

In addition to this figure, we show a comprehensive visual comparison for DUST3R and our method in Figures A.5, A.6, and A.7. We notice that sometimes DUST3R has catastrophic failure and the cameras are often incorrect for our hand-drawn scenes. Toon3D is better equipped to handle inconsistent images.

9. Toon3D Labeler

Figure A.3 shows a screen capture of the Toon3D Labeler. We will make this tool available for others to use.

10. Toon3D Dataset

We choose to use cartoon scenes that are hand-drawn rather than using animated scenes that are rendered or based on an underlying 3D model. We select a variety of cartoons based on popularity. Table A.1 shows our datasets and relevant annotation info, including how many images we use to create each scene and how many point labels are used. We use a varying number of point labels, ranging from only 46 points (Magic School Bus) to as many as 191 points (BoJack Room) in a particular scene. This range is meant to convey the robustness of our method to handle a few or many user-defined correspondences. We also note that “Family Guy House” and “Planet Express” are most likely 3D assets, rather than hand-drawn. We learned about this during the project, and we made the conscious decision to include these scenes to showcase the diversity of our method. DUST3R still struggles on these scenes.

11. Deformable mesh topology

In Figure A.4, we show an illustration of how we go from an image, a depth map, and our labeled correspondences, to a 3D mesh which can be deformed.

12. Sparse-view reconstruction data

We obtain sparse-view images from Airbnb Our overview video shows two rooms and their images. The “Living room”, shown in the paper as well, has 5 images. “Bedroom 2” has 8 images. Videos of our Toon3D reconstructions and renders are shown for both rooms on our supplementary website.



Figure A.1. **Geometrical inconsistencies in cartoons.** Are these orange arrows consistent? It is incredibly difficult to tell as a human, but COLMAP and SfM pipelines fail on these images, even with our hand-labeled correspondences.

Table A.1. **Toon3D Dataset.** Here are some statistics for the Toon3D Dataset. We have ~ 7 images per scene, for a total of 79 images across the 12 scenes. Each image has on average 18.3 points per image, but it varies per scene.

	Num images	Num points	Avg. num points / image
Avatar House	8	156	19.5
Bob's Burgers	7	147	21.0
BoJack Room	12	191	15.9
Family Guy Dining	7	184	26.3
Family Guy House	6	133	22.2
Krusty Krab	9	82	9.11
Magic School Bus	5	46	9.20
Mystery Machine	6	55	9.17
Planet Express	5	137	27.4
Simpsons House	5	137	27.4
Rick and Morty	4	99	24.8
Spirited Away	5	75	15.0
Total	79	1442	18.3

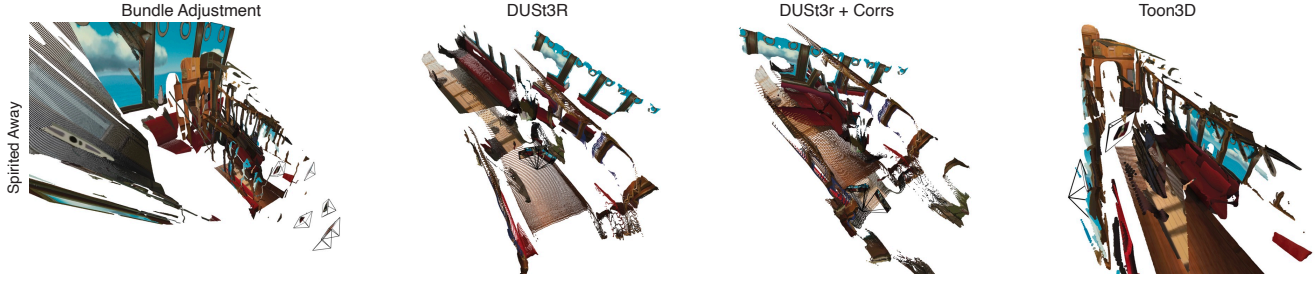


Figure A.2. **Baselines.** We compare our method on the Spirited Away scene with baselines mentioned in the paper. Bundle Adjustment fails because it is unconstrained and doesn't use a prior to recover depth. We visualize the result by backprojecting monocular depths at the recovered camera locations. DUST3R, a data-driven method, performs better and recovers a more plausible result but is still inconsistent. DUST3R + Corrs is slightly improved by using our labeled points at the correspondence locations, but it cannot recover fully from DUST3R's initial prediction. Toon3D (our method) produces the most consistent and realistic structure.

