Comparison of Edge Detection Algorithms Using a Structure from Motion Task

Min C. Shin, *Member IEEE*, Dmitry B. Goldgof, *Senior Member, IEEE*, Kevin W. Bowyer, *Fellow, IEEE*, and Savvas Nikiforou

***Abstract*—This paper presents an evaluation of edge detector performance. We use the task of structure from motion (SFM) as a "black box" through which to evaluate the performance of edge detection algorithms. Edge detector goodness is measured by how accurately the SFM could recover the known structure and motion from the edge detection of the image sequences. We use a variety of real image sequences with ground truth to evaluate eight different edge detectors from the literature. Our results suggest that ratings of edge detector performance based on pixel-level metrics and on the SFM are well correlated and that detectors such as the Canny detector and Heitger detector offer the best performance.**

***Index Terms*—Edge detection, experimental comparison of algorithms, performance evaluation, structure from motion.**

I. Introduction

TABLE I gives a summary of the literature dealing with edge detector performance evaluation. Many of the older works attempted to evaluate edge detector performance using only synthetic images [1]–[5]. The motivation for using synthetic images is that it is easy to have reliable pixel-based ground truth. More recently, some authors have developed evaluation methods using pixel-based ground truth on real images. Use of real images, rather than synthetic, should inspire greater confidence in the results.[1] Heath *et al.* [6] approach the problem from a different perspective, rating performance using human evaluation of edge detector output for real images. However, high performance on pixel-based metrics or human evaluation will not necessarily translate to high performance on computer vision tasks which follow the edge detection step.

Our work is the first to focus on "task-based" empirical evaluation of edge detector performance. We use structure from motion (SFM) as a "black box" through which to evaluate the performance of edge detection algorithms [7], [8]. We have created 18 real image sequences of laboratory scenes. These represent three different scenes of each of two types, and three "densities" of motion sampling ($3 \times 2 \times 3 = 18$ sequences). A mechanical rotation stage was used to independently record accurate motion ground truth. Knowledge of the angle between selected scene features is used to independently record ground truth for structure. Edge detector goodness is measured by how accurately the SFM motion recovers the known structure and motion from the edge detection of the image sequences.

We have conducted experiments with eight edge detectors. For all but the Canny and the Sobel detectors, the edge detector implementation was obtained from its original authors or was validated against their results. We have conducted experiments with different SFM algorithms, due to Taylor and Kriegman [9] and Weng *et al.* [10]. A train-and-test methodology is used for computing performance metrics. We have designed a large and appropriately challenging dataset. The magnitude of the error in the test results shows that the real image sequences we use are sufficiently challenging for the SFM implementations and indeed result in meaningful ranking of the detectors. The result is an objective, quantitative, and fully reproducible comparison of edge detector performance.

TABLE I
SUMMARY OF EDGE DETECTION COMPARISON METHODS

source	nature of evaluation	images	GT required	# of algorithms compared
Fram & Deutsch '75 [11]	MLE of ratio of true edge signal	1 synth 1 real	Yes	3
Bryant & Bouldin '79 [12]	relative & absolute grading	1 real	Yes/No	3
Abdou & Pratt '79 [13]	probability of false/true detection figure of merit	1 synth 3 real	Yes	4
Shaw '77 [14]	relative edge location & direction test	1 real	Yes	5
Kitchen & Rosenfeld '81 [1]	thinness & continuation of edges	2 synth	No	4
Ramesh & Haralick '92 [15]	probability of false alarm & misdetection	1 synth 2 real	Yes	3
Peli & Malah '82 [2]	contour type, avg squared deviation mean abs value of deviation	2 synth	Yes	5
Lunscher & Beddoes '86 [3]	same as Kitchen & Rosenfeld	2 synth	No	5
Venkatesh & Kitchen '92 [4]	combined error measures with quality based weights	3 synth	Yes	4
Strickland & Chang '93 [5]	6 edge qualities	1 synth	Yes	5
Jiang '95 [16]	pixel by pixel comparison of range images	80 real (range)	Yes	9
Salotti & Bellet '96 [17]	pixel level comparison with edge classification	2 real	Yes	2
Zhu '96 [18]	connectivity & width uniformity check	2 synth 2 real	No	3
Heath *et al* 97 [6]	subjective human object recognition	28 real	No	5

A variation of edge detector is not counted as a separate edge detector compared

II. Background

Table II lists 19 edge detector papers published in a selection of major journals in 1995 through 2000. The important point in this table is that there is no generally accepted formal method of evaluating edge detection. The most common form of "evaluation" is to show the results of a proposed new detector side-by-side with those of a version of the Canny detector, and appeal to subjective evaluation of the reader. The ap-

Manuscript received June 23, 2000; revised February 3, 2001. This work was supported by NSF Grants CDA-9724422, EIA-9729904, IRI-9619240, and IRI-9731821. This paper was recommended by Associate Editor B. Draper.

The authors are with the Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620 USA (e-mail: shin@csee.usf.edu; goldgof@csee.usf.edu; kwb@csee.usf.edu; nikiforo@csee.usf.edu).

Publisher Item Identifier S 1083-4419(01)04862-2.

[1]This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. This includes the wood block and Lego house image data sets, a set of Solaris C programs for manipulating and interpreting the images, a setup script and readme file. This material is 34.3 MB in size

TABLE II
SUMMARY OF JOURNAL-PUBLISHED EDGE DETECTION ALGORITHMS

source	nature of algorithm	image set	real image ground truth	compared algorithms	visual comparison	quantitative comparison
PR '98 [?]	single filter	0 synth 1 real	0	2	Yes	None
PR '98 [20]	multi-scale feature binding	0 synth 4 real	0	0	Yes	None
PR '98 [21]	multi-scale morphologic	1 synth 1 real	0	1	Yes	None
PR '95 [22]	simple directional edge detection	1 synth 3 real	0	1	Yes	None
PR '95 [23]	multi-channel filtering	3 synth 5 real	0	0	No	None
PR '95 [24]	isotropical 2-D ISEF cascade	1 synth 1 real	0	1	Yes	None
PR '95 [25]	relaxation labeling	1 synth 4 real	0	2	Yes	Built Own
PR '95 [26]	multi-resolution	1 synth 6 real	0	2/5 [1]	Yes	Abdou & Pratt
PAMI '00 [28]	gradient covariance	1 synth 3 real	0	1	Yes	None
PAMI '98 [28]	local minimum reliable scale	4 real	0	4	Yes	None
PAMI '97 [29]	facet model & relaxation labeling	3 real	0	2	Yes	Built Own
PAMI '96 [30]	fuzzy reasoning	5 real	0	1	Yes	None
PAMI '96 [31]	expansion matching	1 synth 1 real	0	1	Yes	Abdou & Pratt
PAMI '95 [32]	logical/linear operator	3 real	0	1	Yes	None
PAMI '95 [33]	covariance models	3 real	0	0 [3]	No	None
GMIP '96 [34]	computational zero-crossing	1 synth 2 real	0	1	Yes	None
SMC '00 [35]	probabilistic relaxation	2 synth	0	1	Yes	None
SMC '97 [36]	detection on textured and untextured images	4 synth 2 real	0	0	Yes	None
SMC '95 [37]	regularized cubic B-spline fitting	2 synth 1 real	0	2	Yes	Built Own

1. 2 algorithms compared quantitatively
2. 5 algorithms compared quantitatively
3. no algorithms compared for 2D edge detection
4. edge map of Sobel operator on uncompressed image used as ground truth
image set includes any image mentioned in the edge detection work
GMIP (CVGIP: Graphical Models and Image Processing)
PAMI (IEEE Transactions on Pattern Analysis and Machine Intelligence)
PR (Pattern Recognition)
SMC (IEEE Transactions on Systems, Man, and Cybernetics)

TABLE III
PARAMETER RANGE OF EDGE DETECTORS

edge detector	parameters		
Anisotropic	σ_e (0.0 - 10.0)		
Bergholm	S_{start} (0.5 - 5.0)	S_{end} (0.5 - 5.0)	T (5.0 - 60.0)
Canny	σ (0.01 - 5.0)	*low* (0.0 - 1.0)	*high* (0.0 - 1.0)
Heitger	σ (0.5 - 5.0)	T (0.0 - 50.0)	
Rothwell	σ (0.5 - 4.0)	T (0.0 - 60.0)	*alpha* (0.0 - 1.0)
Sarkar	σ (0.01 - 5.0)	*low* (0.0 - 1.0)	*high* (0.0 - 1.0)
Sobel	*low* (0.0 - 1.0)	*high* (0.0 - 1.0)	
SUSAN	T (0.0 - 50.0)		

peal to the reader's visual evaluation may be aided by pointing out specific differences in some area(s) of the edge images. The lack of an accepted formal method of evaluating edge detector performance is not the result of a lack of possible approaches. Table I lists 14 different approaches proposed in the literature since 1975. The important point here is that to date no method has become generally adopted by the research community.

A. Edge Detectors

We have evaluated eight edge detectors in this framework. In this section, the description of edge detectors, their parameters (Table III), and the source of the implementations are given. We have selected the algorithms so that there is a diversity in the approaches of algorithms. We have used the same implementations of detectors as other edge detection evaluation frameworks [6], [38], [39], so that the results of evaluations could also be compared. The implementations of the Bergholm and SUSAN detectors were obtained from the original authors. The implementation of the Rothwell detector was translated from the C++ implementation in the IUE. The implementation of the robust anisotropic edge detector was checked against the original by computing values in [40]. The Canny detector was rewritten at USF from an implementation obtained from the University of Michigan. The Sobel was written at USF. Note that nonmaximal suppression and/or hysteresis were added to several detectors, in order to have all of the detectors produce edge maps with single-pixel edges.

The robust anisotropic edge detector [40] applies nonuniform smoothing to the image and selects the outliers of the robust estimation framework as edges. It estimates the noise level within the image (σ_e) using median absolute deviation (MAD). The detector is intended to be "parameterless," computing the MAD automatically for each image. The implementation we use scales the values of σ_e, giving one parameter. Nonmaximal suppression (directly taken from the Canny detector implementation) was added to the algorithm.

The Bergholm edge detector [41] applies a concept of edge focusing to find significant edges. The image is smoothed using a coarse resolution (high smoothing) from S_{start}. In addition, after applying nonmaximal suppression, the edges exceeding a contrast threshold T are identified. Then the locations of these edges are refined by tracking them to a finer resolution (less smoothing) of S_{end}.

The Canny edge detector [42] is widely considered the standard method of edge detection. The image is smoothed with a Gaussian filter where σ is the standard deviation of the filter. Then, the edge direction and strength are computed. The edges are refined with nonmaximal suppression for thin edges and hysteresis thresholding (*low* and *high*) for continuous edges. The implementation used in this work produces at least a 3×3 smoothing window for any positive σ value.

The Heitger edge detector [43] uses a "suppression and enhancement" concept. First, the image is smoothed by odd and even symmetrical Gabor filters. The filter responses that do not correspond to the position and type of a given image curve are suppressed while features that do correspond are enhanced. The edges are thinned with nonmaximal suppression. The original implementation contains 14 parameters. We have fixed 12 parameters at the default values, and tuned only σ for Gaussian envelope and the edge strength threshold T.

