Volume xx (200y), Number z, pp. 1–10

Gradient-Guided Local Disparity Editing

Leonardo Scandolo, Pablo Bauszat, and Elmar Eisemann †

Delft University of Technology

Figure 1: **Left:** Original stereographic image with disparity map inset **Middle:** An edited version, where we increased the roundness of the lion head and moved the background wall away from the viewer, presents depth-cue conflicts where the edited elements meet **Right:** Our optimization procedure preserves edits while removing inconsistencies.

Abstract

Stereoscopic 3D technology gives visual content creators a new dimension of design when creating images and movies. While useful for conveying emotion, laying emphasis on certain parts of the scene, or guiding the viewer's attention, editing stereo content is a challenging task. Not respecting comfort zones or adding incorrect depth cues, e.g. depth inversion, leads to a poor viewing experience. In this paper, we present a solution for editing stereoscopic content that allows an artist to impose disparity constraints and removes resulting depth conflicts using an optimization scheme. Using our approach, an artist only needs to focus on important high-level indications that are automatically made consistent with the entire scene while avoiding contradictory depth cues and respecting viewer comfort.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

1. Introduction

Stereoscopic images provide the viewer with a better understanding of the geometric space in a scene. Used artistically, it can convey emotions, emphasize objects or regions, and aid in expressing story elements. To achieve this, stereo ranges are increased or compressed and relative depths adapted [GNS11]. Nevertheless, conflicting or erroneous stereo content can result in an uncomfortable experience for viewers. In this regard, stereo editing is a delicate and often time-consuming procedure, performed by specialized artists and stereographers. Our solution supports these artists by allowing high-level definitions to set and modify stereo-related properties of parts of a scene. These indications are propagated automatically, while ensuring that the resulting stereo image pair remains plausible and can be viewed comfortably.

For a known display and observer configuration, the terms *depth* (distance to the camera), *pixel disparity* (shift of corresponding pixels in an image pair), and *vergence* (eye orientation) are linked [Men12]. For the sake of simplicity, we will use these terms interchangeably throughout this paper. Although disparity is typically a function of camera parameters and the object that is observed, stereographers manipulate depth content to influence disparity. While some artists work with 2D footage only [SKK*11], we will focus on 3D productions, where disparity values can be

† e-mail: {l.scandolo,p.bauszat,e.eisemann}@tudelft.nl

changed by interacting with the 3D scene, i.e., changing the depth extent and position of objects.

Modifying depth directly can result in depth cue conflicts and affect the observer's interpretation of the scene, which can cause visual discomfort. For example, in Fig. 1 the background wall was moved away from the viewer (by increasing its disparity), while the lion head was extended in depth. These edits result in conflicts with the rest of the scene: the lion head appears to extend beyond the wall and the wall seems detached from other scene elements.

Depth relationships between objects render depth manipulations complex. Manipulating one object can induce an entire chain of operations and quickly result in a trial-and-error process. Therefore, we propose to manage disparity edits as a global process, taking all parts of a scene into account to avoid unwanted results. Our approach aims at fulfilling the artist's indications while testing for depth-cue errors. As for many artistic tools, providing fast feedback is important. This goal is achieved via an efficient optimization procedure that derives suitable disparity values that are used to produce a new stereoscopic image pair.

Specifically, our contributions include:

- A formalization of stereo editing tools and conflicts;
- A real-time method to optimize scene disparity;
- A solution to avoid disocclusion or temporal artifacts.

2. Related Work

Over the last century, stereo vision and depth perception has received much attention from the clinical and physiological perspective. A detailed explanation of the mechanisms involved in human stereo vision can be found in [Ken01] and [How12]. More recently, work has been devoted to understanding discomfort and fatigue related to distortions present in stereo image displays. Lambooij et. al [LFHI09] and Meester et al. [MIS04] provide reviews that detail distortion effects in stereoscopic displays and their effect on viewer comfort. In particular, vergence and accommodation conflicts [BJ80, HGAB08, BWA*10] are a leading cause of visual fatigue, which can be reduced by keeping depth content to a depth comfort zone. Camera parameters can be automatically adapted for this purpose in virtual scenes [OHB*11] or even real-life stereoscopic camera systems [HGG*11]. Other methods to reduce discomfort rely on post-processing [KSI*02] of the final stereo pair, introducing blur [DHG*14], and depth of field effects [CR15].

Research towards perceptual stereo models can also help reduce or eliminate viewer discomfort [DRE*11, DRE*12b, DMHG13]. Such models can also be used to enhance depth effects, for example using the Cornsweet illusion [DRE*12a], adding film grain to a video [TDMS14] or efficiently compressing disparity information [PHM*14]. Templin et al. [TDMS14] and Mu et al. [MSMH15] modeled user response times for rapid disparity changes, such as video cuts, which allows artists to know when fast vergence changes will be acceptable for observers.

In the context of stereo content editing and post-process, rotoscoping [SKK*11, LCC12] is a widely used technique, where image elements are placed in layers at different depths. The depth of these layers can be moved and scaled, and commercial products [PFT, Mis, Ocu] are available to facilitate this process. Some of these tools can detect color inconsistencies between the stereo pair images and also possible violations to the stereo vision comfort zone, but the detection and correction of depth conflicts is left to the artist. Furthermore, Wang et al. [WLF*11] provide tools to insert depth information to a 2D image via scribble-based tools and the use of an image-aware dispersion method.

Other artistic stereo editing methods focus on globally modifying the available depth range, akin to global tone-mapping used in images. Wang. et al [WZL*16] and Kellnhofer et al. [KDM*16] propose different methods to modify disparity globally in order to enhance depth perception in certain areas of an image pair or stereoscopic video. Lang et al. [LHW*10] present a method to automatically create and apply a global disparity warping that affects the complete scene but does not allow for localized editing (see Fig. 4). Optimizing for depth perception during motion in depth [KRMS13] and parallax motion [KDR*16] have also been explored.

Nevertheless, most of the previous approaches do not allow for user-defined local edits, which are common in movie productions, or they do not ensure consistency after an edit has been made. Our work addresses this problem. We will rely on a global optimization strategy that shares similarity with gradient-guided optimizations that have been explored in different settings, e.g., editing and filtering [PGB03, BZCC10], video editing [FCOD*04], or image stitching [LZPW04]. Luo et al [LSC*12] propose an automated system for stereoscopic image stitching that can preserve borders and correct perspective projection. They do so via a gradient-preserving optimization process similar to Perez et al [PGB03], but unlike the work presented here, it targets images with no defined underlying mesh and only handles the use case of image-stitching.

3. Disparity Editing

The goal of our proposed method is to allow an artist to edit disparity values for a given view of a 3D scene without having to consider potential conflicts. In this context, we strive for real-time performance to be able to provide instant feedback.

