

A Comparative Analysis of Depth-Discontinuity and Mixed-Pixel Detection Algorithms

Pingbo Tang
Carnegie Mellon University
tangpingbo@cmu.edu

Daniel Huber
Carnegie Mellon University
dhuber@ri.cmu.edu

Burcu Akinci
Carnegie Mellon University
bakinci@cmu.edu

Abstract

Laser scanner measurements are corrupted by noise and artifacts that can undermine the performance of registration, segmentation, surface reconstruction, recognition, and other algorithms operating on the data. While much research has addressed laser scanner noise models, comparatively little is known about other artifacts, such as the mixed pixel effect, color-dependent range biases, and specular reflection effects. This paper focuses on the mixed pixel effect and the related challenge of detecting depth discontinuities in 3D data. While a number of algorithms have been proposed for detecting mixed pixels and depth discontinuities, there is no consensus on how well such algorithms perform or which algorithm performs best. This paper presents a comparative analysis of five mixed-pixel/discontinuity detection algorithms on real data sets. We find that an algorithm based on the surface normal angle has the best overall performance, but that no algorithm performs exceptionally well. Factors influencing algorithm performance are also discussed.

1. Introduction

Laser scanners are a popular tool for creating three-dimensional (3D) models of real-world environments. This “modeling-from-reality” problem finds application in a variety of domains ranging from architecture, engineering, and construction (AEC) to cultural heritage preservation [1, 2]. Commercial hardware and software is available for creating these digital models, and companies even provide modeling services. Many, if not most, modeling-from-reality applications place a premium on geometric accuracy of the constructed 3D model. Considering the importance of modeling accuracy, relatively little research has focused on understanding the sources of errors in the

sensing and modeling process [3]. Furthermore, despite the fact that a myriad of algorithms have been put forward for the various stages in the modeling pipeline (e.g., data segmentation [4] and noise filtering, coarse-registration, fine-registration [5], etc.), there have been few comparative analyses, and for most algorithms, it is still not known which ones perform best and under what circumstances. The goal of modeling-from-reality should be not only to create a digital model, but also to know how accurate that digital model is. There is a clear need for a deeper understanding of the sources of artifacts and uncertainties in the modeling-from-reality process and an analysis of the effectiveness of current algorithms in detecting artifacts and in representing uncertainties.

Commercial laser scanners are advertised as being highly accurate devices, with range accuracies from sub-millimeter levels at short distances to tens of millimeters at longer distances [1, 6]. While these specifications may be correct for ideal conditions, real laser scan data exhibits a variety of artifacts that reduce the accuracy of significant regions within the data and, consequently, calls into question the fidelity of the final model. Laser data artifacts that commonly occur in real-world data include: phantom points and surfaces at depth discontinuities (the mixed-pixel effect), range errors for thin structures [7], range jumps at reflectance and color boundaries [8], and large errors due to specular reflections (see Figure 1).

This paper is a first step in investigating these artifacts. Here, we focus on artifacts that occur at depth discontinuities. Specifically, we will analyze the effectiveness of algorithms for detecting depth discontinuities and for eliminating mixed pixels. Depth discontinuity detection is an important part of modeling-from-reality pipelines that utilize surface meshes. In such systems, the point clouds produced by the laser scanner must be triangulated at some point, often directly from range images. Triangles that cross depth discontinuities will create surfaces that do not

correspond to any real surface (Figure 1). These “phantom” surfaces can compromise the reconstructed model directly by their presence and indirectly by corrupting subsequent modeling processes, such as registration.

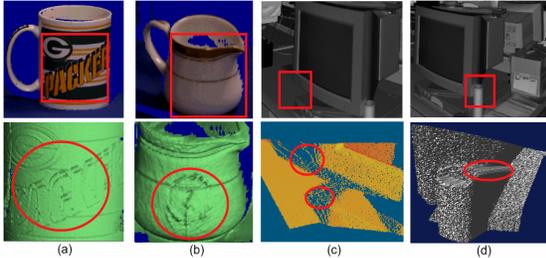


Figure 1. **Artifacts in the laser scanned data: (a) color boundary effects; (b) specular surface effects; (c) mixed pixel; (d) phantom surfaces.**

Mixed pixel detection is closely related to depth discontinuity detection. Mixed pixels occur when the sensed data for a single measurement comes from two different surfaces, for example, when the laser spot bridges a depth discontinuity. The range for a mixed pixel is typically between the ranges of the two surfaces but may be closer than the near surface or further than the far surface in some cases. Mixed pixels can create a curling effect on the surface near depth discontinuities, with the same detrimental effects on the model accuracy that occur with undetected depth discontinuities. Mixed pixels also affect modeling-from-reality pipelines that only use point clouds, since it is the vertex data that is corrupted.

