Image Completion using Planar Structure Guidance

Jia-Bin Huang[1] Sing Bing Kang[2] Narendra Ahuja[1] Johannes Kopf[2]

[1]University of Illinois at Urbana-Champaign [2]Microsoft Research

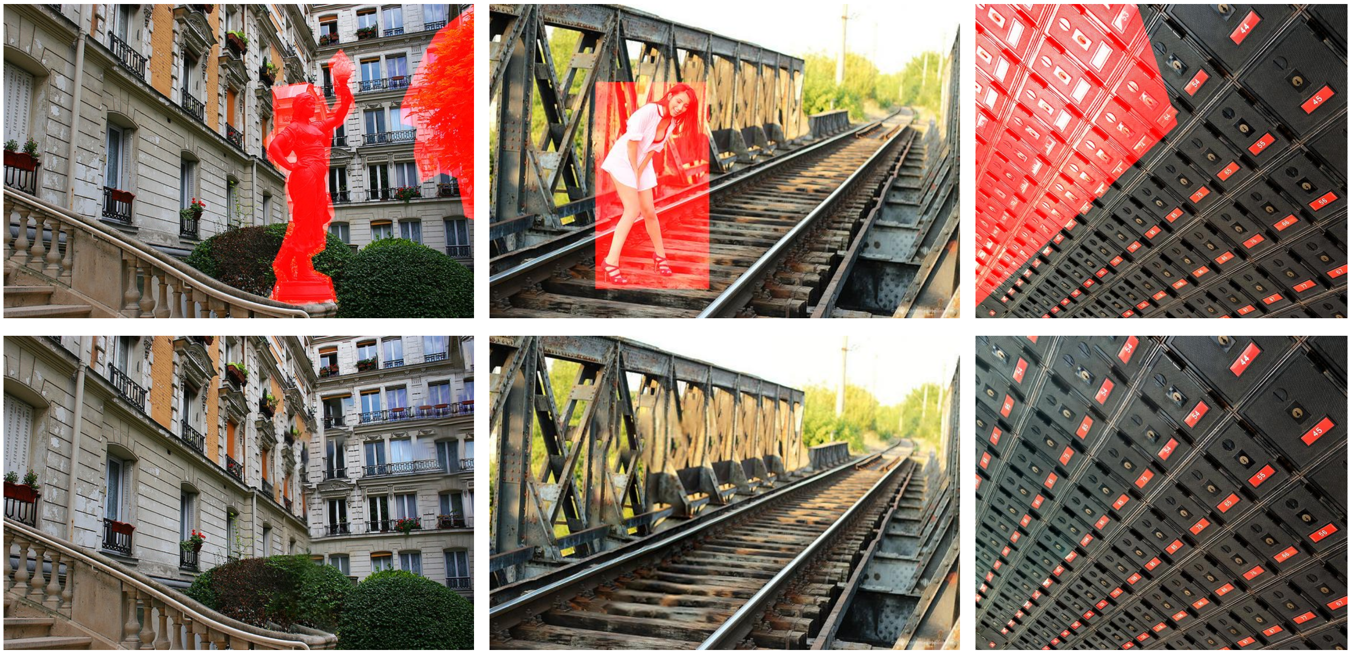


Figure 1: *Our image completion algorithm automatically extracts mid-level constraints (perspective and regularity) and uses them to guide the filling of missing regions in a semantically meaningful way. Our method is capable of completing challenging scenes such as multiple building facades (left), strong perspective distortion (middle) and large regular repetitive structures (right). We significantly outperform three representative state-of-the-art image completion techniques for these images (see Figure 2). Image credits (left to right): Flickr users micromegas, Theen Moy, Nicu Buculei.*

Abstract

We propose a method for automatically guiding patch-based image completion using mid-level structural cues. Our method first estimates planar projection parameters, softly segments the known region into planes, and discovers translational regularity *within* these planes. This information is then converted into soft constraints for the low-level completion algorithm by defining prior probabilities for patch offsets and transformations. Our method handles multiple planes, and in the absence of any detected planes falls back to a baseline fronto-parallel image completion algorithm. We validate our technique through extensive comparisons with state-of-the-art algorithms on a variety of scenes.

CR Categories: I.3.8 [Computer Graphics]: Applications;

Keywords: Patch-based synthesis, image completion, mid-level analysis, guided synthesis

Links: DL PDF WEB DATA CODE

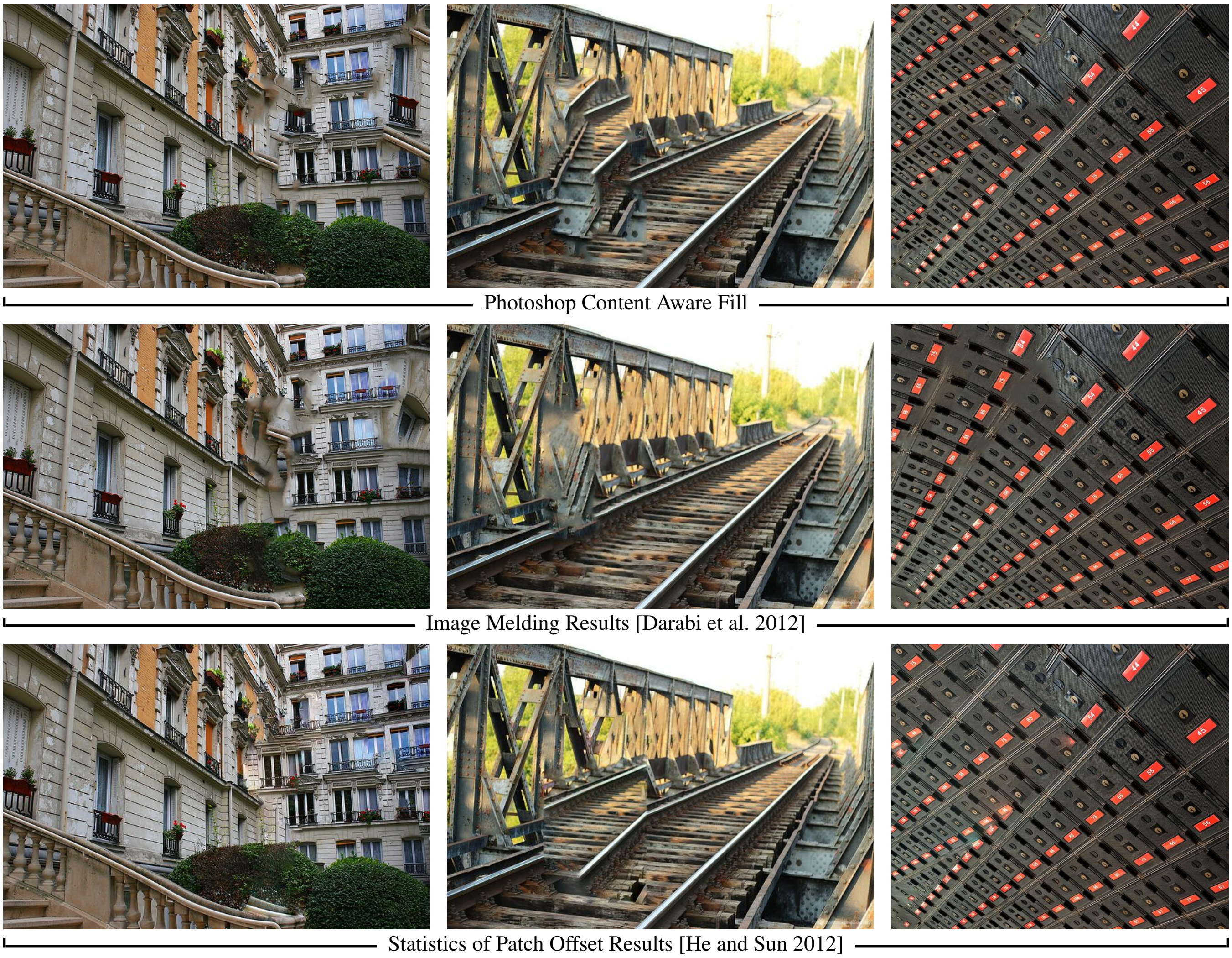
1 Introduction

Replacing or filling regions in images with plausibly synthesized content is a common image editing operation. This task, known as image completion, is used in applications ranging from the removal of unwanted objects in personal photos to movie post-production. It is also an important step in many graphics algorithms, e.g., for generating a clean background plate or reshuffling image contents.

