Depth Image Region Segmentation

Jackson Spell and **Micah Brown**
{jaspell,msbrown}@davidson.edu
Davidson College
Davidson, NC 28035
U.S.A.

Abstract

Segmentation algorithms separate an image into unique regions determined by characteristics of the pixels within those regions. Image segmentation has applications in Artificial Intelligence, including computer vision, 3D modeling, and robotics. We use two different algorithms to segment a depth image, which encodes distance information into rather than conventional RGB values. Whereas color images are commonly segmented into regions of similar color or luminosity, we segment the depth image into cohesive surfaces using gradient difference and Laplacian edge detection. When combined together, these two methods offer a chance to reconstruct three dimensional objects from depth data.

1 Introduction

Image processing is a vital component of computer vision and artificial intelligence. Within image processing, image segmentation continues to be a focus of research as machines are developed to have similar visual abilities to humans. Segmentation groups pixels into regions of similar characteristics, which can be used for automated analysis, tracking, and object recognition. Many image segmentation techniques operate on color images, grouping pixels by color or perceived luminosity. We explore an alternative: segmentation of depth images.

Depth images contain information about the distance from a point in space to the camera sensor. While traditional RGB images contain a matrix of color pixels, each containing a red, green, and blue value, a depth image only contains a matrix of depth values. Depth images can be visually rendered in greyscale, where a point of maximum distance is represented as white and a point of minimum distance is represented as black. The construction of accurate depth images relies heavily on recent technology, including LIDAR and infrared sensors. Notably, the Microsoft Xbox Kinect sensor has the ability to simultaneously record depth images and color images, providing researchers easy access to rich depth data.

The algorithms presented below were tested and tuned using data taken by an Xbox Kinect sensor by a team from UC Berkeley and the Max Planck Institute for Informatics in their work on category-level object recognition (Janoch et al. 2011). As such, it presents an imperfect source of realistic, often cluttered scenes, perfect for useful algorithm development.

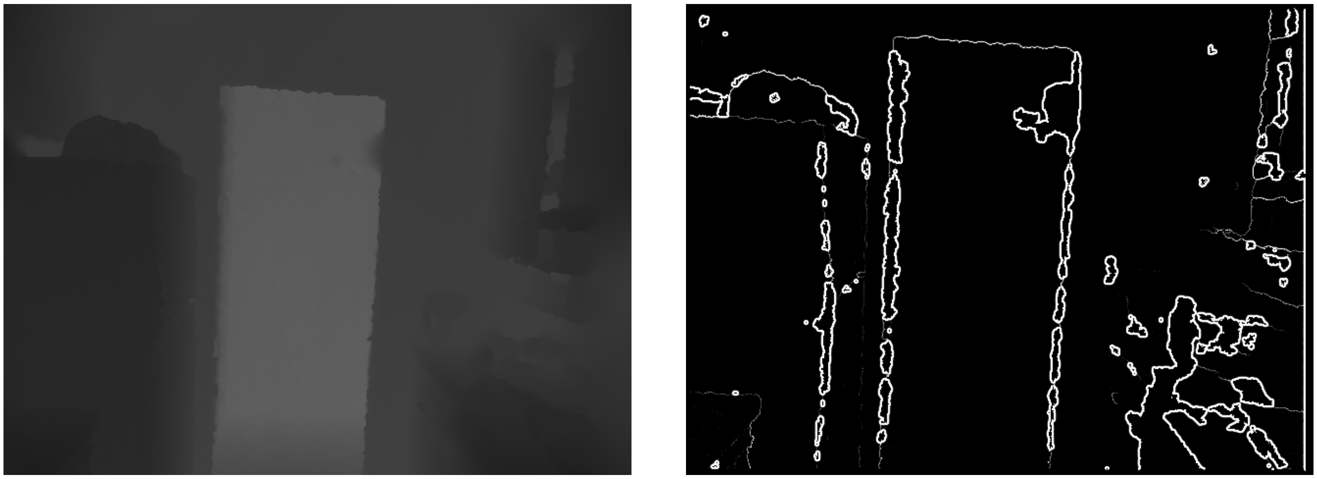
There are two fundamental approaches for segmenting an image: edge-based segmentation and region-based segmentation (Russell and Norvig 2003). Edge-based segmentation locates major discontinuities in the image, indicating separate distinguishable objects, whereas region-based segmentation identifies different surfaces within those objects by calculating the normal vectors at each point. The remainder of this paper covers background information pertaining to the two methods of segmentation used, describes our specific experimental design, and shares our results and conclusions.

