Stereoscopic 3D Copy & Paste

Wan-Yen Lo[1,2] Jeroen van Baar[3] Claude Knaus[2] Matthias Zwicker[1,2] Markus Gross[3,4]
[1]UC San Diego [2]Universität Bern [3]Disney Research Zürich [4]ETH Zürich

Figure 1: *We present an end-to-end system for stereoscopic 3D copy & paste. Left column: stereo input images (only left eye shown). Middle and right column: two results of stereo copy & paste. All anaglyph images in this paper are best viewed with Red/Cyan glasses, with Red = Left eye. For optimal viewing, we encourage the reader to look at the results in the electronic version of our paper.*

Abstract

With the increase in popularity of stereoscopic 3D imagery for film, TV, and interactive entertainment, an urgent need for editing tools to support stereo content creation has become apparent. In this paper we present an end-to-end system for object copy & paste in a stereoscopic setting to address this need. There is no straightforward extension of 2D copy & paste to support the addition of the third dimension as we show in this paper. For stereoscopic copy & paste we need to handle depth, and our core objective is to obtain a *convincing 3D viewing experience*. As one of the main contributions of our system, we introduce a stereo billboard method for stereoscopic rendering of the copied selection. Our approach preserves the stereo volume and is robust to the inevitable inaccuracies of the depth maps computed from a stereo pair of images. Our system also includes an interactive stereoscopic segmentation tool to achieve high quality object selection. Hence, we focus on intuitive and minimal user interaction, and our editing operations perform within interactive rates to provide immediate feedback.

CR Categories: I.3.3 [Computing Methodologies]: Computer Graphics—Picture/Image Generation;

Keywords: Multi-View & 3D, Computational Photography

1 Introduction

Recently stereoscopic 3D has gained in popularity—again—which has resulted in stereoscopic 3D efforts in everything from 3D cinema to 3D TV at home. With this arises the need for 3D content creation and editing tools. The goal of this paper is to address this need. It is not straightforward to extend existing 2D tools to 3D, since the extension to 3D introduces challenges related to recovering the depth, and maintaining comfortable stereo perception, including ensuring the correct handling of occlusions.

In this paper we focus on a specific editing application: copy & paste for stereoscopic 3D images. Our motivation for choosing this application is three-fold: 1) if "3D-at-home" will be successful, many people will eventually own a 3D TV which will be capable of displaying 3D photographs, 2) recently 3D digital cameras have been introduced [Fuji 2009] which make 3D photography easily realizable for many, and 3) photo editing tools for 2D images have proven popular among professionals as well as casual users. We will present a complete system for 3D copy & paste consisting of the following components: depth reconstruction, selection, and composition. We will also describe our contributions for each component.

Copy & paste for 2D images has received a lot of attention in recent years [Pérez et al. 2003; Georgiev 2006; Farbman et al. 2009]. The users' task for a plausible selection is to find objects which match in scale and orientation with that of the target. Objects can then be selected with a "rough" selection. No accurate segmentation of the object is required, provided that backgrounds are either uniformly colored or have similar texture. Simply applying these 2D methods in the source and target to the left and right eye images is not sufficient, since 3D copy & paste has to take stereopsis into account and avoid *stereopsis rivalry*: conflicting cues to the human visual system in the left and right eye images which could severely strain the visual system, or even destroy the 3D *illusion* altogether [Howard and Rogers 2002; Patterson 2007; Lambooij et al. 2009]. More specifically, important aspects are:

- Occlusion, being an important depth due, has to be handled correctly.
- Maintain the copied objects' *stereo volume*, i.e., the anisotropic parallax between pixels that belong to the object and provide the cues for its 3D shape. Loss of this information leads to the so-called "cardboarding" effect, where objects appear as flat planes in depth.
- The composition result should be consistent for both left and right eye images. The pasted object should assume the correct orientation depending on the surface orientation in the target, which varies with the desired location for pasting.

- The copied object disparities in the target should be such that the depth composition is correct with respect to the depth in the target.

To take these aspects into account for 3D copy & paste introduces the problem of recovering the depth information. Many existing methods for two-view stereo have been presented to compute per-pixel disparities [Scharstein and Szeliski 2010]. However, for input images of arbitrary scenes the computed disparities are often inaccurate.

Furthermore, another challenge is to seamlessly composite the copied selection into the target. The aforementioned 2D copy & paste methods may result in smearing artifacts in the case where the backgrounds are dissimilar in texture. Only composition using alpha mattes can seamlessly blend objects with dissimilar backgrounds [Wang and Cohen 2008]. High quality alpha mattes will require accurate segmentation of the object to be copied and pasted.

Finally, direct rendering methods, e.g., forward mapping or geometry mesh approximation, may result in artifacts in the case of inaccurate depth maps.

Contributions To address these challenges, we have developed an end-to-end system for 3D copy & paste with the following additional contributions:

- Automatic propagation of the segmentations from left eye to right eye image (Section 3.2).
- Registration with respect to the local underlying support surface in the target (Section 3.3.1).
- Rendering using stereo billboards, which avoids the so-called "cardboarding" effect and preserves the original stereo volume of the source selection (Section 3.3.3).
- Generation of contact shadows by transferring the disparity map to the target and using an image space ambient occlusion approach (Section 3.3.5).

Paper Organization The remainder of this paper is organized as follows: we discuss related work in Section 2, and a detailed description of our system is given in Section 3, results obtained with our system are presented in Section 4, a discussion of the presented system and outlook on future work is given in Section 5, and finally Section 6 provides concluding remarks.

2 Related Work

We summarize prior work that is relevant to the challenges described in the previous section and to the components of our system, which we describe in the following. We furthermore describe how we address some of the presented problems.

Disparity Maps An important requirement of our system is computing disparity maps. Many two-view stereo disparity map methods, classified according to Scharstein and Szeliski [2002], have been reported in the literature and their relative scores are listed [Scharstein and Szeliski 2010]. However, to date no method can produce accurate disparity maps for arbitrary input images such as those typically found in people's digital photo collections. Our system therefore aims to be robust with respect to depth map inaccuracies.

Segmentation Accurate segmentation is inherently user assisted and iterative. Rother et al. [2004] describe a method that iteratively applies graph cuts optimization. Users may provide additional hints to refine the segmentation. Liu et al. [2009b] let the user paint strokes to denote foreground object and an incremental graph cuts scheme updates the segmentation in real-time. Multi-object segmentation for both methods would require a significant amount of user interaction. Lu et al. [2007] describe a multi-class segmentation method, but this can handle only a small number of distinct classes and is computationally expensive. To allow for easy multiple object segmentation we combine the fast cluster-merging method by Ning et al. [2010], and mean-shift clustering [Comaniciu and Meer 2002].

Pop-up light field [Shum et al. 2004] is an image-based rendering system that models a sparse light field using a layered representation. In this system, the user interactively segments layers for *pop-up* until some desired quality is met. Our system shares some similarities with pop-up light field, but we work with stereoscopic input instead of sparse light fields. In addition, our layers are not flat, but we preserve stereo volume using our stereo billboards. Finally, we focus on editing using copy & paste, while pop-up light field is mainly concerned with high quality rendering.

