EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling (2011)
T. Hammond and A. Nealen (Editors)

StereoBrush: Interactive 2D to 3D Conversion Using Discontinuous Warps

O. Wang[1] and M. Lang[1,2] and M. Frei[2] and A. Hornung[1] and A. Smolic[1] and M. Gross[1,2]

[1]Disney Research Zürich
[2]ETH Zürich

Abstract

We introduce a novel workflow for stereoscopic 2D to 3D conversion in which the user "paints" depth onto a 2D image via sparse scribbles, instantaneously receiving intuitive 3D feedback. This workflow is enabled by the introduction of a discontinuous warping technique that creates stereoscopic pairs from sparse, possibly erroneous user input. Our method assumes a piecewise continuous depth representation, preserving visual continuity in most areas, while creating sharp depth discontinuities at important object boundaries. As opposed to prior work that relies strictly on a per pixel depth map, our scribbles are processed as soft constraints in a global solve and operate entirely on image domain disparity, allowing for relaxed input requirements. This formulation also allows us to simultaneously compute a disparity-and-content-aware stretching of background areas to automatically fill disoccluded regions with valid stereo information. We tightly integrate all steps of stereo content conversion into a single optimization framework, which can then be solved on a GPU at interactive rates. The instant feedback received while painting depth allows even untrained users to quickly create compelling 3D scenes from single-view footage.

1. Introduction

Since its introduction in the 1950s, stereoscopic cinema has experienced several cycles of boom and decline. One reason for its failure to achieve sustained adoption has been inadequacies in display technology, which led to uncomfortable viewing, eye strain, and visual fatigue. While not completely solved, technical advances have greatly alleviated these problems. This is not as true however, for the second fundamental bottleneck: the *creation* of stereoscopic content. Shooting directly with a stereo camera is one option, but shot planning, costly hardware, camera rig control, and extensive post-processing required to fix stereographic errors render this process both cumbersome and expensive.

For these reasons, 2D to 3D conversion has become an important and widely-used approach for stereo content creation, both for cinema and home 3D devices, as it allows producers to retain a well-established production pipeline including hardware, workflows, and software tools. In addition to the creation of novel content, the industry is also in urgent need of solutions for cost-efficient conversion of existing legacy footage and content from ongoing 2D production. Existing high quality solutions are largely manual, consisting of careful segmentation of individual objects using rotoscoping, precise per-pixel depth assignment by drawing or creating proxy scene geometry, and stereo view synthesis, including inpainting missing image content in disoccluded regions. Each of these steps is highly ill-posed, and to date no computational solution exists that could replace the currently employed manual procedures in high-quality production. This leads to costly iterations between production stages, and places a high burden on artists performing the manual conversion. Companies specializing in 2D to 3D conversion charge up to $100,000 per *minute* of converted footage, making high quality conversion prohibitive in many cases.

We address these issues by proposing a novel workflow for stereoscopic content creation that fully integrates all the required steps, providing the user with *instant visual feedback*, and enabling a new intuitive process of "brushing" depth directly into scenes. This novel interaction paradigm is supported at its core by a *single* integrated method that effectively comprises all of the aforementioned 2D to 3D conversion steps, and is specifically designed to process sparse and possibly inaccurate user input.

The basic assumption we make is that our perception of depth in many real world scenes is *piecewise smooth*,

© The Eurographics Association 2011.

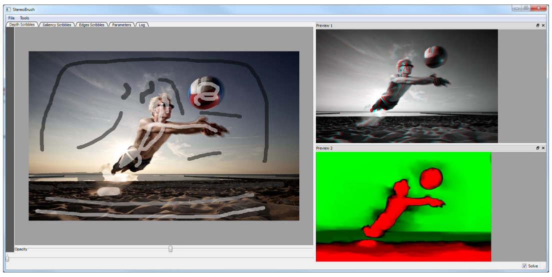

Figure 1: *An example of our user interface. Sparse scribbles are drawn on the left, and the resulting stereo is displayed on the right along with a visualization of disparity.*

with important discontinuities at object boundaries. Stereo 3D display works by presenting different images to the left and right eyes. The horizontal difference between these images is called the "disparity", and is the main source of percieved depth in stereo images. In our approach, the user brushes sparse scribbles on images or keyframes of an input video, indicating rough depth values. These are mapped into a comfortable disparity range following common stereographic practice, and are spread out over the image or video sequence. This forms our disparity hypothesis, which unlike in prior methods, is *not* directly used for view synthesis, but rather serves as a soft constraint in a global, discontinuity preserving method that uses visual saliency information to smooth out deformations, hiding artifacts in less salient image regions. Our framework is also designed to allow the use of a novel hole filling method for removing disocclusions by means of a content-and-disparity aware stretching of background regions.

