Toon3D: Seeing Cartoons from New Perspectives

Ethan Weber[*2] Riley Peterlinz[*2] Rohan Mathur[2]
Frederik Warburg[1] Alexei A. Efros[2] Angjoo Kanazawa[2]

[*]Equal contribution [1]Teton.ai [2]UC Berkeley

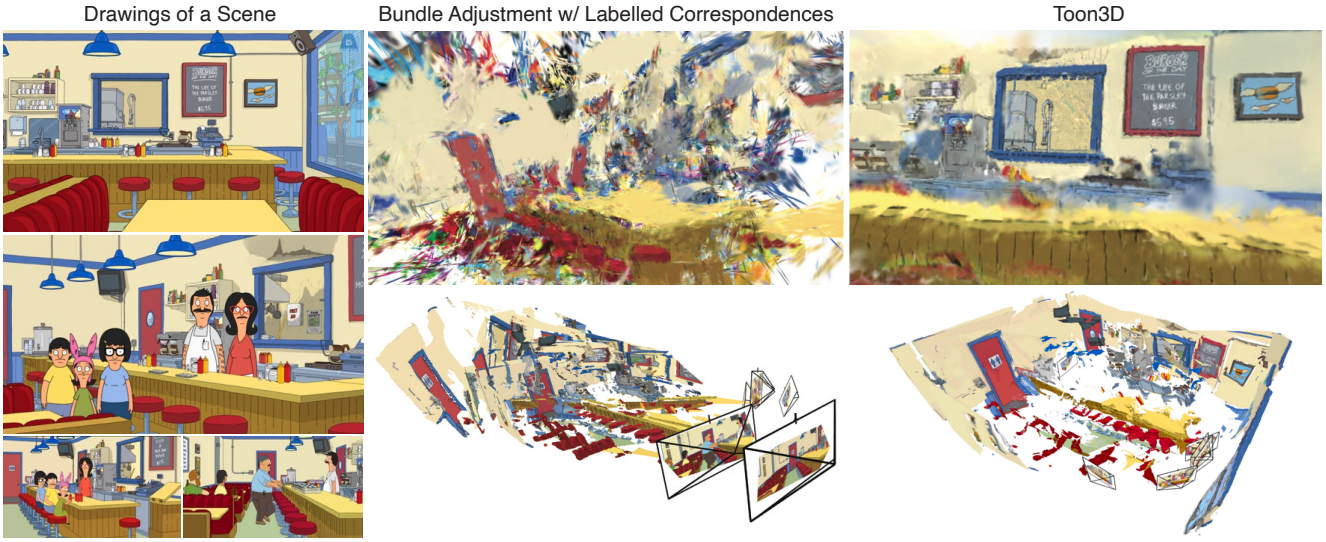

Figure 1. **Reconstructing a 3D scene from 3D inconsistent images.** Cartoons and animations often depict scenes that are not geometrically consistent by design (left), making it challenging for classical Structure-from-Motion (SfM) techniques to reconstruct these scenes as they assume 3D consistency (middle). However, humans can easily perceive the underlying 3D scene from these images. We introduce Toon3D, which addresses these challenges by deforming images during reconstruction to account for geometric inconsistencies and leveraging monocular depth priors. The middle column illustrates how Bundle Adjustment fails, even with manually labeled correspondences, resulting in scattered Gaussian splats (top) and misaligned camera reconstructions visualized by backprojected monodepths (bottom). The right column shows our Toon3D results, with more coherent Gaussian splats (top) and well-structured point clouds and camera views (bottom), demonstrating significantly improved 3D consistency. Our project page is toon3d.studio.

Abstract

We recover the underlying 3D structure from images of cartoons and anime depicting the same scene. This is an interesting problem domain because images in creative media are often depicted without explicit geometric consistency for storytelling and creative expression—they are only 3D in a qualitative sense. While humans can easily perceive the underlying 3D scene from these images, existing Structure-from-Motion (SfM) methods that assume 3D consistency fail catastrophically. We present Toon3D for reconstructing geometrically inconsistent images. Our key insight is to deform the input images while recovering camera poses and scene geometry, effectively explaining away geometrical inconsistencies to achieve consistency. This process is guided by the structure inferred from monocular depth predictions. We curate a dataset with multi-view imagery from cartoons and anime that we annotate with reliable sparse correspondences using our user-friendly annotation tool. Our recovered point clouds can be plugged into novel-view synthesis methods to experience cartoons from viewpoints never drawn before. We evaluate against classical and recent learning-based SfM methods, where Toon3D is able to obtain more reliable camera poses and scene geometry.

1. Introduction

Humans typically have little trouble inferring the relative camera poses and 3D structure from hand-drawn cartoons. However, current structure-from-motion (SfM) pipelines fail to reconstruct these scenes because (1) the images are not geometrically consistent, (2) the images do not obey physically plausible camera models, (3) the scenes are typ-

ically only drawn from a sparse set of views, and additionally, (4) many outlier correspondences from automatic methods. In this work, we overcome these challenges by proposing a piecewise-rigid deformable optimization framework that recovers camera poses and 3D scene from geometrically inconsistent images (see Fig. 1).

Our pipeline consists of a joint optimization to recover cameras and aligned geometry. It takes a set of correspondences as input, which we backproject into 3D using the depth from a monodepth network [21, 41]. We align these sparse correspondences in 3D to estimate the camera intrinsic and extrinsic parameters. Simultaneously, we also deform the image and the associated depth such that images satisfy 3D consistency. We regularize our warps with 2D and 3D rigidity losses to prevent degenerate solutions.

We also propose the Toon3D Dataset and the Toon3D Labeler which is a user-friendly annotation tool, where a user can label point correspondences between images while segmenting transient objects. The Toon3D Labeler is a hosted website with no installation, so anyone can get up and running with it easily. We intentionally highlight Toon3D Labeler as a contribution of our paper because artists work with cartoon drawings regularly, and this tool fits nicely into a human-in-the-loop framework for recovering 3D from these drawings. Our recovered 3D model may help artists draw novel viewpoints. We use our labeler to label 12 scenes from popular cartoons and anime, such as Sponge Bob and Spirited Away, and we release these as the Toon3D Dataset.

