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

Yuqian Zhou¹, Connelly Barnes², Eli Shechtman², Sohrab Amirghodsi²
¹IFP, UIUC, ²Adobe 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. More details are available at the project website.

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. [4]) or deep generative models (e.g. [69, 62, 68]). 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 [39], 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
we refine the warp using a learned per-pixel spatial bilateral color transformation, and we address parallax by using a deep proposal date alignment images a initial image registration. We call each of these candidates according to their estimated depths in the target image, and feature points between the two images, cluster the inliers target and a source image, we first estimate the matched age to the target, each corresponding to an assumption that the scene geometry lies on a different 3D plane. Given a recent work on multiplane images [10, 77, 35, 55], 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 [5] or solving PDEs [3]. 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 [58, 4]. 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 [41], local and global discriminators [19] can improve local texture and overall image layout, and partial [31] and gated convolutions [69] 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 [36], segmentation masks [49], low-frequency structures [45, 29], and other possible maps like depth. The ill-posed nature [76] 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 [65, 71, 67], 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. [58] and Granados et al. [14], which globally optimize patch-based energies, and Newson et al. [37].
which estimates multiple homographies using a piecewise planar assumption for the scene. Xu et al. [63] estimates the optical flow to learn the pixel warping field. Recently, the Onion-Peel Network (OPN) [39] leverages non-local designs inside the network, making it feasible to apply multi-source inpainting for a larger temporal range. Lee et al. [27] proposed a Copy-and-Paste Network to learn the alignment of consecutive frames for video inpainting. Zhao et al. [75] 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. [64] 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 compositing a foreground from one image on a background from another image. Traditional methods transfer color statistics locally and globally [43, 44, 52] and use gradient-domain based blending [42, 22, 53]. 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 [78, 61] are emerging, including methods involving segmentation masks [54] for region selection, and discriminators for domain verification [7]. Deep bilateral filtering has also been used to better preserve edges and details while transforming image color space [13, 56]. 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 [32], image stitching [26, 16], 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 [33] can be matched either using nearest neighbour, or deep models like OANet [72] and SuperGlue [47], and the resulting correspondences can be used to estimate warping models. Recently, deep models have been explored to directly learn homography parameters [8, 73, 38], 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 [32, 28, 30] or pixel-wise dense optical flow [17, 60, 50, 51, 57, 20]. Liu et al. proposed the Content Preserving Warp (CPW) [30] to maintain the rigidity of motions. Recently, Ye et al. proposed deep meshflow [66] 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 $I_t$ with $I_s$ 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 [11]. Therefore, we propose to es-
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 [34] features from $I_t^M$ and $I_s$, and feed all the extracted feature points and their descriptors into a pre-trained OANet [72] for outlier rejection. The lightweight OANet efficiently establishes the correspondences between $I_t^M$ 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 $I_t^M$, 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) [77] idea for scene synthesis. We estimate the depth map $D_t$ from $I_t^M$ using a deep learning based monocular depth estimator [18], and record the depth value for each point in $P_t$. Then we cluster those points into a partition of $N$ subsets $\{P_t^j\}, j \in [1, N]$ by their depth values using an agglomerative clustering method [23], where $P_t = \bigcup_{j=1}^N P_t^j$. The corresponding matched points in $P_s$ are used to form the subsets $P_s = \bigcup_{j=1}^N P_s^j$ accordingly.

For each subset’s pairs of points $(P_t^j, P_s^j)$, we estimate a single homography using RANSAC [9]. 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 $\{I_s^i\}$, where $I_s^i \in \mathbb{R}^{W \times H \times 3}, i \in [1, N+1]$. We set $N = 5$ in our experiments.

Figure 3: Multi-homography Proposal Module. We compute the monocular depth $D_t$ of the non-hole region $I_t^M$, 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.

