

SIRE: SE(3) Intrinsic Rigidity Embeddings

— Supplementary Material

Cameron Smith¹ Basile Van Hoorick² Chonghyuk Song³
 Vincent Sitzmann³ Vitor Guizilini^{2*} Yue Wang^{1*}

¹University of Southern California ²Toyota Research Institute

³Massachusetts Institute of Technology ^{*}*Equal Advising*

Reviewed on OpenReview: <https://openreview.net/forum?id=OZ9HOT0YMt>

1 SIRE Architecture

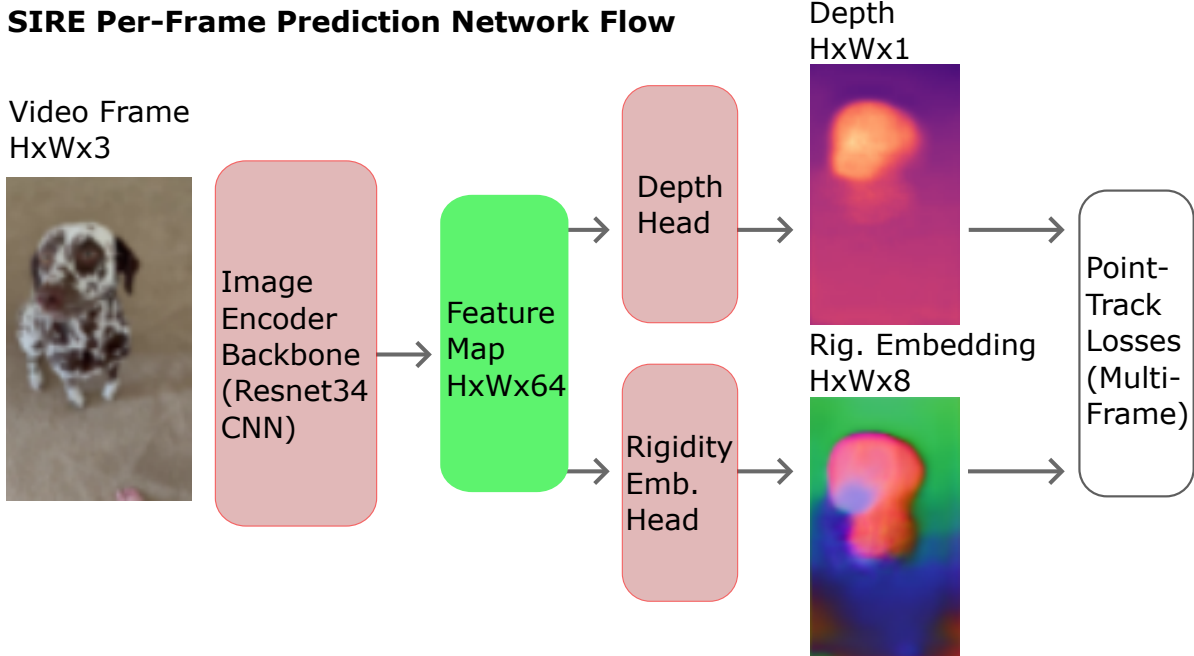


Figure 1: **SIRE Architecture Overview.** Given a video frame $I_t \in \mathbb{R}^{H \times W \times 3}$, a ResNet-34 image encoder produces a shared feature map $F_t \in \mathbb{R}^{H \times W \times 64}$. Two lightweight convolutional heads decode F_t into depth $D_t \in \mathbb{R}^{H \times W \times 1}$ and rigidity embeddings $R_t \in \mathbb{R}^{H \times W \times d_r}$ (here $d_r=8$). These image-aligned predictions are then used by the multi-frame point-track loss.