While much progress has been made, image completion remains a challenging problem. This is because some amount of higher level understanding of the scene is often required. The state-of-the-art automatic algorithms typically rely on low-level cues; they synthesize the missing region as a field of overlapping patches copied from the known region [Wexler et al. 2007]. Here, they attempt to synthesize an image that *locally* appears like the known input everywhere, and such that overlapping patches agree as much as possible. Barnes et al. [2009] showed how this algorithm can be sped up using a random search and propagation scheme.

Most of these algorithms have two important limitations. First, since they only directly copy translated patches from the input, the performance degrades with scenes that are not fronto-parallel. They would not be able to effectively handle the perspective foreshortening as shown in Figure 1. The other limitation is in the tendency of converging to local minima, due to the strong non-convexity of the objective. This second problem is somewhat alleviated by applying the algorithm in a coarse-to-fine manner.

Recent approaches handle the fronto-parallel limitation by considering patch transformations such as rotation, scale, and gain/bias color adjustments [Mansfield et al. 2011; Darabi et al. 2012]. While



Photoshop Content Aware Fill

Image Melding Results [Darabi et al. 2012]

Statistics of Patch Offset Results [He and Sun 2012]

Figure 2: *Limitations of current state-of-the-art methods. Compare these results with ours in Figure 1.*

this improves the algorithm's ability to complete general scenes, it results in an exponential increase of the search space from 2 degrees of freedom per output pixel up to 8 (or more). This adds many local minima to the solution space and hence worsens the tendency of giving rise to bad configurations. This effect can be observed in Figures 2, 8, 9, 10 and in numerous other comparisons in the supplementary material.

In this paper, we show how image completion can be substantially improved by automatically *guiding* the low-level synthesis algorithm using *mid-level structural analysis* of the known region. Specifically, we estimate planar projection parameters (i.e., the local perspective slope of the scene) as well as translational regularity in the affine rectified domain (explained later in Section 4.1); this information is used to constrain the search space in the missing region. These constraints are cast as a prior probability of the patch transformation parameters. As a result, we can use an even richer patch transformation model than previous work (i.e., full homographies) since our constraints effectively reduce this high dimensional model to a lower degree subspace.

We handle multiple detected planes (that may be perspectively distorted) by using a soft proximity-based weighting scheme and relying on the power of the low-level algorithm for finding good transitions. Note that while we model the world as piecewise planar, we are *not* just restricted to such scenes: just as the original completion algorithm was not limited to fronto-parallel scenes, ours allows significant deviation from the piecewise planar model, as evidenced by our results.

Our algorithm significantly improves performance for challenging man-made scenes such as those of architecture and indoors. In the absence of any detected structural cues, e.g., for most natural landscape images (Figure 9), our algorithm falls back to standard unconstrained completion, i.e., our implementation of Wexler et al.'s algorithm [2007]. We validate our method by comparing against state-of-the-art algorithms. We show numerous representative results in the paper and supplementary material.

2 Previous Work

In this section, we review representative techniques for image completion. Image completion techniques can be roughly categorized as diffision-based or example-based.

Diffusion-based techniques fill in small or narrow holes by propagating adjacent image structures. This class of techniques is pioneered by Bertalmio et al. [2000] and later extended by Ballester et al. [2001] and Bertalmio et al. [2003]. These techniques are less effective in handling large missing regions due to their inability to synthesize textures.

By comparison, example-based methods exploit redundancy in natural images for filling missing pixels. They are based on example-based texture synthesis methods [Efros and Leung 1999; Efros and Freeman 2001]. Variants of these methods include using structure-based priority [Criminisi et al. 2004], deterministic EM-like schemes [Kwatra et al. 2005; Wexler et al. 2007; Barnes et al. 2009], or MRF models with patches as labels, which can be solved efficiently using belief propagation [Komodakis and Tziritas 2007] or graph cut [Pritch et al. 2009]. These techniques still fundamentally rely on low-level cues, which are less effective for images with larger structures.

In many real scenes, the appearance can change significantly due to local scene shape variation such as perspective distortion. As a result, it may be difficult to synthesize plausible regions if only translated patches are considered. Recent approaches address this problem by increasing the motion parameter search space to similarity transform with reflection and accommodate slight photometric variations [Mansfield et al. 2011; Darabi et al. 2012]. While the additional motion parameters do help when needed, the increased dimensionality and complexity render the nearest neighbor searching algorithm even harder to find a good solution. We handle this issue by constraining the transformation based on mid-level structural analysis of the image.

Missing regions in images could also be completed with the help of external image datasets. Hays and Efros [2007] retrieve semantically similar images from a large dataset and copy a single large region to fill the missing pixels. A similar scene matching strategy was adopted by Zhang et al. [2013]; the main difference is that they transfer the self-similarity field to guide the completion instead of the actual contents of the matched image. Another example of using external database is through instance-level matching methods to fill in missing regions via appropriate geometric and photometric transformation of the retrieved image [Whyte et al. 2009]. In this work, we consider only the known region in the input image.

The notion of automatic guidance maps for image completion has been used in a number of approaches. For example, Jia et al. [2003] inferred the line and contour continuation in the missing regions and used them for completion. A similar type of salient line matching was used in problems of tele-registration [Huang et al. 2013a]. Kopf et al. [2012] used tile-based search space constraints to constrain the source of texture examples in synthesis. He and Sun [2012] identify a number of representative translation offsets from the known region of the input image and use these offsets to complete the image. However, their method detects regularity only in the image plane, and is, thus, not very effective on larger structures with significant perspective effects.

While it is desirable to have a fully automatic approach, the image completion technique may still fail on occasion because computer vision techniques are typically far from perfect. Interactive methods allows users to explicitly provide high-level expertise to guide the completion. User-specified constraints include label map [Hertzmann et al. 2001], line structure continuation [Sun et al. 2005], perspective [Pavić et al. 2006], lattice [Liu et al. 2004], and symmetry [Huang et al. 2013b].

3 Overview

We implemented as baseline algorithm the non-parametric optimization algorithm of Wexler et al. [2007], and use random search and propagation as in PatchMatch [Barnes et al. 2009]. We use two types of mid-level constraints of the scene to guide the low-level completion process: planar perspective and translational regularity. Given an image and a user-specified mask that specifies the region (or hole) to fill, we first detect multiple planes, estimate their perspective parameters, and determine their spatial supports within the scene (Section 4). To determine translational regularity within each plane, we perform feature matching using SIFT features [Lowe 2004]. The positions of all matched feature pairs are then affine rectified[1] using the corresponding plane parameters. This allows the dominant translational shifts to be easily detected through clusters of displacement vectors in the rectified domain.