By formulating our problem into a single integrated method, all processing steps can be efficiently implemented on the GPU, allowing for iterative depth brushing and refinement that can create plausible stereoscopic output in significantly less time than prior methods. In addition, as our method works on image-domain disparities and makes no assumptions about scene geometry, we can apply 2D to 3D conversion to scenes with no consistent underlying 3D model. This allows for the conversion of drawings (which are particularly difficult for traditional view synthesis approaches), and for artistic 3D depth manipulation. Figure 1 shows how this workflow appears to the user.

To summarize, the main novel contributions of our work are:

- An integrated workflow for 2D to 3D conversion based on sparse scribbles.
- A warping procedure that preserves important stereoscopic discontinuities and automatically handles disocclusions by stretching background regions.

2. Related Work

To create a stereoscopic pair from monoscopic input, a virtual view must be generated close to the original, requiring some notion of scene depth. If accurate depth is available, such view synthesis can be done by a forward-projection method [CW93]. As general 2D to 3D conversion cannot rely on multiple camera depth estimation [ZKU*04] or a priori known depth maps [DRE*10], depth has to be estimated from a single view. Structure from motion (SFM) in video sequences [ZHQ*07, KKS07] is another possibility, but is only applicable for certain scene types and camera motion.

Therefore, high quality 2D to 3D conversion today often relies on user-assisted methods. In these methods, users provide input such as normals, creases and outlines [ZDPSS01], object silhouettes across multiple frames [vdHDT*07], or over/under cues [SSJ*10].

In another recent work by Ward et al. [WKB11] a sophisticated 2D to 3D conversion system is presented that incorporates automatic depth computation techniques with interactive user controls. This work elegantly encapsulates the classical conversion pipeline mentioned above, allowing for various forms of user input at different stages, including SFM corrections, selective object depth scaling and geometric template assignment. While this work presents an interactive user interface, our interaction paradigm is fundamentally different. We design the entirety of our user interaction as an intuitive 'brushing' metaphor. Similar to drawing or painting, our approach allows users to operate entirely within the *image* domain without creating a 3D reconstruction of the scene. By directly painting disparities, our workflow is not limited by perspective projection restrictions, such as a geometrically consistent scene and camera model. As a result, unlike the above methods, we support the conversion of non-realistic animated content, as well as *artistic* modification of scene depth.

Akin to our method, Guttman et al. [GWCO09] also leverages scribbles as input. These scribbles are used to train a support vector machine classifier that assigns depth labels to each pixel in a video sequence. Our solution differs from this in that we envision an iterative brushing interface, where instant visual feedback is a fundamental part of our workflow. A subsequent fundamental difference of our proposed workflow to this and all of the above methods is that in prior work, a dense depth map is estimated and then used to project pixels into a new viewpoint. Instead, we introduce these disparities only as *soft constraints* in a warping-based disparity-space view synthesis optimization, which enforces a piecewise smooth depth model that minimizes visible distortions in a non-local sense, allowing possible inaccuracies in the depth hypothesis to be corrected before display. In addition, we present a *single* optimization procedure that encompasses the entire conversion process, allowing user input to be expressed at a single stage of the workflow.

Our method is motivated by recent locally-varying, continuous image warping methods that have been shown to be successful in numerous other application domains, such as: image retargeting [SS09], perspective modification [CAA09], and stereoscopic editing [LHW*10]. In the last work, it was briefly mentioned that image warping could be used for 2D to 3D conversion. However, by assuming a globally smooth representation of scene depth, this method blurs sharp disparity edges, which in 2D to 3D conversion

© The Eurographics Association 2011.

eliminates important cues for stereo fusion, making stereo fusion difficult. We improve upon this method by introducing a novel, piecewise-smooth image warping procedure that preserves discontinuities at important regions.

Some variations of warping do not enforce image continuity, allowing regions to be removed [AS07], or reshuffled [SCSI08]. However, these methods are not well suited for 2D to 3D conversion, as they assume a piecewise-constant depth representation, and do not preserve input scene composition, making it difficult to enforce stereoscopic consistency.

Our method also proposes a novel solution to the hole filling problem. Traditional automatic inpainting methods find similar appearance image patches to fill hole regions (we refer the reader to a recent state-of-the-art report for a summary of these methods [WLKT09]). However, inpainting is a challenging problem, and it is made more difficult with stereoscopic content as image features need to have consistent disparity across the two generated views. We provide an alternative approach to inpainting, where we modify the scene structure by undetectably stretching background regions into disocclusions.

3. Method

Our discontinuous warping method works as follows. First, the scribbles are mapped into comfortable stereo disparities, and are then spread through the image or video sequence with an edge-aware technique. This forms a "disparity hypothesis" that is used as a confidence-weighted soft-constraint that is combined with a discontinuity preserving smoothness term. Finally our method is designed such that a content-and-disparity-aware resizing method can be used to optimally stretch background regions and fill in disocclusions. We first present our method as three components that together handle the process of 2D to 3D conversion, and afterwards show how all of these three steps can be combined into a single solve. Interaction occurs on images, or on a per-keyframe level for videos, after which an offline process computes a temporally consistent propagation of keyframe information to create the final video.

