An Objective Comparison Methodology of Edge Detection Algorithms Using a Structure from Motion Task

Min C. Shin, Dmitry Goldgof, and Kevin W. Bowyer
Department of Computer Science & Engineering
University of South Florida
Tampa, FL 33620
shin, goldgof or kwb@csee.usf.edu

Abstract

This paper presents a ***task-oriented evaluation methodology for edge detectors. Performance is measured based on the task of structure from motion. Eighteen real*** image ***sequences from 2 different scenes varying in the complexity*** and ***scenery types are*** used. ***The task-level ground truth for each image sequence is*** manually ***specified in*** terms ***of the 3D motion and structure. An automated tool computes the accuracy of the motion and structure achieved using the set of edge maps.*** Parameter ***sensitivity and execution speed are also analyzed. Four edge detectors*** are ***compared. All implementations and data sets are publicly available.***

1 Introduction

The lack of an accepted objective empirical methodology of evaluating even the most widely used computer vision algorithms has delayed the further application of computer vision. Without such a methodology, it is difficult for the users to select the best algorithm for their need. Also, researchers who wish to develop better algorithms cannot document the true contribution of their work.

We found 22 new algorithms published in just 4 major journals (GMIP, PAMI, PR, SMC) since 1992. (Refer to Tables of [3]). The necessity of comparing with other works has been realized, in that 19 of the 22 papers compared their detector with one or more other detectors visually and/or quantitatively. However, only 9 comparisons used any previously published methodology. Three established their own quantitative comparison metrics. Seven others used only visual (qualitative) comparison. That there would be so little serious comparison is surprising considering that the first comparison methodology was published in 1975. Even more importantly, none of authors have used any real image ground truth to evaluate their algorithms. The quantitative comparisons were drawn from synthetic images. Even though the importance of using real images has been acknowledged, real images simply are not used during the quantitative evaluation. An additional concern is that the parameter sensitivity of edge detectors is generally not acknowledged. The output of an algorithm changes significantly with the parameter setting. However, none of 22 edge detection works states any details about how the parameters of edge detectors compared to were searched. This leaves one to wonder if the result of the edge detectors compared to could have been better if a better parameter setting was used.

Lastly, the purpose of edge detection needs to be remembered. Edge detection is not usually a final result by itself; it is an input for further processing. Therefore, the true performance lies in how well it prepares the input for the next algorithm.

We propose that a convincing comparison methodology should have the following features.

1. ***A comparison method*** must ***be objective*** and ***quantitative.***
2. ***A comparison method must be publicly available and easily applicable.***
3. ***A large data set*** must ***be carefully designed using real*** images.
4. ***A comparison method should evaluate the algorithm based on a vision task.***

In this research, the edge detection algorithms have been evaluated based on the task of structure from motion [3]. This is the first attempt to test edge detectors based on another highly researched vision algorithm. Four edge detectors are tested using 18 real image sequences, containing a total of 278 images. Six original sequences were taken from 2 different scenes. Two shorter sequences are derived from each original sequence, resulting in 12 shorter sequences.

0-8186-8497-6/98 $10.00 © 1998 IEEE

The comparison tool is developed to objectively evaluate the performance. Once the edge maps are created by an edge detector, the evaluation program will extract and correspond lines, execute the SFM program and evaluate the result using the comparison tool. The data set, comparison tool, and intermediate process implementations are publicly available at http://figment.csee.usf.edu. The entire comparison process can easily be applied to other edge detectors.

2 Background

2.1 Edge Detectors

Four detectors were studied. The Bergholm edge detector applies a concept of edge focusing to find *significant* edges. The image is smoothed using a coarse resolution (high smoothing) and the possible edges are located. Then, the neighbors of edges from coarser resolutions are checked in finer resolution. The Canny edge detector is considered as the standard methodology of edge detection. The image is smoothed with a Gaussian filter, and the edge direction and strength is computed. The edges are refined with non-maximal suppression and hysteresis. The Rothwell edge detector is similar to the Canny edge detector except 1) non-maximal 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. The Sarkar edge detector is an Optimal Zero Crossing Operator (OZCO).