Figure 4: Structure of the Color-Spatial Transformer Module. $I_s$ will first go through a Color Transformer (CT), and then a Spatial Transformer (ST) to obtain a refined source image $\hat{I}_s$. 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 $\{I_s^i\}$ are regarded as the initialization of the warping of $I_s$. However, as shown in Figure 3 and 4, while directly compositing $I_s^i$ and $I_t^M$ using $I_s^i \odot M + I_t^M$, 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 $I_s^i$. The structure of CST is illustrated in Figure 4. $I_s$ will first go through a Color Transformer (CT), and then a Spatial Transformer (ST) to obtain a refined source image $\hat{I}_s$.

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 [13] and Spatial-Transformer Network (STN) [21], we propose to learn the transformations in a lower resolution, and obtain the full-resolution coefficients using up-sampling. Specifically, given $I_s^i, I_t^M$ and $M$, we down-sample them to $256 \times 256$ to obtain $I_s^i \downarrow, I_t^M \downarrow$ and $M \downarrow$. Then we compute the high-level features $u_s^i = B(I_s^i \downarrow, I_t^M \downarrow, M \downarrow)$ 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 $I_s^i$ to $I_s^i_{sc}$, we learn an affine transformation.
PRO Fill [71] model and freeze its weights while training the learning task and learn confidence and difference features against this PRO Fill reference, to better constrain the overall results.

3.3. Single-Proposal Fusion (SPF) Module

The Single-Proposal Fusion (SPF) module learns to estimate a confidence map $c_i$ and other features for the refined results $\hat{I}_s^i$ from the CST module by merging it with the outputs of a well-trained single image inpainting model called PRO Fill [71]. The inpainting results from PRO Fill often generate results as inputs, where $I_g$ is the output of a single image inpainting method. SPF outputs a confidence map $c_i$, the merged $\hat{I}_s^i$, and the packed features $f_i^s$.

with parameters $A_i^s = [K_i^s, b_i^s] \in \mathbb{R}^{W \times H \times 4}$. Formally, for each pixel at location $p$, $I_{nc}^i(p) = K_i^s(p)f_i^s(p) + b_i^s(p)$, where $K_i^s(p) \in \mathbb{R}^{3 \times 3}$ and $b_i^s(p) \in \mathbb{R}^{1 \times 3}$. To better preserve the edges and textual details, we adopt deep bilateral filtering [13]. Specifically, we learn a bilateral grid $A_i^s = B_c(u_i^s) \in \mathbb{R}^{8 \times 8 \times 8 \times 4}$ in a lower resolution, and a single-channel guidance map $g_i^c = G_c(I_s^i) \in \mathbb{R}^{4 \times W \times H \times 1}$ 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, $A_i^s$ is tri-linearly sampled from $A_i^s$ using the normalized triplet $(x, y, g_i^c(p))$.

Spatial Transformation (ST). We learn the spatial warping offset $A_i^s = [A_i^s_{xx}, A_i^s_{yy}] \in \mathbb{R}^{W \times H \times 2}$ 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 $A_i^s = B_s(u_i^s) \in \mathbb{R}^{8 \times 8 \times 2}$ in a lower resolution, and up-sample it to $A_i^s$ using bi-linear interpolation. Finally, $\hat{I}_s = \text{Warp}(I_{nc}^i; A_i^s)$. The objective loss to learn the CST module is defined by,

$$L_{CS} = || M_v \odot M \odot (I_t - \hat{I}_s) ||_1, \quad (1)$$

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

3.4. Multi-Proposal Fusion (MPF) Module

The Multi-Proposal Fusion (MPF) module merges the $N + 1$ proposals of the refined source images $\hat{I}_s^i$ and the single-image inpainting results $I_g$ together. The module is fed with the packed features $f_i^s$ and $f_g$ from the SPF module. Pixel-wise merging weights $c_i^\tau, i \in [1, N + 1]$ and $c_g$ are learned through a UNet [46] with softmax ($c_g + \sum_{i=1}^{N+1} c_i^\tau = 1$) by merging different portions of proposals as $\hat{I}_m$.

$$\hat{I}_m = c_g \odot I_g + \sum_{i=1}^{N+1} c_i^\tau \odot \hat{I}_s^i, \quad (5)$$