2 Background

Laplacian Edge Detection (LED)

Edge detection distinguishes edges between pixels by locating large discontinuities in the data of adjacent pixels. In color images, two different objects often have very different luminosities, colors, or patterns; edge detection locates the areas where the pixels experience a rapid change in those values. While pixels in RGB images have various characteristics, depth image pixels only have one piece of data (the depth). Thus, edge detection in depth images locates objects in an image by finding the areas in which the depth data rapidly changes, known as jump discontinuities.

The Marr-Hildreth algorithm is one of the simplest forms of edge detection (Marr and Hildreth 1980). A Gaussian blur convolution is applied to the depth image. The blurred image is subtracted from the original image, resulting in an "edged" image, such as in Fig. 1b. Areas with similar values across pixels are relatively unchanged after the blur. The difference between the original image and the blurred image is therefore small over that area, resulting in small values at that location in the edged image. In contrast, areas with large differences in values appear significantly different in the blurred image, and as such have



(a) Depth (in greyscale)　　(b) Edged

Figure 1: Edge detection of a doorway and nearby fridge and countertop using the Marr-Hildreth algorithm.

large values at that location in the edged image. When rendered in color, the edged image is dark in continuous areas of the depth image but bright in areas with significant discontinuities.

It should be noted that the Marr-Hildreth algorithm defines edges, but does not group the pixels within those edges. Grouping of pixels is left to a simple flood-fill algorithm.

Gradient Surface Detection (GSD)

While the Marr-Hildreth algorithm behaves similarly on both color and depth images, it does not take advantage of the different data provided by a depth image. The distance values in a depth image give a pixelated 3D view from a single vantage point, providing volumetric information about the scene. This information is especially valuable when attempting to segment surfaces.

Pulli and Pietikäinen explore several sophisticated approaches in their discussion of depth image segmentation, including fitting continuous differentiable functions, robust estimators, and least squares approximations to the surfaces (Pulli and Pietikäinen 1993). We take a simpler and faster approach, evaluating normal vectors for each pixel, which acts as a local partial derivative, and grouping pixels into surfaces if the normals are sufficiently continuous.

3 Algorithms

Our research sought to determine whether LED or GSD would produce better segmentation on depth images. Our algorithms are outlined below in Alg. 1 and Alg. 2. In LED, we found that a convolution matrix with a radius of 4 pixels and $\sigma = 1$ resulted in a smooth blur for the majority of our dataset. For Fig. 2b, we found a difference threshold of 4 cm was an accurate measure of edges. Using a low threshold resulted in well defined edges but presented several false artifacts as an unintended side-effect. Once the edges have been identified, a simple flood-fill algorithm fills in objects that are completely defined by edges.

Algorithm 1 Laplacian Edge Detection (Image *depth*)

Image *segmented* ← **new** Image
Image *blur* ← **new** Image
blur ← GaussianBlur(*depth*)
for Pixel p **in** *depth* **do**
    $segmented_p \leftarrow (original_p - blur_p)$
    if $segmented_p > threshold$ **then**
        label $segmented_p$ **as** Edge
    end if
end for

for Pixel p **in** *segmented* **do**
    if p **is not** Visited **and** p **is not** Edge **then**
        List *region* ← FloodFill(p)
        for Pixel q **in** *region* **do**
            label q **as** Visited
        end for
    end if
end for
return *segmented*

Gradient Surface Detection relies on accurate calculations of the surface normals. First, the gradient is calculated in the horizontal direction by taking the difference of the two depths on either side of the pixel. This gradient, $\Delta horizontal$, represents the vector of the surface in the horizontal direction. The value of the angle of this vector is found by:

$$\theta_1 = \tan^{-1}\left(\frac{\Delta horizontal}{2}\right) \qquad (1)$$

Similarly, the angle of the vertical gradient, $\Delta vertical$, is found by:

$$\theta_2 = \tan^{-1}\left(\frac{\Delta vertical}{2}\right) \quad (2)$$

Now, using Eqs. (1) and (2), we calculate the normal vector to both of those vectors as:

$$\psi = \cos^{-1}\left(\cos\theta_1 \cos\theta_2 + \sin\theta_1 \sin\theta_2 \cos(\theta_1 - \theta_2)\right) \quad (3)$$

ψ is derived from the spherical law of cosines. For Fig. 2b, a threshold of 35 degrees was used to determine whether one surface normal differed significantly from another. A "maximum jump" parameter was added to solve the case of two similar normals adjacent (due to perspective) in a depth image, despite being physically far away from each other. For example, a depth image of an open doorway, such as in Fig. 1a, may have adjacent pixels on the edge of the doorway; a pixel on the near wall could have a similar normal vector as one through the doorway on the far wall. These are two distinct surfaces that GSD would ideally separate, yet their normals are identical. By setting $max\_jump = 10$, pixels differing by more than 10 centimeters cannot be part of the same surface.