2.2 Structure from Motion

A structure from motion (SFM) algorithm determines the structure (depth information) of a scene and the motion of the camera. For this comparison experiment, the structure and motion from line segments algorithm by Taylor and Kriegman was selected for several reasons [4]. First, the *working* implementation was obtainable from the web site. Second, the work had been extensively tested with synthetic and real data, and this data is also publicly available. Third, the algorithm was already tested against another structure from motion algorithm [5] and was concluded to be more stable under noise [4].

Given n images with m corresponded lines, the SFM algorithm extracts the depth information of the line (3D location of each line in the camera coordinates), and the motion of the camera. It solves the problem in terms of an objective function $\mathcal{O}$ which measures the disparity between the projected 3D line and its corresponding observed 2D line. The algorithm iterates searching for the structure and motion estimate which minimizes $\mathcal{O}$. The algorithm generates an initial random guess of camera positions for each iteration using the initial motion information. It is found that without providing any initial motion information, the algorithm usually managed to converge into a solution but after a far greater number of iterations. In order to speed up the process, a good motion estimate (rotation angle) was provided. A minimum of 3 images and 6 lines is required by the SFM algorithm, and more images or lines are allowed.

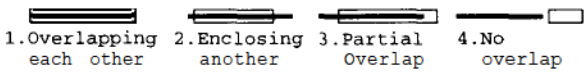


Figure 1: Possible Orientations of Collinear Lines.

3 Framework

The comparison methodology involves four steps : 1) edge detection, 2) intermediate processing, 3) SFM, and 4) evaluation of the result.

3.1 Intermediate Processing

Given the edge maps generated by each edge detector, the intermediate processing involves extracting and corresponding lines. These steps prepare the input for the SFM program. It must be acknowledged that these steps are not perfect nor optimized since the result can vary depending on the sophistication of the algorithm used and/or the parameters selected. The overall goal is to compare the edge detectors, and algorithms that we believe will give the minimum amount of advantage or disadvantage to any particular edge detectors are adopted.

A simple method of line extraction is implemented. Line segments are represented by two endpoints in pixel coordinates. First, the edge links are created from the edge map by scanning from left to right and top to bottom. A link starts with an edge pixel with only 1 neighbor. The 8-connected neighboring edge pixels are recursively linked until 1) there are no more neighbor edge pixels or 2) there is more than one neighbor, indicating a possible junction or branch. Second, the edge links are divided at a high curvature point with the curvature angle greater than $T_{high\_curv\_angle}$. Third, edge links are further broken to form line edge segments using the Polyline Splitting Technique until the farthest point is closer than $T_{point\_to\_line}$. Then, the line edge segment (a chain of edge pixels) is fitted to a line using the Least Squared Estimation, and the ending edge pixels are projected to the line to the nearest pixel. These two projected points are used to describe a line. Finally, lines which are shorter than $T_{min\_line\_length}$ are eliminated.

The input of the SFM is a set of line correspondences across a sequence of images. Manually match-

ing the lines would provide the most error-free method possible. However, it is not practical to match lines from 18 image sequences (total of 278 images) for four edge detectors where each edge detector will be tested on minimum of 177 parameters. Also, the manual matching could possibly give advantage or disadvantage to a particular edge detector since the human can actually use the knowledge of the scene to result in better matches. Therefore, an automated line matching program was developed which will give 1) the minimum advantage to any particular edge detector and 2) the minimum mismatches. Note that the SFM program will produce a **wrong** result if any lines are mismatched, so minimizing mismatches is extremely important.

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. Second, the Machine Estimated (ME) lines (l_{ik} for image i and line k) from the Line Extraction program were corresponded 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 two conditions are met.