The Rothwell edge detector [44] is similar to the Canny edge detector except 1) nonmaximal suppression is not used since it is claimed that it fails at junction points, and 2) hysteresis is not used due to the belief that the edge strength is not relevant for the higher level image processing tasks. Rather than hysteresis, the detector uses "dynamic thresholding" where the thresholding for determining the edges varies across the image to improve the detection of the junctions. The image is smoothed with a Gaussian filter using the size of σ. The pixels are classified as edges if the gradient is greater than αT_{xy}. Then, the edges are thinned.

The Sarkar–Boyer edge detector [45] is an optimal zero crossing operator (OZCO). In order to obtain a good approximation and optimal response, the optimal infinite length response is approximated recursively. First, the infinite impulse response (IIR) filter (smoothing size determined by σ) is applied to the image. Then, after computing the third derivative of the smoothed image, the edges with negative directional

slopes are eliminated. Then, the false positive edges are further reduced by applying hysteresis.

The Sobel detector [46] uses the gradient operators along the x and y direction to compute the gradient magnitude. In the implementation, we added nonmaximal suppression and the hysteresis directly from the Canny detector.

The smallest univalue segment assimilating nucleus [47] (SUSAN) detector calculates the edge strength using the number of edges with similar brightness within circular masks. The USAN of each pixel is computed by placing the mask's center at the pixel. Then, if the area of the USAN is about half the area of the mask, the pixel is considered as an edge candidate. Then, nonmaxima suppression is used to thin edges.

B. Structure from Motion

A structure from motion (SFM) algorithm determines the structure (depth information) of a scene and the motion of the camera. The Taylor algorithm [9] is a nonlinear algorithm and requires multiple iterations for convergence of a solution, while the Weng algorithm [10] is a linear algorithm. These algorithms were tested against each other in [9]. Since the Weng algorithm can only handle three-frame sequences, it is tested only on three-frame sequences. Therefore, the results are presented in two different settings: 1) using long sequences with only Taylor algorithm, and 2) using three-frame sequences with both algorithms. We have obtained the implementations of the Taylor and Weng algorithms from the authors of [9].

The Taylor (nonlinear) SFM algorithm extracts the 3-D location of each line in the camera coordinates, and computes the motion of the camera given n images with m corresponded lines. It solves the problem in terms of an objective function $\mathcal{O}$ that measures the disparity between the projected 3-D line and its corresponding observed two-dimensional (2-D) line. The algorithm iterates searching for the structure and motion estimates which minimize $\mathcal{O}$. A minimum of three images and six lines is required by the SFM algorithm, and more images and lines are allowed. The algorithm generates an initial random guess of camera position for each iteration using the initial motion information. It is found that without providing any initial motion information, the algorithm usually manages to converge to a solution but after a far greater number of iterations. In order to speed up the optimization process for our experiments, the ground truth (GT) rotation angle is provided as a good initial guess.

The Weng (linear) algorithm uses image sequences with three frames only. It accepts m corresponded lines that are present in all three frames. Given the pixel location of the end points of the lines, the SFM algorithm estimates the rigid motion of the camera in terms of rotation and translation. Lines are represented using parametric form. The *intermediate parameters* (E, F, G) are computed from the set of lines. Then, the motion parameters are computed from E, F, and G. The structure vector including the sign of the vector is computed using the *majority positive-z assumption* which states that for the most of the lines, the point on the line that is closest to the origin has positive Z component. It requires minimum of 13 lines.

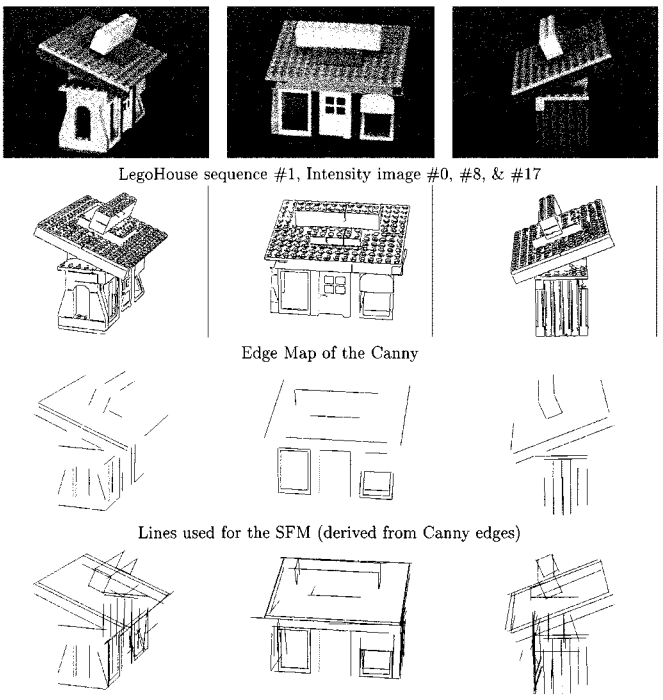


Fig. 1. Overview of the framework: intensity images, Canny edge maps, lines used for SFM, and the projection of the SFM output for corresponding camera motion of estimated structure. Note that the edges found on the right boundary of the intensity image are due to intensity difference along the background and the rightmost columns of the image. This did not affect the results as the edges found along those columns are not used for the SFM algorithm since they were not specified as the GT lines for correspondence.

III. Framework

The comparison methodology involves four steps:

1) edge detection;
2) intermediate processing;
3) SFM;
4) evaluation of the result.

An overview of this process is depicted in Fig. 1.

A. Intermediate Processing

Intermediate processing involves extracting and determining the correspondence of lines from the edge maps. The algorithms used are not necessarily the most sophisticated. Since the overall goal is to compare the edge detectors, it is only important that the intermediate processing algorithms not give any relative advantage to any particular detector.

1) Line Extraction: Initial pixels of edge chains are found in a raster scan. The eight-connected neighboring edge pixels are recursively linked until 1) there are no more neighbors or 2) there is more than one neighbor, indicating a possible junction. Then, edge chains are divided at high curvature points. Next, edge chains are further broken using the polyline splitting technique [48]. Then, least-squares is used to fit a line to each chain [49]. The line segment is represented by its endpoints, found by the position of the end pixels in the chain in the line equation.

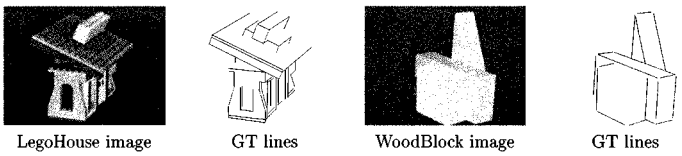


Fig. 2. Manually selected GT lines of LegoHouse1 (LH1) and WoodBlock1 (WB1) image sequences. Note that the lines around the LegoHouse roof are not selected since the lines were not straight due to the bubble of the Lego blocks.

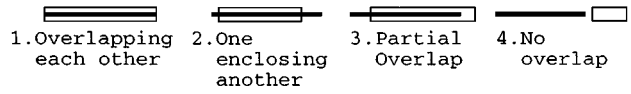


Fig. 3. Four possible orientations of collinear lines.

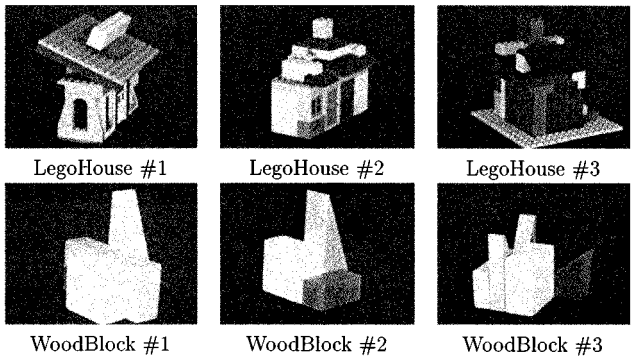


Fig. 4. Image scenes. First images of six original sequences are shown.

Finally, lines shorter than T_{line_length} (50.0 pixels) are eliminated.

2) Line Correspondence: The input to the SFM is a set of line correspondences across a sequence of frames. Manually matching the lines would provide the most error-free matching. However, it is not practical to manually match lines from 18 image sequences for eight edge detectors where each edge detector will be tested on a minimum of 177 (for three parameters), 41 (for two parameters), or 17 (for one parameter) parameter settings (see Section III-E for determining the minimum number of attempted parameter settings.) Therefore, an automated line matching program was developed

1) to provide correct correspondence information to the SFM algorithm so that the quality of the SFM output is due to the quality of the edge map;
2) to make the framework practical since manual matching is not feasible;
3) not to give any particular advantage or disadvantage to any edge detector.

First, the ground truth (GT) lines in all images of the sequence (L_{ij} for image i and line j) are manually defined and corresponded. An example of the GT lines for correspondence is shown in Fig. 2. In GT, we defined the lines that 1) are straight and 2) have enough separation from other lines since lines that are too close might lead to wrong correspondences. Second, the machine estimated (ME) lines (l_{ik} for image i and line k) from the line extraction step are corresponded automatically using the GT lines. If a ME line (l_{ik}) matches to a GT line (L_{ij}), l_{ik} is labeled with the index of L_{ij}. Two lines are corresponded if the following two conditions are met.

1) *Collinearity:* If the sum of the perpendicular distance between the endpoints of L_{ij} to line l_{ik} is less than T_{perp_dist} (5.0 pixels), they are considered to be collinear.
2) *Overlap*: Two line segments could be collinear, yet not belong to the same part of the object/image. So, l_{ik} is projected to L_{ij}, resulting in l'_{ik}. L_{ij} and l'_{ik} could be oriented in four different ways (refer to Fig. 3). Obviously, the orientation #1 indicates overlap while #4 indicates nonoverlap. Since one GT line segment could be broken down into several ME line segments in one image, if l'_{ik} is completely within L_{ij} (#2), they are corresponded. Also, if they partially overlap (#3) by at least $T_{overlap}$ % (80.0%) of the GT line, they correspond. Note that correspondence of many GT lines to one ME line is not allowed.

The SFM requires a minimum of three correspondences (correspondences over minimum of three frames) for each line. Lines with fewer than three correspondences are dropped.

B. Imagery Design

Tables I and II show that almost all previous comparisons have used only a handful of images (synthetic and real). Ideally an evaluation dataset should be large and thorough in order for the users of the methodology to have confidence in the evaluation.

1) Long Sequences: In this work, image sequences were captured for two different scene types: LegoHouse (LH) and WoodBlock (WB) (Fig. 4) For each scene type, three original sequences were obtained (two scene types $\times$ three original sequences per scene type $=$ six original sequences). Then, two shorter "derived" sequences were extracted from each original sequence (six original $\times$ two derived per original $=$ 12 derived), resulting in 18 total sequences (six original $+$ 12 derived). This large dataset of 18 sequences containing 133 unique images (278 including the repeated images in derived sequences) is carefully designed considering different aspects influencing the edge detectors and the structure from motion task, such as

1) number of images in a sequence;
2) total rotation angle;
3) number of lines in the scene;
4) average number of correspondences per line;
5) average line length;
6) number of contrast levels within the image;
7) amount of occlusion, transparency (which results in natural noise) (refer to Table IV).

