ABSTRACT

In Augmented Reality (AR) based remote collaboration, a remote user can draw a 2D annotation that emphasizes an object of interest to guide a local user accomplishing a task. This annotation is typically performed only once and then sticks to the selected object in the local user’s view, independent of his or her camera movement. In this paper, we present an algorithm to segment the selected object, including its occluded surfaces, such that the 2D selection can be appropriately interpreted in 3D and rendered as a useful AR annotation even when the local user moves and significantly changes the viewpoint.

Index Terms: Human-centered computing [Human computer interaction (HCI)]; Interaction paradigms—Mixed / augmented reality

1 INTRODUCTION

AR-based remote collaboration systems such as [3] and [6] allow the remote user to draw 2D annotations to instruct the local user to accomplish a task (e.g., equipment repair or maintenance) that involves the physical environment. In a typical setup, the local user’s camera’s frames are wirelessly streamed over a network to the remote user where he/she can draw onto a frame in order to send an annotation back to the local user. These annotations must “stick” to the selected 3D object as the local user’s camera moves, otherwise they will become useless or misleading. This is challenging for video-based object tracking algorithms since the object of interest can exhibit a large difference in appearance in different viewpoints. Gauglitz et al. [3] approached this problem by assuming a planar scene. To relax the planar scene assumption, they subsequently [4] proposed to incorporate an incrementally built Structure-from-Motion (SfM) 3D model of the unknown scene to infer the 3D positions of the 2D annotations. In particular, for selecting an object in the scene, they investigated several methods such as fitting a 3D plane to the points of a user’s 2D stroke using the median depth of the stroke. These methods consider only the depth information of the stroke points but utilize neither the rich 2D image cues nor geometrical context of the 3D point clouds.

Similar to the work of Gauglitz et al. [4], we aim to consistently render the remote user’s 2D annotation in every view when the local user moves his or her camera (e.g., an ellipse as shown in Figure 1(c)) using the sparse SfM point clouds constructed in the unknown scene. Unlike the planar annotation assumption of [4], however, we propose to take the input 2D annotation (e.g., Figure 1(a)) as a “user prior” and an additional 3D convexity prior to explicitly segment the object of interest in 3D, i.e., to label the 3D keypoints as foreground or background (Figure 1(b)). The idea of using convexity to help segmentation is rooted in psychophysical studies and has been reported in 2D interactive segmentation [5] as well as unsupervised 3D segmentation [10], but not in interactive 2D to 3D object selection.

2 3D POINT CLOUD SEGMENTATION GIVEN 2D USER HINT

We formulate the interactive 2D-3D object co-segmentation problem as minimizing the following energy function:

\[ E = E_{2D}(x, T^{-1}(y)) + E_{3D}(T(x), y), \]  

(1)

where \( x \) are the 2D points in the remote user-annotated frame \( l \) and \( y \) are the 3D points in the SfM model. All \( x \) and \( y \) are to be labeled as foreground or background in the optimization. \( T \) is a transformation that projects \( y \) to the image plane of the annotation, and \( T^{-1} \) projects \( x \) back to 3D. \( E_{2D} \) is a traditional 2D object segmentation energy, e.g., \( E_{2D} = E_{0}(i) + E_{p}(i, j) \), where \( E_{0} \) aims to separate the foreground and background appearances (e.g., color distributions) and \( E_{p} \) is used to encourage neighboring points \( i \) and \( j \) to take the same label. \( E_{3D} \) is a convexity-based term to encourage the user selection to be propagated to a large convex hull, where the transition from convex to concave parts is more likely the separation point between objects.

We solve Equation (1) by a piecewise optimization strategy, i.e., iteratively solving one energy term and refining the solution using the other term. Solving the first term in Equation (1) is known to be NP-hard. Fortunately a user prior, such as a bounding ellipse, can give a good initial estimation of the foreground and background color distributions so that an expectation-maximization-style algorithm can solve it efficiently. Solving the second term requires checking the convexity of every potential foreground labeling configuration and assigning a cost accordingly, which is computation-
Scalability. Solving the segmentation currently takes seconds to complete, which is entirely practical for our application.

User experience. With a manually labeled dataset, the objective mis-classification rate of a segmentation algorithm can be computed. But more important is the user’s subjective evaluation of the algorithm, since with an abstract annotation hint, such as the ellipse, a user may not care about small segmentation errors. In addition, as mentioned in [4], users may have multiple ways to select a 3D object in the given 2D view. We believe that user-centered experiments with a comprehensive dataset will help to better understand general user behavior and preferences and thus advance the design of 2D-3D co-segmentation algorithms.

3 RESULTS

Figure 2 compares the proposed method and the so-called “dominant plane” [4] and “median depth” [4] methods. In (a), the desk is identified as the dominant plane so all of the stroke points are interpreted as painted on the desk. In (b) the median depth of the stroke is assigned to all stroke points. One can see that both (a) and (b) rely on only the depth information on the strokes and thus mistakenly select background regions as foreground. The proposed method explicitly identifies the foreground points and can therefore correctly render the bounding ellipse on the box in any view.

4 LIMITATIONS AND ONGOING WORK

We are planning to build a dataset for a more thorough study on the 2D-3D object co-segmentation problem. There are few such resources available according to a very recent survey from the robotics community [1]. The most relevant one is the Object Segmentation Dataset [8], but its point clouds are constructed by a depth sensor from a fixed viewpoint and have quite different properties to our sparsely constructed SFM point clouds mainly captured surrounding an object of interest. For the same reason, the large body of RGBD datasets used in the computer vision community (e.g., [9]) are not directly suitable in our target AR application. With this new dataset, we will investigate three key aspects of the 2D-3D object selection problem:

- Robustness. While man-made objects are often convex and can be extracted using a convex prior as reported in the segmentation literature [10] and as observed in our preliminary experiments, it is not clear yet how well the algorithm may work for more complex objects, e.g., a paper box like the one shown in Figure 1, but squashed.
- Scalability. Solving the segmentation currently takes seconds for point clouds with tens of thousands of 3D key points and a WVGA resolution input image, with the 2D segmentation the current bottleneck. Plus, given the fact that the 3D keypoints are incrementally added as the local user scans the object and environment, the computational load on solving the 3D energy term will also increase. More investigation is needed on how dense a point cloud can be without leading to prohibitively long computation times for our application and how coarse a point cloud might be to nullify the convexity-based object inference.
- User experience. With a manually labeled dataset, the objective mis-classification rate of a segmentation algorithm can be computed. But more important is the user’s subjective evaluation of the algorithm, since with an abstract annotation hint, such as the ellipse, a user may not care about small segmentation errors. In addition, as mentioned in [4], users may have multiple ways to select a 3D object in the given 2D view. We believe that user-centered experiments with a comprehensive dataset will help to better understand general user behavior and preferences and thus advance the design of 2D-3D co-segmentation algorithms.

REFERENCES