Weak Multi-View Supervision for Surface Mapping Estimation

Nishant Rai [1,2] [*,+], Aidas Liaudanskas [1] [*], Srinivas Rao[1]
Rodrigo Ortiz Cayon[1], Matteo Munaro[1], Stefan Holzer[1]
[1]Fyusion Inc., [2]Stanford University

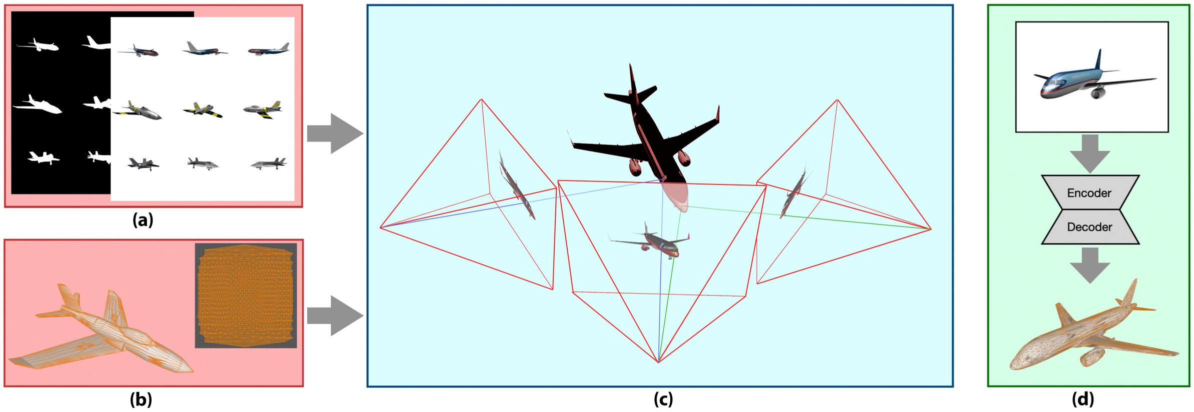

Figure 1: **Approach overview:** Given **(a)** a dataset of multi-view images and segmentation masks of a category-specific object, along with **(b)** a single template mesh with UV coordinates, our method trains a network **(c)** exploiting information from multiple views using reprojection cycles and learn an instance-specific mesh using deformations. **(d)** At test time, our model predicts an instance-specific mesh and a surface mapping from a single image.

Abstract

We propose a weakly-supervised multi-view learning approach to learn category-specific surface mapping without dense annotations. We learn the underlying surface geometry of common categories, such as human faces, cars, and airplanes, given instances from those categories. While traditional approaches solve this problem using extensive supervision in the form of pixel-level annotations, we take advantage of the fact that pixel-level UV and mesh predictions can be combined with 3D reprojections to form consistency cycles. As a result of exploiting these cycles, we can establish a dense correspondence mapping between image pixels and the mesh acting as a self-supervisory signal, which in turn helps improve our overall estimates. Our approach leverages information from multiple views of the object to establish additional consistency cycles, thus improving surface mapping understanding without the need for explicit annotations. We also propose the use of deformation fields for predictions of an instance specific mesh. Given the lack of datasets providing multiple images of similar object instances from different viewpoints, we generate and release a multi-view ShapeNet Cars and Airplanes dataset created by rendering ShapeNet meshes using a 360° camera trajectory around the mesh. For the human faces category, we process and adapt an existing dataset to a multi-view setup. Through experimental evaluations, we show that, at test time, our method can generate accurate variations away from the mean shape, is multi-view consistent, and performs comparably to fully supervised approaches.

1. Introduction

Understanding the structure of objects and scenes from images has been an intensively researched topic in 3D computer vision. Classically, researchers applied Structure from Motion (SfM) and multi-view stereo techniques to sets of images to obtain point clouds [10], which could be converted to meshes using triangulation techniques [28, 16]. Later, representing object shapes as PCA components [1, 2, 22] or as 3D-morphable models [3] gained popularity. Unlike SfM techniques, the benefit was the ability to generate a mesh even from a single image, as mesh generation was reduced to a model fitting problem [20, 27].

[*] Nishant Rai and Aidas Liaudanskas have contributed equally
[+] Work done during Nishant's internship at Fyusion

Subsequently, with the rise of CNNs [17] and their impressive performance in image-to-image tasks [12], many explored the possibility of generating 3D point clouds [25, 26] and meshes [9, 34] with CNNs. However, most of these approaches relied on extensive supervision and well-curated datasets [4, 38, 37], thus requiring a lot of effort to extend them to work on new object categories. Instead of having fully annotated data, unsupervised or self-supervised techniques aim to reduce the amount of data and priors that is needed for training. Among those, some works targeted category-specific reconstruction [14, 13, 36]. Unfortunately, these approaches still rely on moderate supervision, in particular in the form of labelled keypoints [13], which are often hard to compute or require expert annotation. The work of Kulkarni *et al.* [19] dismissed this requirement, but only for computing dense pixel correspondences or surface mappings without actually predicting a mesh. Their approach relies on an underlying static mesh which leads to an approximate surface mapping learning. More recently, [18] relaxed these constraints by allowing articulation for the meshes which alleviates this issue to some extent. However, using unrestrained meshes has not been explored yet. We claim that multi-view cues can provide a useful learning signal for such a setup.

