Interpreting 3D Shapes from 2D Surface Contours

MIT 6.819/6.869 (2017 Fall) **Project Option 1**

October 2, 2017

1 Motivation

Humans are adept at perceiving a 3D surface from simple 2D line drawings, e.g., graphical depictions of continuous functions of two variables as shown in Fig. 1 [1]. What are the features that humans utilize in this extrapolation from 2D to 3D? Can we build a computer system that is also capable of completing such tasks?

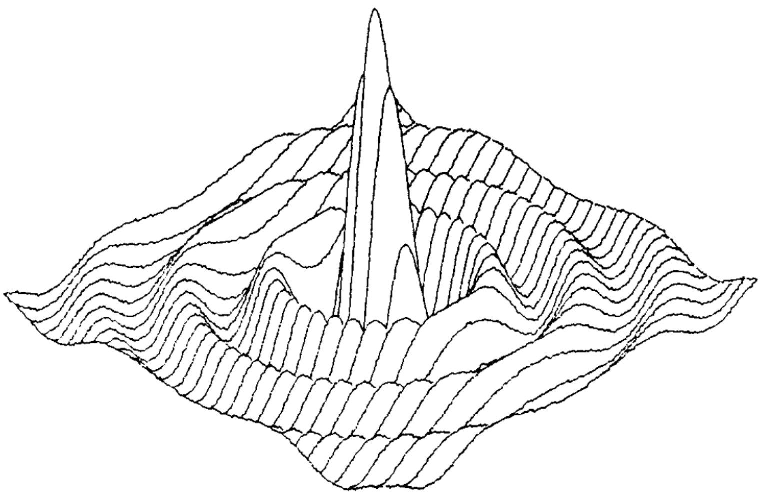

Figure 1: It is commonplace to graphically depict a continuous function of two variables as a surface seen from an oblique viewpoint merely by a set of curves. The 3D surface shape is immediately apparent [1].

2 Problem Formulation

The goal of this project is to build a system that is able to interpret plausible 3D shapes from 2D images of surface contours. Inputs to this system are 2D images of line drawings, which lie on the surface of the unknown 3D shapes (like those in Fig. 1).

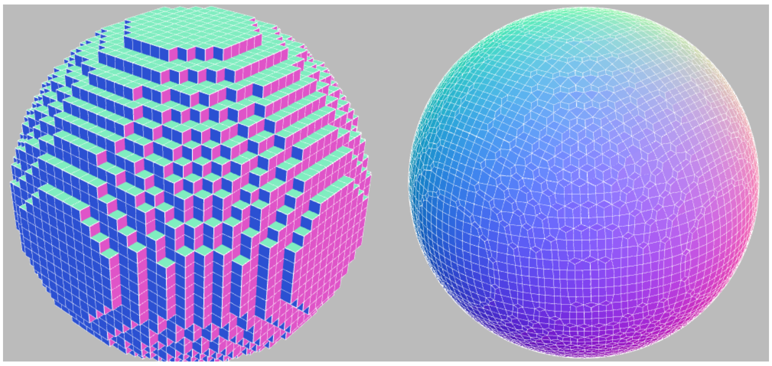

Figure 2: Voxel representation (left) vs. mesh representation (right) of the same sphere. Image source: `https://gamedev.stackexchange.com/questions/120014/smooth-mesh-from-voxel-grid`.

The desired outputs are the underlying unknown 3D shapes, in the format of voxels (Fig. 2 [left]) or mesh (Fig. 2 [right]). You could also represent 3D shapes in terms of 2.5D maps, such as surface normal maps, depth maps (Fig. 3), etc., if you find that makes the problem more tractable.

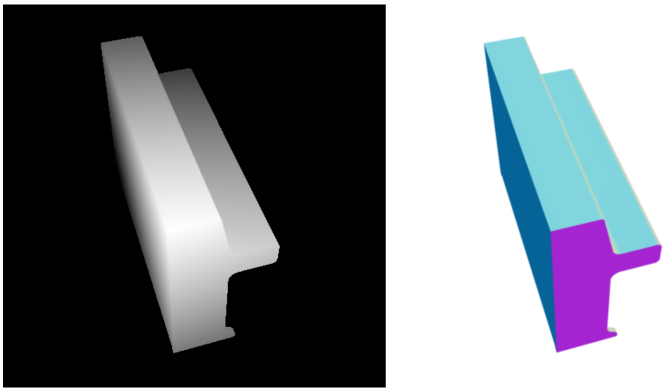

Figure 3: 2.5D maps: depth map (left) and surface normal map (right). In the depth map, brighter color means more shallow depth (i.e., smaller distance to the camera). In the normal map, surface normal vectors (x, y, z) are encoded into RGB channels, so different colors represent surface normal vectors pointing to different directions.

3 Possible Approaches

3.1 Neural Network

One way of tackling this problem is to train a neural network with 2D images of surface contours and their corresponding ground-truth 3D shapes.

The first step would be to generate a synthetic training set of image-shape pairs. If you go along this direction, it may be easier to use the voxel representation (rather than the mesh representation), because the voxel representation is more amenable to network operations, such as 3D convolution. Some neural network-based 3D reconstruction papers include [3] (`http://3dinterpreter.csail.mit.edu/`) and [2] (`https://shubhtuls.github.io/drc/`), both of which have their code publicly available. Alternatively, you could also use the 2.5D map representation, with which you will have an "image-in-image-out" system that outputs 2.5D maps.

After training on the synthetic set, you could test your trained network on some real hand-drawn images. Does it work on the test images? If yes, why do you think it can work? In other words, what features is the network utilizing in achieving the success? If not, can you make it fail even more? This process may reveal some interesting properties of your system.

3.2 Bayesian Probabilistic Approach

You could also take a Bayesian probabilistic approach, where you quantify the constraints or biases provided by each individual line, make assumptions about how lines relate to each other, and infer an optimal solution given those assumptions.

Specifically, the goal is to figure out posterior

$$p(\text{3D shape} \mid \text{surface contours}) \propto p(\text{surface contours} \mid \text{3D shape})p(\text{3D shape})$$

by defining your prior on the underlying 3D shape $p(\text{3D shape})$ and quantifying how likely it is to observe certain surface contour patterns given the 3D shape, i.e., likelihood $p(\text{surface contours} \mid \text{3D shape})$. You could start with a toy world where there are only, say, five types of shapes, including, e.g., spheres, cubes, etc..

References

[1] Kent A. Stevens. The visual interpretation of surface contours. *Artificial Intelligence*, 17(1):47 – 73, 1981.

[2] Shubham Tulsiani, Tinghui Zhou, Alexei A. Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Computer Vision and Pattern Regognition (CVPR)*, 2017.

[3] Jiajun Wu, Tianfan Xue, Joseph J. Lim, Yuandong Tian, Joshua B. Tenenbaum, Antonio Torralba, and William T. Freeman. *Single Image 3D Interpreter Network*, pages 365–382. Springer International Publishing, Cham, 2016.