Given a video, we process each frame independently with a shared image encoder (run individually per frame) to produce a feature map. Two lightweight convolutional heads decode the feature map into (i) a depth map and (ii) a rigidity-embedding map. See Fig. 1 for an overview. These two image-aligned predictions are then used in the multi-frame point-track loss: we use depth to lift the sparse 2D point tracks into 3D, solve for a per-track motion $\in SE(3)$ by a rigidity-weighted fit where weights are derived from rigidity embedding similarity, and finally reproject and compare the 3D tracks to their 2D locations.

The image encoder is a pre-trained ResNet34, and the depth and rigidity embeddings are two randomly initialized convolutional layers (both just one layer with a kernel size of 3).

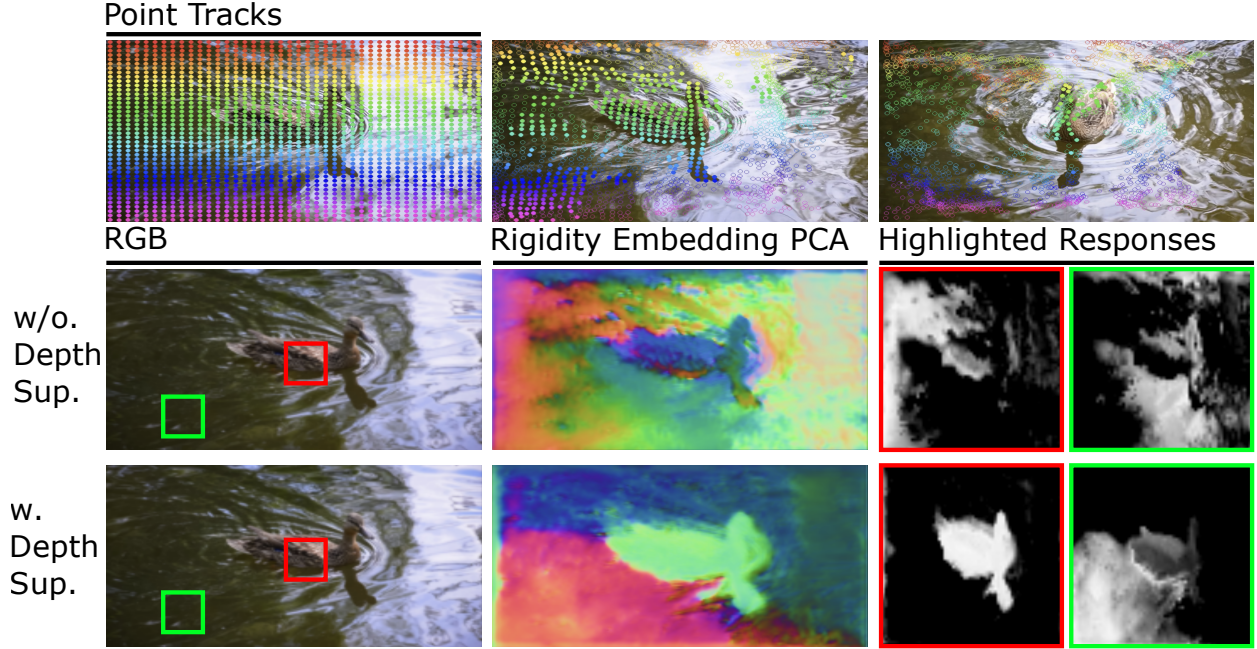


Figure 2: **Ablation: Bad Tracks & Depth Supervision.** We visualize a video with noisy point tracks over water (left), the converged rigidity embedding, and two highlighted rigidity map responses (right) showing per-pixel similarity to a seed in the red/green boxes. With *no depth supervision* (row 2), noisy and lost tracks on the water lead the model to group the duck and water together, yielding degenerate, non-intuitive rigidity clusters. Adding *depth supervision* (row 3) regularizes training: the duck is cleanly separated from the water, producing more geometrically meaningful, piecewise-rigid groupings despite poor track quality.

