	2864	13	849	31	2015	22	1379	22	1477
Testing	42	2712	42	2712	—	—	22	1395	20	1325

(a) (b) (c) (d)

Fig. 10. Low cluttered sample frames with obstacles. (a) Clutter 9. (b) Clutter 19. (c) Clutter 24. (d) Clutter 28.



(a) (b) (c) (d)

Fig. 11. Low cluttered sample frames with no obstacles. (a) Clutter 16. (b) Clutter 17. (c) Clutter 28. (d) Clutter 29.

IV. DATASETS

Search and rescue aircraft data is scarce and hard to obtain due to the significant risk involved with flying nearby obstacles within 15 m. The authors collected videos using a Cannon Optura 20, by manually panning the camera from left to right, right to left, top to bottom, and bottom to top in different locations in Florida and New Zealand, over a period of 4 months under different weather (good weather, mist, and moderate rain) and lighting conditions (time of day). The data was collected without using any stabilization hardware or software. Other videos were taken using unknown hardware and with various types of compression, and provided by the authors from search and rescue groups. There is a total of 9.58 min of video from helicopters and unmanned aerial vehicles (UAVs) that was used, including footage from a UAV crashing into power lines in a search and rescue mission at hurricane Katrina in Pearlington, MS. The only hardware specs known from aircraft are those from the footage taken at the University of South Florida using a Sony FCB-EX 980S from a search and rescue helicopter Maxi Joker 2. The objects present in the images are buildings, trees, power lines and wires. Each video is 30 fr/s and 60–70 frames long. Image sizes are 320×240 and 720×480 px (Florida data) or 640×480 px (Missouri/New Zealand). Ground truth lines were manually drawn in the 3259 frames with obstacles. A single straight line was used per each true obstacle. The training dataset was randomly selected out of the data collected manually.

For the entire dataset the clutter mean and standard deviation per video is $(\mu, \sigma) = (30, 12)$ with a minimum of 5 units and a maximum of 57 units. Low cluttered images are defined as those which have a clutter measure less than 30 units. Figs. 10–11 are sample frames of low cluttered images, Fig. 10(a)–(d) having obstacles and Fig. 11(a)–(d) not having any. Figs. 12–13 are sample frames of high cluttered images, Fig. 12(a)–(d) having obstacles and Fig. 13(a)–(d) not having any. Dataset details are shown in Table I.

V. RESULTS

Results shown in Table II summarize the performance of each algorithm used in this work. Detection is computed by line; thus, each line missed in a frame counts as a misdetection. A line is correctly detected if it is found within an angle and a distance tolerance threshold. The angle threshold is 40°, and the distance threshold is 40 px. The experiments performed in this work aim towards the validation of the algorithm's detection performance on stressed

(a) (b) (c) (d)

Fig. 12. High cluttered sample frames with wires. (a) Clutter 36. (b) Clutter 44. (c) Clutter 48. (d) Clutter 50.

(a) (b) (c) (d)

Fig. 13. High cluttered sample frames with no wires. (a) Clutter 40. (b) Clutter 45. (c) Clutter 52. (d) Clutter 57.

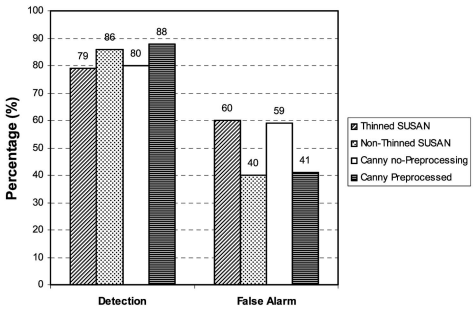


Fig. 14. Detection and false alarm performance comparison using different feature maps.

TABLE II
Detection Results using Test Dataset
(refer to Section IV for dataset details)

Algorithm	Detection	False Alarm
Baseline (*)	48%	75%
Parameter space double thresholds	81%	25%
Clutter-based single threshold	80%	25%
Hybrid	83%	25%
Hybrid (without tracking)	45%	25%

Note: *Baseline performance based on fixed threshold described by algorithm's authors.

conditions, the importance of using video versus still-frame processing, and the necessity of strong preprocessing techniques.

