DSD: Depth Structural Descriptor for Edge-Based Assistive Navigation

David Feng, Nick Barnes, Shaodi You
Data61, CSIRO; RSE, Australian National University
{david.feng, nick.barnes, shaodi.you}@data61.com.au

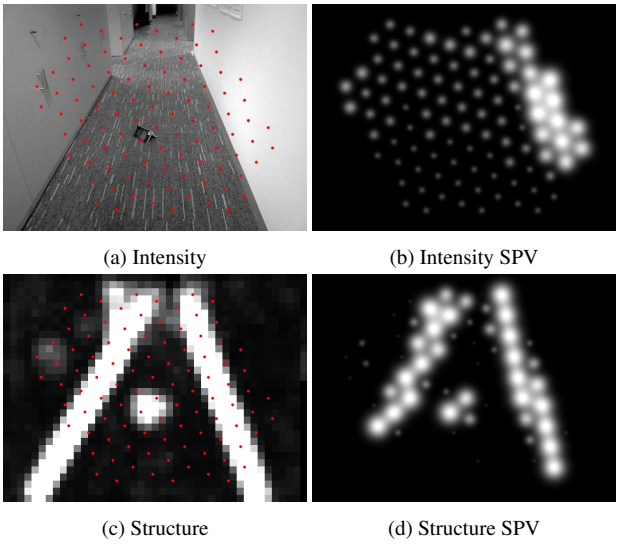
Abstract

Structural edge detection is the task of finding edges between significant surfaces in a scene. This can underpin many computer vision tasks such as sketch recognition and 3D scene understanding, and is important for conveying scene structure for navigation with assistive vision. Identifying structural edges from a depth image can be challenging because surface structure that differentiates edges is not well represented in this format. We derive a depth input encoding, the Depth Surface Descriptor (DSD), that captures the first order properties of surfaces, allowing for improved classification of surface geometry that corresponds to structural edges. We apply the DSD feature to salient edge detection on RGB-D images using a fully convolutional neural network with deep supervision. We evaluate our method on both a new RGB-D dataset containing prosthetic vision scenarios, and the SUNRGBD dataset, and show that our approach produces improved performance compared to existing methods by 4%.

1. Introduction

Edge and contour extraction are classical problems of computer vision, that have received much research attention over a long history [4, 6]. Structural edges are key to understanding a visual scene. This dates from classical approaches such as by Lowe [19] that link edge finding to perceptual organization to understand 3D structure for object recognition, but can also support contemporary ideas of indoor scene understanding (e.g., [24]). Further, finding such structure is important for personal mobility [16] with retinal implants, where the bandwidth of image information that can be represented per frame is quite restricted.

Retinal implants aim to restore visual function that is lost due to degenerative diseases through the electrical stimulation of surviving retinal cells. These devices normally convey incoming light intensity from a head mounted camera to the user, but the resulting perception is constrained by the stimulation process, allowing only low resolution and dynamic range images to be conveyed. Existing devices have a resolution on the order of tens or hundreds of display elements, with less than ten discernable brightness levels for each display element [12]. These display constraints can lead to difficulty in interpreting the content of the display, making it easy to miss details in the environment such as room boundaries and small or low-contrast trip hazards in front of the user. Figure 1b shows a simulation of what a patient might see with a 98 electrode device. This constrained perception motivates the application of computer vision techniques to ensure key information is conveyed to the user.



(a) Intensity (b) Intensity SPV

(c) Structure (d) Structure SPV

Figure 1. Comparison of scene brightness versus structural cues rendered using simulated prosthetic vision. (a) Intensity image with sampling locations marked in red; (b) prosthetic vision rendering of scene brightness; (c) Low level structural edge map with sampling locations marked in red; (d) prosthetic vision rendering of structural edges.

In particular, structural edge detection is well suited to drawing attention to structurally important locations for a retinal implant user attempting to safely navigate their way through the environment. Ground-plane detection has previously been shown to be effective in this scenario [21].

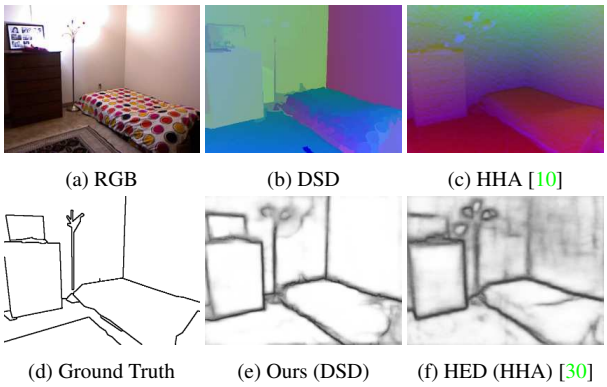
However, structural edge detection methods are more suited to conveying general scene structure for the user to interpret, which is crucial for performing tasks such as self-orientation and building a mental map of the environment. Figure 1d shows a simulated prosthetic vision display with a visual representation based on structural edge detection. The boundary between the ground and wall in the second row provides a useful cue for self orientation, while the presence of the chair and the overhanging obstacle are clearly conveyed. Overhanging objects are known to be difficult for current visual aids.

While there exist many recent RGB-D edge detection methods, these methods focus on contour extraction rather than structural edge detection, identifying objects and suppressing edges within them (e.g.,[30]). The extraction of closed contours reduces noise from texture compared to low level methods and improves results on standard datasets, however, this approach also suppresses structural edges that are internal to objects. Edges such as the corner between two walls could be regarded as internal to the wall object, but are important to indoor scene understanding [24], and for mobility with assistive devices. Similarly, a large object with a leading edge may protrude substantially from the object, posing a collision risk.