To the best of our knowledge, we are the first to present a pipeline for reconstructing cartoon or hand-drawn scenes. Our pipeline yields reliable camera poses, whereas COLMAP [30] and DUSt3R [42] fail to reliably recover camera poses and 3D scene geometry (even with human-annotated correspondences) due to 3D inconsistencies in the input images. In contrast, our 2D image warpings of the original images enable us to reconstruct the full 3D geometry, while also visualizing geometrical inconsistencies in the drawings.

We evaluate our pipeline on 12 popular scenes (10 cartoon TV shows, 1 movie) to highlight the effectiveness of our pipeline in obtaining good camera poses and reconstructions. We show reconstructions of our recovered 3D point clouds and create an immersive visualization by rendering a 3D Gaussian Splatting [22] representation that are initialized from our aligned point cloud. We evaluate our proposed alignment objectives and losses qualitatively and quantitatively. We demonstrate that our warps can highlight geometric inconsistencies in hand-drawn images. We further validate the quality of Toon3D to estimate camera poses, when the scenes are in fact geometrically consistent. We show that we can obtain the 3D geometry of Airbnb rooms with sparse views. Finally, we show that Toon3D is also useful for reconstructing the 3D geometry from paintings depicting the same landmark from different views.

Humans routinely make successful 3D scene inferences from imagery (e.g. cartoons) which is 3D-inconsistent and/or not following perspective projection [14]. Toon3D is a step toward achieving this type of qualitative 3D understanding of cartoons. We validate our pipeline and will release all data, code, and tools to easily process any cartoon. We hope our contribution serves as a useful framework to build tools that, like humans, can reconstruct and understand qualitative 3D.

2. Related work

Multi-view geometry estimation. Structure-from-Motion (SfM) [13, 32] takes in images, detects and matches correspondences, and solves for camera parameters. COLMAP [30] is a popular SfM pipeline, but it fails for wide baseline images (few correspondences), images with a lot of moving objects, or geometric inconsistencies typically present in cartoons. Improvements in keypoint detection [10, 11], matching [29, 34] and optimizations [36] have been proposed to better handle wide baselines [39] and be robust to transient objects [4]. However, all these methods make a fundamental assumption that the input images are geometrically consistent. In contrast, we propose a method that accounts for such inconsistencies by explaining away the inconsistencies when possible via image deformation.

Reconstructing image collections. Facade [8], a seminal early work in image-based modeling and rendering, used a set of photographs of an architectural scene to recover a textured 3D model using structure-from-motion with human-specified volumetric constraints. Phototourism [32] and Building Rome in a Day [1] pioneered the use of large online photo collections for 3D reconstruction. Object-centric methods like CMR [19, 20] recover 3D models of animals through a learned deformation model. For non-rigid dynamic scenes, there exist methods which explain small variations in a video via a 3D model with a time-conditioned warp fied to be as rigid as possible [25, 27, 37]. With methods that require deformation, techniques such as As-Rigid-As-Possible (ARAP) [33] are useful. These problems are relevant in a sense that they need to reconstruct scenes with transient variations in each image. We propose a relevant but novel and under-explored problem setting where the input images are meant to depict the same 3D scene, through geometrically inconsistent multi-view imagery.

Paintings to 3D. Most attempts at recovering 3D from drawings and paintings have focused on the single view setting, with missing 3D information provided either manually by the user or via learning. Important early user-assisted approaches for generating 3D scenes from a single painting include Tour into the Picture [16], which assumed single-point perspective, and the more general Single View Metrol-

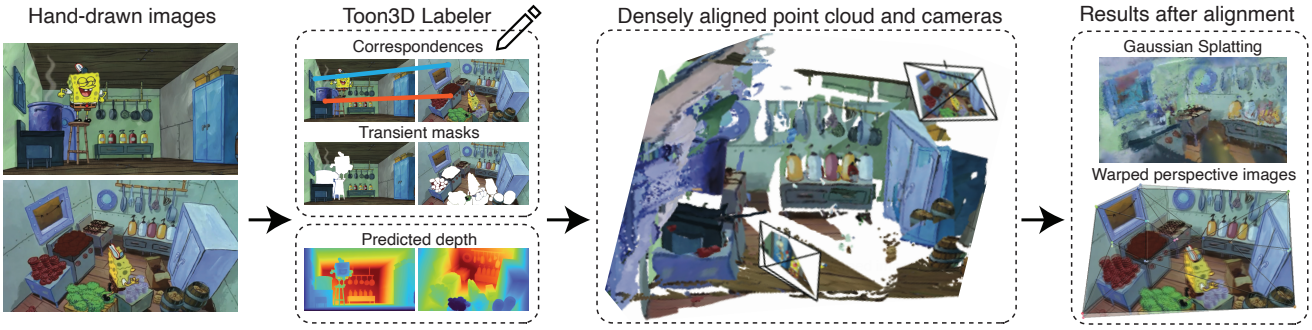

Figure 2. **Toon3D overview.** Our framework consists of labeling images with our interactive Toon3D Labeler tool, recovering camera poses and aligning a dense point cloud, and visualizing the dense reconstruction with Gaussians to create an immersive visual experience.

ogy [6]. Automatic Photo Popup [15] replaced the manual parts of the reconstruction process with early machine learning techniques, and was able to generalize to paintings. Aubry *et al.* [3] is a rare attempt to connect different paintings of the same scene by using a 3D model. There has also been a few attempts to recover a 3D model from a set of sketches of the same object [9, 12]. Our approach similarly explores reconstructing creative expressions (i.e., drawings) but from multiple drawings of the same scene as seen in settings like cartoons instead of a single image.

Computer vision in TV and Film. Previous works have explored reconstructing TV shows and films. Pavlakos *et al.* [26] recover camera shot locations, 3D human poses, and gaze understanding, enabling applications such as post-production re-rendering with novel camera paths. MovieNet [17] proposes a large dataset of popular films annotated with bounding boxes, actions, and cinematic style for a holistic understanding of movies. Zhu *et al.* [47] align movies and books to obtain fine-grained descriptions of appearances of objects and characters, as well as high-level semantic understanding into how characters think and reason. Additionally, some works have looked into character reconstruction for cartoon characters [5, 18, 31] but none have looked at recovering camera poses and reconstructing full 3D environments. Our work is most similar to [26], but we tackle geometrical inconsistencies in cartoons and animation instead of video sequences in sitcoms.