Cosegmentation Propagating the segmentations from one eye image to the other eye image is related to cosegmentation. Cosegmentation aims at segmenting the common parts between a pair or a sequence of images. Rother et al. [2006] exploit histograms for consistency between foreground objects in images. Cheng et al. [2007] encode the consistency between objects in frames within a prior and solve a mixture model. Zitnick et al. [2005] aim for consistent segmentation and motion simultaneously, using segment shape and optical flow between images as constraints and finally solving an energy minimization problem. Motion, optical flow, and tracking have also been proposed in segmentation propagation for video sequences [Chuang et al. 2002; Agarwala et al. 2004]. Rather than relying on multiple frames, or modeling the consistency between objects explicitly, we have chosen to adopt Video Snapcut [Bai et al. 2009] which propagates a set of local windows along the segmentation contour with associated color and shape information.

Copy & Paste Copy & paste using Poisson image editing [Pérez et al. 2003] has the advantage that no accurate segmentation is necessary, but requires care to be taken to avoid smearing in the case of dissimilar backgrounds. Drag and Drop Pasting [Jia et al. 2006] attempts to avoid smear by computing an optimal boundary for Poisson blending. However this method still will not produce desired results for multiple (partially occluding) objects of different textures. Alpha matting [Wang and Cohen 2008] on the other hand will be able to handle such cases, and we compute alpha mattes for all segmentations in our system.

In Photo Clip-Art [Lalonde et al. 2007] objects are inserted into a target image from a database of pre-segmented and labeled images. The 3D scene structure, and lighting are estimated by image analysis and to determine which object to retrieve from the database. In our case the user explicitly selects the objects to be copied from and pasted into stereoscopic 3D images, and we address the challenges that arise with this.

Stereo Editing & Display Several stereoscopic editing approaches exist. Stereoscopic Inpainting [Wang et al. 2008] describes a segmentation-based method which exploits disparity maps to fill in missing depth and color due to occlusion in stereoscopic images. Editing methods for manipulating stereo parameters, e.g., stereo baseline, compute disparity maps to adjust the parameters locally or globally [Lang et al. 2010; Koppal et al. 2010; Wang and Sawchuk 2008]. A commercial stereo editing tool we are currently aware of is the Ocula plug-in for Nuke [The Foundry 2010]. The focus in these methods is either on foreground object removal, color correction, alignment correction, or stereo view synthesis rather than object copy & paste.

Rhee et al. [2007] introduced the concept of stereo billboards as planar proxies for stereoscopic telepresence display under the assumption that objects are always humans and fronto-parallel to the

Figure 2: *Workflow for 3D copy & paste. Given a stereoscopic pair of source and target images, the first component is depth reconstruction, which could be performed offline prior to online editing. Next the user performs segmentation and selection of the object(s) to be copied. Finally the copied object(s) is pasted into the target at some desired location, and the result is a composited stereo pair of images.*

camera. In contrast, our stereo billboards are more general: they can represent arbitrary 3D objects, and the (optimal) orientations are computed using the objects' reconstructed 3D points as constraints.

3 3D Copy & Paste

Our 3D copy & paste system allows a user to select objects from several stereoscopic source images and composite them into a desired stereoscopic target image. The editing workflow for 3D copy & paste is shown in Figure 2. Input to the system are stereoscopic pairs of images for the source and target. The system can be divided into three components:

1. Depth Reconstruction.
2. Selection.
3. Composition.

The first component, Depth Reconstruction (Section 3.1), is an essential step to determine the depth structure of the source and target scenes. The reconstructed depth is then subsequently used during selection to support segmentation; and during composition to support object placement, occlusion handling, and the stereo billboard steps. Our main challenge is in ensuring high quality results in the presence of inaccuracies in the computed depth maps.

In the next component, Selection (Section 3.2), the user selects one or more objects from source images to be copied to any desired location in the target. To support this goal, several steps are necessary in preparing the source and target images. Accurate boundary segmentation of objects, ground planes, backgrounds etc. in both source and target images is required. We have implemented an interactive segmentation tool. To reduce the amount of required user input and ensure consistent segmentations, the segmentation for the left eye is automatically propagated to the right eye image.

In the final component, Composition (Section 3.3), the user determines a desired location for pasting the copied selection in the target. Composition is performed interactively while the user is viewing the resulting composite stereoscopically. The system continuously ensures consistent orientation of the cloned object with the local orientation in the target, by computing a best-fit alignment with the targets' local underlying surface (Section 3.3.1). Furthermore, since only two views are available and to avoid the need for in-painting, the system constrains the amount of rotation and aims to keep the objects "forward facing" (Section 3.3.2). To ensure that the stereo volume of the objects is preserved, and avoid the cloned objects from appearing flat, we have developed a method we refer to as stereo billboards (Section 3.3.3). Copied objects are sorted in depth for correct occlusions (Section 3.3.4). Finally, our system computes approximated contact shadows (Section 3.3.5) to avoid the copied objects from appearing to float. We will next describe the individual components in more detail.

3.1 Depth Reconstruction

Dense depth maps with per-pixel depth values can be recovered from the stereo pair of images by computing disparities. Many methods for computing disparities based on two views have been reported in the literature and the results on reference images are compared to one another, see [Scharstein and Szeliski 2010]. Disparity computation suffers from two main problems: first, the disparities in certain areas may not correspond to the correct disparity values due to the limitations of the particular algorithm and second, the disparity values may be incorrect due to occlusions between the left and right eye images. Our system is thus designed to be able to perform copy & paste editing in the presence of (locally) inaccurate depth maps.

We assume that the camera parameters, both intrinsic and extrinsic, are known prior to loading the images into the system . We use the method presented by Smith et al. [2009] for computing the disparity map between the left and right image, and between right and left image. Using the camera parameters and the disparity maps we compute per-pixel depths [Hartley and Zisserman 2004]. For the remainder of this paper we assume that disparity maps and depth maps are one and the same and we use the terms interchangeably.

3.2 Selection

Our goal is to provide the user with the flexibility of selecting multiple objects from the source, and paste them at any desired location in the target. To support this goal, both source and target should be accurately partitioned into segments corresponding to objects, surfaces and backgrounds. Accurate real-world object segmentation requires a significant amount of user interaction in the form of strokes to mark fore- and background pixels. To reduce the amount of user interaction we have implemented an interactive multiple object segmentation approach with automatic propagation from one eye to the other.

Interactive segmentation Different objects usually have different color distributions compared to the background and we exploit this assumption to reduce the amount of required user strokes. We start the procedure by computing a mean-shift clustering [Comaniciu and Meer 2002] on the image. This results in some initial segmentation of the image into fore- and background objects. Increasing the mean-shift kernel size, more aggressively merges clusters across the entire image. We observe that, compared to the background surfaces, the foreground objects' delicate contours require a smaller mean-shift kernel size to better preserve details. We thus employ the following scheme: the user adjusts the kernel size un-

