

Geometrically Consistent Stereoscopic Image Editing using Patch-based Synthesis

Sheng-Jie Luo, Ying-Tse Sun, I-Chao Shen, Bing-Yu Chen, *Senior Member, IEEE*, and Yung-Yu Chuang, *Member, IEEE*

Abstract—This paper presents a patch-based synthesis framework for stereoscopic image editing. The core of the proposed method builds upon a patch-based optimization framework with two key contributions: First, we introduce a depth-dependent patch-pair similarity measure for distinguishing and better utilizing image contents with different depth structures. Second, a joint patch-pair search is proposed for properly handling the correlation between two views. The proposed method successfully overcomes two main challenges of editing stereoscopic 3D media: (1) maintaining the depth interpretation, and (2) providing controllability of the scene depth. The method offers patch-based solutions to a wide variety of stereoscopic image editing problems, including depth-guided texture synthesis, stereoscopic NPR, paint by depth, content adaptation, and 2D to 3D conversion. Several challenging cases are demonstrated to show the effectiveness of the proposed method. The results of user studies also show that the proposed method produces stereoscopic images with good stereoscopicity and visual quality.

Index Terms—stereoscopic images, patch-based synthesis

1 INTRODUCTION

As stereoscopic 3D media is becoming popular, manipulating stereoscopic 3D images and videos becomes an important demand. Unlike conventional 2D media, editing stereoscopic 3D media allows the depth of the scene to be controllable (e.g., depth adjustment of an object) while maintaining stereopsis. Therefore, the main issues of editing stereoscopic 3D media are twofold. Firstly, the editing results should maintain depth interpretation established through stereopsis by human, which indicates that the corresponding points in the left and right views should be adjusted jointly. Secondly, objects with different depths should be handled separately so that they do not affect each other, implying the needs for depth-aware processing.

It is challenging to faithfully maintain the depth interpretation and provide controllability of the scene depth while editing stereoscopic 3D media. Recently, warping-based approaches [1], [2] have been introduced for editing stereoscopic images, especially for adapting the depth and the size of an image. These approaches place a pair of quad meshes onto the two views and then compute a pair of deformed meshes for satisfying editing constraints. They successfully maintain consistent depth interpretation by incorporating stereopsis constraints into the warping procedure. However, they could suffer from distortion

artifacts when there are significant depth variations within a quad, because all pixels in a quad have to undergo the same transformation despite their depth discrepancy. Therefore, these methods are ineffective for 3D editing problems which involves depth adjustment especially where there are multiple elements with different depths.

In this paper, we propose *StereoSyn* - a versatile and robust stereoscopic image synthesis method which synthesizes the stereoscopic images according to editing operations. To achieve this, patch-based approaches are chosen as the basis of the synthesis algorithm because they have been proven successful for image editing [3], [4]. Existing patch-based algorithms divide the source image into a collection of overlapping patches, and copy the patches to construct the target image while maintaining the coherence of nearby pixels. However, applying these algorithms to synthesize the left and right views individually would unlikely establish a consistent depth interpretation of the resultant stereoscopic image. In addition, these algorithms measure patch similarity using only color appearance. Therefore, they can only synthesize patches similar to the ones in the source image. However, due to occlusion and disocclusion after depth adjustment, it is often necessary to synthesize patches which are not similar to any of the input image for correctly modeling the spatial relationship change between objects.

This leads to two key observations of our method. Firstly, the correspondence information should be embedded into the synthesis process, and the regions that are visible in both views should be jointly synthesized for guaranteeing a consistent depth in-

-
- S.-J. Luo, Y.-T. Sun, B.-Y. Chen, and Y.-Y. Chuang are with National Taiwan University.
E-mail: {forestking, night533, robin, cyj}@cmlab.csie.ntu.edu.tw
 - I.-C. Shen is with Academia Sinica, Taipei, Taiwan.
E-mail: joshshen@citi.sinica.edu.tw

terpretation of the scene. Secondly, it is important to be aware of depth edges when measuring patch similarity and contributing to the final color. It is because a local window (i.e., the patch) often contains multiple elements with different depths. Without considering depth discrepancies within a patch, it is likely to mix up contributions from different elements. Based upon these observations, we develop a depth-dependent patch-pair similarity measurement which takes local depth discrepancies into consideration. Two patch-pairs are similar only if the corresponding pixels with similar depth structures have similar colors. Therefore, the proposed method is capable of dealing with the *spatial relation change* and *objects occlusion/disocclusion* caused by depth modification during stereoscopic image editing. Furthermore, to guarantee the depth interpretation of the stereoscopic images, a joint nearest-neighbor patch-pair search method is introduced to jointly synthesize the left and right views.

We successfully applied our algorithm to an extensive set of stereoscopic image editing applications which provide users with flexibility in specifying different configurations for adjusting the depths and layouts of the stereoscopic images. We also compared our results of depth/size adaptation to warping-based methods, and demonstrated that our method performs better than the state-of-the-art methods specifically designed for individual problems.

2 RELATED WORK

Stereoscopic media editing. Recently, editing of stereoscopic media has attracted lots of attentions. Many techniques have been proposed for specific editing applications. Stereoscopic image inpainting techniques have been proposed by Wang et al. [5] and Morse et al. [6]. Koppal et al. [7] proposed a technique that enables the manipulation of stereo parameters such as the interocular distance and location. The problem of stereoscopic copy and paste has been addressed by using a segmentation-based technique [8] and a gradient-domain technique [9]. Wang and Sawchuck [10] proposed a framework for disparity manipulation of stereoscopic media. Lang et al. [1] proposed a novel spatially varying warping technique to enable manipulation of the disparity range of stereoscopic videos. The warping-based approach is also applied to resolve stereoscopic image retargeting problem [2]. Basha et al. [11], [12] also addressed the stereoscopic image retargeting problem but used a geometrically consistent seam-carving algorithm with the concept of seam coupling for preserving the depth interpretation. More recently, Lee et al. proposed a layer-based stereoscopic image resizing method [13]. They adopted the layer-based idea from scene carving [14] but used warping to adjust layers with different depths separately for resizing a stereoscopic image. Niu et al. [15] proposed a method for

applying the user-specified warping on stereoscopic images. Northam et al. [16] proposed a stereoscopic 3D stylization method that guarantees consistency between the left and right views. However, their method would produce visible layering artifacts in the stylized results because it stylizes different depth layers separately.

Our patch-based method is versatile and can be used in a broader set of interesting stereoscopic image editing problems, such as transferring the stereoscopic texture and editing object shapes by manipulating their depths.

Patch-based image synthesis. Patch-based technique has become popular for image and video synthesis. They were branched from non-parametric texture synthesis [17], [18], [19], [20], [21], [3] which samples patches from the input texture example and pastes them into the output image while maintaining the coherence of nearby pixels in the synthesized result. Recently, the patch-based methods have also been applied to image editing problems, and we will next review some of them.

In early period of patch-based technique development, researchers focused on the texture synthesis problem due to large interests in computer graphics community. Efros and Leung [17] proposed a non-parametric texture synthesis method. The structure preservation problem was later addressed by modifying the search and sampling strategies [18]. Then, Kwatra et al. [3] developed a synthesis procedure based on global optimization for obtaining more consistent synthesized results with respect to the input textures. Patch-based methods have also been proven effective for many 2D natural image editing problems, such as image inpainting [22], [23], [24]. Simakov et al. [25] introduced a bidirectional similarity distance for summarizing an input natural image. Barnes et al. [26] introduced a randomized patch-search method to accelerate the synthesis procedure. Darabi et al. [27] proposed a generalized method for combining two or more images with inconsistent structures.

Although patch-based methods are effective for 2D image editing, they cannot be directly applied to stereoscopic 3D image editing because neither joint synthesis nor local depth discrepancies were addressed in these methods. Therefore, the synthesized results are not guaranteed to maintain a consistent depth interpretation of the scene (Fig. 1(b)). In contrast, our approach samples patch-pairs jointly from both views and can synthesize the result with a satisfactory depth interpretation (Fig. 1(c)). Moreover, existing methods cannot handle local layout changes caused by depth adjustments. Our approach exploits a depth-dependent patch-pair similarity measure, and thus can deal with the cases in which pixels within a patch undergo different transformations after depth

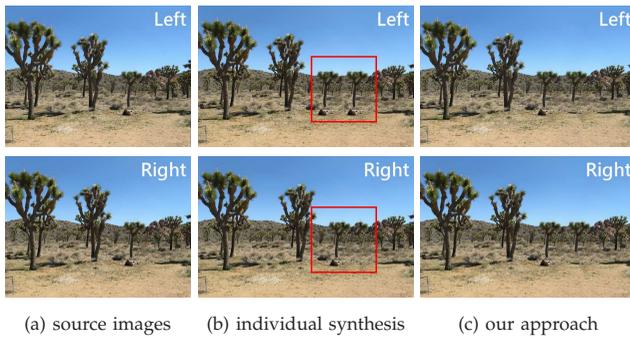


Fig. 1. A comparison to individual synthesis. The source stereoscopic image pair (a) is enlarged using [25] for each view individually (b) and our joint synthesis approach (c).

adjustment.