1.1 Rigidity Embedding Dimensionality

One hyperparameter of consideration is the rigidity embedding dimensionality: intuitively, a large dimensionality should be more flexible at the risk of the trivial solution where each point track only attends to itself, and dimensionality that is too low should be unable to fully explain the motion. We varied the rigidity embedding size $d_r \in \{3, 8, 64\}$ to study its effect on geometry and grouping. With a very small space ($d_r=3$), the model lacks the flexibility to well explain multi-body rigid motion and often collapses to the hollow depth bias (objects pushed toward infinity), and yielding less intuitive object groupings. At the other extreme ($d_r=64$), the embeddings are more flexible but depth often oscillates between hollow and plausible geometry. We found that our chosen dimensionality ($d_r=8$) strikes a balance, yielding more stable, piecewise-rigid groupings and plausible geometry. Quantitatively, the final training reprojection loss (lower is better) is interestingly 2.6 for $d_r=3$, 1.7 for $d_r=8$, and 2.2 for $d_r=64$, suggesting a sweet spot in dimensionality between low and high dimensionality. This makes sense intuitively as the motions we consider are not extreme, such as falling confetti or rain, and are closer to intuitively $\sim \leq 10$ rigid motions.

2 Results on Additional Multi-Object Datasets

Figure 3 shows SIRE on robot manipulation (internal data from Toyota Research) and highway driving scenes (from the PandaSet [Xiao et al. \(2021\)](#) driving dataset). In robot videos, our rigidity embeddings cleanly separate the two end-effectors from the target object across time while predicting reasonable unsupervised depth. In highway scenes, SIRE often groups individual cars, but without depth supervision we observe the standard hollow-depth bias of self-supervised driving data, and cars moving in parallel can be grouped together due to shared motion. Tracker noise in the sky further induces a sky/road split in the rigidity maps, highlighting both the robustness and the current limits of our track-based supervision.