To explain our solution, we will first describe how we will model the tools that influence the original scene disparity (Sec. 3.2). In practice, this process will be linked to a *disparity map*, which, for a given view, stores in each pixel a disparity value (Sec. 3.1). Our algorithm will derive an optimized disparity map, integrating the artist's constraints defined with the aforementioned tools, while avoiding depth conflicts (Sec. 3.3). To additionally prevent artifacts due to hidden geometry and temporal changes, we rely on a scene reprojection technique. It transfers the information from this disparity map to the 3D scene, which is then rendered to an image pair following the disparity map (Sec. 3.4). Fig. 2 showcases the different stages involved in our approach.

3.1. Disparity Map

The *disparity map* stores the final pixel disparity between the left and right view as an image taken from a camera located precisely

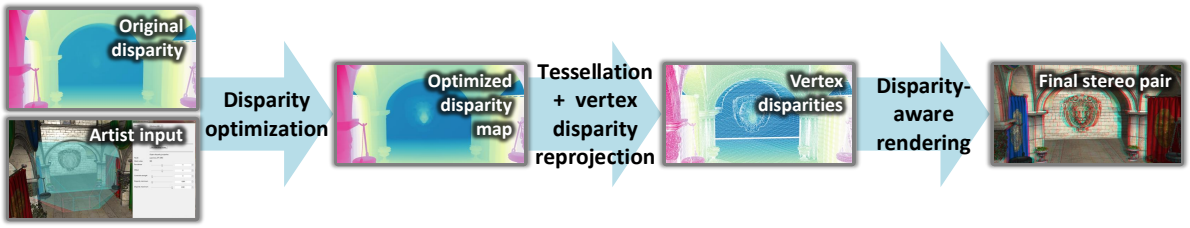


Figure 2: Our pipeline takes the initial disparity values and artist input to create an optimized disparity map, which is used by the vertices of the scene to allow a disparity-aware render algorithm to create a stereographic image pair.

between the left and right view. This map can be derived very efficiently by rendering the scene from the middle camera and converting the depth buffer by taking the focal plane distance and the interaxial distance of the stereoscopic cameras into account [Men12]. We refer to this unedited disparity map as $D : \mathbb{N}_0^2 \rightarrow \mathbb{R}$.

The tools we will provide to the user will influence this disparity map. As user commands might cause conflicts or inconsistencies, a depth conflict resolution strategy will override them where necessary before performing an optimization.

The tools and conflict resolution procedures will shape a target gradient $\bar{G}$ that will be linked to the optimized disparity map $\bar{D}$ via a set of linear equations:

$$\begin{aligned} \bar{D}_{x,y} &= \bar{D}_{x+1,y} - \bar{G}x_{x,y} \\ \bar{D}_{x,y} &= \bar{D}_{x-1,y} + \bar{G}x_{x-1,y} \\ \bar{D}_{x,y} &= \bar{D}_{x,y+1} - \bar{G}y_{x,y} \\ \bar{D}_{x,y} &= \bar{D}_{x,y-1} + \bar{G}y_{x,y-1} \end{aligned} \tag{1}$$

These equations will be part of a larger linear system that will be solved in the least-squares sense in order to obtain $\bar{D}$.

For brevity and readability, we will use subscripts to refer to the sampling of these maps, i.e., D_p instead of $D(p)$. Likewise, the first and second component of $\bar{G}$ will be noted as $\bar{G}x$ and $\bar{G}y$ respectively.

3.2. Disparity Tools

We will describe several editing tools, which are found in actual practice [SKK*11]. These tools act on properties of individual objects, properties relating pairs of objects or world-space points, and global parameters. We will express their effect directly in terms of constraints for the optimized disparity map or its target gradient.

Roundness Roundness refers to a change of an object's disparity range. Increasing roundness is commonly used to put emphasis on main objects or to convey emotion; in the movie *UP*, the roundness of the main character contrasted drastically with the roundness of a happy character when the latter approached his house to express the different emotional states.

Roundness R scales the disparity difference of every point of an object with respect to its center. This operation amounts to enlarging or decreasing the disparity gradient in a pixel p, if it belongs to the pixel set θ_{obj} corresponding to the manipulated object :

$$\forall p \in \theta_{obj} : \bar{G}_p = G_p \cdot R \tag{2}$$

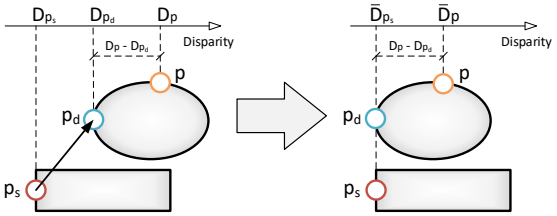


Figure 3: The matching points constraint matches the disparity of two points plus an offset (zero in this case) while maintaining the original disparity difference of all points in their influence set.

Disparity anchoring Disparity anchoring means that a certain disparity value is enforced for a chosen location. This option is important to specify the overall layout of a 3D environment [Men12]. Usually, the artist will enforce a specific depth for certain scene elements, e.g., the main object at screen distance to minimize the vergence-accommodation conflict. In our solution, the artist chooses an offset O_d to the initial disparity. As roundness will affect the disparity as well, we include it in the computation:

$$\forall p \in \theta_{obj} : \bar{D}_p = D_m + (D_p - D_m) \cdot R + O_d \tag{3}$$

where D_m is the object's center point disparity.

Interface preservation Interface preservation is used to maintain the local depth contrast between objects. It is known that local depth contrast can have a global effect [AHR78, DRE*12a]. Further, it helps separating objects clearly in space.

We allow users to specify pairs of objects for which the disparity difference should be maintained. Consequently, the pixels on the shared boundary maintain their disparity gradient $\left(\bar{G}_p = G_p\right)$. This definition can also be extended by allowing the user to draw a pixel mask to indicate where the disparity gradient should remain unaffected. This option is particularly useful for static imagery in the background.

Matching points Besides overlapping objects, a user can also couple the disparity of different elements in the scene. For example, in a view of a soccer ball flying through the air, one might want to keep the disparity between player and soccer ball constantly at the limit of the comfort zone to obtain the highest comfortable depth contrast. A more subtle application is for objects that are in contact. Fig. 1 shows an example, where the wall has been moved back. The attached objects become disconnected and appear to float in the air.

An artist can easily connect the objects to the wall using matching points.