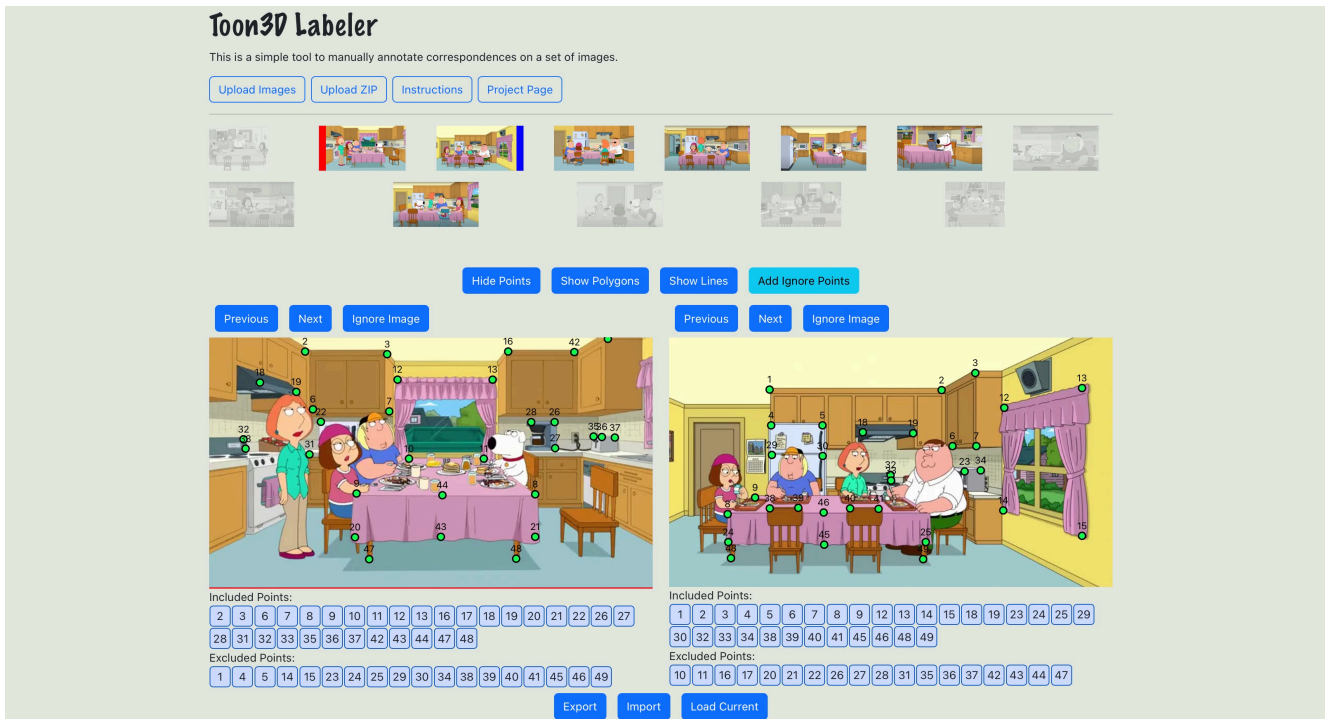


Figure A.3. **Toon3D Labeler.** Here is a screen capture from the Toon3D Labeler interface. Using the labeler, a user can label points and masks, and one can interactively visualize the depth map to avoid labeling on depth boundaries (see the website for a screen recording of this). Our Toon3D Labeler is a general labeling tool for labeling multi-view correspondences.

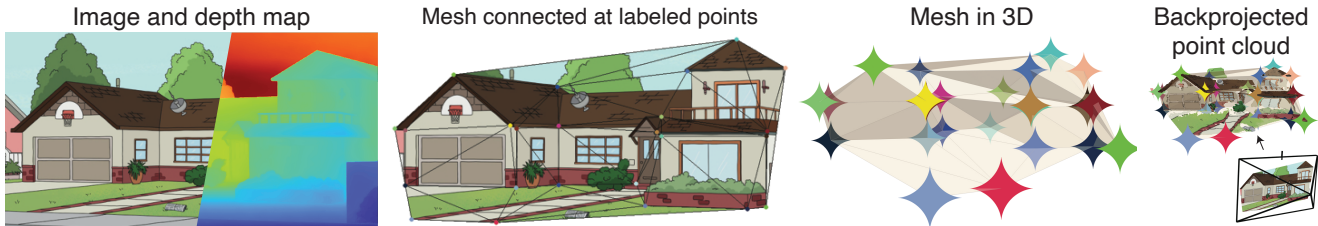


Figure A.4. **Deformable mesh topology.** We start with an image and predicted depth map (left). Then, we create a mesh with the 2D correspondences to define the topology (middle left). This mesh lives in 3D, where larger diamonds are closer to the camera (middle right). We optimize the 3D vertices to achieve multi-view consistency. After convergence, we use barycentric interpolation to query the RGB and depth maps in order to create the dense 3D point cloud, shown on the right.