In this paper, we compare several popular mixed-pixel and depth discontinuity detection algorithms that are known to us (either through published papers or documented in available software):

1) *Normal-angle filter* – Detect discontinuity triangles based on the angle between the local surface normal and viewing direction [9].

2) *Edge length filter* – Detect discontinuity triangles based on the length of triangle edges [9].

3) *Boundary removal variations* – The above normal-angle filter or edge-length filter combined with added step of removing triangles adjacent to depth-discontinuities.

4) *Cone of influence algorithm* – Detect mixed pixels where several neighbors fall within a cone of influence along the viewing direction of the laser scanner [10].

These algorithms will be described in more detail in section 3. It is straightforward to modify a depth discontinuity detection algorithm to detect mixed pixels and vice versa (see Section 3.3), so these variations will be evaluated as well. We use ROC (receiver operator characteristic) curves to quantitatively evaluate the performance of the

algorithms in terms of the rate of correct detection and false alarms.

2. Related work

Mixed pixel artifacts were observed by [11] in early research on the use of laser scanners. The authors suggest two approaches for mixed pixel detection: using a median filter or removing isolated points in 3D. However, neither algorithm was further developed in the paper, and no quantitative results were presented. Adams [8, 12] developed computational models of the mixed pixel effect for amplitude modulated continuous wave (AMCW) scanners. The models, however, require the capability of obtaining many overlapping samples of range and signal strength as the laser beam passes across a depth discontinuity. Such an approach is not practical with commercially available scanners, since they are generally designed with little or no overlap between samples. Ye and Borenstein [13] developed the Certainty Assisted Spatial filter, which operates on elevation maps to filter out mixed pixels. This algorithm has limited applicability, since it relies on the elevation map representation. Tuley et al [14] developed a taxonomy of mixed pixel phenomena, with a focus on thin structures. They observed that mixed pixels are more problematic for small depth discontinuities than for large ones. Edge detection in range images is used for range image segmentation [15]. Since depth discontinuities result in edges in range images, such algorithms could potentially be used for mixed pixel and depth discontinuity detection.

3. Approach

3.1. Depth discontinuity detection

In our evaluation, the two algorithms for explicitly detecting depth discontinuities are the normal-angle filter and the edge-length filter. Both algorithms require triangulated data as input. For individual range images, it is convenient to use an intermediate representation, which we call the 3D grid. In a 3D grid, data points are organized in a grid just as with a range image, but instead of storing a range value at each pixel, the corresponding 3D vertex Cartesian coordinates are stored. Triangulating a 3D grid is straightforward. Triangles are formed by connecting adjacent vertices in a square and then connecting one of the diagonals. In our implementation, we connect the shortest diagonal. This is a conservative approach from the perspective of discontinuity detection,

because discontinuity detection algorithms utilizing triangle edge length information tend to identify triangles with long edges as discontinuity triangles, and constructing triangles with short edge lengths will generate fewer potential discontinuity triangles.

The edge-length algorithm relies on the observation that triangles spanning a depth discontinuity often have long edge lengths. Any triangle with an edge longer than a specified threshold (d_{EL}) is marked as a depth discontinuity triangle. This algorithm does not take into account the fact that edge lengths increase with increasing distance from the sensor, so we can expect that the accuracy will be worse for scenes with large depth of field. In following sections we call this algorithm the *edge* algorithm.

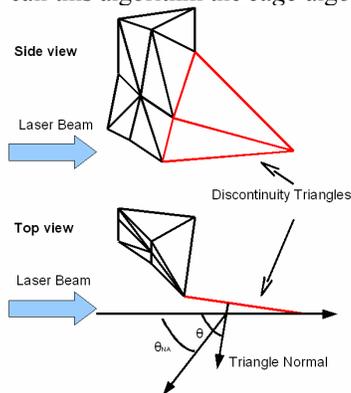


Figure 2. **View angle of the scanner for a discontinuity triangle.**

The normal-angle algorithm relies on the fact that triangles spanning depth discontinuities generally have an oblique orientation with respect to the viewing direction. A triangle is marked as a depth discontinuity if the angle between the surface normal and the line from the triangle's centroid to the sensor origin is larger than a threshold (θ_{NA}) (Figure 2). Hereafter, we call this algorithm the *normal* algorithm.