Definitions The goal of our method is to map input image I into a stereoscopic pair I^l, I^r (l and r indicate left and right). To do this, we compute two image warps Ω^l and Ω^r, where $\Omega : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is a mapping such that for image I, $\Omega(I) = I'$ where I' is the warped image. When creating a stereo pair from a single image, we want to introduce only horizontal disparities between I^l and I^r, as vertical disparity can cause visual fatigue and difficulty in fusing the stereo pair. Therefore, the warp Ω modifies only x coordinates, which has the additional advantage of reducing the complexity of the linear solution. For simplicity, we define subscripts as values at that pixel; so I_i is the image color at pixel i, and Ω_i is the x coordinate of the warp. To find Ω, we define an energy minimization problem that includes a *data* term E_d, enforcing stereo disparity at each pixel, and *discontinuity preserving smoothness* terms E_s^l, E_s^r that preserve visual appearance.

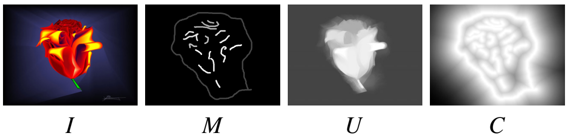

Figure 2: *Scribble Propagation. (I) Input image , (M) user provided scribbles, (U) dense scribble propagation, (C) confidence map, the distance transform to nearest scribble.*

These are combined with a *hole filling* method that stretches backgrounds to fill discontinuities.

3.1. Disparity Hypothesis

Users provide a sparse and rough scribble map M, where higher intensities represent "closer" objects. Our interface allows users to create scribbles with a standard brush, as well as a gradient brush, which is useful for surfaces with smooth depth variation. After placement, users can interactively scale the depth of each scribble while viewing the result in 3D, allowing them to settle on a desired perceptual effect of each stroke.

We then propagate these sparse scribbles into a per-pixel set of constraints using a global sparse data interpolation technique originally proposed for colorization [LLW04], which for completeness we will now describe. Using this method, we compute a dense scribble propagation U, such that the difference between the value at U_i and a weighted sum of its neighbors is minimized for each pixel i:

$$\arg\min_U \sum_{i \notin M} \left\| U_i - \Sigma_{q \in N(i)} w_{iq} U_q \right\|^2 + \sum_{i \in M} \left\| U_i - M_i \right\|^2 \quad (1)$$

where $N(i)$ are the neighbors of pixel i, and w_{iq} is a weighting function that sums to one and is large when I_i and I_q are similar. We allow user scribbles to be sparsely placed, not only spatially, but also temporally at keyframes, propagating values across an entire video sequence by including temporal as well as spatial neighbors in Eq. (1).

We incorporate a confidence weight C into our scribble propagation U, which we compute as a distance transform of M. This incorporates the idea that our disparity hypothesis will be more reliable close to the original scribbles. Our confidence term allows for the scribble propagation to have a lower contribution to the resulting warp in areas that are more likely to be incorrect (farther from any user input). Figure 2 shows each of these components.

User scribbles indicate a rough concept of "closeness". These values are mapped into a pleasant stereo disparity range, using a disparity operator ϕ, as presented in [LHW*10], that adheres to the rules of stereography. For the examples we present in the paper, we choose a simple linear operator ϕ that maps scribbles into a comfortable minimum and maximum disparity range, based on the image size, and expected display size and viewer distance. However, we note that at this stage, any nonlinear operators ϕ could be used, allowing for stereographic rules to be easily enforced in our

© The Eurographics Association 2011.

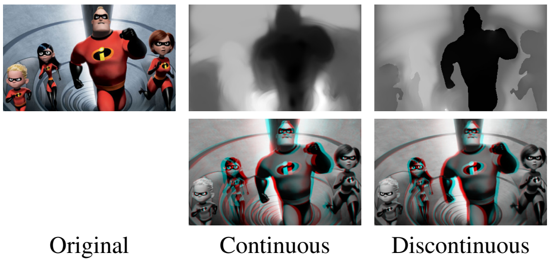

Figure 3: *Comparison of discontinuous versus continuous warping. From left to right: original image, resulting disparity using continuous image warping from Lang et al. [LHW*10], resulting output disparity using our discontinuous warping method. Sharp disparity boundaries are clearly visible in our method. This effect is most visible when viewed zoomed-in on a desktop monitor.*

method. We can write the final per-pixel user input disparity hypothesis D, as

$$D = \phi(U) \tag{2}$$

This is used to construct the data term (E_d) of a minimization problem that ties together spatial warps Ω^r and Ω^l, and enforces a confidence weighted disparity for each pixel i:

$$E_d = \sum_i C_i \left\| (\Omega_i^r - \Omega_i^l) - D_i \right\|^2 \tag{3}$$

Minimizing this error term with respect to Ω^r, Ω^l results in a simple forward mapping operator, where each pixel moves relative to its disparity. This would enforce that the disparity between I^l, I^r be exactly the disparity hypothesis D, but would introduce visual artifacts at locations with any errors in D, as well as holes in the result.