3 STEREO SYN ALGORITHM

Our *StereoSyn* framework is designed for manipulating the depth and layout of a pair of stereoscopic images. To address the above-mentioned issues of stereoscopic image editing, the objectives of our algorithm are to maintain a consistent depth interpretation of the stereoscopic images, and to separately synthesize objects with different depths in the scene.

The input is a pair of rectified source stereoscopic images. In order to obtain the depth information and to associate corresponding pixels in the two views, we first construct the disparity maps for the source image pair, and then generate a pair of visibility maps indicating whether a pixel is visible in both views or in only one view. Depending on the editing goals, the target disparity maps could be manually specified by users or automatically constructed via synthesis [25] for guaranteeing that the layouts of the target disparity maps are similar to those of the source disparity maps. Specifically, for applications in which the target depth information is based on user intention, such as depth-guided texture synthesis, paint by depth, and 2D to 3D conversion, the target disparity map is specified by users and given as the input. On the other hand, for applications where the target depth information can be inferred from the source depth information, such as content adaptation, the target disparity map is automatically generated using a 2D synthesis method [25].

Based on the target disparity maps, the target visibility maps are constructed accordingly. The four pairs of maps (source images, source disparity, visibility and target disparity) are used in the synthesis procedure, which performs patch-based optimization taking into account the disparity and visibility maps. Based on the visibility maps, a joint nearest-neighbor patch-pair search method is introduced so that the regions visible in both views can be synthesized jointly to maintain a consistent depth interpretation

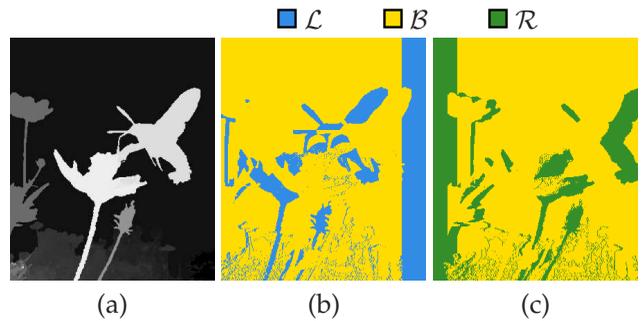


Fig. 2. An illustration of visibility maps. (a) The left disparity map. (b) & (c) the left and right visibility maps. The blue (\mathcal{L}) / green (\mathcal{R}) regions are the pixels only visible in the left / right view, and the yellow regions (\mathcal{B}) are visible in both views.

of the two views. In addition, a depth-dependent patch-pair similarity measure is incorporated into the optimization process such that each patch performs self-segmentation according to its local depth discrepancies while contributing to the final results.

Here we first describe how we obtain the disparity and visibility maps (Sec. 3.1). Then we introduce our optimization-based stereoscopic image synthesis process (Sec. 3.2) and depth-dependent patch-pair similarity measure (Sec. 3.3). Finally we describe the solver used to obtain the final results and also the joint nearest-neighbor patch-pair search mechanism (Sec. 3.4).

3.1 Disparity and visibility map construction

Given a rectified stereoscopic image pair (I^l, I^r), in this step, their associated disparity map pair and visibility map pair are constructed.

Disparity map. To generate the disparity map pair (D^l, D^r), we begin with estimating the disparity map D^l associated with the left view I^l using stereo matching [28], where the disparity value d is defined as $d = p'_x - p_x$ for a pixel pair $(p, p') \in (I^l, I^r)$. To avoid potential inconsistency between the left and right views, instead of estimating the disparity map for the right view, we directly map the disparity values in the left disparity map to the right one, and fill the missing disparities due to occlusions using a segmentation-based disparity filling approach [5]. Note that, although the estimated disparity maps may not be perfect, experiments show that our method can tolerate modest inaccuracy within disparity maps and still produce visually plausible results.

Visibility map. Analysis of visibility is critical in our framework. Although there exist more sophisticated approaches such as the learning-based occlusion analysis technique proposed by Humayun et al. [29], we adopt a light-weight rule-based approach to analyze the visibility for stereoscopic images.

The objective of the visibility maps (V^l, V^r) is to indicate whether a pixel is visible in both views or

only in a specific view. A visibility map V classifies every pixel p into one of the three classes $\{\mathcal{L}, \mathcal{R}, \mathcal{B}\}$ which indicates that p is visible from the left, right, or both views, respectively (Fig. 2). A pixel $p = (p_x, p_y)$ in the left image is marked as \mathcal{B} if satisfying the following two criteria:

$$\begin{aligned} 0 \leq p_x + D^l(p) < w \quad \text{and} \\ \forall q(q_x > p_x) \wedge (q_y = p_y), p_x + D^l(p) \neq q_x + D^l(q), \end{aligned} \quad (1)$$

where w denotes the width of the image, and $q = (q_x, q_y)$ represents a pixel on the right of p on the same horizontal line. The first condition verifies that p stays within the image bound, and the second one ensures that p is not occluded in the right view. If p is labeled as \mathcal{B} , its corresponding pixel in the right view is also labeled as \mathcal{B} . The remaining pixels in the left and right images are marked as \mathcal{L} and \mathcal{R} , respectively. According to the visibility maps, we can partition the stereoscopic image pair into four non-overlapping regions: $\Omega^{\mathcal{L}} / \Omega^{\mathcal{R}}$ represents the pixels in the left / right image labeled as $\mathcal{L} / \mathcal{R}$; $\Omega^{\mathcal{B}^l} / \Omega^{\mathcal{B}^r}$ represents the pixels in the left / right image labeled as \mathcal{B} , where $I^l = \Omega^{\mathcal{L}} \cup \Omega^{\mathcal{B}^l}$ and $I^r = \Omega^{\mathcal{R}} \cup \Omega^{\mathcal{B}^r}$.

3.2 Optimization-based stereoscopic image synthesis

Given a pair of source stereoscopic images (I_S^l, I_S^r) , our goal is to synthesize a pair of target stereoscopic images (I_T^l, I_T^r) while observing the user-specified editing properties. The objective of our stereoscopic image synthesis is to minimize the appearance differences of regions with similar local depth geometry between the result and input images. Therefore, the disparity and visibility maps are associated together to formulate the optimization problem via an energy function. Formally, the stereoscopic image pair and its associated maps are denoted by $\mathbf{I}^p = (\mathbf{I}^l, \mathbf{I}^r)$, where $\mathbf{I}^l = (I^l, D^l, V^l)$ and $\mathbf{I}^r = (I^r, D^r, V^r)$. As a result, for synthesizing the target stereoscopic images, we need to solve for $I_T^l, I_T^r, D_T^l, D_T^r, V_T^l,$ and V_T^r . The problem is complicated because these variables have dependencies which cannot be expressed as closed-form formulas: (1) D_T^r and (V_T^l, V_T^r) can be inferred from D_T^l as described in Sec. 3.1 and (2) D_T^l can be estimated from (I_T^l, I_T^r) . Furthermore, it is often preferable if users can have direct control to the target disparity map. Therefore, we opt to fix the target disparity maps after they are either manually provided by the users or automatically generated. Specifically, the algorithm synthesizes the target stereo image pair (I_T^l, I_T^r) given the source stereo image pair, the computed disparity and visibility maps and the target disparity and visibility maps by minimizing an energy function:

$$\arg \min_{(I_T^l, I_T^r)} E(\mathbf{I}_S^p, \mathbf{I}_T^p). \quad (2)$$

The energy function $E(\mathbf{I}_S^p, \mathbf{I}_T^p)$ is defined as:

$$E(\mathbf{I}_S^p, \mathbf{I}_T^p) = E_{\mathcal{L}}(\mathbf{I}_S^l, \mathbf{I}_T^l) + E_{\mathcal{R}}(\mathbf{I}_S^r, \mathbf{I}_T^r) + E_{\mathcal{B}}(\mathbf{I}_S^p, \mathbf{I}_T^p), \quad (3)$$

where $E_{\mathcal{L}}$, $E_{\mathcal{R}}$ and $E_{\mathcal{B}}$ are the similarities of the regions labeled as \mathcal{L} , \mathcal{R} , and \mathcal{B} , respectively. Pixels within different regions are measured and synthesized separately to better preserve the depth interpretation of the synthesized image.