We build on the aforementioned works and predict dense surface mappings along with a 3D mesh. We train our network to be multi-view consistent by taking advantage of multi-view cycles and using a novel reprojection loss which results in improved performance. To the best of our knowledge, utilizing instance specific deformation to learn using self-consistency in a multi-view setting has not been explored yet. Since we only use cycles, i.e., reprojections updates of the current prediction, our method is computationally more efficient than other approaches that require differentiable renderings [34] or iterative processing approaches [6]. Our method does not necessarily require multiple views, and it can also work with single-view images. Our approach is weakly-supervised as it only requires weak labels like rough segmentation masks, camera poses and an average per-category mesh for training. We discuss how to compute these weak labels for new categories and datasets in section 5.

As our approach relies on exploiting multi-view correspondences, we need a dataset consisting of multiple images or a video of object instances of a given category. For faces, we adapt and use the 300W-LP [43] dataset. For cars and airplanes, we render and release our own dataset, filtering out degenerate meshes from ShapeNet [4]. Besides the weak labels required for our method, we also release additional data like depth maps, origin-centered meshes and images rendered at high resolution to promote research in multi-view computer vision tasks. This task also has several applications as dense mappings are useful, for example, in localizing or replacing certain parts of an object, like license plates for cars, or pasting tattoos/filters onto faces.

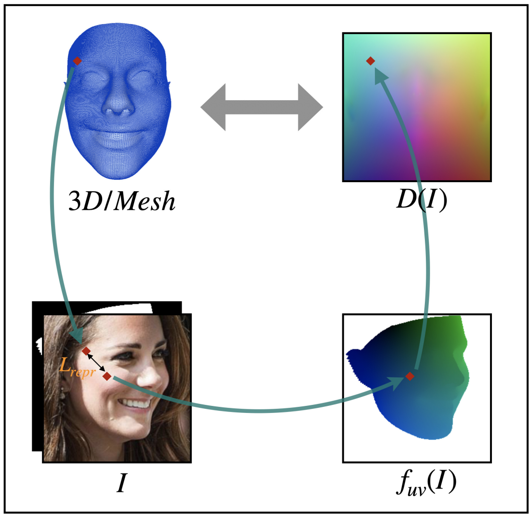

Figure 2: Illustration of a single reprojection cycle. For any image pixel, we: 1. Jump from RGB to UV space (via f_{uv} i.e., a neural network); 2. Jump from UV space to 3D space (via predefined category specific mesh using interpolation); 3. Project the 3D point back into RGB image space and take the displacement of reprojected pixel location against starting pixel location as our error signal for learning better UV predictions. We refer to this procedure as reprojection consistency loss.

The key contributions of our work are:

1. a novel weakly-supervised approach which learns the surface mapping and 3D structure of a category from a collection of multi-view images.

2. a training regime exploiting cycle consistency across different views learning instance-specific meshes by modelling deformations. Provided with an image at test time, we can produce a unique mesh along with a dense correspondence surface map.

3. a multi-view dataset of ShapeNet cars and ShapeNet airplanes created by rendering a smooth camera trajectory around the ShapeNet meshes and an adaptation of the 300W-LP dataset so that it is suited for our approach.

2. Related Work

There has been a lot of research on mesh reconstruction, learning dense correspondences and multi-view constraints. In this section, we focus on the recent approaches which exploit deep architectures.

DNNs for 3D geometry: Recently, there has been a push towards learning novel ways of using neural networks by imparting prior knowledge to solve specific tasks. One such problem is learning 3D geometry from images. Some initial works in this direction are transform-invariant networks [25, 26]. These works are dealing with point clouds, hence they lack the connectivity information needed for meshing. Subsequent works tried to address this by proposing iterative deformation of an ellipsoid [34, 9] to generate a mesh matching a given shape. Alternatively, there has also been research into estimating the mesh by encoding it as an image-like position map, like in [7], where the mesh is represented as a function of UV space. Similarly, [30] encode the mesh in a normalized object coordinate system as an X-NOCS map which is a function of image space. These mesh encoding maps are great for solving category-specific tasks, as these maps are a function of UV space [7], and the same UV can be considered for all instances of a given category. Our work builds over this position map idea (see D in Fig. 2) to express a 3D mesh. More recently, works exploiting differentiable rendering [6, 21] or differentiable ray-marching [29, 23] also seem to be promising. In our work, we try to bypass these expensive differentiable rendering computations by exploiting cycle consistency.

Cycle Consistency: The idea of using cycles or relationships between pixels has been extensively exploited for object tracking, reconstruction and alignment. Its success has been marked by notable works such as unpaired image-to-image translation [42], SfM [40], depth estimation [8] and dense correspondences [41]. Cycle consistency has also been applied in time for learning correspondences in a video [35]. More recent works by [19, 18] use pixel correspondences to form a cycle and establish dense correspondences of a given mesh. We build on this idea by cycling through different views of an object and allowing deformed 3D meshes to improve our generated dense mappings.