2) Three-Frame Sequences: In order to compare the Taylor and Weng algorithms, sequences consisting of three frames are created. Twelve sequences are created from the LegoHouse scenes by taking nonoverlapping sub-sequences of original LegoHouse sequences (refer to Table V). For instance, LegoHouse.3 frame 1.A, B, and C are created by taking three

TABLE IV
PROPERTIES OF LONG IMAGE SEQUENCES WITH DERIVED SEQUENCES DENOTED BY ".A" OR ".B" SUFFIX

image set name	No. of images in sequence	total rotation angle	No. of lines	Avg No. of corres per line	Ave line length (pixel^2)
LegoHouse1	18	160°	122	8.8	80.6
LegoHouse1.A	12	160°	122	5.9	80.3
LegoHouse1.B	9	160°	122	4.5	81.4
LegoHouse2	19	355°	104	6.9	89.4
LegoHouse2.A	13	355°	104	4.8	88.4
LegoHouse2.B	10	355°	104	3.6	89.2
LegoHouse3	20	190°	118	7.6	83.3
LegoHouse3.A	14	190°	118	5.3	84.1
LegoHouse3.B	10	180°	118	3.8	82.2
WoodBlock1	18	170°	29	11.0	132.9
WoodBlock1.A	9	160°	29	5.5	133.9
WoodBlock1.B	12	160°	29	7.3	133.7
WoodBlock2	28	275°	36	15.7	110.9
WoodBlock2.A	14	265°	36	7.9	110.6
WoodBlock2.B	17	275°	36	9.5	111.9
WoodBlock3	30	285°	47	18.2	91.9
WoodBlock3.A	15	270°	46	9.2	92.0
WoodBlock3.B	10	260°	45	6.2	92.8

TABLE V
PROPERTIES OF THREE-FRAME IMAGE SEQUENCES

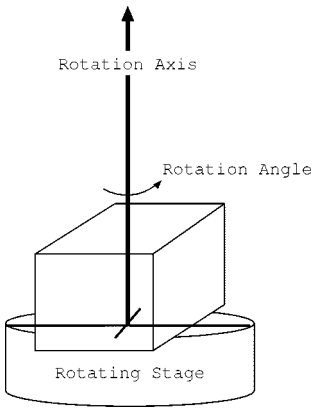
image set name	# of images in sequence	total rotation angle	# of lines	avg # of corres per line	avg line length (pixel^2)
LegoHouse.3frame1.A	3	50.0°	88	2.2	76.4
LegoHouse.3frame1.B	3	65.0°	95	1.9	79.4
LegoHouse.3frame1.C	3	20.0°	75	2.5	74.4
LegoHouse.3frame2.A	3	40.0°	77	1.7	84.6
LegoHouse.3frame2.B	3	45.0°	61	2.4	81.4
LegoHouse.3frame2.C	3	40.0°	104	1.0	95.7
LegoHouse.3frame3.A	3	50.0°	75	1.4	86.4
LegoHouse.3frame3.B	3	50.0°	88	1.5	89.1
LegoHouse.3frame3.C	3	50.0°	118	1.3	81.2

nonoverlapping subsequences from LegoHouse1. The three frames are selected so that 1) the minimum required number of line segments in the scene is available and 2) the total motion is enough to allow accurate estimation (at least 20°).

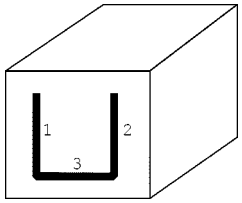
C. Ground Truth

The ground truth is manually defined in terms of motion and structure. The motion of the object is described by the rotation axis and the rotation angle between the frames. Image sequences are captured by rotating an object on a rotation stage by a predetermined angle between frames. These angles correspond to the GT rotation angles [refer to Fig. 5(a)]. In order to determine the GT rotation axis of the stage in camera coordinates, a cube is placed on the calibrated rotating stage so that the straight edge of the cube is aligned with the rotation axis. Intensity and range images are taken using a K^{2}T structured-light range sensor. The three–dimensional (3-D) rotation axis is computed by

1) picking two points defining the endpoints of the rotation;
2) getting the 3-D locations of two points using the range image;
3) normalizing the vector defined by two points. The structure GT is defined by a set of pairs of lines on the object and the angle between them [refer to Fig. 5(b)].



(a)



(b)

Fig. 5. Motion and structure ground truth.

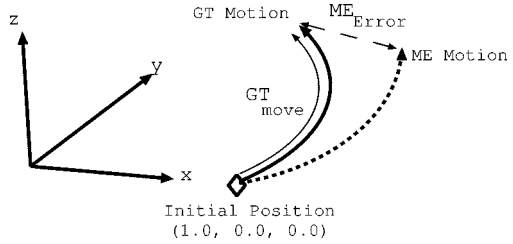


Fig. 6. Motion error measurement.

For instance, lines #1 and #3 in Fig. 5 are a pair making a 90° angle.

D. Performance Metrics

The machine estimated (ME) result is compared with GT in motion and structure. The ME motion of the camera which SFM produces is converted to the motion of the object by reversing the sign of the rotation angle while keeping the same rotation axis. Two measurements for the motion (rotation axis and rotation angle) are combined by the following method (refer to Fig. 6). First, an arbitrary point at $(1, 0, 0)$ is set for P_{GT_0} and P_{ME_0}. For each camera orientation j, P_{GT_j} is computed by moving P_{GT_0} with $Angle_{GT_j}$ and $Axis_{GT_j}$ while P_{ME_j} is computed with $Angle_{ME_j}$ and $Axis_{ME_j}$. Then the motion error is computed by ($MotionError = ME_{error}/GT_{move} \cdot 100\%$), where ME_{error} is the distance traveled from P_{GT_j} to P_{ME_j}, and GT_{move} is the distance between P_{GT_0} and P_{GT_j}. The structure error is measured by computing the absolute angle difference between a ME angle and its corresponding GT angle.

E. Parameter Training

This section describes the automated method of training the parameters of the edge detectors. The goal is to objectively and

automatically find the parameter setting of the edge detector that yields the best SFM performance.

The evaluation of detectors depends on parameters of the edge detector (up to three parameters), line extraction (three parameters), and line correspondence (two parameters). Finding the best setting of up to eight parameters is computationally infeasible. Therefore, the following method was established.

First, good parameter settings for line extraction and line correspondence are found after observing many runs of the experiment: $T_{point\_to\_line} = 5.0$ pixels, $T_{high\_curv\_angle} = 90.0°$, $T_{line\_length} = 50.0$ pixels, $T_{perp\_dist} = 5.0$ pixels, and $T_{overlap} = 80.0\%$. These values are fixed for all experiments with all detectors.

An adaptive method is used to search for the best edge detector parameter values. A $5 \times 5 \times \cdots \times 5$ initial uniform sampling of parameter range is tested. The area around the best parameter point in this coarse sampling is further subsampled at $3 \times 3 \times \cdots \times 3$ with the previous best at the center. A minimum of two subsamplings is executed, resulting in a minimum of $(5\times5\times5)+(3\times3\times3-1)+(3\times3\times3-1) = 177$ different parameter points for three-parameter edge detectors. Subsampling is continued while there is a 5% or greater improvement from the previous best. The parameters are trained separately for structure and motion. In our experiment, the best parameter setting was found after an average of 3.46 subsamplings and a maximum of six subsamplings.

The adaptive search is not *guaranteed* to find the globally best parameter setting. In fact, in some instances, better motion performance was observed with the structure-trained parameters than with the motion-trained parameters, or vice versa. However, only 18 such occurrences (eight for the Anisotropic, one for the Bergholm, three for the Canny, one for the Heitger, one for the Rothwell, one for the Sarkar and three for the SUSAN) were found during 252 (18 sequences $\times$ seven edge detectors $\times$ two error metrics) trainings. In addition, the differences were minimal. The mean difference was 2.12% (motion) and 1.06° (structure). Eleven of 18 instances are from the one-parameter edge detectors (the anisotropic detector and the SUSAN detector). The maximum motion difference of 8.34% was observed by the SUSAN detector trained for WoodBlock #2.B sequence with the motion error of 18.04% (motion-trained) and 9.70% (structure-trained). The maximum structure difference of 3.06° was observed by the Anisotropic detector trained for LegoHouse #3.B sequence with the structure error of 2.38° (motion-trained) and 5.44° (structure-trained).

To further check the effectiveness of the adaptive search, we made one comparison against an exhaustive search. Since the average number of subsamplings observed was 3.46, we have taken the exhaustive search to a similar resolution by parameter sampling of $20 \times 20 \times 20$. We have trained the Canny detector on WoodBlock #1.A. The best motion performance found by the exhaustive search was 3.80% ($\sigma = 0.272\,63$, *low* $= 0.631\,58$, *high* $= 0.947\,37$). Compared to the best of adaptive search (3.82%), the difference (0.02%) was minor. In fact, it was better than the fourth best found by the exhaustive search (3.90%). For structure, the best exhaustive performance was 0.310° ($\sigma = 1.323\,16$, *low* $= 0.894\,74$, *high* $= 0.947\,37$) compared to 0.920° of adaptive search (which was better than 25th best found by the exhaustive search), the difference was 0.610°.

IV. Results

The results are divided into three major sections:

1) long sequence experiment;
2) three-frame sequence experiment;
3) effect of line characteristics on SFM convergence.

In the long sequence experiment, the Taylor algorithm has been used as the task. The section is subdivided into

1) train;
2) test within the same scene type;
3) test across the scene types.

In the three-frame sequence experiment, the Taylor and Weng algorithms have been used as the task. The goal of the three-frame section is to examine 1) the effect of linear and nonlinear nature of SFM algorithms and 2) the effect of using minimum number of frames required in a sequence. The section is divided into 1) train and 2) test within the same scene. The effect of line characteristics on SFM convergence section analyzes how characteristics of edges have influenced the performance. It is divided into

1) line input characteristics;
2) SFM setting;
3) line characteristic analysis for the Sobel detector.

Notice that detailed results of the Sobel are absent. This is because the SFM algorithm could not converge with any of the Sobel's edge output of 5×5 parameter settings with WoodBlock #1, #2, and #3 sequences. We lowered the T_{line_length} threshold to 25 pixels, which should result in more lines and correspondences, and the Sobel still did not converge. We lowered T_{line_length} even further to 15 pixels, and got convergence only with the WoodBlock #1 sequence. The trained motion error was 4.92, which is poor compared to the other detectors. An investigation of the reasons for the poor performance of the Sobel detector is given in Section IV-C.

