SynSin: End-to-end View Synthesis from a Single Image

Olivia Wiles[1] **Georgia Gkioxari**[2] **Richard Szeliski**[3] **Justin Johnson**[2,4]

[1] VGG, University of Oxford [2] Facebook AI Research (FAIR) [3] Facebook [4] University of Michigan

Scan for Code!

Facebook AI Research

PyTorch3D

Scan for project page with video demos and the arXiv paper with more results!

GOAL

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

Figure 1: The task: given an image and new viewpoint, the task is to synthesize an image at that new viewpoint.

Requires:
- 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

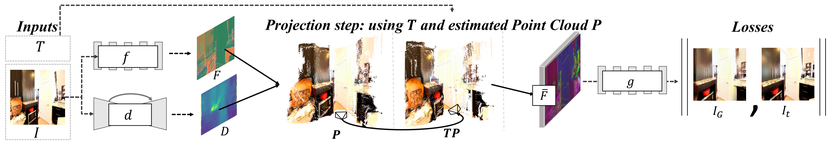


Figure 2: Overview of the training pipeline.

- Given I, SynSin predicts a set of features F at the same resolution as the image and a depth map D
- The depth map 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** is used to render the point cloud into the new view, creating features $\bar{F}$
- The **refinement network** g is used to fill in missing regions and synthesis an image of the scene at the new viewpoint
- A combination of GANs, L1, and perceptual losses are used to train the model

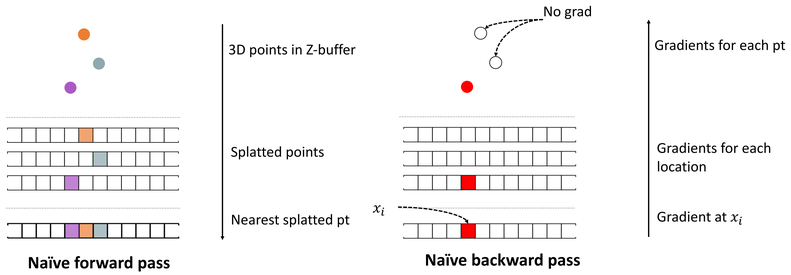
Testing Time:
Given **only** an image of an unseen scene and desired camera viewpoint, the model synthesizes new views of the scene.

NEURAL POINT CLOUD RENDERER

Desired properties:
- End-to-end differentiable
- Differentiate with respect to both the features and 3D positions to train f and d

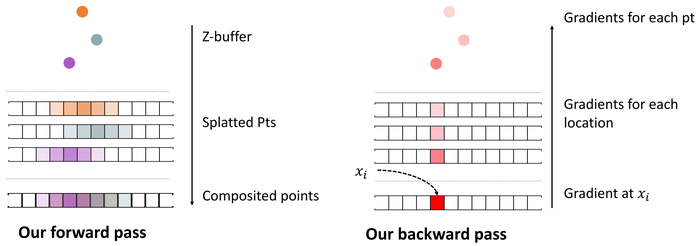
The renderer gives us features $\bar{f}_i$ at 2D locations p_i. As p_i is a function of the depth, this means we need derivatives with respect to both the projected features and the projected location. This is sufficient to train f and d.

NAIVE RENDERER



- Shown in 2D → 1D projections
- **Problems:** Local neighborhoods; Hard Z-buffer

OUR RENDERER



- **Larger neighborhoods:** 3D points are projected to a region with varying weights
  - Weights are proportional to 2D distance, allowing backpropagation to the 2D location p_i
- **Soft z-buffer:** Projected points are composited using alpha compositing
  - All points contribute to the final value
- **Performance:** 36 ms (forward pass), 5 ms (backward pass) for six 512^2 point clouds to 256x256 images. [1] takes ~1000ms (forward pass) and 2000ms (backward pass) for the same set of point clouds.

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]

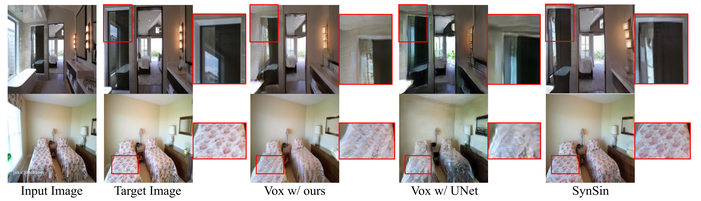


Figure 3: Results on Matterport. SynSin's results are higher quality than those of the baselines.

Figure 4: SynSin's 3D predictions.

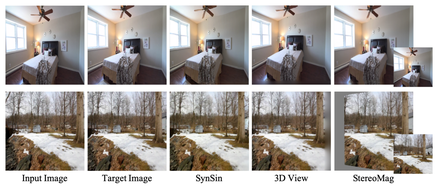


Figure 5: Results on RealEstate10K. SynSin outpaints better and can hallucinate missing regions.

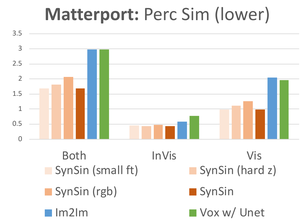


Table 1: Ablations and comparison to baselines on Matterport.

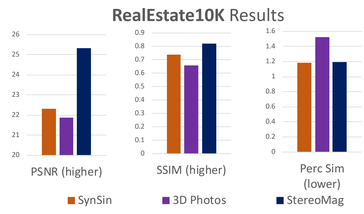


Table 2: Comparison with state of the art methods on RealEstate10K. The other methods use auxiliary information (e.g. depth or multiple views).

REFERENCES

[1] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. NeurIPS 2018.
[2] Vincent Sitzmann et al. "DeepVoxels: Learning persistent 3D feature embeddings." CVPR 2019.
[3]Tinghui Zhou et al. "View synthesis by appearance flow." ECCV 2016.
[4]Tinghui Zhou et al. "Stereo magnification: Learning view synthesis using multiplane images." ACM Transactions on Graphics 2018.
[5] https://ai.facebook.com/blog/powered-by-ai-turning-any-2d-photo-into-3d-using-convolutional-neural-nets/
[6] Chang et al. "Matterport3d: Learning from RGB-D data in indoor environments." 3DV 2017.
[7] Straub et al. "The Replica Dataset: A Digital Replica of Indoor Spaces," arXiv 2019.