3.2. Discontinuity Preserving Smoothness

To fix these problems, we enforce our piecewise-continuous disparity assumption by introducing a discontinuity preserving, visual saliency-weighted smoothness term, E_s. The purpose of the smoothness term is twofold. First, it directs errors in the hypothesized depth to discontinuity locations, preserving object structure, while minimizing visual artifacts. In addition, it determines which parts of a scene can be most stretched for our hole filling method.

As discontinuities are very important for creating realistic looking depth effects, our method yields more realistic reconstructions than prior work [LHW*10]. In this image, we display the difference in methods both in anaglyph, as well as by visualizing the disparity of the final synthesized image pair as a grayscale image.

To compute a saliency map S, we use an existing state of the art method from Goferman et al. [GZMT10] combined with a simple gradient magnitude term. For most cases, this saliency map is sufficient, but our interface allows for user adjustments to S using soft "dodge" and "burn" style brushes when necessary.

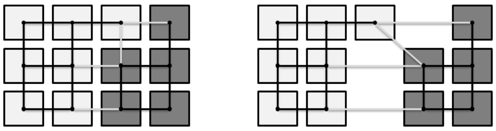

Figure 4: *Discontinuity preserving smoothness. A grid of pixels is shown as grayscale squares before (left) and after (right) warping. Lines connecting pixel centers indicate edges δ_{iq}. Standard edges ($\delta_{iq} = 1$) are black, while edges at potential discontinuity locations ($\delta_{iq} = \varepsilon$) are gray.*

For 2D to 3D conversion, the preservation of depth discontinuities is perceptually very important. Our smoothness term therefore penalizes local distortions, weighted by the saliency S, *except* at discontinuity locations, which are represented by a binary edge map G, which we will shortly discuss how to compute. Figure 4 shows a diagram of how a discontinuity could look like in a pixel representation.

Discontinuities between neighbor pixels $q \in N(i)$ are incorporated into our pipeline as an additional weight δ_{iq} on the smoothness constraint using the following formulation:

$$\delta_{iq} = \begin{cases} \varepsilon & \text{if } i,q \in G \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

Using a small, non-zero ε is advantageous as it yields visually similar results to zero weight, but does not decrease the rank of the coefficient matrix in our linear solve, regardless of the location or number of discontinuities. For all of the examples presented in this paper, we use a value of $\varepsilon = .001$. This gives us the discontinuity preserving smoothness term (evaluated similarly for both left and right warps),

$$E_s = \sum_i S_i \left(\sum_{q \in N(i)} \delta_{iq} \left\| (\Omega_q - \Omega_i) - \Delta_x(i,q) \right\|^2 \right) \tag{5}$$

where, $\Delta_x(i,q)$ is the x distance between i and q in the original image.

Now we discuss how to determine locations for discontinuities in G. Ideally, we would like all discontinuities to match real depth edges in the image, but the problem of distinguishing depth edges from image edges in a single view is a highly under-constrained problem. Fortunately, we can use the existing user scribbles and disparity hypothesis D to help disambiguate the two. Image edges in G that exist where there is no change in D will not produce discontinuities in the final result, as the data term will be similar on both sides of the edge. In addition, locations where depth edges exist in the absence of image edges (such as two white walls at different depths with the same albedo & lighting) are not as crucial to model discontinuities at, due to a lack of features for stereo fusion. Therefore, our algorithm is largely robust in terms of excess edges in G, and we are able to use an over-detection of image edges as an initial guess of discontinuity locations. These edges are computed using a standard gradient magnitude edge detection method. We show the robustness to over-detection of this approach in Figure 5. G can

© The Eurographics Association 2011.

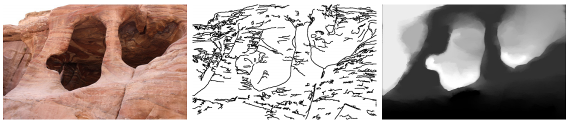

Figure 5: *Original image, edge over-detection G (dilated for clarity) and the output disparity map, showing our method's robustness to over-detection of edges.*

also be modified by the user. This is typically done through a brush that removes extra image edges and allows for adding small corrections where necessary discontinuities cannot be detected by image edges.

We solve for Ω^l, Ω^r by minimizing the data term E_d and discontinuity preserving smoothness term E_s. The stereo per-pixel disparity of our resulting synthesized stereo pair can also now be computed: $D'_i = \Omega^r_i - \Omega^l_i$. We note that $D' \neq D$, as the output disparity D' is computed considering the smoothness term E_s as well. In the resulting stereo pair $\Omega^l(I), \Omega^r(I)$, overlaps and disocclusions will occur at depth discontinuities. In the case of overlaps, our output disparity D' can be used to determine correct ordering, however, disocclusions still need to be addressed.

