NeMo: Neural Mesh Models of Contrastive Features for Robust 3D Pose Estimation

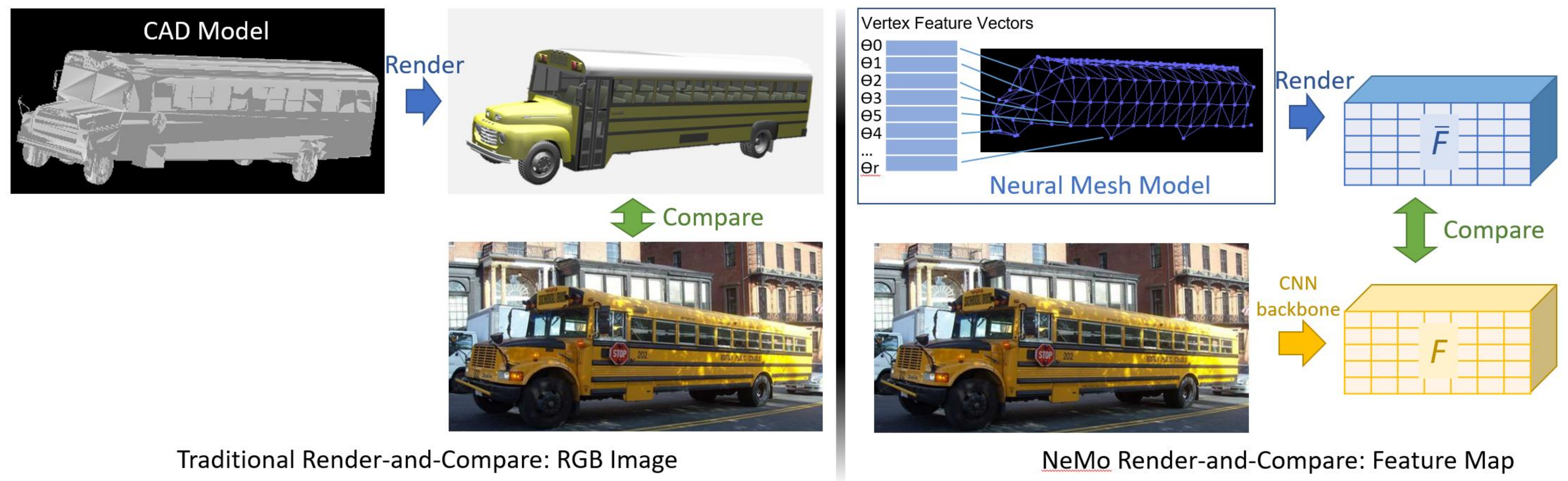
Angtian Wang, Adam Kortylewski, Alan Yuille

Johns Hopkins University

Code: https://github.com/Angtian/NeMo.

Motivation

Computer Vision with Analyze-by-Synthesis



NeMo Perform 3D pose estimation via feature level render-and-compare

Current render-and-compare approaches to image analysis operate on **pixels intensity level**. Which lead these methods have following limitations:

- The reconstruction loss in inherently hard to optimize w.r.t. pose parameters.
- Requires detailed and instance specific mesh models.

Contribution

This work proposes NeMo, a 3D object pose estimation pipeline conducts neural feature level render-and-compare. NeMo combines a prototypical geometric representation of the object with **a generative model of neural network features** that are invariant to object details. Which allows NeMo have following advantages:

- Reconstruction loss is very easy to optimize with standard gradient descent (one global optimum)
- Requires only a very crude prototypical 3D mesh.
- SOTA 3D pose estimation performance and **exceptional robustness** to out distributed cases, i.e. **occlusions, unseen views.**

Method

We define the likelihood of the feature representation F as:

$$p(F|\mathfrak{N}_y, m, B) = \prod_{i\in\mathcal{FG}} p(f_i|\mathfrak{N}_y, m) \prod_{i'\in\mathcal{BG}} p(f_{i'}|B).$$

where $\mathfrak{N}_y$ is the 3D mesh representation, m is the camera pose, B is mixture parameters. Then the foreground and background feature likelihoods:

$$p(f_i|\mathfrak{N}_y, m) = \frac{1}{\sigma_r\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma_r^2}\|f_i - \theta_r\|^2\right) \quad p(f_{i'}|B) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}\|f_{i'} - \beta\|^2\right)$$

where θ is the per vertex feature vector, β is the clutter feature vector, σ is the variance of each distribution.

