3D CONVERSION USING VANISHING POINTS AND IMAGE WARPING

Daniel Donatsch, Nico Färber, Matthias Zwicker

Institute of Computer Science and Applied Mathematics, University of Bern

ABSTRACT

We describe a user assisted technique for 3D stereo conversion from 2D images. Our approach exploits the geometric structure of perspective images including vanishing points. We allow a user to indicate lines, planes, and vanishing points in the input image, and directly employ these as constraints in an image warping framework to produce a stereo pair. By sidestepping explicit construction of a depth map, our approach is applicable to more general scenes and avoids potential artifacts of depth-image-based rendering. Our method is most suitable for scenes with large scale structures such as buildings.

Index Terms — Stereo image processing

1. INTRODUCTION

The conversion of conventional 2D imagery to 3D stereo is one of the main strategies for 3D content production, besides capturing using stereo or multiview camera rigs. Even with modern software tools, however, the 2D to 3D stereo conversion process still requires significant effort by specifically trained users. Although there exist automatic algorithms, they fail to produce high-quality results consistently. In this paper, we describe a technique for user assisted 2D to 3D stereo conversion that exploits geometric structure of perspective images including vanishing points. We build on an image warping framework and exploit constraints derived from the perspective geometry of the input to obtain a stereo image pair. In our approach a user specifies line and plane constraints, and indicates lines that intersect at vanishing points. Instead of explicitly constructing a depth map, we warp the input image according to the user constraints to produce a stereo pair. Our approach is most suitable for scenes with large scale geometric structures such as buildings. It provides flexible user control and requires little user effort to produce visually convincing results.

2. RELATED WORK

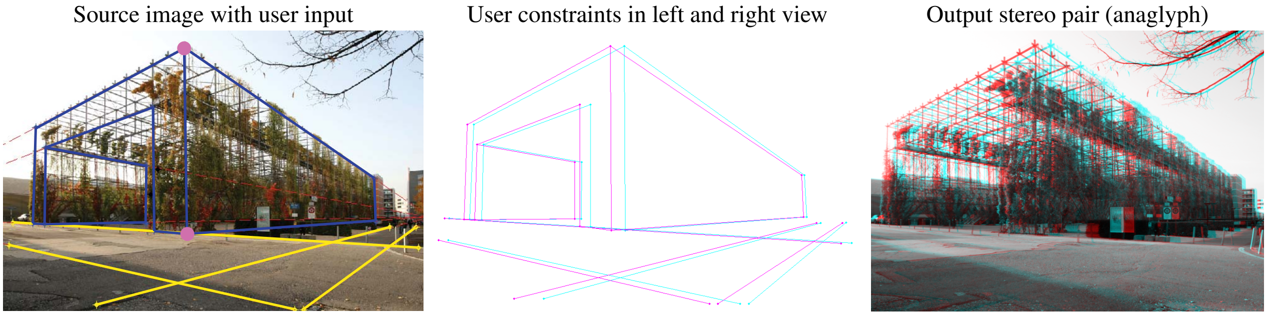
The 2D to stereo 3D conversion process [1] has gained significant attention recently because of the resurgence of stereo display in movie theaters and home TVs. The standard industry workflow for high-quality conversion involves labor intensive manual processing including segmentation (or rotoscoping) and depth map creation [2]. Our work is related to previous academic research that strives to reduce user effort, while still providing enough flexibility to obtain convincing results. We restrict our discussion to the most relevant previous work on user assisted stereo conversion. We refer to overview articles [3] for broader coverage. Harmann et al. [4] describe an early system that combines automatic depth map computation using a machine learning algorithm with user input. Several authors [5, 6, 7, 8] have proposed scribble-based interfaces that allow users to indicate the desired depth at sparse locations in video sequences. An automatic procedure then extrapolates the sparse user input to define dense per-pixel depth, and stereo views are created using depth-image-based rendering. Wang et al. [9] propose a similar scribble-based user interface, but they develop a discontinuous warping technique that can create sharp depth discontinuities at object boundaries. The "depth director" system by Ward et al. [10] is based on segmentation more similar to the standard industry workflow, but it includes a variety of computer vision techniques such as automatic oversegmentation, optical flow, and structure from motion, to support user interaction. A disadvantage of scribble-based systems is that they are less suitable to generate depth maps for large scale geometric structures such as buildings, since the consistency of user scribbles with the underlying geometry is not guaranteed.