1. **Collinearity.** If the sum of the distance between the endpoints of L_{ij} to l_{ik} is less than T_{perp_dist}, 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} and 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 Figure 1). Obviously, the orientation #1 indicates overlap while #4 indicates non-overlap. 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. In case of the partial overlap (#3), if 1) the intersection of l'_{ik} and L_{ij} is greater than $T_{overlap}$ percent of L_{ij} and 2) the intersection of L'_{ik} and l_{ij} is greater than $T_{overlap}$ percent of l_{ij}, then they are considered overlapping. Note that many GT lines to 1 ME line correspondence is not allowed.

The SFM requires a minimum of 3 correspondences for each line. Lines with less than this T_{min_corr} correspondence are dropped.

3.2 Imagery Design

Some comparisons have concentrated primarily on the evaluation criteria and ignored the importance of the dataset; 13 out of 15 edge detection comparison methodologies have used 4 or less synthetic and real images. The dataset should be large and thorough in order for the users of the methodology to have confidence in evaluation. Four images (including synthetic) would not seem to be thorough enough.

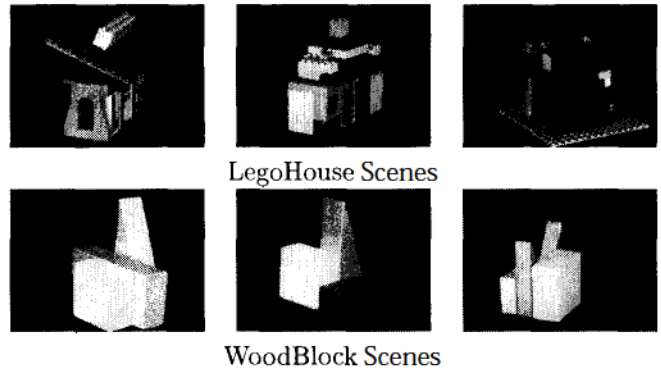


Figure 2: Image Scenes (6 original sequences)

In this work, a large dataset of 18 sequences (6 original and 12 derived) containing 278 images is carefully designed considering different aspects influencing the edge detectors and the structure from motion task. (See Table 1 & Figure 2).

3.3 Ground Truth

The ground truth is manually defined in terms of 1) the rotation angle between two consecutive images, and the rotation axis, and 2) "structure," measured as the angle between selected pairs of lines.

The object is rotated on a rotation stage according to the predetermined GT rotation. To determine the rotation axis of the stage, 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. Two points defining the endpoints of the rotation axis are determined from the intensity image. The 3D location of two points is extracted from the range image, and the vector defined by two points is normalized to represent the rotation axis. The structure GT is defined by a set of two lines of the object and the angle between them.

3.4 Performance Metrics

The ME result is compared with GT in two areas : motion (rotation axis, rotation angle) and structure. Since the object is rotated along the fixed rotation axis with no translation, the 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

Table 1: Properties of Image Sequences

image set name	# of images	total rotation angle	# of lines	Ave # of corres per line	Ave line length (pixel^2)
LH1	18	160.00°	122	8.80	80.64
LH1.A	12	160.00°	122	5.94	80.17
LH1.B	9	160.00°	122	4.46	81.42
LH2	19	355.00°	104	6.89	89.44
LH2.A	13	355.00°	104	4.76	88.41
LH2.B	10	355.00°	104	3.63	89.19
LH3	20	190.00°	118	7.58	83.29
LH3.A	14	190.00°	118	5.31	84.13
LH3.B	10	180.00°	118	3.83	82.15
WB1	18	170.00°	29	11.03	132.91
WB1.A	9	160.00°	29	5.52	133.87
WB1.B	12	160.00°	29	7.31	133.69
WB2	28	275.00°	36	15.72	110.85
WB2.A	14	265.00°	36	7.94	110.56
WB2.B	17	275.00°	36	9.50	111.86
WB3	30	285.00°	47	18.17	91.93
WB3.A	15	270.00°	46	9.24	92.04
WB3.B	10	260.00°	45	6.18	92.77

# of lines, Ave # of correspondence/line, Ave line length are computed from the manually specified GT lines

method. (Refer to Figure 3). 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 $MotionError = \frac{ME_{error}}{GT_{move}}$ 100% where ME_{error} is the distance between P_{GT_j} and P_{ME_j}, and GT_{move} is the distance between P_{GT_0} and P_{GT_j}. The percentage error is used since the absolute distance error would hold different significance depending on the amount of the camera movement. The structure error is measured by computing the angle difference between the ME angle and its corresponding GT angle.