Specifically, a user can mark a destination point p_d to match the disparity of a source point p_s plus an optional offset O_m. This constraint affects the disparity value of a set of screen-space points θ_{mp} defined as belonging to the object indicated by p_d, or, optionally, a specified area around p_d. For all points p within θ_{mp}, the constraint attempts to maintain the original disparity difference between p and p_d, but takes as pivot point $(\bar{D}_{p_s} + O_m)$ instead of $\bar{D}_{p_d}$ (Fig. 3):

$$\forall p \in \theta_{mp} : \bar{D}_p = (\bar{D}_{p_s} + O_m) + (D_p - D_{p_d}) \quad (4)$$

3.3. Disparity Map Optimization

The aim of the optimization stage is to solve the sparse linear system that arises from the constraints imposed by the tools described in Sec. 3.2 . Specifically, the optimization procedure will produce an optimized disparity map $\bar{D}$ which minimizes the sum of a per-pixel energy function E over all pixels:

$$\arg\min_{\bar{D}} \sum_p E_p(\bar{D}) \quad (5)$$

The energy function E is the sum of four terms which arise from the gradient constraints and the editing tools, and whose weights can be adjusted by the user:

$$E_p(\bar{D}) = c_1 E_p^g(\bar{D}) + c_2 E_p^a(\bar{D}) + c_3 E_p^m(\bar{D}) + \varepsilon E_p^r(\bar{D}). \quad (6)$$

The individual energy functions are the square residuals of the linear system formed by Eqs. 1, 3, 4, and a regularizing term:

- $\mathbf{E^g}$, the gradient energy term, is the sum of the square difference of the sides of Eqs. 1, defined for all pixels.
- $\mathbf{E^a}$, the disparity anchor term, is the square difference of Eq. 3. It is present for the pixels corresponding to objects for which an anchor disparity has been defined.
- $\mathbf{E^m}$, the matching points term, comes from the squared difference of the sides of Eq. 4 for each matching point defined. Each set of matching points has a different pixel influence set θ.
- $\mathbf{E^r_p} = |\bar{D}_p - D_p|^2$, the regularization term, ensures that there is a single solution in the absence of user defined constraints. It is defined for all pixels with a very low weight factor.

The gradient energy term ensures that the solution follows the target disparity gradient and that discontinuities or edges are correctly preserved. The target gradients are created using Eq. 2 for intra-object gradients. For inter-object gradients, we use the gradient of $\widetilde{D}$, which is the field we obtain by applying Eq. 3; this is the edited disparity map showcased in Figs. 1, 8, 9, 10, and 11. Finally, for areas where interface preservation is specified we revert to the original disparity map gradient.

Before the optimization procedure is performed, the linear system is inspected and modified to avoid depth inconsistencies which can potentially arise from using the tools. The most important inconsistencies are depth inversions, where for two overlapping objects, one should be behind another but their disparities imply the opposite. Such changes are reflected by differing signs of the gradient in the original and goal map gradients, which makes them easy to detect. In this case, the target gradient can be reset to the original gradient. Our framework can be expanded to deal with other conflicts in a similar fashion. For example, depth conflicts can arise at image borders for objects that are supposed to appear in front of the screen, as they are cut by the screen boundary. This case can be solved by adding an appropriate constraint to the system that penalizes pixel disparities larger than the pixel distance from the nearest vertical image border.

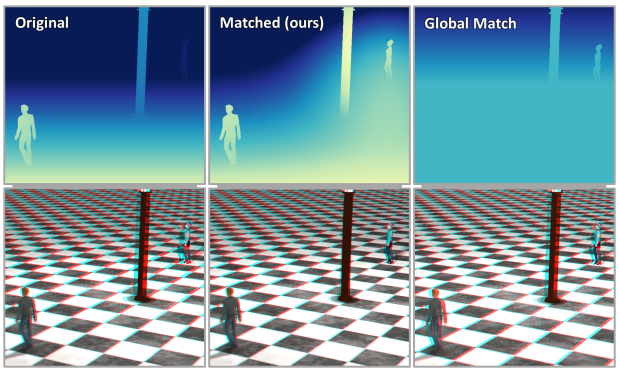


Figure 4: **Left:** Original disparity values and resulting stereoscopic pair for a scene with two characters. **Middle:** Result after using our method, matching the disparity of both characters. **Right:** Adjusting the disparity globally to match the character disparities, as in [LHW*10], results in loss of stereo contrast between the front character and the floor.

In general, the weight of each user-defined constraint is initialized to a default value of one and can be controlled by the user manually and intuitively since we provide instant feedback. However, some effects may only be required when viewing an object from a certain direction, or at a specified distance. Especially in image sequences, an artist may want a smooth transition between different sets of constraints when the camera or scene objects move. Our system provides the means to control the weight of a specific constraint based on different geometrical factors. A video showcasing this use case is included in the supplementary material.

Given that we follow a target gradient, the final optimization method is a modified Poisson reconstruction problem with added screening constraints. For large resolution images, directly solving the linear system is usually infeasible due to memory constraints, and thus iterative methods are preferable, such as Jacobi, SOR, or gradient descent methods [LH95]. For our implementation, we opted to use a GPU-based multi-resolution solver, since it maps well to GPU usage, avoids expensive GPU-CPU memory transfers, and is fast enough to provide real-time results. We create successively halved resolution versions of the full-resolution grid (via rendering or sampling), and solve each one with ten iterations of the Jacobi method. The initial solution for each grid level is obtained by upscaling the solution for the next coarser grid and the coarsest grid is initialized to $\widetilde{D}$. We create the initial disparity value for pixel p_f in the finer grid level f using the optimized disparity value of pixel p_c in coarse grid level c using the formula $\bar{D}^c_{p_c} + (\widetilde{D}^c_{p_c} - \widetilde{D}^f_{p_f})$, where the disparity map superscript denotes the grid level used. This formula uses the nearest pixel at the lower resolution grid, and adds the disparity difference in $\widetilde{D}$ to ensure that discontinuities are preserved between source and destination.

3.4. Stereo image creation

In principle, one can create a stereoscopic image pair by warping a middle-view image according to the optimized disparity map [DRE*10]. Unfortunately, such image-based procedures can lead to holes due to disocclusions that reveal content not visible from the middle view. In the case where the only available information is a single segmented image plus a depth or disparity estimation, this is the best possible solution.

A more interesting case arises when we have access to the complete scene information. In this situation, we can provide a more robust solution, that relies on assigning a disparity value to each mesh vertex based on the optimized disparity map. With a per-vertex optimized disparity value, we can perform a disparity-aware render of the scene to obtain a hole-free stereo image pair that matches the optimized disparity map. This method is similar to the one described in [KRMS13], but since we target real-time performance, several adaptations are needed. As we will detail below, we target a much lower tessellation level and employ a different heuristic for hidden vertices, as well as a bilateral filter pass in order to improve temporal stability. We begin by describing how to perform the stereo rendering step in order to give insight into some restrictions that will apply to the disparity reprojection step.

Disparity-aware rendering In the simplest case of a single triangle and a target disparity map, we want to render a stereo image pair that renders the triangle according to the map. We do this by sampling the disparity map at each projected vertex position. We then render the triangle from the middle-view camera once for each view, and add an offset to the viewport-space position in opposite horizontal directions for each view. This added offset corresponds to half of the disparity value assigned to the vertex being processed. Consequently, disparity values are respected precisely at the vertices, and are a linear interpolation of the vertex disparities at all other locations. Therefore, as described, this method cannot correctly follow non-linear disparity gradients inside the triangle that may be present in the target disparity map. In order to overcome this limitation, we can apply tessellation to the original triangle before projecting the disparity values to the vertices, resulting in a piecewise-linear approximation of the original disparity map.

This procedure can be applied to a complete 3D scene to create a stereo image pair that correctly handles disocclussions. We use the hardware tessellation capabilities of modern GPUs to avoid any modifications to the original mesh. However, the disparity reprojection step needs to carefully handle the cases of vertices that are occluded or fall outside the middle-view camera field of view. Fig. 5 shows the poor approximation of the target disparity for parts of a scene with low triangle count, such as large flat walls, and the improvement achieved when applying tessellation.