3. Toon3D Dataset and Labeler

To study this unique problem, we introduce the Toon3D Dataset, which consists of 12 cartoon scenes (10 TV shows, 1 movie) each with 5-12 images depicting the same environment. An innate challenge in cartoons is that correspondences are difficult to obtain automatically. We tried several SOTA keypoint detectors [2, 10, 29, 34], but they often fail due to extreme viewpoint changes, the presence of transient objects such as characters, and the images' stylistic, low-texture expression. Since our focus is on reconstructing the underlying static 3D scene, we leave the automatic removal of foreground objects and estimation of 2D correspondences to future work. Instead, we develop the Toon3D Labeler, a human-in-the-loop tool for segmenting transient objects and annotating sparse 2D correspondences. The Toon3D Labeler is hosted online with no installation required, making it easily accessible. See the supplement pdf for a visualization of this tool. Next, we discuss how we use the Toon3D Labeler to curate our dataset.

Preprocessing. We start with a set of N images $\{I_i\}$ depicting the same scene in a cartoon. Each scene typically has $N \leq 10$ images with wide baselines. We preprocess these images by running a monocular depth network to obtain predicted depths $\{D_i\}$. We normalize the depth maps by dividing by the maximum depth of a labeled correspondence across all depth maps. We experiment with a variety of depth map predictors [21, 41, 44], while all quantitative evaluations are done with Marigold [21]. We also run Segment Anything (SAM) [23] to get a set of masks per image.

Labeling. We label these images using the Toon3D Labeler web interface. To annotate correspondences, the user clicks on corresponding points across all images. When the point is not visible in an image, it is labeled as invisible. Our interface allows users to visualize the depth map, helping them avoid placing correspondences on depth discontinuities. Each annotated image has on average 18 sparse correspondences (see more details in appendix). To select SAM mask, the user simply hovers over a region, the mask will be highlighted, and it can be toggled on and off to discard transients within the image. After labeling, we have pixel correspondences $\mathcal{X} = \{x_{i,c}\}$ where i is the image index and c is the correspondence index. We also have a valid correspondences mask $m_{i,c} = \{0, 1\}$. When $m_{i,c} = 0$, the correspondence is not visible in that image. We denote the predicted depth of the correspondences with $d_{i,c} = D_i(x_{i,c})$.

4. Toon3D Method

We present Toon3D, a method to reconstruct scenes that are only 3D consistent in a qualitative sense, as opposed to existing SfM methods that requires a geometrical consistent scene. Toon3D takes as input multiple images of the same scene with point correspondences, mask annota-

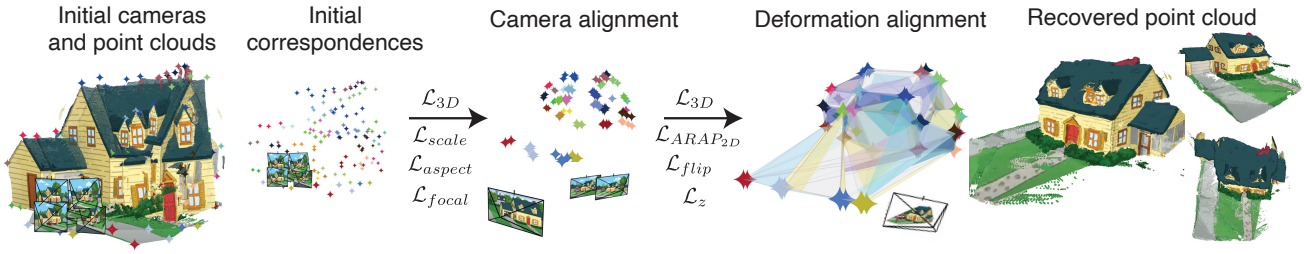

Figure 3. **Toon3D alignment.** The camera alignment objective aligns the point clouds while optimizing for camera intrinsics and extrinsics. Deformation alignment deforms the images to obey a perspective camera model. In practice, our method uses all the losses described here to obtain an aligned point cloud and posed images.

tions, and estimated monocular depth, and outputs camera poses for each image, a 3D point cloud, and a warping of the original images, such that they obey a perspective camera model. The output point cloud can be converted into a Gaussian Splatting [22] representation to create a more immersive novel-view experience. We optimize for cameras and geometry by aligning backprojected point correspondences and allowing the images to deform while still obeying a perspective camera model and multiview geometry. We will now explain our approach in more detail.

4.1. Camera Alignment

The first objective of our pipeline is to obtain camera poses. Since the images are not geometrically consistent, the standard bundle adjustment process that enforces a single 3D point for every corresponding points does not lead to correct camera poses. Instead, we make use of the monocular depth priors in each image and solve for camera poses that align the backprojected correspondences in 3D.

Specifically, we first backproject our sparse correspondences into 3D with

$$p(x_{i,c}) = R_i K_i^{-1} \left((s_i \cdot d_{i,c} + h_i) \begin{bmatrix} x_{i,c,u} + \delta_{i,c,u} \\ x_{i,c,v} + \delta_{i,c,v} \\ 1 \end{bmatrix} \right) + t_i, \tag{1}$$

where the depth d of each point is estimated with a monocular depth network and we solve for camera intrinsics $K_i = \text{Diag}(f_{i,x}, f_{i,y}, 1)$, where $f_{i,x}, f_{i,y}$ are the focal lengths, camera rotations R_i, translations t_i, depth scale s_i, depth shift h_i, and uv shifts $\delta_{i,c,u}, \delta_{i,c,v}$. These parameters are optimized to minimize the 3D correspondence loss

$$\mathcal{L}_{3D} = \frac{1}{|\mathcal{X}|} \sum_{i=1}^{N} \sum_{j<i}^{N} \sum_{c=1}^{M} m_{i,c} \cdot ||p(x_{i,c}) - p(x_{j,c})||_2^2, \tag{2}$$

which pulls the backprojected correspondences together in 3D. N is the number of images and M is the number of correspondences for image i. We found minimizing 3D distance rather than 2D reprojection error works better empirically, which we ablate in experiments.