We use the detected perspective planes and the translational regularity within each plane as soft constraints to guide the low-level image completion (Section 5). We achieve this by integrating these derived constraints as prior probabilities of the search space. The regularity detection step provides "positional" guidance, i.e., where the source patch should be copied from. In contrast, the plane orientation constraints provide "non-positional" guidance of source patches, i.e., how the source patch should be deformed. By incorporating both positional and non-positional constraints on searching source patches from the known region, we show that these two types of mid-level image analysis can significantly improve the quality of the completed region in a semantically meaningful way.

4 Detecting Planar Surfaces and Regularity

In this section, we describe our analysis of the known image region to detect planar surfaces (Section 4.1) and translational regularity within these planes (Section 4.2). The results of this analysis will be used to constrain the low-level completion algorithm, as described in Section 5.

4.1 Planes

Many techniques have been proposed for identifying and rectifying planes [Chum and Matas 2010; Zhang et al. 2012; Aiger et al. 2012], i.e., converting a perspectively distorted plane to a fronto-parallel version. We use a technique that involves line segment extraction, vanishing point estimation [Hartley and Zisserman 2004], and grouping based on vanishing points. Since this part of our algorithm is relatively standard, we provide only a brief description here. We first detect edges and fit line segments in the known region of the image. We then detect up to three vanishing points (VPs) using a RANSAC-based voting approach. This means we assume there are only up to three different plane orientations in the scene. This is reasonable for typical man-made structures. We show a sample result in Figure 3.

Given the three VPs, we can recover up to three plane orientations, one from each pair of the detected VPs. We compactly represent the parameters of plane m using the vanishing line $\mathbf{l}_\infty^m$ (the image of the line at infinity on the world plane connecting the two distinct VPs):

$$\mathbf{l}_\infty^m = [l_1^m, l_2^m, l_3^m]^\top. \quad (1)$$

Note that $\mathbf{l}_\infty^m$ is homogeneous and has two degrees of freedom. The perspective image of a plane can then be affine rectified (so that parallel lines in 3D appear parallel in the image) using a pure perspective transformation matrix

$$\mathbf{H}_m = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1^m & l_2^m & l_3^m \end{bmatrix}. \quad (2)$$

[1] Affine rectification means mapping vanishing points to infinity so that parallel lines in 3D space project to parallel 2D lines in the rectified image.

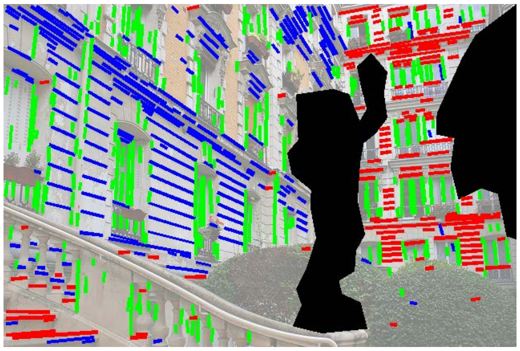


Figure 3: *Vanishing point detection from a man-made environment. The red, green, and blue line segments correspond to three detected vanishing points, respectively.*

However, the plane parameters provide no information on the spatial support of the plane in the image domain. While there are many computer vision algorithms available for automatic single-view reconstruction of man-made environment, they are usually quite sophisticated, e.g., see Barinova et al.'s work [2008]. Instead, we address this problem via a rather simple and straightforward approach.

Our key insight is that a plane typically consists of two sets of parallel 3D lines. In other words, there are usually two sets of the line segments with two distinct VPs that should reside within the same image region. We identify the support of each plane by locating positions where the two sets of line segments corresponding to the two VPs overlap with each other.

We first estimate the spatial support of each VP by diffusing its corresponding line segments using a wide Gaussian kernel. Then, we estimate the spatial support for the planes by performing element-wise multiplication of its VP's support line density maps. These product maps have a high response where the two sets of the line segments overlapped with each other. Note that we always add the fronto-parallel plane with parameters $l_\infty^0 = [0,0,1]^\top$ and assign a fixed density value 10^{-5} uniformly across the image. We then perform *per-pixel* normalization of this density product map so that the sum over the plane membership probability is 1; we call this the "posterior probability" $\Pr[m|\mathbf{x}]$ for assigning plane membership m at pixel $\mathbf{x}$. This process is illustrated in Figure 4. Here, the posterior probability distributions are shown as color-coded density maps on the right column (note that the density map for the fronto-parallel plane is not shown).

As lines can only be detected in the known region of the image, the posterior probabilities within the unknown region are highly unreliable. To address this problem, we assign to every missing pixel the probabilities of the closest boundary pixel. The posterior probability map of an example image is shown in Figure 5.

4.2 Regularity Extraction

Regular and near-regular structures are ubiquitous in man-made environments as well as in many natural objects. Detection of such regularity has been shown to be a compact means for understanding scene structure. Liu et al. [2010] provide a good survey of work in this area.

Similar to He and Sun [2012], we also detect translational regularity using offsets of matched image features. However, we detect regularity in a localized manner and in affine rectified space in order to account for possibly multiple foreshortened planes. We begin with detecting standard Difference of Gaussian feature points in the known image region and compute the SIFT descriptors for each feature point [Lowe 2004]. We choose to extract features in the original image rather than rectified space because the rectification would severely distort the image for slanted planes (e.g., rectifying the ground plane in the middle image in Figure 1 would lead to extreme distortions near the horizon). We compute the two nearest neighbors for each feature using a kd-tree. We only retain matches whose ℓ_2 feature distances are below a threshold of 0.1.

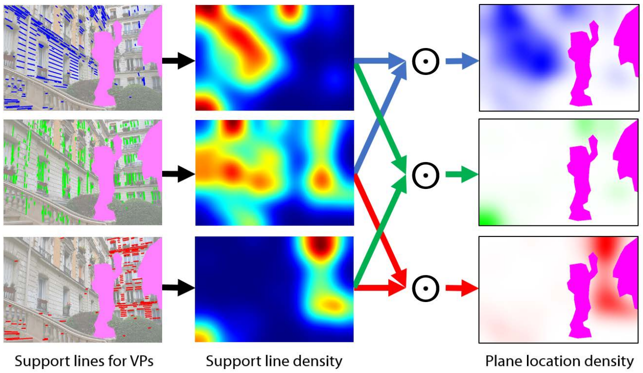


Figure 4: *Plane localization in the known region using support line segments from pairs of vanishing points. In the hole region we assign plane probabilities in a different manner, as shown in Figure 5.*

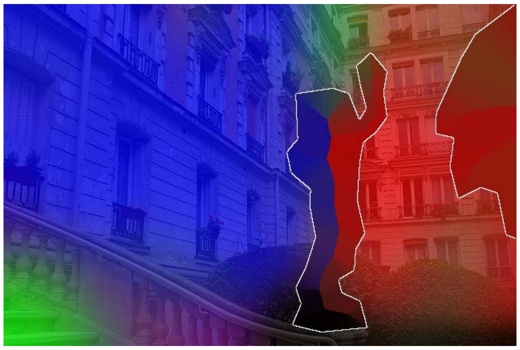
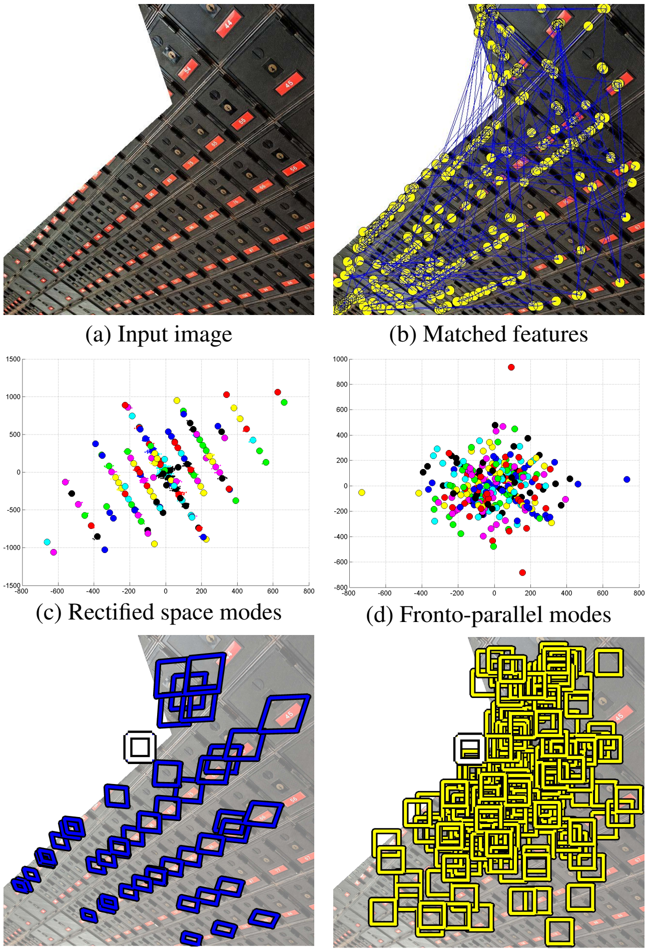