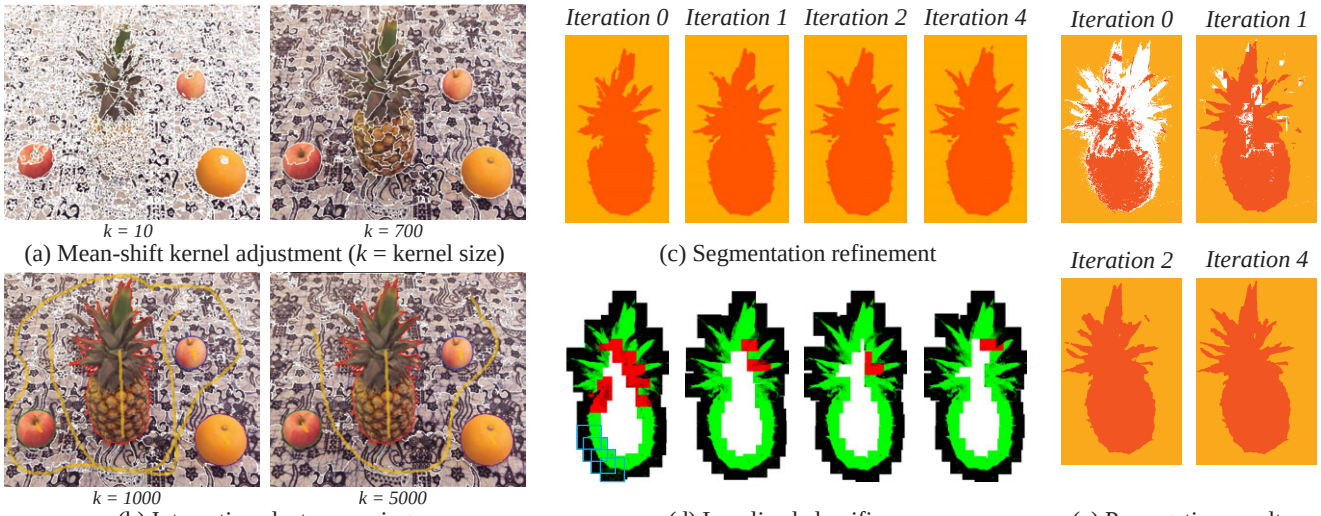


Figure 3: *(a) Adjusting the kernel size can ease multi-object segmentation, because the largest clusters usually correspond to separate objects. (b) With a larger kernel size, the user is able to do the clustering with less strokes. (c) Segmentation refinement of the pineapple through four iterations of graph cuts optimization. The corresponding probability maps estimated from overlapping localized classifiers are shown in (d). Classifier windows are only defined along the segmentation boundary, with some outlined in blue for illustration purposes. Green denote values based on local GMMs, while red values are based on the global GMM. The intensity directly corresponds to the probability. As the segmentation is improved, the number of red windows decreases. (e) Propagation results from the left eye image to the right eye image. Pixels with unknown segmentation are shown in white.*

til the foreground objects are sufficiently clustered into an initial segmentation (Figure 3a), next the user provides strokes to merge clusters and improve the segmentation of the foreground objects (Figure 3b). These two steps can be repeated until some desired segmentation of the foreground objects has been achieved. The remaining clusters of the background surface can then be merged with only a small number of strokes. Please see our accompanying video for further demonstration.

Besides using color information, we further exploit the disparities as a fourth channel in the mean-shift to improve the cluster boundaries, since disparities computed earlier already largely have discontinuities along object silhouettes. Furthermore, the user can adjust the kernel size adaptively for each object.

We merge clusters using the maximal-similarity merging mechanism [Ning et al. 2010]. Clusters covered by the users' stroke are first merged and marked as selected, and the selection is then propagated iteratively. More specifically, if cluster R is selected, we merge cluster Q with R if:

1. R and Q are adjacent, and
2. $\rho(R,Q) = \max_{S \in \mathcal{N}(Q)} \rho(Q,S)$.

Here $\mathcal{N}(Q)$ denotes the set of adjacent clusters to Q, and $\rho(R,Q)$ measures the similarity of two clusters in color and depth. Instant visual feedback is provided to the user during sketching, similar to Paint Selection [Liu et al. 2009b], allowing the user to decide whether to continue or stop sketching.

Segmentation refinement with localized classifiers User input strokes help differentiate objects in the scene. However, due to color ambiguity or estimation errors in the disparity maps, the contours of the merged clusters may not fit the object boundary accurately, as shown in *iteration 0* of Figure 3c. Therefore, after each stroke sketch, the contours are refined by applying graph cuts optimization [Boykov et al. 2001] using *overlapping localized classifiers* [Bai et al. 2009]. Bai et al. use the localized classifiers to propagate a segmentation in a current frame to subsequent frames in a video sequence. In our work, we use a similar method for both contour refinement in the left image and for propagating the segmentation to the right image. We first discuss the contour refinement for the segmentation in the left image.

Bai et al. [2009] assume an accurate segmentation of the first frame as input. They then define a set of overlapping windows whose centers lie on the segmentation boundary, shown in Figure 3d. Each window contains both background pixels (black) and foreground object pixels (green or red). Color statistics for each window are gathered, and a classifier assigns to every foreground pixel within that window, a probability of that pixel belonging to the foreground. Bai et al. advocate using small local windows. However, as stated above, our initial segmentation may be inaccurate and hence, the local statistics for small windows may be incorrect. Larger windows would then be required for the inaccurate areas along the boundary. Since there is no knowledge of where the inaccurate areas are, we create two different sized windows at each sampled location on the boundary: one small (30×30 pixels) and one larger (60×60 pixels). For each window we build a Gaussian Mixture Model (GMM) in the Luv color space using local color statistics. In addition, we use information from the whole image to build a global GMM. For each window size we then compute the model confidence for both local and global GMMs (see Equation 2 in [Bai et al. 2009]), and we pick the one with the highest confidence. We run several iterations of 2-label graph cuts refinement for each input stroke. After each iteration we update the local classifiers along the new boundary. The refinement results are shown in Figure 3c, and the corresponding local windows in Figure 3d (with foreground pixels in windows using local GMM in green, and in windows using global GMM in red).

Consistent propagation. To avoid the need for the user to repeat the segmentation procedure for the right image, we propagate the segmentation result from the left image. We exploit the disparity map and only propagate those pixels with coherent disparities between the left and right images, since those pixels tend to have classifiers with strong confidence. A pixel is said to have coherent

disparities if the difference between the disparity from the left to the right image, and the disparity from the right to the left image is one pixel or less. The initial propagation result, i.e., *iteration 0,* is shown in Figure 3e. Since the image after propagation is initially sparsely segmented, we also propagate the local classifiers from the left image. However, we compute a new global GMM on the second image using only pixels with coherent disparities. We compute the confidence values as described above and pick the one with highest confidence. We perform several iterations of $k+1$-label graph cuts for global refinement for k partitioned segments. An example is shown in Figure 3e. In addition, if the automatic propagation does not give the desired quality of segmentation, the user may provide additional strokes for refinement.

3.3 Composition

In the final component of our system the user composites (pastes) the selection in the target images. To support interactive exploration of the location for pasting, we aim for interactive performance while the user observes the resulting composite in stereo 3D. However, as explained in Section 1, composition needs to take the various aspects related to stereopsis into account: target depth composition, consistency, occlusions, and stereo volume. To address these aspects we perform the following steps:

- Alignment of the pasted object with the local underlying surface in the target.
- Constraining the rotation of the pasted object to avoid the need for in-painting or object completion.
- Stereo volume preservation using stereo billboards.
- Depth sorting to determine the correct visibility, i.e., occlusions.
- Shadow estimation using the depth map and an ambient occlusion technique.

Inaccuracies in the depth maps preclude direct artifact free rendering of the selection, either using, for example, point sample rendering [Zwicker et al. 2002], or mesh fitting [Zitnick et al. 2004]. For robustness with respect to inaccuracies in the depth maps we introduce the stereoscopic extension of billboard rendering which we have labeled stereo billboards. In the remainder of this Section we will explain the above steps in more detail. For all our methods, we represent the geometry (point clouds) of both source and target scenes in a common coordinate frame. We define the center of projection of the left eye camera as the origin of a 3D coordinate system, and align the source and target camera to lie at the origin of this frame.