Estimating these camera poses from just few sparse correspondences is a very under-constrained problem even with a strong depth prior. Therefore, we found that adding the following regularizers were necessary to reliably estimate camera poses across all scenes.

$$\mathcal{L}_{scale} = ||1 - \frac{1}{N} \sum_{i=1}^{N} s_i||^2 \tag{3}$$

$$\mathcal{L}_{aspect} = \sum_{i=1}^{N} ||\frac{f_{i,x}}{f_{i,y}} - \frac{h_i}{w_i}||^2 \tag{4}$$

$$\mathcal{L}_{focal} = \sum_{i=1}^{N} f_{i,x} + f_{i,y}, \tag{5}$$

where $\mathcal{L}_{scale}$ encourages a scale close to 1 such that the scene does not shrink, $\mathcal{L}_{aspect}$ balances $f_{i,x}$ and $f_{i,y}$ to maintain aspect ratio of the camera with the original image's height h_i and width w_i, and $\mathcal{L}_{focal}$ penalizes large focal length to prefer wide-angle cameras over far away and zoomed in shots. We also have losses that penalize scales s_i and shifts h_i if they become negative with $\mathcal{L}_{neg}(x) = ||\frac{1}{N}\sum_{i=1}^{N} \max(0, -x_i)||^2$.

Thus, our final camera alignment objective is as follows

$$\begin{aligned} \mathcal{J}_{\text{camera}} = \mathcal{L}_{3D} + \lambda_{scale}\mathcal{L}_{scale} + \lambda_{aspect}\mathcal{L}_{aspect} \\ + \lambda_{focal}\mathcal{L}_{focal} + \lambda_{neg}(\mathcal{L}_{neg}(s) + \mathcal{L}_{neg}(h)), \end{aligned} \tag{6}$$

which gives us an coarse estimate of the 3D structure and the camera poses.

4.2. Deformation Alignment

Although the previous losses yield coarse estimates of the scene, they do not result in a coherent point cloud due to the geometric inconsistencies in cartoon images. To address this, we propose jointly deforming each image and its corresponding depth map to achieve geometric consistency. Our method introduces a set of dense alignment objectives, which, when optimized, refine the camera poses. This produces a densely aligned, warped 3D point cloud along with images that are geometrically consistent in 3D and adhere to a perspective camera model.

To do this, we use the same optimization objectives from Sec. 4.1 but now with more freedom as we also allow the input image to be warped to further minimize $\mathcal{L}_{3D}$. However, naively warping every pixel location with full degrees of

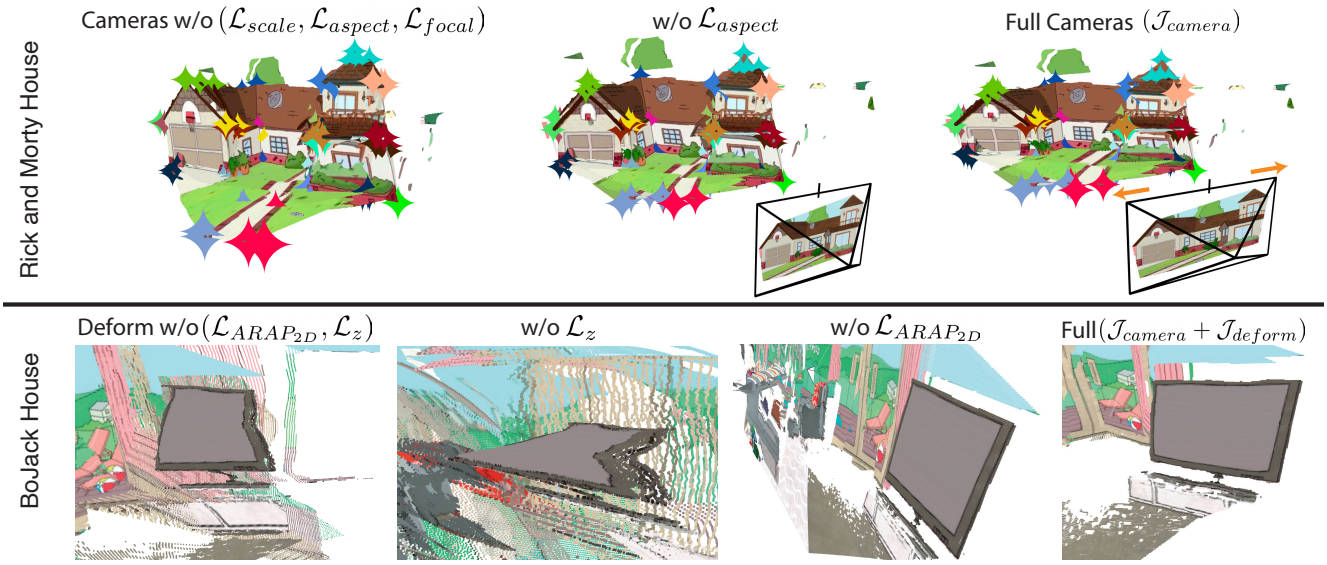

Figure 4. **3D alignment ablations.** Row 1 (Rick and Morty House) shows regularization's impact on scene shaping. Optimized shift and scale parameters can adjust point clouds to better align at correspondences. This is evident as the starred points converge. The aspect regularization keeps the optimized image close to its original aspect ratio. Row 2 (BoJack Horseman House) explores the effects of different warp regularizers ($\mathcal{L}_{ARAP_{2D}}$ and $\mathcal{L}_z$) on scene warping. Without any regularization, warping distorts scene geometry. ARAP alone results in poor 3D warps due to inaccurate depth. z regularization alone limits scene movement, maintaining rigid structures close to the original depth map. Using both strikes a good balance between correctly positioning geometry and preserving structural integrity.

freedom, without any constraints, results in degenerate solutions. To address this, we warp the image using a coarse 3D mesh that approximates the scene and apply a regularizer to ensure the deformation remains piece-wise rigid.