3.5 Parameter Training

This final result depends on parameters of the edge detector (3), line extraction (3), and line correspondence (3). Finding the best setting of 9 parameters seemed 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, $T_{high\_curv\_angle}$=90.0, $T_{min\_line\_length}$=50.0, T_{perp_dist} =5.0, $T_{overlap}$=80.0, and T_{min_corr}=3. These values are fixed for all experiments.

Second, realizing that the edge detector's performance greatly varies with parameters, an adaptive method of searching for the best parameter values is

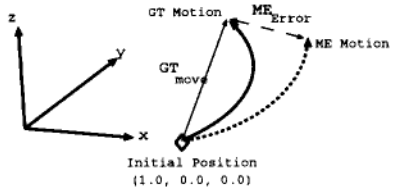


Figure 3: Motion Error Measurement

used. A 5x5x5 initial uniform sampling of parameter space is tested. The area around the best parameter point in this coarse sampling is further subsampled at 3x3x3 (with the previous best at the center). A minimum of 2 subsamplings is executed, resulting in a minimum attempt of 177 different parameter settings. Subsampling is continued while there is a minimum of 5% improvement. The parameters are trained separately for structure and motion. After the average of 2.67 subsampings and the maximum of 8 subsamplings, the best parameters were found.

4 Results

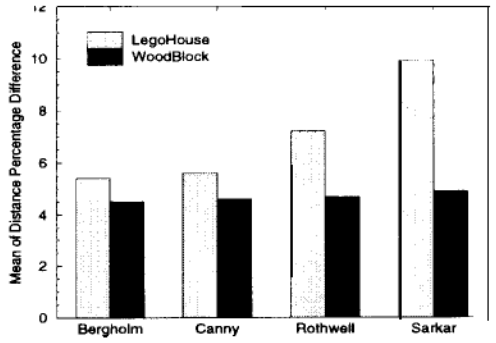
The results of 9 LegoHouse and 9 WoodBlock sequences are presented. The section is divided into three sections: Train, Test, and Parameter Sensitivity & Speed. The motion performance is obtained with the motion-trained parameters and the structure performance with the structure-trained parameters. The results are categorized into two scene groups (LegoHouse and WoodBlock) and by two metrics (motion and structure) resulting in 4 categories.

4.1 Train

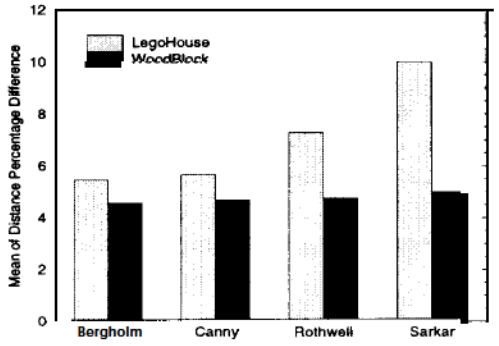
Edge detectors are trained separately for each image sequence with GT information given. The performance with trained parameters is the best possible for the given image sequence with the parameter training algorithm described in the previous section.

First, the Bergholm was able to obtain the best train motion and structure performance in both scene types. (Refer to Figure 4). The Canny and the Rothwell performed the second, while the Sarkar performed the worst in all 4 of them. Most of edge detectors performed better with WoodBlock sequences. This could be the result of longer lines with longer correspondences being available in the WoodBlock scenes. (Refer to Table 1). Another interesting observation is that the rankings were identical between 2 scenes for a given metric (motion or structure).