Figure 5: *Visualization of plane posterior probability in the known region, and propagation into the hole region. The posterior probabilities of the three recovered planes are shown in blue, green and red. (Note that the fronto-parallel plane is not shown here.) The boundary pixels between the known and unknown regions are shown in white. In the hole region every pixel is assigned the plane probabilities of the nearest boundary pixel.*

Next, for each plane m, we extract all feature matches, where both feature positions have a high posterior probability $\Pr[m|\mathbf{x}]$ (defined in Section 4.1). Specifically, we check if the product of two posterior probabilities (from two detected feature positions) exceeds 0.5.

Repetitive structures in man-made environments are usually equidistant in 3D. However, the equal spacing is not preserved in image space (and, hence, in our feature matches), due to perspective distortion. We undo this distortion by affinely rectifying the positions of the matched feature points. The displacement of two rectified feature points is now spatially invariant, and consequently, we can detect translational repetition: if certain regular structures exist, these displacement vectors form a dense cluster in the 2D affine rectified space. We use the mean-shift algorithm [Comaniciu and



(a) Input image

(b) Matched features

(c) Rectified space modes

(d) Fronto-parallel modes

(e) Rectified displacements

(f) Fronto-parallel displacements

Figure 6: *Detecting regularity from modes of displacement vectors between matched features. (a) and (b) show the input image and matched features. (c) and (e) (on the left) show the detected modes and a visualization of candidate source patch locations and shapes for a single target patch (in white) in the* affine rectified space. *The repetitive structure is clearly revealed. (d) and (f) show the corresponding illustrations for the fronto-parallel space. Here, the displacements are scattered and do not reveal any structure.*

Meer 2002] to detect these modes (setting the bandwidth parameter to 10 pixels, and rejecting spurious modes with fewer than 10 members). We denote the set of the modes as $\mathscr{D}_m = \{\mathbf{d}_i\}$, where $\mathbf{d}_i \in R^2$ is the displacement vector in the rectified space.

Figure 6 (c-d) shows detected modes in both rectified (left) and axis-aligned space (right). In (e-f) we also show their positions relative to a target patch in the image (white square). This figure highlights the importance of having perspective correction in computing the displacement vectors. In addition to the accurate position suggestion, the plane parameters explicitly provides how the source patches should deform spatially. The recovered candidates, outlined in blue, have to be significantly deformed to match with the image axis aligned target patch, in white. It is difficult to recover such geometric transformations using low-level algorithms alone.

He and Sun [2012] have shown that regularity using statistics of matched patch offsets can be helpful in the context of image completion. However, they assume global translational regularity in the image plane, i.e., they assume a single fronto-parallel surface. Detection and optimization are both done in image space. For our technique, while the detection is in rectified affine space, the objective function is optimized in image space using constrained homographies (as described in the next section).

Our regularity detection handles more general scenes because we deal with each plane independently (but with soft membership). As a result, we are able to detect different repetitive structures on multiple building facades with different orientations. Given the major differences, He and Sun's technique will have to be substantially modified to work on such scenes.

5 Guided Image Completion

In this section, we describe how the detected planes and extracted regularity from the previous section are used to guide the low-level image completion algorithm. We build on Wexler et al.'s algorithm [2007] using random search and propagation as in PatchMatch [Barnes et al. 2009]. Please refer to these papers for details on the base algorithm.

We incorporate sampling from planes by modifying the patch distance function (Sections 5.1-5.3), and the regularity by modifying the random sample generation (Section 5.4).

5.1 Objective Function

We augment the image completion objective function in two ways: First, we augment the patch distance (called "coherence measure" in the original paper [2007]) by including a guidance term. Second, we augment the search space by the plane index, which determines the patch transformation.

The objective function takes the form

$$\min_{\{\mathbf{t}_i,\mathbf{s}_i,m_i\}} \sum_{i\in\overline{\Omega}} E_{\text{color}}(\mathbf{s}_i,\mathbf{t}_i,m_i)+E_{\text{guide}}(\mathbf{s}_i,\mathbf{t}_i,m_i), \tag{3}$$

where Ω and $\overline{\Omega}$ are the sets of known and unknown pixel indices, $\mathbf{t}_i = (t_i^x, t_i^y)^\top$ is the center position of a target patch in $\overline{\Omega}$, $\mathbf{s}_i = (s_i^x, s_i^y)^\top$ is the center position of the corresponding source patch in Ω, and m_i the plane index of an unknown target patch $\mathbf{t}_i$. The two terms E_{color} and E_{guide} are the appearance and guidance terms, respectively, which together make up the patch distance.

Note that the target patches are image-aligned with no geometric transformation such as scaling or rotation, while the source patches have a geometric transform that is implicitly derived from geometry of the plane they are sampled from. The geometric transform is described in the next section.

5.2 Appearance Cost

Our appearance cost is the sum of the absolute values of two sampled patches in the RGB space:

$$E_{\text{color}}(\mathbf{s}_i,\mathbf{t}_i,m_i) = \left\| q(\mathbf{s}_i,\mathbf{t}_i,m_i) - p(\mathbf{t}_i) \right\|_1, \tag{4}$$

where $p(\mathbf{t}_i)$ denotes the 7×7 patch sampled around the center position $\mathbf{t}_i$ and $q(\mathbf{s}_i,\mathbf{t}_i,m_i)$ denotes the sampled patch centered at $\mathbf{s}_i$ with geometric transformation subject to the plane orientation defined by target patch position $\mathbf{t}_i$ and the plane parameter of plane m_i.

Most prior approaches use pure translational patches [Wexler et al. 2007; Komodakis and Tziritas 2007] or explicitly search geometric transformations, e.g., rotation, scale, and flip [Mansfield et al. 2011;

Darabi et al. 2012]. Instead, we sample patches using homographies. Rather than searching for all parameters of the homography, we derive it implicitly from the combination of the coordinates $\mathbf{s}_i, \mathbf{t}_i$ with their corresponding plane index m_i.