In this paper, we revisit finding structural edges that are significant for 3D scene understanding and mobility. These include the silhouette, but also internal ridges and troughs. These are identified by Banshal *et al*. [2] as occlusion boundaries and surface normal discontinuities. Further, we note the effectiveness of 3D scene understanding using commodity RGB-D sensors (e.g., [26]) and the ability of RGB-D input to aid in differentiating edges of scene structure from edges such as shadows that are only related to appearance. Hence, we investigate improving the recovery of structural edges using RGB-D input. In particular, we contribute a depth input encoding that is suited to finding structural edges in RGB-D images that relate to the important aspects of scene structure. We present an end-to-end fully convolutional CNN approach, incorporating this feature. Finally, we contribute a dataset of 200 RGB-D images of real scenes with ground truth structural edges. The scenarios included cover particularly mobility for prosthetic vision.

2. Related Work

Finding salient edges in an image has background literature in edge and contour detection, and methods for visual saliency. Given the breadth of this research, we focus more on recent methods for CNN-based contour detection that yield the strongest relevant results.



(a) RGB (b) DSD (c) HHA [10]

(d) Ground Truth (e) Ours (DSD) (f) HED (HHA) [30]

Figure 2. HED [30] output when trained on our DSD feature compared to the standard HHA [11] depth feature. Note that our feature provides a better representation of the scene structure, e.g. the corner between the two walls at the right of the image, and is significantly less affected by sensor noise, allowing the CNN to better model edge structure and thus produce a more accurate edge map.

2.1. Edge and contour detection

Edge detection is well-studied. Early approaches directly detected local appearance including classical approaches such as Sobel [13] and the highly successful Canny detector [4]. Classical work by Lowe [19] linked concepts of perceptual organization to algorithms for finding lines, in order to understand 3D structure for object recognition. Extending on this, [28] found 'structural saliency' (objects/regions) based on curvature.

More recent work has found closed contours of objects using combined edge and region methods. For example, Levinshtein *et al*. [17] optimally grouped superpixels to find enclosing salient contours. Dollar and Zitnick [6] show strong results for detecting salient edges by learning using a structured forest and manually designed features. These are sometimes extended to RGB-D data. Ren and Bo [29] train sparse code gradients to detect contours. This approach is extended to RGB-D, and shows a significant performance boost by detecting contours by adding depth data to RGB. Similarly, Dollar and Zitnick [6] show a significant boost incorporating depth data. Raskar *et al*. and Schäfer *et al*. [23] explicitly used depth information to suppress texture intensity edges by requiring co-occurrence between depth and intensity edges [22, 23].

CNN approaches The excellent results yielded by CNNs for high level vision tasks such as object detection have led to revisiting contour detection. Early CNN contour detection approaches include Ganin and Lempitsky [8], Kivinen *et al*. [14], and Shen *et al*. [25]. State-of-the-art results have come from papers that transfer deep learning features from high-level vision tasks to low-level vision problems, includ-

