Detecting Discontinuities for Surface Reconstruction

Yinting Wang\textsuperscript{1} Jiajun Bu\textsuperscript{1} Na Li\textsuperscript{2} Mingli Song\textsuperscript{1} Ping Tan\textsuperscript{3}

\textsuperscript{1} Zhejiang Provincial Key Laboratory of Service Robot
College of Computer Science, Zhejiang University
\{wangyt, bjj, brooksong\}@zju.edu.cn

\textsuperscript{2} Zhejiang International Studies University
nli@zisu.edu.cn

\textsuperscript{3} National University of Singapore
eletp@nus.edu.sg

Abstract

Photometric stereo algorithms produce a map of normal directions from the input images. The 3D surface can be reconstructed from this normal map. Existing surface reconstruction works often assume the normal map is integrable but contaminated by small scale non-integrable noise. However, real surfaces often contain large discontinuities such as occlusion boundaries and sharp depth changes, which break the integrable assumption commonly made in many works. Here, we propose a method to detect these discontinuities by combining multiple geometric cues with trained classifiers and a simple graph optimization. The surface is then reconstructed with the guidance of these detected discontinuities. Experiments show our method outperforms existing works.

1. Introduction

Shape-from-X algorithms such as photometric stereo, shape-from-shading or shape-from-texture, often reconstruct a normal map (also known as a gradient field) of a surface. The underlying surface can be recovered if this normal map is integrable. However, normal maps reconstructed from real data are almost never integrable for a number of reasons. Firstly, they could be noisy due to the noisy input data and/or the reconstruction algorithms. Secondly, the surface itself might not be smooth and contains sharp depth changes.

To handle these problems, variational methods have been proposed to find an integrable surface whose gradient field is closest to the original one [12]. Simchony et al. [11] recovered the surface by solving a Poisson equation. Agrawal et al. [1] computed the result by solving an algebraic system from an optimal set of image gradients. It was further extended to a generalized equation with spatially varying anisotropic weights in [2]. These methods typically work well with Gaussian noise, sparse outliers and some convex and concave edges, but they often generate poor results when there are significant depth discontinuities.

Integrable normal maps form a closed subspace [8] in the space of all normal maps. A common approach to enforce integrability is to represent the original surface by a linear combination of integrable basis functions [5, 9]. The surface is then reconstructed by estimating these combination coefficients. These methods share the common disadvantage as the first category that sharp depth changes will be smoothed out in the result.

However, surface discontinuities such as occlusion boundaries and sharp depth changes are common in real objects. These discontinuities cannot be modeled by Gaussian noise or sparse outliers. Among the previous works, [7, 14] consider large surface discontinuities, which rely on the smoothness of neighboring normals to localize them. Raskar et al. [10] designed an ‘NPR camera’ that can detect large depth discontinuities from photometric stereo images. Each of these cues for discontinuity detection cannot locate all discontinuous edges. Hence, we combine them to design a more reliable detection method.

In this paper, we focus on detecting discontinuities for surface reconstruction from a normal map obtained from photometric stereo. Our input includes the normal map and a set of photometric stereo images captured under the same viewpoint but different directional lighting. We first apply an over segmentation and edge detection (see Section 2) to identify a set of potentially discontinuous edges. We then use classifiers based on various cues (see Section 3) and a graph model (see Section 4) to identify true discontinuities. The final depth map is generated by integrating (see Section 5) the normal map.
mal map with the guidance of the discontinuity map.

2 Candidate Discontinuous Edges

Nearby pixels at surface discontinuities often have different surface orientations. Hence, we might apply an edge detection in the original normal map to identify some edges of shape discontinuities. For each pixel \( i \), we compute the angle formed by its normal and those of its four neighbors. We define the maximum value of these four angles as ‘Local Normal Angle’ of pixel \( i \), which can be used as an indication of local smoothness. We threshold the ‘Local Normal Angle’ to detect part of the discontinuous edges.

At the same time, since the original photometric stereo images are available, we apply the techniques in the ‘NPR camera’ [10] to detect additional discontinuous edges. Basically, this method calculates a ‘N-PR Camera Edge Score’ within \([0, 1]\) at each pixel to indicate the probability of that pixel locating on depth discontinuities. We threshold this probability to detect shape discontinuities.

