GeoFill: Reference-Based Image Inpainting with Better Geometric Understanding

Yunhan Zhao[1]*, Connelly Barnes[2], Yuqian Zhou[2,3], Eli Shechtman[2], Sohrab Amirghodsi[2], Charless Fowlkes[1]

[1]UC Irvine   [2]Adobe Research   [3]IFP, UIUC

{yunhaz5, fowlkes}@ics.uci.edu   {cobarnes, elishe, tamirgho}@adobe.com   yuqian2@illinois.edu

Abstract

Reference-guided image inpainting restores image pixels by leveraging the content from another single reference image. The primary challenge is how to precisely place the pixels from the reference image into the hole region. Therefore, understanding the 3D geometry that relates pixels between two views is a crucial step towards building a better model. Given the complexity of handling various types of reference images, we focus on the scenario where the images are captured by freely moving the same camera around. Compared to the previous work, we propose a principled approach that does not make heuristic assumptions about the planarity of the scene. We leverage a monocular depth estimate and predict relative pose between cameras, then align the reference image to the target by a differentiable 3D reprojection and a joint optimization of relative pose and depth map scale and offset. Our approach achieves state-of-the-art performance on both RealEstate10K and MannequinChallenge dataset with large baselines, complex geometry and extreme camera motions. We experimentally verify our approach is also better at handling large holes.

1. Introduction

Image inpainting aims to plausibly restore missing pixels within a given hole region. Existing single image inpainting models [64, 58, 65] solve this problem without additional information by leveraging the knowledge learned from large scale training data or existing patches within the image. These methods become less reliable when input images contains large holes and the filling regions are complex in structures and textures.

In 2021, Zhou *et al.* [79] proposed a novel inpainting task called reference-based image inpainting. It aims at filling the hole regions of a "target" image using another single source photo taken of the same scene. This foreground re-

Figure 1: Given a reference image and a target image with hole, GeoFill utilizes predicted correspondence matching and depth maps to estimate a 3D mesh and relative camera pose and intrinsics. Compared to TransFill, the previous state-of-the-art approach, GeoFill handles complex scenes better by iteratively refining predicted depth maps and relative pose.

moval application is particularly useful when people take photos at museums or famous landmarks where the background is unique. It is nearly impossible for existing single image inpainting to faithfully restore what was *actually there in the background* for such cases. Inpainting with a reference image is appealing and indispensable, and also feasible, since other photos taken at different viewpoints of the same scene can be available, e.g. in the users' albums or even downloaded from the Internet.

However, reference-based image inpainting is very challenging and not well explored. That is because reference-based inpainting is a photography-oriented task that has only one single reference frame as "source", and has usually large baseline or challenging camera movements between the target and source. Therefore, the target and source images cannot be easily aligned to fill in the hole due to issues such as parallax. To address the alignment problems, the previous state-of-the-art method TransFill [79] has a strong assumption that the scenes can be blended by multiple planar structures. It clusters the predicted depth on matched feature points and utilizes multiple homographies to fill the hole. However, real scenes rarely consist of only a few planar surfaces, and even when they do, it can be hard to iden-

---

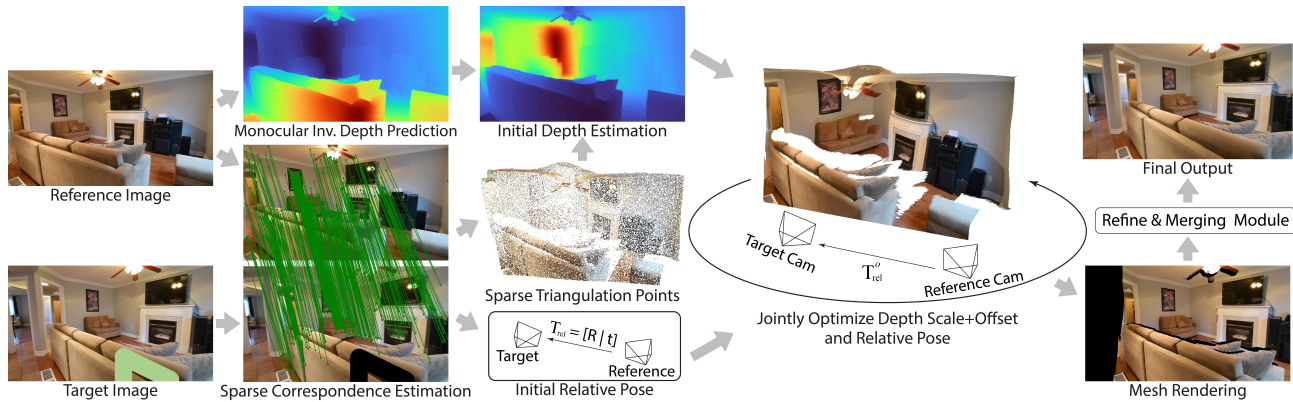*Work done while an intern at Adobe.

Figure 2: Overview of our system pipeline. We are given a target image with a hole and a source image. We aim at warping the single-source image to the target to fill the hole. We first estimate the relative pose as well as predict the monocular depth of the source, and then adjust the scale and offset of the depth map. After that, to mitigate the potential errors caused by deep models, we jointly optimize camera relative pose and depth map scale and offset to make the depth map and image contents well-align near the hole region. Finally, we render the reprojected source and refine it using post-processing.

tify the relevant planar surfaces. This indicates that to better solve the problem of reference-based inpainting, *it is crucial to understand the camera positions and geometry of the 3D scene, especially near the hole region* in order to find the appropriate content for hole filling.

In this work, we propose a more principled approach that fills the hole region by explicitly estimating the 3D scene structure from two limited views. Specifically, we first estimate sparse feature correspondences, from which we derive an initial relative pose between the two views. We then predict a monocular dense depth map of the source image and determine a scale and offset that aligns the depth map with the target using a sparse 3D triangulated point cloud. To mitigate the prediction errors and improve alignment accuracy, we next jointly optimize the depth scale and offset and the relative pose using a fast differentiable 3D reprojection of the source image to the target. We synthesize a warped source image by rendering a textured source mesh with the optimized depth and target pose, and fill dis-occluded regions with single image inpainting. Lastly, we adjust the exposure, white balance, lighting, and correct any residual misalignments before pasting the result back into the hole region.

