TransFill: Reference-guided Image Inpainting by Merging Multiple Color and Spatial Transformations

Yuqian Zhou\(^1\), Connelly Barnes\(^2\), Eli Shechtman\(^2\), Sohrab Amirghodsi\(^2\)
\(^1\)IFP, UIUC, \(^2\)Adobe Research

Abstract

Image inpainting is the task of plausibly restoring missing pixels within a hole region that is to be removed from a target image. Most existing technologies exploit patch similarities within the image, or leverage large-scale training data to fill the hole using learned semantic and texture information. However, due to the ill-posed nature of the inpainting task, such methods struggle to complete larger holes containing complicated scenes. In this paper, we propose TransFill, a multi-homography transformed fusion method to fill the hole by referring to another source image that shares scene contents with the target image. We first align the source image to the target image by estimating multiple homographies guided by different depth levels. We then learn to adjust the color and apply a pixel-level warping to each homography-warped source image to make it more consistent with the target. Finally, a pixel-level fusion module is learned to selectively merge the different proposals. Our method achieves state-of-the-art performance on pairs of images across a variety of wide baselines and color differences, and generalizes to user-provided image pairs.

1. Introduction

Image inpainting is an image restoration task where the goal is to fill in specific regions of the image while making the entire image visually realistic. The regions to be filled are called hole regions, and could contain undesired foreground objects or small distracting elements, or corrupted regions of the image. Much research has been devoted to improving image inpainting either by image self-similarity (e.g. [3]) or deep generative models (e.g. [67, 60, 66]). Such methods synthesize realistic semantics and textures by reusing similar patches from non-hole regions or learning from large collections of images, respectively. However, those methods still struggle in cases when holes are large, or the expected contents inside hole regions have complicated semantic layout, texture, or depth.

These problems can be addressed if there happens to be a second reference image of the same scene that exposes some desired image content that can be copied to the hole. This task is referred to as reference-guided image inpainting in the literature [37], but this topic is less explored. In our paper, we call the image with the hole indicated for removal the target image. In general, there could be multiple other source images used as references. These could be taken by the photographer for the same scene after objects have moved or the photographer moved the camera to a different
viewpoint to expose the background. Alternatively, a source image could be collected from the Internet [57]. If one such source image contains new desired appearance for the target hole region, then we can copy the pixels from the source to fill in the target hole regions. In this paper we assume that the user has identified a particular source image with the new desired appearance, so we refer to this as the source image. We imagine that dedicated apps might be created for aiding the photographer in this process, or for automatically retrieving suitable such source images from the Internet.

Although the reference source image makes the inpainting task easier, reference-guided inpainting is still quite challenging for several reasons. First, the hole regions could be very large, which makes the task of guessing the pixel colors in the hole region less well-posed. Second, we wish for our task to be as general as possible, so we allow an uncalibrated camera to freely translate to different 3D positions for the source and target image, because this can allow the photographer to intentionally reveal regions behind a foreground object to be removed. Such translations, however, can induce large parallax, which cannot be modeled in image space by a simple 2D warp such as a global homography. Unlike video inpainting or multi-view Structure-from-Motion (SfM), we assume the system will not have access to more than two photos. Thus, it is harder in our setting to reliably estimate 3D structures, depth, and point correspondences. Third, depending on the camera and photography setup, the photographs may have substantially different exposure, white balance, or lighting environment, and if one photograph comes from the Internet, then it will have different camera response curves. Existing methods based purely on warping cannot resolve the resulting complex issues of color matching. Finally, there may exist regions in the source image that do not exist after warping due to pixels being out of the image or occluded.