3.3. Hole Filling

A common problem with image-based view synthesis is the need to fill disocclusions that arise from parallax. Traditionally, these regions are filled by hallucinating content. However, our locally-varying warping framework is designed to allow an elegant and novel solution to the hole filling problem. We use a method that is motivated by recent advances in image retargeting, demonstrating that large changes of aspect ratio can be hidden by use of visual saliency [RGSS10]. These methods operate by stretching and squeezing different parts of the image in a way that hides distortions in less salient regions. We present a solution that fills background content enforcing a globally optimal *content-and-disparity*-aware stretching of nearby image content.

Solving for a final, hole filled warp, which we call $\hat{\Omega}^l, \hat{\Omega}^r$ is done by first computing a new set of discontinuity weights $\hat{\delta}_{iq}$, for all $i, q \in G$, such that the penalty for local distortion at disoccluded edges is reinstated unless they form an overlap, in which case the weight remains the same as before:

$$\hat{\delta}_{iq} = \begin{cases} \delta_{iq} & \text{if } i, q \text{ form an overlap} \\ 1 & \text{otherwise} \end{cases} \qquad (6)$$

We fix the locations at discontinuity location $\hat{\Omega}_i = \Omega_i$ if i is in front of q (or in other words, $D'_i < D'_q$).

This serves to "pull-over" neighboring content to the border of the disoccluded region, while preserving the location of the foreground object. Simultaneously, the smoothness term E_s acts to direct distortions to the visually less significant areas, and the data term E_d maintains the stereoscopic relationship between left and right images in stretched regions. Figure 6 shows a diagram of how this looks on a per pixel level for a single view, and Figure 7 shows an example result on an image. Minimizing E_d and E_s with new $\hat{\delta}$ and fixed locations yields $\hat{\Omega}^l, \hat{\Omega}^r$.

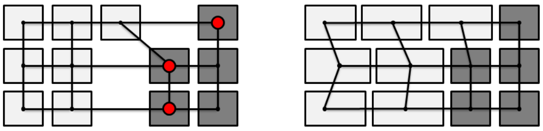

Figure 6: *Hole filling. A disocclusion occurs between light and dark pixels (dark pixels are in front). Pixels on the foreground of the disocclusion are fixed (shown by red circles), and all edges weights are restored $\hat{\delta}_{iq} = 1$. After hole filling (right), the image is stretched to fill the disocclusion.*

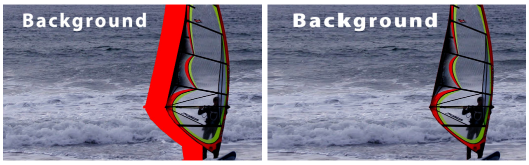

Figure 7: *Extreme example of image stretching for hole filling. In the left image, a disoccluded region is shown in red. After stretching the content, the area has been filled.*

This solution produces compelling hole filling results, maintaining image appearance and disparities in a least squares optimal sense. In addition, it creates valid monocular cues at disoccluded regions (which help with stereo fusion [HW09]). All of these advantages come at the expense of a physically accurately reconstructed scene, but given the amount of distortion required, we did not find that this caused any noticeable conflicts.

3.4. Error Minimization

As it has been described, solving for our optimal warps requires three separate minimization steps. The first, to propagate the scribbles and solve for D, the second to solve for Ω^l, Ω^r (with discontinuities), and the third to create the final output $\hat{\Omega}^l, \hat{\Omega}^r$ after hole filling. However, for efficiency, we can combine the first two steps, using the scribble propagation from Eq. (1) with the data term from Eq. (3). This is accomplished by replacing U with disparities $(\Omega^r_i - \Omega^l_i)$, essentially minimizing the difference in disparity between neighbor pixels, rather than the difference in scribble color.

$$\begin{aligned} E'_d = \sum_{i \notin M} C_i \left\| (\Omega^r_i - \Omega^l_i) - \textstyle\sum_q w_{iq} (\Omega^r_q - \Omega^l_q) \right\|^2 \\ + \sum_{i \in M} \left\| (\Omega^r_i - \Omega^l_i) - \phi(M_i) \right\|^2 \end{aligned} \qquad (7)$$

Our final minimization can now be expressed as:

$$\underset{\hat{\Omega}^l, \hat{\Omega}^r}{\arg\min} \left\| E'_d + w_s (E^l_s + E^r_s) \right\|^2 \qquad (8)$$

where w_s is a user controllable weight to determine the

© The Eurographics Association 2011.

amount of discontinuity preserving smoothness. This value can be interactively adjusted as well, but in most examples we use $w_s = 400$. This large number reflects the smaller range of the smoothness domain $[0,1]$ as compared to the data terms, which is on the order of pixel coordinates. From this we have the disparity of our final synthesized image pair $\hat{D}' = \hat{\Omega}^r - \hat{\Omega}^l$.

Our implementation also allows support for user-drawn line constraints, as well as temporal smoothness in Ω^l, Ω^r, for which we refer the reader to prior work for a more detailed description [KLHG09].