In Eq.(3), $E_{\mathcal{L}}(\mathbf{I}_S^l, \mathbf{I}_T^l)$ and $E_{\mathcal{R}}(\mathbf{I}_S^r, \mathbf{I}_T^r)$ measure the similarity of the regions only visible in one view. We define $E_{\mathcal{L}}(\mathbf{I}_S^l, \mathbf{I}_T^l)$ as the sum of a completeness term and a coherence term:

$$\begin{aligned} E_{\mathcal{L}}(\mathbf{I}_S^l, \mathbf{I}_T^l) = & \frac{1}{|\Omega_S^{\mathcal{L}}|} \sum_{p_s \in \Omega_S^{\mathcal{L}}} \min_{p_t \in I_T^l} s(\mathbf{n}(p_s), \mathbf{n}(p_t)) \\ & + \frac{1}{|\Omega_T^{\mathcal{L}}|} \sum_{q_t \in \Omega_T^{\mathcal{L}}} \min_{q_s \in I_S^l} s(\mathbf{n}(q_t), \mathbf{n}(q_s)), \end{aligned} \quad (4)$$

where p and q are the sampled pixels; $\mathbf{n}(p)$ denotes the spatial neighborhood around a sample p , which is called a patch; and $s(\mathbf{n}(\cdot), \mathbf{n}(\cdot))$ defined in Eq.(6) is the distance between the two patches that will be discussed in Sec. 3.3. The completeness term encourages that all patches in the region $\Omega_S^{\mathcal{L}}$ should be represented in the output image I_T^l , and the coherence term encourages that all patches in the region $\Omega_T^{\mathcal{L}}$ should look similar to those in the input image I_S^l . To compute the similarity metric, for each input sample $p_s \in \Omega_S^{\mathcal{L}}$, we find the corresponding sample p_t with the most similar neighborhood in the output left image I_T^l and cumulate their distances, and vice versa. $E_{\mathcal{R}}$ is defined in a similar way.

For the regions labeled as \mathcal{B} , since they are visible in both views, we should perform a joint patch-pair search for maintaining the depth interpretation as:

$$\begin{aligned} E_{\mathcal{B}}(\mathbf{I}_S^p, \mathbf{I}_T^p) = & \frac{1}{|\Omega_S^{\mathcal{B}}|} \sum_{(p_s, p'_s) \in \bar{\Omega}_S^{\mathcal{B}}} \min_{(p_t, p'_t) \in \bar{\Omega}_T^{\mathcal{B}}} \bar{s}(\bar{\mathbf{n}}(p_s, p'_s), \bar{\mathbf{n}}(p_t, p'_t)) \\ & + \frac{1}{|\Omega_T^{\mathcal{B}}|} \sum_{(q_t, q'_t) \in \bar{\Omega}_T^{\mathcal{B}}} \min_{(q_s, q'_s) \in \bar{\Omega}_S^{\mathcal{B}}} \bar{s}(\bar{\mathbf{n}}(q_t, q'_t), \bar{\mathbf{n}}(q_s, q'_s)), \end{aligned} \quad (5)$$

where (p, p') and (q, q') stand for corresponding pixels in the left and right views, $\bar{\mathbf{n}}(p, p') = (\mathbf{n}(p), \mathbf{n}(p'))$ denotes the pair of corresponding patches, and $\bar{s}(\bar{\mathbf{n}}(\cdot, \cdot), \bar{\mathbf{n}}(\cdot, \cdot))$ defined in Eq.(8) is the distance between the two patch-pairs that will be discussed in Sec. 3.3. Specifically, Eq.(5) sums the local patch-pair distances. That is, for every local patch-pair $\bar{\mathbf{n}}(p_s, p'_s)$ in $\bar{\Omega}_S^{\mathcal{B}} = (\Omega_S^{\mathcal{B}^l}, \Omega_S^{\mathcal{B}^r})$, we search for the most similar patch-pair $\bar{\mathbf{n}}(p_t, p'_t)$ in $\bar{\Omega}_T^{\mathcal{B}} = (\Omega_T^{\mathcal{B}^l}, \Omega_T^{\mathcal{B}^r})$ and accumulate their distances, and do the same vice versa.

3.3 Depth-dependent patch/patch-pair similarity

The patch similarity metric is a core component for patch-based texture synthesis algorithms [30]. In conventional texture synthesis algorithms, the similarity

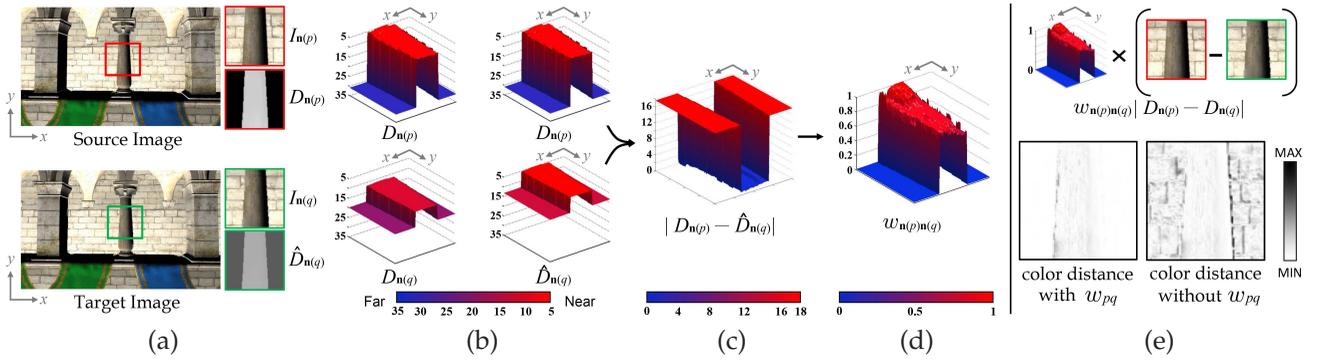


Fig. 3. The illustration of the depth-dependent color similarity term between patches $\mathbf{n}(p)$ and $\mathbf{n}(q)$. In this example, we compress the disparity range of the source image to synthesize the target image. (a) We focus on the two patches $\mathbf{n}(p)$ and $\mathbf{n}(q)$ from the source and target stereoscopic images. (Only one view is shown here.) $D_{\mathbf{n}(p)}$ and $D_{\mathbf{n}(q)}$ are their disparity maps. Note that the textures on the wall are different in the two images. (b) We obtain $\hat{D}_{\mathbf{n}(q)}$ by shifting $D_{\mathbf{n}(q)}$ so that the disparity value of the central pixel of $\mathbf{n}(q)$ equals to that of the central pixel of $\mathbf{n}(p)$. (c) The distance between $D_{\mathbf{n}(p)}$ and $\hat{D}_{\mathbf{n}(q)}$, $|D_{\mathbf{n}(p)} - \hat{D}_{\mathbf{n}(q)}|$ measures the local depth structure similarity for each pixel pair. (d) The depth-dependent weighting kernel w_{pq} is obtained according to $|D_{\mathbf{n}(p)} - \hat{D}_{\mathbf{n}(q)}|$. (e) The comparison of color distances with and without the depth-dependent weighting kernel.

of two patches is defined as the sum of squared distances (SSD) of corresponding pixels' colors within a window (i.e., the patch). For stereoscopic image synthesis, in addition to colors, we need to take 3D geometry differences between patches into account. A naive extension has been proposed by Morse et al. [6], which directly sums the SSD of both colors and depths together (we call it the *simple 4D metric*). However, it does not work well because a window could contain multiple layers of depths, and it does not model the depth adjustment.

In order to separately synthesize contents with different depths and enable depth modification, we present a *depth-dependent weighted patch similarity metric*. The key idea is to separate the pixels in a patch into several regions according to their depths and pay more attention to color differences of the pixels with similar depth structures. Specifically, the pixels' color differences in two patches are weighted by the differences of their own relative depths to their center pixels. Therefore, our depth-dependent similarity metric consists of two terms. (1) The **depth structure resemblance term** measures the distance between their local depth structures (geometry), where two patches are similar if their local depth structures are similar. (2) The **depth-dependent color similarity term** measures the color distance between pixels but emphasizes more on the pixels with similar relative depths, where two patches are similar if the corresponding pixels with similar depth structures have similar colors. Fig. 3 illustrates the idea.

metric is defined as:

$$s(\mathbf{n}(p), \mathbf{n}(q)) = \sum_{k \in \mathbf{n}(p)} (\lambda |D_{\mathbf{n}(p)}(k) - \hat{D}_{\mathbf{n}(q)}(k)|^2 + w_{\mathbf{n}(p)\mathbf{n}(q)}(k) |I_{\mathbf{n}(p)}(k) - I_{\mathbf{n}(q)}(k)|^2), \quad (6)$$

where k is the relative position within a patch, and $I_{\mathbf{n}(\cdot)}(k)$ and $D_{\mathbf{n}(\cdot)}(k)$ denote the colors and disparity values at position k in the patch $\mathbf{n}(\cdot)$, respectively¹.

With the above notations, $D_{\mathbf{n}(p)}(0)$ denotes the disparity value of the central pixel p of the patch $\mathbf{n}(p)$. Thus, in the former term (the depth structure resemblance term) of Eq.(6), $\hat{D}_{\mathbf{n}(q)}(k) = D_{\mathbf{n}(q)}(k) + (D_{\mathbf{n}(p)}(0) - D_{\mathbf{n}(q)}(0))$ is the globally-shifted version of $D_{\mathbf{n}(q)}(k)$ such that the disparity values of the central pixels p and q in the two patches are the same (i.e., $D_{\mathbf{n}(p)}(0) = \hat{D}_{\mathbf{n}(q)}(0)$) (Fig. 3(b)). Fig. 3(c) demonstrates the distance between the two patches, $D_{\mathbf{n}(p)}$ and $\hat{D}_{\mathbf{n}(q)}$.