During **training**, we constrain the variances that $\{\sigma^2 = \sigma_r^2 = 1|\forall r\}$, the max likelihood loss:

$$\mathcal{L}_{ML}(F, \mathfrak{N}_y, m, B) = -C \sum_{i\in\mathcal{FG}} \|f_i - \theta_r\|^2 + \sum_{i'\in\mathcal{BG}} \|f_{i'} - \beta\|^2$$

To efficiently learn θ_r, β, and the feature extractor using the whole training set, we use the contrastive keypoint representation learning pipeline[5] to train NeMo with loss:

$$\mathcal{L}(F, \mathfrak{N}_y, m, B) = \mathcal{L}_{ML}(F, \mathfrak{N}_y, m, B) + \mathcal{L}_{Feature}(F, \mathcal{FG}) + \mathcal{L}_{Back}(F, \mathcal{FG}, \mathcal{BG})$$

where,

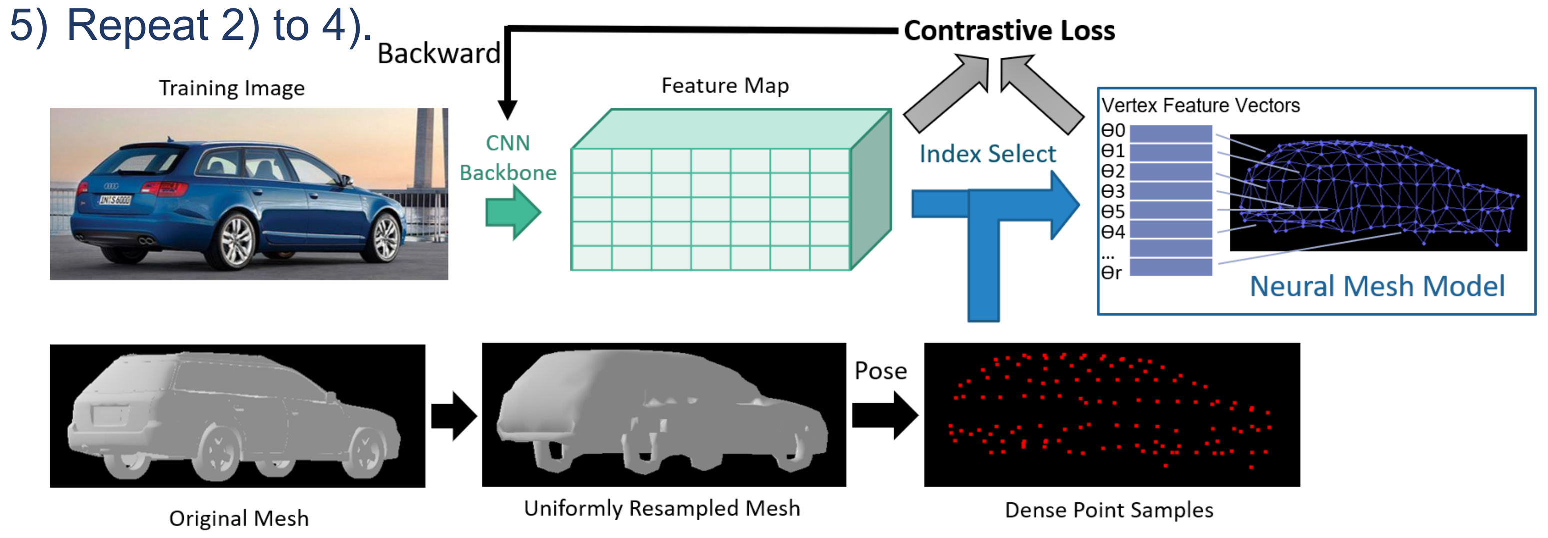
$$\mathcal{L}_{Feature}(F, \mathcal{FG}) = -\sum_{i\in\mathcal{FG}} \sum_{i'\in\mathcal{FG}\setminus\{i\}} \|f_i - f_{i'}\|^2 \qquad \mathcal{L}_{Back}(F, \mathcal{FG}, \mathcal{BG}) = -\sum_{i\in\mathcal{FG}} \sum_{j\in\mathcal{BG}} \|f_i - f_j\|^2$$