To address these challenges, we propose a multi-homography fusion pipeline combined with deep warping, color harmonization, and single image inpainting. We address the issue of parallax by assuming that there may be multiple depth planes inside the hole. Loosely inspired by recent work on multiplane images [9, 75, 33, 53], we propose multiple homographic registrations of an assumption that the scene geometry lies on a different 3D plane. Given a target and a source image, we first estimate the matched feature points between the two images, cluster the inliers according to their estimated depths in the target image, and for each cluster estimate a single homography to perform an initial image registration. We call each of these candidate alignment images a proposal. For each proposal, we then tackle the challenge of color matching by using a deep bilateral color transformation, and we address parallax issues by refining the warp using a learned per-pixel spatial transformation. We then merge all the transformed source image proposals by learning a set of fusion masks. Finally, we address the last challenge regarding regions which do not exist in the source image by using a state-of-the-art single image inpainting method to complete missing regions, and learn to merge it as well.

In summary, the main contributions of our method are:

1. We propose TransFill, a multi-homography estimation pipeline to obtain multiple transformations of the source image, where each aligns a specific region to the target image;
2. We propose to learn a color and spatial transformer to simultaneously perform a color matching and make a per-pixel spatial transformation to address any residual differences after the initial alignment; (3) We learn weights suitable for combining all final proposals with a single image inpainting result.

2. Related Work

**Image inpainting.** Inpainting research can be divided into two categories: traditional methods that work by propagating colors or matching patches, and deep methods that learn semantics and texture from large image datasets.

Some traditional methods propagate pixel colors by anisotropic diffusion [4] or solving PDEs [2]. Such methods work well for thin hole regions but as the hole regions grow larger they tend to result in over-blurring. Patch-based image inpainting methods work by finding similar matches elsewhere in the image and copying the resulting texture [56, 3]. Those methods tend to result in high-quality texture but may give implausible structure and semantics.

Our work is more closely related to deep models for inpainting that use a single image. Context encoders analyze the surroundings of the hole [39], local and global discriminators [17] can improve local texture and overall image layout, and partial [29] and gated convolutions [67] can reduce artifacts from filter responses at the hole boundary.

More recently, some deep methods have focused on inferring other information first: these can be roughly categorized into using edges [34], segmentation masks [47], low-frequency structures [43, 27], and other possible maps like depth. The ill-posed nature [74] of single-image inpainting makes it challenging to complete larger holes and higher-resolution images. Recent works demonstrate neural networks can generate high-resolution images [63, 69, 65], but for large holes, these methods can still generate results that appear semantically implausible or have artifacts in the fine-scale texture. Since our method has a source reference image, we can better establish consistency with the ground truth image by learning appropriate spatial and color transformations for a source image patch.

**Video inpainting.** A few classical works in this area are Wexler et al. [56] and Granados et al. [13], which globally optimize patch-based energies, and Newson et al. [35].
which estimates multiple homographies using a piecewise planar assumption for the scene. Xu et al. [61] estimates the optical flow to learn the pixel warping field. Recently, the Onion-Peel Network (OPN) [37] leverages non-local designs inside the network, making it feasible to apply multi-source inpainting for a larger temporal range. Lee et al. [25] proposed a Copy-and-Paste Network to learn the alignment of consecutive frames for video inpainting. Zhao et al. [73] reuse contents from an unrelated image for a reference-based inpainting. Their method is based on only a single affine transform, which we show is not enough in our experiments, and exhibits residual color and geometric incompatibilities that are problematic in our multi-view scenario. Xue et al. [62] is a specialized method designed to remove reflective or occluding elements near the camera such as fences.

**Image harmonization.** Image harmonization refers to matching the color distribution and appearance when composing a foreground from one image on a background from another image. Traditional methods transfer color statistics locally and globally [41, 42, 50] and use gradient-domain based blending [40, 20, 51]. Digital photomontage [1] also demonstrated copy-and-paste workflows that can change the appearance of a foreground subject. Unlike our method, photomontage required user input and assumes the photographs have been aligned. Recently, CNN-based harmonization models [76, 59] are emerging, including methods involving segmentation masks [52] for region selection, and discriminators for domain verification [6]. Deep bilateral filtering has also been used to better preserve edges and details while transforming image color space [12, 54]. Our work is the first to integrate harmonization with a neural network for reference-guided inpainting. We apply a deep bilateral color transformation to address color inconsistencies while preserving edges.