We express the error minimization in Eq. (8) as a linear system of equations $Ax = b$, and represent Ω as a regularly structured, per-pixel grid. By phrasing our problem in these terms, we can use a multi-scale GPU implementation that iteratively solves for x, achieving interactive rates. At each stage, from coarse-to-fine resolution, each pixel i independently (and in parallel) solves for its local minimum given its neighbors q, and updates itself. This whole procedure is iterated several times. Binary maps, such as the edge map G, are down-sampled using a maximum-filter prior to sub-sampling to prevent continuous lines from becoming disconnected.

3.5. Rendering

The final step is to render the stereo pair I^l and I^r from $\hat{\Omega}^l(I), \hat{\Omega}^r(I)$. We use a hardware-accelerated method, which computes anisotropic pixel splats [ZPvBG02, KLHG09] for each warped pixel and guarantees alias free high quality rendering. We note that our final warp may contain pixels that overlap each other. In these cases, we can use the computed pixel disparity map $\hat{D}'$ to disambiguate over/under regions and properly render occlusions.

Our method preserves sub-pixel image boundaries at disocclusions, as the content-aware stretching is distributed over the image and not strictly at the boundary. However, at overlapped regions, some form of matting should be used. In the results that we present, we use a simple feathering technique on a small number of pixels around overlaps. However, if high-quality alpha mattes are available, they can be incorporating into rendering when drawing foreground regions at overlapped areas.

4. Results

We demonstrate results from our method on a wide range of image types (Figure 8). 3D stereo pairs are shown as grayscale, red-cyan anaglyph images (👓) but we encourage readers to view the included supplementary results on a high quality 3D display if possible. As with any 3D viewing system, perceived depth is a function of the size of the image viewed, and the distance to the viewer. Images shown in this paper have been optimized for zoomed in, on-screen display. The horizontal disparity in the synthesized stereo pair ($\hat{D}'$) is visualized as a grayscale image. This visualization is very helpful to see the exact disparity in the output stereo image pair. Both sharp depth discontinuities and smooth changes in depth are visible in these images. We can see from our results that our method achieves convincing 3D depth impressions on a wide range of images, from photographs to rendered images and drawings. In addition, the sparse and rough nature of user brush strokes used to generate these results is apparent. However, these images can only give a rough idea of our workflow, and we refer to our supplimentary video for a screen capture of an interactive editing session.

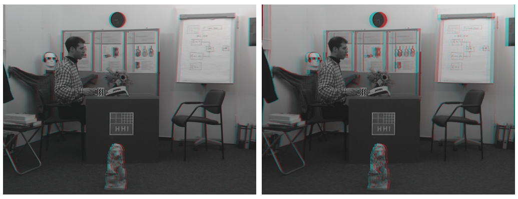

Figure 9: *Ground truth comparison: Left) captured stereo, right) converted from one view of the stereo pair (total time spent on conversion: 2 minutes).*

Timing Images were created by an experienced user of our interface. Rough 3D shape becomes visible after only a few seconds of interaction, and the final versions were all completed in under 5 minutes of refinement. Most interaction centered around painting a small number of unique depth brush strokes, with less frequent modifications to the saliency and edge maps. Running on a desktop computer with NVIDIA GeForce GTX 480 GPU and HD resolution images, our method produces continual updates at a rate of roughly 2 fps. This operates in a separate thread from the interface, so the output is constantly updated without slowing down interaction. Given the time it takes for the user to shift attention between input and 3D visualization, we found that this provided a convincing interactive experience. For video, our workflow involves interactive keyframe editing, with offline video creation. Our method operates on shots and assumes a temporally smooth warp between frames. Scribble information is propagated temporally over the entire shot, allowing sparse input at keyframes, but requiring a larger, offline multi-grid solve, which on our desktop computer, took approximately 1 minute per 4 frames of video (depending on the length of the shot).

Validation For validation of our method, we show a comparison between a ground truth stereo image and a stereo pair created using our 2D to 3D conversion method (Figure 9). We note that we do not aim for an exact replication of the ground truth pair, only to produce an image with a similar depth impression.

In addition, we conducted a user-study measuring the perceived quality of real vs converted stereo images (Figure 10). In this study, we randomly selected 15 of the highest ranked stereo-camera images from Flickr, sorted by interestingness, with 15 of our converted examples. We then asked 31 participants to rate each 3D stereo pair on how convincing its stereo effect was (disregarding image composition). Our study found no measurable preference of image source (stereo camera or converted). To count for user bias, we subtracted the mean score of each user, and then normalized by