We first compute the transformation that maps a 7×7 patch at $\mathbf{t}_i$ to transformed patch sampled at $\mathbf{s}_i$. Let $\tilde{\mathbf{t}_i} = [t_i^x, t_i^y, 1]^\top$ and $\tilde{\mathbf{s}_i} = [s_i^x, s_i^y, 1]^\top$ as homogenous representations of $\mathbf{t}_i$ and $\mathbf{s}_i$, respectively. Let $\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3$ be the row vectors of $\mathbf{H}_{m_i}$. The source and target patch positions in the affine rectified space are computed as:

$$\tilde{\mathbf{t}_i}' = \left[\mathbf{h}_1\tilde{\mathbf{t}_i},\ \mathbf{h}_2\tilde{\mathbf{t}_i},\ \mathbf{h}_3\tilde{\mathbf{t}_i}\right]^\top, \tag{5}$$

$$\tilde{\mathbf{s}_i}' = \left[\mathbf{h}_1\tilde{\mathbf{s}_i},\ \mathbf{h}_2\tilde{\mathbf{s}_i},\ \mathbf{h}_3\tilde{\mathbf{s}_i}\right]^\top. \tag{6}$$

We define (d^x, d^y) as the displacement vector from target to source patch positions in the rectified space. The term $\tilde{\mathbf{s}_i}$ is represented as

$$\tilde{\mathbf{s}_i}' = \begin{bmatrix} \mathbf{h}_1 + \mathbf{h}_3 d^x \\ \mathbf{h}_2 + \mathbf{h}_3 d^y \\ \mathbf{h}_3 \end{bmatrix} \tilde{\mathbf{t}_i}. \tag{7}$$

By applying the inverse of the rectifying matrix $\mathbf{H}_{m_i}^{-1}$, we have

$$\tilde{\mathbf{s}_i} = \mathbf{H}_{m_i}^{-1}\tilde{\mathbf{s}_i}' = \mathbf{H}_{m_i}^{-1} \begin{bmatrix} \mathbf{h}_1 + \mathbf{h}_3 d^x \\ \mathbf{h}_2 + \mathbf{h}_3 d^y \\ \mathbf{h}_3 \end{bmatrix} \tilde{\mathbf{t}_i}. \tag{8}$$

To get the motion parameters of the patch around $\mathbf{s}_i$ (i.e., factoring out the dependency of $\mathbf{t}_i$), we apply a translation matrix with offset $\mathbf{t}_i$:

$$\tilde{\mathbf{s}_i} = \mathbf{H}_{m_i}^{-1} \begin{bmatrix} \mathbf{h}_1 + \mathbf{h}_3 d^x \\ \mathbf{h}_2 + \mathbf{h}_3 d^y \\ \mathbf{h}_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_i^x \\ 0 & 1 & t_i^y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{T}_{\mathbf{s}_i} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{9}$$

where $\mathbf{T}_{\mathbf{s}_i}$ compactly represents the domain transformation of the sampled source patch. Note that in the special case of fronto-parallel plane ($\mathbf{H}_0 = \mathbf{I}_3$), $\mathbf{T}_{\mathbf{s}_i}$ reduces to a translation matrix with offset (s_i^x, s_i^y).

5.3 Guidance Cost

Our guidance cost includes three constraints derived from the analysis stage:

$$\begin{aligned} E_{\text{guide}}(\mathbf{s}_i, \mathbf{t}_i, m_i) \ = {} & \lambda_1 E_{\text{plane}}(\mathbf{s}_i, \mathbf{t}_i, m_i) + \lambda_2 E_{\text{direction}}(\mathbf{s}_i, \mathbf{t}_i, m_i) + \\ & \lambda_3 E_{\text{proximity}}(\mathbf{s}_i, \mathbf{t}_i), \end{aligned} \tag{10}$$

where $\lambda_1 = 10$, $\lambda_2 = 10^3$, and $\lambda_3 = 1$ are the weighting parameters for plane compatibility, orthogonal direction, and proximity cost, respectively. Next, we describe each of these constraints in detail.

Plane compatibility. In the analysis stage, we computed the posterior probability map $\Pr[m_i|\mathbf{x}]$ for assigning plane membership m_i for position located at $\mathbf{x}$. We directly convert this into a penalty term using the negative log-likelihood. Specifically,

$$E_{\text{plane}}(\mathbf{s}_i, \mathbf{t}_i, m_i) = -\log \Pr[m_i|\mathbf{s}_i] - \log \Pr[m_i|\mathbf{t}_i], \tag{11}$$

i.e., the term encourages sampling from a plane that has a high probability both in the source and target location.

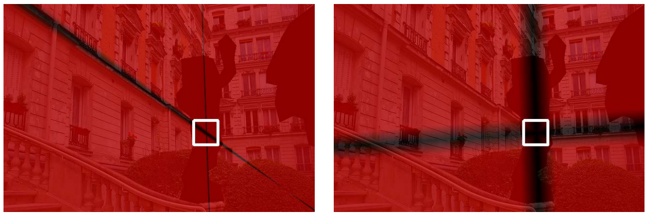


Figure 7: *Visualization of the directional cost given a target patch (shown in white). The dark regions indicate lower costs. The directional cost encourage to find matches along the two dominant, orthogonal directions, leading to semantically more plausible completion results.*

Orthogonal direction cost. Urban scenes often consist of repetitive structures along horizontal and vertical directions, e.g., windows on a building facade. This term encourages using source patches located on either one of the orthogonal directions. Note that affine rectification makes the support lines for each VP parallel, however, the lines from the two VPs are not necessarily orthogonal to each other. We estimate the rotation angle that maps the set of line segments for each VP to align with the horizontal axis. This mapping is denoted as $\mathbf{H}_{m_i}^1$ and $\mathbf{H}_{m_i}^2$ for the two VPs defining the plane m_i:

$$\mathbf{H}_{m_i}^j = \begin{bmatrix} cos(\theta_j) & -sin(\theta_j) & 0 \\ sin(\theta_j) & cos(\theta_j) & 0 \\ l_1^{m_i} & l_2^{m_i} & l_3^{m_i} \end{bmatrix}. \tag{12}$$

We define the orthogonal direction cost as a truncated L1-norm:

$$\begin{aligned} E_{\text{direction}}(\mathbf{s}_i, \mathbf{t}_i, m_i) = \psi\big(& \min(|\mathbf{H}_{m_i}^1(\tilde{\mathbf{s}_i})^y - \mathbf{H}_{m_i}^1(\tilde{\mathbf{t}_i})^y|, \\ & |\mathbf{H}_{m_i}^2(\tilde{\mathbf{s}_i})^y - \mathbf{H}_{m_i}^2(\tilde{\mathbf{t}_i})^y|)\big), \end{aligned} \tag{13}$$

where $\psi(z) = min(|z|, c)$ is the function that caps the cost to a constant $c = 0.02$. To ensure that the cost is invariant to the scale of the image, we divide the distances in y-axis in the rectified space by the largest image dimension. For cases of target patches with no available source samples on the both directions, this constraint has no effect on searching of the source patch because it is constant.

Proximity cost. It has been shown by Kopf et al. [2012] that constraining the search space to nearby regions can improve the synthesis result. In addition to the above mid-level constraints, we also introduce a low-level search space constraint which favors nearby source patches for completion. This constraint implicitly avoid copying patches from extremely different scales. We define the proximity cost as

$$E_{\text{proximity}}(\mathbf{s}_i, \mathbf{t}_i) = \frac{||\mathbf{s}_i - \mathbf{t}_i||_2^2}{\sigma_d(\mathbf{t}_i)^2 + \sigma_c^2}, \tag{14}$$

where $\sigma_d(\mathbf{t}_i)^2$ is the squared distance of target position to the nearest border to the known region and $\sigma_c^2 = (W/8)^2$ is the parameter for adjusting the strength of the proximity constraint (W is the largest image dimension).