In summary, our GeoFill is the first to apply a more principled approach for reference-based inpainting, *i.e.*, the first to leverage an explicit non-planar 3D scene representation given only limited two-view RGB images (no camera pose information). Compared with the previous state-of-the-art, GeoFill better handles complex 3D scene structures within the hole, wide-baseline image pairs and larger holes. Extensive experiments demonstrate that our methods achieve the best perceptual quality on various scenes in benchmarks and real user cases.

2. Related Work

Image inpainting. Traditional image inpainting models rely on hand-crafted heuristics. Diffusion-based approaches [6] propagate pixel colors from the background to the hole region. These approaches generate artifacts when the hole size is large or texture variations are significant. Alternatively, patch-based approaches [56, 3] search for similar patches outside the hole region to complete the missing regions. Although these approaches offer high quality texture by copying texture patches, the filled regions may be inconsistent with regions around the hole due to lack of high-level structural understanding of the entire image.

Recent deep models fill the hole by learning from large amounts of training data. Context encoders [41] generate semantically plausible content in the hole by encoding the surroundings. Iizuka et al. [25] adopt two discriminators to ensure the inpainted content is both locally and globally consistent. Artifacts can be reduced along the hole boundary by filtering using partial [31] or gated [65] convolutions. Some recent inpainting models improve the generated image quality with additional information, such as edges [38], segmentation masks [50], and low frequency structures [46, 30]. Moreover, several papers show deep neural networks can fill holes on high resolution images [61, 63, 67]. Despite the significant advancements in single inpainting models, filling with one single image remains fundamentally an ill-posed problem [75]. Image inpainting with additional information is also explored in the literature, such as inpainting with stereo images [54, 7, 2, 37, 36] and utilizing more than one image [51]. TransFill [79] is closely related to our work: it performs reference-guided inpainting by warping a reference image with multiple homographies. However, due to the planar nature of homographies, TransFill has a limited ability to handle image pairs with complex 3D struc-

tures, wide baselines, or significant disocclusions.

Video Inpainting. Classical works mainly focus on globally optimizing patch-based energies [42, 57, 18]. Recent work often adopts deep generative models for better inpainting performance. Wang *et al.* introduce a data-driven framework that jointly learns temporal structure and spatial details [52]. Onion-Peel Network (OPN) proposes to fill in the missing region progressively with the spatio-temporal attention [39]. Spatial-Temporal Transformer Network (STTN) adopts a deep generative model with adversarial training along the spatial-temporal dimension to mitigate blurriness and temporal artifacts [66]. Note that video inpainting approaches heavily exploit the dense temporal information in the video while we only have one single reference image which is a much harder scenario.

Two-view Geometry. SfM establishes correspondences between two monocular frames and subsequently estimates 3D structure [20, 49, 77, 53, 35]. In classic geometric vision, it is well understood that the camera poses as well as depth for corresponding points can be computed from feature matching points alone [32, 22]. Traditional methods utilize hand-crafted descriptors [33, 5, 47] to build sparse correspondence for the subsequent fundamental matrix estimation with the 8-point algorithm [21]. Learned local features have shown great success in recent works [62, 12, 13] together with the learning-based feature matching models, such as SuperGlue [48] or differentiable formations of RANSAC [8, 44, 9]. Another alternative is to directly estimate relative pose using an end-to-end pose estimation network [26]. We leverage these recent advances (specifically OANet [68]). Our method uses components inspired by SfM, however, our differentiable joint optimization stage is novel and has been carefully formulated for our task.

Monocular Depth Estimation. Predicting depth from a single image is an ill-posed problem. However, learning based approaches have shown impressive performance by either treating monocular depth estimation as a regression or classification task [15, 27, 59, 19, 60, 17, 23, 1, 28, 74, 24, 73]. Recent advances include BTS [28], which introduces local planar guidance layers to guide the features to full resolution instead of standard upsampling layers during the decoding phase. DAV [24] proposes to exploit the coplanarity of objects in the scene via depth-attention volume. DPT [43] leverages the high quality intermediate representation from transformers and become state-of-the-art.

3. Method

Suppose we are given a target image $\mathbf{I}_t$ with a hole $\mathbf{M}$ to be filled, and a reference (source) image $\mathbf{I}_s$ of the same scene. Our goal is to find a 3D-aware warped source image $\mathbf{I}_{s \to t}$ that geometrically aligns the source image to the target image which can be used to fill the hole. The final composite image can be represented as $\mathbf{I}_t^{\text{comp}} = \mathbf{I}_t \odot \mathbf{M} +$

$(\mathbf{M}_{\text{single}}\mathbf{I}_{s \to t} + (1 - \mathbf{M}_{\text{single}}) \odot \mathbf{I}_{\text{single}}) \odot (1 - \mathbf{M})$, where $\mathbf{M}_{\text{single}}$ is the blending map to merge the warped source with the single image inpainting result $\mathbf{I}_{\text{single}}$. Note that ideally there should be enough contents that GeoFill can copy from the source image to the target hole region, *i.e.* the source image I_s is useful. Cases with very few target-source content pixels overlapping inside the hole will cause GeoFill to fall back to the single image inpainting $\mathbf{I}_{\text{single}}$.

To compute the final warping matrix and use it to reproject the source image, as shown in Figure 2, we propose a pipeline consisting of three stages named initialization, joint optimization, and rendering and post-processing. In the first stage, we establish sparse correspondences between $\mathbf{I}_s$ and $\mathbf{I}_t$ and estimate the relative pose $\mathbf{T}_{\text{rel}}$ between two views. Meanwhile, we obtain a dense depth map of the source image using a pretrained deep model. Then we align the scale and offset of the predicted depth map with the sparse 3D triangulated feature points. In the second stage, we mitigate the potential errors of the initial guess of pose and depth by optimizing the related parameters so contents well-align near the hole. Finally, we render the warped image using the optimized parameters, and post-process it to address any residual spatial and color misalignments like TransFill. We will introduce each stage in the following sections.

3.1. Initialization Stage

Initialize Relative Pose Our approach first estimates the relative pose $\mathbf{T}_{\text{rel}}$ based on predicted sparse correspondences. We extract sparse corresponding feature points $\mathbf{P}_t$ and $\mathbf{P}_s$ between the target and source images, and compute the fundamental matrix $\mathbf{F}$ between $\mathbf{I}_s$ and $\mathbf{I}_t$ via the normalized 8-point algorithm [21] using RANSAC [16]. From $\mathbf{F}$, we derive the relative pose $\mathbf{T}_{\text{rel}}$ using the classic multi-view geometry algorithm mentioned in [21].