As demonstrated in the experiments, these two methods might miss some important discontinuities. So, we further apply an over-segmentation to the color coded normal map to generate an overcomplete set of discontinuous edges. We apply the ‘quick-sift’ segmentation algorithm [13] here.

In the next, we smooth and combine these three results to form a set of candidate discontinuous edges. We first apply morphological dilation and erosion a few times to the detected edges to make them smoother. We then keep all edges from the over-segmentation and add the other edges to further split segments.

3 Discontinuity Detection

Each edge \( E \) separates two segments \( S^1 \) and \( S^2 \). We trace along \( E \) from one end to the other, and generate loops with interval of 15 pixels crossing \( E \). These parameters generally work well for images of \( 400 \times 400 \) resolution. We denote these loops as \( C^k (k = 1, 2, \cdots, K) \). The number of loops \( K \) depends on the length of \( E \). Shorter edges have no loops. Here we list all the cues used for discontinuity detection: 1) The ratio between the mean curl value [1] on the edge \( E \) and that within \( S^1, S^2 \). 2) The ratio between the mean ‘Local Normal Angle’ on the edge \( E \) and that within \( S^1, S^2 \). 3) The ratio between the mean ‘NPR Camera Edge Score’ on the edge \( E \) and that within \( S^1, S^2 \). 4) The angle between the mean normal direction within \( S^1 \) and \( S^2 \). 5) The ratio between the max curl value and the mean curl value on a loop \( C^k \) across the edge \( E \). 6) The ratio between the max and mean ‘Local Normal Angle’ on a loop \( C^k \) across the edge \( E \). 7) The ratio between the max and mean ‘NPR Camera Edge Score’ on a loop \( C^k \) across the edge \( E \). 8) We start from a random point in \( C^k \) and integrate along this loop until back to the starting point. If the whole loop \( C^k \) is integrable, the two ends of this integration path should have the same depth. Hence, we can use the difference of these two depths to evaluate the discontinuity at \( E \).

3.1 Discontinuity Classification

We then choose training data to build classifiers based on these cues. We use normal maps generated from known 3D models to get training data. We detect the candidate edges as described in Section 2 and manually label these edges as continuous or discontinuous. The whole training data set contains 200 continuous edges, and 232 discontinuous edges. For edges longer than 15 pixels, we generate closed loops crossing them automatically. Then all the eight cues described above are computed at each loop. For short edges without any loop, we only compute the first four cues.

In general, each of these cues cannot detect all continuous and discontinuous edges free of error. We build two SVM classifiers [4] from the training data. One uses all the eight cues and the second one uses the first four cues. Given a test edge \( E \), we generate \( K \) integration loops across it. When \( K \geq 1 \), we derive an 8D feature vector for each of these loops. We use the first SVM classifier to get \( K \) classification results, each includes a label (i.e. continuous or discontinuous) and a probability associated with this label. We then multiple these probabilities from different loops to obtain the probability for the edge \( E \) to be continuous and discontinuous. If \( K = 0 \), we compute a 4D feature vector which only contains the first four cues. The second SVM classifier is then applied to calculate the probability of \( E \) to be continuous and discontinuous.

4 Graph Optimization

We build a graph model to decide if an edge is continuous or not. Each edge is represented by a node in the graph. Two nodes are connected in the graph if their corresponding edges are connected in the image. We optimize a binary label, i.e. continuous or discontinuous, at each node by minimizing the cost function \( E = \lambda E_d + E_s \). The data cost \( E_d \) is defined as the product of the probability of the edge to take its label as calculated by our classifiers and the length of the edge. We weight the data cost by the length of each edge, such
that longer edges tend to be solely optimized according to their data term. If two neighboring edges have the same label, the smooth cost $E_s$ is 0. Otherwise, it is defined as the cosine of the angle spanned by these two neighboring edges. Basically, we assign stronger smoothness constraint between edges that travel along similar directions. In our experiments, we set $\lambda = 0.5$. We use the graph-cut algorithm [3] to optimize the overall cost function to decide the binary label at each edge.