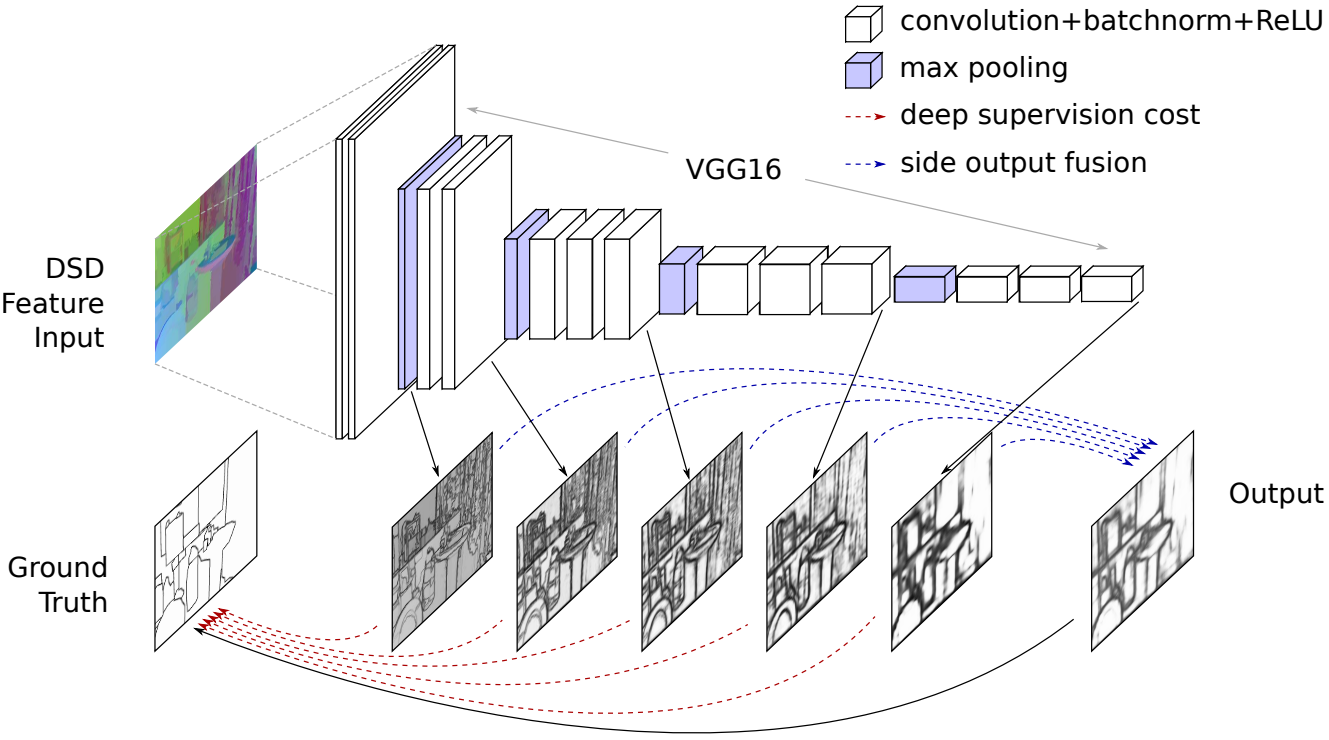


Figure 3. An overview of our edge detection system. Our DSD encoding of the depth map is the input to fully convolutional VGG16 network with deep supervision, as in [30]. We add a batch normalization layer after every convolutional layer to speed up convergence.

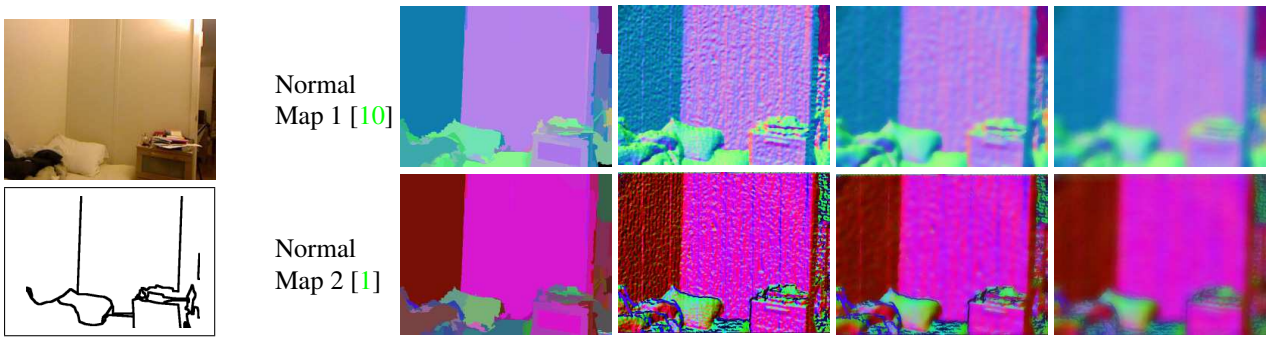
ing edge detection. Both Bertasius *et al*. [3] and Xie and Tu [30] derived contour detection from a base of VGGNet [27]. Using a pre-trained, trimmed VGGNet, [30] incorporate deep supervision to enforce meaningful output from intermediate layers as well as the final layer of the fully convolutional network. This has become the baseline model for deep edge detection, with many subsequent papers proposing improvements to the architecture. Liu and Lew [18] propose relaxed deep supervision, using the output of off-the-shelf edge detectors to guide the learning of intermediate layers in a coarse-to-fine-paradigm. Kokkinos [15] fine tunes the loss function and explicitly incorporates multiple scales as well as global information. Maninis *et al*. [20] include a novel sparse boundary representation for hierarchical segmentation, and show that learning boundary strength and orientation improves results. Yang *et al*. [31] learn to detect contours with a fully convolutional encoder decoder network, which generalizes well to unseen object categories. These methods focus on RGB edge detection, and in particular do not consider alternate encodings of the depth data to improve detection of structural edges.

Depth Image Encoding for Edge Detection Exploiting CNN's for RGB-D depth edges is less common. In the context of detection and scene segmentation, Gupta *et al*. [11] propose the HHA geocentric embedding for depth images to perform RGB-D contour detection. They encode disparity, height above ground, and angle with gravity into the edge learning framework of [6] and show that it produces improved results over naively using depth. The HHA feature is the current state-of-the-art depth representation for CNN-based edge detection, with subsequent CNN-based edge detectors all incorporating this feature when operating on RGB-D input [30, 3, 25]. While this feature is useful for object detection, it is less suited to structural edge detection since it does not incorporate a full model of curvature. In particular the HHA feature does not directly represent vertical joins between two surfaces, such as the boundary between adjacent walls, or the corner on a wardrobe (see Figure 2). These edges are a common occurrence in indoor scenes and are usually salient.

3. DSD Feature

In this section we introduce our proposed depth feature, the Depth Surface Descriptor (DSD), which aims to provide a minimal encoding of the depth information in the scene that captures the distinguishing surface geometry of structural edges, and suppresses sensor noise and other non-edge structure.

We are interested in depth edges as opposed to appear-



(a) RGB + GT (b) $\tilde{\mathcal{N}}$ (c) Unfiltered ($\tilde{N}$) (d) $\sigma = (20, 200)$ (e) $\sigma = (40, 400)$

Figure 4. Visualization of of our surface patch mapping function $\tilde{\mathcal{N}}$ and aggressive bilateral smoothing of pointwise normals computed using two different methods [10, 1]. Our method mitigates noise within surface patches while maintaining contrast between regions bordering structural edges.

ance edges that are treated separately in our architecture. Intrinsically to a surface, depth edges arise for only two reasons, a depth discontinuity in the surface (*i.e.* a step edge), or a first order discontinuity in the surface (*i.e.* a crease edge). To develop a structural edge detector, we require that it can identify these phenomena regardless of the nature of the appearance or embedding of the surface.