Second, the performance varies greatly with the parameter settings for all edge detectors. The mean of motion and structure performance obtained through all parameter setting attempted during the Train is

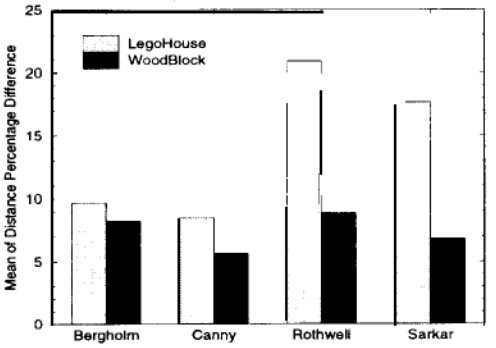


(a) Motion performance

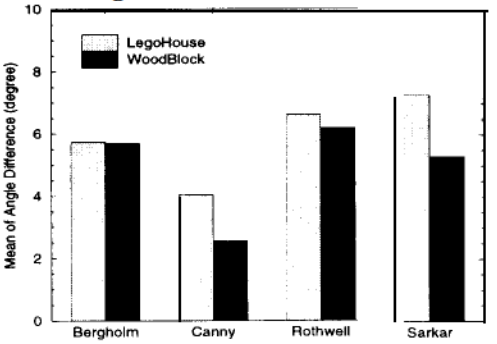


(b) Structure performance

Figure 4: Train Performance



(a) Motion performance
48/72 (LegoHouse) &72/72 (WoodBlock)



(b) Structure performance
40/72 (LegoHouse) &63/72 (WoodBlock)

Figure 5: Performance (on Common Converged Trials.)

analyzed. The mean motion performance was nearly twice the best, and the structure performance was nearly 4 times the best. Especially, the structure performance showed a great variation among the parameter settings. This confirms the importance of the parameter training for edge detector performance. The Canny had the narrowest motion difference, while the Sarkar had the narrowest structure difference.

4.2 Test

In order to test the performance of the edge detectors on sequences different from those trained on, the parameters trained for one sequence were tested on other sequences within the scene group. In our test data, there are 9 image sequences, where each sequence is tested on all other sequences for motion and structure separately. Therefore, for each edge detector we tested 144 times for each image scene type: 9 sequence x 8 trained parameter settings from other sequences in the group x 2 trainings (for motion and for structure).

It is important to realize that under some instances (settings of edge detector parameters), the edge maps resulted in a set of corresponded lines that the SFM program could not converge into any solution after 1750 iterations. In order to take this problem into consideration, the analysis of Test is divided into two sections. First, the convergence rates are compared. Second, the performance of the test trials where all edge detectors successfully converged into a solution is analyzed.

First, even though the testing is performed within the same scene group, not all edge detectors converged into good solutions in all testings. The Canny showed the best convergence rate (overall 98.65%), while the Rothwell had the worst (overall 84.72%). Lower convergence rates were achieved from the structure-trained parameters than for the motion-trained parameters, and also in LegoHouse scenes compared to the WoodBlock scenes.

Second, the test performance is compared. The Canny performed best in all 4 testing categories with all converged sequences. (See Figure 5). The Rothwell performed the worst in 3 categories. The Bergholm, which showed the best train performance, shared the second and the third with the Sarkar. Interestingly, the Sarkar, with the worst trained performance, showed a small degradation in the test. This behavior

could have been expected from a relatively small difference between the train performance (Figure 4) and the mean performance of all attempted parameter settings.

4.3 Parameter Sensitivity & Speed

The effect of each parameter on the performance is analyzed. First, the standard deviations of the measurements of the converged parameter setting (structure and motion separately) where only one parameter was varying are calculated. Then, the mean of standard deviations is calculated for all image sequences.