Deformations: Representing variation in a given category's instance has been a popular idea. The seminal work by Blanz *et al*. [3] tries to represent any given face as a deformed version of a common face model. Similarly, there have been many works which try to represent a shape using PCA shape models [1, 2, 22]. With the popularity of neural networks, several works started using them to estimate these deformation parameters [39, 31, 32, 33, 13]. We build on such ideas and integrate mesh deformation alongside multi-view cycles to improve model performance.

Surface Mapping Understanding: Recent works [19, 18] have attempted to assign semantically consistent meaning to points on objects. A popular approach has been to utilize UV parametrization to create this mapping. While earlier approaches relied on dense supervision, using geometric consistency has been shown to be a promising alternative. However, the use of static underlying meshes hampers the mapping performance when using a consistency loss. Existing approaches do not utilize multi-view signals to improve performance.

Category-Specific Reconstruction: The aim of category-specific reconstruction is to obtain a shape model of an object when provided with a number of images of its instances. Previous works [13, 15] solve this by using extensive pixel-level annotations as supervision. Recent work by Kulkarni *et al*. [19, 18] attempts to relax these constraints. Our work extends them by employing multi-view consistency in order to produce a novel/unique mesh per instance. However, our approach targets a different use-case compared to [18] as they focus on articulation while we attempt to model size and modest shape variations as well.

3. Approach

Our goal is to extract the underlying surface mapping of an object from a 2D image without having explicit annotations during training. We predict an instance-specific shape with a common topology during inference while training the model in a weakly-supervised manner without using dense pixel-wise annotations. We utilize segmentation masks, camera poses and RGB images to learn to predict the 3D structure. We exploit information present in multi-view images of each instance to improve our learning. For each category, we utilize a single mesh as depicted in Fig. 1. Note that we only require a single 2D RGB image for inference.

3.1. Preliminaries

UV Parametrization: Using UVs (refer to Fig. 2) as a parametrization of the mesh onto a 2D plane is an effective technique to represent texture maps of a mesh. We represent the mesh as a function of UV space, i.e., as a position map similar to the representation used in [7]. Given a pixel in the image, instead of directly predicting its 3D position on a mesh, we map each pixel to a corresponding UV coordinate and in turn map each UV coordinate onto the position map (which is equivalent to a mesh). The key difference between the position map used by us and [7] is that our position maps represent a frontalized mesh located in a cube at the origin, whereas the position map used by [7] represents a mesh projected onto the image. We refer to this position map as $D(I)$ which maps UV points to their 3D locations, i.e., $D(I) \in \mathcal{R}^2 \rightarrow \mathcal{R}^3$. D represents an NN which takes an image I as input and predicts a position map. Similarly, we represent the function mapping image locations to their UVs by $f_{uv}(I) \in \mathcal{R}^2 \rightarrow \mathcal{R}^2$. f_{uv} represents a NN which takes an image I as input and predicts the UV location of each pixel. For brevity, we write $D(I)$ as D and $f_{uv}(I)$ as f_{uv} in the single-view case. Refer to Fig. 2, bottom right.

Reprojection Cycle: Since our mesh is frontalized and located in a cube at origin, we represent the transformation from this frontalized coordinate system to the mesh in image by a transformation matrix ϕ_π, where π represents the camera parameters corresponding to this matrix. The cycle starting from pixel $p \in I$, going through UV and 3D and back to UV would be represented by $p = \phi_\pi * D(f_{uv}(p))$. This single reprojection cycle is depicted in Fig. 2 and holds true if we have the ground truth f_{uv}, D and ϕ_π.

3.2. Reprojection Consistency

We train a CNN $f_{uv}(.)$, which predicts a UV coordinate for each input image pixel. Similar to [19], our approach derives the learning signal from the underlying geometry via the reprojection cycle. Starting from a pixel $p \in I$, we can return back to the image space by transitioning through UV, $3D$ and back to image using the transformation matrix ϕ_π to result in a pixel p'. Finally, the difference between p' and p gives us the supervisory signal for learning the involved components:

$$L_{repr} = \sum_{p \in I} (p - p')^2; p' = \phi_\pi * D(f_{uv}(p)) \tag{1}$$

An issue with the cycle defined above is it does not handle occlusion, resulting in occluded points being mapped onto the pixel in front of it. We handle this by the use of an additional visibility loss as proposed in [19]. We consider a point to be self-occluded under a camera π if the z-coordinate of the pixel when projected into camera frame/image space is greater than the rendered depth at the point. To compute the rendered depth map D_π for the given mesh instance under camera π, we use the average mesh D_{avg}. The visibility loss L_{vis} is defined as:

$$\begin{aligned} L_{vis} &= \sum_{p \in I} max(0, p'[2] - p'_{avg}[2]) \\ p'_{avg} &= \phi_\pi * D_{avg}(f_{uv}(p)) \end{aligned} \tag{2}$$

Here, $p'[2]$ represents the z coordinate of the pixel when the corresponding point in 3D is projected into image space. In accordance with earlier works [19] and also our own findings, we utilize segmentation masks to mask the points for which we compute L_{repr} and L_{vis}. This leads to greater stability and performance during training.