**Image alignment.** Image alignment or registration involves placing multiple images in the same coordinate system. It is widely used for video stabilization [30], image stitching [24, 14], and serves as an important pre-processing step for many video and image applications like face analysis. Homography warping is a widely used global parametric method. Sparse local features like SIFT [31] can be matched either using nearest neighbour, or deep models like OANet [70] and SuperGlue [45], and the resulting correspondences can be used to estimate warping models. Recently, deep models have been explored to directly learn homography parameters [7, 71, 36], demonstrating their advantages on low-light and low-texture images.

Issues of parallax due to content at different depths can be better addressed by mesh-based warping [30, 26, 28] or pixel-wise dense optical flow [15, 58, 48, 49, 55, 18]. Liu et al. proposed the Content Preserving Warp (CPW) [28] to maintain the rigidity of motions. Recently, Ye et al. proposed deep meshflow [64] to make mesh estimation more robust on different scenes. Due to the sparsity of the mesh, image contents can be better retained while warping. However, optical-flow based methods can provide greater flexibility in permitted motions. Our pipeline uses multiple global homographies followed by per-pixel warping fields to combine the advantages of various alignment methods.

### 3. Method

We will first give an overview of our pipeline. Suppose we are given a target image $I_t \in \mathbb{R}^{W \times H \times 3}$, an associated mask $M \in \mathbb{R}^{W \times H \times 1}$, and a single source image $I_s \in \mathbb{R}^{W_s \times H_s \times 3}$. Note that $M$ indicates the hole regions with value one, and elsewhere with zero. The masked target image is then denoted by $I_t^M = (1 - M) \odot I_t$. We assume there is sufficient overlap in content between the two images especially nearby (but not necessarily within) the masked regions. Our task is to generate contents inside the masked regions of $I_t$ by effectively reusing contents of $I_s$. More specifically, we wish to geometrically align $I_s$ with $I_t$ in the vicinity of the hole region globally and locally, and adjust any color inconsistency. We fill any regions that are occluded or outside the image using a state-of-the-art single image inpainting method.

Our pipeline follows four steps as shown in Figure 2. It includes an initial registration using multiple homography proposals, per-pixel color and spatial transformations for each proposal, single-proposal fusion and multi-proposal fusion, as introduced in the following sections.

#### 3.1. Multi-homography Proposals

In this stage, we first globally warp the source image $I_s$ to align it with the masked target image $I_t^M$. Provided the contents inside the hole region occur at multiple depth planes, or the camera motion is not a simple rotation, a single homography is not sufficient to perfectly align the source and target image [10]. Therefore, we propose to es-
Figure 3: Multi-homography Proposal Module. We compute the monocular depth $D_i$ of the non-hole region $I^M_i$, and cluster the feature matching points into $N$ sub-groups using the depth values. Each estimated homography $H_i$ will align different regions within the hole. $H_6$ indicates a homography estimated using all the points.

To estimate multiple homography matrices to transform $I_s$. Ideally, each homography-transformed $I_s$ can align with $I_t$ within a specific image depth level range or local spatial area, as shown in Figure 3.

To obtain different transformation matrices, we first extract SIFT [32] features from $I^M_i$ and $I_s$, and feed all the extracted feature points and their descriptors into a pre-trained OANet [70] for outlier rejection. The lightweight OANet efficiently establishes the correspondences between $I^M_i$ and $I_s$ by considering the order of the points in the global and local context. OANet outputs the inliers forming a point set $P_t$ in $I^M_i$, and its corresponding matched point set $P_s$ in $I_s$. To consider different possible depth planes within and nearby the hole region, we are inspired by the Multi-Plane Image (MPI) [75] idea for scene synthesis. We estimate the depth map $D_i$ from $I^M_i$ using a deep learning based monocular depth estimator [16], and record the depth value for each point in $P_i$. We then cluster those points into a partition of $N$ subsets $\{P^j_i\}, j \in [1, N]$ by their depth values using an agglomerative clustering method [21], where $P_t = \cup_{j=1}^N P^j_i$. The corresponding matched points in $P_s$ are used to form the subsets $P_s = \cup_{j=1}^N P^j_i$ accordingly.