5.4 Structure Guided Sampling and Propagation

We extend the random location sampling in the PatchMatch algorithm [Barnes et al. 2009] to incorporate our computed plane probabilities and translational regularity. In addition to the regular random location sampling, we also sample from the clustered regularity modes computed in Section 4.2. We did not include the regularity as a prior term in the previous sections because the detection is sometimes not reliable. Using regularity instead for random location sample generation provides a more robust way of incorporating this constraint. In our implementation of PatchMatch, we use 5 iterations of *plane probability guided* sampling and *regularity guided* sampling (as described below) in the search and propagation stage.

Plane probability guided sampling. For a given target patch $\mathbf{t}_i$, we first sample the plane index m_i according to the posterior probability $\Pr[m_i|\mathbf{t}_i]$. Then, we sample $\mathbf{s}_i$ by drawing random samples from $\Pr[\mathbf{s}_i|m_i]$ using rejection sampling. This effectively biases the search space toward finding the correct patches from the same plane.

Regularity guided sampling. While the plane probability guided random sampling scheme to sample from the right plane, it does not impose constraints on where on the plane it should sample from. This usually leads to visible artifacts when regular structures are present. Sampling from our detected regularity modes alleviates this problem.

For each target patch $\mathbf{t}_i$, we first draw a plane index m_i as above, then, we randomly draw one displacement in rectified space from $\mathscr{D}_m$. Using the target patch position $\mathbf{t}_i$, the plane index m_i, and the displacement in the rectified space, we can then directly compute the candidate positions and their geometric transformations. Examples of candidate source patches are shown in Figure 6. This regularity guided sampling scheme greatly improves the completion quality when repetitive structures exist.

6 Results

We compare our results against several state-of-the-art image completion algorithms. Specifically, we choose Photoshop Content Aware Fill [Barnes et al. 2009; Wexler et al. 2007], He and Sun's method [2012], and Image Melding [Darabi et al. 2012]. In the supplementary material we also compare against Priority Belief Propagation [Komodakis and Tziritas 2007] and GIMP Resynthesizer[2]. All of these methods use fronto-parallel translational patches, except Image Melding, which allows similarity transformation and flip of patches.

6.1 Comparison with the State-of-the-Art Methods

In Figure 8, we show a series of comparisons on challenging scenes. In the first row, the building consist of near regular structures. We can see that the competing algorithms fail to synthesize such large structure because they only minimize localized texture energy without considering a global consistency. Our method, on the other hand, fills in the hole with repetitive pattern similar to the known region. In addition, with the recovered plane orientation, our synthesized result is physically plausible.

In the second row, we show a single planar building facade with regular patterns. Even with only mild perspective distortion, translational patches are insufficient to synthesize the foreshortening effect and thus result in broken line structures. Image Melding, while theoretically equipped with the ability to apply appropriate scaling of the patches, fails to find such solution in high-dimensional space. Our algorithm effectively use the plane constraint to extend the facade with minimally visible artifacts. The results on the 3rd to 6th rows show that our algorithm is *not* limited to ideal piecewise plane scenes with homogeneous textures. With the plane support detection and the weighting scheme, we leverage the low-level algorithm to find a good transition boundary between one structure and another. Examples illustrating good transition boundaries are the stairs regions and the pure texture region around the tree in the fourth row, and multiple unknown surface discontinuity in the fifth row. In the last row, we demonstrate the effectiveness of combining plane constraints and regularity-guided sampling.

From these examples of realistic scenes, we can see that our image completion algorithm is robust to deviations from perfectly textured planar surfaces. In other words, our completion algorithm does not require perfect plane orientation recovery, support estimation, segmentation, and symmetry detection. In fact, the analysis in many regions contains errors because vision algorithms are far from perfect. However, as exemplified here, by combining a powerful low-level algorithm with mid-level constraints, we are able to extend the state-of-the-art in image completion. Please refer to the supplementary materials where we show extensive comparison results on a variety of scenes.

6.2 Comparisons on Natural Scenes

For images of natural scenes our analysis usually does not detect any planes because there are no reliable features to detect planes and translational regularity. In such cases our our algorithm automatically reverts to the baseline image completion algorithm, i.e., our implementation of Wexler et al.'s algorithm [2007]. Four such examples are shown in Figure 9. We compare to the unguided version of our completion algorithm (fourth column) to validate that our result looks visually similar to the baseline. In the supplementary material we present a more extensive comparison on 25 natural images we extracted from the project website of Kopf et al.'s paper [2012].

6.3 Failure Modes

We used relatively simple algorithms in our image analysis stages, which can fail to detect vanishing points or plane regularities, or more severely, return false positives. In the former case our algorithm just reverts to fronto-parallel completion, while the latter case might lead to some artifacts. The performance of the analysis stage could likely be improved using more sophisticated computer vision methods, which we leave to future work.

The first two rows in Figure 10 demonstrate demonstrate the difficulty of finding demarcation lines between different perspective planes when the unknown region is large. The results in the third row shows an example that the falsely detected plane may over-constrain the patch synthesis and lead to poor results near the bushes. Notice, though, that the competing techniques also fail to generate satisfactory results.

7 Concluding Remarks

We have presented an automatic image completion algorithm that exploits extracted mid-level scene structures for guiding the low-level completion. Our algorithm detects multiple planes and their corresponding translational regularity. These constraints are incorporated into the augmented patch distance and the sampling scheme. In the absence of reliable plane detection, our algorithm automatically reverts to a baseline completion algorithm. We demonstrated that our method consistently outperforms state-of-the-art image completion algorithms for a wide range of challenging scenes.

Historically, conventional statistical texture synthesis methods formulate texture synthesis as an "analysis then synthesis" framework. However, this type of framework has been mostly set aside due to the simplicity and the effectiveness of example-based methods. Our

