Which Edges Matter?

Aayush Bansal
RI, CMU
aayushb@cmu.edu

Adarsh Kowdle
Microsoft
adkowdle@microsoft.com

Devi Parikh
Virginia Tech
parikh@vt.edu

Andrew Gallagher
Cornell University
andrew.c.gallagher@gmail.com

Larry Zitnick
Microsoft Research
larryz@microsoft.com

Abstract

In this paper, we investigate the ability of humans to recognize objects using different types of edges. Edges arise in images because of several different physical phenomena, such as shadow boundaries, changes in material albedo or reflectance, changes to surface normals, and occlusion boundaries. By constructing synthetic photorealistic scenes, we control which edges are visible in a rendered image to investigate the relationship between human visual recognition and that edge type. We evaluate the information conveyed by each edge type through human studies on object recognition tasks. We find that edges related to surface normals and depth are the most informative edges, while texture and shadow edges can confuse recognition tasks. This work corroborates recent advances in practical vision systems where active sensors capture depth edges (e.g. Microsoft Kinect) as well as in edge detection where progress is being made towards finding object boundaries instead of just pixel gradients. Further, we evaluate seven standard and state-of-the-art edge detectors based on the types of edges they find by comparing the detected edges with known informative edges in the synthetic scene. We suggest that this evaluation method could lead to more informed metrics for gauging developments in edge detection, without requiring any human labeling. In summary, this work shows that human proficiency at object recognition is due to surface normal and depth edges and suggests that future research should focus on explicitly modeling edge types to increase the likelihood of finding informative edges.

1. Introduction

Over the past several decades, object recognition approaches have leveraged edge-based reasoning [30, 43] or gradient information [14]. The progression of the vision community is marked in some ways by fundamental advances related to capturing edges, be it the classical Canny edge detector [6] or the more recent SIFT [24] and HOG descriptors [28] that capture gradient information effectively. The important role of edges and gradients in computer vision does not come as a surprise to a student of pyschovisual processes: the importance of using edges is also supported by nature; the visual systems of mammals contain cells with gradient-based Gabor-like responses [11].

Though fundamental to visual processing, edges or gradients correspond to several different physical phenomena (Figure 1). For instance, object boundaries typically result in edges due to texture or albedo differences between objects. But an edge may also correspond to texture or albedo changes occurring *within* a single object. Lighting variations due to shadows or changes in surface normals also create edges. In fact, these phenomena are often correlated: depth discontinuities, lighting variations and object boundaries can coincide. This diverse nature of edges raises an important question: are some types of edges (i.e. edges resulting from certain phenomenon) more useful for object recognition than others? If so, which ones? In this paper, we study this question in the context of the human visual system, and then use these findings to gauge progress in computer edge detection by measuring by how well algorithms find edges that contain the most information for (human) object recognition.

Different applications benefit from localizing different types of edges. For instance, image segmentation approaches [1] attempt to find edges corresponding to object boundaries. Edges corresponding to depth discontinuities are used by stereo techniques. Texture classification [29] on the other hand relies on the internal gradients or edges of objects. This raises another relevant question: which types of edges are being found by standard automatic edge detection algorithms in computer vision?

In this paper, we explore the question of which physical phenomena results in the most informative edges for ob-