In the latter term (the depth-dependent color similarity term) of Eq.(6), $w_{\mathbf{n}(p)\mathbf{n}(q)}(k)$ is the depth-dependent weighting kernel for penalizing the color discrepancy more at the pixels with more similar depth structures (Fig. 3(d)). It is formally defined as:

$$w_{\mathbf{n}(p)\mathbf{n}(q)}(k) = \exp(-|D_{\mathbf{n}(p)}(k) - \hat{D}_{\mathbf{n}(q)}(k)|/\sigma^2). \quad (7)$$

With such a kernel, the pixels in a patch will contribute less if their relative depth structures are different from those of the other patch. Fig. 3(e) shows the comparison of color distances with and without the depth-dependent weighting kernel. In this example, we pay more attention to the pillar and less to the

Formally, the depth-dependent patch similarity

1. Note that, when defining patch similarity, the patches $\mathbf{n}(p)$ and $\mathbf{n}(q)$ can come from either the source or the target image pair. Thus, instead of S or T , we use $\mathbf{n}(p)$ and $\mathbf{n}(q)$ as the subscripts to indicate the disparity maps D associated with them. Similar for I .

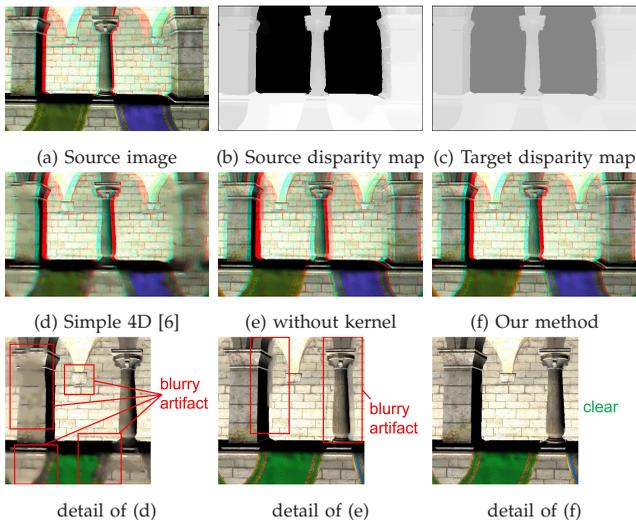


Fig. 4. Comparison of stereoscopic image synthesis using two simpler similarity metrics and the proposed depth-dependent similarity metric. (a) Given a stereoscopic image pair, assume that we want to narrow down its depth range. (b) The estimated source disparity map. (c) The remapped target disparity map. (d) The result using the simple 4D metric [6]. (e) The result of the proposed similarity metric without the depth-dependent weighting kernel. (f) The result using the proposed depth-dependent similarity metric, which does not suffer from blurry artifacts.

wall since the central pixel p is part of the pillar. The proposed kernel shares similarity with the popular bilateral kernel used for edge-preserving filtering [31]. The main difference is that our kernel weights intensity differences by the differences of relative depth structures while the bilateral kernel weights them by differences on locations and intensity values. Finally, the weight λ strikes a good balance between the color similarity and the depth structure resemblance ($\lambda = 1$ in our implementation).

For guaranteeing plausible depth interpretation, we need to synthesize both views jointly. In other words, when putting a patch from the left source image into the synthesized left view, we would like to place its corresponding patch in the right source image into a proper location (specified by the target disparity map) of the synthesized right view at the same time. Therefore, a patch-pair should be formed by associating two corresponding patches centered at the two corresponding pixels in the left and right views together. Assume that a pixel $p = (p_x, p_y) \in I^l$ is labeled as B in the visibility map V^l , so it has a corresponding pixel $p' = (p_x + D^l(p), p_y) \in I^r$. Thus, we can couple the patches $\mathbf{n}^l(p)$ and $\mathbf{n}^r(p')$ together and denote it as a patch-pair $\bar{\mathbf{n}}(p, p')$. Based on Eq.(6), the depth-dependent patch-pair similarity metric is

defined as:

$$\bar{s}(\bar{\mathbf{n}}(p, p'), \bar{\mathbf{n}}(q, q')) = s(\mathbf{n}^l(p), \mathbf{n}^l(q)) + s(\mathbf{n}^r(p'), \mathbf{n}^r(q')). \quad (8)$$

Fig. 4 shows a comparison of the proposed depth-dependent similarity metric with two simpler metrics: *simple 4D*, and *our metric without the depth-dependent weighting kernel*. Given a source stereoscopic image pair (Fig. 4(a)), we compressed its depth range to synthesize the results. Fig. 4(b) shows one of its estimated disparity maps. Then, we remap it using a global remapping function to obtain the target disparity map (Fig. 4(c)) with a narrower depth range. Fig. 4(d), (e), and (f) show the synthesized results using the simple 4D metric, our metric without the depth-dependent weighting kernel, and the proposed depth-dependent similarity metric, respectively. The detail views show that the two simpler metrics often produce blurry results. The simple 4D metric does not model the depth adjustment. Thus, two patches with similar color appearances and local depth structures yet at different depths are regarded as different patches. It would produce very blurry results as less similar patches could be used for synthesis (Fig. 4(d)). Without the depth-dependent weighting kernel, the pixels from different layers could be mixed up. As a result, it would produce blurry artifacts at the places with depth discontinuity (Fig. 4(e)). In contrast, with the proposed depth-dependent similarity metric, the foreground pillars and the background wall can be synthesized separately. Therefore, it generates the results without blurry artifacts (Fig. 4(f)).

3.4 Optimization for stereoscopic image synthesis

Our stereoscopic image synthesis process is achieved by optimizing Eq.(3). Similar to [4], we applied a multi-resolution iterative algorithm to optimize the objective function by refining the target images from lower to higher resolutions.

Our goal is to synthesize a target stereoscopic image pair \mathbf{I}_T^p that minimizes Eq.(3). The problem is complicated because these variables have dependencies which cannot be expressed as closed-form formulas: (1) D_T^r and (V_T^l, V_T^r) can be inferred from D_T^l as described in Section 3.1 and (2) D_T^l can be estimated from (I_T^l, I_T^r) . Furthermore, it is often preferable if users can have direct control to the target disparity map. Therefore, we opt to fix the target disparity maps after they are manually provided by the users or automatically generated by [25].

For each resolution, our iterative algorithm alternates between two steps: patch search and color refinement. In the search step, we fix the output pixels in (I_T^l, I_T^r) , and solve the nearest-neighbor search problem, i.e., for all overlapping patches in the source/target images, we retrieve the most similar

patches and patch-pairs in the target/source images. In the refinement step, we fix the set of matching patches and patch-pairs, and determine the pixel colors in the target images by weighted averaging the overlapping pixels. The two steps are iterated until convergence. The process is similar to previous work [4], [25], but requires several modifications to the patch search and color refinement steps.

Patch/patch-pair search. To find the nearest-neighbor for all the patches and patch-pairs to update the result while maintaining the depth interpretation, joint search of the nearest-neighbor patch-pair is critical for our synthesis process. We separate all the patches into two categories. For each patch in Ω_S^L and Ω_S^R , we use the distance metric described in Eq.(4) to search for the most similar patch in the I_T^L or I_T^R , respectively. For each patch-pair in $(\Omega_S^{B^l}, \Omega_S^{B^r})$, the most similar patch pair in $(\Omega_T^{B^l}, \Omega_T^{B^r})$ is found using the patch-pair similarity metric (Eq.(5)). Similarly, for each patch in the target image, we need to find its nearest-neighbor patch/patch-pair in the source image. To find the most similar patch-pair, we modified the randomized search technique [26] that is an efficient method for constructing nearest-neighbor fields for all patches, consisting of three steps: initialization, propagation, and random search. Every pair of corresponding patches in both views is assigned a pair of corresponding patches in each step. The details of the patch/patch-pair randomized search algorithm is described in the supplemental document for reproducibility.

Pixel color refinement. In this step, we minimize Eq.(3) by fixing the set of matching patches/patch-pairs and updating a new pair of target images. Let $I_T^l(p)$ denote the color of a pixel $p \in I_T^l$, the energy function is minimized by solving $\partial E(\mathbf{I}_S^p, \mathbf{I}_T^p)/\partial I_T^l(p) = 0$. To derive it, we first isolate the contribution of $I_T^l(p)$ to the image distance $E(\mathbf{I}_S^p, \mathbf{I}_T^p)$ of Eq.(3).

