Hyperspectral Vision: A Lightweight 2D-to-3D Reconstruction and Skinning Pipeline (Research Track)

Aditya Aggarwal
IIIT Delhi
aditya22028@iiitd.ac.in

Arpan Verma
IIIT Delhi
arpan22105@iiitd.ac.in

Aditya Upadhyay
IIIT Delhi
aditya22040@iiitd.ac.in

Chandan Sah
IIIT Delhi
chandan22140@iiitd.ac.in

Abstract

This report presents a pipeline that transforms 2D images of object poses into accurate 3D meshes, followed by a hyperspectral inverse skinning module [3] for estimating skinning weights, transformations, and bone structures. For the 2D-to-3D reconstruction, we adopt a baseline inspired by Pixel2Mesh [1] and propose build upon a MobileNet architecture as detailed in Dominique Jack's work on free-form deformations [2], or similar. The report describes our problem statement, relevant literature, data and metrics, experimental analysis, and future research directions.

1. Introduction

Reconstructing accurate 3D shapes and underlying skeletons from a small set of 2D images (poses) is an important challenge in computer vision, graphics, and animation. Applications include character animation, motion capture (mocap), medical imaging, and AR/VR. Our pipeline (see Figure 1) combines:

1. **2D-to-3D Reconstruction:** Our baseline is now based on Pixel2Mesh [1] which deforms a template mesh from a single RGB image. We further propose to build upon this approach using a MobileNet backbone inspired by Dominique Jack's work on free-form deformations [2].
2. **Hyperspectral Inverse Skinning:** We use a hyperspectral inverse skinning method [3] to estimate per-vertex skinning weights and transformations from multiple mesh poses.
3. **Bone Generation Pipeline:** Derivation of bone structures from the estimated handle points.

2. Problem Statement and Scope

Problem Statement: Given a set of 2D images depicting various poses of an object or character, our goal is to:

- *Reconstruct* accurate 3D meshes for each pose using a learned deformation of a template.
- *Estimate* the per-vertex skinning weights and transformations that reproduce observed deformations.
- *Generate* a plausible bone (skeleton) structure to facilitate further animation.

Motivation and Relevance:

- *Applications:* Animation compression, real-time character control, AR/VR avatars.
- *Challenges:* Limited 2D views, unsupervised learning, and the need for high-fidelity 3D reconstructions.
- *Importance:* Minimizes manual rigging and enables accessible, high-quality 3D modeling from readily available 2D data.

3. Related Work and Baselines

3.1. 2D-to-3D Reconstruction Approaches

Several methods have addressed single-view or multi-view reconstruction:

- **Pixel2Mesh** [1]: Generates a 3D mesh by progressively deforming an initial ellipsoid using graph convolutional networks. This method serves as our primary baseline for 2D-to-3D reconstruction.

- **Learning Free-Form Deformations for 3D Object Reconstruction** [2]: This work builds upon a MobileNet backbone and free-form deformation (FFD) technique to deform a template mesh based on a single image, inspiring our design.
- **Occupancy Networks** [4] and **AtlasNet** [5] provide alternative implicit representations, though they output voxels or point clouds.

3.2. Inverse Skinning Methods

- **Hyperspectral Inverse Skinning** [3]: Reformulates inverse skinning as a minimum-volume simplex problem, extracting skinning weights and transformations from a set of deformed meshes.

3.3. Research Gaps

1. Integration of 2D-to-3D reconstruction with inverse skinning in a unified, minimally supervised pipeline.
2. Leveraging a lightweight MobileNet backbone (as used by Dominique Jack) combined with advanced deformation strategies.
3. Enhancing reconstruction fidelity while maintaining efficiency suitable for mobile and embedded platforms.

4. Data and Evaluation Metrics

4.1. Datasets

- **Synthetic Data:** Rendered images from ShapeNet objects.
- **Ground Truth Meshes:** Provided by previous works such as Pixel2Mesh and hyperspectral inverse skinning datasets.