Disparity reprojection For visible vertices, we can directly assign a disparity value by sampling the disparity map. In order to determine vertex visibility, we compare its projected depth to the depth map created during the initial disparity map creation. Instead of relying only on the corresponding disparity map pixel to which each vertex projects, we sample a small neighborhood of pixels to increase robustness. This step also allows us to assign a disparity for vertices just outside the view frustum or close to the occlusion boundary. To integrate the result of the samples, we make use of a cross bilateral filter [ED04, PSA*04], using filter weights based on screen-space position, depth difference, normal orientation, and object id [TM98]. In this way, samples not related to the current vertex will be discarded automatically. Furthermore, filtering values avoids sudden disparity jumps and improves temporal coherency.

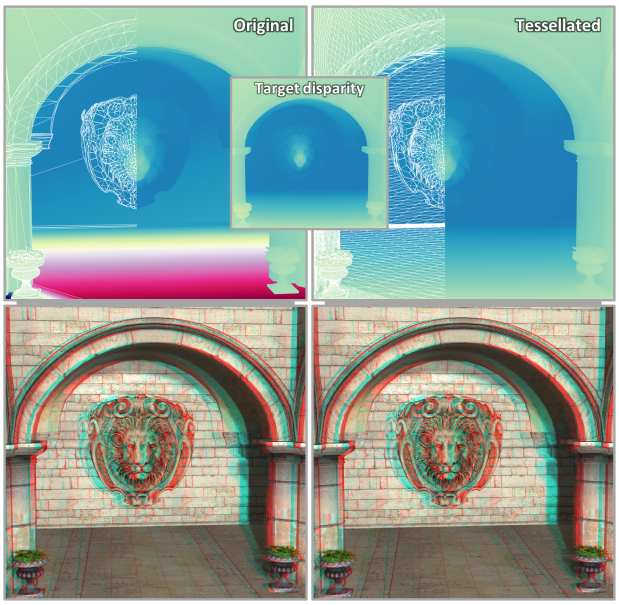


Figure 5: **Top inset:** Optimized screen-space disparity **Left:** Resulting disparity (top) and final stereoscopic image (bottom) when projecting screen-space disparity to original vertices fails to reproduce the target optimized disparity. **Right:** Disparity projected to tessellated geometry closely matches the optimized result.

If we cannot determine any valid sample for a vertex, we can still estimate its disparity by comparing D to $\bar{D}$ at this location and applying the difference to the original disparity of the vertex.

3.5. Implementation Details

We initially render the middle view via a deferred rendering pass, outputting several textures that contain properties used in the optimization pass, such as depth information, normals, object id and an initial disparity value. The optimization pass is then performed as a series of compute shader dispatches that create the target gradients, and is followed by a multi-resolution solver [BFGS03] that acts as outlined in Sec. 3.3, and outputs an optimized disparity texture. Using 16-bit floating point values provides sufficient precision for the optimization procedure and leads to real-time rates. Therefore, an artist can interactively edit the scene, and receive instant feedback.

The final rendering pass uses the optimized disparity map to determine vertex disparities during the tessellation evaluation stage, with the tessellation level determined by screen-space area. The disparity value is then added to the vertex viewport x coordinate in a geometry shader, which is invoked twice with multi-viewport support to efficiently generate a side-by-side image pair.

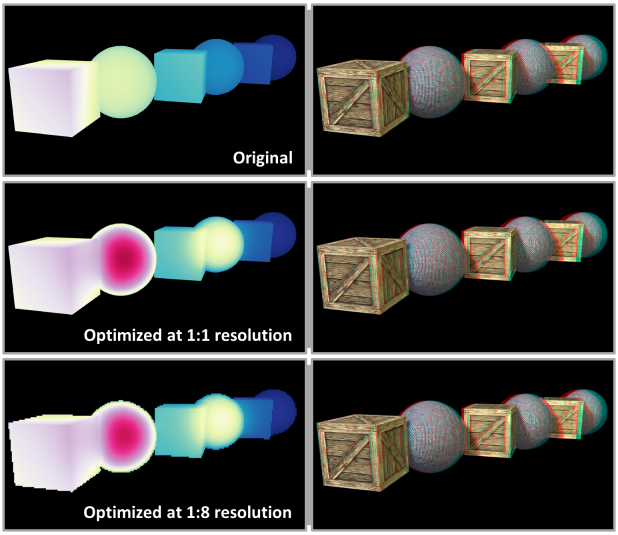
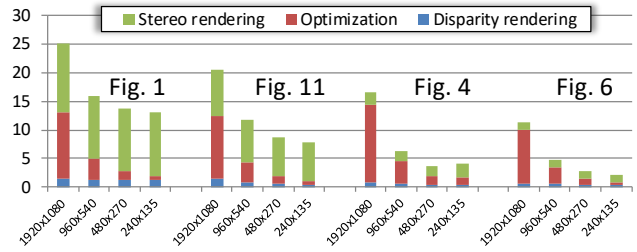
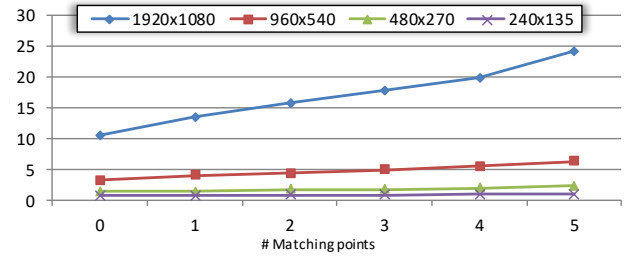


Figure 6: **Top:** Original disparity map and resulting image pair. **Middle:** Optimization result after increasing the roundness of the first two spheres. **Bottom:** Performing the optimization at 1:8 resolution yields very similar results.



(a) Creation times in ms for selected test scenes using different disparity map resolutions and full HD stereo image resolution.



(b) Timings in ms of the optimization stage for Fig. 4 using different amounts of matching points and disparity-map resolutions.

Figure 7: Timings for our optimization procedure

Optimizing Texture Resolution The use of cross bilateral filtering to obtain a per-vertex disparity lifts the strict correspondence in terms of resolution for the optimized disparity map and the final image. In our experiments, the optimization resolution can be much lower than the final stereo image resolution without a noticeable difference in quality. Thus, we can target very large stereo image resolutions, while maintaining low memory usage and real-time performance. Fig. 6 illustrates the result and stereo images using a 1:1 and 1:8 scale between optimized disparity and final image pair. Moreover, this performance gain can be invested into placing the middle view differently and increasing its field-of-view projection to encompass both views to well handle the screen borders.

Optimizing Convergence In most optimization techniques, and ours in particular, a good initial estimate of the solution results in a faster solver convergence. During the course of a typical animation, the resulting disparity maps will be similar from one frame to the next, which implies that a previous frame is a good estimate of the next frame disparity. Using our reprojection technique, we can create a view for the current frame using the previous disparity values and use it as an initial solution for the optimization procedure.