We take the pixels in the left target image as an example. In the coherence term, let $\{\mathbf{n}_i\}_{i=1\dots m} \in I_T^l$ denote the m patches that contain the pixel p , $\{\mathcal{N}(\mathbf{n}_i)\}_{i=1\dots m} \in \Omega_S^l$ denote their nearest-neighbor patches obtained from the search step, and the pixels in $\{\mathcal{N}(\mathbf{n}_i)\}_{i=1\dots m}$ corresponding to the position of p are denoted as $\{q_i\}_{i=1\dots m}$. We assume that the first k patches are located in Ω_T^L , while the remaining patches are located in $\Omega_T^{B^l}$. Then the contribution of $I_T^l(p)$ to the coherence term is

$$E_{coh}(I_T^l(p)) = \frac{1}{|\Omega_T^L|} \sum_{i=1}^k w_{\mathbf{n}_i \mathcal{N}(\mathbf{n}_i)}(p) (I_S^l(q_i) - I_T^l(p))^2 + \frac{1}{|\Omega_T^{B^l}|} \sum_{i=k+1}^m w_{\mathbf{n}_i \mathcal{N}(\mathbf{n}_i)}(p) (I_S^l(q_i) - I_T^l(p))^2, \quad (9)$$

where $w_{\mathbf{n}_i \mathcal{N}(\mathbf{n}_i)}(p)$ is the depth-dependent weighting kernel introduced in Sec. 3.3, and $I_S^l(q_i)$ is the color of pixel $q_i \in \Omega_S^l$.

In the completeness term, assume there are \hat{m}

patches $\{\mathcal{N}(\hat{\mathbf{n}}_j)\}_{j=1\dots \hat{m}} \in I_T^l$ that contain the pixel p ; and they could be matched by the patches $\{\hat{\mathbf{n}}_j\}_{j=1\dots \hat{m}} \in \Omega_S^l$; and $\hat{q}_j \in \hat{\mathbf{n}}_j$ is the pixel at the same relative position as $p \in \mathcal{N}(\hat{\mathbf{n}}_j)$. We also assume that the first k patches are located in Ω_T^L , while the remaining patches are located in $\Omega_T^{B^l}$. The contribution of $I_T^l(p)$ to the completeness term is

$$E_{com}(I_T^l(p)) = \frac{1}{|\Omega_S^l|} \sum_{j=1}^{\hat{k}} w_{\hat{\mathbf{n}}_j \mathcal{N}(\hat{\mathbf{n}}_j)}(p) (I_S^l(\hat{q}_j) - I_T^l(p))^2 + \frac{1}{|\Omega_S^{B^l}|} \sum_{j=k+1}^{\hat{m}} w_{\hat{\mathbf{n}}_j \mathcal{N}(\hat{\mathbf{n}}_j)}(p) (I_S^l(\hat{q}_j) - I_T^l(p))^2. \quad (10)$$

Therefore, the contribution of the color of pixel $I_T^l(p)$ to the image similarity of Eq.(3) is

$$E(I_T^l(p)) = E_{coh}(I_T^l(p)) + E_{com}(I_T^l(p)). \quad (11)$$

To refine the color $I_T^l(p)$ for minimizing the image distance, we solve $\partial E(I_T^l(p))/\partial I_T^l(p) = 0$, and the optimal solution for $I_T^l(p)$ is obtained via the following formula:

$$I_T^l(p) = \left(\frac{\sum_{j=1}^k w_{\hat{\mathbf{n}}_j \mathcal{N}(\hat{\mathbf{n}}_j)}(p) I_S^l(\hat{q}_j)}{|\Omega_S^l|} + \frac{\sum_{j=k+1}^{\hat{m}} w_{\hat{\mathbf{n}}_j \mathcal{N}(\hat{\mathbf{n}}_j)}(p) I_S^l(\hat{q}_j)}{|\Omega_S^{B^l}|} \right) / \left(\frac{\sum_{i=1}^k w_{\mathbf{n}_i \mathcal{N}(\mathbf{n}_i)}(p) I_S^l(q_i)}{|\Omega_T^L|} + \frac{\sum_{i=k+1}^m w_{\mathbf{n}_i \mathcal{N}(\mathbf{n}_i)}(p) I_S^l(q_i)}{|\Omega_T^{B^l}|} \right) + \left(\frac{\sum_{j=1}^k w_{\hat{\mathbf{n}}_j \mathcal{N}(\hat{\mathbf{n}}_j)}(p)}{|\Omega_S^l|} + \frac{\sum_{j=k+1}^{\hat{m}} w_{\hat{\mathbf{n}}_j \mathcal{N}(\hat{\mathbf{n}}_j)}(p)}{|\Omega_S^{B^l}|} \right) / \left(\frac{\sum_{i=1}^k w_{\mathbf{n}_i \mathcal{N}(\mathbf{n}_i)}(p)}{|\Omega_T^L|} + \frac{\sum_{i=k+1}^m w_{\mathbf{n}_i \mathcal{N}(\mathbf{n}_i)}(p)}{|\Omega_T^{B^l}|} \right). \quad (12)$$

The pixel color $I_T^r(p)$ can be refined in the same way. The details and the derivation of the update rules of $I_T^l(p)$ and $I_T^r(p)$ are described in the supplemental document.

4 RESULTS AND DISCUSSION

We applied our method to a wide variety of stereoscopic image editing applications. All the results were produced with the patch size 7×7 . Please notice that the results are presented as red(left)-cyan(right)  anaglyph images. The original left and right images are included in the supplemental materials. We encourage readers to watch them on stereoscopic displays for better visual quality.

4.1 Results

Depth-guided texture synthesis. Fig. 5 shows that our method can successfully synthesize the stereoscopic textures according to user-provided depths. Given an input texture sample and a target disparity map, our method synthesizes a larger stereoscopic texture pair whose depth interpretation is conformed to the given disparity map. To achieve the synthesis, the source texture sample is taken as both the left and right source images of our algorithm. As for disparity

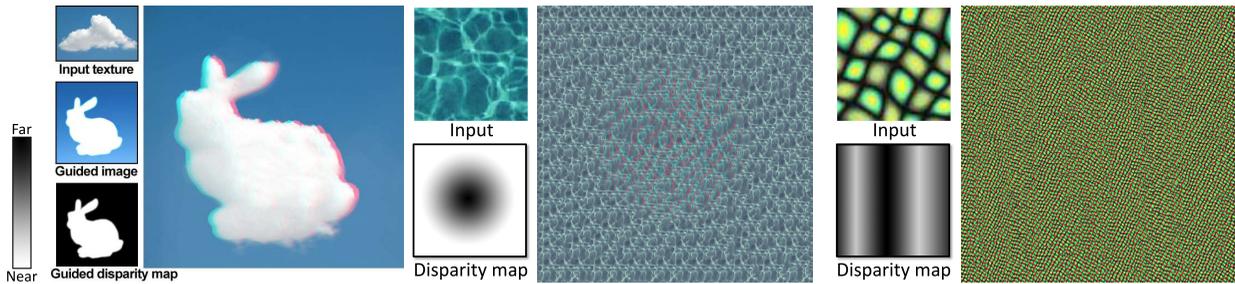


Fig. 5. Examples of stereoscopic texture synthesis for given disparity maps. The intensity levels in the disparity maps are proportional to the disparity values. (The image resolution of both middle and right images are 1200×1200 .)

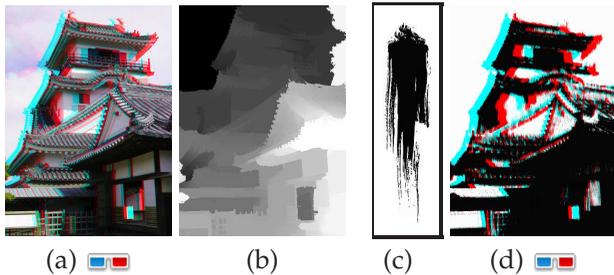


Fig. 6. An example for stereoscopic NPR. (a) The source stereoscopic image. (b) The estimated disparity map. (c) The reference ink brush texture image. (d) The result.

maps, we treat the source disparity map as a flat map with zero disparity values everywhere while the target disparity map is specified by the user. When there are multiple textures in the input texture sample, a guided image can be optionally taken as the initial guess to explicitly guide the synthesis. Taking Fig. 5 Left as an example, there are two types of textures, sky and cloud, in the input. The guided image has two regions: white (a rabbit) and blue (the sky). Guided by it, the proposed method synthesized cloud texture into the white region and sky texture into the blue region.

Stereoscopic NPR. Fig. 6 gives an example of NPR-stylized stereoscopic images. Given a source stereoscopic image, we first estimated its disparity map, and then transferred the reference ink brush texture to the source image to generate a non-photorealistic stereoscopic image. In this application, the reference ink brush texture is taken as both the left and right source images, and the source disparity map is again flat with zeros. The input stereoscopic image and its estimated disparity map are used to initialize the target images and disparity map and guide the synthesis.

Paint by depth. Fig. 7 shows that our method is able to synthesize a plausible stereoscopic image according to users' editing on the disparity map. Given a source stereoscopic image and its disparity map, a user provides the target disparity map by manually painting the desired shape of the tree on the disparity map. Our

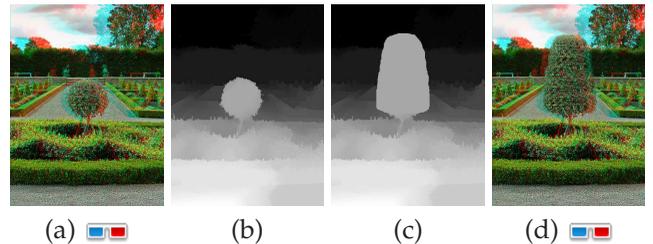


Fig. 7. Disparity map editing. (a) The source stereoscopic image. (b) disparity map. (c) The edited disparity map. (d) The result.