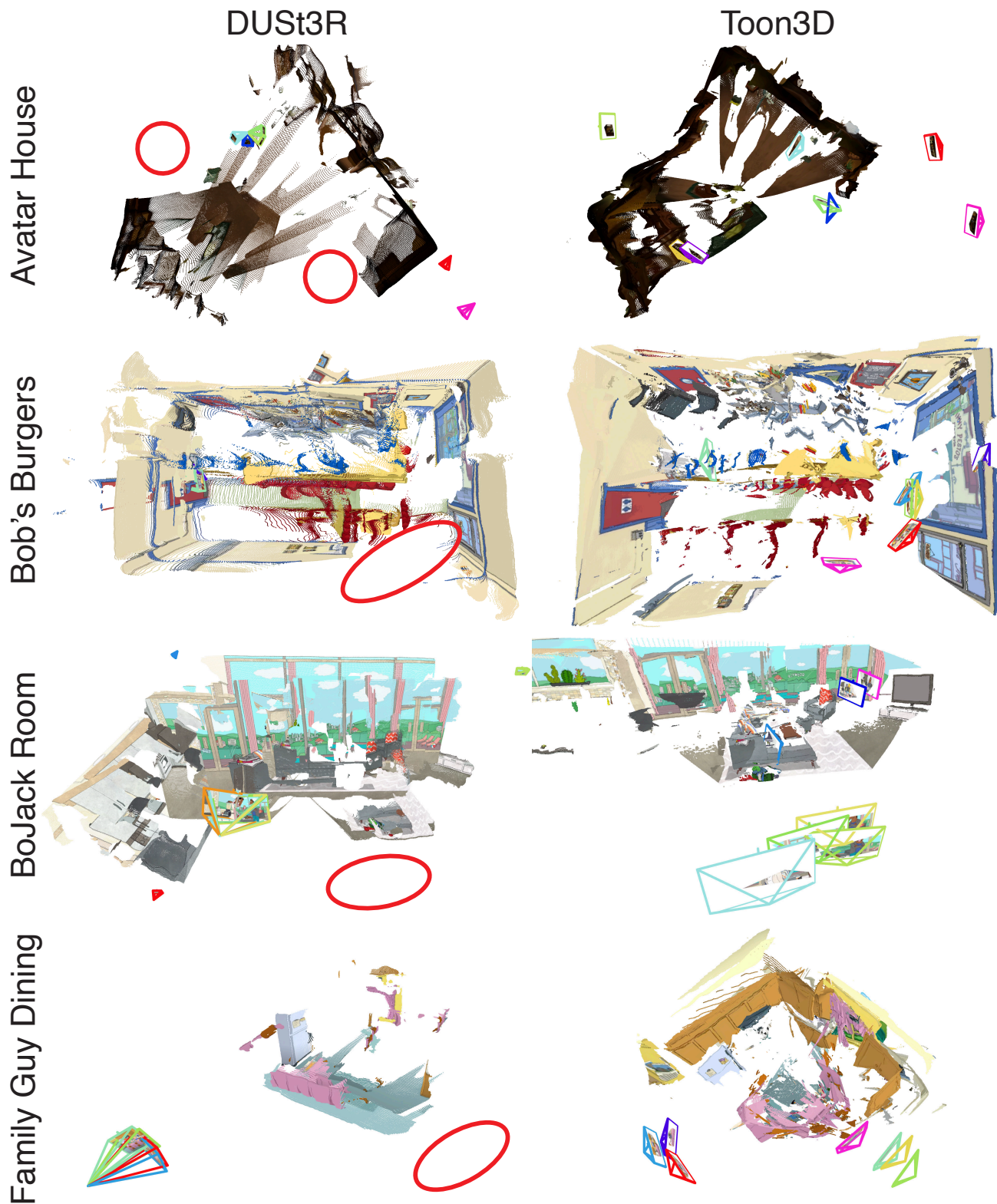


Figure A.5. **DUST3R vs. Toon3D comparison.** We compare results from DUST3R and Toon3D on Avatar House, Bob's Burgers, BoJack Horseman, and Family Guy Dining. The color of each camera frustum corresponds to the same image across DUST3R and Toon3D results. Sometimes DUST3R obtains a good point cloud, but the cameras are in bad locations. However, often times, DUST3R fails catastrophically. The red circles signify where cameras are missing in DUST3R that our method correctly places. (This is the same figure as in the main paper, reproduced for ease of access)