4. Results

We implemented our method into a tool where a user can easily edit depth content in a scene by accessing the stereoscopic properties described in Sec. 3. We tested our method for different scenes and with varied artistic purposes to show the range of stereographic modifications that can be easily performed. In the following, we will detail some examples. Figs. 1, 6, and 11 showcase scenes were elements are highlighted by increasing their roundness or offsetting them in depth, making them more prominent while still harmonizing with the rest of the scene elements. To illustrate the interaction on an example, in the fairy scene, the user simply clicked on both trees in the background and increased their roundness by a factor of around five. The arising conflicts, that are very visible in the image that directly integrates the indications, were fully removed automatically by our algorithm, while maintaining the overall consistency of the scene. In Fig. 4 we show an example of matching the disparity of two characters which are at different depths. This is a useful application in practice since it means an observer does not need to adjust their vergence when switching their gaze from one character to the other. Such quick vergence shifts are known to cause discomfort, and stereographers usually employ various methodologies to avoid them [TDM*14] or in some cases may need to redesign a scene [Men12]. In this case, a user created a matching point constraint between the two characters. As is visible in the result, local contrasts are maintained. This property gives the illusion of maintaining the original scene arrangement, while the disparity of the two characters is actually matched despite them being in different 3D locations in the scene.

Edits are sometimes useful to evoke emotions. We show an example of a disparity manipulation meant to increase the feeling of scale in Fig. 10, by making the cliffs seem more prominent and dangerous. In this case, the user selected parts of the mountain and increased their roundness, while anchoring them at a preferred depth. Additionally, as illustrated in the supplementary video, the constraints can be dampened depending on the view of the camera. During the course of animation, the weight of the user's constraints were linked to the camera location, which made them vanish, when the camera rotated away from the cliffs. Another example to convey emotion is shown in Fig. 8, where depth edits were used to convey a feeling of loneliness in the scene by flattening the character and pushing him away from the viewer. Hereby, a feeling of distance is created. The editing operation moved the person backwards, which created a conflict with the couch, but also the bunny

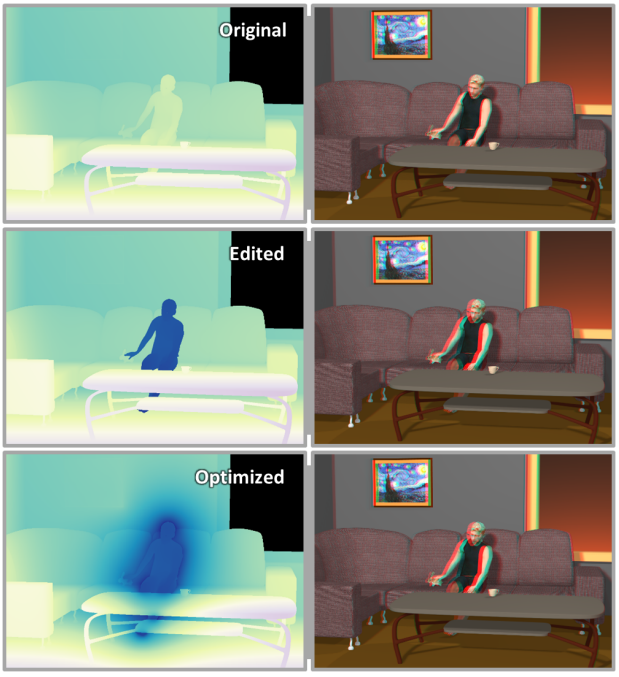


Figure 8: Stereoscopic manipulation can be used to showcase negative feelings as well. In this scene we convey a feeling of loneliness by flattening the sitting man and pushing him back in depth.

in his hand. The optimization process adjusts the disparity to correct for these mistakes. As shown, this solution is robust and also handles smaller objects, such as the bunny. Finally, we also believe our method is useful beyond artistic purposes. In Fig. 9, we show a visualization of a human jaw, where we want to enhance the shape of the three lower left molars. Such a solution is useful in an educational context to focus attention to important elements. The user only manipulated the roundness to increase the shape perception.

All edits in these scenes required less than a few seconds of interaction. By default object interfaces are maintained, which causes the optimization process to spread the deviation induced by the constraints over all objects. In general the user input can be very sparse, which supports our goal of simplifying interaction and having the artist focus only on important indications. Additional examples and animations are presented in the accompanying material.

4.1. Memory usage

Memory consumption is almost entirely linked to the textures used for the optimization. It includes the deferred buffers containing the scene properties, and a series of mipmapped textures used for the multigrid optimization procedure. At full HD resolution, around 160 MB are used, which is directly linked to the disparity map resolution, i.e. at half that resolution, the memory usage is four times smaller.

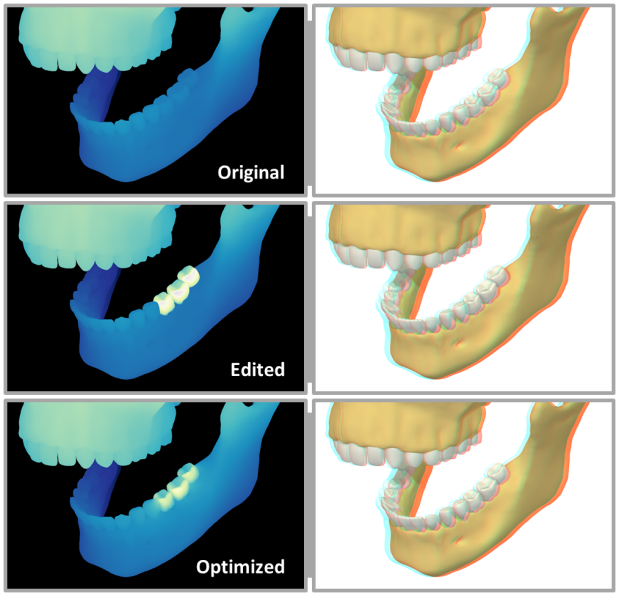


Figure 9: Our method can be used to aid in visualization applications. Here, the three molars are made rounder to improve the understanding of their shape.

4.2. Timing

Our pipeline is implemented using C++ and OpenGL and all tests were run on an Intel i7-5820K CPU running Windows 7, with 32GB of main system memory and an NVidia Titan X GPU.

Our method employs three stages: the disparity map and scene property extraction, the optimization, and the stereo pair rendering. Three parameters affect the efficiency of these stages: the disparity map resolution R_o, the final stereo image pair resolution R_s and the geometric complexity of the scene G. The first stage is only dependent on R_o and G, the second stage depends solely on R_o and the final stage is only affected by R_s and G. The optimization stage can also be affected by the amount of matching point constraints set by the artist as they render the linear system less sparse and requires additional texture lookups. Out of these parameters, R_o is the one an artist has most control over, and can be selected to obtain a desired time/quality balance.