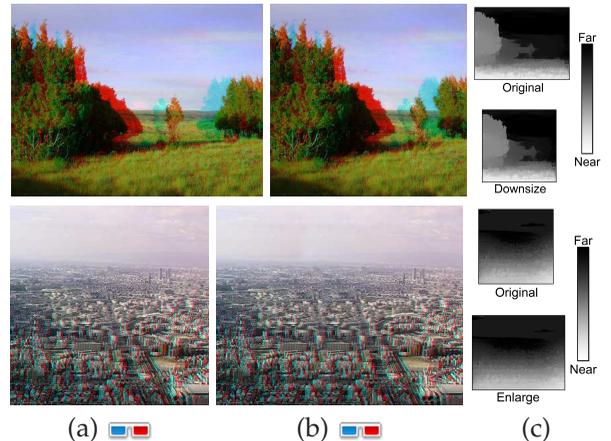


Fig. 8. Stereoscopic image retargeting results: “Tree” (upper) and “City” (lower). (a) The source stereoscopic images. (b) The results of changing the image width. (c) The original and retargeted disparity maps.

method then synthesized the tree with the painted shape according to the modified disparity map.

Stereoscopic content adaptation. Our technique can also be applied to adapt the content of stereoscopic images. Fig. 8 shows that our technique can be used for stereoscopic image retargeting, where the widths were reduced by 22% and increased by 24% in the two examples, respectively. To resize a stereoscopic image, the target disparity map is first obtained by resizing the source disparity map using a 2D synthesis algorithm [25]. Although there is no guarantee that the synthesized target disparity map is a feasible

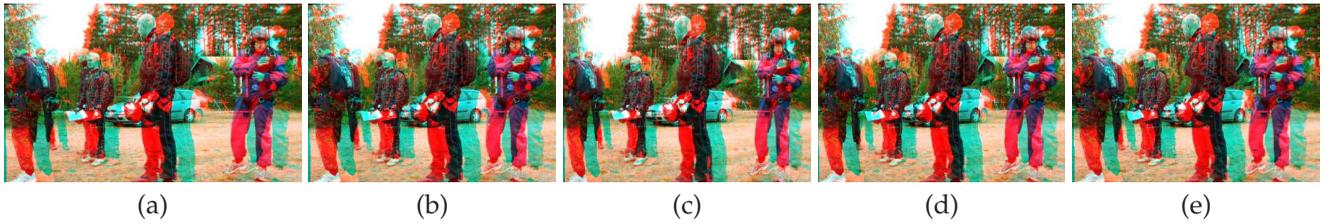


Fig. 9. A comparison of stereoscopic image retargeting using our approach with a warping-based method [2], scene warping [13], and stereo seam carving [12]. The source stereoscopic image (482×286) (a) is resized to 400×286 using our method (b), Chang et al. [2] (c), Lee et al. [13] (d), and Basha et al. [12] (e), respectively.

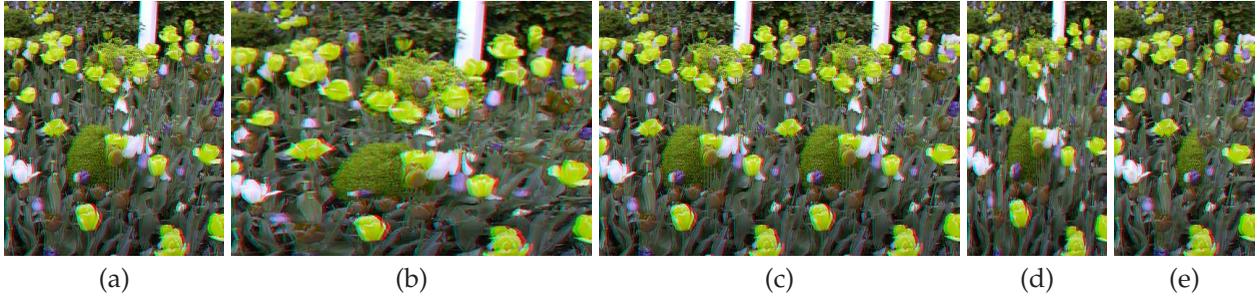


Fig. 10. A comparison of stereoscopic image retargeting using our approach and a warping-based method [2]. (a) The source stereoscopic image (304×351). (b)&(c) The enlarged results (500×351) of Chang et al. [2] and ours, respectively. (d)&(e) The reduced results (194×351) of Chang et al. [2] and ours, respectively. Our method synthesizes the flowers according to the image resolutions, and thus better preserves the shapes and details. In contrast, the warping-based method stretches the images to satisfy the image dimension, thus producing visible shape distortion especially in the cases where the images have rich textures.

one, we found it works well in many cases. Fig. 9 shows a comparison of image resizing using our method (a), Chang et al. [2] (c), Lee et al. [13] (d), and Basha et al. [12] (e). Compared to the algorithms specifically designed for this application, our method can generate equally good results on this example.

Fig. 10 shows a comparison of our approach to a representative warping-based method [2] for more dramatic resizing. The warping-based approach stretched the flowers and leaves and unavoidably produced shape distortion while our method better retained the shapes and sharp details by synthesizing new contents. One thing to note is that our method is usually less efficient compared to those warping-based methods [2], [13]. However, our method offers several advantages. First, it is more versatile and can be used for a set of editing tasks while those methods are specifically designed for image resizing and depth remapping. Second, for image resizing, our results have very different characteristics compared to the results of warping-based methods as shown in Fig. 10. Thus, our method and those methods can be suitable for different situations.

Fig. 11 shows that our patch-based approach can also be used for remapping the disparity range, which is important for stereoscopic content production and display. To achieve this, users can specify the target disparity map via a disparity mapping operator $d_t = f(d_s)$ to guide the synthesis process, where f is a map-

ping function maps the input disparity value d_s to the output disparity value d_t . Fig. 12 shows a comparison of our method to a warping-based method [1]. The warping-based approach leads to visible distortion if there are significant disparity variations within quads. It is because the pixels with different disparity values in the same quad usually need to undergo different transforms, which is impossible for warping-based methods. In contrast, our method allows each patch to contribute only to the regions with similar local depth structures. As a result, our method better preserves the straight lines as shown in Fig. 12(d).

Stereoscopic image inpainting and reshuffling. By specifying the search constraints and/or match constraints for some regions on the source stereoscopic image, our system can also be used for stereoscopic image inpainting and content reshuffling. In this application, the users paint the region(s) to be erased or make layout rearrangements on the source stereoscopic image. They only need to paint on one view of the stereoscopic image. The painted pixels are mapped to the other view automatically based on the disparity map. Note that the completeness terms are removed when performing stereoscopic image inpainting. Fig. 13 shows an example of the constrained synthesis. Given a source stereoscopic image (Fig. 13(a)), and the user-specified region(s) to erase (Fig. 13(b)) and/or layout rearrangement(s) (Fig. 13(d)), our system synthesizes the results with

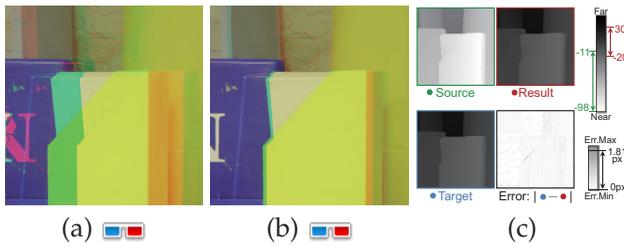


Fig. 11. Disparity remapping results. (a) The source stereoscopic image whose disparity range is too large ($-11 \sim -98$) for human to fuse. (b) The result stereoscopic image after adapting the disparity range to the comfortable zone. (c) The source disparity map (green), the target disparity map (blue), the final voted disparity map (red), and the error map showing the differences between the final voted and target disparity maps.

the search and/or match constraint(s). The results are shown in Fig. 13(c) for inpainting and (e) for reshuffling.

2D to 3D conversion. Given a 2D image and a user-provided target disparity map (which can be obtained by drawing a dense depth map or using a sparse-scribble propagation technique, e.g., [32]), our method can synthesize the stereoscopic 3D images by regarding the target disparity map as D_T^l and duplicating the input image as both I_S^l and I_S^r . Fig. 14 shows three examples with our method. The disparity maps used in the “moon” and “building” examples were manually drawn by users, and the disparity map used in the “cave” example was generated using a propagation technique [32].

4.2 Performance

We implemented our method in C++ and executed it on a desktop PC equipped with an Intel i7 3.5GHz CPU, 16GB RAM, and an Nvidia GeForce GTX 580 GPU. The patch/patch-pair search needs to handle both the left and right views, and is the most time-consuming step. We used OpenMP and GPU to utilize the parallelism for accelerating this step. Overall, for a pair of 0.4-megapixel images (the total pixel number is 0.4×2 megapixels), it took 435.6 seconds in average with our CPU implementation to synthesize images. With the GPU version, the time was reduced to 2 minutes.