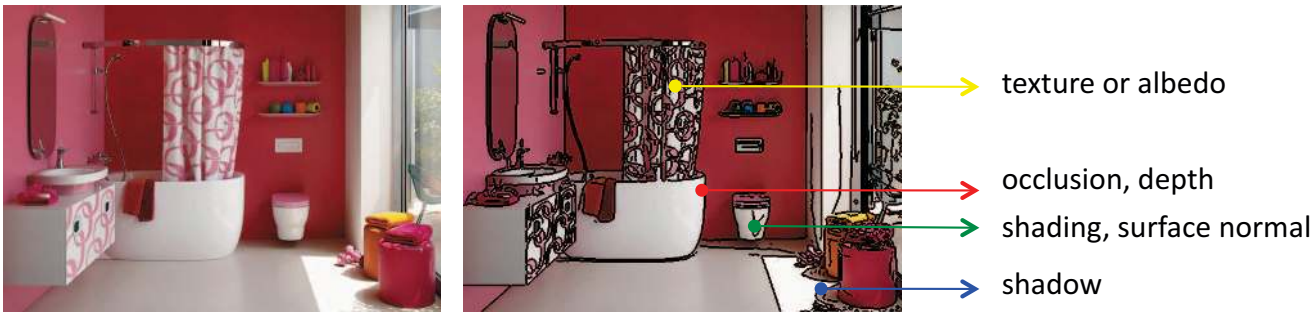


Figure 1. When an edge detector is applied to a color image the resulting edges are a superset of many different edge types. They typically include edges related to occlusion boundaries (red), shape properties such as shading (green), illumination e.g. cast shadows (blue) and the texture of the various surfaces in the scene (yellow).

ject recognition for humans. Further, we investigate which physical phenomena correspond to the edges detected by standard edge detectors in computer vision. Clearly, to answer these questions, we need images for which the ground-truth phenomenon leading to each edge pixel can be determined. Labeling these in natural images would be prohibitively time-consuming. Instead, we leverage the advances made in computer graphics. We synthetically render photorealistic images of different scene models [21, 25, 33]. Since the structure of the scene is known, the physical phenomena that produce each edge in the rendered image can be easily determined. The use of computer graphics also allows us to control the illumination conditions, reflectance properties and albedo variations of objects in the scene. This allows us to control which phenomenon and hence which edge types manifest themselves in the images. For instance, an object's albedo can be forced to be uniform, creating no albedo (i.e. no texture) edges. Similarly, gradients associated with changes in surface normals can be removed by applying uniform ambient lighting.

This paper presents a first investigation of the human visual system's dependance on edges using computer manipulations of scenes. In our study, we find that texture edges confuse object recognition, while edges resulting from surface normal variations and depth discontinuities are most informative. In our analysis of standard edge detection algorithms, we quantify the percentage of informative edges that each captures. We show that edge detectors find edges resulting from numerous physical phenomena, but the majority of detected edges result from object textures, which are also the least informative. However, we also find indications of solid progress, where recent improvements to edge detection are leading to detectors that capture less of the uninformative edges than older edge detectors.

Rather than relying on qualitative intuitions prevalent in the community, this paper explicitly quantifies the usefulness of different edge types for object recognition, using the human visual system as an existence proof, and characterizes different edge detection algorithms. In effect, this paper helps explain the success of recent edge-based methods and the surge in incorporating depth features for object recognition. It also provides inspiration for the design of next-generation features as additional sensing modalities (e.g. direct depth capture along with color) become more widespread. It may be beneficial to explicitly model occlusion, shadows, etc. or train edge detectors to detect more of the informative edges. Developing algorithms that automatically classify an edge into one of its various types to then be selectively fed into a descriptor for recognizing objects could also be beneficial.

2. Related work

This work is related to research in several broad subareas of cognitive science and computer vision, including edge and line analysis, human perception from edges, comparison of different descriptors for computer vision tasks, depth imaging and the use of synthetic photorealistic scenes for advancing computer vision. Humans, of course, do not acquire depth data directly for visual processing, but understanding size, shape, and distance is clearly an important part of the visual system as proposed by Marr [26]. The perception of depth and shape are tasks central to the human visual system. Human depth perception incorporates evidence from stereo cues, motion cues, and monocular cues including: texture gradients, lines and edges, occlusion, shading, and defocus [16]. In this work, we are particularly interested in the relationship between edges and object recognition in human perception.

Edge and line analysis: For decades, edge extraction has been a component of many computer vision and image processing systems. In many early works, [3, 4, 10, 41], image edges were extracted, and scene reconstruction was attempted based on these edges. However, in practice, these methods faced a fundamental problem: image edges are not necessarily related to structural edges in the scene. Instead, some image edges are related to image noise, or illumination and albedo changes in the scene. Our work attempts to explicitly quantify the usefulness of these different types of

edges for object recognition.