Algorithm 2 Gradient Surface Detection (Image *depth*)

```
Image segmented ← new Image
for Pixel p in depth do
    p_norm ← NORMAL(p)
end for
List regions ← new List
for Pixel p in depth do
    if p not visited then
        label p as Visited
        List region ← REGIONIFY(p, region)
    end if
    append regions with region
end for
for region in regions do
    for Pixel p in region do
        label segmented_p as region
    end for
end for
return segmented

procedure REGIONIFY(Pixel p, List region)
    for Pixel q in NEIGHBORS(p) do
        if q not Visited and
                ANGLE(p_norm, q_norm) < threshold and
                |p − q| < max_jump then
            append region with q
            label q as Visited
            REGIONIFY(q, region)
        end if
    end for
    return region
end procedure
```

4 Results

The fundamental difference in the detection techniques between the algorithms has a highly noticeable effect. Because LED is sensitive to jump discontinuities, edges of objects and surfaces go unnoticed if they coincide with the edge of another surface, such as on corners. This effect is visible in Fig. 2c, where the stove in the center of the image has been correctly segmented, but the front-facing corner of the stove is not reflected in the segmented image. The jump discontinuities at the back edges of the stove separate it from the wall and cabinets sufficiently to register as a distinct object.

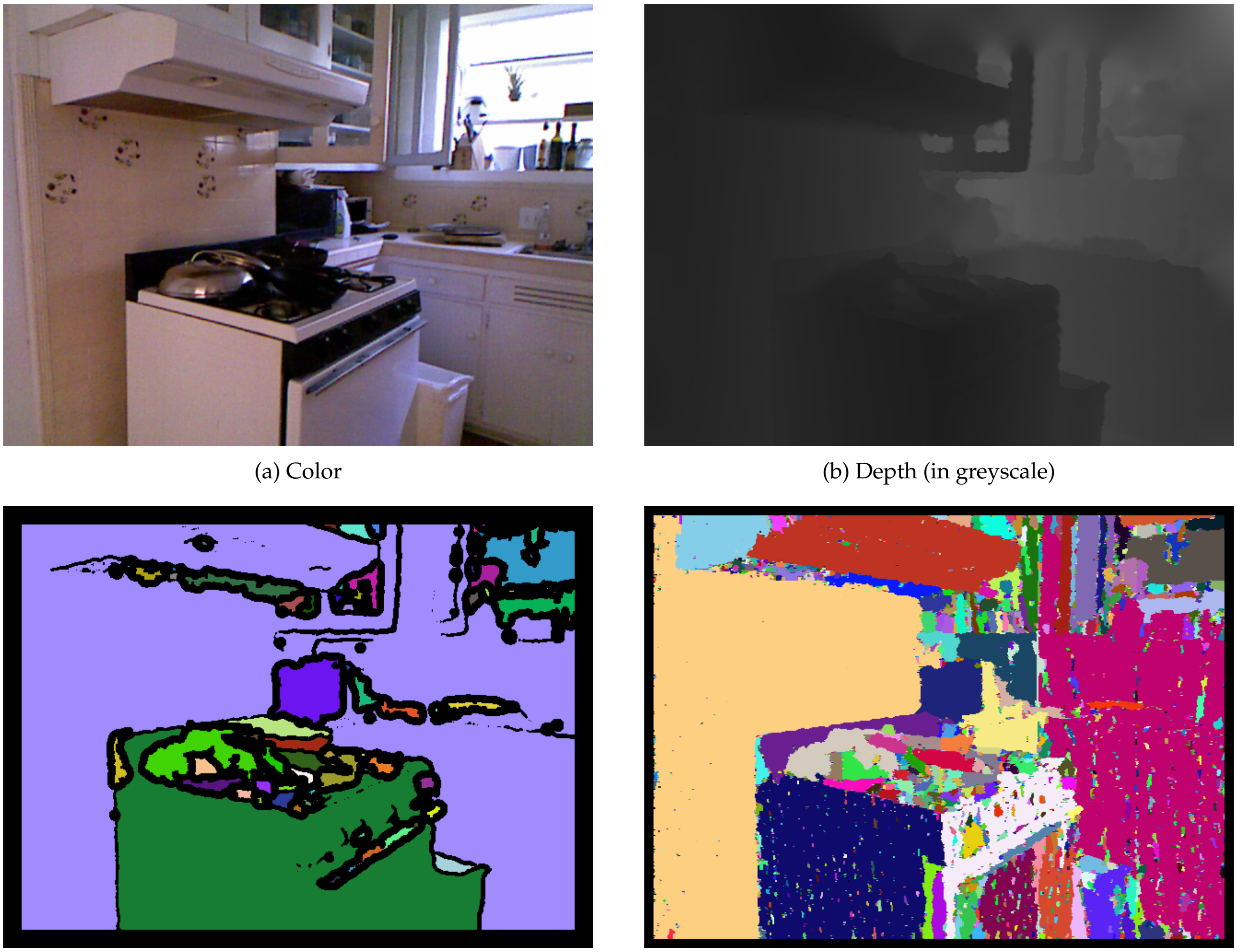
GSD, on the other hand, reliably detects corners and surface edges. The faces of the stove are clearly visible in Fig. 2d, and sufficient difference in normal vector exists between the wall and the edges of the stove to separate the two into separate regions, like in Fig. 2c.