First, the smoothing operators (S_{start}, S_{end}, and sigma) were usually the most influential parameters. Second, low and ***high*** of the Sarkar had little effect on either metric. Therefore, for the Sarkar, the parameter tuning ***could be*** reduced by 2 dimensions within the range of *low* and ***high*** which will lead to converging results. This finding is extremely valuable, since the parameter tuning is important and time consuming. However, the motion performance of the Sarkar in the LegoHouse scene was highly sensitive with the ***sigma.*** All parameters of the Bergholm and the Rothwell were highly influential while all three parameters of the Canny were usually less influential than other edge detectors' parameters. Also, the ranking of the sensitivity of the parameters (such as the ranking of S_{start}, S_{end}, T of the Bergholm) within each edge detector was similar across the 4 categories.

In order to compute the speed of the edge detectors, one image was run with the initial 5x5x5 parameter samplings on a Sun Ultra Sparc Workstation. The average execution times were 40 sec (Bergholm), 8 sec (Canny), 8 sec (Rothwell), and 14 sec (Sarkar). The precise differences in execution time are likely not reliable, as these was no attempt to assure comparable levels of efficiency in the different implementations.

5 Conclusion

First of all, the sensitivity of edge detectors' performance to changes in the parameters is verified. This corresponds with one conclusion of Heath ***et al***'s work [2]. The edge detectors showed the average degradation from the best to the mean of the average factor of 2 (motion) and 4 (structure). (Refer to Figure 4).

Second, the Bergholm had the best "test-on-training" performance in both metrics (Figure 4), while the Canny and the Rothwell were second. With separate test data, the Canny had the best performance with both scene types for both motion and structure (Refer to Figure 5). In Heath ***et al's*** work, the Canny performed the best when the parameter was adapted for each image and the worst when fixed for all images. Theoretically, it can be concluded that once ***the optimal*** parameter setting for the image sequences is found, the Bergholm can achieve the best performance, since it was the best performer in the test-on-train. However, in practice, the Canny performed better with any deviation from the training sequence.

Overall, the Canny had the lowest sensitivity to the parameter variations, the best test performance, the fastest speed, and the robustness of highest convergence rate. Thus we concluded that it performs the best for the task of structure from motion. This conclusion is similar to that reached by Heath et ***al*** [2] in the context of a human visual edge rating experiment, and by Dougherty and Bowyer [1] in the context of ROC curve analysis. Thus it appears that the Canny may be a preferred edge detector for a very broad range of tasks.

There are several desirable directions for extending this work. One is to include additional edge detectors. Another is to include more complex image sequences. A third is to more closely compare the results of this analysis with those of Heath ***et*** al [2] and Dougherty and Bowyer [1].

Acknowledgments

We thank Taylor and Kriegman for answering numerous questions regarding the SFM algorithms; Mike Heath, Sean Dougherty and Sudeep Sarkar for many valuable discussion; and the authors of the edge detectors for making the codes available. The work was supported by the NSF grants CDA-9724422, EIA-9729904, IRI-9619240, IRI-9731821.

References

[1] S. Dougherty and K. W. Bowyer, "Objective Evaluation of Edge Detectors Using a Formally Defined Framework," To appear in *IEEE* Computer *Society Workshop on Empirical Evaluation of* Computer *Vision Algorithms.*

[2] 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* on *PAMI,* 19 (12), 1338-1359, December 1997.

[3] M. Shin, D. Goldgof, and K.W. Bowyer, "An Objective Comparison Methodology of Edge Detection Algorithms for Structure from Motion Task," To appear in *IEEE Computer Society Workshop on* Empirical *Evaluation* ***of*** Computer *Vision Algorithms.*

[4] *C.* Taylor and D. Kriegman, 'Structure and Motion from Line Segments in Multiple Images," *IEEE Transactions on PAMI,* Vol. 17, pp. 1021-1032, Nov. 1995.

[5] J. Weng, Y. Liu, T. Huang, and N. Ahuja, "Structure from Line Correspondences: A Robust Linear Algorithm and Uniqueness Theorems," *IEEE* Conference on *CVPR,* pp. 387-392, 1988.