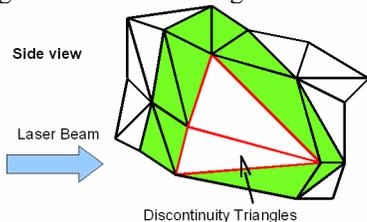


Figure 3. **Boundary removal extension.**

Points immediately adjacent to those depth discontinuity triangles are also likely to be influenced by mixed pixel effects. It is common practice to augment the two aforementioned algorithms to incorporate a boundary removal step. With the boundary removal extension, any triangle adjacent to a previously marked triangle is also marked as a depth discontinuity (see Figure 3). In following sections we

call these two algorithms as the *normal2* and *edge2* algorithms.

3.2. Mixed pixel detection

The Light Form Modeler (LFM) software from Zoller and Frolich incorporates a mixed pixel detection algorithm which is detailed in their documentation [10]. The algorithm uses a 3D grid representation but does not require triangulation. For each 3D grid point, the 3D points in the 8-connected neighborhood are examined to determine how many fall within a cone of a specified angle (θ_{CI}) surrounding the viewing direction line (see Figure 4). If more than N_{CI} points lay within this cone, then the center point is marked as a mixed pixel. In following sections we call this algorithm the *cone* algorithm.

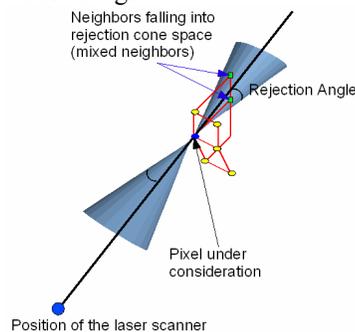


Figure 4. **Illustration of the cone algorithm.**

3.3. Converting between detection methods

It is possible to adapt a depth discontinuity detection algorithm into a mixed pixel detection algorithm and vice versa. We use the following approaches for converting between the two categories of algorithms.

Depth-discontinuity to mixed pixel: After eliminating depth discontinuity triangles, any isolated 3D points (i.e., points not connected to any remaining triangles) are considered mixed pixel points.

Mixed-pixel to depth-discontinuity: Triangles with one or more corners marked as mixed-pixels are considered depth-discontinuity triangles.

We use these conversion methods in our experiments to enable all algorithms to be compared in each category, keeping in mind that the converted algorithms were not explicitly designed for the alternate detection task.

Table 1. **Summary of parameters used in the algorithms.**

Algorithm Name	Parameters
Cone Algorithm	θ_{CI} : cone angle; N_{CI} : number of neighboring points falling into the cone.
Edge and Edge 2	d_{EL} : triangle edge length threshold.
Normal and Normal2	θ_{NA} : viewing angle of the triangle from the scanner.

3.4. Experimental setup

We used a phase-based laser scanner, the Z+F Imager 5003, to scan four different 3D scenes, which were chosen to ensure a variety of scene types and scales and depth discontinuity sizes. These scans include two outdoor scans (a highway bridge and a campus building), and two indoor scans (a laboratory scene and the elevator lobby of a public conservatory). The variety of sizes of discontinuity and contextual conditions from these four scans provides various types of depth discontinuities to comprehensively investigate the performance of the algorithms.

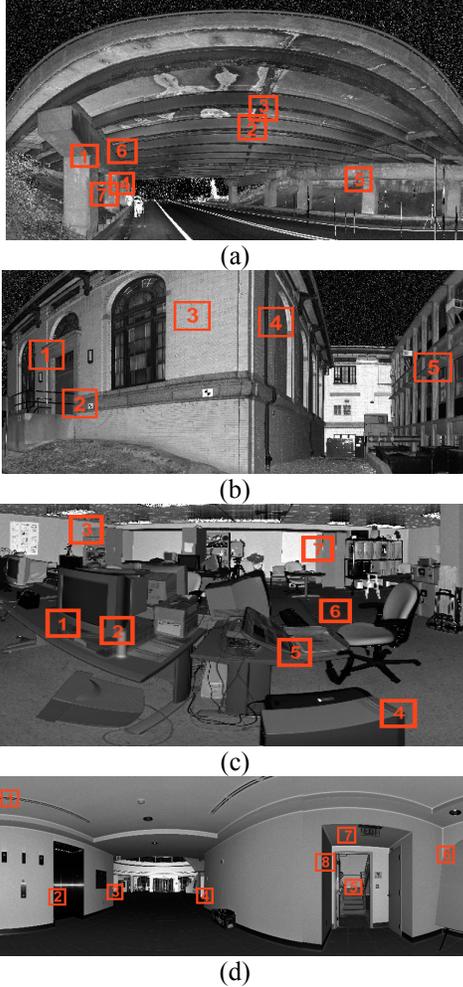


Figure 5. **Selected patches: (a) highway bridge (b) campus building (c) laboratory and (d) public conservatory.**