It is important to note that reliable behavior for these algorithms was hard to find. Different images required tuning to see meaningful results, as a correctly chosen blur radius and edge threshold are crucial to detecting meaningful edges in LED and angle threshold defines GSD's ability to group regions. Segmenting an image containing venetian window blinds with LED, for example, requires a relatively large blur radius and threshold to avoid segmenting every individual slat in the blinds. Alternatively, if that behavior is desired, the radius and threshold can be lowered. However, these changes also alter the algorithm's ability to correctly distinguish other objects within the image, with large radii and thresholds blurring objects out of existence and small values incorrectly grouping small bumps or, as discussed below, flaws in the image.

In general, the GSD algorithm required a cleaning algorithm to partially handle the speckling problem visible in Fig. 2d. We believe this behavior occurs due to random noise in image quality – the result of using a realistic data set. LED is fairly robust to these fluctuations because of the averaging effect of the blur, but because GSD calculates the normal vector for every pixel, it is sensitive to small errors in the data. While a small Gaussian blur would have likely removed most of the flaws, the blur also averages data around corners, rendering the normal vectors less accurate and essentially decreasing the resolution of the depth image. Because GSD is sensitive to small differences in surfaces, it is less reliable on images with small details, such as the venetian window blinds mentioned above, which were frequently grouped into arbitrary, inconsistent regions.

5 Conclusions

We have developed two algorithms for depth image segmentation, one using the Marr-Hildreth algorithm (Laplacian Edge Detection) and one grouping pixels by normal vector (Gradient Surface Detection). Those algorithms were tested on images of normal domes-



(a) Color

(b) Depth (in greyscale)

(c) Laplacian Edge Detection

(d) Gradient Surface Detection

Figure 2: The segmentation of a kitchen scene featuring a stove, oven hood, and cabinets using the Laplacian and Gradient algorithms (Janoch et al. 2011).

tic scenes taken by an Xbox Kinect sensor, providing a realistic and varied environment. We found that LED segmented based on jump discontinuities, locating disconnected objects, and GSD located surfaces within each object.

Because LED and GSD segment an image based on different qualities, their results provide different information. LED finds disconnected objects, but provides no information about the shape of the object. GSD describes the structure of each surface within the image, but cannot group surfaces to approximate a total object. Further research in these algorithms could investigate a combination of the two, using LED to locate objects and GSD to separate them into surfaces, allowing an object to be recognized as a system of planes and facilitating automated modeling.

We frequently found that characteristics in parts of an image – small details, flaws in image quality, etc. – governed the choice of parameters for the entire image, which limited the performance of both algorithms across the image as a whole. An analysis function designed to tune parameters for sections of an image would allow the algorithms to perform effectively across the image, and would significantly increase the value of the algorithms presented.

6 Acknowledgements

Thanks to Dr. Raghu Ramanujan for guidance throughout this project and Dr. Tabitha Peck for algorithmic inspiration.

References

Janoch, A.; Karayev, S.; Jia, Y.; Barron, J. T.; Fritz, M.; Saenko, K.; and Darrell, T. 2011. A category-level 3-d object dataset: Putting the kinect to work. In *ICCV*

Workshop on Consumer Depth Cameras for Computer Vision.

Marr, D., and Hildreth, E. 1980. Theory of edge detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences* 207(1167):pp. 187–217.

Pulli, K., and Pietikäinen, M. 1993. Range image segmentation based on decomposition of surface normals. In *Proceedings of the Scandinavian conference on image analysis*, volume 2, 893–893. PROCEEDINGS PUBLISHED BY VARIOUS PUBLISHERS.

Russell, S. J., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*. Pearson Education.