3.3. Learning Shape Deformations

To allow for learnable deformations, we learn residuals over an average category-specific mesh D_{avg}. We model this residual as a deformed position map $\in \mathcal{R}^2 \rightarrow \mathcal{R}^3$ which is predicted by a CNN (d). We represent the actual position map $D(.) \in \mathcal{R}^2 \rightarrow \mathcal{R}^3$ as $D(I) = D_{avg} + d(I)$.

We add a regularizing loss to enforce smoothness over the predictions. The final loss becomes $L_{def} = Smoothness(d(I)) + L_2Reg(d(I))$.

3.4. Multi-view Cycle Consistency

[19, 18] have looked at independent instances to learn effective surface mappings in the absence of dense labels. However, there have not been many attempts to exploit multiple views of an instance in such a weakly supervised setting. Utilizing multiple views of an instance allows extension to new modalities such as videos as well. We explore the setup where we have multiple corresponding views of an object along with the associated camera poses.

In order to exploit multi-view information during training and learn effective $f_{uv}(.)$ and $D(.)$, we propose to introduce a multi-view consistency loss which goes from a pixel from one view and jumps into another view (refer to the red and blue correspondences in Fig 1 Middle). Take two images from different views of the same object, I_1 and I_2, which have camera parameters π_1, π_2. ϕ_π represents the transformation from 3D space to the image space with camera parameters π. D_1, D_2 represents $D(I_1)$ and $D(I_2)$. Then, we can define preliminaries for UV consistency loss as follows:

$$\begin{aligned} \widetilde{p}_{1\rightarrow 2} &= \phi_{\pi_2} * D_1(f_{uv_1}(p_1)), \quad \text{where } p_1 \in I_1 \\ \widetilde{p}_{2\rightarrow 1} &= \phi_{\pi_1} * D_2(f_{uv_2}(p_2)), \quad \text{where } p_2 \in I_2 \end{aligned} \tag{3}$$

Note that $\widetilde{p}_{1\rightarrow 2}$ refers to the projection of point p_1 from image space I_1 to I_2. Assuming correct predictions, it should map p_1 to its corresponding semantic location in I_2. Therefore, the UV prediction of the corresponding point in I_2 should remain the same as the one in I_1. We follow the same route in the opposite direction to get an additional error signal. We summarize the above in 5. $f_{uv_1}(.)$ and $f_{uv_2}(.)$ represent the learnt functions mapping pixel locations in image I_1, I_2 to their corresponding UVs respectively.

$$\begin{aligned} L_{uv}^{(1\rightarrow 2)} &= \sum_{p_1 \in I_1} (f_{uv_1}(p_1) - f_{uv_2}(\widetilde{p}_{1\rightarrow 2}))^2 \\ L_{uv}^{(2\rightarrow 1)} &= \sum_{p_2 \in I_2} (f_{uv_2}(p_2) - f_{uv_1}(\widetilde{p}_{2\rightarrow 1}))^2 \\ L_{uv} &= L_{uv}^{(1\rightarrow 2)} + L_{uv}^{(2\rightarrow 1)} \end{aligned} \tag{4}$$

3.5. Overall Model

In this section, we put all the above components together and summarize the overall model. We use Deeplab V3+ [5] with more skip connections and a Resnet 18 encoder to model $f_{uv}(.)$ and $D(.)$. We have a separate decoder sub-network for each task (UV prediction, segmentation, deformation-field prediction). We train our system end-to-end to optimize the combination of the losses discussed earlier:

$$\begin{aligned} L = \lambda_{repr} * L_{repr} + \lambda_{vis} * L_{vis} \\ + \lambda_{def} * L_{def} + \lambda_{uv} * L_{uv} \end{aligned} \tag{5}$$

We use $\lambda_{repr} = 1, \lambda_{vis} = 1, \lambda_{uv} = 1, \lambda_{def} = 0.025$ in our experiments. Although it is preferred to have multi-

Dataset	Num Imgs	Num.	RGB	D.	Front.	ST.
300WLP	550,878	13000	✓	✓		✓
WSM-Faces	50,000	13000	✓	✓	✓	✓
XNOCS	102,000	5100	✓	✓	✓	
WSM-Cars	50,000	500	✓	✓	✓	✓
WSM-Planes	50,000	500	✓	✓	✓	✓

Table: Statistics for comparable datasets. Num. refers to number of unique instances. D. refers to the depth. Front. refers to frontalized meshes; ST. refers to having smooth multi-view transitions.