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

$$L_o = || M \odot (I_t - I_o) ||_1 + VGG(M \odot I_t, M \odot I_o), \quad (6)$$
where the VGG loss loss matches features of the pool5 layer of a pre-trained VGG19 [48]. Similarly, total variance losses are imposed to the weighting maps $\tau_c$ and $g_c$, so we have the losses $L_{TV}^c = L_{TV}(\tau_c)$ and $L_{TV}^g = L_{TV}(g_c)$. Therefore, the overall loss function with $\lambda_1 = 1, \lambda_2 = 1$ becomes

$$L_{all} = L_o + \lambda_1 L_g^o + \sum_{i=1}^{N+1} (L_{cS}^i + L_{cE}^i + \lambda_2 (L_{c}^i + L_{g}^i)). \tag{7}$$

4. Experimental Results

4.1. Datasets and Implementation

Datasets. We trained the model on the RealEstate10K dataset [77]. 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 [69]. 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 [6], 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 [8]. We make two binary variables for whether there is a color difference ($C$) and whether there is spatial misalignment ($S$), and synthesized pairs with $CS, CS, C\bar{S}$ 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. [18] to estimate the depth map from a single target image. We also obtained a pre-trained ProFill [71] 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 $\times$ 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 [24] with betas (0.9, 0.999). The code is implemented in PyTorch [40].

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 [70], optical flow-guided video inpainting [63], non-local patch matching for multiple photo inpainting [39], and a state-of-the-art single image inpainting method [71] with the reference image concatenated so the method has access to the same inputs as the rest.

APAP [70]: As-Projective-As-Possible is a baseline image stitching algorithm that resolves depth parallax. We used the official Matlab $^1$ implementation for testing.

DFG [63]: 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 [20]. We used their official$^2$ pre-trained model for testing.

OPN [39]: Onion-Peel Network is a recent work addressing video and group photo inpainting using non-local attention blocks. We used their official PyTorch code$^3$.

ProFill [71]: ProFill is a state-of-the-art single-image inpainting method that also contains a contextual attention module [68]. 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 appendix. 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 appendix 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

$^1$APAP: https://cs.adelaide.edu.au/~tjchin/apap/
$^2$DFG: https://github.com/nb9/Deep-Flow-Guided-Video-Inpainting
$^3$OPN: https://github.com/seoungwugoh/opn-demo
$^4$ProFill: https://zengxianyu.github.io/iic/
composition. The generated results are more faithful to and compatible with the target image. The multi-homography proposal approach provides more options for warping initialization when the matched features are too complex for a single homography. It helps to resolve challenging cases when the hole regions do not belong to the dominant plane in the image as shown in row four. In Figure 8 we show intermediate results indicating regions selected to form the final results and how the CST resolves misalignment.

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 [74] based on AlexNet [25]. 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 appendix. 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.
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 \( c_t \) 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 features. 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 grid color matching, but do not incorporate any specialized models that reason further about lighting (e.g. [12]). Additionally, we utilize the pre-trained ProFill to fill the missing pixels so the final generation quality highly depends on the performance of the single image inpainting module ProFill. That module could be replaced by other 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

[1] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. In ACM SIGGRAPH 2004 Papers, pages 294–302. 2004. 3

[2] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. International journal of computer vision, 92(1):1–31, 2011. 12

[3] Coloma Ballester, Marcelo Bertalmio, Vincent Caselles, Guillermo Sapiro, and Joan Verdera. Filling-in by joint interpolation of vector fields and gray levels. IEEE transactions on image processing, 10(8):1200–1211, 2001. 2

[4] 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. 1, 2

[5] 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. 2

[6] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédéric Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition, 2011. 6

[7] Wenyan Cong, Jianfu Zhang, Li Niu, Liu Liu, Zhixin Ling, Weiyan Li, and Liqing Zhang. Dovenet: Deep image harmonization via domain verification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8394–8403, 2020. 3

[8] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. arXiv preprint arXiv:1606.03798, 2016. 3, 6