Specifically, we first parameterize two meshes, one unwarped with vertices $V_i \in \mathbb{R}^{M\times 2}$ and another warped with $V_i' \in \mathbb{R}^{M\times 2}$, both with faces $F_i \in \mathbb{R}^{K\times 3}$. Each row of V_i is a 2D point $V_{i,c} = \left[x_{i,c,u}, x_{i,c,v}\right]$ while $V_{i,c}' = \left[x_{i,c,u} + \delta_{i,c,u}, x_{i,c,v} + \delta_{i,c,v}\right]$. We use Delaunay triangulation to create the mesh topology. See Fig. 3 for illustrations of this 3D mesh that represents the scene for each image.

We optimize the $\delta_{i,c,u}$ and $\delta_{i,c,v}$ for each image with regularizers to constrain the warps to be as-rigid-as-possible to prevent degenerate solutions. First, we regularize such that the optimized vertices are encouraged to follow a rigid transform in the 2D image plane via

$$\mathcal{L}_{ARAP_{2D}} = \frac{1}{N \times |\mathcal{F}|} \sum_{i=1}^{N} \sum_{f \in F_i} ||\pi(V_{i,c}'[f]) - A_{i\to j}\pi(V_{i,c}[f])||^2, \tag{7}$$

where π denotes the 2D projection with the current camera parameters, $V_{i,c}[f] \in \mathbb{R}^{3\times 2}$ are vertices indexed at face f with no z component, $V_{i,c}$ are the optimized 2D projected vertices, and $A_{a\to b}$ is the best fit 2D rigid transform in the image plane that transforms vertices $V_{i,c}[f]$ to the new vertices $V_{i,c}'[f]$.

Additionally, we use these two losses

$$\mathcal{L}_{flip} = \frac{1}{N \times |\mathcal{F}|} \sum_{i=1}^{N} \sum_{f \in F_i} ||\max(0, t_{area} - \det(V_{i,c}'[f]))||^2, \tag{8}$$

$$\mathcal{L}_z = \frac{1}{N \times |\mathcal{X}|} \sum_{i=1}^{N} \sum_{c=1}^{M} m_{i,c} \cdot ||d_{i,c}' - d_{i,c}||, \tag{9}$$

where $d_{i,c}' = s_i \cdot d_{i,c} + h_i$, $\mathcal{L}_{flip}$ penalizes if the triangle face gets too small or flips, and $\mathcal{L}_z$ encourages the warped depth to be close to the original predicted depth. t_{area} is the minimum area a face can be, and $\det$ gives the signed face area. We set t_{area} to 10% of the original face area.

Finally, we use barycentric interpolation to densely warp the RGB and depth maps according to our deformed vertices V'. We warp the RGB image with barycentric interpolation according to the original vertices V and the deformed mesh V'.

Our deformation alignment objective becomes an extension of our camera alignment, where besides optimizing for poses, focal lengths, rotation, scale, and shift, we also optimize the mesh topology. Our final objective is

$$\text{argmin}_{R,t,f,s,h,\delta} \mathcal{J}_{\text{align}} = \mathcal{J}_{\text{camera}} + \mathcal{J}_{\text{deform}} \tag{10}$$

$$\mathcal{J}_{\text{deform}} = \lambda_{ARAP_{2D}}\mathcal{L}_{ARAP_{2D}} + \lambda_{flip}\mathcal{L}_{flip} + \lambda_z\mathcal{L}_z \tag{11}$$

Optimizing this objective results in an accurate 3D poses as well as an image and depth map that obey the perspective camera model as well as global 3D geometry consistency.

4.3. Gaussian visualization

At this point, we have aligned depth maps which are back-projected into a combined 3D point cloud. We could visualize the point cloud as-is, but we find that Gaussian Splatting