Initialize Dense Depth Map We then predict the inverse depth map on the source image using a pretrained monocular depth estimator as the cues to estimate the real source depth. Mathematically, we only need the source depth and $\mathbf{T}_{\text{rel}}$ to compute the 3D-aware warp $\mathbf{I}_{s \to t}$. However, the source depth is predicted up to an unknown scale and offset by a pretrained deep model, therefore, we need to solve for these such that the initial source depth $\mathbf{D}_s^i$ best matches the estimated relative pose. Note the estimated translation in $\mathbf{T}_{\text{rel}}$ is normalized and will not match the arbitrary scale in the predicted depth maps from pretrained deep models. As suggested in [72], aligning dense depth to sparse triangulated points is much simpler than rescaling the relative pose. Therefore, we first triangulate points with the relative pose, then align the scale of depth predictions with triangulation to subsequently match the scale of the relative pose.

Specifically, a 3D triangulated point $\mathbf{x}$ with point $q_s \in$

$\mathbf{P}_s$ and $q_t \in \mathbf{P}_t$ is computed as,

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}}[E(\mathbf{r}_s, \mathbf{x})]^2 + [E(\mathbf{r}_t, \mathbf{x})]^2, \qquad (1)$$

where $\mathbf{r}_s$ represents the ray shooting from the source camera center through the point q_s on the image plane, $\mathbf{r}_t$ is the ray from the target camera following a similar analogy, and E measures the Euclidean distance between two inputs. In this way, we compute a set of 3D triangulated points $\mathbf{X}$ using all matching sparse correspondences. In order to form the linear problem to compute the scale and offset, we first compute a sparse triangulated depth map $\mathbf{D}_{\mathrm{tri}}$ by projecting 3D triangulated points to source camera coordinates. Note $\mathbf{D}_{\mathrm{tri}}$ is in the same scale as the relative pose. Therefore, we correct $\mathbf{D}_s$ to match $\mathbf{D}_{\mathrm{tri}}$, which subsequently matches the scale of the relative pose. We correct $\mathbf{D}_s$ by estimating two scalars, the scale s^i and offset b^i associated with the depth map, by solving a linear least square problem. The initial depth maps are then expressed as: $\mathbf{D}_s^i = s^i \mathbf{D}_s + b^i$.

3.2. Joint Optimization Stage

To mitigate the effects of potential errors in sparse correspondence and depth estimation since deep models can be not robust or generalized enough, we further introduce an optimization module to improve the quality of $\mathbf{I}_{s \to t}$. We optimize the depth scale, offset, and the relative pose that jointly define $\mathbf{I}_{s \to t}$ in the 3D scene. Specifically, we convert the rotation matrix into quaternions, which leads to a total of 9 parameters to optimize. Both relative pose and the initial depth computed in the previous section are used as the initial guess for the optimization. Our optimization contains 3 different loss functions: a multiscale photometric loss $\mathcal{L}_{\mathrm{photo}}$, a feature correspondence loss $\mathcal{L}_{\mathrm{feat}}$, and a negative depth penalty $\mathcal{L}_{\mathrm{negD}}$.

Multiscale Photometric Loss $\mathcal{L}_{\mathrm{photo}}$ measures the pixel-level color difference between $\mathbf{I}_{s \to t}$ and the $\mathbf{I}_t$ outside the hole region. We downsample both $\mathbf{I}_{s \to t}$ and $\mathbf{I}_t$ and sum the normalized color difference across different resolutions. Specifically, we build Gaussian pyramids on both $\mathbf{I}_{s \to t}$ and $\mathbf{I}_t$ using an RGB representation for the source image and an alpha-premultiplied RGBA representation on the target image to incorporate the hole region properly into the target image. Computing a multi-scale photometric loss within each iteration is obviously computationally expensive. Moreover, the optimization might also get trapped into the local minima associated with the finest resolution due to poor initialization [80]. To accelerate the computation speed and find better solutions, we adopt a coarse-to-fine optimization strategy, which means we first compute photometric loss on the most coarse level and move to the finer level once the convergence criteria at the current pyramid level are met. Additionally, instead of building a 3D triangle mesh and rendering from the target view at each iteration,

we use a much more efficient differentiable 3D reprojection to find a warping field that computes $\mathbf{I}_{s \to t}$ from $\mathbf{I}_s$ with bilinear interpolation. Mathematically, we have:

$$\mathbf{I}_{s \to t} = \mathrm{bilinear}(\mathbf{I}_s, \mathrm{reproj}(\mathbf{K}, \mathbf{T}_{\mathrm{rel}}, \mathbf{D}_s^o)), \qquad (2)$$

where $\mathrm{reproj}()$ represents the reprojection operation. The photometric loss at a given resolution with the pyramid is:

$$\mathcal{L}_{\mathrm{photo}} = \frac{1}{|\mathbf{M}|} \sum \mathbf{W} \odot ||\mathbf{I}_{s \to t} \odot \mathbf{M} - I_t \odot \mathbf{M}||^2. \quad (3)$$

Here $\mathbf{W}$ is a pixel importance weight map discussed shortly.

Feature Correspondence Loss $\mathcal{L}_{\mathrm{feat}}$ computes the distance between reprojected matching feature points in the source images and the target images. We use the $\mathrm{reproj}()$ operator on all 2D image coordinates in $\mathbf{P}_s$ to get another set $\mathbf{P}_{s \to t}$. Then, we compute the average distance between $\mathbf{P}_{s \to t}$ and $\mathbf{P}_t$. However, the average distance of all points is very sensitive to outliers, i.e., very few outliers dominate the loss function. To reduce the effects of the outliers on the loss function, we adopt the general robust loss function from [4]. The general form of the loss function is:

$$f(x, \alpha, c) = \frac{|\alpha - 2|}{\alpha} \left(\left(\frac{(x/c)^2}{|\alpha - 2|} + 1 \right)^{\alpha/2} - 1 \right), \quad (4)$$

where α and c are the shape and scale parameters, respectively. In our experiments, we set $\alpha = -2$ and $c = 10$. Then, feature correspondence loss is written as:

$$\mathcal{L}_{\mathrm{feat}} = \frac{1}{|P_{s \to t}|} \sum_{m=0}^{|P_{s \to t}|} \mathbf{W}(\mathbf{q}_t^m) f(||\mathbf{q}_{s \to t}^m - \mathbf{q}_t^m||, \alpha, c), \quad (5)$$

where f is the general robust loss function, $\mathbf{q}_{s \to t}^m$ and $\mathbf{q}_t^m$ is the mth point in $P_{s \to t}$ and P_t, respectively.