A. Long Sequences

In this section, 18 long sequences are tested on seven detectors. First, the "train" section describes the results of training the parameters of edge detectors for each sequence. Second, the "test within the same scene type" section takes the trained parameters and tests on the sequences within the same scene type. Third, the "test across scene type" section tests the trained parameters of a sequence of one scene type on the sequences of other scene type. All the results are divided into four categories of:

1) LH-motion;
2) LH-structure;
3) WB-motion;
4) WB-structure.

Each section contains a table containing mean of error and convergence rate (Tables VI, VIII, and XVII) and a table of relative ranking (Tables VII, XVI, and XVIII). The mean error is computed from all converging training or testing sequences within each category. Since the nonconvergence of the SFM algorithm occurs during training and testing, the convergence

TABLE VI
TRAIN RESULTS AND THE CONVERGENCE RATE OF THE "ALL" SEQUENCES WHERE FIRST NUMBER IS MEAN ERROR. CONVERGENCE RATE IS SPECIFIED BY NUMBER OF CONVERGING SEQUENCES/NUMBER OF SEQUENCE IN THE SCENE TYPE. BEST IN EACH CATEGORY IS HIGHLIGHTED IN BOLDFACE. MOTION UNIT IN "DISTANCE % DIFFERENCE" AND STRUCTURE UNIT IN "ANGLE DIFFERENCE IN DEGREES"

edge detector	motion (LH)	motion (WB)	structure (LH)	structure (WB)
Anisotropic	5.14 (6/9)	4.31 (9/9)	2.36 (6/9)	1.38 (9/9)
Bergholm	5.08 (9/9)	**3.71** (9/9)	1.19 (9/9)	**0.71** (9/9)
Canny	**5.04** (9/9)	3.82 (9/9)	1.48 (9/9)	1.04 (9/9)
Heitger	9.01 (9/9)	4.04 (9/9)	2.60 (9/9)	1.12 (9/9)
Rothwell	5.52 (9/9)	3.87 (9/9)	**1.10** (9/9)	0.77 (9/9)
Sarkar	7.90 (9/9)	4.19 (9/9)	4.49 (9/9)	1.54 (9/9)
SUSAN	18.07 (7/9)	6.65 (9/9)	5.82 (9/9)	2.78 (9/9)
mean	7.97	4.37	2.72	1.33

TABLE VII
RELATIVE TRAINING PERFORMANCE WHERE EACH CELL INDICATES a/b WHERE THE EDGE DETECTOR IN THE ROW PERFORMED "a" INCIDENCES BETTER THAN THE EDGE DETECTOR IN THE COLUMN OUT OF "b" TIMES AND "*" HAS BEEN PLACED IN THE CASES OF THE EDGE DETECTOR IN THE ROW "SIGNIFICANTLY OUTPERFORMING" THE EDGE DETECTOR IN THE COLUMN

LH - motion

	aniso	berg	canny	heitger	roth	sarkar	susan
aniso		0/6	0/6	1/6	3/6	4/6 *	4/5 *
berg	6/6 *		6/9 *	8/9 *	8/9 *	9/9 *	7/7 *
canny	6/6 *	3/9		9/9 *	9/9 *	9/9 *	7/7 *
heitger	5/6 *	1/9	0/9		4/9	7/9 *	6/7 *
roth	3/6	1/9	0/9	5/9		8/9 *	7/7 *
sarkar	2/6	0/9	0/9	2/9	1/9		5/7 *
susan	1/5	0/7	0/7	1/7	0/7	2/7	

LH - struct

aniso		0/6	0/6	0/6	2/6	5/6 *	4/4 *
berg	6/6 *		6/9 *	7/9 *	6/9 *	9/9 *	6/6 *
canny	6/6 *	3/9		6/9 *	3/9	8/9 *	5/6 *
heitger	6/6 *	2/9	3/9		3/9	7/9 *	5/6 *
roth	4/6 *	3/9	6/9 *	6/9 *		9/9 *	6/6 *
sarkar	1/6	0/9	1/9	2/9	0/9		4/6 *
susan	0/4	0/6	1/6	1/6	0/6	2/6	

WB - motion

aniso		2/9	2/9	4/9	4/9	4/9	6/9 *
berg	7/9 *		7/9 *	7/9 *	6/9 *	8/9 *	8/9 *
canny	7/9 *	2/9		6/9 *	6/9 *	7/9 *	8/9 *
heitger	5/9	2/9	3/9		5/9	5/9	8/9 *
roth	5/9	3/9	3/9	4/9		5/9	6/9 *
sarkar	5/9	1/9	2/9	4/9	4/9		6/9 *
susan	3/9	1/9	1/9	1/9	3/9	3/9	

WB - struct

aniso		1/9	2/9	3/9	2/9	5/9	7/9 *
berg	8/9 *		7/9 *	7/9 *	4/9	9/9 *	8/9 *
canny	7/9 *	2/9		4/9	2/9	8/9 *	7/9 *
heitger	6/9 *	2/9	5/9		2/9	7/9 *	8/9 *
roth	7/9 *	5/9	7/9 *	7/9 *		7/9 *	9/9 *
sarkar	4/9	0/9	1/9	2/9	2/9		4/9
susan	2/9	1/9	2/9	1/9	0/9	5/9	

is shown in (x, y) format where "x" is the number of convergence and "y" is the number of attempts. The table of relative ranking is shown to examine the statistical significance of training and testing. Each table consists of four subtables corresponding to four categories. Each subtable is a $seven \times seven$ matrix holding "(a/b)" in each cell. "b" is the number of sequences that both detectors in the row and the column converged. "a" is the number of times the detector in the row outperformed the detector in the column. "$*$" is added in the cell where the detector in the row "significantly outperformed" the detector in the column, one is better than other two-thirds or more of the times. If the entire row contains "$*$," it indicates that the detector in the row significantly outperformed all other detectors. If the entire column contains "$*$," it indicates that all other detectors outperformed the detector in the column.

TABLE VIII
TEST WITHIN THE SAME SCENE RESULTS AND THE CONVERGENCE RATE WHERE BEST IN EACH CATEGORY IS HIGHLIGHTED IN BOLDFACE, MOTION UNIT IN "DISTANCE % DIFFERENCE," AND STRUCTURE UNIT IN "ANGLE DIFFERENCE IN DEGREES"

edge detector	motion (LH)	motion (WB)	structure (LH)	structure (WB)	converge rate
anisotropic	8.12 (30/72)	6.15 (72/72)	**3.57** (27/72)	4.08 (54/72)	63.5%
bergholm	7.86 (63/72)	7.13 (72/72)	4.87 (65/72)	6.30 (72/72)	94.4%
canny	**7.14** (70/72)	**4.77** (72/72)	4.28 (71/72)	**2.84** (71/72)	**98.6%**
heitger	8.82 (60/72)	4.90 (72/72)	5.84 (59/72)	4.36 (70/72)	90.6%
rothwell	14.96 (57/72)	7.58 (72/72)	7.07 (53/72)	6.16 (66/72)	86.1%
sarkar	16.61 (63/72)	5.65 (72/72)	6.17 (63/72)	5.13 (69/72)	92.7%
susan	20.34 (39/72)	9.06 (30/30)	8.30 (32/72)	6.69 (66/72)	58.0%
converge rate	75.8%	91.7%	73.4%	92.9%	

1) Train: First, the training results show that in all 28 instances (four categories $\times$ seven edge detectors $=$ 28), each edge detector performed better in WoodBlock scenes than LegoHouse scenes, indicating that the LegoHouse scenes are harder. Second, the Bergholm detector performed the best in two categories of WB-motion and WB-structure (refer to Table VI). The Canny detector and the Rothwell detector shared second place by each having the best performance in one category. The Anisotropic, the Heitger, and the Sarkar detectors ranked closely as fourth. The SUSAN detector was last in all four categories. Third, referring to the relative ranking (Table VII), the Bergholm detector was significantly better than other detectors 23 out of 24 times (six other detectors $\times$ four categories $=$ 24) while the SUSAN detector was significantly outperformed by all other detectors nearly all times (23 out of 24). Though the mean error might show that the Sarkar detector's performance was similar to the Rothwell detector, Table VI shows that the Rothwell detector significantly outperformed the Sarkar detector three out of four categories.

As can be seen in Figs. 7 and 8, "false positive" edges do not seem to play a significant role since they were usually short edge chains that were eliminated during the short line removal step. Also, a general trend of more and longer lines giving better results seems visually verified in Fig. 7.

In Tables IX–XV, the trained parameter settings in 18 sequences are shown. Note that the parameter settings are sparsely distributed within the parameter space.

2) Test within the Same Scene Type: The parameters selected by training for one sequence were tested on other sequences within the same scene using "leave one out" type of testing. Each sequence is tested on all other sequences of the same type, for motion and structure separately. Therefore, for each edge detector we had 72 tests (nine sequences $\times$ eight trained parameter settings from other sequences) for four categories (motion and structure, and LegoHouse and WoodBlock).

The test edge maps sometimes resulted in a set of corresponded lines that the SFM algorithm could *not converge* into any solution even after 1750 iterations. In order to take this

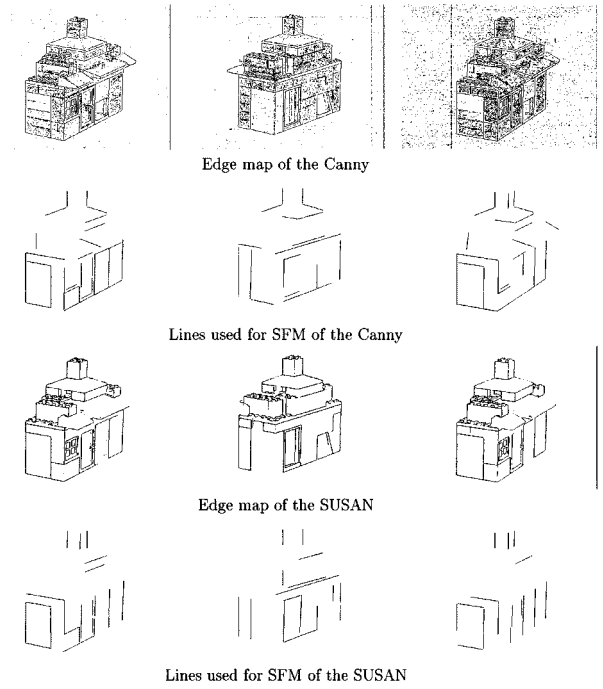


Fig. 7. Dataset where the least motion error difference between a pair of edge detector is shown: LegoHouse #2.A. Canny detector ($\sigma = 0.01$, $low = 1.0$, $high = 0.812$) with 6.82 and the SUSAN ($T = 12.5$) with 68.43.