Our approach exploits the geometric structure of perspective images including vanishing points, inspired by the seminal work by Horry et al. introducing the "tour into the picture" [11]. In contrast to this work, however, we do not explicitly construct 3D geometry, which allows us to work with more general scenes. Instead, we exploit line, plane, and vanishing points indicated by the user directly as constraints for image warping to produce a stereo pair. Our warping algorithm builds on the work by Carroll et al. [12], which we extend with constraints specifically for stereo conversion. Since we sidestep explicit depth image creation, we also avoid potential artifacts commonly associated with depth-image-based rendering.

3. OVERVIEW

We show an overview of our method in Figure 1. Given a source image, our main idea is to construct the left and right view of a stereo pair using constrained image warping. A user specifies various constraints such as straight line constraints (yellow in Figure 1), which preserve linearity in the warped image, and planar region constraints (blue), which locally restrict the image warp to a homography. The user can also select sets of lines that converge in vanishing points (dotted red lines). Finally, he can place target disparity constraints at individual locations in the source image (pink). In addition to the user inputs, our system automatically enforces additional constraints specific to stereo conversion: we restrict the image warp to generate horizontal disparities, we set the target disparity of vanishing points to zero (since they are at infinity), and we enforce that the disparity along line constraints varies linearly. We feed these constraints into an optimization-based image warping algorithm to map the input view, which is used as the left view, to the right view. Figure 1 shows the line and plane constraints provided by the user after warping in the middle, and the final stereo output on the right. We next provide details of the image warping algorithm, and then discuss results.

978-1-4799-1369-5/13/$31.00 2013 IEEE


Source image with user input

User constraints in left and right view

Output stereo pair (anaglyph)

Figure 1. Overview of our method: We show the source image with the user provided input consisting of line (yellow) and plane constraints (blue), vanishing points (intersections of dotted red lines), and disparity constraints (pink dots) on the left. The middle shows the line and plane constraints in the left and right view after warping (red-cyan anaglyph encoded), and the output stereo pair (anaglyph) is shown on the right.

4. CONSTRAINED IMAGE WARPING

Our constrained image warping algorithm is based on an energy minimization framework following the work by Carroll et al. [12]. We next describe our user interface, the mathematical formulation of the warping problem, and finally the energy minimization solver.

4.1. User interface

The user interface allows a user to define constraints that describe the geometric structure of the scene. The image warper then employs the constraints to obtain the novel views of the scenes required for stereo output. The user may indicate the following constraints:

Planar Regions. Regions indicated as planar will be warped locally using a homography.

Straight Lines. The user can specify straight lines, which will be preserved as straight during the warp. Further it is possible to mark subsegments of line constraints as inactive. This is useful if a constrained line is partially occluded by other objects.

Vanishing Points. The user can indicate lines and edges of planar regions that are parallel in 3D. These lines define a vanishing point in the image plane. Vanishing points are fixed during image warping, since they correspond to points at infinity in 3D, and the projection of points at infinity do not change under a translation of the camera parallel to the image plane.

Line Orientation. Lines and edges of planar regions can be restricted to become vertical or horizontal after the warp.

Disparities. The user can fix a desired output disparity at any point on the image. Often it is necessary to define the disparity at only two or a few more locations. We allow users to indicate relative disparities between the fixed locations, which provides the ability to scale the disparities easily later.