Negative Depth Penalty $\mathcal{L}_{\mathrm{negD}}$ aims to penalize negative values in the remapped depth. Although depth predictions have arbitrary scale and offset, they should have all positive values, meaning that a fragment of geometry associated with a pixel should never move behind the camera. Mathematically, we adopt a hinge loss function:

$$\mathcal{L}_{\mathrm{negD}} = \sum \max\{0, -\mathbf{D}_s^o\}. \qquad (6)$$

Our final objective function is written as: $l = \lambda_1 \mathcal{L}_{\mathrm{photo}} + \lambda_2 \mathcal{L}_{\mathrm{feat}} + \lambda_3 \mathcal{L}_{\mathrm{negD}}$, where $\{\lambda_j\}$ are weights. The convergence criteria is discussed in the supplemental material.

Pixel Importance Weight Map W. When computing the photometric loss or feature correspondence loss, we assign weights to each pixel to replace uniform weighting. It encourages our optimization to better align the warped source with the target image both locally and globally.

The first type of weighting strategy is hole-distance weighting $\mathbf{W}_h$. We encourage the optimization to focus

on the regions close to the hole boundary since those pixels are more important for filling the hole regions. To achieve this, we apply the distance transform to the hole image and obtain a distance map $\mathbf{M}_h$, where each pixel records the Euclidean distance to the closest boundary pixel of the hole $\mathbf{M}_h$. We compute the weight at each pixel using a Gaussian function $\mathbf{W}_h = \exp(-\mathbf{M}_h^2/2\sigma^2)$, where σ is a hyperparameter that adjusts how weights change w.r.t distance to the hole boundary.

The second type is the edge-based weighting $\mathbf{W}_e$. This is because high-gradient edge regions are more salient while checking the alignment quality so we intend to give strong edges larger weights. We compute a multi-scale Canny edge map by first applying Gaussian blur on $\mathbf{I}_t$ with N different kernel sizes, running the Canny edge detector [10], and dilating each edge to get $\{e_1, e_2, \cdots, e_N\}$. Our pixel-level edge-based weight map becomes $\mathbf{W}_e = \sum_{k=1}^{N} e_k / \sum_p e_k(p)$, where the inner sum is over spatial coordinates. Our overall weighting map is $\mathbf{W} = \mathbf{W}_h \odot \mathbf{W}_e$.

3.3. Rendering and Postprocessing Stage

Mesh Rendering. After optimization, we find $\mathbf{T}_{\text{rel}}^o$, s^o, and b^o, the camera relative pose and depth map scale and offset, respectively. However, computing $\mathbf{I}_{s \to t}$ using depth reprojection always has some gaps between valid pixels due to it relying on a forward warping for efficiency in the inner loop of the optimization. One way to overcome this problem is interpolation but that has two disadvantages. First, the reprojected image is sparse and for regions with multiple layers of depth, pixels for a far-away depth layer might splat between pixels of a closer depth layer, which could result in interpolations that do not fully remove hidden surfaces. Additionally, interpolation cannot distinguish between holes due to disoccluded regions or simple gaps between the pixels due to the forward warping.

To address the above problems, we choose to render a textured mesh to get the final $\mathbf{I}_{s \to t}$. We first build a triangle mesh with a regular grid from the source view. The mesh vertices are computed by projecting the optimized depth $\mathbf{D}_s^o$ to the 3D space and the texture is the RGB colors of the source image. After obtaining the mesh, we drop the edges around depth discontinuities. We adopt a simplified version of the footprint algorithm [55] by comparing the depth values between connected vertices. We drop the edge between two vertices v_i and v_j if $\frac{2|d(v_i)-d(v_j)|}{d(v_i)+d(v_j)} > \epsilon_{\text{edge}}$, where $d(v_i)$ is the depth value of the vertex v_i and ϵ_{edge} is a predefined threshold. After building the triangle mesh, we render the target view with $\mathbf{T}_{\text{rel}}^o$. Note that we also normalize the mesh to the unit size before rendering. The textured mesh densely fills in pixels and removes hidden surfaces. It also allows us to use the rendered alpha channel to find pixels where there is no ray intersection with the mesh, which represent disoccluded regions near depth discontinuities or regions outside

the photo: these are later filled by single image inpainting. We use PyTorch3D [45] as our renderer.

Refinement and Merging. With the rendered image, we apply the color-spatial transformation (CST) module from TransFill to further improve any small residual spatial misalignments and correct color and exposure differences. Lastly, we merge the output from CST with results from the single image inpainting model as in TransFill to handle the disocclusions and regions outside the photo.

4. Experiments

Datasets. No large-scale image-based dataset for reference-based inpainting is available. So we follow Trans-Fill to randomly sample multiple image pairs from video-based datasets because it is easier to simulate user behaviours and analyze the target-source difference (or camera view changes). During evaluation we only use one single reference frame. This is different from what video inpainting works did. We follow DeepFillv2 [65] to generate random free-form brush stroke masks and evaluate GeoFill and other baselines on the following datasets.

RealEstate10K [78]: It contains a diverse collection of YouTube video sequences shot from a moving camera for both indoor and outdoor scenes. Each video clip contains a variety of views of the same scene. We randomly sample 500 videos and select one pair of images in each video sequence with a specific frame difference (FD). Specifically, we sample FD=25, 50, and 75 to build three different sets with a resolution of 720×1280 while automatically filtering out image pairs without enough overlapping content inside the hole by checking the number of matching sparse features. Note that our filtering mechanism is only for the purpose of simulating user behaviors and removing non-useful image pairs. In practice, we believe users could simply check overlap between photos by visual inspection.

MannequinChallenge [29]: This is a challenging dataset with video sequences shot from a hand-held camera freely roaming in a scene of people with frozen poses. This dataset contains more than 170K frames and corresponding camera poses derived from about 2k YouTube videos. The camera motion in this dataset is more extreme and scene complexity is much higher due to diverse frozen human poses and rich background objects. We reduce the sampling FD to ensure enough overlap between image pairs. Similar to the previous dataset, we randomly sample 3 subsets with FD=10, 20, 30, where each contains 500 image pairs of 720×1280.