We selected representative data patches from each scan based on several considerations. First, the data samples should cover different sizes of discontinuities in terms of the length of the discontinuity as well as the amount of depth difference between front surface and back surface. Second, the selected data samples should be taken from various object surfaces in terms of

reflectivity as well as texture, since it is known that data noise levels vary according to reflectivity and other material properties, and the experiments should not be biased to any specific type of surface. Third, the incident angle of laser beam on the object surface also influence the data accuracy, so the selected data patches should include different incident angle cases. Fourth, we also select some patches without any depth discontinuities to evaluate the false detection capabilities of the algorithms. Fifth, all data patches are roughly 200 x 200 pixels in size, and the number of samples from different contextual cases are roughly the same in order to make this comparative analysis neutral to various application environments. Figure 5 shows the four scans used in this comparative analysis and the selected data patches. We selected 7 patches from the highway bridge scan, 5 patches from the campus building scan, 7 patches from the laboratory scan, and 8 patches from the public conservatory scan.

We manually labeled mixed pixels and discontinuity triangles to create the ground-truth for algorithm performance evaluation. For labeling mixed pixels, we imported the point cloud data into Polyworks IMInspect package [9] from InnovMetric Inc., manually removed mixed pixels and exported the cleaned point clouds as the ground truth. For labeling discontinuity triangles, we used the 3D grid triangulation method described in section 3.3.1 to triangulate the point cloud and generate a triangulated VRML model, imported that model into 3D model editing environment AC3D 6.1 from Inivis [16], manually removed all discontinuity triangles, and then exported the manual processed triangulated model as the ground-truth model.

By comparing the manually processed point clouds and triangulated 3D model with the labeling results of selected algorithms, we can label each pixel or triangle as one of the following categories:

- 1) *True positive*: a discontinuity triangle or mixed pixel is correctly identified by the algorithm.
- 2) *True negative*: a normal triangle or pixel is correctly identified by the algorithm.
- 3) *False positive*: a normal triangle or pixel is falsely reported as discontinuity triangle or mixed pixel by the algorithm.
- 4) *False negative*: the algorithm fails to identify a discontinuity triangle or mixed pixel.

$$R_{tp} = \frac{n_{tp}}{n_{tp} + n_{fn}}; \quad R_{fp} = \frac{n_{fp}}{n_{fp} + n_{fn}} \quad (1)$$

We calculate *true positive rate* (R_{tp}) and *false positive rate* (R_{fp}) as shown in (1); where n_{tp} is the number of pixels or triangles in the true positive category, n_{fn} is the number of false negatives, n_{fp} is the number of false positives, and n_{fn} is the number of true

negatives. An algorithm with high true-positive rate and low false-negative rate is preferred. By varying the threshold parameters of each algorithm, we obtain a series of R_{tp} - R_{fp} value pairs, which are plotted to form an ROC curve. In an ROC curve, the ideal algorithm would touch the top left corner of the graph (100% true positives with 0 false positives). In practice, the parameter setting that comes closest to the ideal (in distance) is frequently used. An algorithm is superior to another algorithm if its ROC curve is consistently above and to the left of the other algorithms' curves.

4. Results and discussion

Among the selected algorithms, the two variations of the normal algorithm (normal and normal2) and the two variations of the edge algorithm (edge and edge2) have only one changeable parameter, while the cone algorithm has two parameters. The ROC curves for the single parameter algorithms are straightforward. For the two parameter algorithm, we evaluated all possible values of NCI (from 1 to 8) and values for the rejection angle (θ_{CI}) of 5, 9, 15, 17 and 20 degrees. Among these variants, we found that setting θ_{CI} to 5 and 15 degrees were the two best-performing settings, so we use these fixed θ_{CI} values when comparing the cone algorithm to the others. In the figures, these variants are labeled cone5 and cone15 respectively. As we have discussed above, we can convert a depth discontinuity detection algorithm into a mixed pixel detection algorithm and vice versa, so for all above algorithms, we generated ROC curves for both depth discontinuity detection (see Figure 6) and mixed pixel detection (see Figure 7).