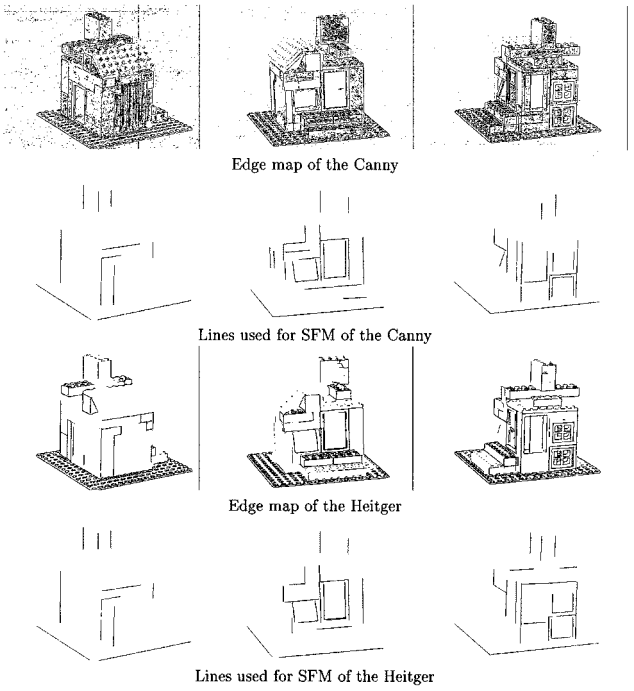


Fig. 8. Dataset where the least motion error difference between a pair of edge detector is shown: LegoHouse #3.A. Canny detector ($\sigma = 0.321\,88$, $low = 0.968\,75$, $high = 0.781\,25$) with 6.002. The Heitger ($\sigma = 1.906\,250$, $T = 9.375$) with 6.005.

problem into consideration, 1) the convergence rates are compared and 2) the test results of all converging trials of each edge detector are presented.

The Canny detector showed the best convergence rate at 98.6%, while the SUSAN detector had the worst at 58.0% (refer to Table VIII). Lower convergence rates were observed for LegoHouse scenes (75.8% for motion and 73.4% for structure) compared to WoodBlock scenes (91.7% for motion and 92.9% for structure). This suggests that the LegoHouse scene type is more difficult than the WoodBlock, as was also shown in train results.

The Canny detector performed the best in three test categories: motion-LH, motion-WB, and structure-WB (refer to Table VIII). The Anisotropic detector placed first in structure-LH. As shown in the training performance, the SUSAN detector performed the worst in all four categories. The Bergholm detector, which showed the best train performance, was behind the Heitger detector. The Canny detector was never outperformed (significantly) by any other edge detector. In fact, the Canny detector outperformed the Rothwell detector and the SUSAN detector in all four categories. The Heitger

TABLE IX
TRAINED PARAMETER SETTINGS FOR THE ANISOTROPIC EDGE DETECTOR WHERE n/a INDICATES THAT NONE OF PARAMETER SETTINGS RESULTED IN SFM CONVERGENCE

Image Sequence	Motion-trained	Structure-trained
LH1	1.25000	2.50000
LH1.A	1.87500	1.87500
LH1.B	3.75000	3.75000
LH2	n/a	n/a
LH2.A	n/a	n/a
LH2.B	n/a	n/a
LH3	1.87500	1.87500
LH3.A	2.50000	2.50000
LH3.B	1.87500	1.87500
WB1	6.87500	6.87500
WB1.A	7.50000	7.50000
WB1.B	7.50000	7.50000
WB2	2.50000	2.50000
WB2.A	3.12500	3.12500
WB2.B	6.25000	6.25000
WB3	5.00000	5.00000
WB3.A	1.71875	1.71875
WB3.B	1.56250	1.56250

TABLE X
TRAINED PARAMETER SETTINGS FOR THE BERGHOLM EDGE DETECTOR

Image Sequence	Motion-trained	Structure-trained
LH1	4.69281, 0.01000, 11.25000	9.37562, 0.01000, 5.62500
LH1.A	3.75625, 0.01000, 22.50000	5.00500, 0.01000, 60.00000
LH1.B	5.00500, 2.50750, 15.00000	8.75125, 1.25875, 7.50000
LH2	6.25375, 0.63437, 3.75000	6.25375, 2.50750, 3.75000
LH2.A	8.12688, 3.13187, 0.00000	8.43906, 3.13187, 1.87500
LH2.B	3.75625, 2.50750, 3.75000	8.75125, 1.25875, 7.50000
LH3	6.25375, 2.50750, 7.50000	9.37562, 4.38062, 5.62500
LH3.A	7.50250, 2.50750, 45.00000	10.62438, 0.01000, 33.75000
LH3.B	8.75125, 2.19531, 5.62500	7.58055, 0.01000, 11.71875
WB1	10.00000, 3.75625, 15.00000	10.00000, 5.00500, 60.00000
WB1.A	6.87812, 3.75625, 11.25000	7.50250, 1.25875, 67.50000
WB1.B	11.24875, 0.01000, 3.75000	7.50250, 5.00500, 60.00000
WB2	6.87812, 5.62937, 56.25000	4.38063, 3.13187, 3.75000
WB2.A	3.75625, 2.50750, 7.50000	8.75125, 3.13187, 13.12500
WB2.B	5.00500, 1.88312, 45.00000	9.37562, 2.50750, 13.12500
WB3	11.24875, 2.50750, 22.50000	10.00000, 3.75625, 22.50000
WB3.A	10.00000, 2.50750, 15.00000	8.75125, 0.01000, 22.50000
WB3.B	10.00000, 7.50250, 0.00000	9.37562, 2.50750, 56.25000

TABLE XI
TRAINED PARAMETER SETTINGS FOR THE CANNY EDGE DETECTOR

Image Sequence	Motion-trained	Structure-trained
LH1	0.63375, 0.37500, 0.87500	1.88125, 0.75000, 0.75000
LH1.A	0.32188, 0.81250, 0.87500	1.88125, 0.37500, 0.81250
LH1.B	1.88125, 0.00000, 0.62500	2.50500, 0.50000, 0.75000
LH2	0.01000, 1.00000, 0.81250	0.01000, 0.75000, 0.50000
LH2.A	0.01000, 1.00000, 0.81250	4.37625, 0.00000, 0.00000
LH2.B	0.01000, 0.00000, 0.00000	1.56937, 0.00000, 0.00000
LH3	0.32188, 0.93750, 0.68750	3.75250, 0.00000, 0.00000
LH3.A	0.32188, 0.96875, 0.78125	0.32188, 0.68750, 0.78125
LH3.B	5.31187, 0.00000, 0.00000	3.75250, 0.00000, 0.00000
WB1	0.01000, 0.50000, 0.50000	0.78969, 0.51562, 0.89062
WB1.A	0.63375, 0.12500, 0.62500	0.01000, 0.50000, 0.50000
WB1.B	0.01000, 0.00000, 0.00000	0.01000, 0.00000, 0.00000
WB2	1.56937, 0.00000, 0.00000	0.94563, 1.00000, 0.93750
WB2.A	1.25750, 0.00000, 0.00000	2.50500, 0.00000, 0.00000
WB2.B	1.56937, 0.00000, 0.00000	0.63375, 0.00000, 0.00000
WB3	1.88125, 0.00000, 0.00000	3.28469, 0.00000, 0.00000
WB3.A	0.01000, 0.50000, 0.50000	0.94563, 0.00000, 0.00000
WB3.B	5.00000, 0.00000, 0.00000	0.01000, 0.00000, 0.00000

TABLE XII
TRAINED PARAMETER SETTINGS FOR THE HEITGER EDGE DETECTOR

Image Sequence	Motion-trained	Structure-trained
LH1	2.46875, 3.12500	3.87500, 25.00000
LH1.A	3.87500, 3.12500	1.62500, 6.25000
LH1.B	4.01562, 6.25000	4.01562, 6.25000
LH2	3.87500, 12.50000	3.87500, 12.50000
LH2.A	5.35156, 3.90625	5.35156, 3.90625
LH2.B	3.87500, 37.50000	3.87500, 37.50000
LH3	2.75000, 12.50000	2.75000, 12.50000
LH3.A	1.90625, 9.37500	2.18750, 9.37500
LH3.B	3.03125, 6.25000	3.87500, 3.12500
WB1	3.87500, 3.12500	2.18750, 18.75000
WB1.A	2.75000, 3.12500	2.18750, 31.25000
WB1.B	3.87500, 6.25000	3.03125, 12.50000
WB2	3.03125, 9.37500	2.75000, 50.00000
WB2.A	2.75000, 12.50000	2.89062, 35.93750
WB2.B	2.46875, 9.37500	1.62500, 12.50000
WB3	4.15625, 6.25000	1.34375, 9.37500
WB3.A	4.01562, 9.37500	1.62500, 6.25000
WB3.B	1.69531, 2.34375	0.78125, 15.62500

TABLE XIII
TRAINED PARAMETER SETTINGS FOR THE ROTHWELL EDGE DETECTOR

Image Sequence	Motion-trained	Structure-trained
LH1	1.37500, 0.00000, 0.81250	0.93750, 3.75000, 0.93750
LH1.A	0.93750, 3.75000, 0.81250	0.50000, 15.00000, 0.50000
LH1.B	0.93750, 3.75000, 0.81250	0.28125, 24.37500, 0.71875
LH2	1.15625, 3.75000, 0.93750	0.66406, 5.62500, 0.29688
LH2.A	0.50000, 7.50000, 0.12500	0.93750, 3.75000, 0.43750
LH2.B	1.15625, 0.00000, 0.93750	0.50000, 30.00000, 0.50000
LH3	0.71875, 5.62500, 0.90625	0.50000, 0.00000, 0.75000
LH3.A	1.81250, 15.00000, 0.56250	0.93750, 7.50000, 0.62500
LH3.B	0.33594, 7.96875, 0.98438	0.50000, 15.00000, 0.18750
WB1	1.59375, 15.00000, 0.81250	1.37500, 30.00000, 0.00000
WB1.A	0.71875, 26.25000, 0.00000	1.59375, 0.00000, 0.56250
WB1.B	1.59375, 15.00000, 0.53125	3.12500, 0.00000, 0.31250
WB2	1.15625, 11.25000, 0.68750	0.50000, 45.00000, 0.50000
WB2.A	0.71875, 11.25000, 0.93750	0.39062, 43.12500, 0.00000
WB2.B	0.93750, 22.50000, 0.00000	1.81250, 0.00000, 0.50000
WB3	1.15625, 3.75000, 0.75000	0.71875, 3.75000, 0.68750
WB3.A	0.93750, 7.50000, 0.00000	0.71875, 15.00000, 0.00000
WB3.B	0.93750, 7.50000, 0.37500	1.26562, 2.81250, 0.57812

TABLE XIV
TRAINED PARAMETER SETTINGS FOR THE SARKAR EDGE DETECTOR

Image Sequence	Motion-trained	Structure-trained
LH1	0.63375, 0.00000, 0.75000	0.94563, 0.06250, 0.81250
LH1.A	1.10156, 0.06250, 0.15625	0.94563, 0.43750, 0.50000
LH1.B	0.63375, 0.37500, 0.50000	0.63375, 0.25000, 0.37500
LH2	1.2575, 0.00000, 0.25000	1.25750, 0.00000, 0.25000
LH2.A	1.25750, 0.00000, 0.25000	0.63375, 0.00000, 0.12500
LH2.B	3.12875, 0.37500, 0.50000	2.50500, 0.00000, 0.25000
LH3	1.88125, 0.31250, 0.37500	1.88125, 0.31250, 0.37500
LH3.A	1.56937, 0.31250, 0.43750	1.88125, 0.50000, 0.62500
LH3.B	1.56937, 0.06250, 0.31250	2.19312, 0.12500, 0.31250
WB1	1.88125, 0.00000, 0.12500	2.19313, 0.00000, 0.06250
WB1.A	0.94563, 0.43750, 0.68750	0.32188, 0.00000, 0.06250
WB1.B	1.88125, 0.00000, 0.12500	1.25750, 0.31250, 0.43750
WB2	0.63375, 0.12500, 0.37500	1.88125, 0.25000, 0.37500
WB2.A	0.63375, 0.00000, 0.12500	0.63375, 0.12500, 0.25000
WB2.B	0.94563, 0.56250, 0.68750	2.50500, 0.00000, 0.25000
WB3	0.94563, 0.25000, 0.31250	0.78969, 0.03125, 0.15625
WB3.A	2.19312, 0.00000, 0.06250	0.94563, 0.18750, 0.43750
WB3.B	0.94563, 0.43750, 0.68750	2.03719, 0.00000, 0.15625

detector significantly outperformed the SUSAN detector in all four categories.