A. Using Alternative Edge Detectors for the Feature Map Creation

The Canny edge detector has been identified [30] to consistently offer one of best edge detector performances with respect of pixel-based ground truth metrics with real (nonsynthetic) images. Also, it has been found in [31] that for particular conditions, few approaches offer any general improvements over Canny. However, in order to address the validity of Canny's performance in the urban low-altitude domain, specifically for thin lines, a comparison between detection results using feature maps created by Canny and small univalue segment assimilating nucleus (SUSAN) [32] edge detectors is presented. SUSAN has been identified to have good performances compared with other edge detectors, including Canny, with particular edge-based recognition tasks such as in [33]. Fig. 14 shows that the performance improvement for using preprocessing (refer to Section IIIB), are similar regardless of the edge detection technique used to compute the feature map; furthermore, SUSAN and Canny edge detectors offer a similar performance in terms of detection of thin lines.

B. Detection Results

Table II shows a comparison between the baseline (refer to Section IIB) and the different techniques explored in this work. The detection and false alarm percentages are computed dividing the number of true and false positives by the number of ground truth wires. The baseline performance is based on the fixed threshold shown in (4). The second entry in Table II shows the detection performance of the algorithm using parameter space double-thresholds (12) and (13) considered in previous work [34]. In (12) and (13) the threshold becomes a function of the parameter space alone

$$\text{threshold}_{\text{dynamic}} = 0.75(\max(\text{votes})) \quad (12)$$

$$\text{threshold}_{\text{static}} = 0.12(\max\_\text{line}(\text{window})). \quad (13)$$

Equation (12) represents a dynamic threshold that depends on the parameter space, and it is the 75% of the maximum peak found in the window being

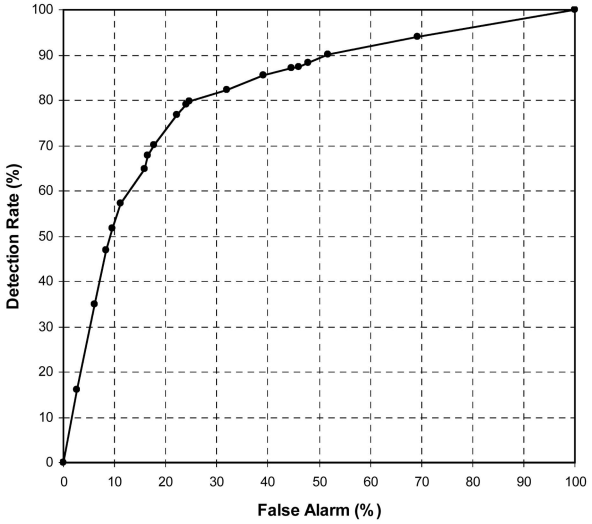


Fig. 15. ROC of performance with test dataset using hybrid thresholds.

processed. Equation (13) represents a static threshold since it is based on the window size being used (constant within a video), and it is 12% of the maximum line possible in a window. The parameter space double threshold performs best for low cluttered images; thus, Table II shows the performance of the hybrid threshold strategy that uses (12) and (13) for low cluttered windows, and (11) for high cluttered windows. Also, the last entry in Table II shows the performance using hybrid thresholds, but without using the tracking model. The tracking model allows the algorithm to achieve high detection rates. Without tracking the algorithm detection rate caps at 65% with FA greater than 75%; however, at 62% detection the FA is 72%, which represents a 15% detection and 3% FA improvement over the baseline. Fig. 15 shows a performance receiver operating characteristic (ROC) of the hybrid technique. The hybrid technique offers the best performance with the dataset used in this work.

VI. CONCLUSIONS

This work describes a thin line detection algorithm capable of providing high detection rate performance with low-altitude video in urban environments. The system was designed to support its use in USAR missions, which often involve highly cluttered urban scenes and uncontrolled weather conditions. Results are computed using a large and challenging multi-site low-video-quality dataset with 10160 thin obstacles spanning across 5576 frames. The detection process is performed on a feature map created by the use of estimated motion at the pixel level combined with an edge map computed by an edge detector. Similar to previous studies, it was determined that Canny and SUSAN edge detectors perform similarly. A windowed Hough transform converts the feature map in a multi-window parameter space in which tracking takes place.

Detection performance is determined by comparing the algorithm output with manual ground truth. Related previous work was used as a baseline for the performance comparison of the algorithm. For the same detection rate, the FA improvement over the baseline is 67%. Combined improvements over the baseline, of 34% FA and 41% detection, are shown. Also, improvements due to preprocessing and video usage are computed. The preprocessing filters account for an 8% detection and 18% FA performance improvement. The use of tracking in video enables the algorithm to achieve high detection rates, and opens the possibility to detect over 65% of the thin objects, that otherwise would not be possible.