Classically, Gaussian curvature encodes the intrinsic curvature of a surface regardless of embedding [9]. Hence, two principal curvatures are all that is required to encode a surface. This information can be represented as a Gauss map $N : X \rightarrow S^2$ of surface normals, which maps a point $\mathbf{p} \in X$ on an input surface $X \subset \mathbb{R}^3$ to the point $\mathbf{n} \in S^2$ on the unit sphere corresponding to the surface normal at $\mathbf{p}$. Since we seek a minimal encoding, we use an approximation function $\mathcal{N}$ defined as follows:

$$\mathcal{N}(N(\mathbf{p})) = \left(\cos^{-1}\left(N(\mathbf{p}) \cdot \mathbf{u}\right),\ \cos^{-1}\left(N(\mathbf{p}) \cdot \mathbf{v}\right)\right) \quad (1)$$

where $\mathbf{u}, \mathbf{v} \in S^2$ are fixed and orthogonal. $\mathcal{N}$ is injective, i.e. $\mathcal{N}(\mathbf{a}) = \mathcal{N}(\mathbf{b}) \rightarrow \mathbf{a} = \mathbf{b}$, because

$$\cos^{-1}(\mathbf{a}\cdot\mathbf{u}) = \cos^{-1}(\mathbf{b}\cdot\mathbf{u}) \rightarrow \mathbf{a}\cdot\mathbf{u} = \mathbf{b}\cdot\mathbf{u} \rightarrow \mathbf{a} = \mathbf{b} \quad (2)$$

since $\mathbf{u} \cdot \mathbf{u} = 1$, and since we are only interested in the range $[0, \pi]$ where $\cos^{-1}$ is bijective. Therefore $\mathcal{N}$ does not reduce the discriminability of the representation.

To find edges, we seek change in the curvature, hence a spatial operator that finds such change is required over some local region R.

$$E_N(\mathbf{p}) = \int_{\mathbf{p}\in R} f\left(\mathcal{N}(N(\mathbf{p}))\right) d\mathbf{p} \quad (3)$$

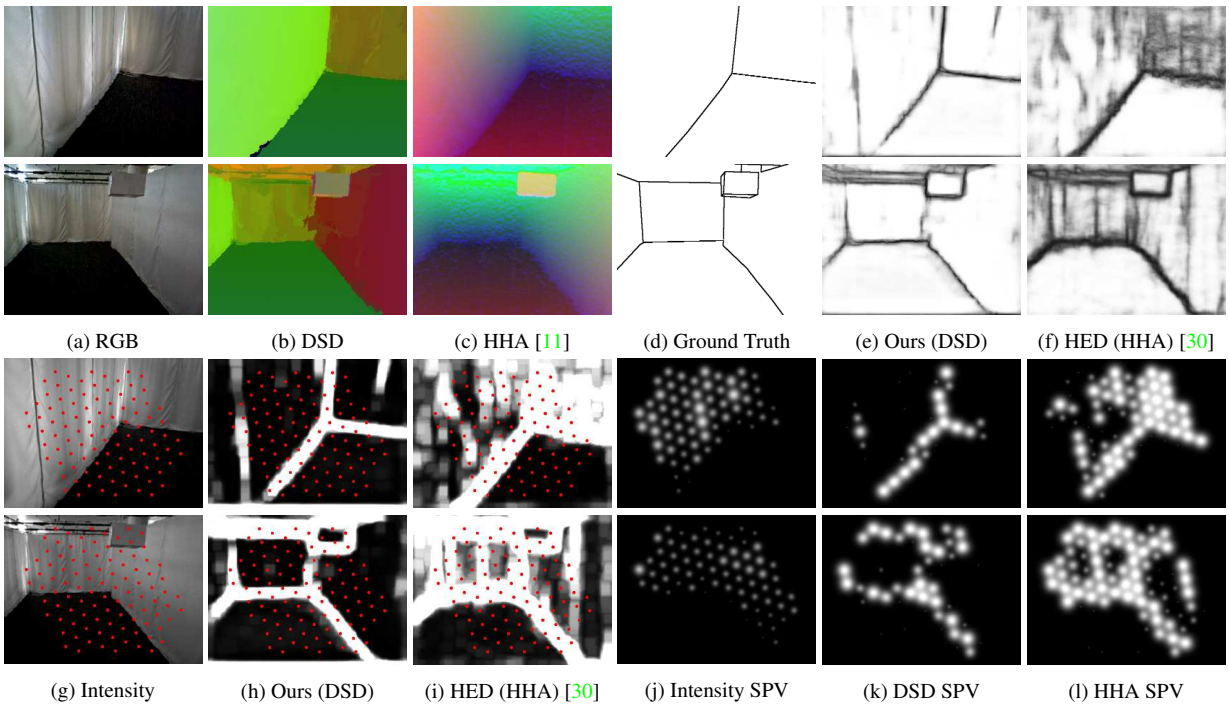
As we seek a minimal encoding of the surface, then theoretically we could simply take the depth map of the scene. In this case, f can also compute surface normals as required. However, in practice, depth sensor readings have a component of noise, and care must be taken in the computation of the surface normals. Hence, both parameters of surface normals need to be represented directly. Further, if we minimally code surface normals with two parameters in 3D, we lack the original depth data, and will be unable to identify step edges that do not result in a visible change of surface normal (e.g., stairs viewed from directly above). Hence we propose a minimal encoding of depth by incorporating the disparity map D

$$E_{N,D}(\mathbf{p}) = \int_{\mathbf{p}\in R} F\left(\mathcal{N}(N(\mathbf{p})), D(\mathbf{p})\right) d\mathbf{p} \quad (4)$$

We can hand-craft such an operator directly, however its construction is not simple. It must account for all surface shapes, such as two corrugated iron fences that abut at an angle, or a corner in rippled curtains (see Figure 5). In addition, sensor noise is complex and scale is problematic, in short, "mathematics has nothing to say about scale" - O. Faugeras [7]. A rippling curtain does have changing curvature, but yet it is the joint between the surfaces that would be considered structurally salient by humans for most tasks (see Figure 5).