3) Test Across Scene Type: We also tested the parameters trained for all nine WoodBlock sequences on LegoHouse #1 and the parameters trained for all LegoHouse sequences on WoodBlock #1. The results are organized in a similar fashion. There are four categories:

1) tested on LegoHouse motion;
2) tested on LegoHouse structure;
3) tested on WoodBlock motion;
4) tested on WoodBlock structure.

We present the results of "all" converging sequences.

The Canny detector performed the best in two categories (motion-LH1 and motion-WB1) and had the highest convergence rate of 100%. Those two observations are consistent with the observations from testing within scene type. We conclude that the Canny detector has the "best motion testing performance" with the most robust convergence. The Anisotropic, Bergholm, and Heitger detectors performed similarly, ranking behind the Canny detector.

The Canny detector significantly outperformed others the most number of times (16 out of 24). The Anisotropic detector outperformed all other detectors in LH-structure category. The SUSAN detector was significantly outperformed by others 23 out of 24 times.

B. Three-Frame Sequences

This section describes the results of using two different SFM algorithms on three-frame sequences. Due to the difficulty of convergence for the Taylor algorithm on three-frame sequences, the adaptive parameter setting started with $10 \times 10 \times \cdots \times 10$ parameter sampling. The results are using nine LegoHouse three-frame sequences (Table V). All other experiment methods on train, test, and other analysis are identical to the procedure followed in long sequence section.

1) Train: The train results of using three-frame sequences are shown in Table XIX. First, note the great performance

TABLE XV
TRAINED PARAMETER SETTINGS FOR THE SUSAN EDGE DETECTOR WHERE n/a INDICATES THAT NONE OF PARAMETER SETTINGS RESULTED IN SFM CONVERGENCE

Image Sequence	Motion-trained	Structure-trained
LH1	15.62500	18.75000
LH1.A	n/a	n/a
LH1.B	6.25000	6.25000
LH2	n/a	n/a
LH2.A	12.50000	12.50000
LH2.B	6.25000	4.68750
LH3	12.50000	12.50000
LH3.A	17.18750	9.37500
LH3.B	9.37500	n/a
WB1	25.00000	25.00000
WB1.A	12.50000	25.00000
WB1.B	12.50000	43.75000
WB2	25.00000	20.31250
WB2.A	25.00000	18.75000
WB2.B	21.87500	18.75000
WB3	10.93750	15.62500
WB3.A	12.50000	12.50000
WB3.B	18.75000	9.37500

TABLE XVI
RELATIVE TESTING WITHIN THE SAME SCENE PERFORMANCE

LH - motion

	aniso	berg	canny	heitger	roth	sarkar	susan
aniso		16/29	17/29	19/30	18/26 *	22/30 *	18/22 *
berg	13/29		22/61	33/53	39/53 *	42/56 *	32/37 *
canny	12/29	39/61		43/58 *	43/55 *	51/61 *	35/38 *
heitger	11/31	20/53	15/58		32/49	39/55 *	30/34 *
roth	8/26	14/53	12/55	17/49		27/49	23/33 *
sarkar	8/30	14/56	10/61	16/55	22/49		23/36
susan	4/22	5/37	3/38	4/34	10/33	13/36	

LH - struct

aniso		11/26	13/27	9/27	19/26 *	17/26	9/12 *
berg	15/26		30/64	30/55	34/50 *	40/58 *	26/32 *
canny	14/27	34/64		35/58	38/53 *	48/62 *	23/32 *
heitger	18/27 *	25/55	23/58		32/46 *	29/53	17/26
roth	7/26	16/50	15/53	14/46		25/49	8/15
sarkar	9/26	18/58	14/62	24/53	24/49		13/25
susan	3/12	6/32	9/32	9/26	12/23	11/29	

WB - motion

aniso		37/72	23/72	26/72	45/72	30/72	25/30 *
berg	35/72		18/72	24/72	41/72	28/72	16/30
canny	49/72 *	54/72 *		37/72	56/72 *	48/72 *	25/30 *
heitger	46/72	48/72 *	35/72		58/72 *	46/72	24/30 *
roth	27/72	31/72	16/72	14/72		27/72	19/30
sarkar	42/72	44/72	24/72	26/72	45/72		24/30 *
susan	5/30	14/30	5/30	6/30	11/30	6/30	

WB - struct

aniso		31/54	21/53	27/52	36/48 *	31/51	32/50
berg	23/54		18/71	22/70	35/66	37/69	41/66
canny	32/53	53/71 *		42/69	48/65 *	51/68	55/65 *
heitger	25/52	48/70 *	27/69		45/64 *	36/67	48/64 *
roth	12/48	31/66	17/65	19/64		32/63	35/61
sarkar	20/51	32/69	17/68	31/67	31/63		39/63
susan	18/50	25/66	10/65	16/64	26/61	24/63	

degradation between long sequences and three-frame sequences. For the Sarkar detector, the error with three-frame sequences was nearly five times greater. Second, given the three-frame sequences, the edge detectors performed better with the Taylor algorithm than with the Weng algorithm. Third, the relative ranking among results 1) using Taylor algorithm with three-frame sequences and 2) using Taylor algorithm with long sequences, are the same. In addition, for the relative ranking among all three (including Weng's three-frame), the Canny detector was the first or the second in all three, the Sarkar detector was the worst in all three, and the Canny detector and the Bergholm detector were both close in ranking.

TABLE XVII
TEST ACROSS SCENE OF "ALL" SEQUENCES RESULTS AND THE CONVERGENCE RATE WHERE THE RESULTS UNDER EACH COLUMN INDICATE THE PERFORMANCE WHICH WAS "TESTING ON" THE DATASET AND CONVERGENCE IS SHOWN BY "# OF CONVERGING SEQUENCES/# OF SEQUENCE IN THE SCENE TYPE," AND THE BEST IN EACH CATEGORY IS HIGHLIGHTED IN BOLDFACE, MOTION UNIT IN "DISTANCE % DIFFERENCE," STRUCTURE UNIT IN "ANGLE DIFFERENCE IN DEGREES"

edge detector	motion (LH1)	motion (WB1)	struct (LH1)	struct (WB1)	converge rate
anisotropic	2.30 (9/9)	5.95 (6/9)	2.39 (8/9)	**2.30** (9/9)	81%
bergholm	4.18 (8/9)	6.34 (9/9)	**1.43** (6/9)	6.15 (8/9)	89%
canny	**1.86** (9/9)	**5.59** (9/9)	2.11 (9/9)	3.82 (9/9)	**100%**
heitger	1.88 (9/9)	6.10 (9/9)	1.96 (7/9)	2.82 (9/9)	92%
rothwell	6.68 (8/9)	7.40 (8/9)	6.32 (5/9)	9.78 (8/9)	81%
sarkar	3.82 (9/9)	5.75 (9/9)	2.44 (7/9)	3.77 (9/9)	94%
susan	6.53 (5/9)	5.95 (3/9)	2.98 (3/9)	24.84 (5/9)	36%
mean	3.89	6.15	2.80	7.64	82

TABLE XVIII
RELATIVE TESTING ACROSS THE SCENE TYPE PERFORMANCE WHERE EACH CELL INDICATES a/b WHERE THE EDGE DETECTOR IN THE ROW PERFORMED "a" INCIDENCES BETTER THAN THE EDGE DETECTOR IN THE COLUMN OUT OF "b" TIMES AND "*" HAS BEEN PLACED IN THE CASES OF THE EDGE DETECTOR IN THE ROW "SIGNIFICANTLY OUTPERFORMING" THE EDGE DETECTOR IN THE COLUMN

LH - motion

	aniso	berg	canny	heitger	roth	sarkar	susan
aniso		5/6 *	1/6	3/6	1/6	2/6	2/3 *
berg	1/6		3/9	3/9	3/8	3/9	2/3 *
canny	5/6 *	6/9 *		5/9	6/8 *	3/9	3/3 *
heitger	3/6	6/9 *	4/9		5/8	4/9	3/3 *
roth	5/6 *	5/8	2/8	3/8		2/8	3/3 *
sarkar	4/6 *	6/9 *	6/9 *	5/9	6/8 *		3/3 *
susan	1/3	1/3	0/3	0/3	0/3	0/3	

LH - struct

aniso		6/6 *	4/6 *	5/6 *	5/6 *	4/6 *	1/1 *
berg	0/6		1/9	1/9	5/8	2/9	2/2 *
canny	2/6	8/9 *		4/9	7/8 *	6/9 *	2/2 *
heitger	1/6	8/9 *	5/9		6/8 *	5/9	2/2 *
roth	1/6	3/8	1/8	2/8		2/8	2/2 *
sarkar	2/6	7/9 *	3/9	4/9	6/8 *		2/2 *
susan	0/1	0/2	0/2	0/2	0/2	0/2	

WB - motion

aniso		5/8	2/9	4/9	8/8 *	7/9 *	5/5 *
berg	3/8		1/8	0/8	5/7 *	5/8	4/5 *
canny	7/9 *	7/8 *		4/9	8/8 *	9/9 *	5/5 *
heitger	5/9	8/8 *	5/9		8/8 *	9/9 *	5/5 *
roth	0/8	2/7	0/8	0/8		2/8	3/5
sarkar	2/9	3/8	0/9	0/9	6/8 *		5/5 *
susan	0/5	1/5	0/5	0/5	2/5	0/5	

WB - struct

aniso		3/5	4/8	2/7	3/5	2/6	3/3 *
berg	2/5		3/6	4/5 *	3/4 *	4/5 *	1/1 *
canny	4/8	3/6		5/7 *	4/5 *	4/7	2/3 *
heitger	5/7 *	1/5	2/7		4/4 *	3/6	2/2 *
roth	2/5	1/4	1/5	0/4		0/3	1/1 *
sarkar	4/6	1/5	3/7	3/6	3/3 *		2/2 *
susan	0/3	0/1	1/3	0/2	0/1	0/2	