REFERENCES

[1] Avizonis, P., and Barron, B.
Low cost wire detection system.
In *Digital Avionics Systems Conference*, vol. 1, 1999, 3.C.3-1–3.C.3-4.

[2] Frew, E., McGee, T., ZuWhan K., Xiao X., Jackson, S., Morimoto, M., Rathinam, S., Padial, J., and Sengupta, R.
Vision-based road-following using a small autonomous aircraft.
In *IEEE Proceedings of Aerospace Conference*, vol. 5, 2004, 3006–3015.

[3] McGee, T. G., Sengupta, R., and Hedrick, K.
Obstacle detection for small autonomous aircraft using sky segmentation.
In *IEEE International Conference on Robotics and Automation*, 2005, 4679–4684.

[4] Bermudez, S., Pyk, P., and Verschure, P. F. M. J.
A biologically based flight control system for a blimp-based UAV.
In *IEEE International Conference on Robotics and Automation*, 2005, 3053–3059.

[5] Ollero, A., Ferruz, J., Caballero, F., Hurtado, S., and Merino, L.
Motion compensation and object detection for autonomous helicopter visual navigation in the COMETS system.
In *IEEE International Conference on Robotics and Automation*, vol. 1, 2004, 19–24.

[6] Kurdila, A., Nechyba, M., Prazenica, R., Dahmen, W., Binev, P., DeVore, R., and Sharpley, R.
Vision-based control of micro-air-vehicles: Progress and problems in estimation.
Presented at the IEEE Conference on Decision and Control, 2004.

[7] Braithwaite, R. N., and Bhanu, B.
Hierarchical Gabor filters for object detection in infrared images.
In *Conference on Computer Vision and Pattern Recognition*, vol. 21–23, 1994, 628–631.

[8] Casasent, D., and Ye, A.
Detection filters and algorithm fusion for ATR.
IEEE Transactions on Image Processing, **6**, 1 (1997), 114–125.

[9] Arnold, J., Shaw, S. W., and Pasternack, H.
Efficient target tracking using dynamic programming.
IEEE Transactions on Aerospace and Electronic Systems, **29** (1993), 44–56.

[10] Barniv, Y.
Dynamic programming solution for detecting dim moving targets.
IEEE Transactions on Aerospace and Electronic Systems, **21**, 1 (1985), 144–156.

[11] Gandhi, T., Yang, M. T., Kasturi, R., Camps, O., Coraor, L., and McCandless, J.
Performance characterization of the dynamic programming obstacle detection algorithm.
IEEE Transactions on Image Processing, **15**, 5 (2006), 1202–1214.

[12] Blostein, S. D., and Huang, T. S.
Detecting small, moving objects in image sequences using sequential hypothesis testing.
IEEE Transactions on Signal Processing, **39**, 7 (1991), 1611–1629.

[13] Kasturi, R., Camps, O., Huang, Y., Narasimhamurthy, A., and Pande, N.
Wire detection algorithms for navigation.
NASA Technical Report, 2002.

[14] Muratet, L., Doncieux, S., Briere, Y., and Meyer, J.
A contribution to vision-based autonomous helicopter flight in urban environments.
Robotics and Autonomous Systems, **50** (2005), 195–209.

[15] Hills, M., Pridmore, T., and Mills, S.
Object tracking through a Hough space.
In *International Conference on Visual Information Engineering*, 2003, 53–56.

[16] Gandhi, T., Yang M. T., Kasturi, R., Camps, O., Coraor, L., and McCandless, J.
Detection of obstacles in the flight path of an aircraft.
IEEE Transactions on Aerospace and Electronic Systems, **39**, 1 (2003), 176–191.

[17] Yang, M. T., Gandhi, T., Kasturi, R., Coraor, L., Camps, O., and McCandless, J.
Real-time obstacle detection system for high speed civil transport supersonic aircraft.
In *Proceedings of the National Aerospace and Electronics Conference*, 2000, 595–601.

[18] Savan, M., and Barr, D. N.
Reflectance of wires and cables at 10.6 micrometer.
Center for Night Vision and Electro-optics, MSEL-NV-TR-0063, 1988.