During **Inference**, the object pose is optimized via maximizing the model likelihood:

$$p(F|\mathfrak{N}_y, m, B, z_i) = \prod_{i\in\mathcal{FG}} [p(f_i|\mathfrak{N}_y, m)p(z_i{=}1)]^{z_i} [p(f_i|B)p(z_i{=}0)]^{(1-z_i)} \prod_{i'\in\mathcal{BG}} p(f_{i'}|B)$$

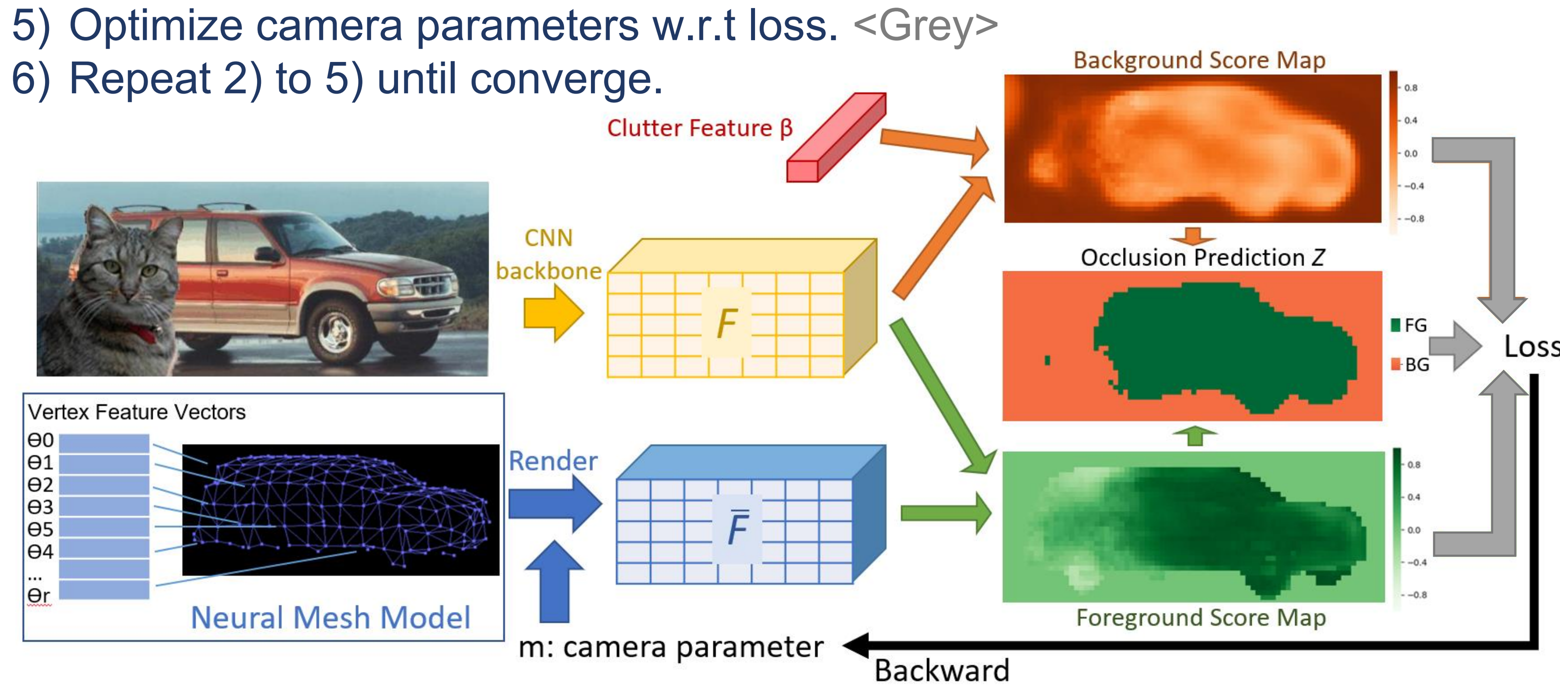
Train NeMo

1) Project vertices to the image manifold with the pose annotation. <Black>
2) Extract features from training image. <Aquamarine>
3) Learn a per vertex feature representation (NMM) via dense point position on feature map. <Blue>
4) Optimize the CNN Backbone with contrastive loss such that vertex features become different from each other.<Grey>
5) Repeat 2) to 4).