TABLE XIX
MOTION TRAINED RESULTS OF THE THREE-FRAME SEQUENCES WITH MOTION UNIT IN "DISTANCE % DIFFERENCE"

edge detector	Weng (3-frame)	Taylor (3-frame)	Taylor (long sequences) (result from all LH sequences)
bergholm	72.5	7.0	4.5
canny	71.7	11.4	4.6
rothwell	51.2	21.7	4.9
sarkar	86.2	49.8	5.5

TABLE XX
MOTION TEST RESULTS OF THE THREE-FRAME SEQUENCES WITH MOTION UNIT IN "DISTANCE % DIFFERENCE"

edge detector	Weng (3-frame)	Taylor (3-frame)	Taylor (long sequences) (result from all LH sequences)
bergholm	87.7 (16.7%)	67.6 (43.1%)	7.86 (87.5%)
canny	67.9 (43.1%)	59.0 (29.2%)	7.41 (97.2%)
rothwell	95.6 (29.2%)	106.0 (27.8%)	14.96 (79.2%)
sarkar	76.3 (31.9%)	151.0 (45.8%)	16.61 (87.5%)

2) Test: The test results are based on 72 test attempts (nine trained sequences $\times$ eight test sequences). First, all edge detectors suffered greatly on the convergence rate (refer to Table XX). All edge detectors went below 50%. Interestingly, the Canny detector which showed the highest convergence rate in long sequence study, plunged to nearly the worst convergence rate. Second, the Canny detector performed the best in all three categories, which again shows that the Canny detector has the best test results. Third, the relative test rankings among results using Taylor algorithm with long sequences and using Taylor algorithm with three-frame sequences, are the same. This indicates that the relative testing rankings are preserved with different length of the sequence. Fourth, compared to the results for long sequences, the edge detectors performed much (six to ten times) worse.

C. Effect of Line Characteristics on SFM Convergence

This section analyzes reasons behind the performance level of edge detectors for the task of SFM. Two factors could affect the SFM algorithm: the input for the SFM and the SFM setting. The line input is characterized by the number of lines, the number of correspondences for each line, and the length of the lines. The SFM algorithm's setting is the number of global iterations.

1) Line Input Characteristics: The average line characteristics of converging and nonconverging points are shown in Table XXI. Note that *in average* over all edge detectors, converging cases have more than twice as many lines as the nonconverging cases. In addition, higher numbers of correspondences and line lengths are observed with converging cases. The Bergholm detector has the least difference between convergences and nonconvergences. In fact, the average line length is actually higher with nonconvergences.

In order to examine the role of minimum criteria on nonconvergence, the points in nonconvergences are organized in four categories as shown in Table XXII. Obviously, all the converging cases met the minimum criteria. Since the line correspondence program deletes the lines with less than minimum correspondences, the category of $<$ *corr* and $<$ *line* is the cases where no lines were used as the input of the SFM algorithm. This does *not* indicate that the edge detectors did not detect any edges. It is possible that the lines that were formed from edges could not meet the minimum line length or the minimum number of correspondence criteria. Note the percentage of nonconvergences coming from *valid* SFM inputs. Except for the Anisotropic detector, all edge detectors have greater than 70% of nonconvergences from valid inputs. This raises several questions regarding the SFM algorithm. 1) *Practical* minimum criteria: Even though the theoretical minimum criteria are satisfied, the recovery of motion is not guaranteed. 2) *Practical* number of SFM global iterations: The current train setting of 50 global iterations might not be enough for some of the valid inputs.

TABLE XXI
EFFECT OF LINE CHARACTERISTICS ON CONVERGENCE

	LegoHouse						WoodBlock					
	converging			non-converging			converging			non-converging		
	# of line	# of corr	line length	# of line	# of corr	line length	# of line	# of corr	line length	# of line	# of corr	line length
Anisotropic	39.1	7.0	93.1	14.1	3.5	53.3	25.0	8.8	104.3	7.8	2.6	34.9
Bergholm	36.5	6.5	94.0	20.8	4.9	77.3	24.5	8.3	115.0	14.8	5.3	84.4
Canny	46.3	6.6	94.2	29.8	5.7	95.7	27.4	9.6	113.7	21.6	8.2	113.0
Heitger	38.3	6.4	98.1	24.1	5.7	97.1	23.7	8.4	117.4	15.1	6.2	103.8
Rothwell	37.6	6.5	95.8	18.0	5.5	100.9	21.7	8.0	116.2	11.8	6.1	108.7
Sarkar	38.8	6.0	107.2	19.7	4.5	102.8	25.1	9.0	121.6	14.4	6.1	92.2
SUSAN	36.3	6.3	91.4	18.8	4.8	87.7	22.6	7.9	110.6	104.	4.0	71.3
average	39.0	6.5	96.2	20.8	4.9	87.8	24.3	8.6	114.1	13.7	5.5	86.9

TABLE XXII
MEETING MINIMUM CRITERIA WITH NONCONVERGENCE (NUMBERS IN PERCENT)

	LegoHouse				WoodBlock			
	$<$ corr $<$ line	$<$ corr $\geq$ line	$\geq$ corr $<$ line	$\geq$ corr $\geq$ line	$<$ corr $<$ line	$<$ corr $\geq$ line	$\geq$ corr $<$ line	$\geq$ corr $\geq$ line
Anisotropic	46.5	0.0	0.0	53.5	66.7	0.0	0.0	33.3
Bergholm	21.1	0.0	0.0	78.9	26.9	0.0	0.0	73.1
Canny	0.0	22.0	0.0	78.0	0.0	9.6	0.0	90.4
Heitger	4.2	1.6	0.0	94.2	7.8	4.9	0.0	87.2
Rothwell	0.8	20.3	0.0	78.8	0.9	25.4	0.0	74.0
Sarkar	18.7	6.9	0.0	74.4	19.3	2.1	0.0	78.6
SUSAN	16.8	0.0	0.0	83.2	37.0	0.0	0.0	63.0

2) SFM Setting: There were instances where the algorithm was able to converge with more iterations. We examined three different aspects of whether 50 iterations for training and 1750 for testing are "good enough."

First, the feasibility of executing the algorithm for a large number of iterations needs to be examined. With 50 global iterations, the algorithm consumes up to 10 min of CPU time on a Sun Ultra Sparc. The training attempts on the average nearly 200 parameter settings on 18 image sequences for each edge detector in two different metrics (motion and structure), corresponding to $200 \times 18 \times 2 = 7200$ executions of the SFM algorithm and about 1200 h of CPU time. This could be multiplied by a factor of two for every 50 extra global iterations for every edge detector. Without tremendous computing power, increasing the global iterations would be infeasible.

Second, consider the average number of iterations used in cases that did converge. The average of SFM attempts during the train for all image sequences is computed for each edge detector:

1) Anisotropic (3.4);
2) Bergholm (3.5);
3) Canny (3.3);
4) Heitger (3.6);
5) Rothwell (3.6);
6) Sarkar (3.4);
7) Sobel (4.4).

TABLE XXIII
EFFECT OF LINE CHARACTERISTICS ON CONVERGENCE COMPARED WITH THE SOBEL

WoodBlock #1, #2, #3

	converging			non-converging		
	# of line	# of corr	line length	# of line	# of corr	line length
Anisotropic	21.5	8.4	114.1	7.3	2.7	37.7
Bergholm	26.5	10.1	114.9	27.1	9.9	115.4
Canny	29.1	12.0	113.2	14.9	6.6	90.8
Heitger	27.2	10.8	116.7	6.1	2.8	41.2
Rothwell	25.4	10.0	115.5	7.5	4.0	57.8
Sarkar	27.8	11.3	121.0	6.4	3.3	41.2
SUSAN	20.0	7.2	120.5	12.9	4.9	80.8
average	25.4	10.0	116.5	11.7	4.9	66.4
Sobel ($T_{line\_length} = 50$)				18.99	6.97	86.77
Sobel ($T_{line\_length} = 25$)				24.11	8.39	61.37

The average of all attempts for all edge detectors is 3.46. Note that the previous number is different from the average of all edge detectors' average, since the edge detectors have a different number of parameter attempts during the train. This indicates that if the SFM input is to converge, it will converge with very few iterations.

Third, the effect of a high number of iterations on the train and test performance is examined. We have tested the Bergholm detector, which tends to have the most overlapping between convergence and nonconvergence, by performing the adaptive parameter searching with 100 iterations on a sequence (WoodBlock1.A). It is verified that the best parameter setting did not change with 100 iterations; therefore, no change of train and test performance.

3) Line Characteristic Analysis for the Sobel Detector: As we have mentioned in the beginning of the results section, the Sobel did not converge into any solution during the training with three WoodBlock sequences. We will compare the line characteristics for the Sobel with other edge detectors.

First, the data from the only three image sequences that the Sobel detector was tested with (WoodBlock #1, #2, #3) are shown in Table XXIII. The characteristic of the nonconvergences of the Sobel falls between the mean of the convergences and nonconvergences of other five edge detectors. Even with the T_{line_length} lowered to 25, the line characteristic could not reach the mean convergences. Though the reason for the Sobel's performance cannot be totally explained by the line characteristics, it certainly strengthens the argument of more lines, longer lines, and more correspondences tend to converge more often.

V. CONCLUSION

The Canny detector had the best test performance and the best robustness in convergence. It is also one of the faster-executing detectors. Thus we conclude that it performs the best for the task of structure from motion. This conclusion is similar to that reached by Heath *et al.* [6] in the context of a human visual edge rating experiment, and by Bowyer *et al.* [38] in the context of ROC curve analysis.

The Bergholm detector had the best "test-on-training" performance in three categories (Table VI) while the Canny detector and the Rothwell detector were second. With separate test data, the Canny detector had the best performance with all image sequences for motion (Table VIII). Theoretically, it can be concluded that once *the optimal* parameter setting for the image sequences is found, the Bergholm detector can achieve the best performance, since it was the best performer in the test-on-train. However, in practice, the Canny detector performed better with any deviation from the training sequence. The SUSAN detector consistently was the worst performer in all categories in training, testing within same scene type, testing across scene type, and convergence rates.

The results obtained by varying length of dataset to the minimum 3) shows that the relative rankings of train and testing results were identical. When the "SFM algorithm" was varied by using the Weng's algorithm, the results were not as consistent. However, the Canny detector was still the best in test results.

ACKNOWLEDGMENT

The authors would like to thank C. Tayor and D. Kriegman for willingly answering numerous questions regarding the SFM algorithms. The authors would also like to thank M. Heath, S. Dougherty, C. Kranenburg, and S. Sarkar for many valuable discussions; and the authors of the edge detectors for making the implementations available. Early results of this work appear in [7], [8].