Several observations are noticeable from these ROC curves. First, for highway bridge patches, the performance of the edge2 and normal2 algorithms are almost equally the best, and these two algorithms' capabilities for detecting discontinuity triangles are better than their capabilities for detecting mixed pixels. Both the edge2 algorithm and normal2 algorithm can detect 86 percent of the mixed pixels while sacrificing about 9 percent of normal pixels. For discontinuity detection, the edge2 and normal2 algorithm both detect about 93 percent of the discontinuity triangles while sacrificing about 8 percent of valid triangles.

Second, for laboratory patches, which represent the indoor and small-discontinuity environments, all algorithms perform relatively poorly. The edge2 algorithm performs the best among them: both its mixed-pixel ROC curve and discontinuity-triangle ROC curve are closer to point (0, 1) than that of any other algorithms. It detects about 93 percent of mixed pixels while sacrificing sacrifices more than 10 percent of normal pixels. It detects about 93 percent of

discontinuity triangles while sacrificing almost 20 percent of normal triangles. It is also noticed that in the indoor and small-discontinuity environment all algorithms perform worse in the task of detecting discontinuity triangles than in the task of detecting mixed pixels.

Third, for public conservatory patches, which represent the indoor and medium-discontinuity environments, all algorithms perform better than that in the highway bridge case. In this case, normal2 algorithm performs the best among all algorithms. It can detect 96 percent of mixed pixels while only sacrificing about 8 percent of normal pixels. It can detect 96 percent of discontinuity triangles while only removing about 7 percent of valid triangles.

Fourth, for campus building patches, which represent the outdoor and various-discontinuities environments, all algorithms perform better than any of the cases stated above. In this case, normal2 algorithm performs the best in terms of mixed-pixel detection. It only sacrifices about 6 percent of valid pixels while detecting about 96 percent of mixed pixels. However, the performance of edge2 algorithm is only a little bit worse than it: it can detect about 95 percent of mixed pixels while sacrificing about 6 percent of normal pixels. The edge2 algorithm performs the best in terms of depth-discontinuity detection. It sacrifices about 5 percent of the valid triangles while detecting about 96 percent of discontinuity triangles.

Fifth, by summing up all ROC curves to evaluate the overall performance of all algorithms (see Figure 8), we found that normal2 algorithm performs the best: it can detect about 92 percent of all mixed pixels on all selected patches while keeping 93 percent of all valid pixels on all selected patches untouched; it can detect about 90 percent of all discontinuity triangles on all those selected patches while only removing 5 percent of valid triangles.

We visualized the algorithm evaluation results for each selected patch by labeling pixels or triangles falling into different categories (true positive, true negative, false positive and false negative) with different colors. Figure 9 (patch 1 from laboratory scan) and Figure 10 (patch 2 from bridge scan) are two representative visualization results. For each algorithm, we illustrate the parameter value that minimizes the distance to the top-left corner of the ROC curve.

Several observations are noticeable in these visualization results. First, for cone algorithm, even though the ROC curves shows no substantial performance difference between cone5 and cone15, the visualization results show that cone5 is better than cone15. In both examples, most discontinuities detected by cone5 are around actual discontinuities while cone15 has many false-positive cases scattered