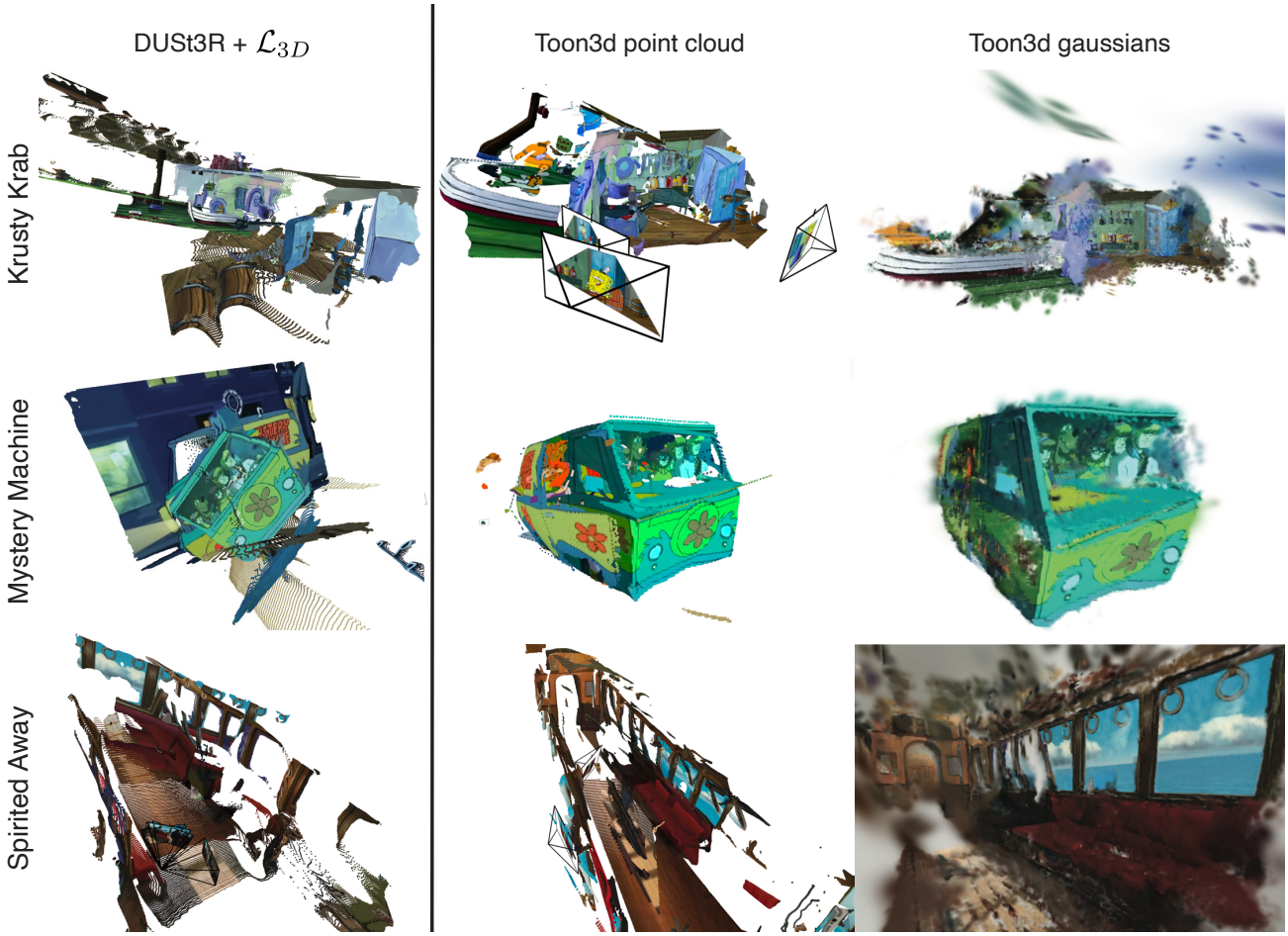

Figure 5. **3D reconstructions of cartoons.** Off-the-shelf methods like COLMAP fail completely. State-of-the-art learning based method DUSt3R [42] also fails catastrophically on many scenes even with labeled correspondences (left). Our method (middle), recovers reliable cameras and plausible pointclouds, which can be visualized with Gaussians for a more immersive experience. For the SpongeBob scene (top), we label point correspondences between walls to reconstruct two rooms together. Notably, our method works with different depth predictors. From top to bottom, we show results with MoGe [41], Depth Anything V2 [44], and Marigold [21].

can create a more immersive experience. Gaussian Splatting [22] is typically initialized by a sparse point cloud from COLMAP, but instead, we initialize it with our dense point cloud. We add a few sparse-view regularizers including the ranking loss from [40] (to reconstruct scenes to be consistent with the predicted depth) and a total variation [46] loss in novel views interpolated between pairs of training views. The transient regions are not ignored in the objective.

5. Experiments

First we show results on cartoon scenes, and then we evaluate our design choices and compare our method with DUSt3R [42], a state-of-the-art learning based 3D reconstruction method. We further test the correctness of our approach on a similar setup but with geometrically consistent photos from an AirBnB listing. We also evaluate our approach on paintings and finally, visualize which parts of the images need to warp to become consistent with each other.

5.1. Cartoon reconstruction

In Fig. 5 we show the results from our pipeline on multiple popular cartoon scenes. On the left, we show results using DUSt3R [42], which often fails catastrophically even with our labeled correspondences. The center column shows our point cloud reconstruction. The right column shows rendered novel views after the Gaussian visualization. We also show a traditional bundle adjustment (BA) baseline in Fig. 1 that optimizes a single 3D point for each labeled correspondence, which recovers inaccurate poses. For clarity, we visualize the dense result by backprojecting monocular depths. Approaches that don't account for geometrical inconsistencies result in poor camera poses. Please see our overview video for better visualization. From start to completion, our method takes on the order of minutes. Finding a few images of a cartoon scene and labeling points is quick due to the web-based viewer, and running our camera alignment and warping takes approximately 1 minute on an

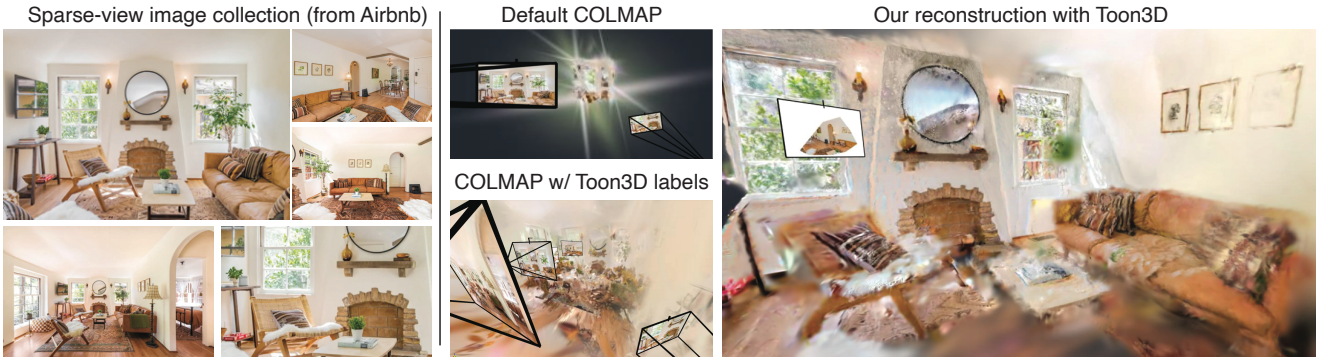

Figure 6. **Sparse-view Reconstruction.** Our pipeline can reconstruct sparse-view image collections that are geometrically consistent as well (left). COLMAP by default only registers 2 out of 5 images and fails to recover structure (middle top). Using Toon3D Labeler correspondences, we get COLMAP to work (middle bottom) but it is initialized with a very sparse point cloud and cannot recover dense details properly. Using Toon3D, we can fully reconstruct the room.

Method	PCC↑	Method	PCC↑
Camera Alignment		*Deformation Alignment*	
$\mathcal{J}_{\text{camera}}$	0.26	$\mathcal{J}_{\text{camera}} + \mathcal{J}_{\text{deform}}$	0.47
$-\mathcal{L}_{focal}$	0.26	$-\mathcal{L}_{z}$	0.42
$-\mathcal{L}_{aspect}$	0.24	$-\mathcal{L}_{ARAP_{2D}}$	0.42
$-\mathcal{L}_{scale}$	0.18	$-\mathcal{J}_{\text{deform}}{}^{*}$	0.36
Traditional BA	0.10	Switch $\mathcal{L}_{3D}$ to $\mathcal{L}_{2D}$	0.31

Table 1. **Quantitative ablations.** We report reprojection error for 5 holdout points on our 12 scenes. PCC is evaluated using a threshold radius set to 3% of the image size ($\alpha = 0.03$). (* Includes $\mathcal{L}_{flip}$)

NVIDIA RTX A5000. Running Gaussian Splatting in Nerfstudio [35] with our additional losses takes ∼3 minutes.

