Gradient Surfing for Depth Prediction from DIGIT Tactile Images

Vikram Khandelwal
University of Maryland, College Park
vikramkh@umd.edu

Gavin Hung
University of Maryland, College Park
ghung@umd.edu

Yuliang Peng
University of Maryland, College Park
ypeng12@umd.edu

Abstract

In this paper, we propose a novel mathematical approach to estimating depth from tactile images captured by the DIGIT sensor. Leveraging a mapping from the RGB color space to 2-dimensional gradient space, generated during a training step with over 200 tactile images, we achieve high-accuracy depth maps. Our method involves image subtraction, calculation of depth gradients, and a K-D Tree mapping between colors and gradients. The reconstructed depth map, obtained by "surfing" the gradients, is further refined through a clipping process to eliminate artifacts. Experimental results demonstrate the effectiveness of our approach on both training and test data, showcasing its potential for accurate depth estimation in tactile imaging.

1. Introduction

We propose a mathematical approach to estimating depth from tactile images generated by the DIGIT sensor [2]. We generate a mapping from the RGB color space to the 2-dimensional gradient space in the "training" step, from over 200 tactile images. We then reconstruct the 3D geometry by surfing the gradients, resulting in a high-accuracy depth map. We present methods for cleaning and calibrating the generation of this depth map as well.

2. Machine Learning Approach

The problem of computing depth from an image in the color space has historically been solved through machine learning approaches. MiDaS, for example, implements a transformer architecture with a convolutional decoder to calculate high-resolution depth [4] [3]. Earlier approaches involve utilizing a multi-scale deep neural network, consisting of both global and local approximations using fully-connected and convolutional layers simultaneously [1].

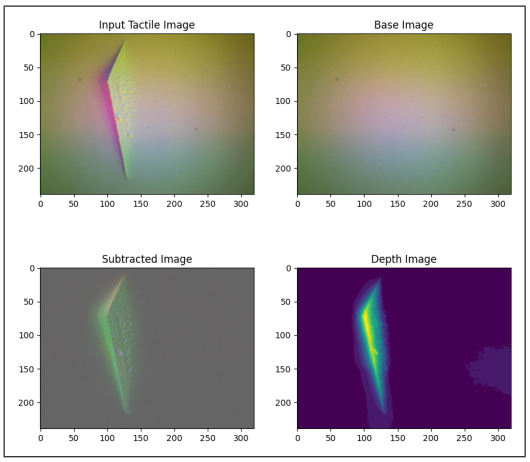

Figure 1. The difference between a sample tactile image and the base tactile image is compared with the ground-truth depth image.

3. Mathematical Approach

3.1. Image Subtraction

The key observation for this approach is that an injective map exists between the RGB color space of the tactile images, and the gradients of the depth map. Thus, the first step of this approach is to get the color at every position in the image. It was also observed that every input tactile image had the same background. In order to isolate the color at every position of the image, all of the input images were subtracted from a base image. The base image represents a tactile input with 0 depth. By comparing the subtracted tactile input image with the ground truth depth, we observed that colors are constant when the height is increasing. Figure 1 compares an input tactile image, the difference between a tactile image and the base image, and its corresponding ground-truth depth map.

3.2. Calculate Depth Gradients

After calculating the color at each coordinate of the image, the depth gradient at each coordinate will be calculated in the x- and y- direction using a 5 by 5 x-direction Sobel operator and a 5 by 5 y-direction Sobel operator respectively. The operators are shown in Figures 2 and 3. The gradient vectors in the x- and y-directions will then be extracted by looping through every coordinate of the two output images. Figure 4 is a visualization of the depth gradients through a HSV image.