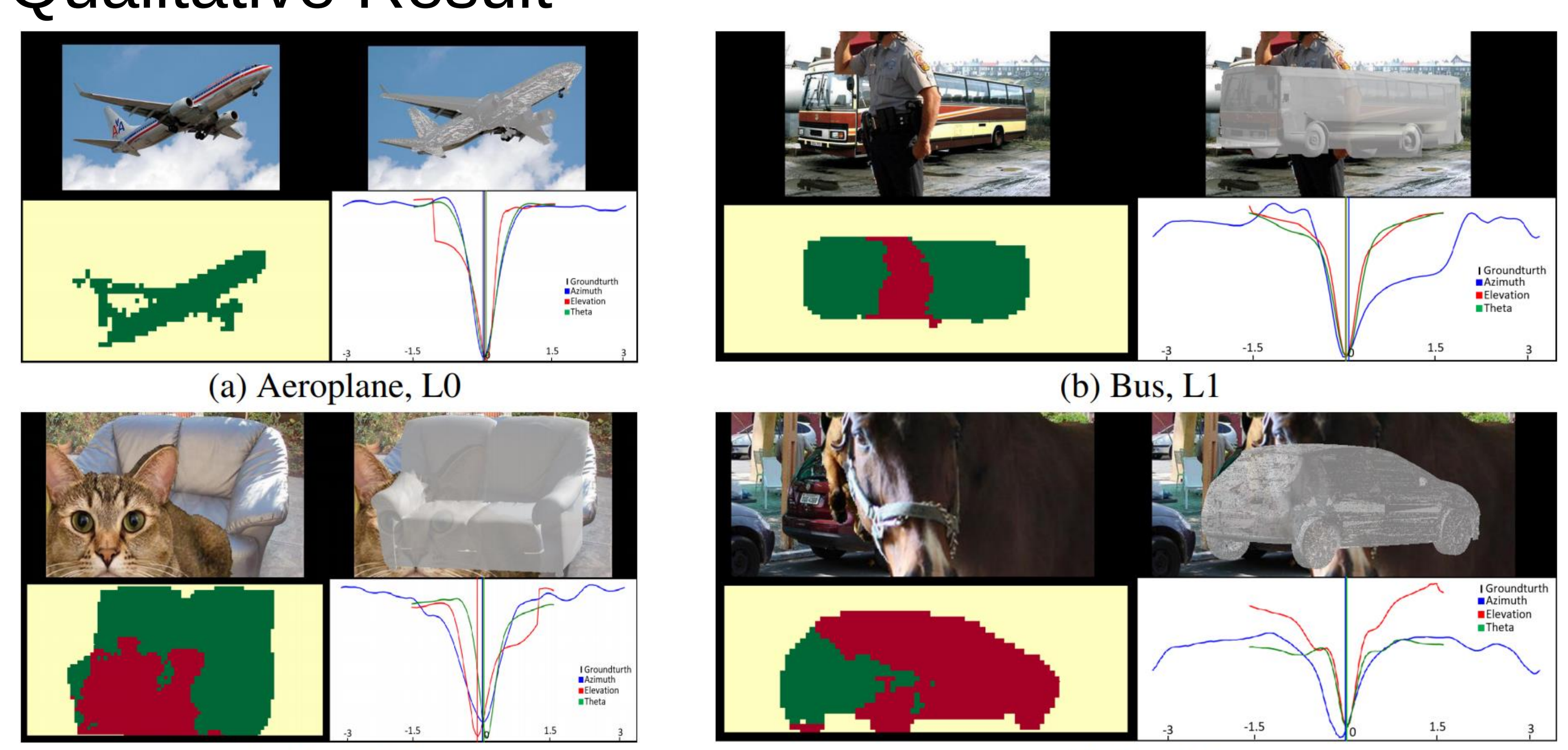


Pose Estimation with NeMo

1) Extract features (F) using the trained backbone. <Yellow>
2) Render a feature map (F') using the trained NMM under a (random initialized) camera pose. <Blue>
3) Compare F and F' to compute a foreground score map. <Green>
4) Occlusion prediction using the clutter feature β. <Orange>
5) Optimize camera parameters w.r.t loss. <Grey>
6) Repeat 2) to 5) until converge.



Qualitative Result

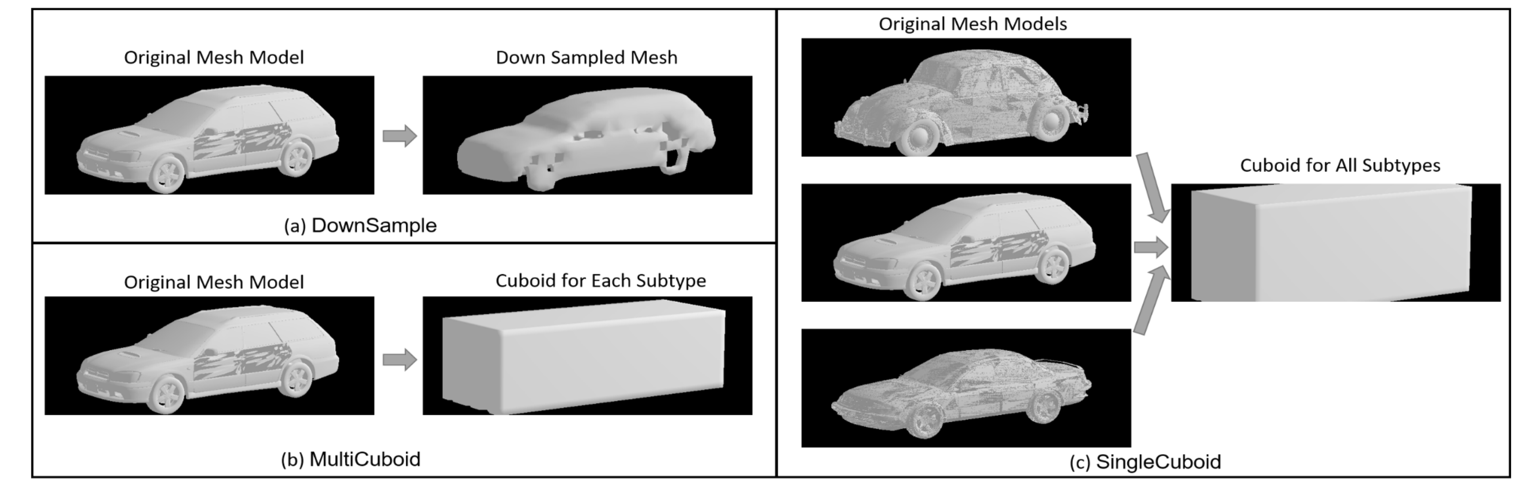


(a) Aeroplane, L0 (b) Bus, L1 (c) Sofa, L2 (d) Car, L3

Top-left: the input image; Top-right: A mesh superimposed on the input image in the predicted 3D pose; Bottom-left: The occluder localization result, yellow -> background, green -> non-occluded area, red -> occluded; Bottomright: The loss landscape for each individual camera parameter respectively.

Pre-process Meshes

Experiment setup including 3 mesh sampling methods: DownSample, MultiCuboid and SingleCuboid.