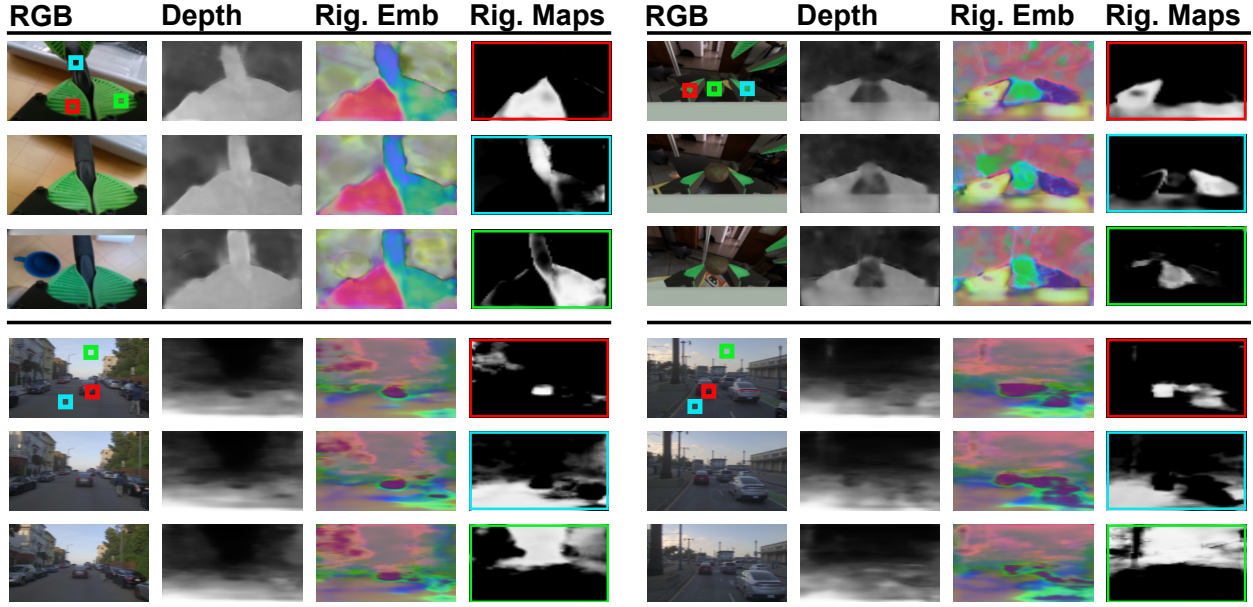


Figure 3: **Results on Additional Multi-Object Datasets** We run SIRE on two additional datasets: a *robot gripper* dataset (top two rows) and *highway cars* videos (bottom two rows). For each dataset we show three frames from two scenes with columns: RGB, predicted depth (unsupervised), rigidity embeddings, and *highlighted rigidity maps* (per-pixel similarity to seeds in red/green/blue boxes). **Robot Gripper Datasets:** SIRE often produces intuitive, piecewise-rigid groupings, segmenting the two grippers and the manipulated object as distinct components. **Cars:** The model sometimes captures intuitive groupings of vehicles, but without depth supervision exhibits the common *hollow depth* failure mode in driving videos (cars are placed towards large depth values). Parallel common motion between cars also causes occasional merging of nearby cars. Incorrect tracks in the sky also yield separate sky and road partitions.

3 Depth Supervision and Inaccurate Point Tracks

Fig. 2 evaluates SIRE on a sequence with noisy point tracks on moving water (the Mallard-Water DAVIS sequence). Without depth supervision, rigidity embeddings couple the duck with the surrounding water. With depth supervision, the embeddings emerge as more semantically and geometrically coherent, cleanly separating the duck from the water and yielding stable piecewise-rigid regions. This experiment suggests that depth acts as a strong geometric prior which mitigates the impact of poor tracks and prevents degenerate rigidity groupings.

4 Additional Rigidity Response Grids

In Fig. 4, we plot additional rigidity response grids for more DAVIS scenes. Recall that each result here is from a per-scene optimization, i.e. trained from-scratch on each video, highlighting our strong data-efficiency. For each video, we plot the first frame of the video and the 16x16 track response grid for space constraints, but note that in practice our track grids are 64x64 resolution. We observe compelling rigid body decompositions for each scene.

5 Self-Supervised Depth

In Fig. 6, we plot a random set of generalizable depth estimates on the CO3D-hydrants pre-training to demonstrate that we can learn compelling depth estimation on static scenes. This result makes sense since our method reduces to FlowMap Smith et al. (2024) in the static case and FlowMap demonstrated strong depth-learning.

In Fig. 7, we plot a random set of generalizable depth estimates on the CO3D-dogs dataset, as well as PCA visualization of rigidity embeddings. While depth is not perfect in this setting, we observe that most regions have plausible dog reconstructions. Also note that the feature embedding tends to empirically group canonical dog parts in consistent embedding groups – consider how the dogs are often colored in green in the first three embedding PCA channels and their dogs are colored red in the last three.

6 Epipolar-Rigidity Masking

In Fig. 5, we plot an example of the epipolar rigidity mask for an input video and below further describe its computation and usage. Given an input video, we run off-the-shelf optical flow Xu et al. (2022), and similar to prior works Ye et al. (2022), compute the fundamental matrix per-frame using a median RANSAC least-median solver. We then compute the Sampson re-projection error and consider all flow errors above the 80th quantile to be non-rigid. Note that this approach necessarily will segment rigid regions as non-rigid, but will be unlikely to consider non-rigid regions as rigid. We only use this mask during self-supervised depth training of CO3D-Dogs, as we find it better constrains the geometry. The way we use it is that we mark a point track as static if it is rigid in every frame, and for all static tracks, we set their rigidity weights to be constant so that they share the same rigid body. This has the effect of enforcing the static scene geometry to be explained by a single rigid motion (likely the camera motion).