[19] Yamamoto, K., and Yamada, K.
Analysis of the infrared images to detect power lines.
In *IEEE Conference on Speech and Image Technologies for Computing and Telecommunications*, vol. 1, 1997, 343–346.

[20] Sarabandi, K., Pierce, L., Oh, Y., and Ulaby, F. T.
Power lines: Radar measurements and detection algorithm for polarimetric SAR images.
IEEE Transactions on Aerospace and Electronic Systems, **30**, 2 (1994), 632–643.

[21] Fontana, R. J., Richley, E. A., Marzullo, A. J., Beard, L. C., Mulloy, R. W. T., and Knight, E. J.
An ultra wideband radar for micro air vehicle applications.
Presented at the IEEE Conference on Ultra Wideband Systems and Technologies, 2002.

[22] Garcia-Pardo, P., Sukhatme, G., and Montgomery, J.
Toward vision based safe landing for an autonomous helicopter.
Robotics and Autonomous Systems, **38** (2002).

[23] Schmieder, E., and Weathersby, M. R.
Detection performance in clutter with variable resolution.
IEEE Transactions on Aerospace and Electronic Systems, **AES-19** (1983).

[24] Candamo, J., Kasturi, R., Goldgof, D., and Sarkar, S.
Vision-based on-board collision avoidance system for aircraft navigation.
Proceedings of SPIE, vol. 6230 (Unmanned Systems Technology VIII), 2006.

[25] Steger, C.
An unbiased detector of curvilinear structures.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **20**, 2 (1998), 113–125.

[26] Canny, J.
A computational approach to edge detection.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **8** (1986), 679–698.

[27] Huang, L., and Aloimonos, J. Y.
Relative depth from motion using normal flow: An active and purposive solution.
In *Proceedings of the IEEE Workshop on Visual Motion*, 1991.

[28] Jung, C. R., and Schramm, R.
Rectangle detection based on a windowed Hough transform.
In *Proceedings on Computer Graphics and Image Processing*, 2004, 113–120.

[29] Saether, B., Rueslatten, H., and Gronlie, A.
Application of the Hough transform for automated interpretation of linear features in imageries.
In *Geoscience and Remote Sensing Symposium, Surface and Atmospheric Remote Sensing*, vol. 2, 1994, 847–850.

[30] Shin, M. C., Goldgof, D. B., Bowyer, K. W., and Nikiforou, S.
Comparison of edge detection algorithms using a structure from motion task.
Transactions on Systems, Man, and Cybernetics, Pt. B, **31**, 4 (2001), 589–601.

[31] Bowyer, K., Kranenburg, C., and Dougherty, S.
Edge detector evaluation using empirical ROC curves.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 1999, 354–359.

[32] Smith, S. M., and Brady, M.
SUSAN—A new approach to low level image processing.
International Journal of Computer Vision, **23**, 1 (1997), 45–78.

[33] Shin, M. C., Goldgof, D. B., and Bowyer, K. W.
Comparison of edge detector performance through use in an object recognition task.
Computer Vision and Image Understanding, **84**, 1 (2001), 160–178.

[34] Candamo, J., Kasturi, R., Goldgof, D., and Sarkar S.
Vision-based on-board collision avoidance system for aircraft navigation.
Proceedings of SPIE, vol. 6230, 2006.

Joshua Candamo received his B.S. in computer science in 2003 from the University of South Florida, Tampa, FL.

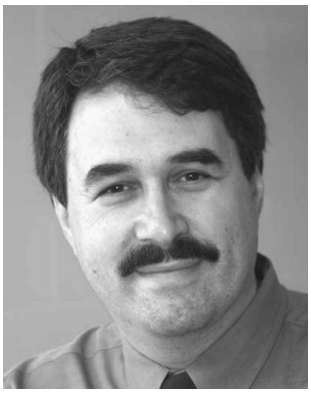
He is currently pursing a Ph.D. degree in computer science at the University of South Florida where he currently works as a graduate research assistant. In the past, he has taken part in several internships opportunities as a software engineer. He received the Pan-American Studies Institute scholarship to attend internship in the Universidad de Aquino Bolivia in 2004, focusing on digital signal processing, communications, instrumentation, and control systems. In 2003, as a software engineer co-op for REIMELT Corporation he focused on application development and interface design for programmable logic controllers. As an intern for the Department of Defense in 2007, he developed an analytical tool to estimate learning effects in recurring production of the joint light tactical vehicle program. His research interests include computer vision, pattern recognition, and artificial intelligence applied to homeland security applications.