Real User-provided Images: We also evaluate our approach on the real user-provided images like TransFill to validate the generalization ability and practicability.

Baselines. In addition to TransFill, which is directly related to our work, we compare our approach against several different types of baselines to evaluate final inpainting performance. The first type is state-of-the-art video

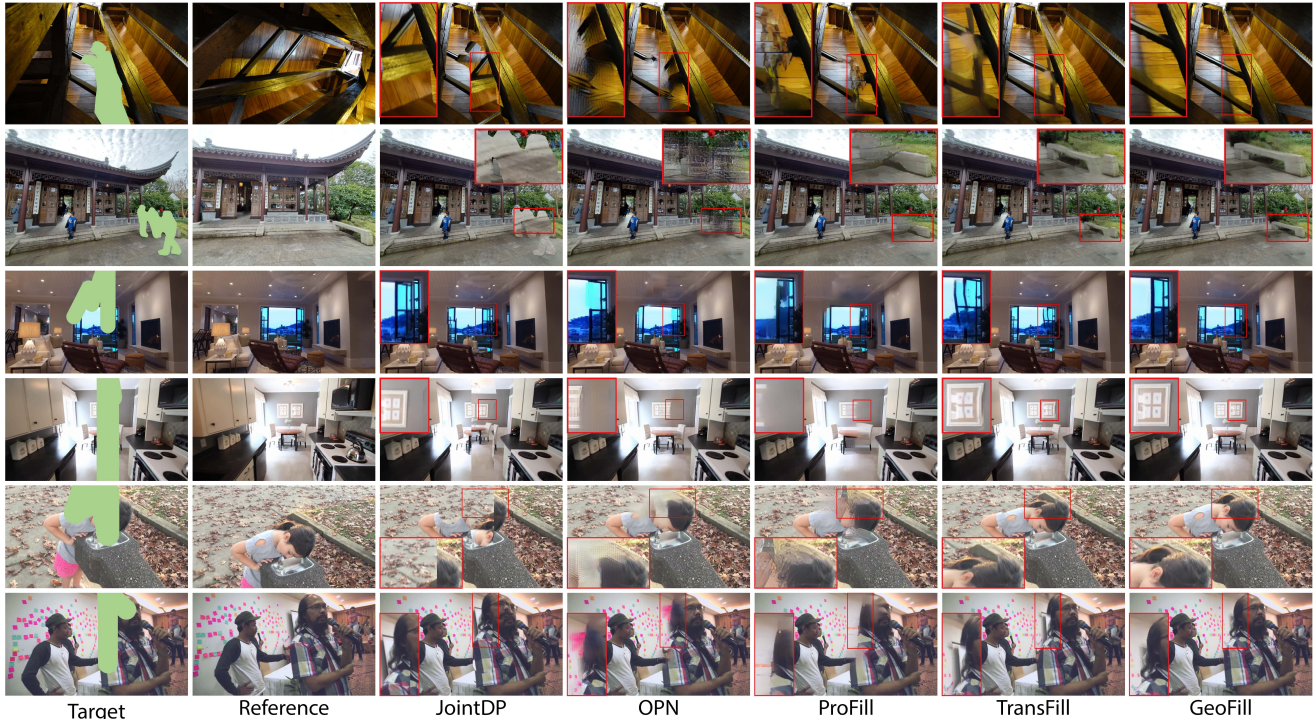| Target | Reference | JointDP | OPN | ProFill | TransFill | GeoFill |

Figure 3: Qualitative comparison of GeoFill against other baselines on user-provided images (top 2 rows), RealEstate10K (mid 2 rows), and MannequinChallenge dataset (last two rows).

completion models. OPN [39] achieves high quality inpainting results with spatio-temporal attention. STTN [66] proposes to optimize a spatial-temporal adversarial loss function. In addition, we compare against state-of-the-art single-image inpainting methods including ProFill [67] and CoModGAN [71]. We use their off-the-shelf pretrained weights directly since they are both trained on more diverse scenes in Places2 [76] than RealEstate10K and have strong generalization ability. Lastly, we compare a two-view SfM based approach [72] — which we refer to as JointDP — by warping the source image with the jointly estimated relative pose and depth using dense correspondence. To ensure fairness in the comparison, we use the same depth predictor and rendering process as in GeoFill while keeping all other settings the same as in the original work.

Implementation Details. We follow TransFill to extract SIFT features [34] and feed them into OANet [68] to reject outliers and establish correspondences. This combination was already proven to be quite robust. A better matching strategy could always be adopted to further improve results while leaving the rest of our framework intact. Our pretrained monocular depth predictor is DPT [43]. No groundtruth camera intrinsic information is required to run our approach. We use fixed camera intrinsic parameters for all images by setting focal length to 750 and principal point to the image center. An ablation study with different focal lengths is included in the supplemental material. We use the pretrained CST module from TransFill [79] without

finetuning. This module generalizes well to MannequinChallenge and user provided images. Our pipeline is implemented with PyTorch [40] and we choose DiffGrad [14] as our optimizer due to its fast convergence speed. In the optimization step, we use a constant learning rate 10^{-2} and the maximum number of iteration is set to 10^4. The loss weights λ_1, λ_2, and λ_3 are 10, 10, 0.5, respectively. In the coarse-to-fine optimization strategy, the number of pyramid levels is 4 and the maximum number of cumulative iterations at each level from coarse to fine are 4×10^3, 7×10^3, 9×10^3, 10^4. We set the σ in hole-based weighting to 192 pixels. In edge-based weight, we compute 4 different Canny edge maps and dilate each of them with a kernel size equal to 4. In mesh rendering, the edge threshold ϵ_{edge} is 4×10^{-2}.

4.1. Quantitative Results

The quantitative results comparing our approach with other baselines are shown in Table 1. We report the PSNR, SSIM and LPIPS[70] on the RealEstate10K and the MannequinChallenge datasets. Single image inpainting models are not competitive enough for image pairs with larger scale differences and wider baselines. Video inpainting approaches also show bad performance due to the lack of dense temporal information and multiple reference frames. JointDP is based on optical flow and is not able to accurately estimate parameters like the camera pose needed to correctly align the image pairs. Our method demonstrates superiority over TransFill because we have a better under-

Table 1: Quantitative comparisons of GeoFill against other baselines on both RealEstate10K and MannequinChallenge.