3.3. Repeat for Training Data

In order to get a good mapping between the colors and the depth gradients, the depth gradients have to be calculated across all of the given training images. Given the folder structure of the training data, we created a script to move all of the training images to one folder.

$$\begin{bmatrix} -1 & -2 & 0 & 2 & 1 \\ -2 & -3 & 0 & 3 & 2 \\ -3 & -5 & 0 & 5 & 3 \\ -2 & -3 & 0 & 3 & 2 \\ -1 & -2 & 0 & 2 & 1 \end{bmatrix}$$

Figure 2. 5 x 5 Sobel Operator in the X-Direction

$$\begin{bmatrix} -1 & -2 & -3 & -2 & -1 \\ -2 & -3 & -5 & -3 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 3 & 5 & 3 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix}$$

Figure 3. 5 x 5 Sobel Operator in the Y-Direction

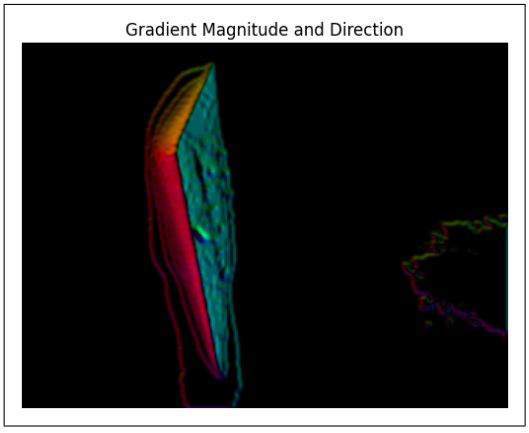

Figure 4. The gradients in the x- and y-direction can be visualized through a HSV image by making the hue the normalized direction and the value the normalized magnitude of the x- and y- gradient magnitudes.

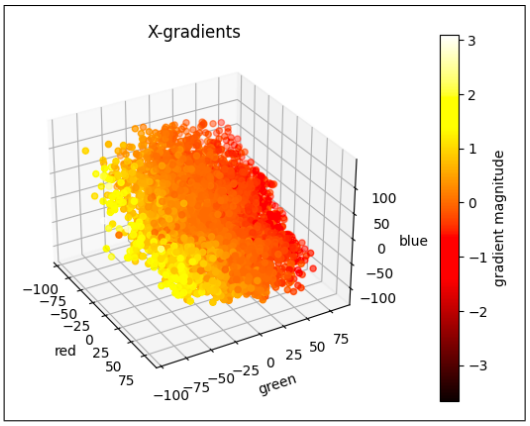

Figure 5. The depth gradients in the x-direction are clustered by the color of the pixel, showing that color corresponds to the depth derivatives.

3.4. Map Colors and Gradients with K-D Tree

With the gradients in the x- and y-direction now calculated for each training image, colors can be mapped to depth gradients by iterating through every coordinate of every image and mapping the color at the coordinate to the depth gradient. The K-D Tree data structure was chosen to create the mapping between colors and depth gradients. Recall that the red, green, and blue (RGB) color space was used. The mapping between colors and gradients can be visualized by mapping the red, green and blue values to the x, y, and z coordinates with a three dimensional scatter plot. The color of the points represent the depth gradient. There is a clear difference in depth gradients with different colors, as shown from the different clusters of depths in Figures 5 and 6.

3.5. Reconstruct Depth Map

The reconstruction step is the analogue to the inference step in the typical machine learning system. The K-D trees for the x- and y-gradients serve to translate the tactile image from the color space to the gradient space. After that, we reconstruct the depth map by ”surfing” the gradients (refer to Algorithm 1). In particular, either the x- or y-axis is selected for surfing, based on the direction in which the aggregate sum of gradients is largest. Once the direction is selected, the image is surfed from both the positive and negative directions. The resulting two depth maps are combined by taking the maximum value at each pixel location, while escaping to zero if either of the depth maps yields zero at that location. Figure 7 shows reconstruction step output.