4.2. Mathematical Formulation

We define our warp using a rectangular mesh consisting of quad faces, which is overlaid on the input image. Given the warped locations of the four vertices of a quad, we warp the interior of the quad using bilinear interpolation. To compute the left and right view of the desired stereo output, we formulate an energy minimization problem that determines two deformed meshes $\mathbf{u}^l(\mathbf{x}_{i,j})$ and $\mathbf{u}^r(\mathbf{x}_{i,j})$ that best fulfill a set of constraints. Here, l and r denote the left and right view, respectively, i and j are vertex indices, $\mathbf{x}_{i,j}$ are locations of undeformed mesh vertices on the input image, and $\mathbf{u}^*(\mathbf{x}_{i,j})$, with $* \in \{l, r\}$, are the warped locations of the vertices in the left and right output views, respectively. We also denote input coordinates by $\mathbf{x} = (x, y)$ and output coordinates in the left and right view by $\mathbf{u}^* = (u^*, v^*), * \in \{l, r\}$. Next we briefly discuss the energy terms for our constraints. In addition to the user provided constraints introduced in Section 4.1, we impose further constraints to ensure the output is a valid stereo pair.

Avoiding Vertical Disparities. We avoid vertical disparities by penalizing differences between the v coordinates in the left and right output views, which leads to an energy term summing up over all mesh vertices,

$$E_a = \sum_{i,j} (v^l_{i,j} - v^r_{i,j})^2. \tag{1}$$

User Provided Disparities. Each user specified disparity constraint is given by a location $\mathbf{x}^d = (x^d, y^d)$ and a target relative disparity Δ, where d denotes the disparity constraint. Each disparity constraint corresponds to a target location $\mathbf{u}^{l,d} = (x^d + f\Delta, y^d)$ in the left, and $\mathbf{u}^{r,d} = (x^d - f\Delta, y^d)$ in the right view, where f is a user specified global disparity scaling factor. Hence the energy term for each disparity constraint is

$$E_d = \left|\mathbf{u}^l(\mathbf{x}^d) - \mathbf{u}^{l,d}\right|^2 + \left|\mathbf{u}^r(\mathbf{x}^d) - \mathbf{u}^{r,d}\right|^2. \tag{2}$$

We also introduce a disparity constraint for each vanishing point, where the target disparity simply is $\Delta = 0$, that is, vanishing points remain fixed.

Note that the constrained location $\mathbf{x}^d$ is unlikely to coincide with a grid vertex. Hence we express the location as a linear combination of its surrounding quad vertices, where we compute weights (a, b, c, d) according to Heckbert's inverse bilinear mapping [13], similar as proposed by Carroll et al. [12]. The corresponding output location is expressed using the same weights, that is, $\mathbf{u}^*(\mathbf{x}) = a\mathbf{u}^*_{i,j} + b\mathbf{u}^*_{i+1,j} + c\mathbf{u}^*_{i+1,j+1} + d\mathbf{u}^*_{i,j+1}$.

Ratios from Vanishing Points. While line constraints preserve straightness of lines, they do not penalize deformations along the line direction. We exploit the additional information provided by vanishing points to avoid such undesired deformations. Assume we have two points $\mathbf{x}_1 = (x_1, y_1), \mathbf{x}_2 = (x_2, y_2)$ on a line with vanishing point $\mathbf{x}_\infty = (x_\infty, y_\infty)$, as shown in Figure 2. Let us consider the ratios $|x_1 - x_\infty|/|x_2 - x_\infty| = c_x$, and $|y_1 - y_\infty|/|y_2 - y_\infty| = c_y$. Because after the warp the line given by $\mathbf{x}_1$ and $\mathbf{x}_2$ still goes through the same vanishing point $\mathbf{u}_\infty = \mathbf{x}_\infty$, and with the intercept theorem, we can see that the ratios c_x and c_y stay constant during the warp. In our case the vanishing point never lies between $\mathbf{x}_1$ and $\mathbf{x}_2$, hence we can omit

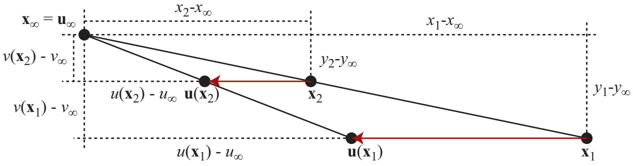

$\mathbf{x}_\infty = \mathbf{u}_\infty$
x_2-x_∞
x_1-x_∞
$v(\mathbf{x}_2)$ - v_∞
y_2-y_∞
y_1-y_∞
$u(\mathbf{x}_2)$ - u_∞
$\mathbf{u}(\mathbf{x}_2)$
$\mathbf{x}_2$
$v(\mathbf{x}_1)$ - v_∞
$u(\mathbf{x}_1)$ - u_∞
$\mathbf{u}(\mathbf{x}_1)$
$\mathbf{x}_1$