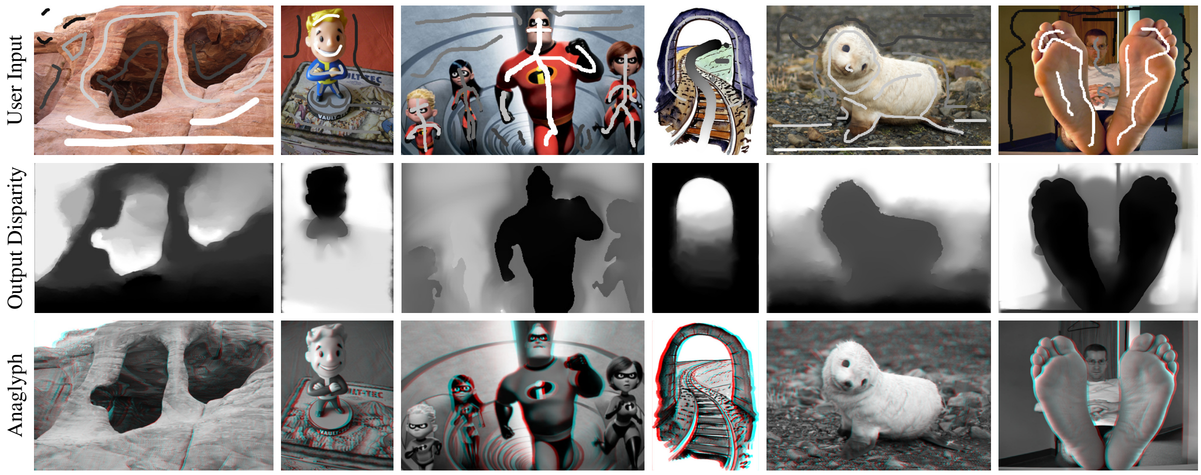

Figure 8: *2D to 3D conversion results, showing user input, an output disparity visualization, and grayscale anaglyph output.*

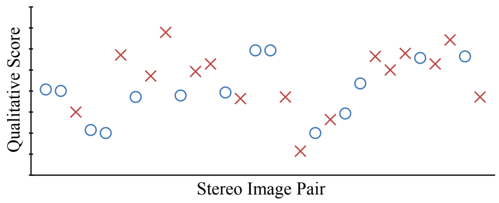

Figure 10: *User study showing no preference for real stereo over converted images. Scores for images converted with our method are shown as blue* ○*'s. Scores for high"interestingness" Flickr stereo pairs are shown as red* ×*'s.*

variance. The point-biserial correlation coefficient between the bias-corrected user ratings of each image and its source was $r_{pb} = -0.0019$, with a p-value of $p = 0.9921$. This indicates with high statistical confidence that the null hypothesis is true, and that in our study, there was no correlation between image source and user rating.

Additional Applications In addition to standard 2-view stereo, our method can be used to generate N-view images for autostereoscopic displays by using a data term that defines disparities between $\Omega_1 \ldots \Omega_n$ image warps. Our method provides convincing results up to a certain distance that is typically sufficient for the limited range of autostereoscopic displays.

Our discontinuous warping technique can also be used for applications other than 2D to 3D conversion, such as in image retargeting. We apply our method to retargeting by replacing the data term with a set of border constraints at the new aspect ratio. By comparing our results to the highest performing retargeting method from the same comparison study (Figure 11), we can see that our method introduces less distortion in the foreground character, as we are able to "fold under" less important background areas. This exploits the findings of a recent survey that the loss of information can occasionally be preferable to the introduction of visual artifacts [RGSS10].

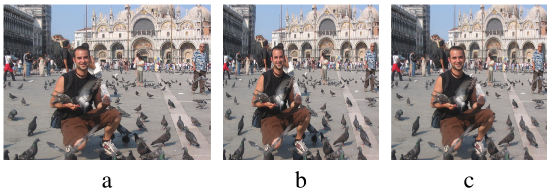

Figure 11: *Discontinuous warps for image retargeting. a) Original image, b) retargeting using Krähenbühl et al. [KLHG09], c) retargeting using our method*

Limitations Our method shares some limitations with image warping in general. Scenes that contain too much visually salient content do not allow modification without introducing visible artifacts. Fortunately, in stereo view synthesis, the amount of warping needed is on the order of tens of pixels, which is small enough such that these cases are rare. When such images do exist and our automatically detected edges are insufficient, our method defaults to requiring similar input as existing 2D to 3D conversion solutions (manual object segmentation and depth assignment). Additionally, transparent objects, where multiple depths exist per pixel, are not correctly modeled by our approach. The construction of warps that represent such multi-modal data is an interesting area of future research.

One main difference between our method and prior 2D to 3D conversion techniques is that we operate directly on image-domain disparities. While this allows for artistic depth effects and robust view synthesis, monocular depth cues such as perspective changes, texture gradient, and relative size cannot be reproduced by our method. However, in neighboring stereo views, incorrect changes in perspective are often too small to be detected, and in our analysis, did not cause apparent depth conflicts or unconvincing results.

5. Conclusions

We have presented a novel workflow for creating 3D content from 2D input. We describe 2D to 3D conversion through a brushing metaphor, presenting an alternative to existing conversion methods. This interaction paradigm is made possible by the introduction of a novel discontinuous warping method that handles all steps of 2D to 3D conversion, including hole filling. Using our interactive workflow, we were able to significantly reduce the amount of user time required to generate convincing stereoscopic pairs. Furthermore, we validated our method by comparing converted images with ground truth and by conducting a user study, both of which indicate that our method can create a convincing depth experience. Many interesting directions for future work remain. Our discontinuous warping method has many possible extensions, such as combining image-domain and projection-based methods. In addition, we plan on integrating our workflow in actual production pipelines which will give further insight into useful interfaces for user-assisted stereoscopic depth creation.