For each subset’s pairs of points $(P^j_i, P^j_s)$, we estimate a single homography using RANSAC [8]. By further including the homography estimated from the full set of points $(P_t, P_s)$, we obtain $N+1$ homography matrices overall. We denote them by $H_i$, $i \in [1, N+1]$. Finally, we transform the source image $I_s$ using the estimated $H_i$, and obtain a set of warped source images $\{I^w_i\}$, where $I^w_i \in \mathbb{R}^{W \times H \times 3}, i \in [1, N+1]$. We set $N = 5$ in our experiments.

Figure 4: Structure of the Color-Spatial Transformer Module. $I^w_i$ will first go through a Color Transformer (CT), and then a Spatial Transformer (ST) to obtain a refined source image $I^r_i$. The bottom row shows examples of the refinement stages. Blocks with blue color indicate there are learned parameters, otherwise they are parameter-free.

3.2. Color-Spatial Transformation (CST) Module

The global homography-warped source image sets $\{I^w_i\}$ are regarded as the initialization of the warping of $I_s$. However, as shown in Figure 3 and 4, while directly compositing $I^w_i$ and $I^M_i$ using $I^w_i \odot M + I^M_i$, due to the possibly inaccurate homography estimation or challenges of large parallax, there may be small misalignments inside and near the hole region, especially along the hole boundary. Additionally, the composite image may suffer from color and exposure differences. Therefore, we propose another refinement step that we call a Color-Spatial Transformer (CST). This simultaneously adjusts the color and alignment for each $I^w_i$. The structure of CST is illustrated in Figure 4. $I^w_i$ will first go through a Color Transformer (CT), and then a Spatial Transformer (ST) to obtain a refined source image $I^r_i$.

In our design of the color and spatial transformers, we would like to retain the texture details and the rigidity of the source image contents. Additionally, we prefer the color transformation and warping operations to be decoupled and not have to use auxiliary losses for each component. Inspired by deep bilateral filtering [12] and Spatial-Transformer Network (STN) [19], we propose to learn the transformations in a lower resolution, and obtain the full-resolution coefficients using up-sampling. Specifically, given $I^w_i$, $I^M_i$ and $M$, we down-sample them to 256 $\times$ 256 to obtain $I_s \downarrow$, $I^M_i \downarrow$ and $M \downarrow$. Then we compute the high-level features $u_i = B(I_s \downarrow, I^M_i \downarrow, M \downarrow)$ using a shared network $B$. After that, the color and spatial transformation coefficients will be learned by the CT and ST sub-networks.

Color Transformation (CT). To transform the color in RGB space of $I^w_i$ to $I^r_i$, we learn an affine transformation
with parameters $A_i^c = [K_i^c, b_i^c] \in \mathbb{R}^{W \times H \times 3 \times 4}$. Formally, for each pixel at location $p$, $I_{sc}^i(p) = K_i^c(p)I_s^i(p) + b_i^c(p)$, where $K_i^c(p) \in \mathbb{R}^{3 \times 3}$ and $b_i^c(p) \in \mathbb{R}^{1 \times 3}$. To better preserve the edges and textual details, we adopt deep bilateral filtering [12]. Specifically, we learn a bilateral grid $A_i^c = B_c(u_i^c) \in \mathbb{R}^{s \times s \times d \times 4}$ in a lower resolution, and a single-channel guidance map $g_i^c = G_c(I_s^i) \in \mathbb{R}^{W \times H \times 1}$ in full-resolution. We fix $s = 8$ and $d = 8$ in our experiments. $B_c$ and $G_c$ are the trainable networks for estimating the grid and guidance map. Finally, $A_i^c$ is tri-linearly sampled from $A_i^c$ using the normalized triplet $(x, y, g_i^c(p))$.