Model	RealEstate10K: PSNR ↑/ SSIM ↑ / LPIPS ↓			MannequinChallenge: PSNR ↑/ SSIM ↑ / LPIPS ↓		
	FD=25	FD=50	FD=75	FD=10	FD=20	FD=30
JointDP [72]	22.46 / 0.9469 / 0.1011	21.76 / 0.9457 / 0.1063	20.89 / 0.9423 / 0.1122	20.13 / 0.9346 / 0.1087	19.52 / 0.9290 / 0.1195	19.38 / 0.9315 / 0.1177
OPN [39]	28.41 / 0.9684 / 0.0525	27.80 / 0.9669 / 0.0570	26.91 / 0.9634 / 0.0624	25.63 / 0.9628 / 0.0605	24.92 / 0.9584 / 0.0698	24.84 / 0.9591 / 0.0702
STTN [66]	28.83 / 0.9696 / 0.0710	28.26 / 0.9697 / 0.0721	27.59 / 0.9680 / 0.0751	25.60 / 0.9623 / 0.0803	25.09 / 0.9602 / 0.0865	24.94 / 0.9613 / 0.0844
ProFill [67]	27.45 / 0.9642 / 0.0775	27.67 / 0.9654 / 0.0755	27.37 / 0.9639 / 0.0768	25.04 / 0.9589 / 0.0808	25.02 / 0.9582 / 0.0836	25.22 / 0.9599 / 0.0810
CoModGAN [71]	26.02 / 0.9594 / 0.0703	26.14 / 0.9607 / 0.0686	25.88 / 0.9596 / 0.0697	23.39 / 0.9504 / 0.0770	23.14 / 0.9486 / 0.0808	23.36 / 0.9503 / 0.0791
TransFill [79]	32.03 / 0.9764 / **0.0461**	30.64 / 0.9732 / 0.0540	29.24 / 0.9694 / 0.0608	28.01 / 0.9680 / 0.0569	26.56 / 0.9628 / 0.0688	26.17 / 0.9632 / 0.0701
GeoFill (Ours)	**32.57 / 0.9775 /** 0.0467	**31.47 / 0.9748 / 0.0525**	**30.43 / 0.9717 / 0.0581**	**28.85 / 0.9702 / 0.0553**	**27.72 / 0.9658 / 0.0652**	**27.44 / 0.9664 / 0.0665**

standing of the 3D structures of the scenes, and better leverage the depth estimation. Note that GeoFill has higher performance gain over TransFill on MannequinChallenge than on RealEstate10K dataset. TransFill relies on the fusion module to handle multiple homographies for the final hole filling and it is less robust on the images different from the training data. In contrast, GeoFill only has a single proposal to merge during inpainting, which also greatly reduces blending artifacts that arise from multiple proposals. Therefore, our GeoFill is robust and stable on image pairs with even larger frame differences.

4.2. Qualitative Results

Figure 3 shows the visual comparisons with other baseline algorithms on the user-provided images, the RealEstate10K and the MannequinChallenge dataset. JointDP utilizes estimated optical flow for the initial matching, thus the results fail to obtain accurate depth and camera pose if the baseline of the image pair is wide. The contents inside the hole are often misaligned with the target image. The original OPN uses five reference frames to achieve a more efficient non-local matching among frames, but a single reference frame makes the results less visually-pleasing. ProFill is not able to take advantage of the reference image contents, and TransFill usually has blending artifacts or content misalignment issues when objects inside the hole regions occupy multiple depth planes. However, GeoFill avoids the blending artifacts by using one single proposal, and aligns the objects well by understanding the camera poses and reconstructing the 3D scene from two images.

4.3. Ablation Study

In the following sections, we study how each optimization loss function and the pixel importance weight maps contribute to the final results. Additionally, we present experimental results of TransFill and GeoFill on cases with larger holes and their alignment accuracy without CST. All experiment results in the following section are reported for the RealEstate10K FD=50 subset unless specified. We also include more in-depth analyses of the joint depth pose optimization module on RealEstate10K and ScanNet [11], the timing of each module in our system pipeline, the convergence criteria of the optimization step, and more visualizations in the supplemental material.

Table 2: Ablations on the objective functions in the joint optimization stage of GeoFill.

$\mathcal{L}_{\text{photo}}$	$\mathcal{L}_{\text{feat}}$	$\mathcal{L}_{\text{negD}}$	PSNR↑	SSIM↑	LPIPS↓
✓	✓	✓	**31.47**	**0.9748**	**0.0525**
✗	✓	✓	31.19	0.9742	0.0533
✓	✗	✓	30.88	0.9734	0.0554
✓	✓	✗	31.23	0.9742	0.0532

Table 3: Ablations on the pixel importance weight map $\mathbf{W}$.

Hole $\mathbf{W}_h$	Edge $\mathbf{W}_e$	PSNR↑	SSIM↑	LPIPS↓
✓	✓	**31.47**	**0.9748**	**0.0525**
✓	✗	31.20	0.9740	0.0534
✗	✓	31.12	0.9739	0.0539
✗	✗	30.95	0.9734	0.0552

Optimization: Objective Functions. We claim that all the objective functions used in the optimization process contribute to a higher perceptual and reconstruction quality as shown in Table 2. Comparing GeoFill without photometric loss against without feature correspondence loss, the feature loss contributes to the performance the most. We find that photometric loss by itself may get distracted by local textures and fall into local minima, such that it achieves lower RGB errors on average but ignores the global structure. However, using the photometric loss does help improve alignments. Lastly, GeoFill without negative depth penalty still has a performance drop, suggesting negative depth penalty is able to prevent corner cases where depth scale or offset estimates are non-robust.

Optimization: Pixel Importance Weight Map. We present results of GeoFill with different combinations of pixel importance weight maps in the optimization in Table 3. This suggests that hole-distance weighting $\mathbf{W}_h$ which puts higher weights around the hole lead to better local alignment around the hole. Edge-based weighting $\mathbf{W}_e$ also helped by matching strong edges within the image. Using a uniform weighting map leads to the worst performance, indicating the effectiveness of the weighting maps.

