A model (SynSin) that performs view synthesis from a single image.

Figure 2: The task: given an image and new viewpoint, the task is to synthesize an image at that new viewpoint.

Requirements:
- 3D scene understanding to model the 3D geometry
- Context understanding to fill in missing regions

Our key contributions to solve these challenges:
- A differentiable point cloud renderer to model the 3D geometry
- Using GAN techniques to fill in missing regions
- Training the whole model end to end in a self-supervised manner

SELF-SUPERVISED TRAINING

Training Inputs:
Assume we have an image \( I \), and a target image \( I_t \) of the same scene and the corresponding change in pose \( T \).

Setup:
- Input: \( I \)
- Output: \( I_t \)
- Project points using \( T \) and estimate point cloud \( P \)

Figure 3: Overview of the training pipeline.

- Given \( I \), SynSin predicts a set of features \( F \) at the same resolution as the image and a depth map \( D \)
- The depth map \( D \) is used to project the features into 3D, creating a point cloud \( P \)
- The point cloud is transformed according to the new view using \( T \)
- Our differentiable point cloud renderer \( g \) is used to render the point cloud into the new view, creating features \( P \)
- The refinement network \( R \) is used to fill in missing regions and synthesize an image of the scene at the new viewpoint
- A combination of GANs, L1, and perceptual losses are used to train the model

SynSin's 3D predictions.

Figure 4: SynSin's 3D predictions.

RESULTS

Baselines:
- Uses same info at train/test: Vox, based on [2]; Im2Im [3]
- Uses additional info at test: StereoMag [2 views] [4]; 3D Photos (depth), similar to [5]

Datasets:
- Train on Matterport [6], test on Matterpot and Replica [7]
- Train/test on RealEstate10K [4]

Table 1: Ablations and comparison to baselines on Matterport.

Table 2: Comparison with state of the art methods on RealEstate10K. The other methods use auxiliary information (e.g., depth or multiple views).

REFERENCES