Hence we take the approach of forming a minimum surface encoding and using a deep CNN that takes semantic information into account to form a spatial operator to detect structurally salient edges. An advantage of deep CNNs for such problems is that the encoding weighs depth values from the entire image and so supports a multi-scale framework. Further, contour processing generally employs a broader region of support to suppress noise as well as a local gradient operator to find the edge.

Our minimum encoding consists of absolute depth and surface normals. Next we present how we compute stable surface normals.



(a) RGB (b) DSD (c) HHA [11] (d) Ground Truth (e) Ours (DSD) (f) HED (HHA) [30]

(g) Intensity (h) Ours (DSD) (i) HED (HHA) [30] (j) Intensity SPV (k) DSD SPV (l) HHA SPV

Figure 5. (a-f) Challenging examples from our dataset, with output from our method and HED-HHA. Note the ripples in the curtains that produce high local surface normal variation, illustrating the importance of scale for structural edge detection. (g-l) Prosthetic vision inputs and renderings of the scene from intensity, our method, and HED-HHA, with sampling locations shown in red. SPV denotes simulated prosthetic vision renderings. The errors due to surface normal noise in HED-HHA (l) can make it difficult for a prosthetic vision user to interpret the scene when performing navigation. Our method (k) reduces noise, providing a clearer depiction of scene structure.

3.1. Region-Based Normal Computation

Sensor noise has an adverse effect on the learning process, slowing convergence and reducing accuracy with a limited set of training examples. The effects of sensor noise are magnified in surface normal estimations from depth sensors, since surface orientation is a first order property of the sensor output. As shown in Figure 5, a seemingly flat surface can have an wide distribution of surface orientations due to a small amount of noise in the depth reading. Thus the noisy discretized normal map $\tilde{N}$ computed from sensor data is a poor approximation to N and does not accurately express the surface structure of the scene.

Filtering the image can address this issue to an extent by smoothing spurious local normal variations, but still leaves a considerable amount of noise in the input. Furthermore, over filtering will blur the structural boundaries of the scene, reducing edge localization accuracy, as shown in Figure 4. Due to the unknown required scale of surface curvature, filter size cannot be defined *a priori*.

We reduce the effect of sensor noise by performing region-based smoothing of the point-wise normal image. First, we over-segment the image into surface patches using the Mean Shift algorithm [5]. Let $P(\mathbf{p}) \subseteq X$ map a point $\mathbf{p}$ to its containing surface patch. Then the region-aggregated normal map is given by:

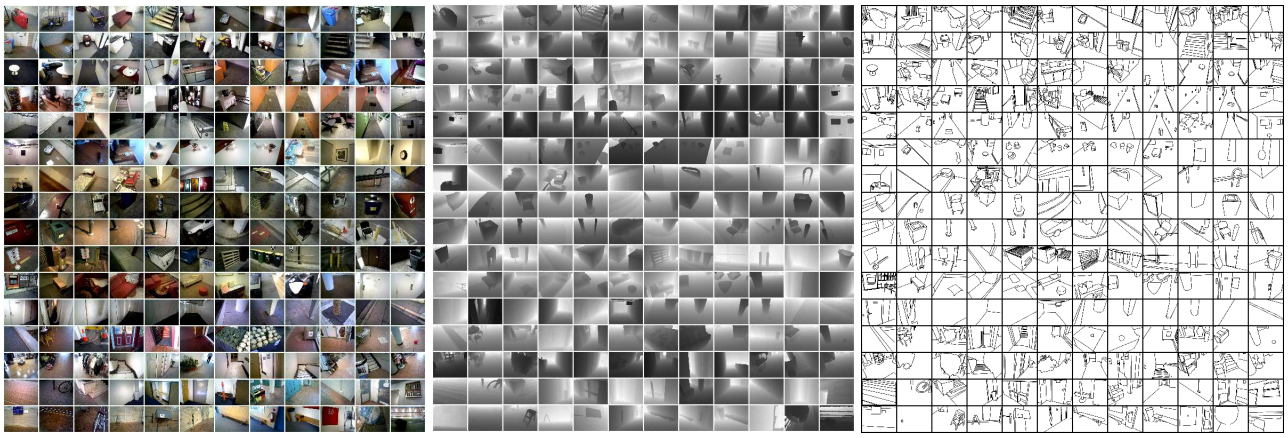
$$\tilde{\mathcal{N}}(N(\mathbf{p})) = \frac{1}{|P(\mathbf{p})|} \int_{\mathbf{x} \in P(\mathbf{p})} \mathcal{N}(\tilde{N}(\mathbf{x})) d\mathbf{x} \qquad (5)$$

This maps regions with consistent surface orientation to a single representative normal value, smoothing normals within a surface while maintaining contrast between surfaces that border structural edges.

3.2. Normal Computation Frame of Reference

The ground orientation is a key piece of semantic information in many scenes, as it provides an absolute reference point for object surfaces in the scene. For example, a boundary between the ground and a vertical surface, or between two walls, may be more likely to be labeled as salient, particularly for tasks such as mobility. We parameterize $\mathcal{N}$ with respect to the ground plane by fixing the first coordinate axis $\mathbf{u}$ to the inferred direction of gravity. To provide a stable reference frame, set the second axis $\mathbf{v}$ to be orthogonal to both the camera axis $\mathbf{z}$ and $\mathbf{u}$.

$$\mathbf{v} = \mathbf{u} \times \mathbf{z} \qquad (6)$$



(a) RGB (b) Depth (c) Ground Truth

Figure 6. Our proposed dataset, containing a range of assistive vision scenarios.