[2] `http://www.logarithmic.net/pfh/resynthesizer`

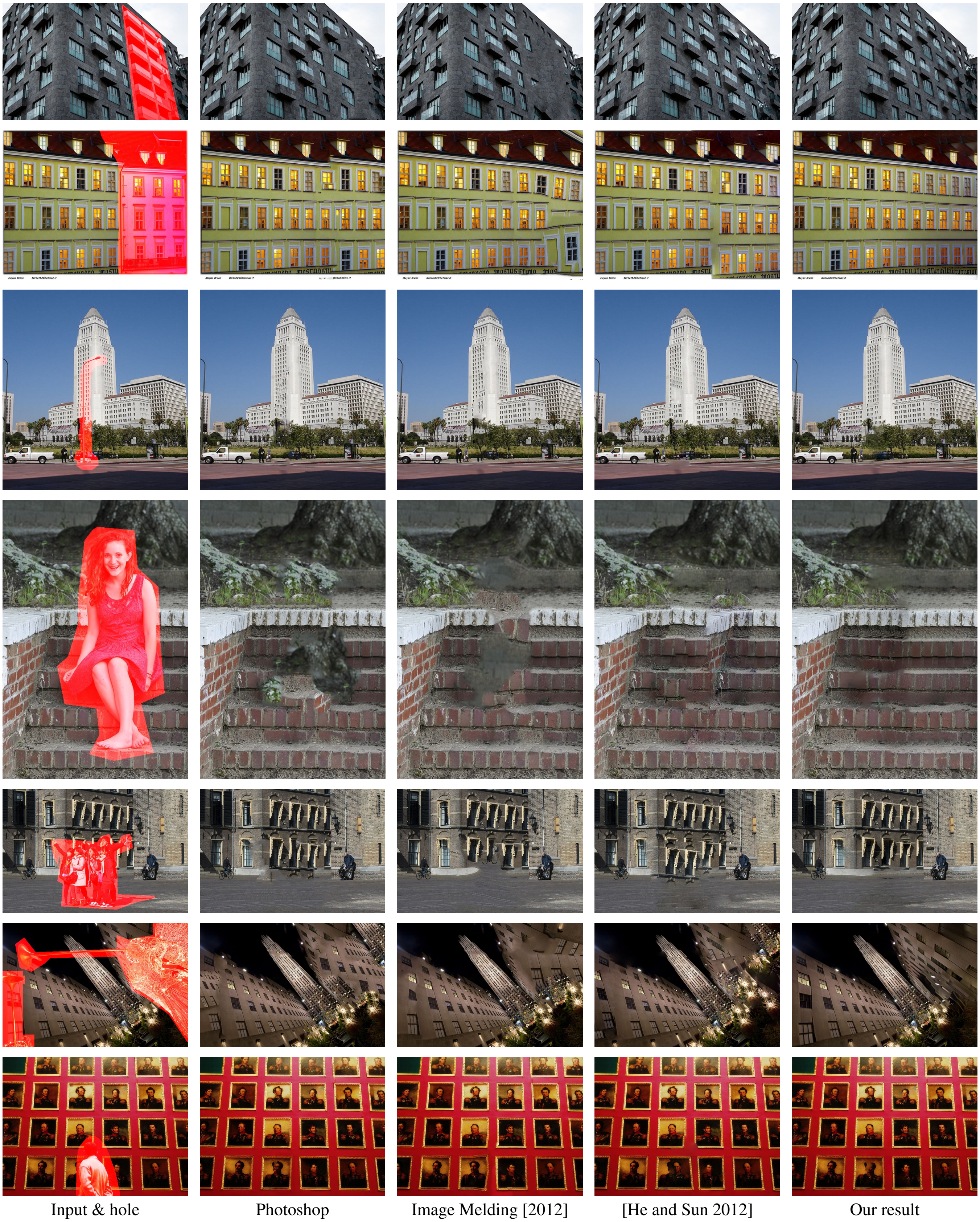


Figure 8: *Comparisons with representative state-of-the art algorithms. Please refer to the supplementary material for a more extensive comparison on over 80 images and against a wider selection of existing algorithms. Image credits: Flickr users Daniel Foster, Moyan Brenn, Savannah Roberts, Remon Rijper, Chris Ford, and marie-ll.*