[9] 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. 4

[10] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world’s imagery. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5515–5524, 2016. 2

[11] Junhong Gao, Seon Joo Kim, and Michael S Brown. Constructing image panoramas using dual-homography warping. In CVPR 2011, pages 49–56. IEEE, 2011. 3

[12] Mathieu Garon, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, and Jean-François Lalonde. Fast spatially-varying indoor lighting estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6908–6917, 2019. 8

[13] Michaël Gharbi, Jiawen Chen, Jonathan T Barron, Samuel W Hasinoff, and Frédéric Durand. Deep bilateral learning for real-time image enhancement. ACM Transactions on Graphics (TOG), 36(4):1–12, 2017. 3, 4, 5, 12

[14] 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. 2

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision, pages 1026–1034, 2015. 19

[16] Charles Herrmann, Chen Wang, Richard Strong Bowen, Emil Keyder, Michael Krainin, Ch Liu, and Ramin Zabih. Robust image stitching with multiple registrations. In Proceedings of the European Conference on Computer Vision (ECCV), pages 53–67, 2018. 3

[17] Berthold KP Horn and Brian G. Schunck. ”determining optical flow”: A retrospective. 1993. 3

[18] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. 2019. 4, 6

[19] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. ACM Transactions on Graphics (ToG), 36(4):1–14, 2017. 2

[20] Eddy Ilg, Nikolaus Mayer, Tommoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2462–2470, 2017. 3, 6

[21] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In Advances in neural information processing systems, pages 2017–2025, 2015. 4

[22] Jiaya Jia, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Drag-and-drop pasting. ACM Transactions on graphics (TOG), 25(3):631–637, 2006. 3

[23] Stephen C Johnson. Hierarchical clustering schemes. Psychometrika, 32(3):241–254, 1967. 4

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014. 6

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6):84–90, 2017. 7

[26] Kyu-Yul Lee and Jae-Young Sim. Warping residual based image stitching for large parallax. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8198–8206, 2020. 3

[27] Sunghee Lee, Seong-Wug Oh, Dae-Young Won, and Seon-Joo Kim. Copy-and-paste networks for deep video inpainting. In Proceedings of the IEEE International Conference on Computer Vision, pages 4413–4421, 2019. 3

[28] Shiwei Li, Lu Yuan, Jian Sun, and Long Quan. Dual-feature aware image inpainting for mixed scenes. arXiv preprint arXiv:2003.06877, 2020. 2
[30] Feng Liu, Michael Gleicher, Hailin Jin, and Aseem Agarwala. Content-preserving warps for 3D video stabilization. *ACM Transactions on Graphics (TOG)*, 28(3):1–9, 2009. 3

[31] Guolin Liu, Futsun 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. 2

[32] Shuaicheng Liu, Ping Tan, Lu Yuan, Jian Sun, and Bing Zeng. Meshflow: Minimum latency online video stabilization. In *European Conference on Computer Vision*, pages 800–815. Springer, 2016. 3

[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. 3

[34] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 4, 8

[35] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 2

[36] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Rozenn Dahyot. N-dimensional probability density function transfer and its application to color transfer. In *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*, volume 2, pages 1434–1439. IEEE, 2005. 3

[37] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001. 3

[38] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H. Li, Shan Liu, and Ge Li. Structureflow: Image inpainting via structure-aware appearance flow. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 5

[40] 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. 3

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[42] Yuhang Song, Chao Yang, Yei 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. 2

[43] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 2432–2439. IEEE, 2010. 3

[44] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018. 3

[45] Kalyan Sunkavalli, Micah K Johnson, Wojciech Matusik, and Hanspeter Pfister. Multi-scale image harmonization. *ACM Transactions on Graphics (TOG)*, 29(4):1–10, 2010. 3

[46] Michael W Tao, Micah K Johnson, and Sylvain Paris. Error-tolerant image compositing. In *European Conference on Computer Vision*, pages 31–44. Springer, 2010. 3