3.3.1 Local Surface Orientation Alignment

In the real world, objects are typically placed on some supporting surface, e.g., a table or a sidewalk. Therefore, when an object is copied from a source to a target image, our system aims to orient it in such a way that its support surface in the source becomes aligned with an appropriate support surface in the target. As an example consider the situation in Figure 4. When the pineapple from the source scene on the left is copied into the target on the right, we aim to align the supporting table surfaces. This registration problem could be solved using a general point cloud registration technique [Besl and McKay 1992]. We observe, however, that in practice objects are mostly placed onto planar support surfaces. Therefore we use a simple strategy to align supporting planes.

During the Selection step the images have been segmented, and each foreground object and background surface is represented by a segment. For a selected object in the source we define a set $\mathcal{S}$ of neighbor segments. For example in Figure 4 the pineapple has the table as its neighbor segment. When the object is pasted into the

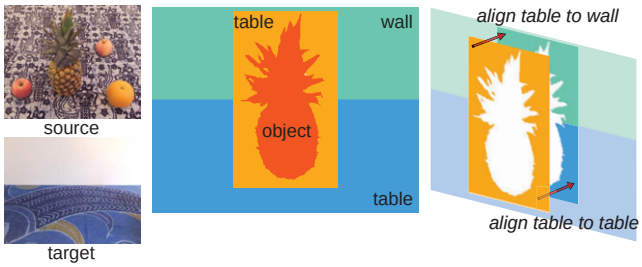


Figure 4: *Left: Source and target scene images (left eye) with different orientations of the support surfaces. Middle+Right: After the pineapple is copied and pasted into the target scene, we compute a transformation for best alignment. In this case, there are two possible alternatives, but the best choice would be to align the source's table surface to the target's table surface.*

target, $\mathcal{S}$ will overlap with a set $\hat{\mathcal{S}}$ of segments in the target scene. In the example of Figure 4, $\hat{\mathcal{S}}$ contains the target's table and wall segments. Exploiting the fact that support surfaces typically are planar, we estimate a least squares fitting plane for each $s \in \mathcal{S}$ and $\hat{s} \in \hat{\mathcal{S}}$. For each segment in $\{\mathcal{S}, \hat{\mathcal{S}}\}$ we define a coordinate frame $(R, \mathbf{t})$, with rotation $R : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ and translation $\mathbf{t}$. We define $\mathbf{t}$ as the centroid of the 3D points associated with the segment, and R is computed from the normal of the estimated plane. We then aim to find the two segments s^* and $\hat{s}^*$ with the most similar orientation, i.e., they minimize the rotation required to align the source and target segment:

$$(s^*, \hat{s}^*) = \underset{s \in \mathcal{S}, \hat{s} \in \hat{\mathcal{S}}}{\operatorname{argmin}} \parallel R_{\hat{s}} {R_s}^{1} \parallel. \tag{1}$$

The desired alignment transformation $T_A(\mathbf{x}) = R_A(\mathbf{x}) + \mathbf{t}_A$ is the transformation that aligns these two segments. It can be computed as

$$\begin{aligned} R_A &= R_{\hat{s}^*} {R_{s^*}}^{1}, \\ \mathbf{t}_A &= \mathbf{t}_{\hat{s}^*} \quad R_A(\mathbf{t}_{s^*}). \end{aligned} \tag{2}$$

Instead of using the entire segments, in practice we only use information from partial segments. Partial segments are determined by taking a predefined area around the selected object, e.g., the rectangular orange area around the pineapple in Figure 4. We denote such partial segments as patches.

3.3.2 Rotation Constraints

If one could move an object freely in 3D, parts that previously were hidden would become visible, as shown in Figure 5a. With stereoscopic input images we have no data available for the invisible parts and hence, in-painting or object completion techniques would be required to handle such rotations. To avoid these difficult tasks of in-painting and object completion, we try to keep the object's "forward facing" orientation of the source images. We accomplish this by rotating the object around the normal of the support plane computed during the alignment step.

Assume $\mathbf{t}$ is the centroid of the object in the source scene, and $\hat{\mathbf{t}}$ is its new location after being pasted into the target scene. With the alignment transformation in Equation 2 we get:

$$\hat{\mathbf{t}} = T_A(\mathbf{t}) = R_A(\mathbf{t}) + \mathbf{t}_A. \tag{3}$$

We denote the up vector of the camera as $\mathbf{u}$, and determine the angle θ between the projections of $\mathbf{t}$ and $\hat{\mathbf{t}}$ onto the ground plane (see Figure 5(a)). We can then apply a corresponding rotation R_F to ensure a target orientation as close as possible to the source orientation of the object. R_F is defined as:

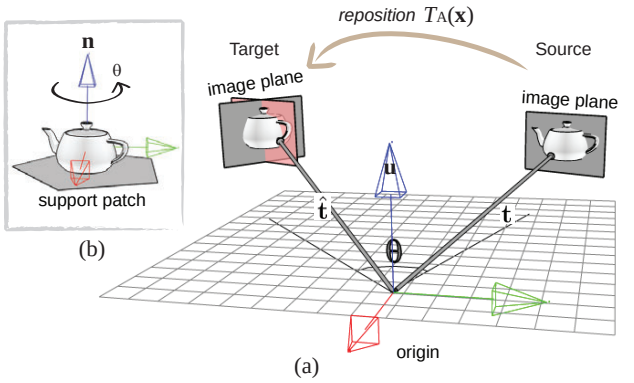


Figure 5: *(a) Due to missing data (shaded in red), only part of the repositioned object can be rendered onto the composited image. (b) We compensate the perspective change by rotating the object back, around the normal of the support patch, so that most of the available data still faces the viewer.*

$$R_F = \theta\mathbf{n}, \tag{4}$$

where $\theta\mathbf{n}$ is the so-called Euler axis–angle representation, and $\mathbf{n}$ denotes the normal of the support segment, as shown in Figure 5b. We can compute θ as:

$$\theta = \sin^{-1}\left(\frac{\|(\mathbf{t}-(\mathbf{t}\cdot\mathbf{u})\mathbf{u})\times(\hat{\mathbf{t}}-(\hat{\mathbf{t}}\cdot\mathbf{u})\mathbf{u})\|}{\|\mathbf{t}-(\mathbf{t}\cdot\mathbf{u})\mathbf{u}\|\|\hat{\mathbf{t}}-(\hat{\mathbf{t}}\cdot\mathbf{u})\mathbf{u}\|}\right), \tag{5}$$

In other words, we rotate the object around $\mathbf{n}$ at its centroid $\hat{\mathbf{t}}$ with angle θ. The rotation constrained result may not be fully satisfying to the user and we thus provide additional user control over the rotation for each pasted object in the scene.