Qualitative ablations. For our default method, we have all parameters free (including scale and shift) with all regularization losses turned on. We show the qualitative tradeoffs for our various losses in Fig. 4. We find our losses help align structure while maintaining an accurate aspect ratio, preventing degenerate warps, and favoring cameras inside walls rather than far away and zoomed in (see caption).

Quantitative evaluation Since our task and data have no ground truth cameras, we design a metric to evaluate 3D consistency, with results reported in Tab. 1. We randomly remove 5 labeled correspondence points from each image of our 12 Toon3D scenes and report the average percentage of correct correspondences (PCC) across all scenes on these held-out points. Similar to PCK [45], PCC considers a correspondence correct if the reprojected point lies within a radius defined as a percentage $alpha$ of the image size. We run our method with various parameters and regularizations turned on and off for ablations and also compare against strong baselines like DUSt3R [42] adapted to our setting, all shown in Fig. 8. In order for a fair comparison, we also compare with a version of DUST3R that uses our correspondences via adding $\mathcal{L}_{3D}$ in their global optimization stage along with our labeled masks applied to the confidence map. Results show that our proposed approach obtains the best PCC across all methods. We find that DUST3R works well occasionally, but when it fails, it fails catastrophically (see supplement pdf). Using our labels helps, but not significantly. Adding the dense alignment warp is necessary to significantly increase the performance. Our experiments validate the need for methods designed to deal with geometrically inconsistent input images.

5.2. Sparse-view Reconstruction Validation

In this section, we validate the correctness of our approach on image collections that are geometrically consistent. Airbnb listings provide suitable test cases, as their photos are often geometrically consistent but sparse with wide baselines. For this evaluation, we reconstruct sparse photo collections from two Airbnb rooms from a listing (8 photos of a bedroom, shown in the supplement webpage, and 5 photos of a living room, shown in Fig. 6). This task is very difficult because SfM pipelines like COLMAP fail to find enough correspondences to accurately recover all poses. Furthermore, even with accurate camera poses, the sparse-view reconstruction setting is especially hard without priors or specialized methods like RegNeRF [24] or ReconFusion [43]. We tackle this sparse-view Airbnb setting with our method for two reasons: (1) to show that we can get COLMAP to work with labeled correspondences from the Toon3D Labeler and (2) to show that our approach works for real sparse photo collections.

When running COLMAP on our Airbnb collections, default COLMAP only registers 46% of the images. This could be possibly improved with better correspondences, e.g. [10, 11, 38], but there is no guarantee of finding enough inlier correspondences if automated methods are used. With our Toon3D Labeler, however, we can manually label the images quickly and get COLMAP to succeed for all images. Manual labels take less than 10 minutes per scene with our tool. We compare the recovered COLMAP cameras with our correspondences with the cameras recovered from Toon3D. The mean relative rotation distance between

Figure 7. **Reconstructing paintings with Toon3D.** Our method enables reconstructing paintings. On the left, we show a few paintings of The Trevi Fountain. In the middle, we show the recovered point cloud and cameras (with warped and cropped images). On the right, we densify the point cloud with Gaussian Splatting.

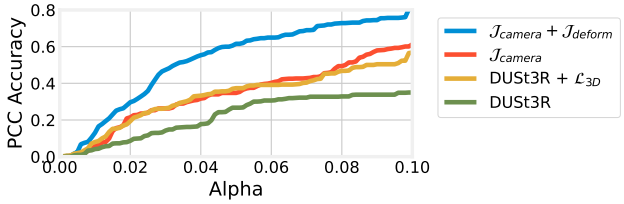

Figure 8. **Baselines evaluation.** We obtain best metrics for percent correct correspondences at image size % thresholds α. We beat DUSt3R and improve it with our labels & $\mathcal{L}_{3D}$.

corresponding pairs in our reconstructions vs. COLMAP's is quite low at only 8.29°, indicating our cameras are similar to ones recovered by COLMAP with human-labeled correspondences. We do not compare translations or focal lengths due to ambiguity between the two, but we note that our camera relative rotations match COLMAP quite well, suggesting that our camera pose estimation is accurate. We show qualitative results for sparse-view reconstruction on real images in Fig. 6 and videos on the supplement.

5.3. Reconstructing paintings

We also show our pipeline can reconstruct paintings of the same scene. Fig. 7 shows results of Toon3D on paintings of The Trevi Fountain found in the Oxford Dataset [7]. This setup requires multiple paintings of the same scene from diverse viewpoints, which is uncommon. However, it presents an interesting problem, and notably, we are able to apply the Toon3D pipeline successfully without any modifications.

5.4. Visualizing inconsistencies

One unique aspect of Toon3D is that we keep the original images around rather than discarding them. They are warped in 2D to obey the global 3D consistency through a perspective camera model. This is fundamentally different than alternative sparse-view generative methods, e.g. Dreambooth3D [28] which fine-tunes on a collection of images and then hallucinates a scene. In Fig. 9 we show where the images deform the most to create a unified consistent 3D structure. Additionally, it provides insights into the artistic

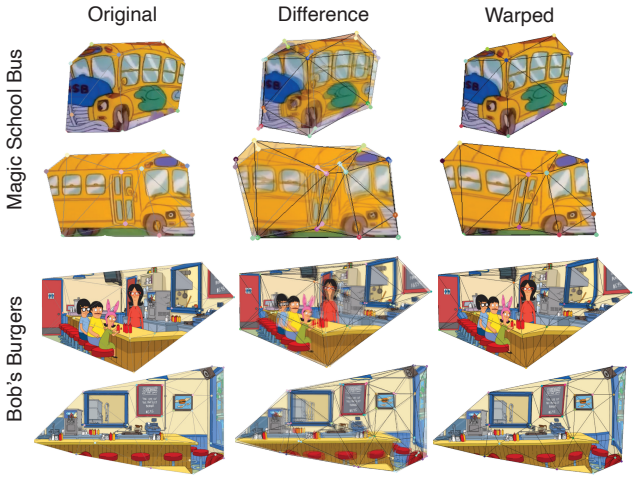

Figure 9. **Visualizing inconsistencies.** We overlay the original (left) and deformed image (right) to form a difference image (middle). Blurry regions indicate where strong warping occurs.

techniques used to convey 3D or to emphasize regions in drawings without strictly adhering to physical laws.