**Spatial Transformation (ST).** We learn the spatial warping offset $A_i^s = [A_{i,xx}^s, A_{i,yy}^s] \in \mathbb{R}^{W \times H \times 2}$ along the horizontal and vertical axes. To better preserve the rigidity of the image contents inside hole region, we propose to learn the warping field $A_i^s = B_s(u_i^s) \in \mathbb{R}^{s \times s \times 2}$ in a lower resolution, and up-sample it to $A_i^s$ using bi-linear interpolation. Finally, $I_s^i = \text{Warp}(I_{sc}^i; A_i^s)$. The objective loss to learn the CST module is defined by,

$$\mathcal{L}_{CS} = \|M_v \odot (M \odot (I_h - I_s^i))\|_1,$$

where $M_v = \mathbb{1}(I_s^i > 0)$ is the valid mask indicating the pixel regions after initial homography warping.

### 3.3. Single-Proposal Fusion (SPF) Module

The Single-Proposal Fusion (SPF) module learns to estimate a confidence map and other features for the refined results $I_s^i$ from the CST module by merging it with the outputs of a well-trained single image inpainting model called ProFill [69]. The inpainting results from ProFill often generate good structures, so the intuition for the SPF module is that we independently do an image comparison of each proposal with parameters $A_i^c$, the merged $I_s^i$, and the packed features $f_s^i$.

Figure 5: Single-Proposal Fusion (SPF) module. This takes $I_s^M$, $M$, a single $I_s^i$ and $I_g$ as inputs, where $I_g$ is the result of a single image inpainting method. SPF outputs a confidence map $c_i$, the merged $I_s^i$, and the packed features $f_s^i$.

### 3.4. Multi-Proposal Fusion (MPF) Module

The Multi-Proposal Fusion (MPF) module merges the $N + 1$ proposals of the refined source images $I_s^i$ and the single-image inpainting results $I_g$ together. The module is fed with the packed features $f_s^i$ and $f_g$ from the SPF module. Pixel-wise merging weights $\tau_i$, $i \in [1, N + 1]$ and $c_g$ are learned through a UNet [44] with softmax $(c_g + \sum_{i=1}^{N+1} \tau_i = 1)$ by merging different portions of proposals as $I_m$,

$$I_m = c_g \odot I_g + \sum_{i=1}^{N+1} \tau_i \odot I_s^i,$$

Then the final result $I_o = I_s^M + M \odot I_m$ is learned by the objective functions,

$$\mathcal{L}_o = \|M \odot (I_t - I_o)\|_1 + \text{VGG}(M \odot I_t, M \odot I_o),$$
where the VGG loss matches features of the pool5 layer of a pre-trained VGG19 [46]. Similarly, total variance losses are imposed to the weighting maps \( \pi_i \) and \( \varepsilon_g \), so we have the losses \( L_i^\pi = L_{TV}(\pi_i) \) and \( L_g^\varepsilon = L_{TV}(\varepsilon_g) \). Therefore, the overall loss function with \( \lambda_1 = 1, \lambda_2 = 1 \) becomes
\[
L_{all} = L_o + \lambda_1 L_i^\pi + \sum_{i=1}^{N+1} (L_{CS}^i + L_E^i + \lambda_2 (L_c^i + L_p^i)).
\] (7)

4. Experimental Results

4.1. Datasets and Implementation

Datasets. We trained the model on the RealEstate10K dataset [75]. This was collected from YouTube videos labelled as real estate footage. In total it consists of more than 8000 video clips with length from 1 to 10 seconds. For each clip, we randomly sampled pairs of images with a displacement of 10, 20, and 30 frames apart. We call this “Frame Displacement” (FD). This resulted in 188184 frame pairs for training, and 20290 pairs for testing. We generated random free-form brush-and-stroke holes like in DeepFillv2 [67]. We also collected 3K more pairs of real user-provided image pairs to serve as practical user cases for testing.

