Canonical Surface Mapping via Geometric Cycle Consistency

Nilesh Kulkarni, Abhinav Gupta, Shubham Tulsiani

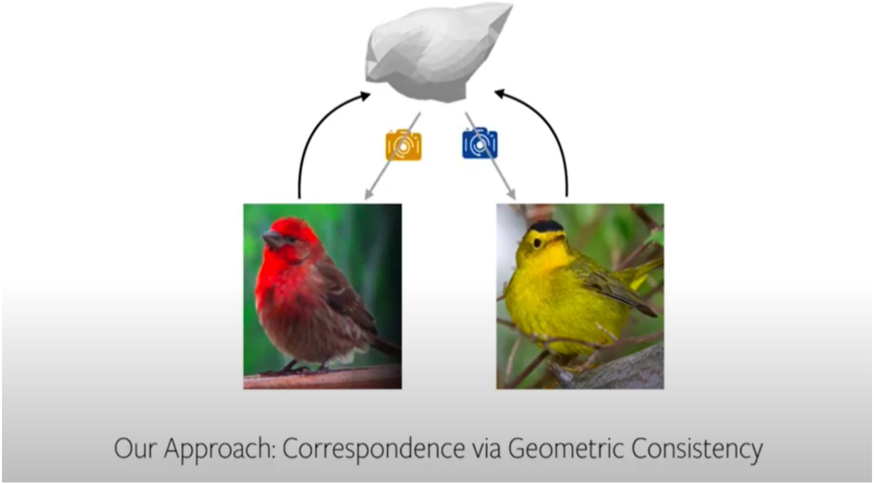
ICCV 2019

Problem this paper is solving

Given an image, map pixels on the object to the corresponding locations on a 3D model of the category the image lies in.

Why we care about mapping to canonical 3D models?

We care about pixel correspondence between two images.

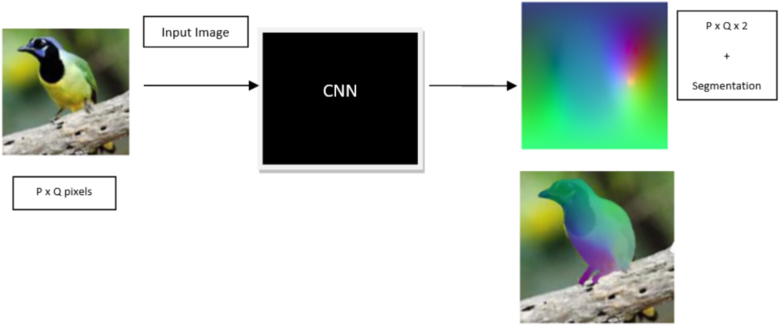


Key Insights

- Using consistency loss as an objective
- Allows dense correspondences without any correspondence supervision

Approach Explained - CNN Model

The CNN is a 5-layer UNet Model with 4x4 kernel size at each layer.



Approach Explained - How learning happens?

Geometric Cycle Consistency Loss

$$L_{cyc} = \Sigma_{p \in I_f} ||\mathbf{p'} - \mathbf{p}||_2^2; \mathbf{p'} = \pi(\phi(C[\mathbf{p}])) \tag{1}$$

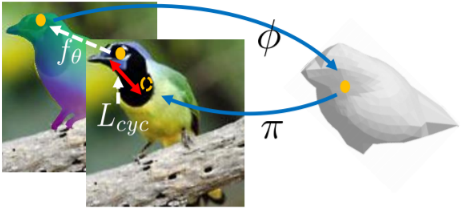


Figure 3: Geometric Cycle Consistency Loss. A pixel mapped to **u** by CSM function f_θ gets mapped onto the 3D template via ϕ. Our loss enforces that this 3D point, when projected back via the camera π, should map back to the pixel.

Approach Explained - Visibility Constraints

Occluded points can also minimize loss and misguide it. The author's solution is to discourage the CNN from predicting values that map to self-occluded regions.

$$L_{vis} = \Sigma_{p \in I_f} max(0, z_p - D_\pi[\mathbf{p'}]) \tag{2}$$

Approach Explained - Foreground Mask Prediction

- Background pixels are ignored*, hence an additional per-pixel mask predictor is trained using ground-truth masks.
- The CNN model itself is modified to include the mask prediction and one output is added per-pixel as a probability of it belonging to the foreground.

*This also helps in creating the visuals.

Approach Explained - Without Camera Pose Supervision

- Use predicted camera parameters instead of known ones (since this data might not be available). Differentiate w.r.t camera parameters as well.
- Known - Foreground Mask annotations (from a Mask-RCNN model) and canonical template shape for each category
- Learns - Jointly learn pose and CSM prediction

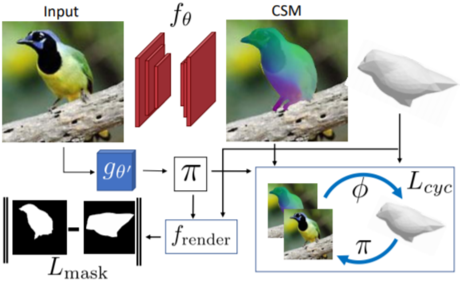
Pose predictor (A Resnet Model)

$$\pi = g_{\theta}(I) \tag{3}$$

Mask Loss

$$L_{mask} = ||f_{render}(S, \pi) - I_f||^2 \tag{4}$$

Approach Explained - Overall Training Objective and Procedure



$$L_{tot} = L_{div}(g_{\theta'}(I)) + \Sigma_{i=1}^{N_c} c_i(L^i_{cyc} + L^i_{vis} + L^i_{mask}) \qquad (5)$$

Results

Keypoint Transfer Task - on CUB-200-2011 (birds) and PASCAL 3D+ (cars) datasets

Metrics used - Percentage of Correct Keypoints (PCK) and Keypoint Transfer AP (APK)

Annotation	Method	Birds		Cars	
		PCK	APK	PCK	APK
KP + Seg. Mask	CMR [18]	47.3	22.4	44.1	16.9
Pose + Syn. Data	Zhou et. al [54]	-	-	37.1	10.5
Pose + Seg. Mask	CSM (ours) w/ pose	56.0	30.6	51.2	21.0
Seg. Mask	Dense Equi [40]	34.8	11.1	31.5	5.7
	VGG Transfer	17.2	2.6	11.3	0.6
	CSM (ours)	48.0	22.4	40.0	11.0

Limitations and Future Work

- Trained using unoccluded images in video, so some errors are present
- If shapes vary significantly across instances, then there might be inconsistencies
- The segmentation depends on Mask-RCNN model - could be a point of failure for videos
- Future work may try to include temporal consistency for videos. Also, try prediction using only consistency losses, without segmentation masks.

Discussion Questions

- Why is a Mask Loss needed? Why didn't the authors just use consistency between camera pose and CSM prediction?
- How are all the models - the renderer, the ResNet, the UNet trained together?