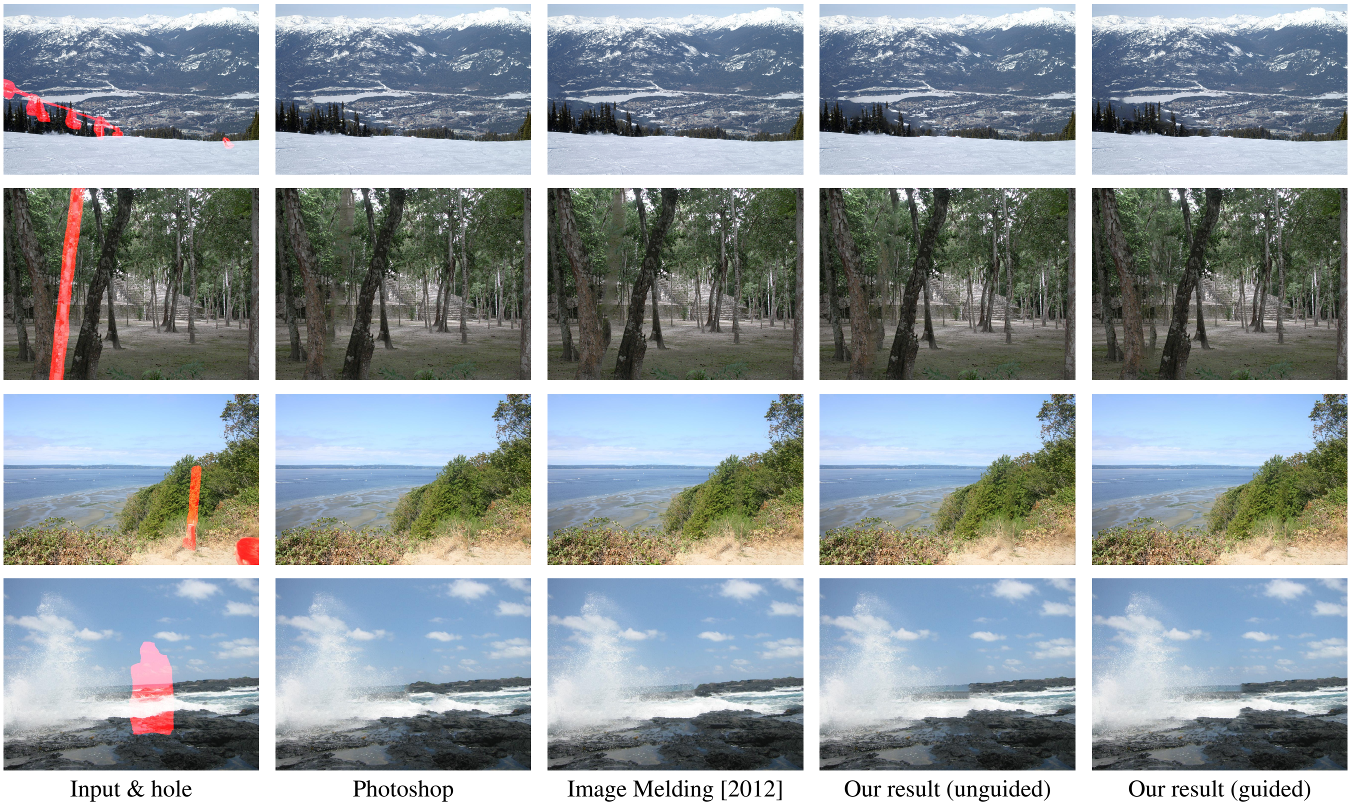


Figure 9: *Image completion on images of natural scenes.*

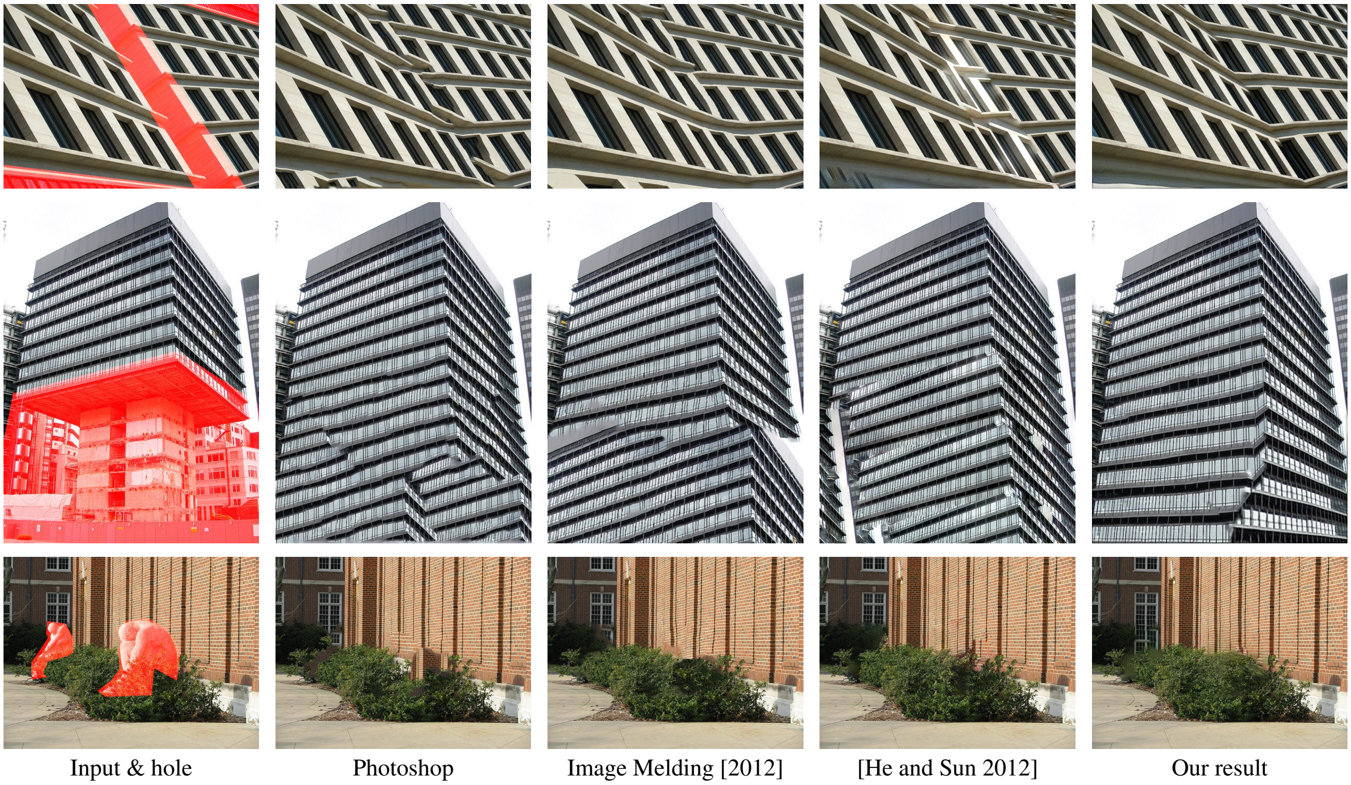


Figure 10: *Failure examples. First two rows: our algorithm has difficulty in finding the good demarcation lines when the missing regions are too large. The last row: our method may overconstrain the patch synthesis with falsely-detected planes, which leads to artifacts near the bushes. Image credits: Flickr users Reto Fetz, David Barrie, and brokenthoughts.*

method demonstrates the benefit and the need of image analysis. We show that the quality of image completion can be significantly improved by striking a balance between analysis and synthesis.

Acknowledgements

We thank the flickr users who put their images under Creative Commons license or allowed us to use them. For a detailed list of contributors to our image dataset, please refer to the accompanying project website. The support of the Office of Naval Research under grant N00014-12-1-0259 is gratefully acknowledged.

References

Aiger, D., Cohen-Or, D., and Mitra, N. J. 2012. Repetition maximization based texture rectification. *Computer Graphics Forum (EUROGRAPHICS) 31*, 2pt2, 439–448.

Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G., and Verdera, J. 2001. Filling-in by joint interpolation of vector fields and gray levels. *IEEE TIP 10*, 8, 1200–1211.

Barinova, O., Konushin, V., Yakubenko, A., Lee, K., Lim, H., and Konushin, A. 2008. Fast automatic single-view 3-d reconstruction of urban scenes. In *ECCV*.

Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. 2009. PatchMatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. on Graphics (Proc. of Siggraph) 28*, 3, 24.

Bertalmio, M., Sapiro, G., Caselles, V., and Ballester, C. 2000. Image inpainting. *ACM Trans. on Graphics (Proc. of Siggraph) 19*, 3, 417–424.

Bertalmio, M., Vese, L., Sapiro, G., and Osher, S. 2003. Simultaneous structure and texture image inpainting. *IEEE TIP 12*, 8, 882–889.

Chum, O., and Matas, J. 2010. Planar affine rectification from change of scale. In *ACCV*.

Comaniciu, D., and Meer, P. 2002. Mean Shift: A robust approach toward feature space analysis. *IEEE TPAMI 24*, 5, 603–619.

Criminisi, A., Pérez, P., and Toyama, K. 2004. Region filling and object removal by exemplar-based image inpainting. *IEEE TIP 13*, 9, 1200–1212.

Darabi, S., Shechtman, E., Barnes, C., Goldman, D. B., and Sen, P. 2012. Image Melding: Combining Inconsistent Images using Patch-based Synthesis. *ACM Trans. on Graphics (Proc. of Siggraph) 31*, 4.

Efros, A. A., and Freeman, W. T. 2001. Image quilting for texture synthesis and transfer. *ACM Trans. on Graphics (Proc. of Siggraph) 20*, 3, 341–346.

Efros, A. A., and Leung, T. K. 1999. Texture synthesis by non-parametric sampling. In *ICCV*.

Hartley, R. I., and Zisserman, A. 2004. *Multiple View Geometry in Computer Vision*, second ed. Cambridge University Press.

Hays, J., and Efros, A. A. 2007. Scene completion using millions of photographs. *ACM Trans. on Graphics (Proc. of Siggraph) 26*, 3, 4.

He, K., and Sun, J. 2012. Statistics of patch offsets for image completion. In *ECCV*.

Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B., and Salesin, D. H. 2001. Image analogies. *ACM Trans. on Graphics (Proc. of Siggraph) 20*, 3, 327–340.

Huang, H., K.Yin, Gong, M., Lischinski, D., Cohen-Or, D., Ascher, U., and Chen, B. 2013. Mind the gap: Teleregistration for structure-driven image completion. *ACM Trans. on Graphics (Proc. of Siggraph Asia) 32*, 174:1–174:10.

Huang, J.-B., Kopf, J., Ahuja, N., and Kang, S. B. 2013. Transformation guided image completion. In *ICCP*.

Jia, J., and Tang, C. 2003. Image repairing: Robust image synthesis by adaptive nd tensor voting. In *CVPR*.

Komodakis, N., and Tziritas, G. 2007. Image completion using efficient belief propagation via priority scheduling and dynamic pruning. *IEEE TIP 16*, 11, 2649–2661.

Kopf, J., Kienzle, W., Drucker, S., and Kang, S. B. 2012. Quality prediction for image completion. *ACM Trans. on Graphics (Proc. of Siggraph Asia) 31*, 6.

Kwatra, V., Essa, I., Bobick, A., and Kwatra, N. 2005. Texture optimization for example-based synthesis. *ACM Trans. on Graphics (Proc. of Siggraph) 24*, 3, 795–802.

Liu, Y., Lin, W.-C., and Hays, J. 2004. Near-regular texture analysis and manipulation. *ACM Trans. on Graphics (Proc. of Siggraph) 23*, 3, 368–376.

Liu, Y., Hel-Or, H., and Kaplan, C. 2010. *Computational symmetry in computer vision and computer graphics*. Now Publishers.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2, 91–110.

Mansfield, A., Prasad, M., Rother, C., Sharp, T., Kohli, P., and Van Gool, L. 2011. Transforming image completion. In *BMVC*.

Pavić, D., Schönefeld, V., and Kobbelt, L. 2006. Interactive image completion with perspective correction. *The Visual Computer 22*, 9, 671–681.

Pritch, Y., Kav-Venaki, E., and Peleg, S. 2009. Shift-map image editing. In *ICCV*.

Sun, J., Yuan, L., Jia, J., and Shum, H. 2005. Image completion with structure propagation. *ACM Trans. on Graphics (Proc. of Siggraph) 24*, 3, 861–868.

Wexler, Y., Shechtman, E., and Irani, M. 2007. Space-time completion of video. *IEEE TPAMI 29*, 3, 463–476.

Whyte, O., Sivic, J., and Zisserman, A. 2009. Get out of my picture! internet-based inpainting. In *BMVC*.

Zhang, Z., Ganesh, A., Liang, X., and Ma, Y. 2012. TILT: transform invariant low-rank textures. *International Journal of Computer Vision 99*, 1, 1–24.

Zhang, Y., Xiao, J., Hays, J., and Tan, P. 2013. FrameBreak: Dramatic image extrapolation by guided shift-maps. In *CVPR*.