Human perception from edges and lines: It is well known that humans can perceive depth from an image of edges (i.e. from a line drawing). Studies have explored the ability of humans to perceive shape from various types of images [5, 7, 8, 20, 22]. In [8], subjects demonstrated effective surface normal estimation when shown artist renditions of various objects. Lines drawn by a skilled artist can convey shapes nearly as well as shaded objects. However, the mechanisms used by the artists to produce the line drawings are not computational models. Hence, unlike our study, they do not give us insights into which types of edges matter. In [40], it is shown that humans exhibit roughly similar fMRI patterns when viewing either line or color images, suggesting that the interpretation of both modalities is similar (and edge-based). However, line drawings do not capture all available scene information: [39] shows that subjects have systematic misinterpretations of shape in line drawings. Algorithmic approaches to render line drawings of 3D meshes that effectively convey the visual information to viewers have been proposed [12, 19].

Comparison of image features: Several works have compared performance of various image features, be it for low-level tasks such as matching image patches [21, 27, 34] or for high-level tasks such as image classification [42]. The goal of this paper is not to compare existing computational image descriptors. Instead, it is to evaluate the impact of the manifestations of different physical phenomenon in image content on object recognition performance. Moreover, while we characterize the properties of various edge detection algorithms in computer vision, we study recognition in humans and not machines.

Depth imaging: The past several years have seen an explosion of practical systems for capturing depth either directly (e.g. the PointGrey BumbleBee camera and Microsoft Kinect), or reasoning about depth in the scene from many images [37], or inferring depth from single images by extracting mid- and low-level features [18, 32]. For computer vision applications, depth-based features have shown remarkable efficacy for accuracy. For example, detection of humans and pose is accomplished with depth-based difference features *and no visual features* in [36]. Incorporating depth features in addition to RGB features was shown to boost patch-matching performance [21]. With the advent of many new sensing modalities, it is useful to analyze what information we should be attempting to extract from these. For instance, do we need the absolute depth values, or simply the depth discontinuities? Or perhaps the discontinuities in surface normals? Our study attempts to address these questions. By studying human object recognition, our findings are independent of specific algorithmic choices. [1]

[1] It would be more accurate to say that we choose humans as the "algorithm" to study since it is the best known vision system to date.

Use of synthetic scenes: Synthetic images and video data have been used to advance computer vision in a variety of ways including evaluating the performance of tracking and surveillance algorithms [38], training classifiers for pedestrian detection [25] or human pose estimation [36], learning where to grasp objects [31], evaluating image features for matching patches [21], etc. In this work we use images of synthetic scenes to understand the informativeness of different types of edges for human object recognition.

3. Creating Virtual Worlds

In this work, we consider the following four edge types: **Shadow edges:** observed at the boundaries of cast shadows, e.g. [23]. **Texture edges:** corresponding to a surface texture or a change in the reflectance properties (e.g. albedo) across the smooth surface of one or more materials (e.g. a stripe on a zebra). **Occlusion boundaries:** often observed at the boundary of an object, which likely occludes another object having different illumination, surface properties, or surface normal. **Surface normal discontinuities:** indicating intersecting surfaces or creases where surfaces have distinct surface normals, and therefore are illuminated differently. An edge observed in an image can be a result of a number of different arrangement of light and materials in a scene (Figure 1). We create synthetic scenes and manipulate the renderings so that specific types of edges can be selectively rendered in the scene.

We model and render our virtual scenes using SketchUp 8 [2] and Trimble 3D Warehouse powered by Google [3]. 3D Warehouse is a vast repository of user-created models of objects, scenes, and buildings, some of which are incredibly detailed. SketchUp allows for the creation of rendering styles that control the appearance of faces in the 3D model and edges, and has a simple user interface for placing objects into scenes. Further, controls are available for illumination sources (ambient and direct illumination). Shadows can be separately toggled on or off. Although more sophisticated rendering packages could be used, SketchUp has user interface advantages, and we find that our subjects could not reliably distinguish the rendered images from actual photographs.

3.1. Objects and Scenes

We assembled a 3D scene dataset containing 46 unique scenes by searching Google 3D Warehouse for models from the following 7 scene categories: bathroom (6), kitchen (10), living room (8), office (5), house (10), patio (3), and playground (4). Our object dataset contains 54 different objects including sandal, teddy bear, frying pan, tricycle, chair, faucet, grill, tree, laptop, car, person, books,

[2] www.sketchup.com

[3] sketchup.google.com/3dwarehouse

etc. In each scene, 4 to 12 objects are placed manually by the authors in a contextually appropriate position (e.g. a corkscrew is placed on a countertop, not on the floor). The same object may be placed in multiple scenes (e.g. a teddy may be on a living room sofa and also on a kitchen floor); however, with a different viewpoint.