This increases the amount of information encoded in our minimal representation with no representation cost, providing a stable reference frame from which further relationships between edges and scene structure can be inferred.

4. Experiments

In this section we detail the implementation of the DSD CNN, and describe the experiments run to evaluate the effectiveness of the encoding.

4.1. Implementation

We use VGG-16 as the base system for testing the DSD encoding. Since the main contribution is the DSD encoding, the selection of VGG-16 is to provide fair comparison of our encoding with existing methods. We trim the fully connected layers of VGG and incorporate deep supervision by adding a side output to the last convolutional layer of each of the five VGG blocks, as in [30]. The network produces one fusion output which linearly combines the side outputs using learned weights. We add a batch normalization layer immediately after each convolutional layer, to help speed up convergence.

We merge the output depth edge maps with rgb maps from the HED architecture [30] in order to assess the contribution of the system as part of an RGB-D edge detector. When merging depth edge with rgb edge maps, we first take the product of the fusion output with all the up-sampled side outputs, since this produces the best results. We observe that the later side outputs produce more semantically meaningful output with some false positives due to blurry edges from up-sampling, whereas the earlier side outputs have excellent edge localization but a high number of false positives due to incorrect edge detections within non-boundary regions. Thus taking the product of all layers reduces false positives while ensuring that the meaningful edges retain a high response. Multiplying the side outputs in this way increases F-score but decreases average precision. However, when merging with the rgb saliency map, average precision is not reduced.

We tune the hyper-parameters of the network using the method in [30], using deviations of the F-score on the validation set as a measure of convergence. We select the following hyper-parameter values for our experiments: image size 500×500 mini-batch size $= 10$, learning rate $= 1e5$, momentum $= 0.9$, weight decay $= 0.0002$, training iterations $= 15000$, with learning rate divided by 10 every 5000 iterations.

We fix the coordinate system of the surface normal map N as follows. We set u as the inferred direction of gravity and v as the intersection between the camera plane and the plane define by u. This provides a stable reference frame for surface orientation measurements, allowing the system to learn extrinsic priors relating to structural edge placement.

4.2. Datasets

We evaluate our method on the SUNRGBD dataset, which contains 10335 RGB-D image pairs taken with a variety of commodity depth cameras. As in [6], we convert the segmentation ground truth to edge maps using [10]. Note that SUNRGBD is a superset of the NYU dataset that existing methods use for evaluation, and thus provides a better indication of model performance. We split the SUNRGBD dataset into 6201 training, 2067 validation and 2067 test images.

We also introduce a new dataset, which contains 200 RGB-D image pairs with hand-labeled ground truth. The images were taken with an Asus Xtion Pro depth camera and represent a wide variety of indoor environments, particularly those which would be encountered within robotic

Method	ODS	OIS	AP
HED HHA	.647	.668	.570
Ours DSD	**.678**	**.712**	**.653**
HED RGB	.641	.679	.591
HED RGB+HHA	.679	**.729**	.676
Ours RGB+DSD	**.685**	**.729**	**.685**

Table 1. Results on our new depth edge dataset, reflecting performance on a variety of prosthetic vision navigation scenarios.

Method	ODS	OIS	AP
HED HHA	.615	.634	.548
Ours DSD	**.630**	**.652**	**.577**
HED RGB	.629	.652	.545
HED RGB+HHA	.649	.672	.606
Ours RGB+DSD	**.652**	**.676**	**.610**

Table 2. Results on the SUNRGBD Dataset.

grasping or prosthetic vision mobility tasks. Ground truth was provided by a group of volunteers using custom annotation software. Labelers were asked to mark significant structural boundaries in the scene. The dataset is shown in Figure 6, and has been made publicly available [1].

We do not perform training on our dataset due to its small size. Rather, we use it to evaluate the generalization ability of the learned edge maps on novel scenes likely to be encountered during mobility tasks.

4.3. Comparison with Existing Methods

We compare the surface representation capacity of the DSD feature with the state-of-the-art HHA feature for structural edge detection, by evaluating the quality of learned edge maps on the two features using the HED architecture [30]. Since our main contribution, the DSD input encoding, is agnostic of the learning framework, we do not provide further comparison with different deep learning architectures. HED is the state-of-the-art base architecture for edge detection, and any optimizations to the framework [18, 15] would likely improve learned results from our feature.

Evaluation Metrics We use three standard performance metrics for edge detection evaluation. These are the F-score for the best threshold over the dataset (ODS), best per image threshold (OIS), and average precision (AP).

5. Results and Discussion

To evaluate the suitability of DSD and HHA for for prosthetic vision navigation scenarios, as well as cross-dataset generalisation performance, we test the pretrained networks on our new RGB-D edge dataset. The results are shown in Table 1 and Figure 7. Our method outperforms the HHA-based system for depth-only edge detection, demonstrating that the DSD feature provides a more general representation of surface structure, and that it is more suitable for detecting scene structure for navigation with prosthetic vision. Thus, our method gives a clearer indication of the structure of the environment for navigation scenarios than the HHA-based system, as shown in Figure 5.