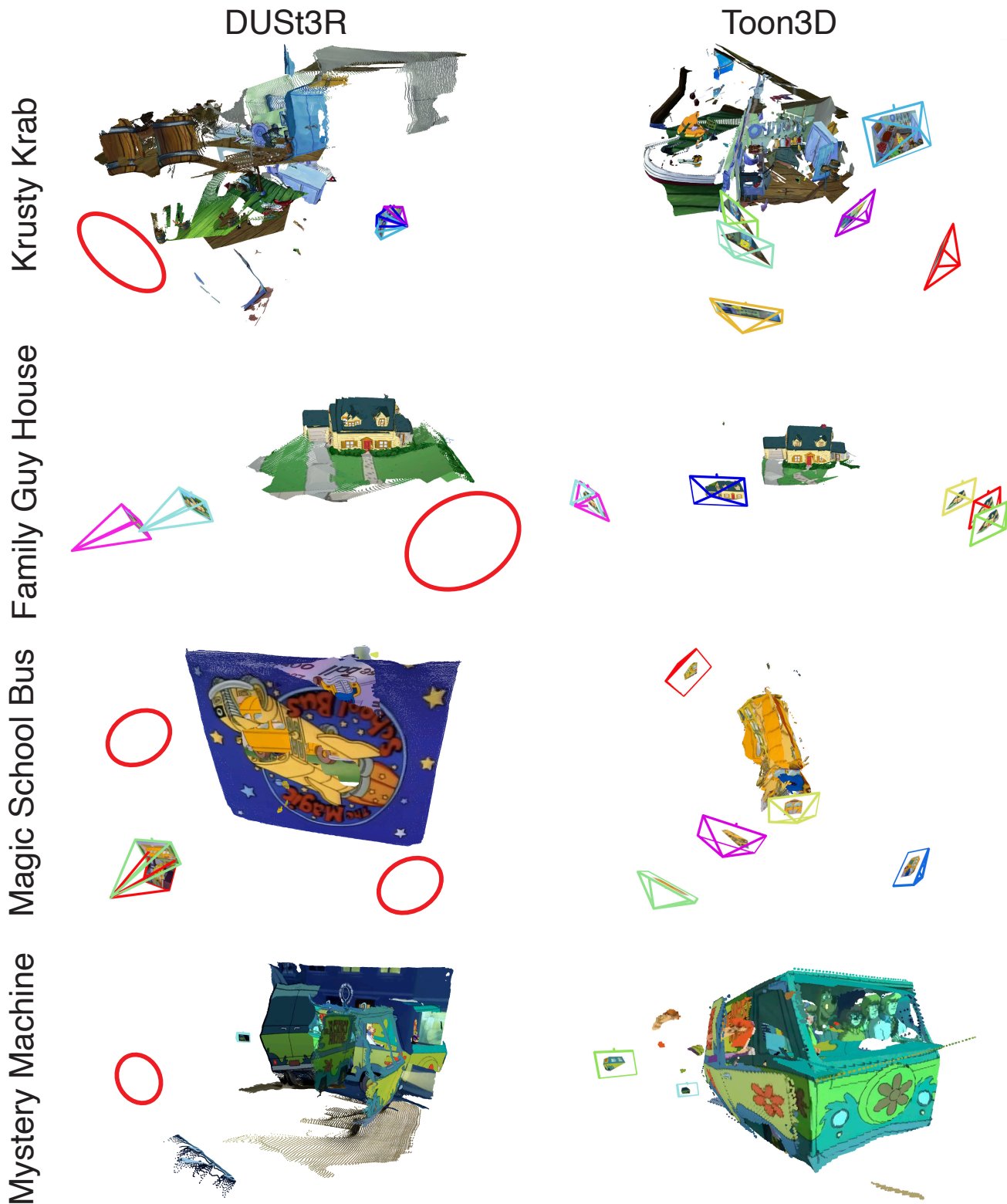


Figure A.6. **DUST3R vs. Toon3D comparison.** We compare results from DUST3R and Toon3D on Family Guy House, Krusty Krab, Magic School Bus, and Mystery Machine. The color of each camera frustum corresponds to the same image across DUST3R and Toon3D results. Sometimes DUST3R obtains a good point cloud, but the cameras are in bad locations. However, often times, DUST3R fails catastrophically. It does well on “Family Guy House” since this data is likely not hand-drawn (see Section 10). The red circles signify where cameras are missing in DUST3R that our method correctly places.