3.2. Rendering

We render each of the 46 synthetic scenes in 6 different ways. Edges in these images correspond to different combinations of the edge types. In addition, a total of 5 edge images are produced, for a total of 11 images per scene. Table 1 describes the renderings, the edge images, and the edges types that are contained in each.

For each scene, all renderings are from the same camera viewpoint. The edge renderings (7-11) are made by applying a basic edge detector to the corresponding rendered image (more details in Section 3.3). A few points about specific steps for producing other renderings mentioned in Table 1 using Google SketchUp: 1) **RGB**: RGB is obtained by rendering with textures, shading, and shadows *on*. This rendering corresponds most closely to what we experience in everyday life. 2) **BW**: The grayscale rendering is produced simply by converting the RGB rendering to gray. 3) **Albedo**: To render only the albedo of the scene the rendering is performed with textures *on*, but uniform lighting. 4) **GraySurf**: All Lambertian surfaces are replaced with uniform-gray material having constant albedo. Shading is still visible, but shadows are turned *off*. 5) **GrayShad**: This rendering is identical to GraySurf, but the shadows (on faces in the 3D model and on the ground) are *on*. 6) **Depth**: There is no direct way in SketchUp to export an image of the depth map. We use the following trick: after turning off all textures and shading, the scene is rendered with linear black fog. The rendered image code value is then linearly related to the distance from the camera.

3.3. Extracting edges

We compute the edges in five of the six [4] SketchUp renderings described above to produce renderings 7-11 as in Table 1. A typical approach to visualizing edges in images is to obtain the gradient profile and perform global normalization, However, that approach retains only gradient edges that are globally dominant while globally weak edges are eliminated. To obtain a rendering where the user can observe all the locally dominant gradients clearly, we use the idea of locally normalized gradients by Zitnick [44] where the gradient profile is normalized with respect to the average gradient magnitude in a small local neighborhood. Examples of all 11 renderings of a scene can be seen in Figure 2.

[4]Most edge detectors convert an RGB image to grayscale before extracting edges, hence edges from BW and RGB are assumed to be identical.

Similar renderings of all 46 scenes, along with a list of the 54 objects in our dataset can be found in the supplementary material.

4. Experiments and Results

4.1. Calibration

We first wish to assess how realistic our synthetic scenes are. For each of the 7 scene categories in our dataset, we collected 15 real and 15 synthetic images[5]. We analyze the distribution of gradients for the synthetic images and compare them to the profile of real images. Figure 5(a) shows the result. We see that the gradient profiles are very similar. We also conducted a study where 10 subjects were shown each (grayscale) scene for 107ms (as in [13]) and asked to classify it as being real or synthetic. Their average accuracy was 56%, not much higher than chance (50%). We also had subjects recognize the scene category of the image. Average accuracy was 68%, which is significantly above chance ($\sim$15%). This suggests that 107ms is sufficient for subjects to understand the context of the scene and yet they can not reliably tell if the scene is synthetic or real. These tests indicate that our synthetic images are realistic looking.

4.2. Human Object Recognition Studies

The goal of our human studies is to determine what types of edges are most informative to humans for recognizing objects. Our experimental setup is shown in Figure 3. We focus the attention of our subjects by flashing a 3 second countdown followed by a prompt to focus on a following red dot. The red dot is located on a white field, and indicates the location of an object to be recognized in the following scene. The scenes, as described earlier, are rendered according to one of our 11 renderings. After viewing the scene, each subject is asked to identify the indicated object using free-form text. They were instructed to use one word or a short phrase for the object name. The scene is flashed for either 107ms or 500ms following the approach of Fei-Fei et al. in [13]. The shorter time frame does not allow for the subject to change fixation point, while the longer time frame allows for one or two quick saccades before the scene disappears.

We conducted these tests for each combination of object, scene and rendering style as described in Section 3. We manually marked the location of the objects in the scenes to generate the red dots, which resulted in 265 unique object instances for human subjects to recognize in each of the 11 renderings. Each of these 2915 tests was performed by 10 different subjects. All experiments were performed on Amazon Mechanical Turk [6], a crowd-sourcing service that

[5]A subset of these were the 46 images in our dataset.

