Abstract

Handling depth images as a point cloud in a 3D data framework and performing planar segmentation in real-time requires heavy computation and it is a major challenge. Available planar-segmentation algorithms are mostly based on surface normals and/or curvatures, and consequently, do not provide real-time performance. In this abstract paper, we introduce a real-time planar-segmentation method for depth images avoiding any surface-normal calculation. A possible 3D application would strongly benefit from this real-time algorithm, more specifically, aiming at the reconstruction of indoor environments, which mainly consist of planar surfaces.

1. Introduction

Multiple planar segmentation algorithms have been proposed for point cloud datasets. Considering depth images as point clouds and performing planar segmentation requires heavy computation, because available planar segmentation algorithms are mostly based on surface normals and/or curvatures. In such algorithms, a real-time segmentation of planes is normally achieved by sacrificing the image quality.

As illustrated in Figure 1, a straightforward way to extract planes out of a depth image is to conventionally convert it to a point cloud and then apply an appropriate algorithm. In this abstract paper, as an alternative to this approach, we propose a real-time planar segmentation algorithm, which directly deploys the depth images as 2D frames containing 3D information. More specifically, the proposed algorithm involves high-level region growing based on a geometrical proposition stating that each pair of intersecting lines lies on a plane. To this end, first, edge contours should be detected in order to extract surfaces bounded between the 3D edges. After detecting 3D edges of a depth image, the algorithm searches for line-segments between the opposite edges and then merges the detected line-segments into planes, thereby facilitating planar segmentation. This abstract paper summarizes an original work which has been already published (submitted) by the authors [1, 2].

2. From 3D Edges to Segmented Planes

The proposed algorithm segments planar surfaces in three principal steps: (1) edge detection, (2) plane extraction, and (3) enhancement.

In the first step, four different types of 3D edges in a depth image are extracted: jump, extremum, corner, and curved. The jump edges result from occlusions or holes. The extremum edges include the local-minima or -maxima. The corner-edges emerge where two planes meet each other, and the curved-edges are resulting from intersection of actual planar and non-planar surfaces.
Figure 3. Planar segmentation of an depth image: (a) 3D edges detected; (b) detecting and labelling each individual line-segment; (c) merging of intersecting line-segments; (h) final outcome of the planar segmentation process (Note that for the sake of clarity, only diagonal lines have been shown. Besides this, the proposed enhancement methods improving the final outcome have not been applied here).

In the second step, we first extract all the (1-pixel-wide) string-segments bounded in between the edges. This step commences with scanning all the strings in all four directions (vertically, horizontally, left- and right-diagonally). Then, for each of these string-segments, we evaluate whether it is a line-segment or not. After finding all the line-segments, the algorithm attempts to merge the points on each pair of intersecting lines into a plane candidate. This step segments a depth image into its plane candidates.

In the third and final step, to improve the segmentation outcome, we perform several validation checks. First of all, we evaluate each plane candidate in terms of its curvature in a 3D space (for instance as a point cloud). Second, due to occlusions, a planar surface can be detected as various disconnected planar segments. Therefore, a merging process is needed to coalesce these apart segments into one actual plane. Finally, we evaluate the resulting segments in terms of their size to reject diminutive planes. Figure 3 illustrates the steps for planer segmentation of a depth image.

3. Evaluation Results

To evaluate the proposed algorithm, we have prepared a rich collection of datasets captured via Kinect as a depth-sensor. Figure 3 depicts three snapshots of the datasets and the corresponding results. The proposed algorithm has been initially designed to maximally benefit from parallel computing. The OpenMP library (OMP) and Compute Unified Device Architecture (CUDA) have been utilized in order to implement the parallel versions running on multi-core CPUs and GPU platforms, respectively. Table 1 shows the average executing time and the corresponding speedup factors for the various implementations of the proposed algorithm, which is implemented as a dual-layer pipeline, consisting of 3D edge detection and plane extraction. On the average, it takes 1102 ms to segment a depth image into its planes for a single-threaded implementation. A multi-threaded implementation decreases this execution time to 391 ms per frame (speedup factor of ≈ 3). And finally, this pipeline can produce planar segments in 17 ms per depth image by employing a GPU-based implementation (speedup factor of ≥ 60). Utilizing an interleaved pipeline enables us to gain an even higher speedup factor of ≥ 100 fps (9.4 ms per interleaved cycle).

4. Conclusion

In this abstract paper, we have introduced a real-time planar segmentation algorithm, which enables plane detection in depth images avoiding any normal-estimation calculation. First, the proposed algorithm searches for 3D edges in a depth image and then finds the line-segments located in between these 3D edges. Second, the algorithm merges all the points on each pair of the intersecting line-segments into a plane candidate. The developed 3D edge detection algorithm considers four different types of edges: jump, corner, curved and extremum edges. The complete system is implemented as a dual-layer execution architecture: the 3D edge detection and plane extraction layers. This enables a fast execution of both algorithms in parallel. By exploiting the GPU-based implementation, on the average, 3D edges are detected in 9.4 ms and planes are extracted in 7.8 ms. Therefore, the planar segmentation pipeline is capable of segmenting planes in a depth image with a rate of 58 fps. Utilizing pipeline-interleaving techniques further increases the rate up to 100 fps.

Acknowledgments

This research has been performed within the PANORAMA project, co-funded by grants from Belgium, Italy, France, the Netherlands, the United Kingdom, and the ENIAC Joint Undertaking.

References