4.2. Evaluation Metrics

- **2D-to-3D Reconstruction:**
  1. *Chamfer Distance (CD)* between predicted and ground-truth point clouds.
  2. *F-Score* computed on the mesh surfaces.
- **Pose Reconstruction Error:** Distance between predicted and ground-truth vertex positions.
- **Bone Structure Consistency:** Qualitative comparison.

5. Proposed Architecture for 2D-to-3D Reconstruction

Our reconstruction architecture builds upon the Pixel2Mesh model as employed by nywang16's work. A brief outline is as follows:

5.1. Overall Pipeline Overview

1. **Data Prep & Initialization**
   - Unpack ShapeNetP2M and generate per-view `.dat` files (point + normal + image) under `Data/.../rendering/00.dat{23.dat`.
   - Load a fixed coarse ellipsoid mesh ($N_0 \approx 256$ vertices) as the starting template.
2. **Image Feature Extraction**
   - Pass the input RGB image through a small CNN (e.g., 18-layer VGG) to produce four multi-scale feature maps: $56 \times 56 \times 64$, $28 \times 28 \times 128$, $14 \times 14 \times 256$, $7 \times 7 \times 512$.
3. **Multi-Stage Deformation (3 refinement blocks)**

   For $t = 1 \to 3$:

   [label*=0.]**Dynamic Graph Construction**

   (a) • Build a k-NN graph in 3D (or learned feature space) on the current vertex set V^t.

   (b) **Attention-Augmented Graph Convolution**
   - For each vertex v_i, compute pairwise attentions α_{ij} over its neighbors via a learned "query-key" MLP on $(v_i \parallel v_j)$.
   - Aggregate neighbor features with the α_{ij} weights.
   - Pass through a small MLP to produce an updated vertex feature h_i^t.

   (c) **Offset Regression & Update**
   - From h_i^t predict a 3D offset Δv_i^t; update vertex positions: $v_i \leftarrow v_i + \Delta v_i^t$.

   (d) **Unpool (if $t < 3$)**
   - Subdivide each edge, insert midpoints $\to$ double the vertex count. Transfer features via simple average pooling.
4. **Losses & Regularizers**
   - Chamfer distance to ground-truth point cloud
   - Laplacian smoothness (with masked neighbors)
   - Edge-length and normal-cosine losses

5.2. Detailed Architectural Steps

1. **CNN Backbone + Projection**
   - Input image $224 \times 224 \times 3 \to$ conv blocks $\to$ feature maps $\{F_1, \ldots, F_4\}$.
   - For each vertex v_i, project $(x, y, z) \to$ pixel (u, v) in image via camera intrinsics, sample corresponding vectors from each $F_k \to$ per-vertex feature f_i.

2. **For** $t = 1 \dots 3$ **(refinement stage)**

   [label*=0.]**Rebuild adjacency** A^t

   (a) • Compute k-nearest neighbors in Euclidean 3D space ($k \approx 8$) on V^t or use a small "affinity" MLP to predict edge weights and select top-k.

   (b) **Attention GraphConv block (×2 layers per stage)**

   • $e_{ij} = \text{LeakyReLU}(a^\top [W_q v_i \parallel W_k v_j])$

   • $\alpha_{ij} = \text{softmax}_{(j)}(e_{ij})$

   • $h_i = \text{ReLU}\left(\sum_j \alpha_{ij} W_v v_j + b\right)$

   (c) **Offset MLP**

   • $\Delta v_i = \text{MLP}(h_i) \rightarrow$ new coordinate

   (d) **Unpool if** $t < 3$

   • For each edge (i, j), insert vertex at $(v_i + v_j)/2$; pool features by averaging.

5.3. Rationale and Benefits

- **Attention-Augmented GCN**
  - Learns data-driven neighbor weights instead of fixed Chebyshev supports $\rightarrow$ sharper detail preservation, robust to irregular meshes.
  - Attentions filter out noisy or overly distant neighbors, stabilizing training.
- **Dynamic Graph Topology**
  - Recomputing k-NN on the deformed mesh ensures local connectivity reflects the new shape $\rightarrow$ prevents feature "bleeding" across distant regions.
  - Learned affinities can discover semantic or geometric adjacency beyond pure spatial proximity.
- **Combined Impact**
  - More expressive vertex interactions $\rightarrow$ faster convergence, crisper edges, and improved generalization on unseen categories compared to static-graph GCN.