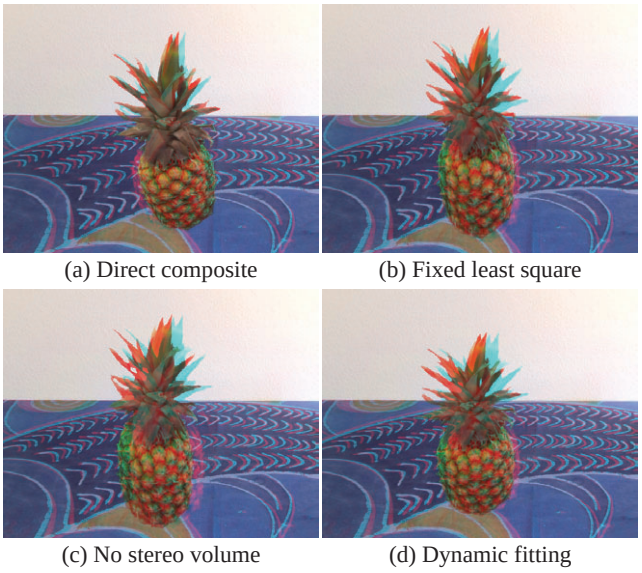
3.3.3 Stereo Billboards

The transformations T_A and R_F from above determine the desired pose of the selected object copied into in the target. Due to the inaccuracies in the computed disparities, the objects' corresponding 3D point clouds are not suitable for direct rendering. To overcome this problem we adopt the motivation from Liu et al. [2009a] to compute parametric warps for rendering. We approximate the 3D point clouds with planar proxies, and we compute homographies for the left and right eye as our parametric warps for rendering. However, the stereo volume of the source object is implicitly encoded by the 3D point cloud, and representing them by a plane could make the composited object appear flat: the so-called cardboarding effect in stereo. In order to preserve the stereo volume of the source objects in stereoscopic 3D, we introduce an approach we call stereo billboards. The goal is then to determine a *single* planar proxy, such that the error between points projected by the parametric warp, and points from the projected 3D point clouds, is minimized.

We define stereo billboards as finding an optimal common plane v, from which a pair of consistent homographies can be computed. Figure 6 sketches a particular configuration. We denote pixels of a segmented object in the left and right source images as $\mathbf{l}_i$ and $\mathbf{r}_i$ respectively, where $\mathbf{l}_i$ and $\mathbf{r}_i$ are homogeneous pixel coordinates, and each pixel $\mathbf{l}_i$, and $\mathbf{r}_i$ has associated 3D points $\mathbf{x}_i^l$ and $\mathbf{x}_i^r$. For a given plane parameterization we can project the points $\mathbf{x}_i^l$ and $\mathbf{x}_i^r$ onto the plane $p = (v^T, 1)$ resulting in $\tilde{\mathbf{x}}_i^l$ and $\tilde{\mathbf{x}}_i^r$, such that,

$$\begin{aligned} v^T\tilde{\mathbf{x}}_i^l + 1 = 0 \qquad & P^l\tilde{\mathbf{x}}_i^l = \mathbf{l}_i, \\ v^T\tilde{\mathbf{x}}_i^r + 1 = 0 \qquad & P^r\tilde{\mathbf{x}}_i^r = \mathbf{r}_i \end{aligned} \tag{6}$$

where P^l and P^r denote the camera projection matrices for the source image pairs.



(a) Direct composite (b) Fixed least square

(c) No stereo volume (d) Dynamic fitting

Figure 7: *(a) Direct composite result with source images from Figure 4. (b) Result with least square fitted plane proxy. (c) Same as (b), but now using only a single image for both left and right eye emphasizes the "cardboard" effect. (d) Our stereo billboards using dynamically optimized common plane. Our result better preserves the stereo volume.*

Given Equation 2 and 5 we can define $T(\mathbf{x}) = T_A(R_F(\mathbf{x}))$. We can then solve the following minimization problem:

$$\begin{aligned} v^* = \text{argmin}_v \textstyle\sum_i \Big(&\|\hat{P}^l T(\mathbf{x}_i^l) - \hat{P}^l T(\tilde{\mathbf{x}}_i^l)\|^2 + \\ &\|\hat{P}^r T(\mathbf{x}_i^r) - \hat{P}^r T(\tilde{\mathbf{x}}_i^r)\|^2\Big), \end{aligned} \tag{7}$$

where $\hat{P}^l$ and $\hat{P}^r$ denote the camera projection matrices for the target image pairs. Equation 7 aims to find the optimal common plane which minimizes the image space difference between the original points and the plane approximated points, in order to faithfully represent the stereo object during rendering. Figure 7 compares direct compositing, straightforward least squares fitting, and our common plane optimization of Equation 7. Stereo billboards can better preserve the stereo volume of the object.

To solve Equation 7 in the presence of inaccuracies in the disparity map, we incorporate an outlier removal step by performing an erosion on the 2D image pixels and remove the corresponding 3D points. While this does not guarantee that all outliers will be removed, in practice we found that the resulting common planes that were fitted to the remaining points gave acceptable results.

3.3.4 Occlusion

Composition of pasted objects behind other objects requires the correct handling of occlusions. Furthermore each segmented object has an associated alpha matte for handling the mixed pixels along the segmentation boundaries. Therefore, to ensure the correct order for occlusions and transparencies in rendering, we have to perform depth sorting on the objects. Methods that require per-pixel depth values for depth sorting, e.g., depth peeling [Mammen 1989], lead to interweaving objects due to the inaccuracies in the computed depth maps. We instead use the planar proxies of Section 3.3.3 for depth sorting. The overhead of having to recompute the proxy ordering is negligible since we typically only have a limited number of planes to consider in the ordering. We can interactively move the pasted layers while correctly handling the occlusions for the composite, see Section 4 for more details.

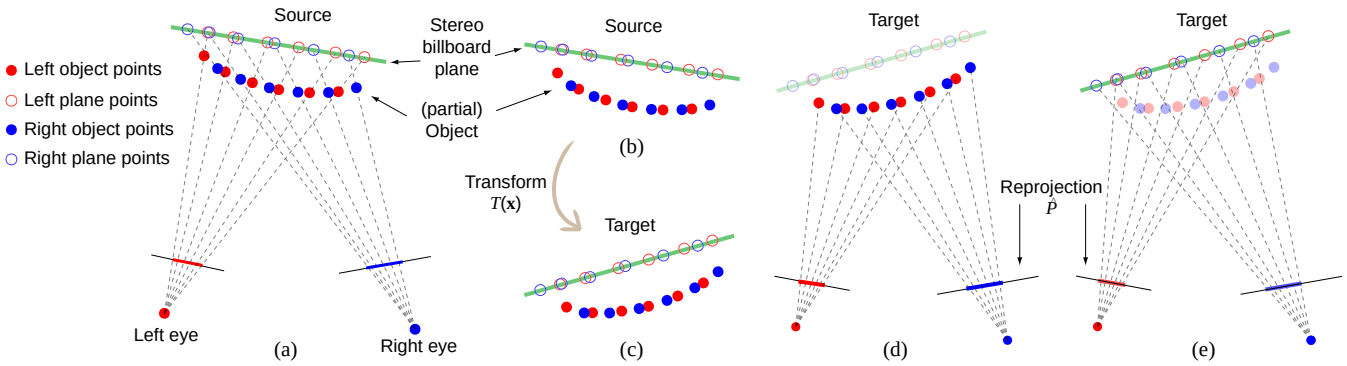


Figure 6: *This figure illustrates the computation of the common plane as the planar proxy geometry. We show pixels belonging to the left (red) and right (blue) source images with their corresponding 3D points. (a) Source pixels are back-projected onto a current estimate for a common plane (green). (b) Close-up of pixels projected onto the current common plane. (c) We use the plane proxy to reposition the object from the source to the target scene. (d) Transformed 3D points are projected onto the target left and right image. (e) Transformed 3D plane points projected onto target left and right image. The optimal common plane minimizes the difference between projected points in (d) and (e).*