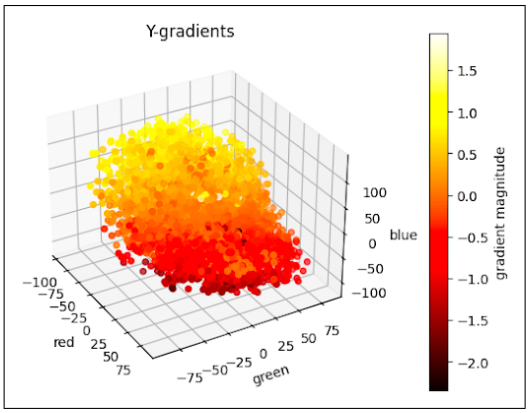

Figure 6. The depth gradients in the y-direction are clustered by the color of the pixel, showing that color corresponds to the depth derivatives.

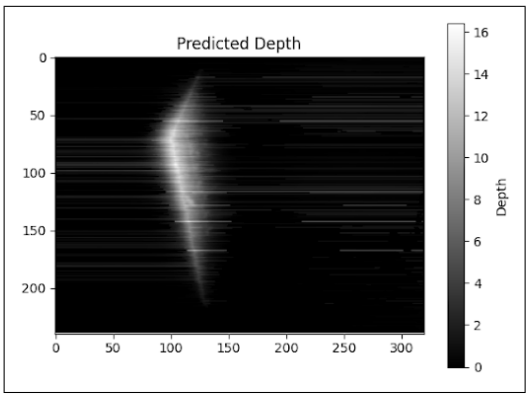

Figure 7. The reconstructed depth map after integrating the depth gradients.

3.6. Clipping Depth Map

There are some artifacts from surfing that remain (See Figure 7 for reference). Specifically, streaks of constant height can be observed outside the border of the object. To clip these streaks away, we zero all rows/columns where the height is below a certain threshold, when surfing that specific row/column. This yields a "clipped" version of the depth map, which can be seen in Figure 8.

4. Results

We were able to achieve an accuracy of **73.7%** on a test data set of 114 unseen tactile images. A prediction was deemed "correct" when its mean squared error fell under a threshold of 2 millimeters. Figure 9 shows the reconstructed depth map on an image from the testing data set. Figure 10 shows the reconstructed depth on an image from the testing data set.

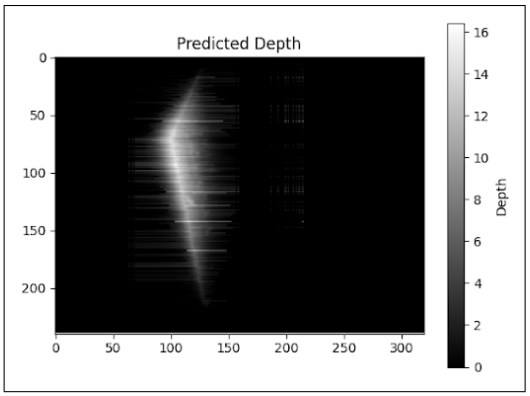

Figure 8. The reconstructed depth map after clipping in the x- and y-direction.

5. Discussion

In the future, there are improvements to make the code even more efficient and to resolve a boundary condition. The current bottleneck for the time complexity of the algorithm is the time to perform a query on the K-D tree. Data structures with fast lookup times can be explored. In addition, a different color space can be utilized to make the data more sparse, allowing clearer distinctions between colors and depth gradients. The current implementation integrates the depth gradients at the edges of the image with the assumption that the depths start at 0. In conditions where the depth is not 0 at the edges, different gradient surfing techniques can be explored to resolve this edge case. These enhancements will aim to refine the performance and reliability of our method.

