

Abstract

Nowadays, three-dimensional (3D) point cloud has been an emerging medium to represent real-world scenes and objects. However, there is a considerable proportion of point clouds whose color attribute information is not captured during the acquisition process due to the device or environment limitations. This poses a great challenge for efficient management and utilization of point clouds. To address this problem, we introduce an automatic colorization scheme based on a deep generative network for 3D point clouds. The proposed approach uses the range images of point cloud geometry and trains a conditional generative adversarial network to predict the color of those images. Later, the color of each pixel in the colorized image is projected back to its corresponding point in the 3D point cloud. The experimental results demonstrate the efficacy of the proposed colorization approach in facilitating users to recognize and handle 3D point cloud data better.

Introduction

Due to the limitation of point cloud acquisition process, there is a considerable proportion of point clouds whose color attribute information is not captured. However, research on 3D point cloud colorization is relatively limited.

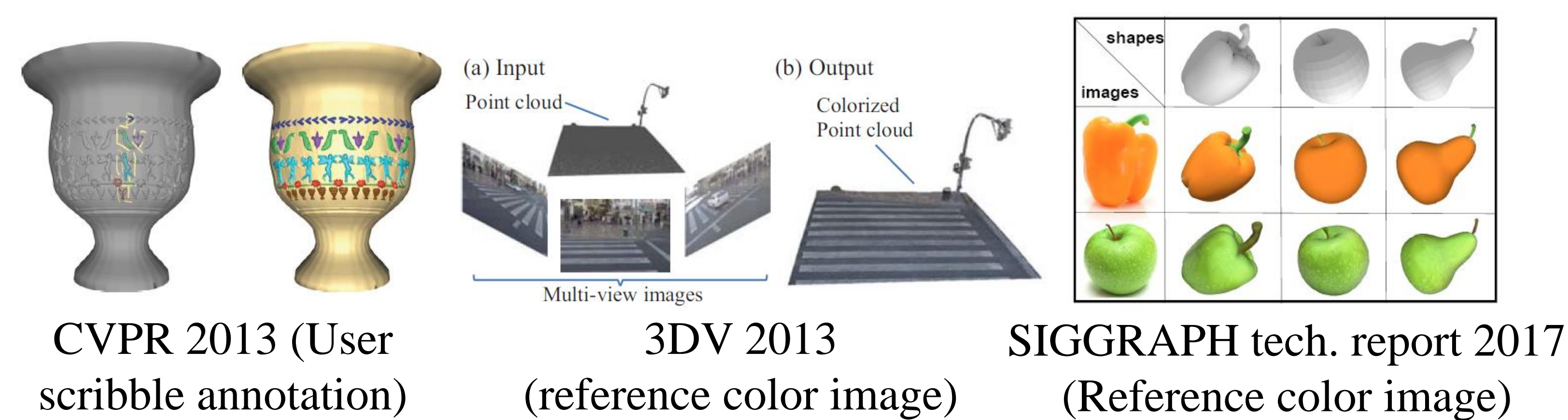


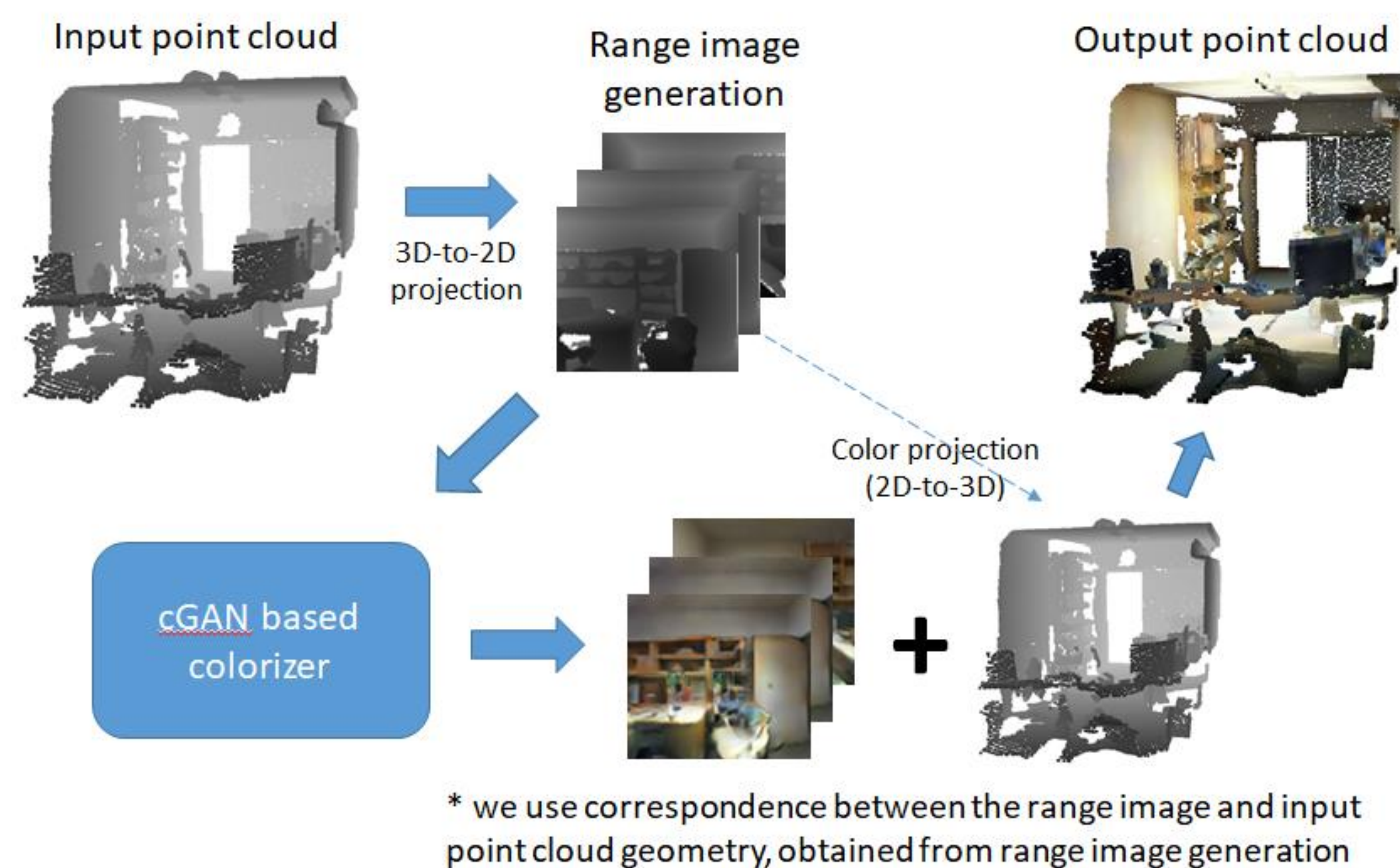
Figure 1. Two popular 3D object colorization method based on reference information: (a) scribble-based method, (c) and (b) color transfer method.