6. Bone Formation and Pruning

In this we use the calculation from the paper [**?**]. Given an un-rigged mesh sequence $\{\mathbf{v}_{f,i} \in \mathbb{R}^3\}_{i=1}^{N_v}$, $f = 1, \dots, N_f$, our goal is to discover a compact skeleton $\mathcal{S} = (\mathcal{J}, \mathcal{B})$ (joint set $\mathcal{J}$, bone set $\mathcal{B}$) together with *linear blend skinning* (LBS) parameters that minimise the reconstruction error

$$E_{\text{LBS}} \;=\; \sum_{f=1}^{N_f} \sum_{i=1}^{N_v} \Big\| \mathbf{v}_{f,i} - \sum_{b \in \mathcal{B}} w_{i,b} \left(\mathbf{R}_{f,b}\, \mathbf{v}_{0,i} + \mathbf{t}_{f,b} \right) \Big\|_2^2, \tag{1}$$

where $\mathbf{v}_{0,i}$ is vertex i in the rest frame, $(\mathbf{R}_{f,b}, \mathbf{t}_{f,b}) \in SE(3)$ is the rigid transform of bone b at frame f, and $\{w_{i,b}\}$ are non-negative skinning weights with $\sum_b w_{i,b} = 1$.

6.1. Initial Bone Generation

Transformation residual clustering. We first compute per-vertex rigid motions

$$(\widehat{\mathbf{R}}_{f,i}, \widehat{\mathbf{t}}_{f,i}) \;=\; \underset{(\mathbf{R},\mathbf{t}) \in SE(3)}{\arg\min} \sum_f \big\| \mathbf{v}_{f,i} - \mathbf{R}\mathbf{v}_{0,i} - \mathbf{t} \big\|_2^2. \tag{2}$$

Collecting the residuals $\mathbf{e}_{f,i} = \mathbf{v}_{f,i} - \widehat{\mathbf{R}}_{f,i}\mathbf{v}_{0,i} - \widehat{\mathbf{t}}_{f,i}$ into a feature vector $\mathbf{z}_i = [\mathbf{e}_{1,i}^\top, \dots, \mathbf{e}_{N_f,i}^\top]^\top$, we apply k-means++ to cluster the vertices. Each cluster induces an *initial bone*. The rest-pose joint position is set to the barycentre of its vertices:

$$\mathbf{c}_b \;=\; \frac{\sum_{i \in \mathcal{V}_b} \mathbf{v}_{0,i}}{|\mathcal{V}_b|}.$$

Greedy refinement. We iteratively split the bone with the largest reconstruction error (Eq. 1), until either (i) the error drops below a user threshold $\varepsilon_{\text{split}}$, or (ii) a maximum bone limit $B_{\max}$ is reached. A split plane orthogonal to the principal eigenvector of the cluster's covariance yields two child bones.

6.2. Joint Placement

For two neighbouring bones b_1, b_2 the joint is initialised at the weighted least-squares intersection of their motion lines [**?**]:

$$\mathbf{j}_{b_1,b_2} \;=\; \Big(\sum_f \mathbf{A}_f^\top \mathbf{A}_f \Big)^{-1} \Big(\sum_f \mathbf{A}_f^\top \left(\mathbf{b}_f \right) \Big), \quad \mathbf{A}_f = \big[\mathbf{I} - \mathbf{R}_{f,b_1} \,|\, \mathbf{R}_{f,b_2} - \mathbf{I} \big], \tag{3}$$

where $\mathbf{b}_f = \mathbf{t}_{f,b_1} - \mathbf{t}_{f,b_2}$.