Performance w.r.t Hole Size. This ablation study aims to examine the performance of our approach against TransFill under more difficult settings. As the hole becomes larger, inevitably we have fewer matching points, which makes $\mathbf{I}_{s \rightarrow t}$ hard to align with $\mathbf{I}_t$. We generate holes with different average stroke widths ranging from 90 to 210 pixels. As shown in Figure 4, GeoFill has an increasing performance gain over TransFill until the hole average stroke width reaches 180 pixels. As the hole size grows, the complexity of the content in the hole also increases. In other words, we are more likely to encounter a growing number
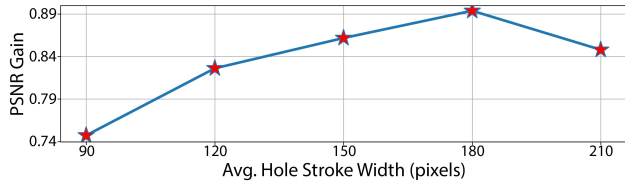
Figure 4: Performance gain of our method compared to TransFill w.r.t the average hole size. GeoFill has a greater advantage when the hole is larger.

Figure 5: Per-sample performance gain of ours compared to TransFill. The blue vertical line separates positive and negative PSNR gain.

Table 4: Ablations on Initial alignment comparisons of our method compared to TransFill without the CST Module.

Model	PSNR↑	SSIM↑	LPIPS↓
GeoFill	**31.47**	**0.9748**	**0.0525**
GeoFill (no CST)	29.35	0.9688	0.0579
TransFill	30.64	0.9732	0.0540
TransFill (no CST)	26.03	0.9598	0.0742

of depth layers, a more complicated objects layout, and a higher chance of occlusions due to camera translation in a larger hole. These problems are harder for homography-based models, therefore, GeoFill has a greater advantage when the hole is larger. The drop in performance gain at the end of the curve can be because there are not enough matching points for GeoFill to infer an accurate 3D structure.

Initial Alignment Comparisons against TransFill. We adopted the CST module from TransFill to adjust auto exposure, lighting conditions, and potential remaining misalignments. In this experiment, we analyze the performance of using the aligned images directly, e.g. without using CST for both models. We retain the multiple homographies used by TransFill, drop the CST module, and keep the merging module so TransFill can merge its different regions. Table 4 shows that GeoFill maintains better performance, dropping 2.12 in PSNR, while TransFill drops 4.62 without CST. This experimentally validates the proposal from GeoFill is much more accurate than TransFill by better leveraging the depth.

Per-Sample Improvement Study. This ablation is designed to break down the numbers shown in Table 1 by comparing the performance of GeoFill against TransFill over each individual sample. We sort and plot the per-image PSNR difference over TransFill, shown in Figure 5. Specifically, PSNR difference is computed as GeoFill PSNR − TransFill PSNR, therefore a positive PSNR gain indicates GeoFill is better. GeoFill improves performance on the majority of samples, specifically for around 75.4% of the entire subset. Moreover, the PSNR gains when GeoFill outperforms are much greater: up to 3 dB, as opposed to the PSNR losses when TransFill outperforms.

5. Discussion, Limitations, and Conclusion

Limitations and Future Work. GeoFill inpaints the hole with one image pair with no auxiliary pose or depth
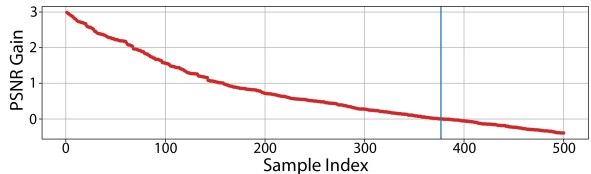
information from sensors, therefore, our pipeline may not work well when the quality of feature matching points is poor, e.g., matching points are inaccurate or too few. Under these cases, our relative pose and triangulated points can be inaccurate, which may be hard for optimization to correct. Additionally, our pipeline is also sensitive to depth prediction quality: artifacts such as blurry depth discontinuities or wrong order of depth planes can lead to potential bad inpainting results. Future work might mitigate these problems by jointly reasoning about monocular depth and the stereo cues established by triangulation. In the optimization, we used a 3D reprojection based on forward warping because it is much faster than rendering a triangle mesh even though it does not fully remove hidden surfaces in the rare cases where a mesh occludes itself: this could be addressed in future work by testing and pruning those splatted points. GeoFill utilizes the CST module to adjust auto exposure and lighting condition changes, which still suffers when the scene environment changes drastically, e.g., day to night, spring to fall. Future work could better address these by incorporating specialized lighting estimation (e.g., [69]) and relighting modules. One last limitation of GeoFill is that our pipeline only handles static scenes. Any dynamic objects, e.g., walking people or moving cars, not in the hole could lead to bad relative pose estimation and our optimization may cause misalignments under such cases. A simple solution would be masking out objects that are likely to move such as people and cars, but how to correctly identify all moving objects is still an open question.

Conclusion. In this work, we proposed a novel reference-based inpainting approach that achieves state-of-the-art performance. Our method relies on off-the-shelf depth estimation and sparse correspondence methods, and thus we can expect its performance to improve if these modules improve in performance. Compared to previous homography-based work, GeoFill is better at image pairs with large camera motions and scenes with complicated structures in the hole. GeoFill ensures accurate alignment between the filling content and the hole by jointly optimizing depth predictions and relative poses without explicitly knowing camera intrinsics. Moreover, our approach inpaints the hole in a more principled way by utilizing a single proposal without heuristic planar assumptions.

References

[1] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941*, 2018.

[2] Seung-Hwan Baek, Inchang Choi, and Min H Kim. Multiview image completion with space structure propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 488–496, 2016.

[3] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009.

[4] Jonathan T Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4331–4339, 2019.

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[6] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, 2000.

[7] Arnav V Bhavsar and Ambasamudram N Rajagopalan. Inpainting in multi-image stereo. In *Joint Pattern Recognition Symposium*, pages 172–181. Springer, 2010.

[8] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6684–6692, 2017.

[9] Eric Brachmann and Carsten Rother. Neural-guided ransac: Learning where to sample model hypotheses. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4322–4331, 2019.

[10] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[11] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[12] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Toward geometric deep slam. *arXiv preprint arXiv:1707.07410*, 2017.

[13] 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.

[14] Shiv Ram Dubey, Soumendu Chakraborty, Swalpa Kumar Roy, Snehasis Mukherjee, Satish Kumar Singh, and Bidyut Baran Chaudhuri. diffgrad: An optimization method for convolutional neural networks. *IEEE transactions on neural networks and learning systems*, 31(11):4500–4511, 2019.

[15] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *arXiv preprint arXiv:1406.2283*, 2014.

[16] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[17] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018.