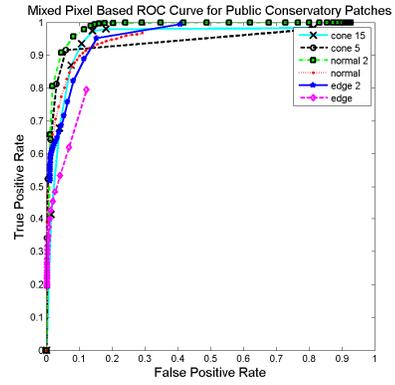
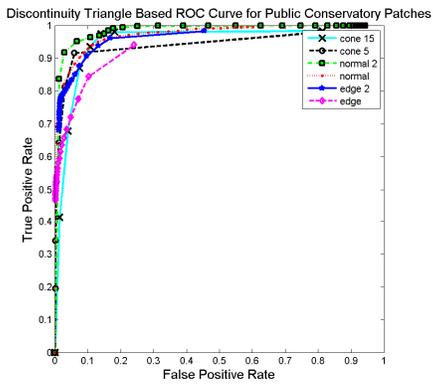
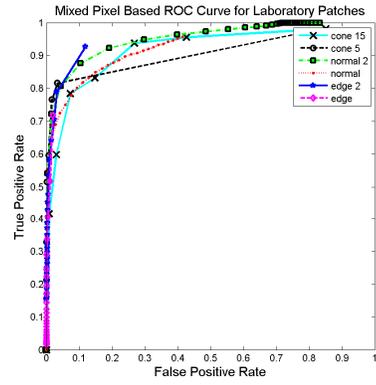
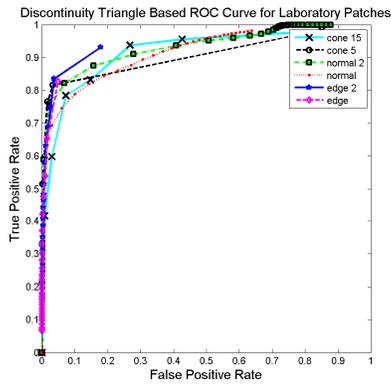
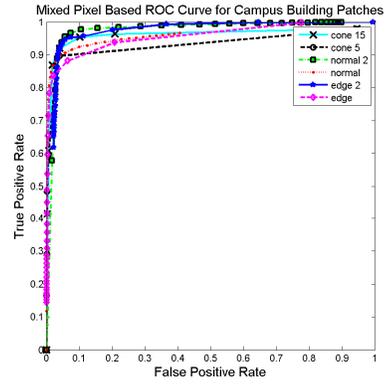
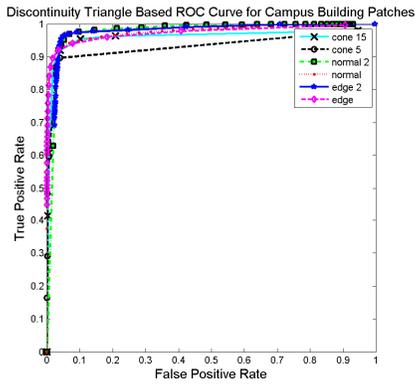
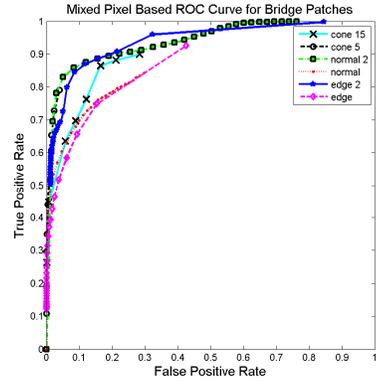
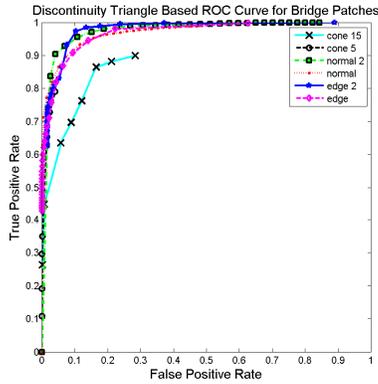


Figure 6. Discontinuity triangle ROC curves of single scans.

Figure 7. Mixed-pixel ROC curves of single scans.

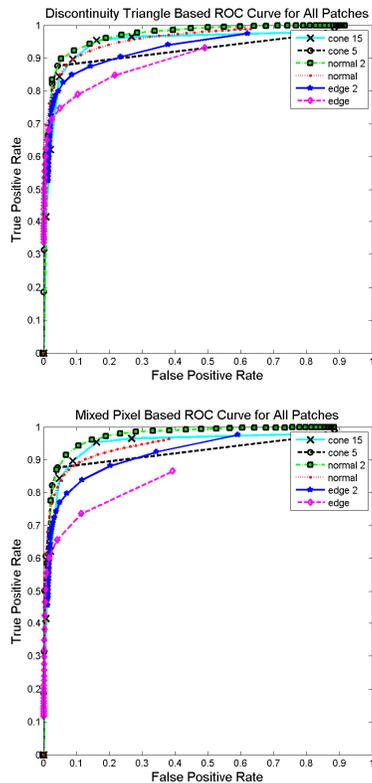


Figure 8. **Discontinuity triangle ROC curves (a) and mixed-pixel ROC curves (b) on all data patches.**