On the SUNRGBD dataset our method gives the highest ODS, OIS and AP scores for the depth-only methods, as seen in Table 2 and Figure 8. This demonstrates that our DSD input encoding makes available to the learning framework a more discriminative surface representation

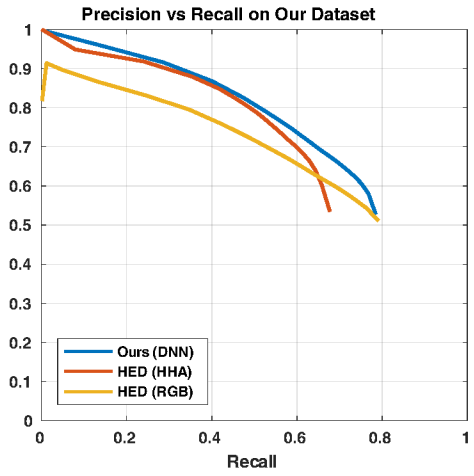


Figure 7. PR curve on our new dataset.

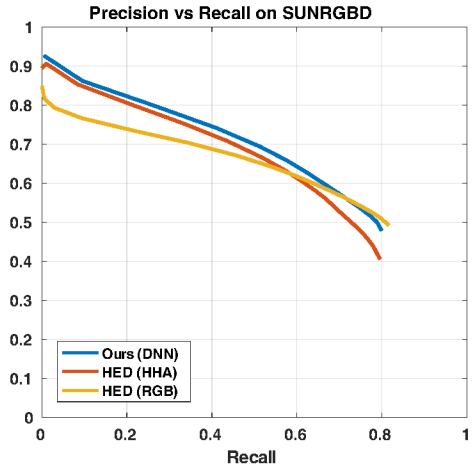
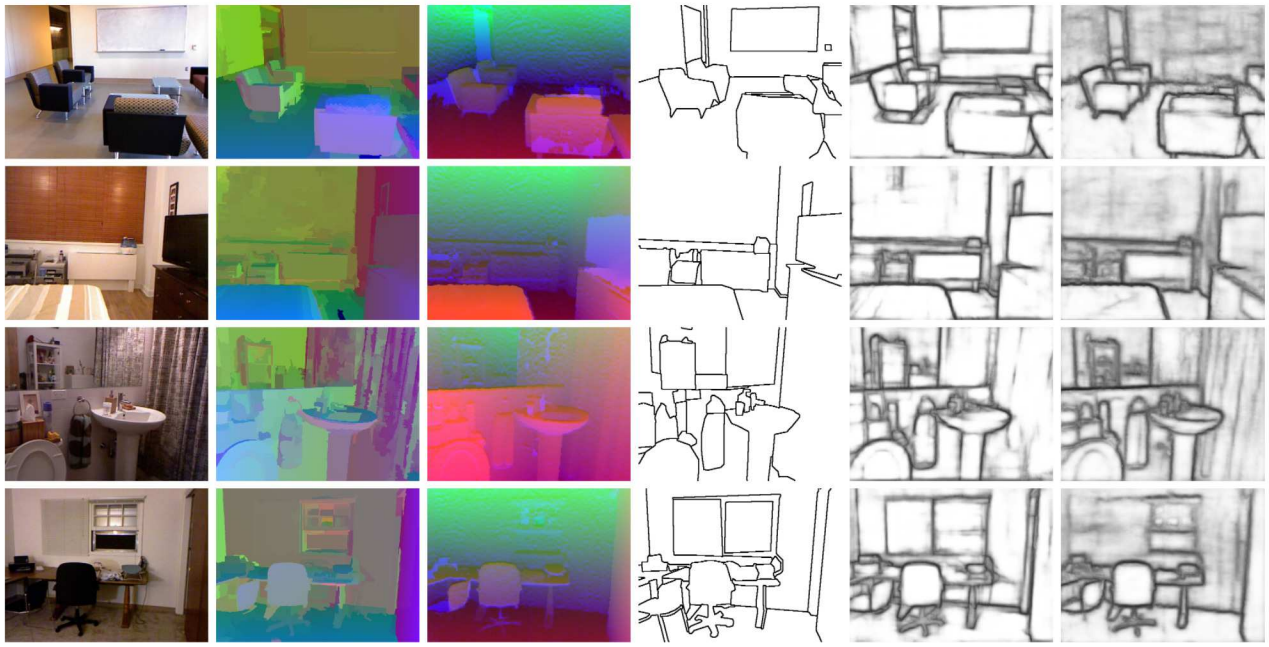


Figure 8. PR curve on SUNRGBD.

[1] http://users.cecs.anu.edu.au/ u4673113/dsd.html



(a) RGB (b) DSD (c) HHA [11] (d) Ground Truth (e) Ours (DSD) (f) HED (HHA) [30]

Figure 9. Example outputs from the SUNRGBD dataset. Our DSD feature provides a cleaner and more descriptive representation of the boundaries between underlying surfaces, which results in an improved final edge map compared to HHA.

than HHA, enabling more effective classification of edge structure. Note that this dataset features semantic ground truth and therefore internal edges of objects are not marked, despite many of these edges, such as protruding edges from objects, being important for navigation scenarios. This issue is addressed in our dataset, in which all structural edges relevant for assistive navigation, including internal edges, are labelled.

From Figure 8, we see that while HED-RGB overall does not perform as well as depth, it performs relatively well in the high recall region. Thus we see from the graph that the two HED methods can compensate for each other's performance. However, since our curve is mainly above the curve for HED-RGB, there is less potential for performance improvement from a naive combination with HED-RGB. Despite this, our method obtains superior performance when merged with HED-RGB as shown in Table 2. The focus of this work is on the depth representation, and further investigation of merging depth and RGB edge maps would increase the RGB-D performance of our system.

Figure 9 shows some example edge outputs generated by our method. Generally our DSD encoding provides a more effective expression of surface geometry, allowing for a cleaner separation of edge and non-edge structure. For example, in the second last row our method correctly suppresses the depth texture of the shower curtain, demonstrating the effectiveness of our encoding when combined with learned high level information.