Figure 2. The ratios of points on a line from a vanishing point stay constant during the warp.

the norm. Reordering terms gives $x_1 - c_x x_2 = (1 - c_x)x_\infty$, and similarly for the y-coordinate. This leads to the energy

$$E_r = \sum_k \left(\frac{u(\mathbf{x}_k) - c_x u(\mathbf{x}_0)}{1 - c_x} - u_\infty \right)^2 + \left(\frac{v(\mathbf{x}_k) - c_y v(\mathbf{x}_0)}{1 - c_y} - v_\infty \right)^2, \quad (3)$$

where $\mathbf{x}_k, k \geq 0$ are locations sampled along the line. To sample the line regularly, we split it into intervals obtained by intersecting it with the mesh, and we use the middle point of each interval. We express these locations using bilinear interpolation from mesh vertices as above. Note that these constraints may seem redundant with the constraint to avoid vertical disparities. With lines that are nearly horizontal, however, even small vertical disparities can lead to significant undesired deformations. We found the ratio constraint to be necessary to avoid these in practice. Finally, observe that we divide the energy by $(1 - c_x)$ and $(1 - c_y)$. This scales the error to pixel units and makes it comparable to all the other energy terms, which measure the error in pixels, too.

Additional Constraints. We implemented the remaining user constraints described in Section 4.1 similarly as proposed in the work by Caroll et al. [12], which we refer to for a more detailed description. We use an energy term E_h to constrain the warp to a homography in planar regions. If two planar regions have a common edge, it is necessary to constrain the homographies, which gives rise to an additional term E_{hc}. The straight line constraint yields an energy term E_l. This energy may also constrain the line orientation, if desired. Next there is an energy E_v that keeps lines pointing to vanishing points. Finally, there are two regularization energies. The conformal energy E_c keeps the mesh rectangular, and a smoothness term E_s prevents abrupt changes from one mesh cell to the next.

4.3. Optimization

The total energy E our algorithm minimizes is a weighted sum of all the energies from the previous subsection,

$$\begin{aligned} E =& w_a^2 E_a + w_d^2 \sum E_d + w_r^2 \sum E_r \\ &+ w_h^2 \sum E_h + w_{hc}^2 \sum E_{hc} + w_l^2 \sum E_l \\ &+ w_v^2 \sum E_v + w_c^2 E_c + w_s^2 E_s, \end{aligned} \quad (4)$$

where the summations are over the number of the respective types of constraints. We also multiply each type of energy with a weight factor. The disparity constraints are the most important. Further, they act on only one grid cell, where all other constraints affect larger parts of the mesh. So we weight this energy highest. The other user constraints are more important than the regularization terms, since the later are not meant to be hard constraints. Hence we also weight them more heavily. Although we normalize the energy of the ratio constraint in (Equation 3), this energy is often about ten times larger than the others. We balance this by giving ten times less weight to it. Besides, we found the best weights by experimenting and with regard to [12]. We produced all results shown in this paper with weights $w_d = 1000$, $w_h = w_{hc} = 200$, $w_l = w_v = 100$, and $w_r = 10$ for the user given constraints, and $w_a = 20$, $w_s = 12$, and $w_c = l$ for the others.