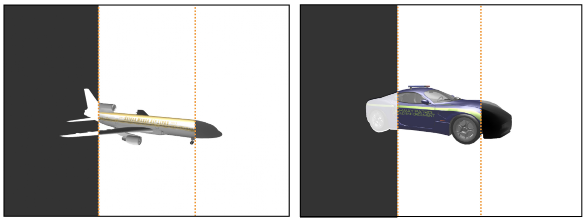

Figure 3: Data from WSM-Planes and WSM-Cars being released. We also release camera poses and origin-centered ShapeNet meshes along with segmentation mask, image and depth maps.

view instances in our dataset, our model extends to datasets without multiple instances as well.

4. Experiments

The goal of our framework is to train a model to infer underlying instance-specific geometry without explicit pixel-level labels. In this section, we look at various experiments to individually validate the effectiveness of our proposed modules. We measure the performance of our models using ground-truth annotations present in our face dataset. We objectively measure model performance of the predicted instance-specific mesh and surface mapping.

4.1. Datasets

Our framework attempts learning instance-specific geometry by exploiting multi-view consistency. To perform evaluation in a fair manner, we propose a multi-view dataset of RGB images, segmentation masks and their corresponding camera poses. Our dataset contains instance from three categories: faces, cars and airplanes.

Faces: For faces, an existing dataset 300WLP [43] contains RGB images, 3D facial mesh and 3D morphable model (3DMM) parameters. For our work, we adapt the 300WLP [43] by frontalizing all the meshes and the corresponding position maps. We also generate ground truth depth and f_{uv} to help in evaluating supervised baselines.

Cars and Airplanes: Our dataset consists of manually selected 500 high-quality car and airplane meshes. For each instance, we generate 100 view-points per instance in a 360° smooth camera trajectory around the mesh (refer Fig. 3). We use Blinn-Phong shading model for rendering in OpenGL, along with 8 point lights and one single directional light attached to the virtual camera looking direction.

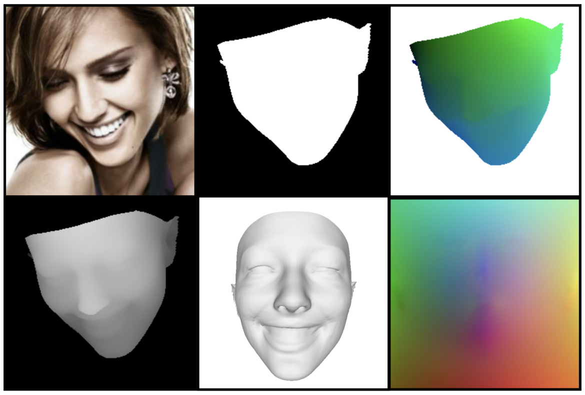

Figure 4: Figure illustrating data being released as part of face dataset. The first row is the input 2D image, segmentation mask and the corresponding f_{uv} map. The second row is the depth map, frontalized mesh and the corresponding frontalized position map D. Beside this, we also release the transformation matrix for projecting mesh into image space.

We plan to release all of our datasets containing multi-view images, segmentation masks, depth maps, mesh, and camera poses. For faces, we also provide UV maps and position maps in frontalized coordinate system. We believe that apart from solving category-specific reconstruction, our dataset would be useful in propelling research in multi-view supervised as well as weakly-supervised tasks such as image segmentation, depth prediction and mesh estimation. Because our camera trajectory is smooth, it also has applications in turntable and handheld multi-view captures.

4.2. Implementation Details

We implement our network in PyTorch [24] and its architecture is based on DeepLabV3+ [5]. UV and position-map prediction have separate decoders. All our training and testing experiments are performed on an NVIDIA GeForce GTX 1080 Ti GPU with 8 cores each running @ 3.3 GHz.

4.3. Evaluation Metrics

We evaluate our approach by computing the Percentage of Correct Keypoints (PCK). We focus our quantitative evaluations and loss ablations on face dataset, because this is the only dataset with dense UV annotations and ground-truth position maps.

4.4. Quantitative Results

We analyze various aspects of our approach through ablation studies, experiments on the multi-view datasets, and controlled variation of settings to understand the individual effectiveness of the proposed modules. In the following sections, we aim to understand: 1) the effectiveness of reprojection loss and its utility as a self-supervised signal; 2) the effectiveness of our deformation module and impact on performance; 3) the effectiveness of multi-view training compared to the single-view model; 4) the effectiveness of our overall model.

4.4.1 Effectiveness of Reprojection

We start off by considering scenarios where we initially utilize ground truth annotations to learn each component. Specifically, *'Learning only UVs'* refers to learning UV mapping while using ground truth meshes for each instance; *'Learning only PosMaps'* refers to learning meshes while using ground truth UV mapping for each instance. We then move on to the weakly supervised setting where we do not have pixel-level labels. *'Learning UVs with fixed mesh'* involves learning the UV mapping with an average mesh instead of an instance-specific ground truth mesh. Finally, we use supervision with pixel-level annotations to get an upper bound for performance. *'Learning with dense labels'* involves learning the UV mapping and PosMap using direct supervision from the labels.