References

- Cameron Smith, David Charatan, Ayush Tewari, and Vincent Sitzmann. Flowmap: High-quality camera poses, intrinsics, and depth via gradient descent. In *arXiv*, 2024. 3
- Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, Yunlong Wang, and Diange Yang. Pandaset: Advanced sensor suite dataset for autonomous driving, 2021. URL <https://arxiv.org/abs/2112.12610>. 2
- Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Reza Tofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8121–8130, 2022. 4
- Vickie Ye, Zhengqi Li, Richard Tucker, Angjoo Kanazawa, and Noah Snavely. Deformable sprites for unsupervised video decomposition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 4

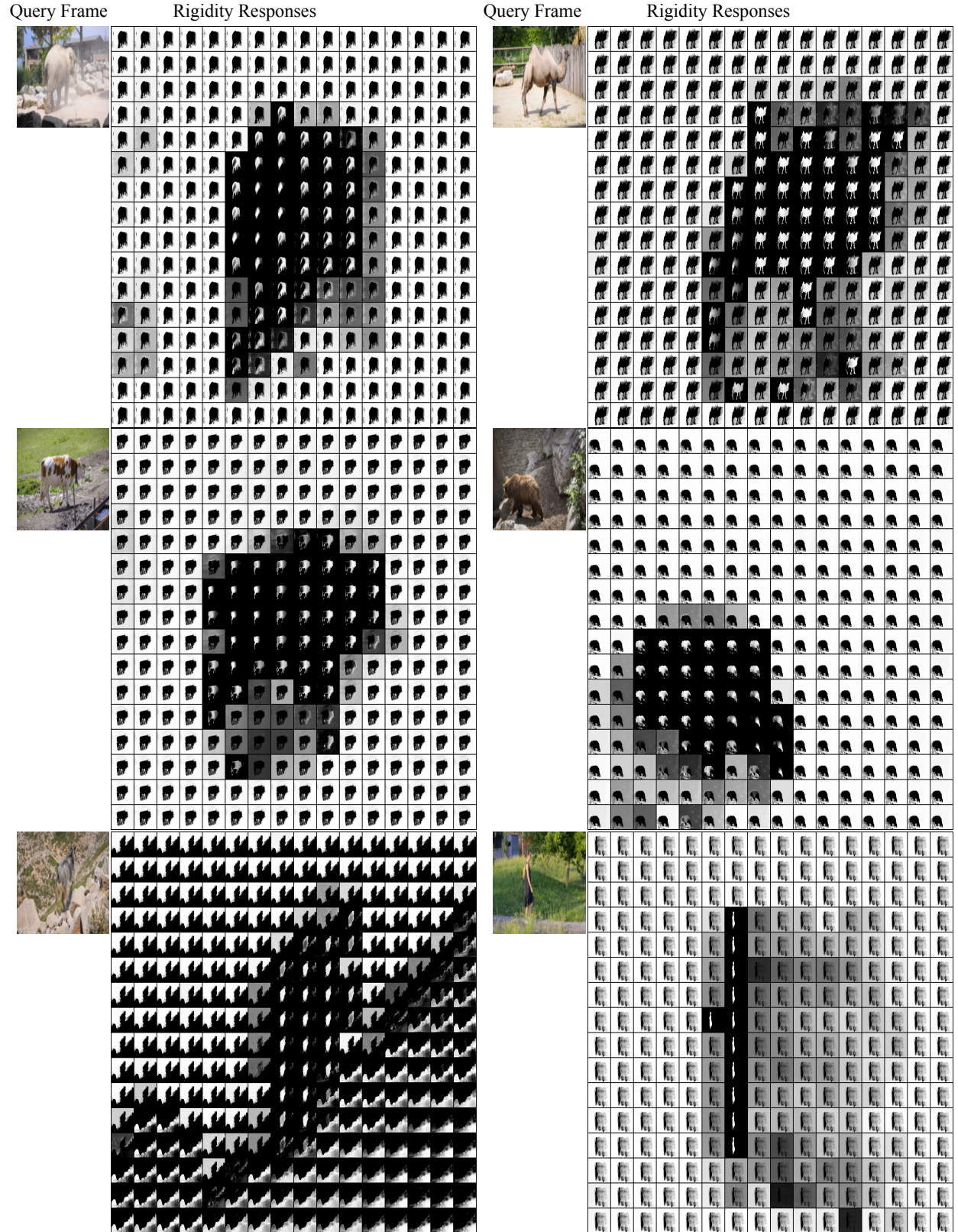