Our current implementation aims for pursuing the output quality and thus the computation cost is high. There are some possible ways toward more efficient computation. Firstly, currently we sample patches for all pixels. The computation time could be reduced if we sample patches every k pixels along x and y directions, where k is the sample step. Secondly, the patch/patch-pair search is the bottleneck of the computation time. A multiscale strategy can be adopted to accelerate the step. Specifically, the stereoscopic

image is firstly downscaled to a coarse level, and the patch/patch-pair search is performed on that level. Then the matching result is propagated back to the finer level and is further refined within a local region.

4.3 Evaluation

To validate the stereoscopic quality and the effectiveness of disparity remapping of our synthesized stereoscopic images, we conducted two experiments using a 22-inch 120Hz 3D LCD monitor with active shutter glasses. We recruited 30 participants, and 4 of them had no experience on watching stereoscopic contents before. Participants passing the depth perception ability test were asked to perform the following tasks.

Stereoscopic quality evaluation. In this task, we randomly selected 10 natural stereoscopic images and adjusted their contents with one of the above-mentioned applications². These 20 images (both source and synthesized images) were displayed to the participants in a random order. The participants were asked to watch each image and rate it in 30 seconds on (1) how convincing its stereoscopic effect is, and (2) how natural it is from 1(Bad) to 5(Good). In total, we received 300 pairs of ratings for each question. Fig. 15 shows the average scores for the two questions on all ratings and the ratings for each editing application. For stereoscopic 3D quality, there are 74% ratings showing that the synthesized results have equal or higher score than the source images; and for naturalness, there are 83% ratings showing the same result.

Disparity remapping. In the second task, we evaluated whether the disparity remapping is effective. We randomly selected 4 natural stereoscopic images and enhanced their depth ranges by a factor of 1.5 using the proposed method. The remapped results and the original images were placed side-by-side with randomized orders. Participants were asked to perform pairwise comparisons. For each comparison, the participants had to answer the following question: “Which image has a larger depth range?”. We totally received 120 valid votes on this task. Overall, 87.3% correctly recognized the images with larger disparity ranges. Kendall’s coefficient of agreement was adopted to measure the interobserver variability for the pairwise comparison tests and the p -value < 0.01 . Therefore, the disparity remapping results with our method are perceptually effective.

4.4 Discussion

Our method requires disparity maps in order to separately synthesize contents with different depths. The target disparity maps can be either synthesized using a 2D patch-based approach (e.g.[25]) or specified

2. Some of the images are presented in the paper, including Fig. 8 Upper, Fig. 8 Lower, Fig. 10(a)(c), Fig. 11, and Fig. 13(a)(e). For all left and right images, please refer to supplemental materials.

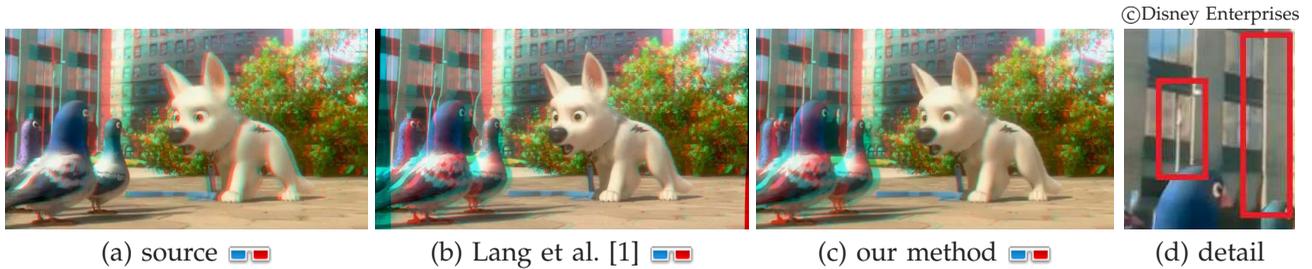


Fig. 12. A comparison of disparity remapping using our approach and a warping-based method [1]. (a) The source stereoscopic image. (b)&(c) The disparity remapping results using Lang et al. [1] and ours, respectively. (d) A detail view of our result shows that our technique can preserve the straight lines in the regions with significant disparity variations. The warping-based method bends the straight lines because the disparity values of the pigeon and the building are very different. It thus produces visible distortions.

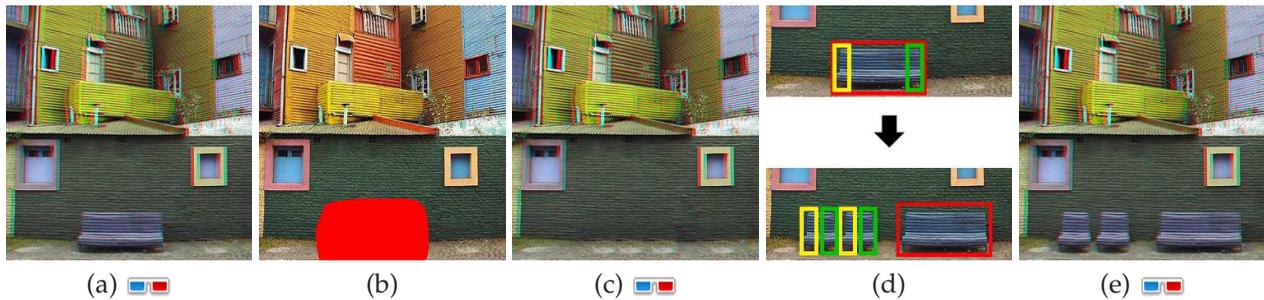


Fig. 13. Results of stereoscopic image inpainting and reshuffling. (a) The source stereoscopic image. (b) The region to be erased is painted by red color. (c) The stereoscopic image inpainting result. (d) The layout rearrangements specified as red, green, and yellow rectangles. (e) The stereoscopic image reshuffling result.

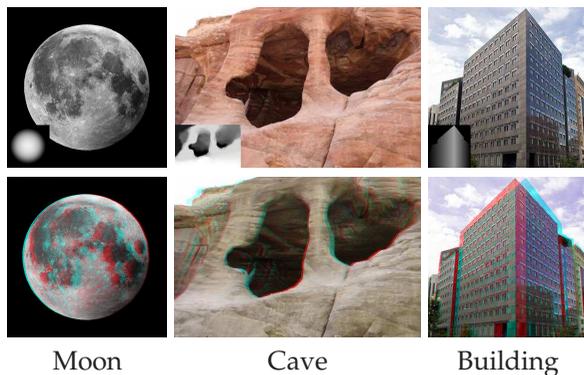


Fig. 14. Examples of 2D to 3D conversion. Upper: The source images and given disparity maps (inset). Lower: The 2D to 3D results.

by users. Synthesized disparity maps often lead to reasonable results because they are guaranteed to be similar to the source disparity maps. However, for user-specified disparity maps, if they are too different from the source, the synthesized results could be less meaningful.

We have also evaluated how well the depth interpretations of the synthesized stereoscopic images are conformed to the target disparity maps. Although stereo matching algorithms can be applied to estimate the result disparity map by analyzing the synthesized images, existing stereo matching algorithms are still



Fig. 15. The average ratings of the synthetic images and the original images. (The error bars indicate the standard deviations.)

far from perfect and could contain errors. Therefore, instead of applying stereo matching algorithms, we adopt the refinement rule described in Eq.(12) to vote the result disparity map by substituting the color information with depth information. The disparity maps obtained in this way are closer to what we perceive from the synthesized images. As shown in Fig. 11(c), the voted disparity maps (red outlined) are

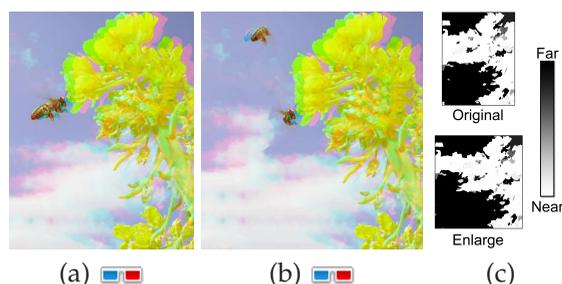


Fig. 16. A failure case due to the inaccurate disparity map. (a) The source stereoscopic image. (b) The enlarged result. (c) The disparity maps. Note that the disparity values of the sky at the up-left corner are not accurate. They should be further back but misestimated as the same as the tree.

consistent to the target disparity maps (blue outlined). The errors between the voted disparity maps and the target disparity maps are shown in the lower right corner of Fig. 11(c), in which the maximal error is less than 2 pixels. It shows that the proposed framework can synthesize stereoscopic images whose depth interpretations are very close to the target disparity maps.

Limitation. Although the proposed method can be applied to a wide variety of applications, it suffers from a few limitations. Firstly, the current computation cost is still very high. The method certainly benefits from faster nearest neighbor search algorithms. Additionally, the method can also be accelerated by more aggressive sampling of patches as discussed in Sec. 4.2.