For training the Color-Spatial Transformer (CST), although RealEstate10K contains sufficient samples with real multi-view data and different exposures across image pairs, it lacks image pairs with large color inconsistency. Therefore, we synthesized misaligned color-different images from the MIT-Adobe5K dataset [5], and uniformly mixed these data with RealEstate10K for training. Adobe5K contains 5000 images, and for each image it provides five additional expert-retouched images to form 5000 sets in total. We regard the original samples as target images and synthesized the misaligned source images using the method in [7]. We make two binary variables for whether there is a color difference \( C \) and whether there is spatial misalignment \( S \), and synthesized pairs with \( CS, CS, CS, \hat{C}S \) and \( \bar{C}S \) with equal probability from 4000 sets to form a balanced training set, leaving 1000 sets for validation.

Implementations. We obtained a pre-trained OANet model for image feature matching and outlier rejection. We applied the pretrained model of Hu et al. [16] to estimate the depth map from a single target image. We also obtained a pre-trained ProFill [69] from the authors. All the above-mentioned model weights were frozen during training. Additionally, we pre-trained the CST module using the mixed dataset in advance for 400 epochs, and froze its weights afterwards. Finally, the whole pipeline was trained end-to-end for 400 more epochs. We used a patch size of 256 × 256 for training and arbitrary size for inference, and a learning rate of \( 10^{-4} \) with decay rate 0.5 after 200 epochs. We used the Adam optimizer [22] with betas (0.9, 0.999). The code is implemented in PyTorch [38].

4.2. Baseline Models

We chose baselines that are similar to, but may not exactly the same as our task, including approaches addressing image stitching [68], optical flow-guided video inpainting [61], non-local patch matching for multiple photo inpainting [37], and a state-of-the-art single image inpainting method [69] with the reference image concatenated so the method has access to the same inputs as the rest.

APAP [68]: As-Projective-As-Possible is a baseline image stitching algorithm that resolves depth parallax. We used the official Matlab 1 implementation for testing.

DFG [61]: Deep Flow-Guided Video Inpainting treats video inpainting as pixel propagation. It fills the holes by completing the optical flow field estimated by FlowNet2.0 [18]. We used their official2 pre-trained model for testing.

OPN [37]: Onion-Peel Network is a recent work addressing video and group photo inpainting using non-local attention blocks. We used their official PyTorch code3.

ProFill [69]: ProFill is a state-of-the-art single-image inpainting method that also contains a contextual attention module [66]. We used the official pre-trained model 4 from the authors. When testing, we fed in the target with the homography-warped source image. Before testing on RealEstate10K, we also fine-tuned OPN and ProFill on RealEstate10K training frames for fairness.

4.3. Qualitative Comparison

Results on User-Provided Images. In Figure 7, we show visual results of testing on real user-provided images. We indicate the hole region on the target image, and crop only the region of interest due to the space limits. More results can be found in the supplementary material. APAP and DFG well-preserve the source image contents due to the global homography warping, but they still suffer from color inconsistencies and alignment issues. We also experimented with combining Poisson blending with APAP but found it gives color bleeding artifacts: see the supplemental for details. OPN usually works well when there are multiple reference frames which have similar scales and color distributions within the same video clips. However, if only one source reference image exists, the non-local attention module struggles to search for similar local patches and fails. ProFill with the contextual attention module usually does well in searching for textures, but the estimated intermediate coarse results cannot be matched with specific image contents. Thus the reference-based ProFill can only achieve texture or object removal but not background contents recovery. Compared to them, ours better reuses the background patterns and achieves a content-aware alignment and
4.4. Quantitative Comparison