Figure 4: **Rigidity Response Grids on More DAVIS Scenes.** Here we plot rigidity response grids for more DAVIS scenes. We show the query frame to the left of each rigid response grid. Zoom in to observe the per-track rigidity maps.

Video Input

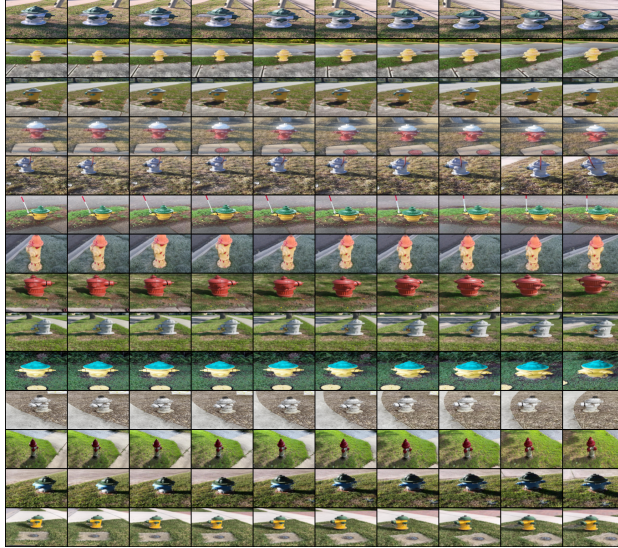


Epipolar Flow Mask



Figure 5: **Epipolar Rigidity Masks.** For the self-supervised depth setting, we compute per-frame optical flow and estimate scene elements which are potentially non-rigid by thresholding the Sampson re-projection error. Here we plot a video and corresponding epipolar rigidity mask.

Video Input



Depth Estimate

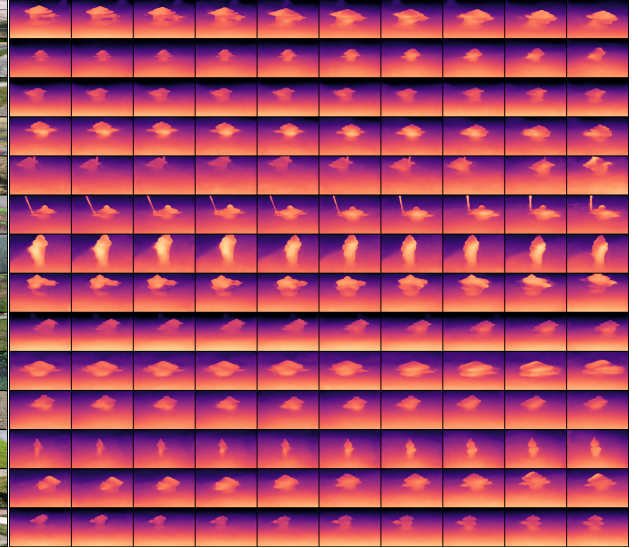


Figure 6: **Hydrant Depth Estimates.** Here we plot a random set of generalizable depth estimates for the CO3D Hydrant scenes which we perform pre-training on.