6.3. Bone Pruning

Weight-mass pruning. A bone is discarded if its accumulated influence is negligible:

$$\sum_{i=1}^{N_v} w_{i,b} \;<\; \eta_{\text{mass}} \quad (\text{default } \eta_{\text{mass}} = 0.01).$$

Linear-chain collapsing. For any triplet of consecutive joints (j_1, j_2, j_3), if the enclosed angle $\angle(j_1, j_2, j_3) < \theta_{\min}$ (e.g. 5°) for *all* frames, bone $\overline{j_1 j_2}$ is merged into $\overline{j_1 j_3}$ and the weights of its vertices are re-normalised.

Pose variance pruning. The positional variance of joint b across time

$$\sigma_b^2 \;=\; \frac{1}{N_f} \sum_f \big\| \mathbf{j}_b^{(f)} - \bar{\mathbf{j}}_b \big\|_2^2, \qquad \bar{\mathbf{j}}_b = \frac{1}{N_f} \sum_f \mathbf{j}_b^{(f)},$$

is compared against a threshold σ^2_{min}. Bones whose motion is below that threshold are removed as they contribute little to animation fidelity.

6.4. Final Optimisation

After pruning, we re-optimise the remaining $\{\mathbf{R}_{f,b}, \mathbf{t}_{f,b}, w_{i,b}\}$ by alternating least-squares as in Algorithm 2 of [**?**] until the RMSE improvement $\Delta E_{\text{LBS}}/E_{\text{LBS}} < 10^{-4}$.

7. Analysis of Results

Preliminary experiments indicate that our Pipeline is building up and coming into reality. In our analysis, we note:

- The Pixel2Mesh baseline [1] provides a strong foundation for mesh deformation and has nice results.
- While Dominique Jack's free-form deformation approach [2] inspires our proposed lightweight MobileNet-based feature extraction and deformation parameter inference.
- The Hyperspectral Inverse Skinning provides very similar results to input Meshes.

8. Compute Requirements

- **Hardware:**
  1. Multi-core CPU for data preprocessing and Hyperspectral Inverse Skinning.
  2. GPU (e.g., from Kaggle) for training the 2D-to-3D reconstruction pipeline.

9. Individual Tasks and Contributions

- **Aditya Aggarwal:** Bone construction Pipeline and Implementation of hyperspectral inverse skinning module [3].
- **Aditya Upadhyay:** Implementation of 2D to 3D reconstruction Baseline Pixel2Mesh [1].
- **Arpan Verma:** Implementation of the hyperspectral inverse skinning module [3]. Proposed Architecture (Mobile Net + Transformer).
- **Chandan Sah:** Implementation of 2D to 3D reconstruction Baseline Pixel2Mesh [1].

10. Next Steps

1. **2D to 3D Reconstruction Challenges:** Despite architectural proposals like MobileNet fused with Transformer-style self-attention, the 2D to 3D reconstruction task remains unsolved. While efforts can be made to improve results, there is currently no guarantee of consistently accurate output.
2. **Rigging Complex Shapes:** Rigging still faces issues when dealing with complex or highly articulated shapes. Strategies to simplify or generalize handle generation must be explored further.
3. **Pipeline Simplification:** Our current approach remains a two-stage process—reconstruction followed by inverse skinning. The long-term goal is to shift towards a single-stage pipeline.
4. **Exploration of Unified Pipelines:** The work presented in DrawingSpinUp provides a foundation for a single-stage pipeline, albeit limited to simple sketching tasks. A future direction is to extend and generalize this approach for more complex 3D reconstruction workflows. [6]
5. **Adversarial Extensions (Future):** This whole Pipeline is essentially still a 2 step process, Reconstruction and Inverse Skinning. Ideally one Adversarially or any other learned Architecture for the entire process should be sought after, though beyond the immediate project timeline.

This builds upon the MobileNet architecture as introduced in Dominique Jack's work and leverages the Pixel2Mesh baseline for 2D-to-3D reconstruction.

Conclusion: This report outlines a research-track project combining 2D-to-3D reconstruction and hyperspectral inverse skinning. By building upon established methods such as Pixel2Mesh [1] and Dominique Jack's free-form deformation approach [2], we aim to achieve a lightweight yet high-fidelity, unsupervised pipeline from 2D images to fully rigged and skinned 3D models.