[6]MTurk has been used by several works analyzing abilities of human subjects at visual tasks e.g. [45].

Index	Label	Description	Edge Types			
			Shadow	Text	Occ	Norm
1	RGB	Full color image	✓	✓	✓	✓
2	BW	Grayscale image	✓	✓	✓	✓
3	Albedo	No shading or shadows		✓		
4	GraySurf	Lambertian gray surfaces			✓	✓
5	GrayShad	Same as 4, with shadows	✓		✓	✓
6	Depth	Linear depth			✓	
7	RGB_E	Edges of 1	✓	✓	✓	✓
8	Albedo_E	Edges of 3		✓		
9	GraySurf_E	Edges of 4			✓	✓
10	GrayShad_E	Edges of 5	✓		✓	✓
11	Depth_E	Edges of 6			✓	

Table 1. A summary of the 11 renderings that are produced for each scene.

(a) RGB

(b) BW (c) Albedo (d) GraySurf (e) GrayShad (f) Depth

(g) RGB_E (h) Albedo_E (i) GraySurf_E (j) GrayShad_E (k) Depth_E

Figure 2. Example renderings (Table 1) used in our work. (a)-(k) show the various renderings for a kitchen scene.

allows workers around the world to participate and get paid for online tasks. Each subject was paid 7 cents for 10 such sequences. To avoid biases introduced by workers "learning" the scenes or individual objects, workers were allowed to perform at most 10 recognition sequences.

To evaluate the accuracy of the responses from the subjects, we implemented a worker grading task also on Amazon Mechanical Turk. We displayed each scene in full color. We also showed graders the same scene with the red dot overlaid on the image at the precise location where it was displayed to the subjects of the previous task. We showed graders a list of object names provided by subjects, and asked them to mark the names that correctly refer to the object beneath the red dot. Each name was graded by 10 different graders. We consider a name to be correct if 5 or more graders mark it as correct. An illustration of the grading interface is shown in Figure 4.

In Figure 5(b) we show results of human subjects on the first six (1-6) renderings. Recognition performance is higher for renderings containing more information such as RGB, whereas rendering with less information such as GraySurf perform worse. This may be due to multiple reasons. First, humans may be good at filtering out irrelevant information, allowing them to make use of more information effectively. Second, renderings such as RGB and BW appear more natural, and thus are easier to recognize. The Depth rendering has less contrast and the images are very unnatural to humans. The trends from 107ms to 500ms are

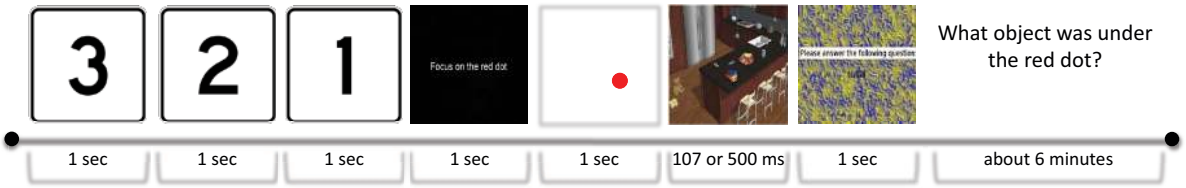


Figure 3. The timing sequence used in our human object recognition studies.

Figure 4. The grading interface used to evaluate human object recognition performance.

also interesting. As more time is given, renderings with more information such as RGB, BW, and GrayShad obtain a more significant performance boost. This may suggest a larger duration is needed for information discovery.