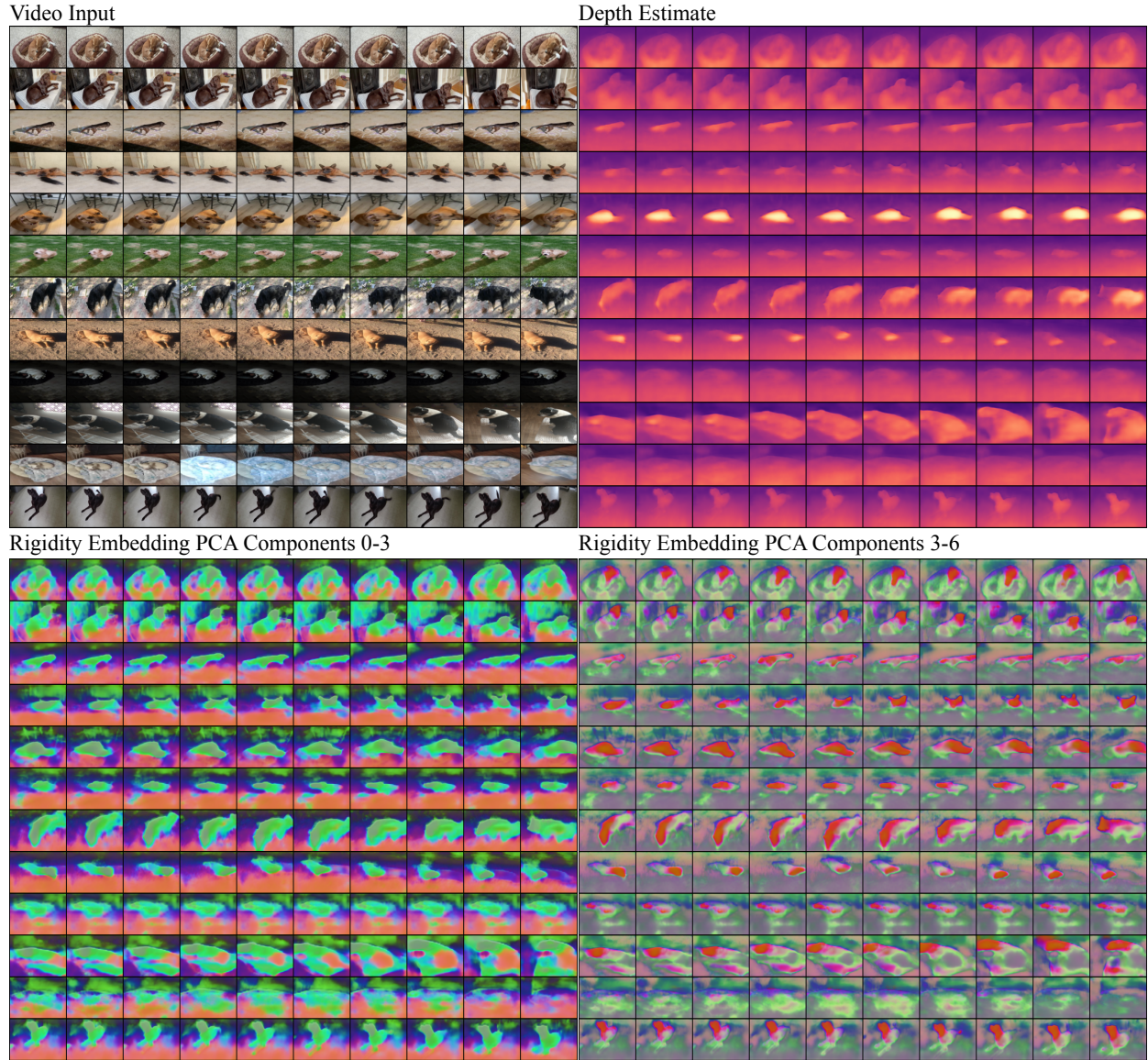


Figure 7: **Dog Depth Estimates.** Here we plot a random set of generalizable depth (top right) and rigidity embeddings (bottom) for an input video (top left). For the rigidity embeddings, we plot the first (bottom left) and next (bottom right) three PCA components.