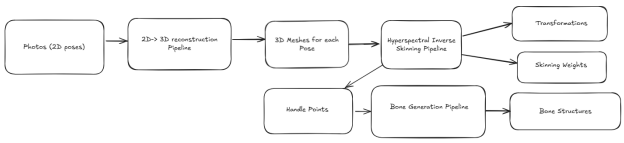

Figure 1. Overall project pipeline. 2D images are transformed into 3D meshes via the reconstruction network (baseline: Pixel2Mesh), then fed into the hyperspectral inverse skinning pipeline for skinning weights and transformations, followed by bone structure generation.

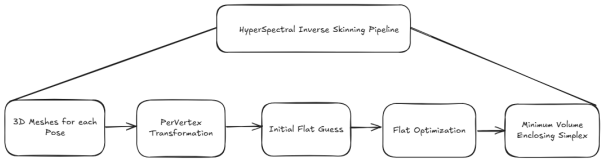

Figure 2. HyperSpectral Inverse Skinning Pipeline- Consists of 5 stages: 1: Loading of 3D mesh data for all the poses of an object/character. 2: Applying Per Vertex Transformation and mapping it to R^{12p}. 3: Initial Flat Guess by finding the h-1 dimensional line that is closest to all the intersectional points of the Per vertex Transforms. 4: Flat Optimization. 5: Minimum Volume Enclosing Simplex provides the vertices which are the Transformation matrices and the Barycentric Coordinates as the Skinning Weights.

Figure 3. Twisted Rod: pose-1 Left is the reconstructed in red which is the output of the pipeline, right one is the input green GT. Similar in all rest images.

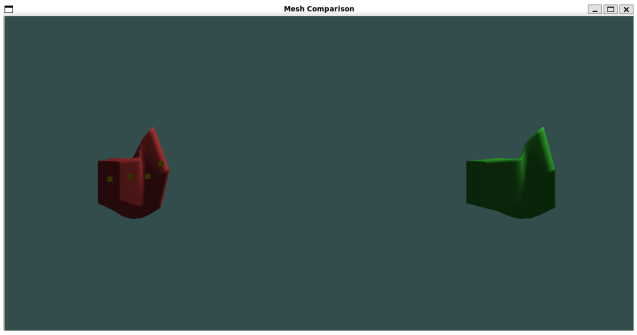

Figure 4. Twisted Rod: pose-2

Category	CD
plane	0.477
bench	0.624
table	0.498

Table 1. CD on the ShapeNet test set for plane, bench, and table for the Baseline Pixel2Mesh [1]. Smaller is better.