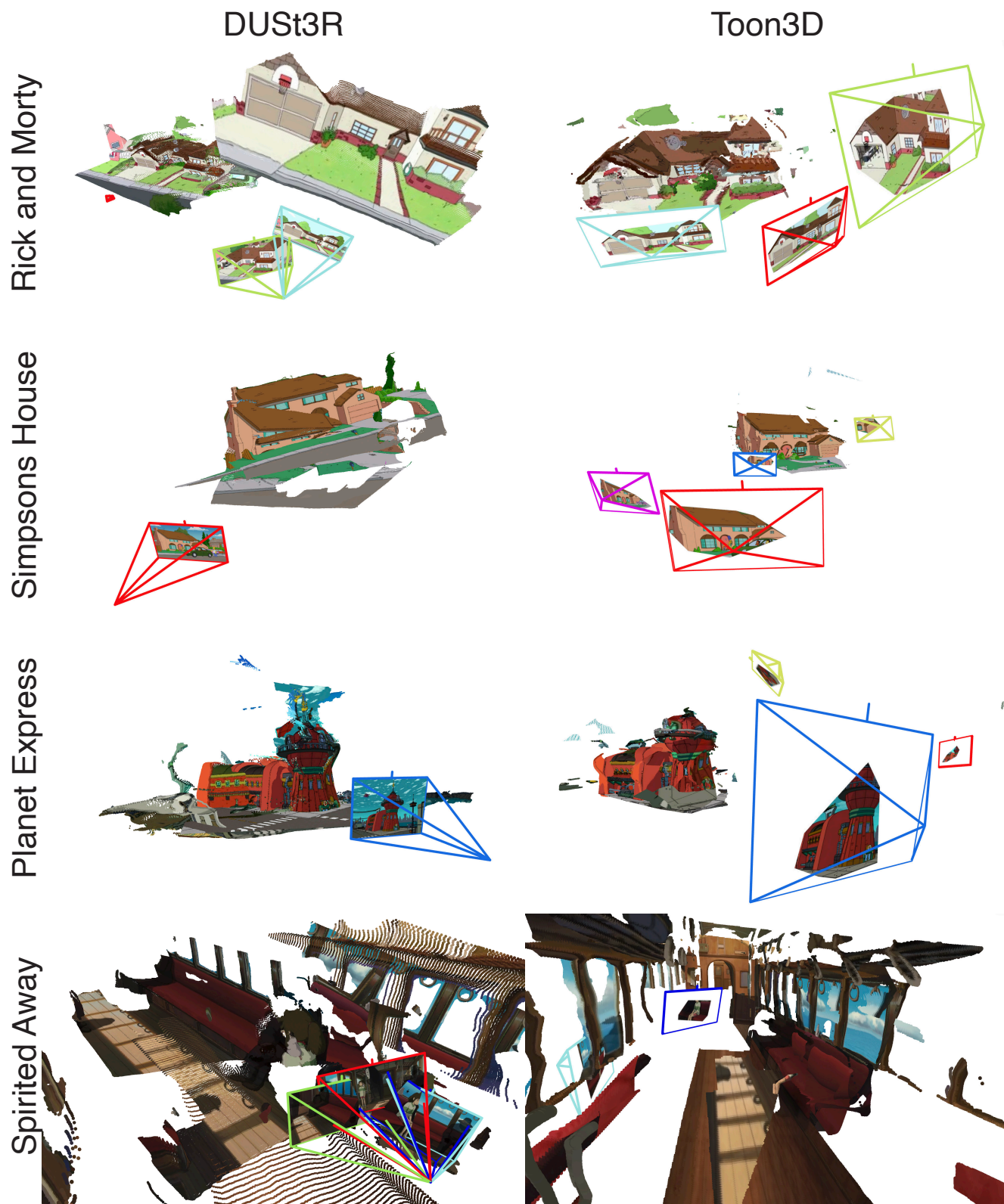


Figure A.7. **DUST3R vs. Toon3D comparison.** We compare results from DUST3R and Toon3D on Planet Express, Simpsons House, Rick and Morty, and Spirited Away. The color of each camera frustum corresponds to the same image across DUST3R and Toon3D comparison. Sometimes DUST3R obtains a good point cloud, but the cameras are in bad locations. However, often times, DUST3R fails catastrophically. It does well on “Planet Express” since this data is likely not hand-drawn (see Section 10).