(a) No shadow (b) With shadow (c) Depth map

Figure 8: *(a) Even objects with the same orientation, when copied & pasted into the same scene, will appear to be floating in the absence of contact shadows. (b) Synthesized contact shadows generated with our method. (c) Depth map used for shadow synthesis. Note: we only render the selected object and the underlying surfaces into the depth buffer.*

3.3.5 Shadow Synthesis

Shadows are an important cue for judging contact between surfaces. In the absence of knowledge about the light direction in the scene, we approximate contact shadows by using screen-space volumetric ambient occlusion [Loos and Sloan 2010]. A depth map of the composite scene is required to synthesize the shadows. We could use the point clouds to obtain depth images, but this would be inconsistent with the stereo billboard warp. Therefore, we obtain disparities for the composited scene directly from the warp instead and achieve more accurate shadows. Let $(\mathbf{l}_i, \mathbf{r}_i)$ and $\mathbf{x}_i^l$ and $\mathbf{x}_i^r$ denote a pair of corresponding points on the selected object in the source images and their associated 3D points respectively. Using v^* from Equation 7 we can project $\mathbf{x}_i^l$ and $\mathbf{x}_i^r$ onto v^* to obtain $\tilde{\mathbf{x}}_i^l$ and $\tilde{\mathbf{x}}_i^r$, and compute $\hat{\mathbf{l}}_i = \hat{P}^l T(\tilde{\mathbf{x}}_i^l)$ and $\hat{\mathbf{r}}_i = \hat{P}^r T(\tilde{\mathbf{x}}_i^r)$. The image points $(\hat{\mathbf{l}}_i, \hat{\mathbf{r}}_i)$ denote the new positions of $(\mathbf{l}_i, \mathbf{r}_i)$ in the composite target images. We then render the disparity value $\hat{\mathbf{r}}_i \quad \hat{\mathbf{l}}_i$ at pixel $\hat{\mathbf{l}}_i$ into the depth buffer of the left composite image, and vice versa for the right one. This method better preserves contours in the depth map and is also more consistent with the stereo volume. Synthesized shadows therefore exhibit less noise and better approximate the object. A computed depth map with our method is shown in Figure 8c.

(a) Direct composite (b) Our result

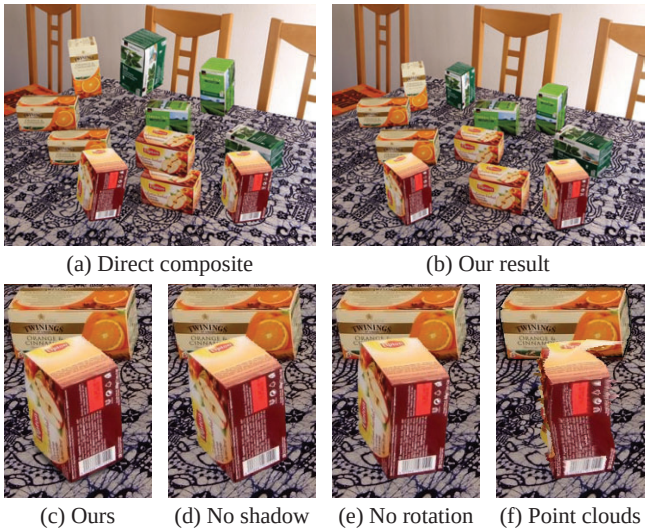
Figure 9: *(a) Direct composite result of the same object copied from different source images. Without alignment the composition looks unnatural. (b) Our alignment method can construct a more plausible composite automatically. Furthermore, we also support automatic shadow synthesis, adding to the plausibility.*

Figure 10: *The objects can be arranged onto different planes in the target scene.*

4 Results

Figure 9 demonstrates our local surface orientation alignment (Section 3.3.1). The same object is copied from different source images in Figure 15, and the object has a different orientation and scale in each source image. Without alignment the orientation and scale of the objects are not adjusted. Our alignment method on the other hand results in a (correct) composition, where the object appears to have been copied from the same source. The synthesized shadows further add to the believability of the composition. Figure 10 shows another example of our alignment. We used the color transfer method described by Reinhard et al. [Reinhard et al. 2001] in all the results presented in this paper.

Figure 11 shows that our system can handle multiple objects composed in depth. Occlusions are continuously updated while the user



(a) Direct composite (b) Our result

(c) Ours (d) No shadow (e) No rotation (f) Point clouds

Figure 11: *(a) The user can roughly place the objects into the target scene. (b) Our system will automatically arrange the objects in a right perspective and depth order. (c) Close-up of a region in (b). (d) Without shadow, the object appears to be floating. (e) Without the rotation constraint, the copied object edges are not parallel with target object edges, resulting in unnatural results. (f) Rendered results with point clouds show artifacts due to inaccurate depth reconstruction and missing data.*

determines a final location of the copied object in the target. Furthermore, by comparing Figure 11c with Figures 11d - f it is clear that shadow synthesis, local surface orientation alignment, rotation constraints, and stereo billboards significantly add to the quality of the result.

Figure 12 shows several more results of compositions with objects copied from the source images shown in Figure 15. The source objects can be copied from a wide variety of different source images. The resulting compositions look plausible with respect to the targets' scene composition. However, especially for the center image in Figure 12, the color difference between the source object and target scene breaks the believability of the composition.

5 Discussion and Future Work

We have proposed a system and methods which build on previous work for computing depth maps and performing segmentations. To address inaccuracies in the current depth maps, we have specifically aimed to make the segmentation refinement, segmentation propagation, alignment, stereo billboards, and occlusion methods robust to those inaccuracies. As explained, this is largely achieved by approximating the objects' associated 3D point clouds with proxy geometry. For simplicity, we currently use planar proxy geometry.

Approximating geometry with planar proxies has its limitations. Planar proxies do not preserve detail depth structure, such as the grass surface in Figure 13a. As a result of the absence of partial occlusions the copied object appears to float. Large orientation changes using planar proxies can introduce distortions. An example of this is shown in Figure 13b. The copied object (left person) appears distorted compared to the person in the target image. An alternative would be to use in-painting [Wang et al. 2008]. However, high quality in-painting is a difficult task and therefore typically limited to only paint in relatively small areas.

Another important problem is that planar stereo billboards may no longer respect the epipolar geometry, which could result in vertical disparities that could strongly interfere with the stereopsis. To

(a) No partial occlusion (b) Large orientation change

Figure 13: *(a) The lack of fine depth structures after planar approximation makes the copied object appear to float. (b) Large warps with planar proxy geometry leads to distortions of the copied (left) object.*

evaluate the amount of vertical disparity that is introduced, we use an object which is not well represented by a plane, shown in Figure 14. The object is copied from the source scene into two target scenes with the support surface at a different orientation: 10° and 35°. For comparison we also show the ground truth images for each case. The vertical disparities for the 10° case are around 0.8% of the object height, and for the 35° case around 2.4%. The reader can evaluate that even for the 35° orientation change the stereo images can still be comfortably fused. The maximal vertical disparity for all other result images used in this paper is around 0.5%. Although the vertical disparity tolerance varies depending on scene content, for comparison Fukuda et al. [2009] report a tolerance of 45 arcmin for random dot stereograms. Given a display at 100 dpi, viewed at a distance of 50 cm, this amounts to a vertical disparity tolerance of about 26 pixels. The vertical disparity for our 35° case is about 10 pixels. This is well within the reported tolerance, however a more thorough analysis should be conducted. In summary, our system produces plausible results for moderate orientation changes. The limitations for larger orientation changes could be overcome with more accurate depth reconstruction, but this problem of obtaining more accurate depth maps is notoriously difficult to solve robustly.