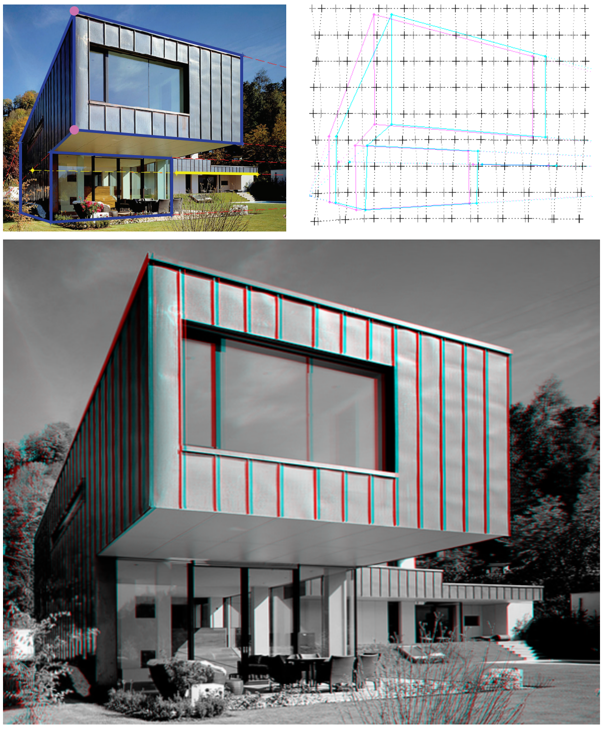


Figure 3. The top left image shows the input including the user given constraints. Next to it we show the geometry and a subsample of the warped meshes, and at the bottom the gray scale output encoded in anaglyph.

The total energy we minimize is a least-squares problem. Because of the homography and line constraints, however, it is nonlinear. We implemented a simple iterative Gauss-Newton method to solve for the minimum. We stop the minimization as soon as the total error becomes smaller than 10^{-5}. Keep in mind that we measure the energy in pixel units. In the vast majority of our experiments we reached the stopping condition after at most ten iteration steps.

5. RESULTS

We show all results in gray scale because they are more suitable for anaglyph glasses than color images. Images that need few constraints, as in Figure 3, take only a couple of minutes of user interaction. For more complex scenes indicating appropriate constraints may require trial and error, and our algorithm is fast enough to enable an iterative workflow. While it may be challenging for the user to set geometrically plausible disparities, our system allows a user to handle even complex scenes by specifying only a small number of disparity constraints as shown in Figure 4. In Figures 3 and 4 we also shift the produced images horizontally towards each other by $\Delta/2$ after warping. Hence we can adjust the zero disparity plane such that the scene appears partially in front and behind the screen.

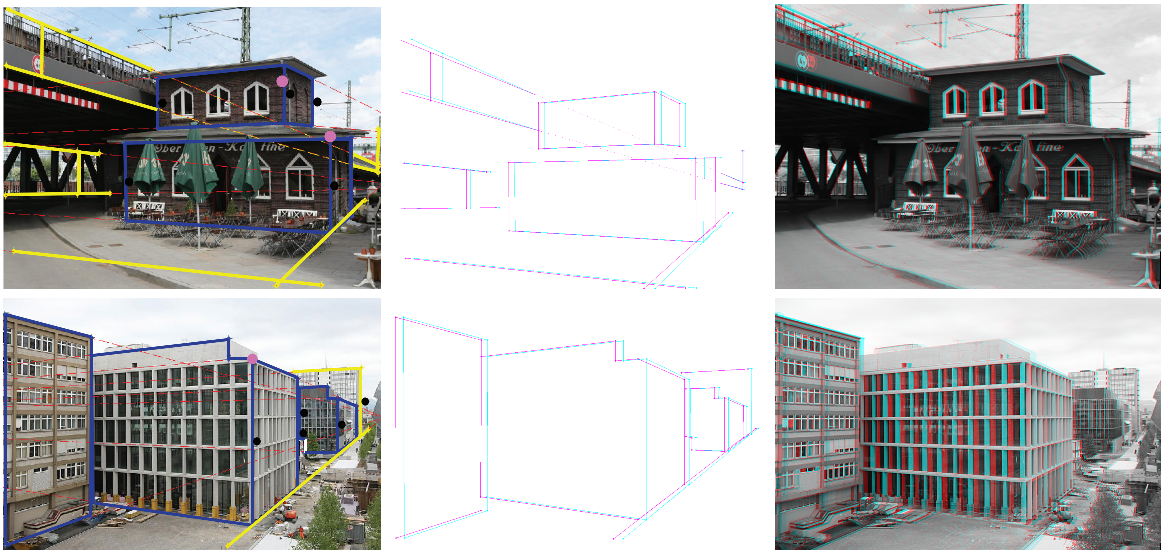


Figure 4. These images need many user constraints. In each we constrain five lines to be vertical (black dots). We prescribe disparities only at one (bottom) respectively two (top) points. On the bridge in the top image we use the ability to mark lines as partially hidden. This constraint guarantees that the disparity is correct along the whole bridge.