Mr. Candamo is a recipient of the Alfred P. Sloan fellowship. He received 1st place in the 2005 USF computer science and engineering Ph.D. research competition, for his work in visual navigation using video for search and rescue unmanned aerial vehicles. His research work currently includes vision-based surveillance for mass transit systems, and low-altitude aircraft reconnaissance in urban areas.

Rangachar Kasturi (S'82—M'82—SM'88—F'96) received his B.E. (electrical) degree from Bangalore University, India, in 1968 and M.S.E.E. and Ph.D. degrees from Texas Tech University, Lubbock, in 1980 and 1982, respectively.

He was a professor of Computer Science and Engineering and Electrical Engineering at the Pennsylvania State University during 1982–2003 and was a Fulbright Scholar during 1999. His research interests are in document image analysis, video sequence analysis and biometrics.

Dr. Kasturi is the 2008 President of the IEEE Computer Society. He was the President of the International Association for Pattern Recognition (IAPR) during 2002–04. He has served as the editor-in-chief of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1995–98) and the *Machine Vision and Applications* (1993–94) journals. Dr. Kasturi is a Fellow of IAPR. He is an author of the textbook, *Machine Vision*, (McGraw-Hill, 1995).



Dmitry Goldgof (M'83—SM'90—F'07) received his M.S. degree in computer engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1985 and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign in 1989.

He is currently a professor and an associate chair in the Department of Computer Science and Engineering and a member of H. Lee Moffitt Cancer Center and Research Institute, where during 2002–2003 he held a position of professor in Bioinformatics and Cancer Control. Previously, Dr. Goldgof held visiting positions at the Department of Computer Science at the University of California at Santa Barbara and at the Department of Computer Science at University of Bern in Switzerland. His research interests include motion and deformation analysis, image analysis and its biomedical applications, bioinformatics, and pattern recognition.

Dr. Goldgof has graduated 12 Ph.D. and 35 M.S. students, edited four books, published over 60 journals and over 130 conference papers, and was granted US patent. He has served as IEEE Distinguished Visitor 2004–2006 and on the Board of Governors of Systems, Man and Cybernetics Society in 2007. Dr. Goldgof was awarded Annual Pattern Recognition Society Awards in 1993 and 2002. One of his papers was selected by International Medical Informatics Association for 2000 Yearbook containing "the best of medical informatics." Dr. Goldgof is an associate editor for *IEEE Transactions on Systems, Man and Cybernetics* Pt. B, and for the *International Journal of Pattern Recognition and Artificial Intelligence*. He has been involved in numerous professional society activities, has served as North American editor for the *Image and Vision Computing Journal*, member of the Editorial Board of the Pattern Recognition, and associate editor for *IEEE Transactions on Image Processing*. Currently, Professor Goldgof is a local arrangement chair for ICPR 2008 to be held in Tampa in December of 2008.

Sudeep Sarkar (M'93—SM'95) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, in 1988. He received the M.S. and Ph.D. degrees in electrical engineering, on a University Presidential Fellowship, from The Ohio State University, Columbus, in 1990 and 1993, respectively.

Since 1993, he has been with the Computer Science and Engineering Department at the University of South Florida, Tampa, where he is currently a professor. His research interests include perceptual organization in single images and multiple image sequences, automated American Sign Language recognition, biometrics, gait recognition, and performance evaluation of vision systems. Dr. Sarkar is the coauthor of the book, *Computing Perceptual Organization in Computer Vision*, published by World Scientific. He is also the coeditor of the book, *Perceptual Organization for Artificial Vision Systems*, published by Kluwer Publishers. He is the recipient of the National Science Foundation CAREER award in 1994, the USF Teaching Incentive Program Award for undergraduate teaching excellence in 1997, the Outstanding Undergraduate Teaching Award in 1998, and the Theodore and Venette Askounes-Ashford Distinguished Scholar Award in 2004. He served on the editorial boards for the *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1999–2003) and *Pattern Analysis & Applications Journal* during (2000–2001). He is currently serving on the editorial board of the *Pattern Recognition Journal* and the *IEEE Transactions on Systems, Man, and Cybernetics*, Pt. B.