Stereo billboards help to preserve the stereo volume of the copied source object. However, if the initial depth volume in the source image is relatively flat, such as for narrow baselines (or interocular), stereo billboards will not be able to increase the stereo volume in the target. Furthermore, for large differences in baseline between source and target, stereo billboards may not be able to preserve volume. In particular achieving artistic stereo effects such as hypostereo (gigantism) and hyperstereo (miniaturization) [Koppal et al. 2010] in copy & paste is an interesting topic for future work. We may be able to exploit the work by Lang et al. [2010] in such scenarios.

For plausible appearance of copied objects, we approximate contact shadows to avoid objects from appearing to float. However, illumination differences between the source and target images is a larger problem that we did not address in this paper. This problem is not specific to 3D, see for example [Lalonde et al. 2007]. Although we use the color transfer method described by Reinhard et al. [Reinhard et al. 2001], this does not always give the desired results. For truly plausible appearance of pasted objects, more information about the scene illumination should be recovered, and exploited to relight the objects. The depth map could then also be used for shadow casting and light attenuation. However, relighting is an active area of research with no good solution to date.

Segmentations and disparity maps are closely related in that segmentation boundaries often correspond to depth discontinuities. Some disparity map methods compute an initial segmentation as a starting point for the disparity computation [Zitnick and Kang 2007]. However, in our current system segmentations and disparities are computed more or less independently. In areas where the

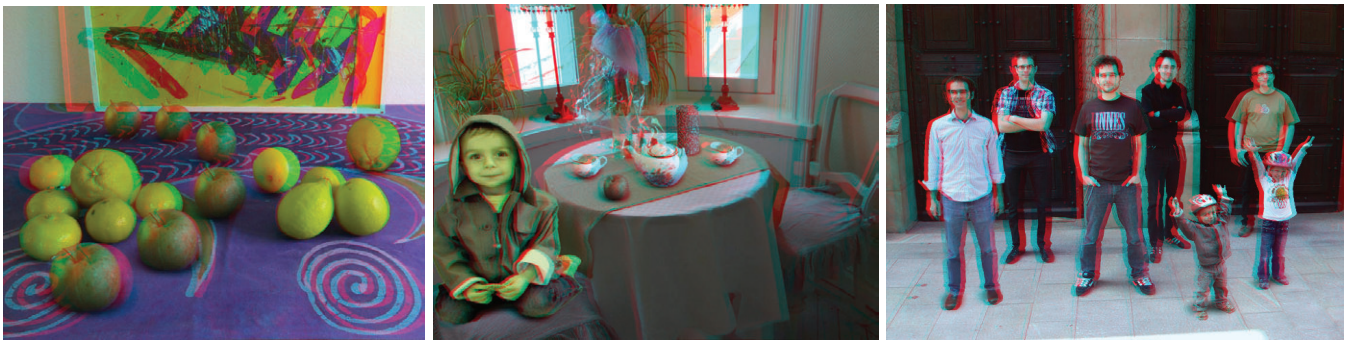


Figure 12: *Three examples of differently composed scenes using objects copied from the source images of Figure 15.*

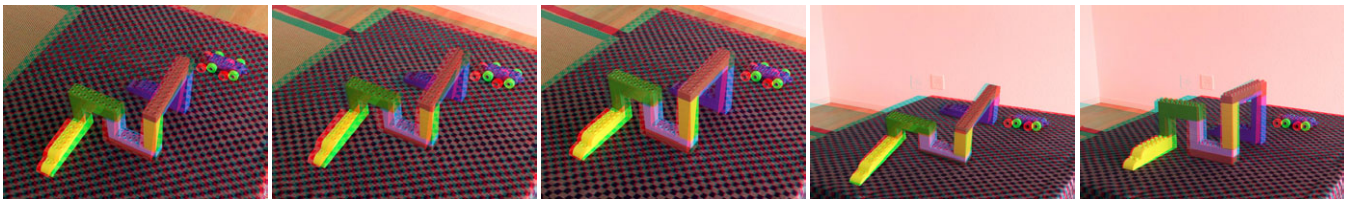


Figure 14: *Comparison of object not well represented by a plane under different orientations using stereo billboards. Two orientations are shown, together with their ground truth images.*

user needs to refine the initial segmentation in our system, the disparities are likely incorrect, and it would thus be desirable to refine the disparities for these areas simultaneously. Handling mixed pixels along the boundaries may also benefit from simultaneous refinement [Taguchi et al. 2008].

With the rapid growth in popularity of 3D the need for stereoscopic 3D compositing tools in general will grow as well. A desirable extension to this work would be 3D copy & paste for stereoscopic video. Depth reconstruction, alignment, occlusions, and depth composition will now all have to be done for dynamic objects and scenes. This would be an interesting area for further exploration.

Figure 15: *Source images for copy & paste.*

6 Conclusions

We have presented an end-to-end system for 3D copy & paste, which extends 2D copy & paste editing for still images to stereoscopic 3D. We found that for stereoscopic input images captured under casual conditions, the reconstructed depth maps are rarely accurate enough for direct artifact-free composition. Furthermore, since composition of objects is done in 3D, one has to ensure correct and comfortable stereo viewing of the resulting composite. Our main insight is that inaccurate depth maps can be appropriately approximated with simpler geometry, while still achieving high quality compelling composition results and convincing stereo viewing. To this end we have introduced stereo billboards, which approximate the reprojection of reconstructed geometry using planar warps with optimal planar proxy geometries. In addition, to increase the realism of the results, we support automatic alignment of copied objects, occlusion handling, and the generation of contact shadows. We discussed several limitations of our proposed methods such as distortions due to large warps, vertical disparities, and handling stereo baseline changes, which are interesting avenues of future work. Finally, we hope that our system serves as a start in the exploration of more general stereoscopic 3D compositing tools, in particular for stereoscopic 3D video editing.

References

AGARWALA, A., HERTZMANN, A., SALESIN, D. H., AND SEITZ, S. M. 2004. Keyframe-based tracking for rotoscoping and animation. *ACM Trans. on Graph. 23*, 3 (Aug.), 584–591.

BAI, X., WANG, J., SIMONS, D., AND SAPIRO, G. 2009. Video snapcut: robust video object cutout using localized classifiers. *ACM Trans. on Graph. 28*, 3 (Aug.).

BESL, P. J., AND MCKAY, N. D. 1992. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell. 14*, 2, 239–256.

BOYKOV, Y., VEKSLER, O., AND ZABIH, R. 2001. Fast approximate energy minimization via graph cuts. *IEEE Trans. on*

Pattern Anal. and Mach. Intell. 23, 11, 1222–1239.

CHUANG, Y.-Y., AGARWALA, A., CURLESS, B., SALESIN, D. H., AND SZELISKI, R. 2002. Video matting of complex scenes. In *Proc. of SIGGRAPH 2002*, ACM Press / ACM SIGGRAPH, J. F. Hughes, Ed., ACM, 243–248.

