Unsupervised Learning of Shape and Pose with Differentiable Point Clouds

Eldar Insafutdinov[1] and Alexey Dosovitskiy[2]

[1]Max Planck Institute for Informatics [2]Intel Labs

Motivation

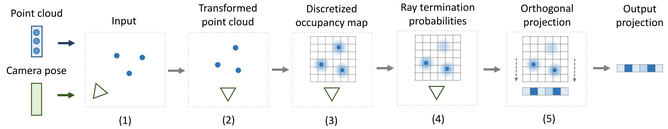
We live in a three-dimensional world and perceive it mainly via its two-dimensional projections. Based on these projections, we are able to infer the three-dimensional shapes and poses of the surrounding objects. Is it possible to design a learning system that perceives 3D from observing only two-dimensional projections?

Contributions:

- A system that predicts a detailed 3D shape from a single view of an object
- It learns only from **2D projections** and with **unknown camera poses**
- **Differentiable rendering of point clouds** that enables learning from 2D projections
- **An ensemble of pose estimators** that overcomes pose ambiguity

Code available!
https://eldar.github.io/PointClouds

Differentiable Point Cloud Rendering



1. Inputs: a point cloud $P = \{\langle \mathbf{x}_i, \mathbf{s}_i, \mathbf{y}_i \rangle\}_{i=1}^N$ and a camera pose c
2. Apply the projective transformation to the points in the point cloud: $\mathbf{x}'_i = T_c \mathbf{x}_i$
   - Transform T_c includes extrinsic c and intrinsic camera parameters
3. Represent each point as a Gaussian density function to enable gradient flow:

$$o(\mathbf{x}) = \text{clip}(\sum_{i=1}^{N} f_i(\mathbf{x}),\ [0,1]) \qquad f_i(\mathbf{x}) = c_i \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{x}'_i)^T \Sigma_i^{-1} (\mathbf{x}-\mathbf{x}'_i)\right)$$

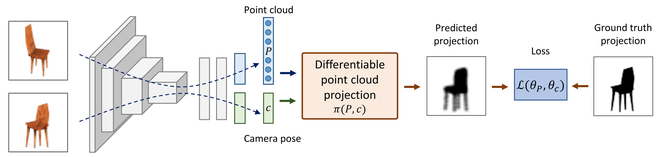
   - Covariance Σ_i can be a fixed isotropic or learned per point
   - The occupancy function $o(\mathbf{x})$ is discretized in two steps:
     - Put the points on a grid with trilinear interpolation using **tf.scatter_nd** op
     - Apply convolution with the kernel to the volume
4. Compute ray termination probabilities r from the occupancies o, similar to [1]:

$$r_{k_1,k_2,k_3} = o_{k_1,k_2,k_3} \prod_{u=1}^{k_3-1} (1 - o_{k_1,k_2,u}) \ \ if \ \ k_3 \leqslant D_3, \quad r_{k_1,k_2,D_3+1} = \prod_{u=1}^{D_3} (1 - o_{k_1,k_2,u}).$$

5. Finally, project the volume to the plane:

$$p_{k_1,k_2} = \sum_{k_3=1}^{D_3+1} r_{k_1,k_2,k_3} y_{k_1,k_2,k_3}.$$

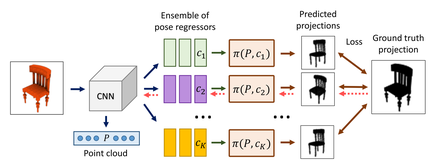
Learning Shape and Pose from 2D



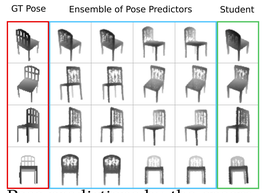
- Training data: a dataset D of views of K objects, with m_i views available for the i-th object: $D = \cup_{i=1}^{K} \{\langle x_j^i, \mathbf{p}_j^i \rangle\}_{j=1}^{m_i}$
- Inputs: two images of the same object at different viewpoints: x_1 and x_2
- We train a deep neural network F to predict:
  - the 3D shape (as a point cloud) from the first image: $\hat{P}_1 = F_P(x_1, \theta_P)$
  - the camera pose (a quaternion of rotation) from the second image: $\hat{c}_2 = F_c(x_2, \theta_c)$
- Compute 2D projection using Differentiable Point Cloud Renderer: $\hat{\mathbf{p}}_{1,2} = \pi(\hat{P}_1, \hat{c}_2)$
- Minimize reconstruction error between the computed projection and the ground truth:

$$\mathcal{L}(\theta_P, \theta_c) = \sum_{i=1}^{N} \sum_{j_1,j_2=1}^{m_i} \left\| \hat{\mathbf{p}}_{j_1,j_2}^i - \mathbf{p}_{j_2}^i \right\|^2.$$

Ensemble of Camera Pose Predictors



Our **ensemble of pose regressors** is designed to resolve camera pose ambiguity.



Pose predictions by the ensemble of regressors.

- Instead of a single pose regressor $F_c(\cdot, \theta_c)$, we introduce an ensemble of K pose regressors $F_c^k(\cdot, \theta_c^k)$
- Each regressor learns to specialize on a subset of poses and together they cover the whole range
- We train the system with the "hindsight" loss [2]:

$$\mathcal{L}_h(\theta_P, \theta_c^1, \ldots, \theta_c^K) = \min_{k \in [1,K]} \mathcal{L}(\theta_P, \theta_c^k).$$

- In parallel we train a single regressor by using the best model from the ensemble as the teacher
- The loss for training the student is an angular difference between two quaternions of rotation:

$$L(q_1, q_2) = 1 - \text{Re}(q_1 q_2^{-1} / \left\| q_1 q_2^{-1} \right\|)$$

Pose ambiguity: segmentation masks used for supervision look very similar from different camera views.