Results on RealEstate10K. The quantitative comparison on RealEstate10K is shown in Table 1. OPN and ProFill are more suitable for large batch testing. We tested them on the entire testing set. Results on cropped image pairs with Frame Displacement (FD) 10, 20 and 30 are reported in terms of PSNR, SSIM and LPIPS scores [72] based on AlexNet [23]. APAP and DFG are not suitable for large batch testing and their performance may be influenced by non-existing regions, so we sampled a 300-image subset from FD=10 as Small Set to test. Results showed that contextual-attention based ProFill failed to faithfully reconstruct the source contents. Optical-flow based DFG achieved better results by smoothly completing the flow field. OPN with atomic patch matching was not better than our warping-based approach. The TransFill thus demonstrated its superiority in faithful reconstruction.

User Study on User-Provided Images. To better evaluate the performance on our user-provided images, we conducted a user study via Amazon Mechanical Turk (AMT). We compared our method with each baseline separately and presented users with binary choice questions. We requested the users to choose one fill result which looks more realistic and faithful. To ensure the reliability, we used a pre-qualification test as well as check questions, as we explain in the supplementary material. For each method pair, we randomly sampled 80 examples, and each example was evaluated by 7 independent users. For each sample, one method was regarded as “preferred” if at least 5 users selected it. Samples voted by 3 or 4 users are considered confusing samples and filtered out. We reported TransFill’s Preference Rate (PR) in Table 1. The high preference rate demonstrates the effectiveness of TransFill. We also conducted a one-sample permutation t-test with 10^6 samples by assuming a null hypothesis that on average 3.5 users prefer one method. The p-values are all sufficiently small so we can draw the conclusion that the preference for our method was statistically significant.
Figure 8: Visualization of intermediate results. $I_s$ are the initialized homography warping of the source image $I_s$. $I^*_s$ are the learned spatial and color transformation of $I^*_s$. The final result $I_o$ is the merging of $I^*_s$ and the results of ProFill $I_g$ by pixel-wise weights $c_g$ and $c_o$ overlaying on the images. The result draws from regions that are better aligned: on the left from $I^*_s$, in the center from the single image inpainting $I_g$, and on the right from $I^*_s$ and $I^*_g$. Zoom in for better visualization.

Table 1: Quantitative Comparisons and User Study. **FD**: Frame Displacement. **PR**: Preference Rate.

<table>
<thead>
<tr>
<th>Model</th>
<th>FD=10</th>
<th>FD=20</th>
<th>FD=30</th>
<th>All</th>
<th>Small Set</th>
<th>PR</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>APAP [68]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>31.94 / 0.9738 / 0.0253</td>
<td>90.76%</td>
<td>$p &lt; 10^{-4}$</td>
</tr>
<tr>
<td>SPF [69]</td>
<td>32.43 / 0.9734 / 0.0259</td>
<td>32.47 / 0.9969 / 0.0320</td>
<td>32.43 / 0.9963 / 0.0321</td>
<td>32.40 / 0.9971 / 0.0207</td>
<td>95.65%</td>
<td>$p &lt; 10^{-4}$</td>
<td></td>
</tr>
<tr>
<td>OPN [37]</td>
<td>33.45 / 0.9765 / 0.0201</td>
<td>33.47 / 0.9734 / 0.0258</td>
<td>33.45 / 0.9737 / 0.0261</td>
<td>33.40 / 0.9793 / 0.0252</td>
<td>97.50%</td>
<td>$p &lt; 10^{-4}$</td>
<td></td>
</tr>
<tr>
<td>TransFill (Ours)</td>
<td>39.59 / 0.9919 / 0.0116</td>
<td>39.62 / 0.9889 / 0.0162</td>
<td>39.57 / 0.9867 / 0.0164</td>
<td>38.83 / 0.9914 / 0.0126</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Ablation Study on Multi-Homography Proposals.