6. Conclusion

We present Toon3D, a pipeline for 3D reconstruction from geometrically inconsistent images. Toon3D leverages labeled correspondences and predicted depth priors, explaining inconsistencies via image deformations regularized to satisfy a shared perspective camera model. Many exciting future directions remain, such as incorporating generative priors or data-driven methods to reconstruct cartoons end-to-end. Our recovered cameras and point clouds could even be useful to train supervised methods like DUSt3R. We encourage ethical and responsible use of our work.

Acknowledgements. This project is supported in part by IARPA DOI/IBC 140D0423C0035. The views and conclusions herein are those of the authors and do not represent the official policies or endorsements of IARPA, DOI/IBC, or the U.S. Government. We thank Qianqian Wang, Justin Kerr, Brent Yi, David McAllister, Matthew Tancik, Evonne Ng, Anjali Thakrar, Christian Foley, Abhishek Kar, Georgios Pavlakos, the Nerfstudio team, the KAIR lab, Ian Mitchell and Roland Jose.

References

[1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building rome in a day. In *2009 IEEE 12th International Conference on Computer Vision*, pages 72–79, 2009. 2

[2] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021. 3

[3] M. Aubry, B. Russell, and J. Sivic. Painting-to-3D model alignment via discriminative visual elements. *ACM Transactions on Graphics*, 2013. 3

[4] Berta Bescos, José M Fácil, Javier Civera, and José Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, 2018. 2

[5] Shuhong Chen, Kevin Zhang, Yichun Shi, Heng Wang, Yiheng Zhu, Guoxian Song, Sizhe An, Janus Kristjansson, Xiao Yang, and Matthias Zwicker. Panic-3d: Stylized single-view 3d reconstruction from portraits of anime characters. In *CVPR*, 2023. 3

[6] Antonio Criminisi, Ian Reid, and Andrew Zisserman. Single view metrology. *IJCV*, 2000. 3

[7] Elliot J Crowley and Andrew Zisserman. In search of art. In *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part I 13*, pages 54–70. Springer, 2015. 8

[8] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *SIGGRAPH'96*. 1996. 2

[9] Johanna Delanoy, Mathieu Aubry, Phillip Isola, Alexei A Efros, and Adrien Bousseau. 3d sketching using multi-view deep volumetric prediction. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 1(1):1–22, 2018. 3

[10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 2, 3, 7

[11] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint detection and description of local features. *arXiv preprint arXiv:1905.03561*, 2019. 2, 7

[12] Benoit Guillard, Edoardo Remelli, Pierre Yvernay, and Pascal Fua. Sketch2mesh: Reconstructing and editing 3d shapes from sketches. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13023–13032, 2021. 3

[13] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2

[14] Aaron Hertzmann. Toward a theory of perspective perception in pictures. *Journal of Vision*, 24(4):23–23, 2024. 2

[15] Derek Hoiem, Alexei A Efros, and Martial Hebert. Automatic photo pop-up. In *ACM SIGGRAPH 2005 Papers*, pages 577–584. 2005. 3

[16] Youichi Horry, Ken-Ichi Anjyo, and Kiyoshi Arai. Tour into the picture: using a spidery mesh interface to make animation from a single image. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 225–232, 1997. 2

[17] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 709–727. Springer, 2020. 3

[18] Eakta Jain, Yaser Sheikh, Moshe Mahler, and Jessica Hodgins. Three-dimensional proxies for hand-drawn characters. *ACM Transactions on Graphics (ToG)*, 31(1):1–16, 2012. 3

[19] Angjoo Kanazawa, Shahar Kovalsky, Ronen Basri, and David Jacobs. Learning 3d deformation of animals from 2d images. In *Computer Graphics Forum*, 2016. 2

[20] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018. 2

[21] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation, 2023. 2, 3, 6

[22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 2, 4, 6

[23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 3

[24] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 7

[25] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 2

[26] Georgios Pavlakos*, Ethan Weber*, , Matthew Tancik, and Angjoo Kanazawa. The one where they reconstructed 3d humans and environments in tv shows. In *ECCV*, 2022. 3

[27] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 2

[28] Amit Raj, Srinivas Kaza, Ben Poole, Michael Niemeyer, Nataniel Ruiz, Ben Mildenhall, Shiran Zada, Kfir Aberman, Michael Rubinstein, Jonathan Barron, et al. Dreambooth3d: Subject-driven text-to-3d generation. *arXiv preprint arXiv:2303.13508*, 2023. 8

[29] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 2, 3

[30] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[31] Harrison Jesse Smith, Qingyuan Zheng, Yifei Li, Somya Jain, and Jessica K. Hodgins. A method for animating children's drawings of the human figure. *ACM Trans. Graph.*, 42(3), 2023. 3

[32] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pages 835–846. 2006. 2

[33] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, pages 109–116. Citeseer, 2007. 2

[34] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 2, 3

[35] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023. 7

[36] Javier Tirado-Garín, Frederik Warburg, and Javier Civera. Dac: Detector-agnostic spatial covariances for deep local features. *arXiv preprint arXiv:2305.12250*, 2023. 2

[37] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12959–12970, 2021. 2

[38] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *Advances in Neural Information Processing Systems*, 33:14254–14265, 2020. 7

[39] Andrea Vallone, Frederik Warburg, Hans Hansen, Søren Hauberg, and Javier Civera. Danish airs and grounds: A dataset for aerial-to-street-level place recognition and localization. *IEEE Robotics and Automation Letters*, 7(4):9207–9214, 2022. 2

[40] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. *arXiv preprint arXiv:2303.16196*, 2023. 6

[41] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. *arXiv preprint arXiv:2410.19115*, 2024. 2, 3, 6

[42] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 2, 6, 7

[43] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. *arXiv preprint arXiv:2312.02981*, 2023. 7

[44] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. 3, 6

[45] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2878–2890, 2012. 7

[46] Chao Zhou, Hong Zhang, Xiaoyong Shen, and Jiaya Jia. Unsupervised learning of stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1567–1575, 2017. 6

[47] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015. 3