Fig. 7a showcases the timing of the three stages in some of our test scenes for different resolutions of the optimized disparity map. The optimization procedure is most heavily affected by different matching points and the scene from Fig. 4 shows an increased time spent on the optimization procedure. Fig. 7b shows the effect of different amount of matching point constraints on the optimization timing.

4.3. User studies

Stereo perception study We performed a small-scale user study to evaluate the effectiveness of the stereographic images created through our algorithm. For this, we presented the different versions of the stereo output of Fig. 10 to a sample of seven participants, before first verifying that they were able to perceive stereoscopic con-

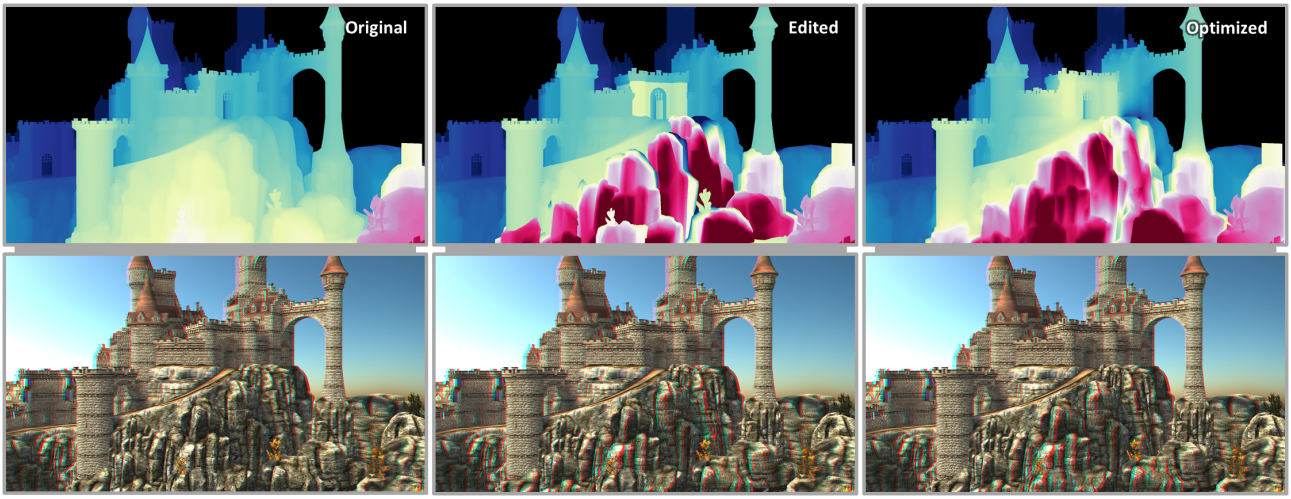


Figure 10: A more menacing look for the rock cliffs can be achieved by making them more prominent in depth. This is a hard case for manual editing, since many small features such as trees and bushes need to be made consistent with the edited parts of the mountain.

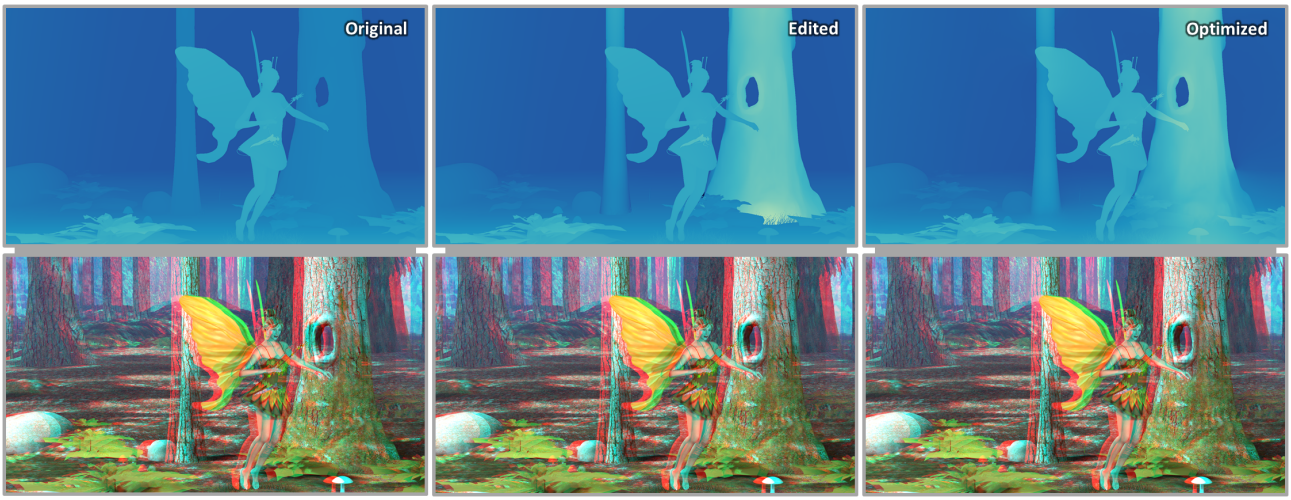


Figure 11: In the fairy forest test scene, the trees are made more prominent by increasing their roundness and slightly offsetting their disparity towards the viewer. The unoptimized version shows depth conflicts where the trees meet the ground and the fairy character, which are fixed in the optimized version.

tent. Fig. 12 showcases the view frustum and some of the parameters used for this task. All participants had normal or corrected-to-normal vision and no knowledge in the field of stereoscopic content creation or editing.

In the first part of the trial, the participants were shown the original stereoscopic image and our optimized version, and they were freely able to switch between both versions with no time constraints. They were asked to explain the difference between both images and recorded whether they were correctly able to identify the expected edition effect, namely that the cliffs look rounder and more prominent. All participants noticed that the difference between both images were constrained to the cliff area, and five out of the seven (71.5%) described the effect stating that the cliffs were better defined and there was better depth perception in their area.

In the second part, we allowed the participants to observe both our optimized version and the unoptimized edited version, and again they could switch between both images. No time constraint was imposed and they were asked to choose their preferred image. In this case, 100% of the participants expressed a preference for the optimized version, alleging either discomfort or visible artifacts when looking at the unoptimized version.

Usage study In order to test our solution against a traditional 3D modeling approach, we tasked an expert 3D modeler to create a similar modification as was created with our method in Fig. 13, namely to enhance the disparity of two rock models in the scene. The task was performed in Autodesk Maya and the artist reported that around 45 minutes of work were required. The results are shown at the bottom of Fig. 13. The same edition was done in under one minute with our framework. The artist mentioned difficulties to correctly maintain the geometric interfaces between the

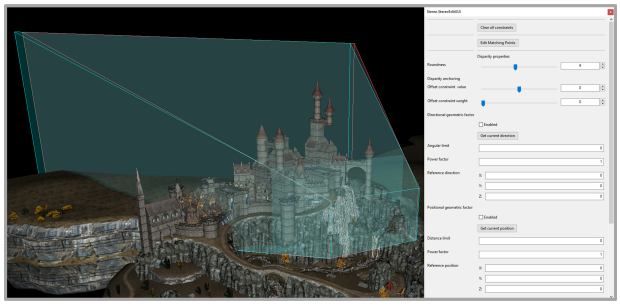


Figure 12: A screenshot of the editing interface used for our experiments, showing the camera frustum and some of the parameters used for our perceptual user study.

mesh parts intended to be enhanced and the rest of the scene. Additionally, when asked if the effect could be enhanced, he reported that he would have to basically start over. He also underlined that he considers the task as very challenging. The final image shows a large depth enhancement for the target regions, but as expected, the geometrical shape of the area has been significantly altered. Furthermore, some objects are missing, such as one of the trees. This highlights a key feature in our proposed solution: the ability to largely decouple disparity edition from geometrical shape, thus altering only depth perception while maintaining the geometrical shape of the scene.