Experiments

- We evaluate our method on the 3 shape categories (chairs, cars, airplanes) of the ShapeNet dataset [4]
- We render 5 views of each model (random camera azimuth and elevation, sampled from $[0°, 360°)$ and $[-20°, 40°]$)
- We use Chamfer Distance as an evaluation metric for shape reconstruction:

$$d_{Champf}(P^{gt}, P^{pred}) = \frac{1}{|P^{pr}|} \sum_{\mathbf{x}^{pr} \in P^{pr}} \min_{\mathbf{x} \in P^{gt}} \left\| \mathbf{x}^{pr} - \mathbf{x} \right\|_2 + \frac{1}{|P^{gt}|} \sum_{\mathbf{x}^{gt} \in P^{gt}} \min_{\mathbf{x} \in P^{pr}} \left\| \mathbf{x}^{gt} - \mathbf{x} \right\|_2$$

- To evaluate pose estimation we report the accuracy (at a 30° threshold) and the median error (in degrees)

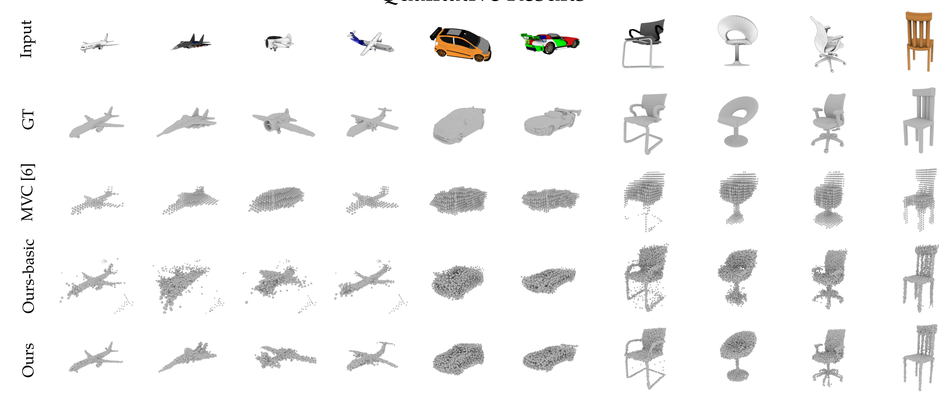
	Resolution 32				Resolution 64		Resolution 128	
	DRC [1]	PTN [4]	Ours-V	Ours	Ours-V	Ours	EPCG [5]	Ours
Airplane	8.35	3.79	5.57	4.52	4.94	3.50	**4.03**	**2.84**
Car	4.35	3.94	3.88	4.22	3.41	2.98	**3.69**	**2.42**
Chair	8.01	5.10	5.57	5.10	4.80	4.15	**5.62**	**3.62**
Mean	6.90	4.27	5.01	4.61	4.39	3.55	**4.45**	**2.96**

Table 1: Shape prediction accuracy for the experiments with known camera pose.

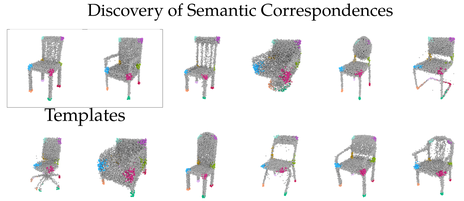
	Shape (D_{Champf})			Pose (Accuracy & Median error)					
	MVC [6]	Ours-basic	Ours	GT pose [6]		MVC [6]		Ours-basic	Ours
Airplane	4.43	7.22	**3.91**	**0.79**	10.7	0.69	14.3	0.20 100.2	0.75 **8.2**
Car	8.43	4.14	**3.91**	**0.90**	7.4	0.87	**5.2**	0.49 42.8	0.82 7.4
Chair	6.51	4.79	**4.30**	0.85	11.2	0.81	**7.8**	0.50 31.3	**0.86** 8.1
Mean	6.46	5.38	**4.04**	**0.85**	10.0	0.79	9.0	0.40 58.1	0.81 **7.9**

Table 2: Shape and pose prediction accuracy. Ours-basic is our model with a single pose regressor.

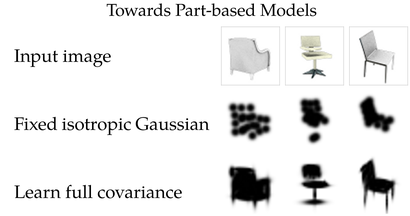
Qualitative Results



Discovery of Semantic Correspondences



Towards Part-based Models



References

[1] S. Tulsiani et al. "Multi-view supervision for single-view reconstruction via differentiable ray consistency" in CVPR, 2017.
[2] A. Guzmán-rivera et al. "Multiple Choice Learning: Learning to Produce Multiple Structured Outputs" in NIPS, 2012.
[3] A. X. Chang et al. "ShapeNet: An information-rich 3D model repository." Technical Report.
[4] X. Yan et al.. "Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision" in NIPS, 2016.
[5] C.-H. Lin et al. Learning efficient point cloud generation for dense 3D object reconstruction" in AAAI, 2018.
[6] S. Tulsiani et al. "Multi-view consistency as supervisory signal for learning shape and pose prediction" in CVPR, 2018.