COMANICIU, D., AND MEER, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Anal. and Mach. Intell. 24*, 603–619.

DONG SEON, C., AND FIGUEIREDO, M. A. T. 2007. Cosegmentation for image sequences. *Int. Conf. on Image Anal. and Proc.*, 635 –640.

FARBMAN, Z., HOFFER, G., LIPMAN, Y., COHEN-OR, D., AND LISCHINSKI, D. 2009. Coordinates for instant image cloning. *ACM Trans. on Graph. 28*, 3 (Aug.).

FUJI, 2009. Finepix REAL 3D W1. http://www.fujifilm.com/products/3d/camera/finepix_real3dw1/.

FUKUDA, K., WILCOX, L. M., ALLISON, R., AND HOWARD, I. P. 2009. A reevaluation of the tolerance to vertical misalignment in stereopsis. *Journal of Vision 9*, 2 (February), 1–8.

GEORGIEV, T. 2006. Covariant derivatives and vision. *Proc. of European Conf. on Comp. Vision 4*, 56–69.

HARTLEY, R. I., AND ZISSERMAN, A. 2004. *Multiple View Geometry in Computer Vision*, second ed. Cambridge University Press, ISBN: 0521540518.

HOWARD, I. P., AND ROGERS, B. J. 2002. *Seeing in Depth, Basic Mechanics & Depth Perception*, vol. 1 & 2. I Porteous, Thornhill, Ontario.

JIA, J., SUN, J., TANG, C.-K., AND SHUM, H.-Y. 2006. Drag-and-drop pasting. *ACM Trans. on Graph. 25*, 3 (July), 631–637.

KOPPAL, S., ZITNICK, C., COHEN, M., KANG, S., RESSLER, B., AND COLBURN, A. 2010. A viewer-centric editor for stereoscopic cinema. *IEEE Comp. Graph. and Appl. Preprint*.

LALONDE, J.-F., HOIEM, D., EFROS, A. A., ROTHER, C., WINN, J., AND CRIMINISI, A. 2007. Photo clip art. *ACM Trans. on Graph. 26*, 3 (July).

LAMBOOIJ, M., IJSSELSTEIJN, W., FORTUIN, M., AND HEYNDERICKX, I. 2009. Visual discomfort and visual fatigue of stereoscopic displays: A review. *Journal of Imaging Science and Tech. 53*, 3, 030201.

LANG, M., HORNUNG, A., WANG, O., POULAKOS, S., SMOLIC, A., AND GROSS, M. 2010. Nonlinear disparity mapping for stereoscopic 3d. *ACM Trans. on Graph. 29*, 4 (July).

LIU, F., GLEICHER, M., JIN, H., AND AGARWALA, A. 2009. Content-preserving warps for 3d video stabilization. *ACM Trans. on Graph. 28*, 3.

LIU, J., SUN, J., AND SHUM, H.-Y. 2009. Paint selection. *ACM Trans. on Graph. 28*, 3 (Aug.).

LOOS, B. J., AND SLOAN, P.-P. 2010. Volumetric obscurance. In *ACM Symp. on Interactive 3D Graph. and Games*, ACM, New York, NY, USA, 151–156.

LU, F., FU, Z., AND ROBLES-KELLY, A. 2007. Efficient graph cuts for multiclass interactive image segmentation. *Proc. of the Asian Conf. on Comp. vision*, 134–144.

MAMMEN, A. 1989. Transparency and antialiasing algorithms implemented with the virtual pixel maps technique. *IEEE Comp. Graph. and Appl. 9*, 4, 43–55.

NING, J., ZHANG, L., ZHANG, D., AND WU, C. 2010. Interactive image segmentation by maximal similarity based region merging. *Pattern Recogn. 43*, 2, 445–456.

PATTERSON, R. 2007. Human factors of 3d displays. *Journal of the Soc. for Information Disp.*, 15, 861–871.

PÉREZ, P., GANGNET, M., AND BLAKE, A. 2003. Poisson image editing. *ACM Trans. on Graph. 22*, 3 (July), 313–318.

REINHARD, E., ASHIKHMIN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *IEEE Comp. Graph. and Appl. 21*, 5, 34–41.

RHEE, S.-M., ZIEGLER, R., PARK, J., NAEF, M., GROSS, M., AND KIM, M.-H. 2007. Low-cost telepresence for collaborative virtual environments. *IEEE Trans on Vis. and Comp. Graph. 13*, 1, 156–166.

ROTHER, C., KOLMOGOROV, V., AND BLAKE, A. 2004. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. on Graph. 23*, 3 (Aug.), 309–314.

ROTHER, C., MINKA, T., BLAKE, A., AND KOLMOGOROV, V. 2006. Cosegmentation of image pairs by histogram matching. *IEEE Conf. on Comp. Vision and Pattern Recog.*, 993–1000.

SCHARSTEIN, D., AND SZELISKI., R. 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. Journal of Comp. Vision 47*, 1/2/3, 7–42.

SCHARSTEIN, D., AND SZELISKI, R., 2010. Middlebury Stereo Repository. http://vision.middlebury.edu/stereo/.

SHUM, H. Y., SUN, J., YAMAZAKI, S., LI, Y., AND TANG, C. K. 2004. Pop-up light field: An interactive image-based modeling and rendering system. *ACM Trans. on Graph. 23*, 2 (Aug.), 143–162.

SMITH, B., ZHANG, L., AND JIN, H. 2009. Stereo matching with nonparametric smoothness priors in feature space. *IEEE Conf. on Comp. Vision and Pattern Recog.*, 485–492.

TAGUCHI, Y., WILBURN, B., AND ZITNICK, C. L. 2008. Stereo reconstruction with mixed pixels using adaptive over-segmentation. *IEEE Conf. on Comp. Vision and Pattern Recog.*, 1–8.

THE FOUNDRY, 2010. Nuke - Ocula Plug-in. http://www.thefoundry.co.uk/.

WANG, J., AND COHEN, M. F. 2008. Image and video matting: A survey. *Foundations and Trends in Comp. Graph. and Vision 3*, 2, 97–175.

WANG, C., AND SAWCHUK, A. A. 2008. Disparity manipulation for stereo images and video. *Stereoscopic Disp. and Appl. 6803*, 1, 68031E.

WANG, L., JIN, H., YANG, R., AND GONG, M. 2008. Stereoscopic inpainting: Joint color and depth completion from stereo images. In *IEEE Conf. on Comp. Vision and Pattern Recog.*, 1–8.

ZITNICK, C. L., AND KANG, S. B. 2007. Stereo for image-based rendering using image over-segmentation. *Int. Journal of Comp. Vision 75*, 1, 49–65.

ZITNICK, C. L., KANG, S. B., UYTTENDAELE, M., WINDER, S., AND SZELISKI, R. 2004. High-quality video view interpolation using a layered representation. *ACM Trans. on Graph. 23*, 3 (Aug.), 600–608.

ZITNICK, C. L., JOJIC, N., AND KANG, S. B. 2005. Consistent segmentation for optical flow estimation. *IEEE Int. Conf. on Comp. Vision*, 1308–1315.

ZWICKER, M., PFISTER, H., VAN BAAR, J., AND GROSS, M. 2002. Ewa splatting. *IEEE Trans. on Vis. and Comp. Graph. 8*, 3, 223–238.