4.4. Limitations and Future Work

We obtain good results for realistic use cases. When introducing highly contradictory constraints, our optimization technique might create artifacts in image sequences. In such cases, adjusting the constraint weights can achieve good results. A limitation exists for thin geometry, and large disparity changes. A large disparity gradient can result in a stretched version of the object to fulfill the indicated disparity constraints. We do not explicitly tackle the problem of temporal stability for image sequences, but as seen in the videos provided with the supplementary material, the produced disparity values do not show stability problems, as the optimization procedure has a well-defined behavior and provides a smooth fit to the artist's constraints. If the constraint changes are smooth, the result is typically smooth. We could envision reprojecting the optimized disparity map between consecutive frames using optical flow and rely on equalizing constraints with a small weight to avoid large changes but found it unnecessary in practice.

5. Conclusion

We have presented a method for editing stereoscopic content in 3D scenes by modifying high-level properties of the scene elements. Our approach then identifies regions where depth conflicts may arise from the user input and creates and performs an optimization procedure to obtain a conflict-free disparity map. Although image-based, coupling the map to our reprojection leads to a hole-free stereoscopic image pair. The solution runs fully on the GPU, which leads to instant feedback even for very large image resolutions. Our approach is an important addition to the toolbox of stereographers

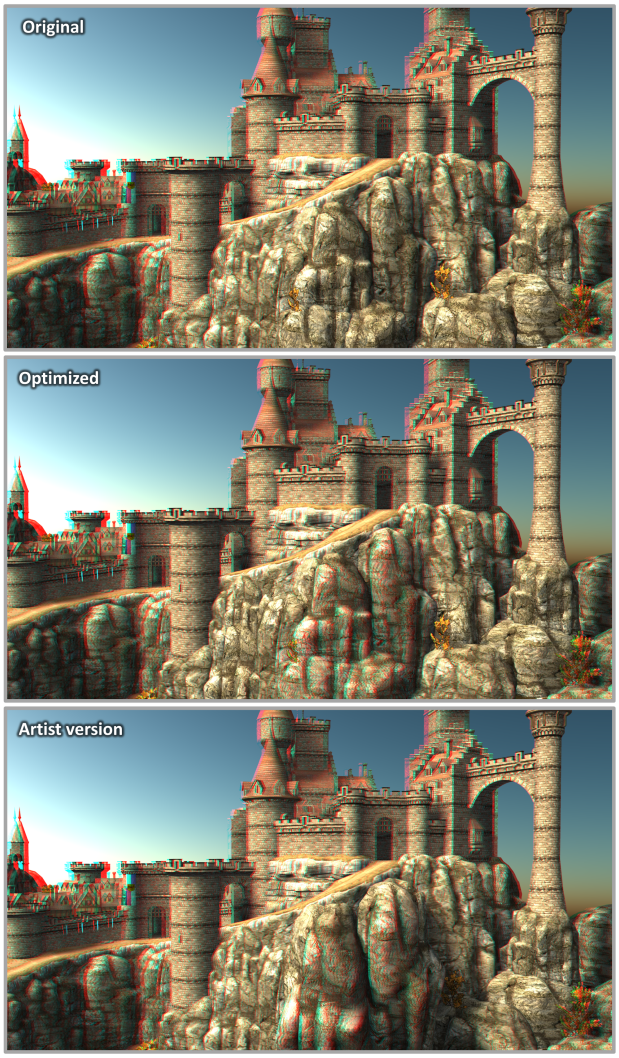


Figure 13: A comparison of our optimized version with an artist edited version where the geometry has been modified. Directly modifying geometry is time consuming and alters the original look of the scene, while our solution is much faster to create and preserves the geometric shape, only altering stereo perception.

that simplifies dealing with the various conflicts. It allows the artist to focus on semantics instead of the technical underpinnings and delivers convincing results even when used by novice users.

6. Acknowledgments

We would like to thank Dr. Timothy Balint for his help proofreading this article and Dr. Leila Schemali for insights regarding stereoscopic content editing. The fairy scene is provided by the University of Utah, Epic Citadel by Epic Games, and Sponza by Marko Dabrovic. The remaining scenes were courtesy of blendswap/archive3D users ChameleonScales, hilux, TiZeta, and Rahman Jr. This work is partly supported by VIDI NextView, funded by NWO Vernieuwingsimpuls

References

[AHR78] ANSTIS S. M., HOWARD I. P., ROGERS B.: A Craik-O'Brien-Cornsweet illusion for visual depth. *Vision Research 18*, 2 (1978). 3

[BFGS03] BOLZ J., FARMER I., GRINSPUN E., SCHRÖODER P.: Sparse matrix solvers on the gpu: conjugate gradients and multigrid. *ACM Trans. Graph. 22*, 3 (2003), 917–924. 5

[BJ80] BURT P., JULESZ B.: A disparity gradient limit for binocular fusion. *Science 208*, 4444 (1980), 615–617. 2

[BWA*10] BLUM T., WIECZOREK M., AICHERT A., TIBREWAL R., NAVAB N.: The effect of out-of-focus blur on visual discomfort when using stereo displays. In *ISMAR* (2010), IEEE, pp. 13–17. 2

[BZCC10] BHAT P., ZITNICK C. L., COHEN M., CURLESS B.: Gradientshop: A gradient-domain optimization framework for image and video filtering. *ACM Trans. Graph. 29*, 2 (2010), 10. 2

[CR15] CARNEGIE K., RHEE T.: Reducing visual discomfort with hmds using dynamic depth of field. *IEEE Comput. Graph. Appl. 35*, 5 (2015), 34–41. 2

[DHG*14] DUCHOWSKI A. T., HOUSE D. H., GESTRING J., WANG R. I., KREJTZ K., KREJTZ I., MANTIUK R., BAZYLUK B.: Reducing visual discomfort of 3d stereoscopic displays with gaze-contingent depth-of-field. In *ACM SAP* (2014), pp. 39–46. 2

[DMHG13] DU S., MASIA B., HU S., GUTIERREZ D.: A metric of visual comfort for stereoscopic motion. *ACM TOG 32*, 6 (2013), 222. 2

[DRE*10] DIDYK P., RITSCHEL T., EISEMANN E., MYSZKOWSKI K., SEIDEL H.-P.: Adaptive image-space stereo view synthesis. In *VMV* (2010), pp. 299–306. 5