[47] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3789–3797, 2017. 3

[48] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 551–560, 2020. 2

[49] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed photo enhancement using deep illumination estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6849–6857, 2019. 3

[50] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical
flow with deep matching. In *Proceedings of the IEEE international conference on computer vision*, pages 1385–1392, 2013. 3

[58] Yonatan Wexler, Eli Shechtman, and Michal Irani. Spacetime completion of video. *IEEE Transactions on pattern analysis and machine intelligence*, 29(3):463–476, 2007. 2

[59] Oliver Whyte, Josef Sivic, and Andrew Zisserman. Get out of my picture! internet-based inpainting. In *Proceedings of the 20th British Machine Vision Conference, London*, 2009. 2

[60] Jonas Wulff and Michael J Black. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 120–130, 2015. 3

[61] Cun Xiaodong and Pun Chi-Man. Improving the harmony of the composite image by spatial-separated attention module. *arXiv preprint arXiv:1907.06406*, 2019. 3

[62] Wei Xiong, Jiahui Yu, Zhe Lin, Jimei Yang, Xin Lu, Connelly Barnes, and Jiebo Luo. Foreground-aware image inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5840–5848, 2019. 1

[63] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Deep flow-guided video inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4723–4732, 2019. 3, 6, 8

[66] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting with iterative confidence feedback and guided upsampling. *arXiv preprint arXiv:2005.11742*, 2020. 2, 5, 6, 8

[71] Yu Zeng, Zhe Lin, Jimei Yang, Jianming Zhang, Eli Shechtman, and Huchuan Lu. High-resolution image inpainting with iterative confidence feedback and guided upsampling. *arXiv preprint arXiv:2005.11742*, 2020. 2, 5, 6, 8

[72] 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. *International Conference on Computer Vision (ICCV)*, 2019. 3, 4

[73] Jirong Zhang, Chuan Wang, Shuaicheng Liu, Lanpeng Jia, Nianjin Ye, Jue Wang, Ji Zhou, and Jian Sun. Content-aware unsupervised deep homography estimation. *arXiv preprint arXiv:1909.05983*, 2019. 3

[74] 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. 7

[75] Yu Zeng, Zhe Lin, Jimei Yang, Jianming Zhang, Eli Shechtman, and Huchuan Lu. High-resolution image inpainting with iterative confidence feedback and guided upsampling. *arXiv preprint arXiv:2005.11742*, 2020. 2, 5, 6, 8

[76] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1438–1447, 2019. 2

[77] 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. 2, 4, 6

[78] Jun-Yan Zhu, Philipp Krahenbuhl, Eli Shechtman, and Alexei A Efros. Learning a discriminative model for the perception of realism in composite images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3943–3951, 2015. 3
A. Appendix A: More Ablation Studies

A.1. Color-Spatial Transformer

Figure A.1: Ablation study on the resolution setting in the Color-Spatial Transformer. First row, direct composition of target image $I^M_t$ and one of the single-homography transformed source images $I^s_i$. Second row, the learned pixel-wise warping field $A^s_i$ visualized using color wheel in [2]. Third row, the color-spatial transformed image $\hat{I}^s_i$. Last row, the final merging result $I_o$.

Recall that while introducing the Color-Spatial Transformer, we intend to preserve the texture details and the rigidity of the source image contents. Therefore, given $A^s_i = [K^s_i \ b^s_i] \in \mathbb{R}^{W \times H \times 3 \times 4}$, and $\bar{A}^s_i = B_s(u^s_i) \in \mathbb{R}^{s \times s \times 2}$, we fix $s = 8$ and $d = 8$ in our experiments. We find $d$ does not influence the performance a lot, and the guidance map is automatically learned to uniformly span the necessary bins like in the HDRNet[13]. Figure A.1 shows the comparison when we set different $s$ values. It suggests that increasing $s$ gives more degrees of freedom to the learned warping field $A^s_i$. However, while encountering larger holes like in Figure A.1, better flexibility does not better align the contents as expected, but distorts the contents inside the hole. The transformed color field also becomes less smooth as $s$ increases. In an extreme case, suppose we replace the deep bilateral grid and directly learn a full-resolution pixel-wise color-warping field with total variance constraints as in the last column, the model struggles to infer a reasonable color-warping operation within a large hole.