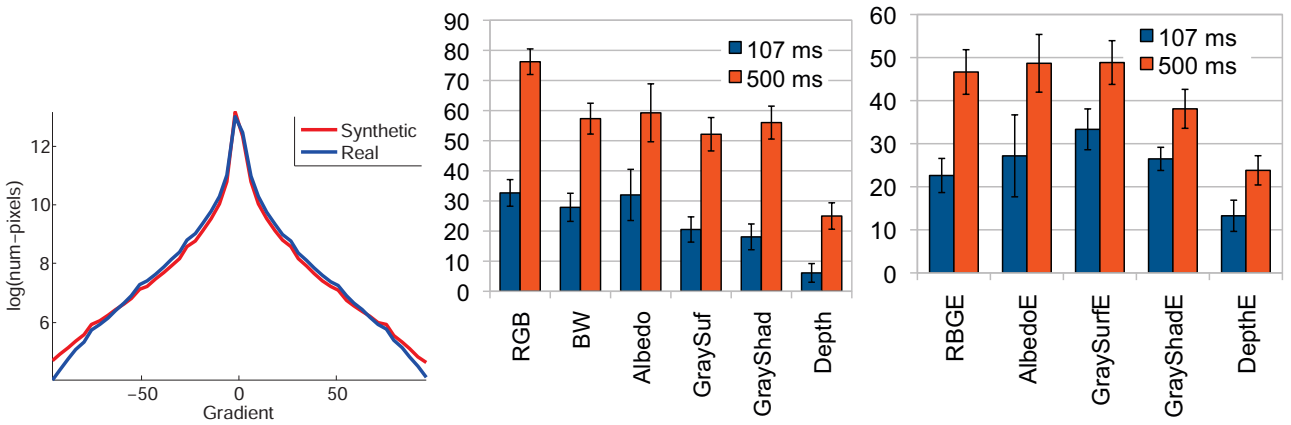
More interestingly, Figure 5(c), shows human performance on the *edge* images. At 107ms viewing time, the presence of more edges in RGB_E and GrayShad_E reduce recognition performance when compared against the reduced edges in GraySurf_E. Also, GraySurf_E performs better than the similarly rendered GrayShad_E that has the addition of shadows, indicating that shadows may prove distracting. We also find that information in depth edges alone (Depth_E) are not enough for object recognition and result in poor performance. The shape information provided by surface normals (visible in GraySurf_E) is critical. As the viewing time increases to 500 ms, the performance gap (between GraySurf_E, Albedo_E, and RGB_E) disappears, perhaps as higher-level processes kick in.

4.3. Edge Analysis

We now study the types of edges that are found using a variety of machine edge detection and segmentation algorithms: thresholding the gradient of a pixel (grad), canny edge detection [6], normalized cuts [35] (ncut), mean shift [9] (ms), Felzenszwalb and Huttenlocher [15] (fh), ultrametric contour maps [1] (ucm) and Zitnick's approach [44] (ngrad) presented in Section 3.3. The implementation details of the first six approaches can be found in [45]. We used the same parameter settings as in [45] which were chosen to match human edge detection performance. We detected edges using each of these approaches in the RGB image. Since the geometry and lighting of each scene is known, each detected edge can be matched to the physical phenomena associated with the edge.

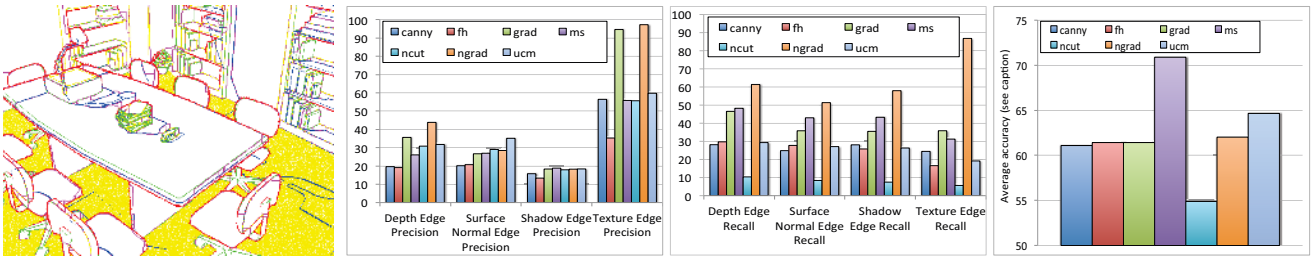
We produce a map for each type (texture, occlusion, shadow, surface normal) of edge by analyzing the original six renderings of each scene. For instance, edge detection on the depth maps corresponds to occlusion edges. Edges that appear in GrayShad_E but not in GraySurf_E correspond to shadow edges. Edges that appear in Albedo_E are texture edges. Edges in GraySurf_E that are not occlusion edges are surface normal edges. An illustration of our edge classification for an example scene is shown in Figure 6(a). For the visualization, if an edge pixel is associated to multiple edge types, we assign it to the type that is more rare. The list of precedence is Depth, GraySurf, GrayShad and Albedo. An analysis of the types of edges found by the different edge detection algorithms is in Figure 6(b) and 6(c), where each detected edge pixel can be assigned to multiple source edge types if applicable. We obtain the precision as the proportion of the edges retrieved by each edge detector that are relevant edges (of the specific type) (Figure 6(b)). UCM has relatively high precision for detecting the informative surface normal and depth edges, while a larger proportion of the edges found by other detectors (e.g. ngrad) correspond to the less informative texture edges. Figure 6(c) shows the recall i.e. proportion of the relevant edges (of the specific type) retrieved by each edge detector. We consolidate these statistics in Figure 6(d). We treat the occlusion and surface normal edges as the positive class, and the remaining pixels in the image as the negative class. We compute the proportion of the positive pixels detected by an edge detector and the proportion of negative pixels left undetected. The average of these two accuracies is shown in Figure 6(d). We see that the mean-shift segmentation algorithm performs surprisingly well. This is followed by UCM, a recent state-of-art edge detector. We see that it outperforms several classical approaches even on this new evaluation.