Secondly, the patch-based approaches do not guarantee to preserve global structures. As presented in Fig. 1 (c), the lower part of the new tree is not synthesized, and thus the tree looks floating. This is inherited from the patch-based synthesis approaches that lack the knowledge of global structures, and could be improved by incorporating semantic object detection algorithms.

Finally, although with some degree of tolerance, the proposed method suffers from bad quality of disparity maps. Fortunately, existing stereo methods can produce good disparity maps for plausible results in most cases. Fig. 16 shows a failure case, in which the bee is separated into two parts because most of the disparity values are inaccurate. In addition, the proposed method synthesizes the results by utilizing only existing patches in the source images. When there are significant view changes in the target, the existing patches in the source need to be re-projected and warped according to the depth structure changes between the original view and the novel view. The current method cannot handle such cases.

5 CONCLUSION AND FUTURE WORK

We have presented a stereoscopic patch-based synthesis framework that handles the corresponding in-

formation in two views and separately synthesizes contents with different depth structures. The combination of depth-dependent patch/patch-pair similarity metric and joint nearest-neighbor search contributes to the realism of the synthesized stereoscopic images and plausibility of their depth interpretations. The method has potential to be useful for many stereoscopic image processing applications as demonstrated in the experiments. A few interesting research directions are worth of exploration. First, the current method only compensates the local depth structures of patches by global depth shifts. To accommodate for large perspective changes, the depth structures of patches should be compensated by proper rigid transforms before evaluating patch similarity. Second, to synthesize results with large view changes, view interpolation or depth-image-based rendering could be combined with our approach.

REFERENCES

- [1] M. Lang, A. Hornung, O. Wang, S. Poulakos, A. Smolic, and M. Gross, "Nonlinear disparity mapping for stereoscopic 3D," *ACM TOG*, vol. 29, no. 4, pp. 75:1–75:10, 2010.
- [2] C.-H. Chang, C.-K. Liang, and Y.-Y. Chuang, "Content-aware display adaptation and interactive editing for stereoscopic images," *IEEE TMM*, vol. 13, no. 4, pp. 589–601, 2011.
- [3] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, "Texture optimization for example-based synthesis," *ACM TOG*, vol. 24, no. 3, pp. 795–802, 2005.
- [4] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE TPAMI*, vol. 29, no. 3, pp. 463–476, 2007.
- [5] L. Wang, H. Jin, R. Yang, and M. Gong, "Stereoscopic inpainting: Joint color and depth completion from stereo images," in *Proc. IEEE CVPR '08*, 2008.
- [6] B. Morse, J. Howard, S. Cohen, and B. Price, "Patchmatch-based content completion of stereo image pairs," in *Proc. 3DIMPVT '12*, 2012, pp. 555–562.
- [7] S. Koppal, C. L. Zitnick, M. Cohen, S. B. Kang, B. Ressler, and A. Colburn, "A viewer-centric editor for 3D movies," *IEEE CG&A*, vol. 31, no. 1, pp. 20–35, 2011.
- [8] W.-Y. Lo, J. van Baar, C. Knaus, M. Zwicker, and M. Gross, "Stereoscopic 3D copy & paste," *ACM TOG*, vol. 29, no. 6, pp. 147:1–147:10, 2010.
- [9] S.-J. Luo, I.-C. Shen, B.-Y. Chen, W.-H. Cheng, and Y.-Y. Chuang, "Perspective-aware warping for seamless stereoscopic image cloning," *ACM TOG*, vol. 31, no. 6, pp. 182:1–182:8, 2012.
- [10] C. Wang and A. A. Sawchuk, "Disparity manipulation for stereo images and video," in *Proc. SPIE*, vol. 6803, 2008, p. 68031E.
- [11] T. Basha, Y. Moses, and S. Avidan, "Geometrically consistent stereo seam carving," in *Proc. IEEE ICCV '11*, 2011, pp. 1816–1823.
- [12] T. Dekel Basha, Y. Moses, and S. Avidan, "Stereo seam carving a geometrically consistent approach," *IEEE TPAMI*, vol. 35, no. 10, pp. 2513–2525, 2013.
- [13] K.-Y. Lee, C.-D. Chung, and Y.-Y. Chuang, "Scene warping: Layer-based stereoscopic image resizing," in *Proc. IEEE CVPR '12*, 2012, pp. 49–56.
- [14] A. Mansfield, P. Gehler, L. Van Gool, and C. Rother, "Scene carving: scene consistent image retargeting," in *Proc. ECCV '10: Part I*, 2010, pp. 143–156.
- [15] Y. Niu, W.-C. Feng, and F. Liu, "Enabling warping on stereoscopic images," *ACM TOG*, vol. 31, no. 6, pp. 183:1–183:7, 2012.
- [16] L. Northam, P. Asente, and C. S. Kaplan, "Consistent stylization and painterly rendering of stereoscopic 3D images," in *Proc. NPAR '12*, 2012, pp. 47–56.
- [17] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *Proc. IEEE ICCV '99*, vol. 2, 1999, pp. 1033–1038.

- [18] L.-Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," in *Proc. ACM SIGGRAPH '00*, 2000, pp. 479–488.
- [19] M. Ashikhmin, "Synthesizing natural textures," in *Proc. I3D '01*, 2001, pp. 217–226.
- [20] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum, "Real-time texture synthesis by patch-based sampling," *ACM TOG*, vol. 20, no. 3, pp. 127–150, 2001.
- [21] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proc. ACM SIGGRAPH '01*, 2001, pp. 341–346.
- [22] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," in *Proc. IEEE CVPR '03*, vol. 2, 2003, pp. 721–728.
- [23] I. Drori, D. Cohen-Or, and H. Yeshurun, "Fragment-based image completion," *ACM TOG*, vol. 22, no. 3, pp. 303–312, 2003.
- [24] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum, "Image completion with structure propagation," *ACM TOG*, vol. 24, no. 3, pp. 861–868, 2005.
- [25] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, "Summarizing visual data using bidirectional similarity," in *Proc. IEEE CVPR '08*, 2008.
- [26] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: a randomized correspondence algorithm for structural image editing," *ACM TOG*, vol. 28, no. 3, pp. 24:1–24:11, 2009.
- [27] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, and P. Sen, "Image melding: combining inconsistent images using patch-based synthesis," *ACM TOG*, vol. 31, no. 4, pp. 82:1–82:10, 2012.
- [28] B. M. Smith, L. Zhang, and H. Jin, "Stereo matching with nonparametric smoothness priors in feature space," in *Proc. IEEE CVPR '09*, 2009, pp. 485–492.
- [29] A. Humayun, O. M. Aodha, and G. J. Brostow, "Learning to find occlusion regions," in *Proc. IEEE CVPR '11*, 2011, pp. 2161–2168.
- [30] L.-Y. Wei, S. Lefebvre, V. Kwatra, and G. Turk, "State of the art in example-based texture synthesis," in *EG '09 STARS*, 2009, pp. 93–117.
- [31] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. IEEE ICCV '98*, 1998, pp. 839–846.
- [32] O. Wang, M. Lang, M. Frei, A. Hornung, A. Smolic, and M. Gross, "StereoBrush: interactive 2D to 3D conversion using discontinuous warps," in *Proc. SBIM '11*, 2011, pp. 47–54.



I-Chao Shen received B.B.A and M.B.A degrees in information management from National Taiwan University, in 2009 and 2011, respectively. He is currently a research assistant with the Research Center for Information Technology Innovation (CITI), Academia Sinica, Taipei, Taiwan. His research interests include image processing, computational photography, and geometry processing.



Bing-Yu Chen received the B.S. and M.S. degrees in computer science and information engineering from National Taiwan University, in 1995 and 1997, respectively, and the Ph.D. degree in information science from The University of Tokyo, Japan, in 2003. He is currently a professor with National Taiwan University. His current research interests include computer graphics, image and video processing, and human-computer interaction. He is a senior member of ACM and

a member of Eurographics.



Yung-Yu Chuang received the B.S. and M.S. degrees from National Taiwan University in 1993 and 1995 respectively, and the Ph.D. degree from the University of Washington at Seattle in 2004, all in Computer Science. He is currently a professor with the Department of Computer Science and Information Engineering, National Taiwan University. His research interests include computational photography, rendering and computer vision. He is a member of the IEEE and the IEEE

Computer Society, and a member of ACM.



Sheng-Jie Luo received the B.B.A. and M.B.A degrees in information management from National Taiwan University, in 2007 and 2009, respectively, and the Ph.D. degree in computer science from National Taiwan University, in 2013. He is currently a senior engineer with HTC Corporation, Taipei, Taiwan. His research interests include computer graphics, computational photography, and information visualization. He is a member of ACM.



Ying-Tse Sun received the B.S. degree in computer science from National Chengchi University (Taipei, Taiwan) in 2010, and the M.B.A. degree in information management from National Taiwan University in 2012. She is currently an engineer with Vivotek Inc. Her research interests include computer graphics, computational photography and image editing.