Acknowledgments

We would like to thank Pixar, HHI, and Flickr users; Mikebaird, Magnusfranklin, Gualtiero, Haojan, Denemiles, Theklan, Wiki_hybrid, TJ.Blackwell, Hikingartist, Malser and DeiselDemon for allowing the use of their images.

References

[AS07] Avidan S., Shamir A.: Seam carving for content-aware image resizing. *ACM Trans. Graph. 26*, 3 (2007), 10. 3

[CAA09] Carroll R., Agrawala M., Agarwala A.: Optimizing content-preserving projections for wide-angle images. *ACM Trans. Graph. 28*, 3 (2009). 2

[CW93] Chen S. E., Williams L.: View interpolation for image synthesis. In *SIGGRAPH* (1993), pp. 279–288. 2

[DRE*10] Didyk P., Ritschel T., Eisemann E., Myszkowski K., Seidel H.-P.: Adaptive image-space stereo view synthesis. In *VMV Workshop* (Siegen, Germany, 2010), pp. 299–306. 2

[GWCO09] Guttmann M., Wolf L., Cohen-Or D.: Semi-automatic stereo extraction from video footage. In *Computer Vision, 2009 IEEE 12th International Conference on* (Oct 2009), pp. 136 –142. 2

[GZMT10] Goferman S., Zelnik-Manor L., Tal A.: Context-aware saliency detection. In *CVPR* (2010), pp. 2376–2383. 4

[HW09] Harris J. M., Wilcox L. M.: The role of monocularly visible regions in depth and surface perception. *Vision Research 49*, 22 (2009), 2666 – 2685. Vision Research Special Issue - Vision Research Reviews. 5

[KKS07] Knorr S., Kunter M., Sikora T.: Super-resolution stereo- and multi-view synthesis from monocular video sequences. In *3DIM* (2007), pp. 55–64. 2

[KLHG09] Krähenbühl P., Lang M., Hornung A., Gross M. H.: A system for retargeting of streaming video. *ACM Trans. Graph. 28*, 5 (2009). 6, 7

[LHW*10] Lang M., Hornung A., Wang O., Poulakos S., Smolic A., Gross M. H.: Nonlinear disparity mapping for stereoscopic 3d. *ACM Trans. Graph. 29*, 4 (2010). 2, 3, 4

[LLW04] Levin A., Lischinski D., Weiss Y.: Colorization using optimization. *ACM Trans. Graph. 23*, 3 (2004), 689–694. 3

[RGSS10] Rubinstein M., Gutierrez D., Sorkine O., Shamir A.: A comparative study of image retargeting. *ACM Trans. Graph. 29*, 5 (2010), to appear. 5, 7

[SCSI08] Simakov D., Caspi Y., Shechtman E., Irani M.: Summarizing visual data using bidirectional similarity. In *CVPR* (2008). 3

[SS09] Shamir A., Sorkine O.: Visual media retargeting. In *SIGGRAPH ASIA Courses* (2009). 2

[SSJ*10] Sýkora D., Sedlacek D., Jinchao S., Dingliana J., Collins S.: Adding depth to cartoons using sparse depth (in)equalities. *Comput. Graph. Forum 29*, 2 (2010), 615–623. 2

[vdHDT*07] van den Hengel A., Dick A. R., Thormählen T., Ward B., Torr P. H. S.: Videotrace: rapid interactive scene modelling from video. *ACM Trans. Graph. 26*, 3 (2007), 86. 2

[WKB11] Ward B., Kang S. B., Bennett E. P.: Depth director: A system for adding depth to movies. *IEEE Computer Graphics and Applications 31* (2011), 36–48. 2

[WLKT09] Wei L.-Y., Lefebvre S., Kwatra V., Turk G.: State of the art in example-based texture synthesis. In *Eurographics 2009, State of the Art Report, EG-STAR* (2009), Eurographics Association. 3

[ZDPSS01] Zhang L., Dugas-Phocion G., Samson J.-S., Seitz S. M.: Single view modeling of free-form scenes. In *CVPR (1)* (2001), pp. 990–997. 2

[ZHQ*07] Zhang G., Hua W., Qin X., Wong T.-T., Bao H.: Stereoscopic video synthesis from a monocular video. *IEEE Trans. Vis. Comput. Graph. 13*, 4 (2007), 686–696. 2

[ZKU*04] Zitnick C. L., Kang S. B., Uyttendaele M., Winder S. A. J., Szeliski R.: High-quality video view interpolation using a layered representation. *ACM Trans. Graph. 23*, 3 (2004), 600–608. 2

[ZPvBG02] Zwicker M., Pfister H., van Baar J., Gross M. H.: Ewa splatting. *IEEE Trans. Vis. Comput. Graph. 8*, 3 (2002), 223–238. 6