5.1. Future Work and Assistive Vision Application

The results demonstrate that our method is well suited for extracting salient edge structure relevant to scene understanding with assistive vision. Future work will measure the effectiveness of our approach compared to standard methods during practical implant use. The authors, in collaboration with the Centre for Eye Research Australia and The Bionics Institute (Melbourne, Australia) will be undertaking clinical trials involving three implanted patients using a suprachoroidal retinal prosthesis this year. This will include a focus on orientation and mobility. These clinical trials will provide an opportunity for further evaluation of our proposed method for the target application of navigation with assistive vision.

6. Conclusion

We have presented a new depth encoding, the DSD feature, for salient structural edge detection. The DSD feature captures the first order properties of surfaces, allowing for improved classification of surface geometry that corresponds to structural edges. We have incorporated this feature into a fully convolutional CNN framework to achieve state-of-the-art structural edge detection results on both a large scale existing dataset as well as a new RGB-D edge dataset.

References

[1] H. Badino, D. Huber, Y. Park, and T. Kanade. Fast and accurate computation of surface normals from range images. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3084–3091. IEEE, 2011. 4

[2] A. Bansal, A. Kowdle, D. Parikh, A. Gallagher, and L. Zitnick. Which edges matter? In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 578–585, 2013. 2

[3] G. Bertasius, J. Shi, and L. Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4380–4389, 2015. 3

[4] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 1, 2

[5] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002. 5

[6] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1841–1848, 2013. 1, 2, 3, 6

[7] O. Faugeras, J. Mundy, N. Ahuja, C. Dyer, A. Pentland, R. Jain, K. Ikeuchi, and K. Bowyer. Why aspect graphs are not (yet) practical for computer vision. *CVGIP: Image Understanding*, 55(2):212–218, 1992. 4

[8] Y. Ganin and V. Lempitsky. Nˆ 4-fields: Neural network nearest neighbor fields for image transforms. In *Asian Conference on Computer Vision*, pages 536–551. Springer, 2014. 2

[9] C. F. Gauss. *General investigations of curved surfaces of 1827 and 1825*. The Princeton University Library, 1902. 4

[10] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 564–571, 2013. 2, 4, 6

[11] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *European Conference on Computer Vision*, pages 345–360. Springer, 2014. 2, 3, 5, 8

[12] M. S. Humayun, J. D. Weiland, G. Y. Fujii, R. Greenberg, R. Williamson, J. Little, B. Mech, V. Cimmarusti, G. Van Boemel, G. Dagnelie, et al. Visual perception in a blind subject with a chronic microelectronic retinal prosthesis. *Vision research*, 43(24):2573–2581, 2003. 1

[13] J. Kittler. On the accuracy of the sobel edge detector. *Image and Vision Computing*, 1(1):37–42, 1983. 2

[14] J. J. Kivinen, C. K. Williams, N. Heess, et al. Visual boundary prediction: A deep neural prediction network and quality dissection. In *AISTATS*, volume 1, page 9, 2014. 2

[15] I. Kokkinos. Pushing the boundaries of boundary detection using deep learning. *arXiv preprint arXiv:1511.07386*, 2015. 3, 7

[16] M. Leo, G. Medioni, M. Trivedi, T. Kanade, and G. M. Farinella. Computer vision for assistive technologies. *Computer Vision and Image Understanding*, 154:1–15, 2017. 1

[17] A. Levinshtein, C. Sminchisescu, and S. Dickinson. Optimal contour closure by superpixel grouping. In *European Conference on Computer Vision*, pages 480–493. Springer, 2010. 2

[18] Y. Liu and M. S. Lew. Learning relaxed deep supervision for better edge detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 231–240, 2016. 3, 7

[19] D. Lowe. *Perceptual organization and visual recognition*, volume 5. Springer Science & Business Media, 2012. 1, 2

[20] K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. Van Gool. Convolutional oriented boundaries. In *European Conference on Computer Vision*, pages 580–596. Springer, 2016. 3

[21] C. McCarthy, J. G. Walker, P. Lieby, A. Scott, and N. Barnes. Mobility and low contrast trip hazard avoidance using augmented depth. *Journal of neural engineering*, 12(1):016003, 2014. 1

[22] R. Raskar, K.-H. Tan, R. Feris, J. Yu, and M. Turk. Non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. In *ACM transactions on graphics (TOG)*, volume 23, pages 679–688. ACM, 2004. 2

[23] H. Schäfer, F. Lenzen, and C. S. Garbe. Depth and intensity based edge detection in time-of-flight images. In *3DTV-Conference, 2013 International Conference on*, pages 111–118. IEEE, 2013. 2

[24] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient structured prediction for 3d indoor scene understanding. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2815–2822. IEEE, 2012. 1, 2

[25] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3982–3991, 2015. 2, 3

[26] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012. 2

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[28] S. Ullman and A. Sha'ashua. Structural saliency: The detection of globally salient structures using a locally connected network. 1988. 2

[29] R. Xiaofeng and L. Bo. Discriminatively trained sparse code gradients for contour detection. In *Advances in neural information processing systems*, pages 584–592, 2012. 2

[30] S. Xie and Z. Tu. Holistically-nested edge detection. *International Journal of Computer Vision*, pages 1–16, 2017. 2, 3, 5, 6, 7, 8

[31] J. Yang, B. Price, S. Cohen, H. Lee, and M.-H. Yang. Object contour detection with a fully convolutional encoder-decoder network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 193–202, 2016. 3