6. LIMITATIONS

Our algorithm is able to produce 3D images from a variety of single input images only with limited user input. In images showing many objects with round or organic shapes, however, it may be difficult to indicate the required constraints, because we rely on planes, straight lines and vanishing points. The downside of the cell-wise image warp is that it is not possible to create depth discontinuities. It is also not possible to have objects of different depths in one and the same cell. This can be seen in Figure 4, where the umbrellas are warped together with the building and therefore appear at the same depth. In the future, we plan to combine our approach with scribble and segmentation based techniques to handle such cases.

7. REFERENCES

[1] Wa James Tam and L. Zhang, “3d-tv content generation: 2d-to-3d conversion,” in *Multimedia and Expo, 2006 IEEE International Conference on*, 2006, pp. 1869–1872.

[2] A. P. Van Pernis and M. S. DeJohn, “Dimensionalization: converting 2D films to 3D,” in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Mar. 2008, vol. 6803 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*.

[3] A. Smolic, P. Kauff, S. Knorr, A. Hornung, M. Kunter, M. Muller, and M. Lang, “Three-dimensional video postproduction and processing,” *Proceedings of the IEEE*, vol. 99, no. 4, pp. 607–625, 2011.

[4] Philip V. Harman, Julien Flack, Simon Fox, and Mark Dowley, “Rapid 2d-to-3d conversion,” in *Proc. SPIE 4660, Stereoscopic Displays and Virtual Reality Systems IX*, 2002, pp. 78–86.

[5] M. Guttmann, L. Wolf, and D. Cohen-Or, “Semi-automatic stereo extraction from video footage,” in *Computer Vision, 2009 IEEE 12th International Conference on*, 2009, pp. 136–142.

[6] Xun Cao, Zheng Li, and Qionghai Dai, “Semi-automatic 2d-to-3d conversion using disparity propagation,” *Broadcasting, IEEE Transactions on*, vol. 57, no. 2, pp. 491–499, 2011.

[7] Xi Yan, You Yang, Guihua Er, and Qionghai Dai, “Depth map generation for 2d-to-3d conversion by limited user inputs and depth propagation,” in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*, 2011, pp. 1–4.

[8] Miao Liao, Jizhou Gao, Ruigang Yang, and Minglun Gong, “Video stereolization: Combining motion analysis with user interaction,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 7, pp. 1079–1088, 2012.

[9] O. Wang, M. Lang, M. Frei, A. Hornung, A. Smolic, and M. Gross, “Stereobrush: interactive 2d to 3d conversion using discontinuous warps,” in *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*, New York, NY, USA, 2011, SBIM ’11, pp. 47–54, ACM.

[10] B. Ward, Sing Bing Kang, and E.P. Bennett, “Depth director: A system for adding depth to movies,” *Computer Graphics and Applications, IEEE*, vol. 31, no. 1, pp. 36–48, 2011.

[11] Youichi Horry, Ken-Ichi Anjyo, and Kiyoshi Arai, “Tour into the picture: using a spidery mesh interface to make animation from a single image,” in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1997, SIGGRAPH ’97, pp. 225–232, ACM Press/Addison-Wesley Publishing Co.

[12] Robert Carroll, Aseem Agarwala, and Maneesh Agrawala, “Image warps for artistic perspective manipulation,” *ACM Transactions on Graphics*, vol. 29, no. 4, pp. 1, Jul. 2010.

[13] Paul S. Heckbert, *Fundamentals of Texture Mapping and Image Warping*, Master’s thesis, 1989.