over actual object surfaces. Second, the boundary removal extensions substantially improve the performance of the edge and normal algorithms. In both examples both the edge and normal algorithm produce many false-positives scattered throughout the images while edge2 and normal2 algorithms avoid most such errors. Third, for small discontinuities, the edge2 algorithm performs the best among all algorithms. In Figure 9, which contains many small discontinuities, edge2 only produces a few false-positives on the screen of the computer while capturing most discontinuity artifacts, while all other algorithms including normal2 produce more false-positives cases on the table, the screen and the surface of the desktop computer. In Figure 10, only edge2 and cone5 can detect the small horizontal discontinuity produced by the bottom flange of the I-beam without producing many false-positive cases on actual beam surfaces, while the normal2 algorithm produces many false negatives at that small discontinuity and other algorithms produce many false-positives while detecting the small discontinuity.

Several conclusions can be drawn from above discussions. Small discontinuities are challenging for all these algorithms, because all ROC curves indicate that the performance of the algorithms is worse on

patches with more small discontinuities such as the laboratory patches and bridge patches with small depth discontinuities at the edges of the steel I-beams (bridge patches 2, 3, and 6). Second, on the indoor and outdoor patches, the performance of the algorithms does not show any important difference. Hence, we can deduce that whether a scan is taken indoors or outdoors does not essentially influence the discontinuity detection operation. Third, the boundary erosion variant of the normal-based algorithm is robust to the application environments, because on most patches it performs the best among all algorithms, and even in those exceptions, the performance is close to that of the best one.

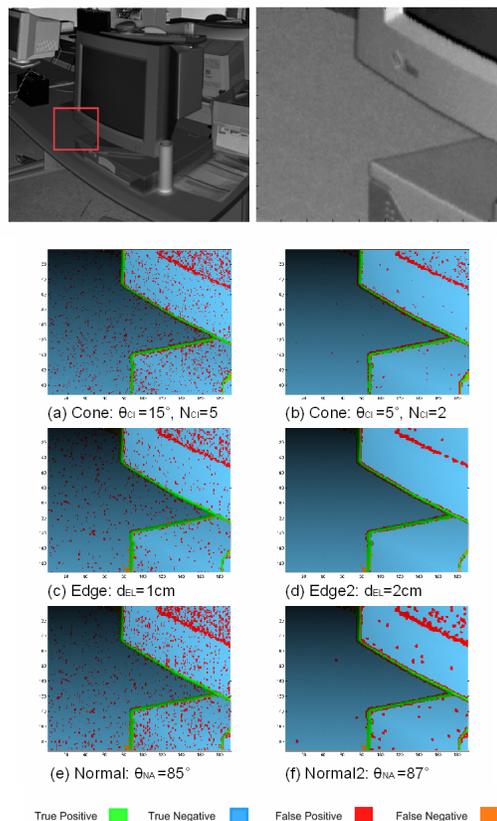


Figure 9. **Comparison of algorithm performance for patch 1 of the laboratory scan (shown in first row with close-up).**

5. Summary and future work

In this research, we have quantitatively compared the performance of several algorithms for detecting depth discontinuities and mixed pixels in 3D data. We show how depth discontinuity detection algorithms can be converted to mixed pixel detection algorithms and vice versa. The algorithms were evaluated using test patches extracted from a variety of scenes. Patches

and scene types were selected to ensure a full spectrum of depth discontinuity scales (in terms of depth and extent) as well as different surface types in an effort to make the comparative analysis as comprehensive and unbiased as possible. The results indicate that while no algorithm performs exceptionally well, filtering based on surface normal followed by a single boundary removal step (the normal2 algorithm) performs the best overall among the algorithms evaluated. We confirmed that discontinuity size is one of the major factors influencing the discontinuity detection, and that the performance of the selected algorithms is roughly the same for indoor and outdoor cases.