References

[1] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network, 2014. 1

[2] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, Dinesh Jayaraman, and Roberto Calandra. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. 2020. 1

[3] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021. 1

[4] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 1

Algorithm 1 Gradient Surfing Algorithm (Positive X-Direction)

procedure PREDICTONIMAGE(grad_x2, grad_y2)
    reconstructed_image ← zeros_like(grad_x2)
    for $i \leftarrow 1$ **to** reconstructed_image.shape[0] $- 1$ **do**
        for $j \leftarrow 1$ **to** reconstructed_image.shape[1] $- 1$ **do**
            reconstructed_image$[i, j] \leftarrow$ reconstructed_image$[i - 1, j] -$ grad_y2$[i - 1, j]$
            reconstructed_image$[:, j] \leftarrow$ reconstructed_image$[:, j - 1] -$ grad_x2$[:, j - 1]$
        end for
    end for
end procedure

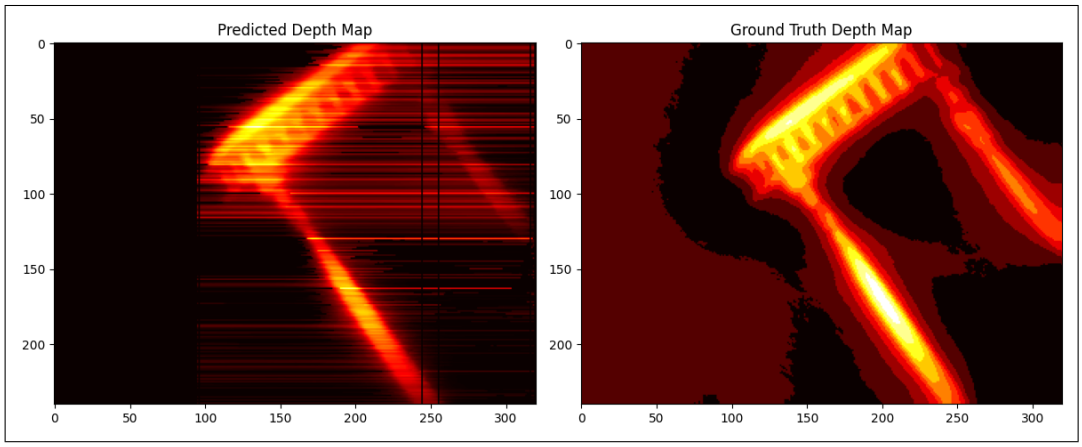

Figure 9. The predicted depth on a tactile image in the training data set.

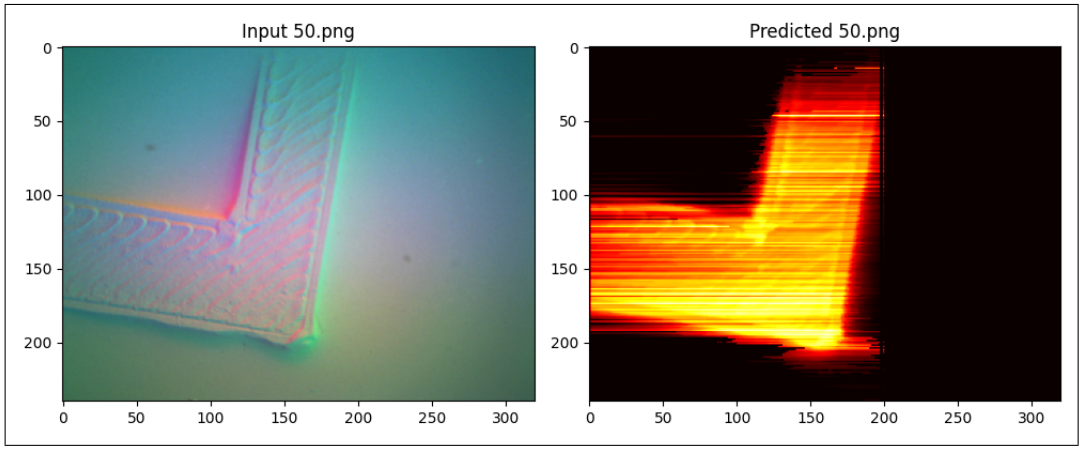

Figure 10. The predicted depth on a tactile image in the test data set.