To design improved edge detectors, we suggest that it may be useful to perform this type of detailed evaluation of



(a) Gradient histograms (b) Accuracy of Humans with Images (c) Accuracy of Humans with Edges

Figure 5. (a) We verified that real and our computer-rendered images have similar low-level statistics, and were difficult for subjects to distinguish. Consequently, we are confident that our human studies provide insight into visual processes on natural imagery. (b) and (c) show object recognition performance by humans when viewing (b) various renderings of the scene and (c) edges extracted from these renderings. When viewing edge images (c), humans achieve the best performance when there is an absence of texture and shadow edges (as in GraySurfE at 107ms), and the visible edges correspond to occlusion and surface normal edges.



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

Figure 6. (a) Edges found in the RGB image using the state-of-the-art UCM [1] contour detector classified by edge type. Yellow: texture edges, Blue: shadow edges, Green: surface normal edges, Red: occlusion edges, and Magenta: none of the above. The majority of detected edges are texture, which lacks discriminative information for object recognition. The precision (b) and recall (c) of each edge detector for each of the four edge types is also shown. UCM has relatively high precision for detecting the informative surface normal and depth edges, while other detectors (e.g. ngrad) detect a larger proportion of the less informative texture edges. We summarize the performance of each detector in identifying the informative (occlusion + surface normal) edges in (d). Best viewed in color.

edge detectors. [7] Useful edge detectors are those that, on a large variety of (possibly computer generated) images, find informative edges associated with surface normals and occlusions, but ignore shadow and texture edges. Notice that this methodology removes the need for manual annotation of images to evaluate edge detectors.

5. Conclusion

We study the usefulness of different edge types for object recognition in humans. We approach this problem in a novel way by constructing scenes in a virtual world, controlling surface properties and illumination when rendering images of the scenes, and then applying edge detectors on the resulting images. This allows us to identify the source of each edge pixel in an image. A battery of human studies were conducted to show that not all edges are equally informative. Edges related to surface normal changes and occlusion were the most informative, and edges associated with shadows and textures make recognition tasks more difficult. We show that edge detectors find edges resulting from numerous physical phenomena, but a larger portion of the edges detected correspond to shadows and albedo (uninformative) than to surface normal and depth changes (informative). We believe that this paper takes steps toward explaining the success of edge-based methods, as well provides inspiration for the design and evaluation of next-generation features as additional sensing modalities become more widespread. Automatically identifying the sources of the different edges, the first steps towards which are being taken in works like [2], may be beneficial. This is a preliminary exploration. There are several avenues for future

[7] This is in similar philosophy to Hoiem *et al*. [17] that suggests more details evaluation of object detectors.

work: more realistic renderings of synthetic scenes, evaluating machine vision algorithms for different edge types, analyzing outdoor scenes, etc.

Acknowledgements: This work was supported in part by NSF IIS-1115719/1341772.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 2011.

[2] J. T. Barron and J. Malik. Shape, albedo, and illumination from a single image of an unknown object. In *CVPR*, 2012.