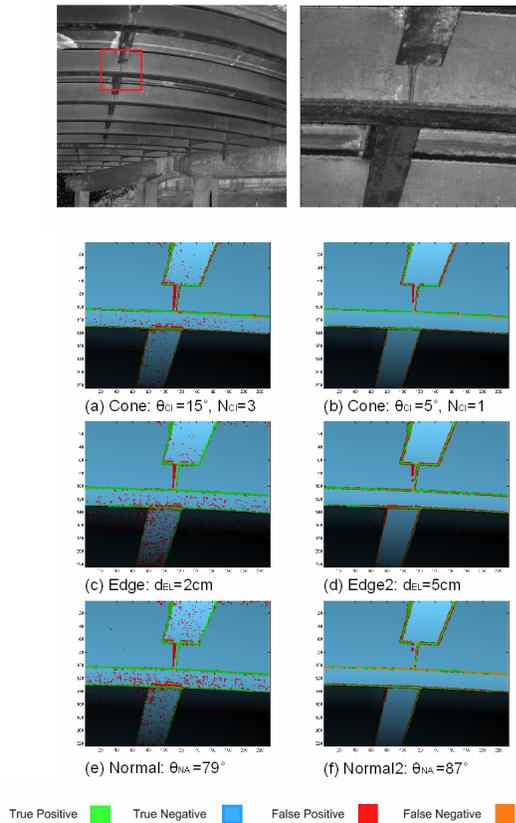


Figure 10. Comparison of algorithm performance for patch 2 of the bridge scan (shown in first row with close-up).

Our future work on this subject includes several areas of exploration. First, we intend to create a standard test-bed for mixed pixel and depth discontinuity detection. Such a test-bed would allow different sensors to be evaluated using a consistent data set. Similarly, such a testbed would enable the generation of a standard data set against which other researchers and developers can evaluate their algorithms. Second, we are planning on working with the ASTM Committee E57 on 3D Imaging Systems to develop standardized evaluation procedures for mixed

pixels and other data artifacts [17]. Third, we hope to mathematically relate the existing theory of the mixed pixel effect (e.g. works in [8]) with the performance characteristics of the discontinuity detection algorithms. A better understanding of this relationship could lead to improved algorithms in the future. Finally, we plan to analyze the impact of mixed-pixel removal on domain independent as well as domain-specific algorithms and applications.

References

- [1] T. Kersten, H. Sternberg, and E. Stiemer, "First Experience with Terrestrial Laser Scanning for Indoor Cultural Heritage Applications using Two Different Scanning Systems," in *Panoramic Photogrammetry Workshop*, Berlin, Germany, 2005.
- [2] G. Chris, B. Frank, H. Daniel, L. Edward, P. Kuhn, and A. Burcu, "Combining Reality Capture Technologies for Construction Defect Detection: A Case Study," 2003.
- [3] W. Boehler, M. B. Vicent, and A. Marbs, "Investigating Laser Scanner Accuracy," in *the XIXth CIPA Symposium Antalya*, Turkey, 2003.
- [4] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher, "An experimental comparison of range image segmentation algorithms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, pp. 673-689, 1996.
- [5] D. F. Huber and M. Hebert, "Fully automatic registration of multiple 3D data sets," *Image and Vision Computing*, vol. 21, pp. 637-650, 2003.
- [6] Zoller+Fröhlich, "Technical data IMAGER," Z. F. I. T. Specification, Ed., 2005.
- [7] M. K. Reed and P. K. Allen, "3-D Modeling from Range Imagery: An Incremental Method with a Planning Component," *Image and Vision Computing*, vol. 17, pp. 99-111, 1999.
- [8] B. Liu, M. Adams, W. S. Wijesoma, and J. Ibañez-Guzmán, "Range error detection caused by occlusion in non-coaxial LADARs for scene interpretation," *Journal of Robotic Systems*, vol. 22, pp. 549-567, 2005.
- [9] InnovMetric, "Polyworks Reverse Engineering Environment Version 9.0.2," 2005.
- [10] Zoller+Fröhlich, "LFM Register 3.80 Manual," Zoller+Fröhlich, 2005.
- [11] M. Hebert and E. Krotkov, "3-D measurements from imaging laser radars: how good are they?," 1991, pp. 359-364 vol.1.
- [12] M. D. Adams and P. J. Probert, "The interpretation of phase and intensity data for AMCW light detection sensors for reliable ranging." vol. 15: Sage Publications, Inc., 1996, pp. 441-458.
- [13] Y. Cang and J. Borenstein, "Characterization of a 2D laser scanner for mobile robot obstacle negotiation," 2002, pp. 2512-2518.
- [14] J. Tuley, N. Vandapel, and M. Hebert, "Analysis and Removal of Artifacts in 3-D LADAR Data," in *IEEE International Conference on Robotics and Automation*, 2005.
- [15] A. D. Sappa and M. Devy, "Fast range image segmentation by an edge detection strategy," 2001, pp. 292-299.
- [16] Inivis, "AC3D 6.1," 2007.
- [17] ASTM, "Committee E57 on 3D Imaging Systems," West Conshohocken, 2007, p. <http://www.astm.org/COMMIT/SUBCOMMIT/E57.htm>