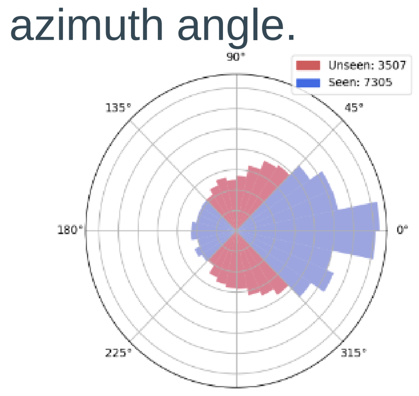
Robust 3D Pose Estimation Under Occlusion

Experiments on PASCAL3D+[1] (L0) and Occluded-PASCAL3D+[2] dataset (L1 to L3 with increasing ratio of occluded area on the object). We use accuracy under given threshold of the error between the predicted and groundtruth rotation matrix.

Evaluation Metric	$ACC_{\frac{\pi}{6}}\uparrow$				$ACC_{\frac{\pi}{18}}\uparrow$			
Occlusion Level	L0	L1	L2	L3	L0	L1	L2	L3
Res50-General	85.9	66.5	50.8	38.0	38.6	26.5	18.8	12.2
Res50-Specific	86.5	71.4	56.4	41.3	39.7	29.9	21.5	13.8
StarMap[3]	**89.4**	71.1	47.2	22.9	59.5	34.4	13.9	3.8
NeMo	84.1	73.1	59.9	41.3	60.1	45.1	30.2	14.5
NeMo -MultiCuboid	86.7	**77.3**	**65.2**	**47.1**	**63.2**	**49.9**	**34.5**	**17.8**
NeMo -SingleCuboid	85.0	75.8	63.5	45.8	57.7	43.7	30.4	15.1

Generalization to Unseen Views

The PASCAL3D+ dataset is split into 4 bins based on the ground-truth azimuth angle.



Evaluation Metric	$ACC_{\frac{\pi}{6}}\uparrow$		$ACC_{\frac{\pi}{18}}\uparrow$	
Data Split	Seen	Unseen	Seen	Unseen
Res50-General	91.7	37.2	47.9	5.3
Res50-Specific	91.2	34.7	47.9	4.0
StarMap	**93.1**	49.8	68.6	13.5
NeMo-MultiCuboid	88.6	**54.7**	**70.2**	**31.0**
NeMo-SingleCuboid	88.5	54.3	68.6	27.9

The image on the left shows how we separate the PASCAL3D+ dataset based on the azimuth annotations. The Blue bins indicates azimuth range used during training. The number and histogram shows the azimuth distribution of PASCAL3D+ testing set.

ObjectNet3D[4]

$ACC_{\frac{\pi}{6}}\uparrow$	bed	bookshelf	calculator	cellphone	computer	cabinet	guitar	iron	knife
StarMap	40.0	**72.9**	21.1	**41.9**	62.1	79.9	38.7	2.0	6.1
NeMo-MultiCuboid	**56.1**	53.7	**57.1**	28.2	**78.8**	**83.6**	**38.8**	**32.3**	**9.8**
$ACC_{\frac{\pi}{6}}\uparrow$	microwave	pen	pot	rifle	slipper	stove	toilet	tub	wheelchair
StarMap	86.9	**12.4**	45.1	3.0	**13.3**	79.7	35.6	46.4	17.7
NeMo-MultiCuboid	**90.3**	3.7	**66.7**	**13.7**	6.1	**85.2**	**74.5**	**61.6**	**71.7**

References

[1] Yu Xiang (2014) Beyond pascal: A benchmark for 3d object detection in the wild. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)

[2] Angtian Wang (2020) Robust object detection under occlusion with context-aware compositionalnets. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

[3] Xingyi Zhou (2018) Starmap for category-agnostic keypoint and viewpoint estimation. In Proceedings of the European Conference on Computer Vision (ECCV)

[4] Yu Xiang (2016) Objectnet3d: A large scale database for 3d object recognition. In Proceedings of the European Conference Computer Vision (ECCV)

[5] Yutong Bai (2020) Coke: Localized contrastive learning for robust keypoint detection. arXiv preprint arXiv:2009.14115