[3] H. G. Barrow and J. M. Tenenbaum. Recovering intrinsic scene characteristics from images. In *Computer Vision Systems*, 1978.

[4] H. G. Barrow and J. M. Tenenbaum. Interpreting line drawings as three-dimensional surfaces. *AI*, 1981.

[5] K. Burns. Mental models of line drawings. *Perception*, 2001.

[6] J. Canny. A computational approach to edge detection. *PAMI*, 1986.

[7] F. Cole, F. Durand, W. Freeman, and E. Adelson. Interpreting line drawings of smooth shapes. *Journal of Vision*, 2011.

[8] F. Cole, K. Sanik, D. DeCarlo, A. Finkelstein, T. Funkhouser, S. Rusinkiewicz, and M. Singh. How well do line drawings depict shape? *ACM Trans. Graph.*, 2009.

[9] D. Comanicu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 2002.

[10] M. C. Cooper. Interpretation of line-drawings of complex objects. *Image and Vision Computing*, 1993.

[11] J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A*, 1985.

[12] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella. Suggestive contours for conveying shape. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 22(3):848–855, July 2003.

[13] L. Fei-Fei, A. Iyer, C. Koch, and P. Perona. What do we perceive in a glance of a real-world scene? *Journal of Vision*, 2007.

[14] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 2009.

[15] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004.

[16] J. J. Gibson. Perception of the visual world. In *Houghton Mifflin*, 1950.

[17] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *ECCV*, 2012.

[18] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. In *SIGGRAPH*, 2005.

[19] T. Judd, F. Durand, and E. H. Adelson. Apparent ridges for line drawing. *ACM Trans. Graph.*, 26(3):19, 2007.

[20] N. Kawabata. Depth perception in simple line drawings. *Perceptual Motor Skills*, 1997.

[21] B. Keneva, A. Torralba, and W. Freeman. Evaluation of image features using a photorealistic virtual world. In *ICCV*, 2011.

[22] J. Koenderink, A. van Doorn, A. Kappers, and J. Todd. Ambiguity and the 'mental eye' in pictorial relief. *Perception*, 2001.

[23] J. F. Lalonde, A. A. Efros, and S. G. Narasimhan. Detecting ground shadows in outdoor consumer photographs. In *ECCV*, 2010.

[24] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[25] J. Marin, D. Vazquez, D. Geronimo, and A. Lopez. Learning appearance in virtual scenarios for pedestrian detection. In *CVPR*, 2007.

[26] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Freeman, 1982.

[27] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 2005.

[28] D. N and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[29] T. Randen and J. Husoey. Filtering for texture classification: A comparative study. *PAMI*, 1999.

[30] L. Roberts. Machine perception of 3-D solids. Ph.D. Thesis, 1965.

[31] A. Saxena, J. Driemeyer, and A. Y. Ng. Robotic grasping of novel objects using vision. *IJRR*, 2008.

[32] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *PAMI*, 2009.

[33] J. Schels, J. Liebelt, K. Schertler, and R. Lienhart. Building a semantic part-based object class detector from synthetic 3d models. In *ICME*, 2011.

[34] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *IJCV*, 2000.

[35] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 2000.

[36] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, 2011.

[37] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH*, 2006.

[38] G. R. Taylor, A. J. Chosak, and P. C. Brewer. Ovvv: Using virtual worlds to design and evaluate surveillance systems. In *CVPR*, 2007.

[39] J. T. Todd. The visual perception of 3d shape. In *Trends in Cognitive Science*, 2004.

[40] D. B. Walther, B. Chai, E. Caddigan, D. M. Beck, and L. Fei-Fei. Simple line drawings suffice for functional mri decoding of natural scene categories. In *PNAS*, 2011.

[41] D. Waltz. Generating semantic descriptions from drawings of scenes with shadows. Technical report, MIT, 1972.

[42] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[43] L. Zhu, Y. Chen, and A. L. Yuille. Learning a hierarchical deformable template for rapid deformable object parsing. *PAMI*, 2010.

[44] C. L. Zitnick. Binary coherent edge descriptors. In *ECCV*, 2010.

[45] C. L. Zitnick and D. Parikh. The role of image understanding in contour detection. In *CVPR*, 2012.