To gain a holistic understanding of model performance, we consider evaluations on both UV and PosMap. We perform evaluation on multiple thresholds to gain both fine- and coarse-grained understanding. Table 1 and Table 2 contain UV and PosMap evaluations respectively and summarize our results when comparing training with only reprojection to other approaches.

Approach	UV-Pck@			
	0.01	0.03	0.1	AUC
Learning UVs with fixed mesh	5.3	32.2	90.6	94.0
Learning only UVs	**12.1**	**48.6**	**91.1**	**94.8**
Learning with dense labels	55.1	94.9	99.5	98.7

Table 1: Comparison of UV performance. We notice sharp degradation in performance while looking at smaller Pck thresholds when only using reprojection. The biggest performance gap between supervised and weakly-supervised models emerge at finer scales, suggesting that reprojection is a good signal to give rough predictions but not enough for finer-grained ones.

Table 1 shows the effectiveness of reprojection as a supervisory signal even in the absence of dense labels. Our approach is comparable to the supervised baseline at coarse α's despite not having any dense label supervision at all.

Table 2 shows the effectiveness of reprojection in learning the underlying 3D structure without having the underlying geometry during training. We observe higher Pck-PosMap values when using ground truth UVs, as the network optimizes for the ideal mesh based on the provided UV mapping, leading to a slight boost in performance compared to the weakly-supervised variant.

Approach	PosMap-Pck@		
	0.01	0.03	0.1
Learning UVs with fixed mesh	56.3	71.7	98.6
Learning only PosMaps	**56.9**	**72.2**	**99.5**
Learning with dense labels	59.0	82.0	99.8

Table 2: Comparison of PosMap performance. We are able to approach coarse-level supervised performance (at $\alpha = 0.1$) with reprojection while lagging at finer scales.

4.4.2 Effectiveness of Deformation

In the previous section, we observed an improvement in UV performance with accurate underlying meshes. Now we investigate the effectiveness of learning deformations for better position maps along with their effect on UV performance.

With Pixel-Level Supervision: We first evaluate the effectiveness of our deformation module by studying its impact on performance in a supervised setting. We consider two variants, 1) *Unconstrained*: our position map prediction head directly predicts a $256 \times 256 \times 3$ position map with around $43k$ valid points; 2) *Deformed Mesh*: we predict a $256 \times 256 \times 3$ 'residual' position map and combine it with the mean mesh.

We summarize our results in Table 5. We see improved performance when learning deformations instead of an unconstrained position map. Overall, we observe that 1) our modules lead to improved performance, especially at finer scales; 2) using such deformations allow us to converge much more quickly compared to the unconstrained counterpart. We argue this is due to the intuitive nature of the formulation as well as the ease of predicting residuals over inferring an unconstrained position map.

Without Pixel-Level Supervision: After seeing the efficiency of residual deformation learning, we proceed to study its effectiveness in the absence of pixel-level labels. For these experiments, we perform only single-view training and only utilize the components proposed in Sections 3.2 and 3.3. We evaluate the effectiveness of both the proposed deformation residual formulations. *'Reprojection with Deformed Mesh'* utilizes direct prediction of position map residuals. Both approaches are discussed in Section 3.3. We use *'Reprojection with Fixed Mesh'* as a baseline. Table 3 summarizes the results and shows the benefit of utilizing deformations over a fixed mesh. We observe considerable performance improvement, especially for UV predictions.

4.4.3 Effectiveness of Multi-View Training

So far, we have demonstrated the effectiveness of reprojection and mesh deformation for a single-view setting. We

Approach	UV-Pck@				PosMap-Pck@		
	0.01	0.03	0.1	AUC	0.01	0.03	0.1
Reprojection with Fixed Mesh	5.3	32.2	90.6	94.0	**35.8**	58.8	**98.1**
Reprojection with Deformed Mesh	**13.5**	**57.8**	**96.0**	**95.7**	**35.8**	**59.3**	97.9

Table 3: Performance of deformed models. We notice a considerable increase in UV performance when allowing deformed meshes.

Approach	UV-Pck@				PosMap-Pck@		
	0.01	0.03	0.1	AUC	0.01	0.03	0.1
Single-view Reprojection with Fixed Mesh	5.3	32.2	90.6	94.0	56.3	71.7	98.6
Multi-view Reprojection with Fixed Mesh	**5.8**	**34.0**	**90.8**	**94.2**	56.3	71.7	98.6
Deformed Single-view Reprojection	**13.5**	57.8	96.0	95.7	56.4	72.4	98.5
Deformed Multi-view Reprojection	13.4	**58.4**	**96.2**	**95.9**	**56.5**	**72.7**	**98.6**

Table 4: Comparison of Single-View and Multi-View Training. We observe consistently improved UV performance with multi-view training with both fixed and deformed meshes. We also notice a slight improvement in position map performance.

Approach	PosMap-Pck@		
	AUC	0.1	0.03
Mean-Fixed Mesh	96.2	97.6	42.7
Unconstrained	97.6	99.8	74.8
Deformed Mesh	**98.0**	**99.9**	**82.0**

Table 5: Performance comparison between different approaches for 3D mesh prediction. We notice that residual mesh learning approaches perform much better than the unconstrained prediction.