REFERENCES

[1] L. Kitchen and A. Rosenfeld, "Edge evaluation using local edge coherence," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-11, pp. 597–605, Sept. 1981.
[2] T. Peli and D. Malah, "A study of edge detection algorithms," *Comput. Graph. Image Process.*, vol. 20, no. 1, pp. 1–21, Sept., 1982.
[3] W. Lunscher and M. Beddoes, "Optimal edge detector evaluation," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-16, pp. 304–312, Apr. 1986.
[4] S. Venkatesh and L. Kitchen, "Edge evaluation using necessary components," *CVGIP: Graph. Models Image Understand*, vol. 54, pp. 23–30, Jan. 1992.
[5] R. N. Strickland and D. K. Chang, "Adaptable edge quality metric," *Opt. Eng.*, vol. 32, pp. 944–951, May 1993.
[6] M. Heath, S. Sarkar, T. Sanocki, and K. W. Bowyer, "A robust visual method for assessing the relative performance of edge detection algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 1338–1359, Dec. 1997.
[7] M. Shin, D. Goldgof, and K. W. Bowyer, "An objective comparison methodology of edge detection algorithms for structure from motion task," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Santa Barbara, CA, 1998, pp. 190–195.
[8] ——, "An objective comparison methodology of edge detection algorithms for structure from motion task," in *Empirical Evaluation Techniques in Computer Vision*, K. W. Bowyer and P. J. Phillips, Eds. Los Alamitos, CA: IEEE Comput. Soc. Press, 1998, pp. 235–254.
[9] C. Taylor and D. Kriegman, "Structure and motion from line segments in multiple images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 1021–1032, Nov. 1995.
[10] J. Weng, Y. Liu, T. Huang, and N. Ahuja, "Structure from line correspondences: A robust linear algorithm and uniqueness theorems," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 1988, pp. 387–392.
[11] J. R. Fram and E. S. Deutsch, "On the quantitative evaluation of edge detection schemes and their comparison with human performance," *IEEE Trans. Comput.*, vol. C-24, pp. 616–628, June 1975.
[12] D. J. Bryant and D. W. Bouldin, "Evaluation of edge operators using relative and absolute grading," in *Proc. IEEE Comput. Soc. Conf. Pattern Recognit. Image Process.*, Chicago, IL, Aug. 1979, pp. 138–145.
[13] I. E. Abdou and W. K. Pratt, "Quantitative design and evaluation of enhancement/thresholding edge detectors," *Proc. IEEE*, vol. 67, pp. 753–763, May 1979.
[14] G. B. Shaw, "Local and regional edge detectors: Some comparison," *Comput. Graph. Image Process.*, vol. 9, pp. 135–149, Feb. 1979.

[15] V. Ramesh and R. M. Haralick, "Performance characterization of edge detectors," *SPIE Applicat. Artificial Intell. X: Mach. Vision Robot.*, vol. 1708, pp. 252–266, Apr. 1992.
[16] X. Y. Jiang, A. Hoover, G. Jean-Baptiste, D. Goldgof, K. Bowyer, and H. Bunke, "A methodology for evaluating edge detection techniques for range images," in *Proc. Asian Conf. Comput. Vision*, Singapore, Dec. 1995, pp. 415–419.
[17] M. Salotti, F. Bellet, and C. Garbay, "Evaluation of edge detectors: Critics and proposal," in *Proc. Workshop Performance Characterization Vision Algorithms*, Cambridge, U.K., Apr. 1996.
[18] Q. Zhu, "Efficient evaluations of edge connectivity and width uniformity," *Image Vision Comput.*, vol. 14, pp. 21–34, Feb. 1996.
[19] J. Siuzdak, "A single filter for edge detection," *Pattern Recognit.*, vol. 31, pp. 1681–1686, Nov. 1998.
[20] M. Brown, K. Blackwell, H. Khalak, G. Barbour, and T. Vogl, "Multiscale edge detection and feature binding: An integrated approach," *Pattern Recognit.*, vol. 31, pp. 1479–1490, Oct. 1998.
[21] B. Chanda, M. Kundu, and Y. Padmaja, "A multi-scale morphologic edge detector," *Pattern Recognit.*, vol. 31, pp. 1469–1478, Oct. 1998.
[22] P. Tadrous, "A simple and sensitive method for directional edge detection in noisy images," *Pattern Recognit.*, vol. 28, pp. 1575–1586, Oct. 1995.
[23] T. N. Tan, "Texture edge detection by modeling visual cortical channels," *Pattern Recognit.*, vol. 28, pp. 1283–1298, Sept. 1995.
[24] J. Shen, "Multi-edge detection by isotropical 2-D ISEF cascade," *Pattern Recognit.*, vol. 28, pp. 1871–1885, Dec. 1995.
[25] S. Iyengar and W. Deng, "An efficient edge detection algorithm using relaxation labeling technique," *Pattern Recognit.*, vol. 28, pp. 519–536, Apr. 1995.
[26] D. Park, K. Nam, and R. Park, "Multiresolution edge detection techniques," *Pattern Recognit.*, vol. 28, pp. 211–229, Feb. 1995.
[27] S. Ando, "Image field categorization and edge/corner detection from gradient covariance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 179–190, Feb. 2000.
[28] J. Elder and S. Zucker, "Local scale control for edge detection and blur estimation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 699–716, July 1998.
[29] I. Matalas, R. Benjamin, and R. Kitney, "An edge detection technique using the facet model and parameterized relaxation labeling," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 328–341, Apr. 1997.
[30] T. Law, H. Itoh, and H. Seki, "Image filtering, edge detection, and edge tracing using fuzzy reasoning," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 481–491, May 1996.
[31] Z. Wang, K. Rao, and J. Ben-Arie, "Optimal ramp edge detection using expansion matching," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 1092–1097, Nov. 1996.
[32] L. A. Iverson and S. W. Zucker, "Logical/linear operators for image curves," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 982–996, Oct. 1995.
[33] F. Heijden, "Edge and line feature extraction based on covariance models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 16–33, Jan. 1995.
[34] R. Mehrotra and S. Zhan, "A computational approach to zero-crossing-based two-dimensional edge detection," *Graph. Models Image Process.*, vol. 58, pp. 1–17, Jan. 1996.
[35] P. Papachristou, M. Petrou, and J. Kittler, "Edge postprocessing using probabilistic relaxation," *IEEE Trans. Syst., Man, Cybern. B*, vol. 30, pp. 383–402, June 2000.
[36] L. Ganesan and P. Bhattacharyya, "Edge detection in untextured and textured images—A common computational framework," *IEEE Trans. Syst., Man, Cybern. B*, vol. 27, pp. 823–834, Oct. 1997.
[37] G. Chen and Y. Yang, "Edge detection by regularized cubic B-spline fitting," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 636–643, Apr. 1995.
[38] K. W. Bowyer, C. Kranenburg, and S. Dougherty, "Edge detector evaluation using empirical ROC curves," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, vol. 1, Fort Collins, CO, June 1999, pp. 354–359.
[39] M. Shin, D. Goldgof, and K. W. Bowyer, "Comparison of edge detectors using an object recognition task," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, vol. 1, Fort Collins, CO, June 1999, pp. 360–365.
[40] M. Black, G. Sapiro, D. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Trans. Image Processing*, vol. 7, pp. 421–432, Mar. 1998.
[41] F. Bergholm, "Edge focusing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, p. 726, 741, Nov. 1987.
[42] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 679–698, Nov. 1986.
[43] L. Rosenthaler, F. Heitger, O. Kubler, and R. von der Heydt, "Detection of general edges and keypoints," in *Proc. ECCV*, G. Sandini, Ed., 1992, pp. 78–86.
[44] C. A. Rothwell, J. L. Mundy, W. Hoffman, and V. D. Nguyen, "Driving vision by topology," in *Proc. Int. Symp. Comput. Vision*, Coral Gables, FL, Nov. 1995, pp. 395–400.
[45] S. Sarkar and K. L. Boyer, "Optimal infinite impulse response zero crossing based edge detectors," *Comput. Vision, Graph., Image Process.: Image Understanding*, vol. 54, pp. 224–243, Sept. 1991.
[46] I. E. Sobel, *Camera Models and Machine Perception*. Stanford, CA: Stanford Univ. Press, 1970, pp. 277–284.
[47] S. M. Smith and J. M. Brady, "SUSAN—A new approach to low level image processing," *Int. J. Comput. Vision*, vol. 23, no. 1, pp. 45–78, May 1997.
[48] R. Jain, R. Kasturi, and B. G. Schunck, *Mach. Vision*. New York: McGraw-Hill, 1995.
[49] W. Press *et al.*, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge, U.K.: Cambridge Univ. Press, 1992.

Min C. Shin (M'88) received the B.S. and M.S. degrees, with honors, in computer science from the University of South Florida (USF), Tampa, in 1992 and 1996, respectively. He is currently pursuing the Ph.D. degree at USF.

His research interests include nonrigid motion analysis and evaluation of algorithms.

Mr. Shin is a member of ACM, UPE, and the Golden Key Honor Society.

Dmitry B. Goldgof (S'88–M'89–SM'93) received the M.S. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1985 and the Ph.D. degree in electrical engineering from the University of Illinois, Urbana-Champaign, in 1989.

He is currently a Professor in the Department of Computer Science and Engineering at the University of South Florida, Tampa. His research interests include motion analysis of rigid and nonrigid objects, computer vision, image processing and its biomedical applications, and pattern recognition. He has published 39 journal and 90 conference publications, 14 books chapters, and one book. He is currently the North American Editor for *Image and Vision Computing Journal* and a member of the Editorial Board of *Pattern Recognition*.

Dr. Goldgof has served as General Chair for the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'98) and as an Area Chair for CVPR'99. He has just recently completed his term as Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING. He is a member of the IEEE Computer Society, IEEE Engineering in Medicine and Biology Society, SPIE—The International Society for Optical Engineering, Pattern Recognition Society, and the Honor Societies of Phi Kappa Phi and Tau Beta Pi.

Kevin W. Bowyer (F'91) received the Ph.D. degree in computer science from Duke University, Durham, NC.

He is currently a Professor in Computer Science and Engineering at the University of South Florida (USF), Tampa. He was previously on the faculty of the Department of Computer Science, Duke University, and at the Institute for Informatics of the Swiss Federal Technical Institute, Zurich. His current research interests include general areas of image understanding, pattern recognition, and medical image analysis. From 1994 to 1998, he served as North American Editor of the *Image and Vision Computing Journal*. In addition, he is a member of the editorial boards of *Computer Vision and Image Understanding, Machine Vision and Applications, and the International Journal of Pattern Recognition and Artificial Intelligence*.

Dr. Bowyer served as Editor-in-Chief of IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE from 1999 to 2000. He served as General Chair for the 1994 IEEE Computer Vision and Pattern Recognition Conference and served as Chair of the IEEE Computer Society Technical Committee on Pattern Analysis and Machine Intelligence from 1995 to 1997.

Savvas Nikiforou received the Higher National Diploma in electrical engineering from the Higher Institute of Technology, Cyprus, Greece, in 1991 and the B.Sc. degree in computer engineering from the University of South Florida, Tampa, in 1999, where he was also named the outstanding graduate of the Computer Science and Engineering Department. He is currently pursuing the M.S. degree in computer science at USF.

His research interests include artificial intelligence.