5 Surface Reconstruction

Once the discontinuities have been detected, we use it to guide the surface reconstruction. Basically, we solve a depth $h(i)$ at each pixel $i$ by minimizing

$$\sum_i \sum_{j \in N_i} w_{ij} |h(i) - h(j) - g_{ij}|^2.$$  

Here, $N_i$ is the four neighbors of the pixel $i$, and $g_{ij}$ is the gradient given by the input normal map. $w_{ij}$ is set to 0 when one of $i, j$ is on the detected discontinuous edges. Otherwise, $w_{ij}$ equals to 1.

The detected discontinuities are used as a binary mask for reconstructing the surface. All integration paths across the discontinuous edges are cut off completely to preserve the shape discontinuities. Our method has a clear disadvantage that the detected discontinuous edges often break the image into multiple disconnected regions. Within each region, we can reconstruct a surface patch up to an unknown constant depth. The relative depth between different patches cannot be solved from the input normal map or the photometric stereo images. We might solve them if there are additional images from other viewpoints, like the configuration of the multiview photometric stereo [6]. Since the unknown relative depth is intrinsic to the input data, we believe it is better to reconstruct the surface up to its intrinsic ambiguity rather than to estimate a connected but distorted shape regardless of this ambiguity. In our experiments, for synthetic data, we set the average depth of each patch to its ground truth average value; for real data, we set the average depth manually.

6 Experiments

We first evaluated our discontinuous edge detection with the marked training data. We left one training image out in training and use it to test our method. Our method successfully identified 93.9% of the discontinuous edges and 96.5% of the continuous edges. To demonstrate the strength of combining multiple cues, we also trained classifiers with each one of the eight cues. We then used these classifiers to replace the one in our graph model to detect discontinuities. The success rates on discontinuous / continuous edges of these eight trials were 77.16% / 91%, 84.48% / 97%, 83.62% / 90.5%, 46.55% / 57%, 78.02% / 92%, 80.6% / 92%, 80.6% / 63.5%, 55.6% / 80% respectively. It is clear a single cue cannot detect many of the discontinuities. The first column of Figure 1 shows the discontinuous edges detected for the first example in Figure 2 by thresholding the ‘NPR Camera edge score’ (in green) and the ‘Local Normal Angle’ (in red). Yellow pixels are detected by both of them. The second column is the result generated by our method. Some important discontinuous edges are missed by the threshold based methods. This will cause imprecise surface reconstruction. The right part of Figure 1 compares their corresponding reconstruction error. The root mean square error (RMSE) are 7.3767, 7.0521 respectively. (The ground truth depth varies between 0 and 240.)

We compared with previous methods on synthetic data in Figure 2. For each example, we provide a color coded depth error in the first row and a rendering of the reconstructed surface in the second row. The first column shows the color coded input normal map and the ground truth surface. The other columns are the results generated from Diffusion [2], Shapelets [9], EM-based method [14] and our method respectively. Our method generate much smaller error in the reconstructed surface. In these two examples, our method broke the surface into 12 and 12 patches respectively.

We also compared with these methods on real data in Figure 3. For each example, we show the input normal map, one of the input images, the candidate discontinuous edges set, the detected discontinuities and the depth map generated by our method in the first row. The second row is the reconstructed surface rendered from a different viewpoint. The two examples were broken into 3 and 5 patches. We provide an image of similar viewpoint for validation. Note that this image is not used in our surface reconstruction. We framed some surface
7 Conclusion

We propose a method to reconstruct a surface from its normal map and some photometric stereo images. We use trained classifiers and a graph model to detect discontinuities to guide the surface reconstruction. In comparison with previous methods, our results preserve large discontinuities such as occlusion boundaries and sharp shape changes, which are common in real objects.

8 Acknowledgement

This work is supported by National Natural Science Foundation of China (61170142), National Key Technology R&D Program (2011BAG05B04), the Zhejiang Province Key S&T Innovation Group Project (2009R50009), and the Fundamental Research Funds for the Central Universities (2012FZA5017). Ping Tan is supported by the Singapore MOE Grant R-263-000-555-112 and R-263-000-620-112.

References