Approach	UV-Pck@		PosMap-Pck@	
	0.03	0.1	0.03	0.1
CSM*	32.2	90.6	71.7	**98.6**
Our Approach	**58.4**	**96.2**	**72.7**	**98.6**
Fully Supervised	94.9	99.5	82.0	99.8

Table 6: Comparison of Single-View and Multi-View Training

now compare the single-view training with the multi-view training setting. We consider performance with both a fixed and deformed mesh. We first consider the fixed mesh setting, where *'Single-view Reprojection with Fixed Mesh'* and *'Multi-view Reprojection with Fixed Mesh'* are the single and multi-view training settings. We then consider our overall model with deformations on top, where *'Deformed Single-view Reprojection'* and *'Deformed Multi-view Reprojection'* refer to the single and multi-view settings for training with deformed meshes.

Table 4 summarizes our results and demonstrates consistent performance gains with the usage of multi-view training, proving the effectiveness of our approach. We see relatively lower gains in our position map performance as we directly optimize for consistency of our UV predictions.

4.4.4 Comparison with Existing Approaches

We summarize comparisons of our overall approach to the only directly comparable state-of-the-art approach, i.e., CSM [19] in Table 6. CSM is functionally the same as *'Single-view Reprojection with Fixed Mesh'* described in the previous section. Note that performance metrics for CSM on our dataset has been computed using our re-implementation of CSM. We have closely followed the implementation of CSM except for performing optimization in 2D space as opposed to 3D space. We also consider the fully-supervised baseline, i.e., *'Learning with Dense Labels'*, to give an estimate of how our approach compares to it.

4.5. Qualitative Results

Fig. 5 shows a few instances of our predicted surface mappings of a model trained with a reprojection cycle. The model predicts smooth dense correspondences between different instances.

Fig. 6 shows a few examples of instance-specific deformations. We can see that both the supervised and weakly-supervised networks learn to predict open/closed mouth, elongated/rounded, bigger/smaller face shapes. The weakly supervised model shown is trained with multi-view reprojection consistency cycle. We did not add any symmetry constraints. We can see that the supervised approach learns implicit symmetry, whereas the weakly-supervised one focuses only on the visible parts of the face (note that even a sun hat covered portion is ignored). This is expected as the supervised network could see the full position map during training, while the reprojection-cycle-trained model had error signal only for the foreground (because of the visibility loss described earlier). While the predictions make intuitive sense, we also notice a significant increase in PCK UV as shown in Table 5.

4.6. Learning on new datasets

In order to train on a new dataset, we require segmentation masks, camera poses and, optionally, multiple views of different instances. Camera poses can be inferred in case we have a few keypoint annotations provided by methods like PnP (Perspective-n-Point). Segmentation masks for a given

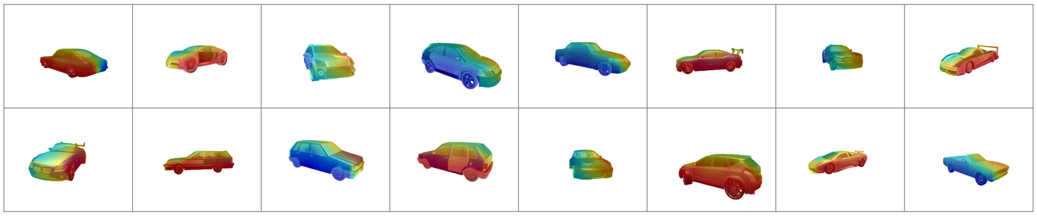

Figure 5: Figure illustrating images and the learnt UV mappings overlaid on top. Predictions shown are from the model trained with the reprojection cycle.

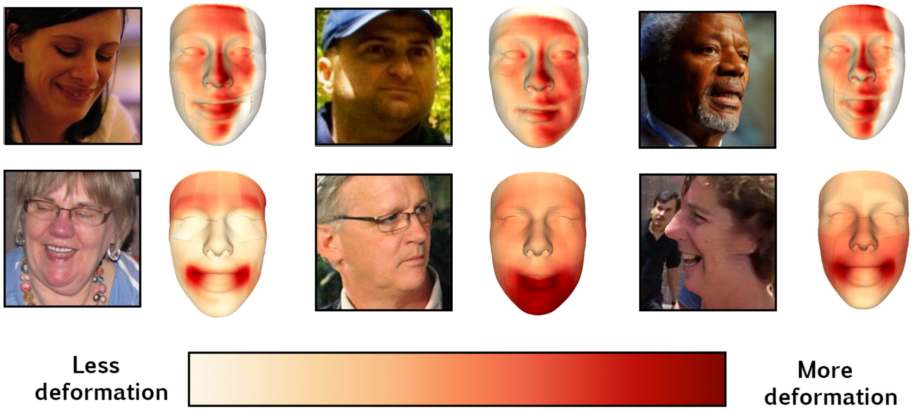

Figure 6: Figure illustrating images and the corresponding predicted meshes color-coded by deformation magnitude with respect to the mean mesh. Top row shows weakly-supervised model predictions, bottom row shows supervised predictions. We can see that both the supervised and weakly-supervised networks learn to predict open/closed mouth and elongated/rounded face shapes.

category can be inferred via an off-the-shelf model [11]. Alternatively, for any new multi-view dataset or other categories, we can run SfM [28] on the images to compute poses and a point cloud. These point clouds can then be aligned to each other to ensure that all the point clouds are in a single common coordinate system. Finally, we can scale these point clouds to ensure that they are in a unit cube.