[18] Miguel Granados, Kwang In Kim, James Tompkin, Jan Kautz, and Christian Theobalt. Background inpainting for videos with dynamic objects and a free-moving camera. In *European Conference on Computer Vision*, pages 682–695. Springer, 2012.

[19] Zhixiang Hao, Yu Li, Shaodi You, and Feng Lu. Detail preserving depth estimation from a single image using attention guided networks. In *2018 International Conference on 3D Vision (3DV)*, pages 304–313. IEEE, 2018.

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

[21] Richard I Hartley. In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 19(6):580–593, 1997.

[22] Richard I Hartley and Peter Sturm. Triangulation. *Computer vision and image understanding*, 68(2):146–157, 1997.

[23] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1043–1051. IEEE, 2019.

[24] Lam Huynh, Phong Nguyen-Ha, Jiri Matas, Esa Rahtu, and Janne Heikkilä. Guiding monocular depth estimation using depth-attention volume. In *European Conference on Computer Vision*, pages 581–597. Springer, 2020.

[25] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):1–14, 2017.

[26] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015.

[27] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE, 2016.

[28] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019.

[29] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4521–4530, 2019.

[30] Liang Liao, Jing Xiao, Zheng Wang, Chia-Wen Lin, and Shin'ichi Satoh. Guidance and evaluation: Semantic-aware image inpainting for mixed scenes. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 683–700. Springer, 2020.

[31] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018.

[32] H Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135, 1981.

[33] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.

[34] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[35] Haoyu Ma, Liangjian Chen, Deying Kong, Zhe Wang, Xingwei Liu, Hao Tang, Xiangyi Yan, Yusheng Xie, Shih-Yao Lin, and Xiaohui Xie. Transfusion: Cross-view fusion with transformer for 3d human pose estimation. *arXiv preprint arXiv:2110.09554*, 2021.

[36] Liqian Ma, Stamatios Georgoulis, Xu Jia, and Luc Van Gool. Fov-net: Field-of-view extrapolation using self-attention and uncertainty. *IEEE Robotics and Automation Letters*, 6(3):4321–4328, 2021.

[37] Wei Ma, Mana Zheng, Wenguang Ma, Shibiao Xu, and Xiaopeng Zhang. Learning across views for stereo image completion. *IET Computer Vision*, 14(7):482–492, 2020.

[38] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*, 2019.

[39] Seoung Wug Oh, Sungho Lee, Joon-Young Lee, and Seon Joo Kim. Onion-peel networks for deep video completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4403–4412, 2019.

[40] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

[41] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[42] Kedar A Patwardhan, Guillermo Sapiro, and Marcelo Bertalmio. Video inpainting of occluding and occluded objects. In *IEEE International Conference on Image Processing 2005*, volume 2, pages II–69. IEEE, 2005.

[43] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021.

[44] René Ranftl and Vladlen Koltun. Deep fundamental matrix estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 284–299, 2018.

[45] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020.

[46] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H Li, Shan Liu, and Ge Li. Structureflow: Image inpainting via structure-aware appearance flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 181–190, 2019.

[47] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[48] 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.

[49] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[50] Yuhang Song, Chao Yang, Yeji Shen, Peng Wang, Qin Huang, and C-C Jay Kuo. Spg-net: Segmentation prediction and guidance network for image inpainting. *arXiv preprint arXiv:1805.03356*, 2018.

[51] Theo Thonat, Eli Shechtman, Sylvain Paris, and George Drettakis. Multi-view inpainting for image-based scene editing and rendering. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 351–359. IEEE, 2016.

[52] Chuan Wang, Haibin Huang, Xiaoguang Han, and Jue Wang. Video inpainting by jointly learning temporal structure and spatial details. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5232–5239, 2019.

[53] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Stan Birchfield, Kaihao Zhang, Nikolai Smolyanskiy, and Hongdong Li. Deep two-view structure-from-motion revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8953–8962, 2021.

[54] Liang Wang, Hailin Jin, Ruigang Yang, and Minglun Gong. Stereoscopic inpainting: Joint color and depth completion from stereo images. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[55] Lee Westover. Footprint evaluation for volume rendering. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 367–376, 1990.

[56] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time video completion. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE, 2004.

[57] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *IEEE Transactions on pattern analysis and machine intelligence*, 29(3):463–476, 2007.

[58] Wei Xiong, Jiahui Yu, Zhe Lin, Jimei Yang, Xin Lu, Connelly Barnes, and Jiebo Luo. Foreground-aware image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5840–5848, 2019.

[59] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5354–5362, 2017.

[60] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3917–3925, 2018.

[61] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6721–6729, 2017.

[62] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *European conference on computer vision*, pages 467–483. Springer, 2016.

[63] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7508–7517, 2020.

[64] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018.

[65] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4471–4480, 2019.

[66] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *The Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[67] Yu Zeng, Zhe Lin, Jimei Yang, Jianming Zhang, Eli Shechtman, and Huchuan Lu. High-resolution image inpainting with iterative confidence feedback and guided upsampling. In *European Conference on Computer Vision*, pages 1–17. Springer, 2020.

[68] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5845–5854, 2019.

[69] Jinsong Zhang, Kalyan Sunkavalli, Yannick Hold-Geoffroy, Sunil Hadap, Jonathan Eisenman, and Jean-Francois Lalonde. All-weather deep outdoor lighting estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10158–10166, 2019.

[70] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[71] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. *arXiv preprint arXiv:2103.10428*, 2021.

[72] Wang Zhao, Shaohui Liu, Yezhi Shu, and Yong-Jin Liu. Towards better generalization: Joint depth-pose learning without posenet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9151–9161, 2020.

[73] Yunhan Zhao, Shu Kong, and Charless Fowlkes. Camera pose matters: Improving depth prediction by mitigating pose distribution bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15759–15768, 2021.

[74] Yunhan Zhao, Shu Kong, Daeyun Shin, and Charless Fowlkes. Domain decluttering: Simplifying images to mitigate synthetic-real domain shift and improve depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3330–3340, 2020.

[75] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1438–1447, 2019.

[76] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

[77] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017.

[78] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

[79] Yuqian Zhou, Connelly Barnes, Eli Shechtman, and Sohrab Amirghodsi. Transfill: Reference-guided image inpainting by merging multiple color and spatial transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2266–2276, 2021.

[80] Shizhan Zhu, Cheng Li, Chen Change Loy, and Xiaoou Tang. Face alignment by coarse-to-fine shape searching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4998–5006, 2015.