We conclude that CST with smaller $s$ value like $s = 8$ generalizes better to inference images with varying spatial resolutions. It is mainly due to the ill-posedness of image completion. Unlike conventional image registration tasks where all the pixels of the matched regions are available, hole regions are missing in the inpainting task. Less freedom in the hole area preserves better content integrity and semantics.

Figure A.3: One example of user-involved interactive editing. For the given target and source images on the left, we generate seven proposals including one $I^g$ from ProFill. For the upper group of images, we visualize the regions selected by our model to synthesize the final results. The image with red bounding box yields an unexpected artifacts of stairs. By zeroing out its corresponding $c_2$, we can correspondingly obtain zero-valued $\bar{c}_2$ as shown in the lower image group. Other maps are also correspondingly redistributed. The final result on the lower-right position is then generated by merging the other selected proposals with nonzero weighted masks. As we can see, the artifact disappears.

A.2. Network Components

Figure A.2: Final fusion masks $\bar{c}_i$ learned by the model with or without the Single-Proposal Fusion (SPF) module. By using SPF outputs as guidance to learn the MPF, the final weights learned tend to be more sparse.

Importance of Single-Proposal Fusion (SPF) Our experiments exhibit that the proposed Single-Proposal Fusion (SPF) module before the Multi-Proposal Fusion (MPF) is necessary for effectively learning the final merging weights of all the proposals. We find directly learning the weights to fuse all the proposals is very challenging. The learned weights have a hard time becoming sparse even though the same total variance loss is imposed. A comparison of the merging mask $\bar{c}_i$ between the model with and without SPF is shown in Figure A.2. Using SPF outputs $c_i$ as a structure guidance for learning the fusion of multiple proposals works better in practice.

Correlation between $c_i$ and $\bar{c}_i$ In our experiments, the learned single-proposal fusion mask $c_i$ and multi-proposal
fusion mask $c_i$ demonstrate strong correlation. Specifically, by zeroing out one of the $c_i$, the values in $\bar{c}_i$ will also vanish. This shows the MPF constructs the correspondence to make $\bar{c}_i$ be conditioned on $c_i$. This provides more flexibility for our model to incorporate user interactions. Suppose users want to eliminate the elements in some proposals, one can simply zero them out and the final results will only be merged from other selected proposals. Such a process is demonstrated in Figure A.3.

![Ablation study on APAP with Poisson blending post-processing.](image)

**Figure A.4:** Ablation study on APAP with Poisson blending post-processing. The color bleeding artifacts are significant in some cases when there are strong color mismatches or non-existing regions. APAP does not include image inpainting, so regions that are outside the source image appear as black.

A.3. APAP with Poisson Blending

We experimented with using Poisson blending [42] combined with APAP. The testing result on the Small Set of images with only few non-existing regions is increased from 31.94dB / 0.9738 to 32.56dB / 0.9754 in terms of PSNR / SSIM. However, we did not incorporate Poisson blending in the baseline because we found in some cases there could be significant color bleeding artifacts due to strong color mismatches and non-existing regions especially along the boundary of the hole. Some visual comparisons are shown in Figure A.4.

A.4. Using ProFill with Partial Masks

As we stated that single image techniques don’t work well for larger holes, while in our work, the single-image inpainting is computed over the full mask area. We also thought about using ProFill or other single-image inpainting method with partial mask, but could not find a principled and an end-to-end way to do this. However, we analyzed an approximation of this approach where we used the confidence map $c_g$ estimated by our method, and binarized it to do a post-hoc fill (with ProFill) of each hole region of the target that corresponds to single image inpainting (where the content is not visible in the source image or not well reused). Comparisons are shown in Figure A.5. This reveals that since the mask was learned for merging purposes, a post-hoc filling using the mask may introduce other artifacts like broken door frames.