5. Conclusion

We propose a framework to learn surface mapping and category-specific geometric reconstruction in a weakly supervised setting. By modelling the underlying instance-specific deformations along with utilizing multi-view cues, we allow our model to learn consistent UV mappings without explicit annotations for the same. We demonstrate the effectiveness of each of the proposed modules through controlled experiments and see a significant improvement in performance. We believe that our approach is the first to exploit multi-view cycle consistencies to generate instance-specific meshes by modelling deformations. We term our approach geometric reconstruction, as our meshes can also be surface-mapped back onto the images. We also present and release a new multi-view dataset of ShapeNet Cars and Airplanes generated by rendering filtered ShapeNet meshes with a smooth camera trajectory and an adapted 300WLP dataset with frontalized face meshes and position maps. We hope these contributions will spark interest in multi-view approaches to learn geometry without the need for labels.

6. Acknowledgement

We would like to thank our co-workers at Fyusion for their guidance and assistance, and anonymous reviewers for helpful discussions and feedback. We also thank Abhishek Kar for his valuable help in getting us started with the idea and the fruitful discussions without which this work would not have been possible.

References

[1] Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: Reconstruction and parameterization from range scans. *ACM Trans. Graph.*, 22(3):587–594, July 2003.

[2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: Shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, July 2005.

[3] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, page 187–194, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[4] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous

separable convolution for semantic image segmentation. In *ECCV*, 2018.

[6] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaako Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *NeurIPS*, 2019.

[7] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *ECCV*, 2018.

[8] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[9] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[10] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, USA, 2 edition, 2003.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[13] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018.

[14] Abhishek Kar, Shubham Tulsiani, João Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *CVPR*, pages 1966–1974. IEEE Computer Society, 2015.

[15] Abhishek Kar, Shubham Tulsiani, João Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *CVPR*. 2015.

[16] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, page 61–70, Goslar, DEU, 2006. Eurographics Association.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. 2012.

[18] Nilesh Kulkarni, Abhinav Gupta, David Fouhey, and Shubham Tulsiani. Articulation-aware canonical surface mapping. In *Computer Vision and Pattern Recognition (CVPR)*, 2020.

[19] Nilesh Kulkarni, Abhinav Gupta, and Shubham Tulsiani. Canonical surface mapping via geometric cycle consistency. 2019.

[20] Youn Lee, Sung Lee, Kang Park, Jaeik Jo, and Jaihie Kim. Single view-based 3d face reconstruction robust to self-occlusion. *EURASIP Journal on Advances in Signal Processing*, 2012, 08 2012.

[21] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019.

[22] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: A skinned multi-person linear model. *ACM Trans. Graph.*, 34(6), Oct. 2015.

[23] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020.

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alche-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[25] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[26] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5099–5108, 2017.

[27] S. Romdhani and T. Vetter. Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 986–993 vol. 2, 2005.

[28] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[29] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, 2019.

[30] Srinath Sridhar, Davis Rempe, Julien Valentin, Sofien Bouaziz, and Leonidas J. Guibas. Multiview aggregation for learning category-specific shape reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[31] A. Tewari, M. Zollhöfer, P. Garrido, F. Bernard, H. Kim, P. Perez, and C. Theobalt. Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2549–2559, 2018.

[32] Luan Tran, Feng Liu, and Xiaoming Liu. Towards high-fidelity nonlinear 3d face morphable model. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1126–1135, 2019.

[33] Luan Tran, Feng Liu, and Xiaoming Liu. Towards high-fidelity nonlinear 3d face morphable model. In *In Proceeding of IEEE Computer Vision and Pattern Recognition*, Long Beach, CA, June 2019.

[34] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*, 2018.

[35] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019.

[36] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *CVPR*, 2020.

[37] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, pages 75–82, 2014.

[38] Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *SIGGRAPH Asia*, 2016.

[39] Yumer and Mitra. Learning semantic deformation flows with 3d convolutional networks. In *European Conference on Computer Vision (ECCV 2016)*, pages –. Springer, 2016.

[40] C. Zach, M. Klopschitz, and M. Pollefeys. Disambiguating visual relations using loop constraints. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1426–1433, 2010.

[41] Tinghui Zhou, Philipp Krahenbuhl, Mathieu Aubry, Qixing Huang, and Alexei Efros. Learning dense correspondence via 3d-guided cycle consistency. pages 117–126, 06 2016.

[42] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[43] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z. Li. Face alignment across large poses: A 3d solution. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 146–155, 2016.