Conditional GAN based color generator

- We apply **conditional GANs** (cGANs) to generate color attribute from the input. We use Pix2pix architecture (CVPR 2017) to implement our color generator. (RGB-D data from S3DIS dataset)



Proposed Colorization Method

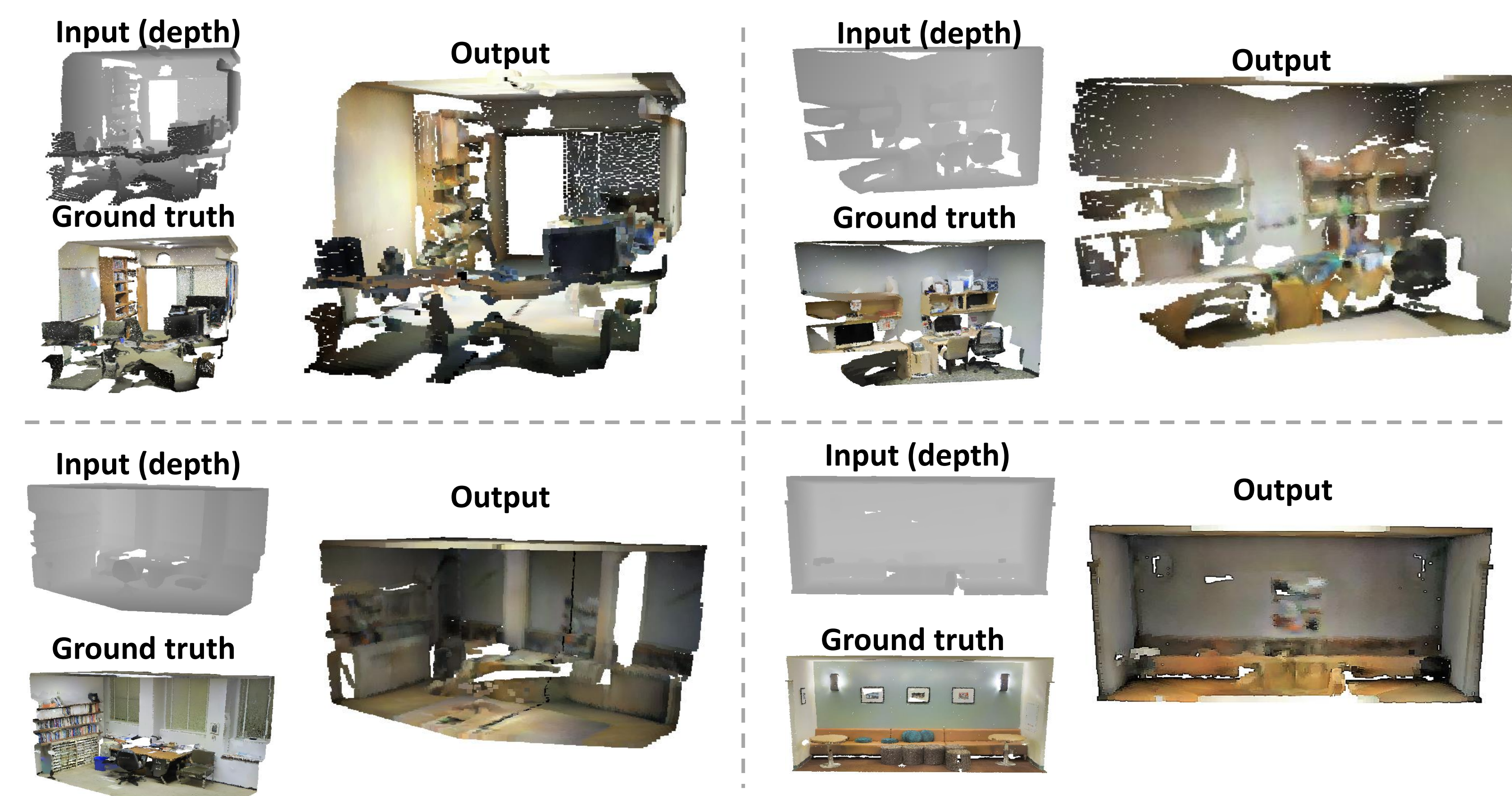


- Range (depth) image based point cloud colorization
 1. Range image generation from a point cloud
 2. Color attribute generation using conditional GAN (cGAN)
 3. Color attribute projection and merging

Test Summary and Samples



3D geometry of S3DIS Area 4 (22 point clouds for each office)



Limitation



Ambiguity of input depth map

Color mismatch from separated range images