![Post-hoc refilling results using ProFill.](image)

**Figure A.5:** Post-hoc refilling results using ProFill. The TransFill columns show a zoom of the original output, the post-hoc filling result (TransFill + Post), and the region to be refilled from the confidence $c_g$ (Binary Mask). The refilling with a partial mask may introduce additional artifacts like broken door frames.

B. Appendix B: User Study Details

![User study interface and check questions.](image)

**Figure B.6:** User study interface and check questions. We set up 10 trivial questions to let the users choose which one is more completed.

The GUI of our user study at AMT is shown in Figure B.6. To guarantee the reliability of the users’ feedback, we require the users to take a qualification test before they evaluate. The test presents users with the 10 trivial pairs $I_t$ and $I^M_t$ and users who answer correctly more than 8 questions are approved to take the official test. We also mix 10 random sanity check questions with the real questions. No
users had to be disqualified due to failing the initial test, and only very few users (4 users) got check questions later in the study wrong (5.7% of total opinions), so we conclude that the user responses are reliable.

C. Appendix C: Failure Cases

Figure C.7: Failure cases. These demonstrate limitations with large changes in viewing angle, outpainting artifacts from the off-the-shelf single image inpainting module ProFill, and challenges in handling dramatically different lighting environments.

Figure C.7 shows some examples of failure cases when the viewing angle changes are large. The color matching module may struggle if there are extreme lighting differences. We may also encounter outpainting artifact issues caused by ProFill.

D. More Visual Results Comparison

Visual Results on RealEstate10K We present more visual results in Figure D.8 on the RealEstate10K dataset. Compared with the baselines, our proposed TransFill achieves better spatial alignment and content faithfulness.

Visual Results on Synthetic Adobe-5K In Figure E.9, we show more results on the synthetic Adobe-5K dataset to evaluate the performance of our color transformation. As stated in the paper, we synthesize misaligned and color inconsistent images from Adobe-5K dataset. The spatial transformation is a simple homography-based warping, so the CST module works well to align the images and match the color. More challenging cases can be visualized in user-provided images.

Appendix D: More Results on User-provided Images
Additional higher-resolution results can be found at the following link: Additional Results.

E. Appendix E: Unfolding the Model: Intermediate Results

In Figure E.10 and E.11, we unfold the whole pipeline of TransFill to visualize the intermediate results of each proposed module. We demonstrate the process of image completion in a more intuitive way. After proposing different homography-warped images, the CST effectively adjusts the misalignment and color mismatching. Then the proposed TransFill fills in the holes by selectively merging the well-aligned and color-consistent regions from different proposals. Imperfect regions are finally filled with the output from ProFill.
Figure D.8: Visual results comparison on the RealEstate10K dataset with FD=10. These have been cropped. Please zoom in so that there are about 3-4 images across the width of the screen to reveal the significant differences in fine details. Compared with the baselines, our proposed TransFill achieves better spatial alignment and faithfulness to the source image content.
Figure E.9: Visual results on the synthetic Adobe-5K dataset. For each group of photos, the left one is the composition of $I^M_t$ and $I_s$. We transform the color and warp $I_s$ to make it consistent with $I^M_t$ and composite them as the right image. Our CST module resolves the color mismatches and spatial misalignment problems.
Figure E.10: Unfolding the whole pipeline to visualize the intermediate results of each module.
Figure E.11: Unfolding the whole pipeline to visualize the intermediate results of each module. For some challenging cases when the line alignment is hard, our model can also leverage the outstanding performance of line generation of ProFill to synthesize the door frame.
F. Appendix F: Network Structures

The network structures are summarized in Table 5, 6 and 7. The structures follow a UNet structure with minor modifications such as some shared structures and some parameter-free components like the tri-linear interpolation layer in the CST. SPF is implemented by a shallower UNet than MPF.

Table 5: Network structure of CST. Prior to each convolution except EncConv1_1, a PReLU [15] is applied as a pre-activation.

Table 6: Network structure of SPF.

Table 7: Network structure of MPF.