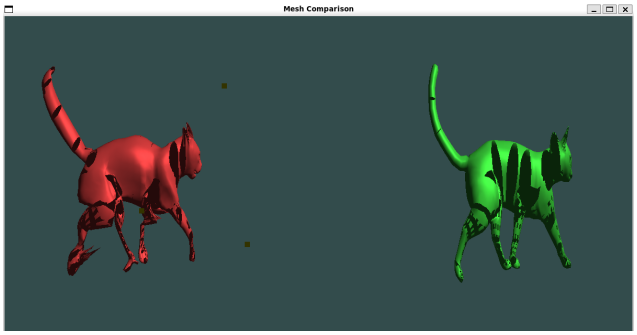

Figure 5. Cat: pose-1

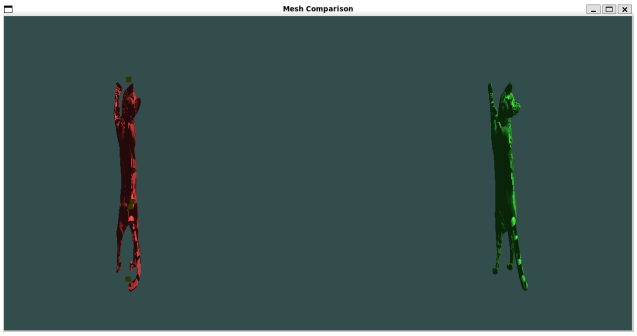

Figure 6. Cat: pose-2

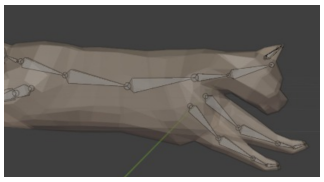

Figure 7. Rigged cat side view

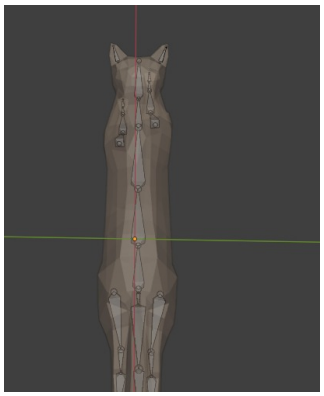

Figure 8. Rigged cat top view

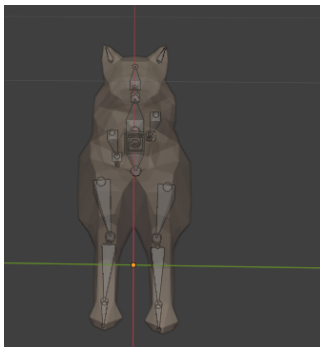

Figure 9. Rigged cat back view

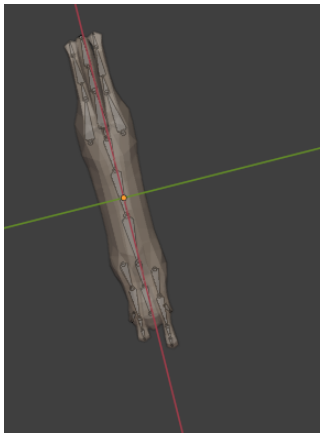

Figure 10. Rigged cat bottom view

Category	CD
plane	0.477
bench	0.624
table	0.498

Table 2. CD on the ShapeNet test set for plane, bench, and table for the Baseline Pixel2Mesh [1]. Smaller is better.

Category	CD
plane	0.431
bench	0.519
table	0.437

Table 3. CD on the ShapeNet test set for plane, bench, and table for our 2d-3d GCN+Attention model

Category	τ	2τ
plane	71.12	81.38
bench	57.57	71.86
table	66.30	79.20

Table 4. F-score (%) on the ShapeNet test set for plane, bench, and table under thresholds $\tau = 10^{-4}$ and 2τ, for the Baseline Pixel2Mesh [1]. Larger is better.

Pose	Average Error	Max Error
0	0.000825	0.0035373072
1	0.000673	0.0035463072
2	0.000863	0.0038273072
3	0.000810	0.0026063072

Table 5. Average and Maximum Error for the Twisting Rod across four poses.

Pose	Avg Error	Max Error
0	0.028669	0.132117
1	0.030794	0.233011
2	0.040084	0.173699
3	0.034170	0.163282

Table 6. Average and Maximum Error for Cat Model Across Poses

References

[1] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images. *ECCV*, 2018. :contentReference[oaicite:0]index=08203;:contentReference[oaicite:1]index=1 1, 4, 5, 6

[2] Dominic Jack, Jhony K. Pontes, Sridha Sridharan, Clinton Fookes, Sareh Shirazi, Frederic Maire, and Anders Eriksson. Learning Free-Form Deformations for 3D Object Reconstruction. *ACCV*, 2018. :contentReference[oaicite:2]index=28203;:contentReference[oaicite:3]index=3 1, 2, 4

[3] Mengfei Liu *et al.* Hyperspectral Inverse Skinning for Unsupervised Animation Extraction. *ACM Trans. Graph.*, 2020. 1, 2, 4

[4] Lars Mescheder *et al.* Occupancy Networks: Learning 3D Reconstruction in Function Space. *CVPR*, 2019. 2

[5] Thibault Groueix *et al.* AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. *CVPR*, 2018. 2

[6] Xinyu Liang, Yifan Jiang, Yujun Shen, Deli Zhao, Jingren Zhou, and Bolei Zhou. Sketch Your Own Pose: Pose-Guided Text-to-Image Generation using Pose-Contrastive Attention. *arXiv preprint arXiv:2209.11730*, 2022. Code 4