<table>
<thead>
<tr>
<th>Sampling</th>
<th>N</th>
<th>Multi-Proposal</th>
<th>PSNR</th>
<th>SSIM</th>
<th>LPIPS</th>
<th>FD</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth</td>
<td>N=5</td>
<td>DANN</td>
<td>37.576</td>
<td>0.9879</td>
<td>0.0164</td>
<td>37.576</td>
<td>0.9879</td>
</tr>
<tr>
<td>Depth</td>
<td>N=5</td>
<td>Ratio Test [12]</td>
<td>37.444</td>
<td>0.9876</td>
<td>0.0168</td>
<td>37.444</td>
<td>0.9876</td>
</tr>
<tr>
<td>Random</td>
<td>N=5</td>
<td>DANN</td>
<td>37.499</td>
<td>0.9873</td>
<td>0.0166</td>
<td>37.499</td>
<td>0.9873</td>
</tr>
<tr>
<td>Spatial</td>
<td>N=5</td>
<td>DANN</td>
<td>37.384</td>
<td>0.9876</td>
<td>0.0169</td>
<td>37.384</td>
<td>0.9876</td>
</tr>
<tr>
<td>Depth</td>
<td>N=5</td>
<td>DANN</td>
<td>37.537</td>
<td>0.9878</td>
<td>0.0162</td>
<td>37.537</td>
<td>0.9878</td>
</tr>
<tr>
<td>None</td>
<td>N=1</td>
<td>DANN</td>
<td>37.092</td>
<td>0.9868</td>
<td>0.0172</td>
<td>37.092</td>
<td>0.9868</td>
</tr>
</tbody>
</table>

Table 3: Color-Spatial Transformation. C: Color, S: Spatial

<table>
<thead>
<tr>
<th>Order</th>
<th>PSNR</th>
<th>SSIM</th>
<th>LPIPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>C → S</td>
<td>37.576</td>
<td>0.9879</td>
<td>0.0164</td>
</tr>
<tr>
<td>S → C</td>
<td>37.566</td>
<td>0.9879</td>
<td>0.0163</td>
</tr>
<tr>
<td>Only S</td>
<td>36.717</td>
<td>0.9866</td>
<td>0.0182</td>
</tr>
<tr>
<td>Only C</td>
<td>36.228</td>
<td>0.9849</td>
<td>0.0179</td>
</tr>
</tbody>
</table>

4.5. Ablation Study

Type and Number of Multi-Homography Proposals. This ablation study was conducted on the testing set of the RealEstate10K. For each alternative, we re-trained the model. We compared the proposed depth-based points clustering methods with other alternatives including random and spatial clustering in Table 2. When we proposed five homography matrices, depth-based clustering works best. The results were fairly close when we set N to either 3 or 5, but N = 5 was slightly better in PSNR. However, these are much better than using just one global homography.

Color-Spatial Transformation Module. Table 3 shows that the order of the Color-Spatial Transformer did not make too much difference. However, according to our experiments, adjusting the color first made the training converge faster since the guidance map was computed from a fixed $I^*_s$. Table 3 also demonstrated that both the Color and Spatial Transformer were necessary.

Pipeline Components. Table 4 indicates that refining the source image with CST outperforms directly merging the initialized homography-warped images. SPF and its output confidence $c_i$ effectively guided the learning of MPF. The proposed full pipeline achieved the best performance.

5. Limitations, Discussion and Conclusions

The proposed method has limitations in certain situations. First, the pipeline may not work well on extreme low-light or texture inputs containing very few SIFT feature points. Second, our homography-based transformation is not suited for image pairs with extreme viewpoint changes. Third, the current model may struggle to transfer color if the lighting environment is very different, such as day to night. This is because we use an effective bilateral color harmonization. The results outperform state-of-the-art models, and could potentially be optimized with the multi-fusion pipeline together. We leave that for future work.

In conclusion, we contribute a multi-source image inpainting model based on multiple homography, deep warping and color harmonization. The results outperform state-of-the-art single image and multi-source inpainting methods, especially when the hole contains complicated depth.
References


[34] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. 2019. 2


2275