[DRE*11] DIDYK P., RITSCHEL T., EISEMANN E., MYSZKOWSKI K., SEIDEL H.-P.: A perceptual model for disparity. In *ACM Trans. Graph.* (2011), vol. 30, p. 96. 2

[DRE*12a] DIDYK P., RITSCHEL T., EISEMANN E., MYSZKOWSKI K., SEIDEL H.-P.: Apparent stereo: The cornsweet illusion can enhance perceived depth. In *IS&T/SPIE Electronic Imaging* (2012). 2, 3

[DRE*12b] DIDYK P., RITSCHEL T., EISEMANN E., MYSZKOWSKI K., SEIDEL H.-P., MATUSIK W.: A luminance-contrast-aware disparity model and applications. *ACM Trans. Graph. 31*, 6 (2012), 184. 2

[ED04] EISEMANN E., DURAND F.: Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph. 23*, 3 (2004), 673–678. 5

[FCOD*04] FLEISHMAN S., COHEN-OR D., DRORI I., LEYVAND T., YESHURUN H.: Video operations in the gradient domain. *Tel-Aviv Univ, Tech. Rep* (2004). 2

[GNS11] GATEAU S., NEUMAN R., SALVATI M.: Siggraph 2011 stereoscopy course, 2011. 1

[HGAB08] HOFFMAN D. M., GIRSHICK A. R., AKELEY K., BANKS M. S.: Vergence–accommodation conflicts hinder visual performance and cause visual fatigue. *Journal of vision 8*, 3 (2008), 33–33. 2

[HGG*11] HEINZLE S., GREISEN P., GALLUP D., CHEN C., SANER D., SMOLIC A., BURG A., MATUSIK W., GROSS M.: Computational stereo camera system with programmable control loop. In *ACM Trans. Graph.* (2011), vol. 30, p. 94. 2

[How12] HOWARD I. P.: *Perceiving in depth, volume 1: basic mechanisms*. Oxford University Press, 2012. 2

[KDM*16] KELLNHOFER P., DIDYK P., MYSZKOWSKI K., HEFEEDA M. M., SEIDEL H.-P., MATUSIK W.: Gazestereo3d: seamless disparity manipulations. *ACM Trans. Graph. 35*, 4 (2016), 68. 2

[KDR*16] KELLNHOFER P., DIDYK P., RITSCHEL T., MASIA B., MYSZKOWSKI K., SEIDEL H.-P.: Motion parallax in stereo 3d: model and applications. *ACM Trans. Graph. 35*, 6 (2016), 176. 2

[Ken01] KENT D. M.: Foundations of binocular vision: A clinical perspective., 2001. 2

[KRMS13] KELLNHOFER P., RITSCHEL T., MYSZKOWSKI K., SEIDEL H.-P.: Optimizing disparity for motion in depth. In *CGF* (2013), vol. 32, pp. 143–152. 2, 5

[KSI*02] KAWAI T., SHIBATA T., INOUE T., SAKAGUCHI Y., OKABE K., KUNO Y.: Development of software for editing stereoscopic 3-d movies. In *Proc. of SPIE* (2002), vol. 4660, pp. 58–65. 2

[LCC12] LEE K.-Y., CHUNG C.-D., CHUANG Y.-Y.: Scene warping: Layer-based stereoscopic image resizing. In *CVPR* (2012), IEEE, pp. 49–56. 2

[LFHI09] LAMBOOIJ M., FORTUIN M., HEYNDERICKX I., IJSSELSTEIJN W.: Visual discomfort and visual fatigue of stereoscopic displays: A review. *JIST 53*, 3 (2009), 30201–1. 2

[LH95] LAWSON C. L., HANSON R. J.: *Solving least squares problems*, vol. 15. Siam, 1995. 4

[LHW*10] LANG M., HORNUNG A., WANG O., POULAKOS S., SMOLIC A., GROSS M.: Nonlinear disparity mapping for stereoscopic 3d. *ACM Trans. Graph. 29*, 4 (2010), 75. 2, 4

[LSC*12] LUO S.-J., SHEN I., CHEN B.-Y., CHENG W.-H., CHUANG Y.-Y., ET AL.: Perspective-aware warping for seamless stereoscopic image cloning. *ACM Trans. Graph. 31*, 6 (2012), 182. 2

[LZPW04] LEVIN A., ZOMET A., PELEG S., WEISS Y.: Seamless image stitching in the gradient domain. *Computer Vision-ECCV 2004* (2004), 377–389. 2

[Men12] MENDIBURU B.: *3D movie making: stereoscopic digital cinema from script to screen*. CRC Press, 2012. 1, 3, 6

[Mis] Mistika. URL: http://www.sgo.es. 2

[MIS04] MEESTERS L. M., IJSSELSTEIJN W. A., SEUNTIËNS P. J.: A survey of perceptual evaluations and requirements of three-dimensional tv. *IEEE Trans. Circuits Syst. 14*, 3 (2004), 381–391. 2

[MSMH15] MU T.-J., SUN J.-J., MARTIN R. R., HU S.-M.: A response time model for abrupt changes in binocular disparity. *The Visual Computer 31*, 5 (2015), 675–687. 2

[Ocu] Ocula. URL: https://www.foundry.com/products/ocula. 2

[OHB*11] OSKAM T., HORNUNG A., BOWLES H., MITCHELL K., GROSS M. H.: Oscam-optimized stereoscopic camera control for interactive 3d. *ACM Trans. Graph. 30*, 6 (2011), 189. 2

[PFT] Pftrack. URL: http://www.thepixelfarm.co.uk/pftrack/. 2

[PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. *ACM Trans. Graph. 22*, 3 (2003), 313–318. 2

[PHM*14] PAJĄK D., HERZOG R., MANTIUK R., DIDYK P., EISEMANN E., MYSZKOWSKI K., PULLI K.: Perceptual depth compression for stereo applications. In *CGF* (2014), vol. 33, pp. 195–204. 2

[PSA*04] PETSCHNIGG G., SZELISKI R., AGRAWALA M., COHEN M., HOPPE H., TOYAMA K.: Digital photography with flash and no-flash image pairs. *ACM Trans. Graph. 23*, 3 (2004), 664–672. 5

[SKK*11] SMOLIC A., KAUFF P., KNORR S., HORNUNG A., KUNTER M., MULLER M., LANG M.: Three-dimensional video postproduction and processing. *Proc. of the IEEE 99*, 4 (2011), 607–625. 1, 2, 3

[TDM*14] TEMPLIN K., DIDYK P., MYSZKOWSKI K., HEFEEDA M. M., SEIDEL H.-P., MATUSIK W.: Modeling and optimizing eye vergence response to stereoscopic cuts. *ACM Trans. Graph. 33*, 4 (2014), 145. 6

[TDMS14] TEMPLIN K., DIDYK P., MYSZKOWSKI K., SEIDEL H.-P.: Perceptually-motivated stereoscopic film grain. In *CGF* (2014), vol. 33, pp. 349–358. 2

[TM98] TOMASI C., MANDUCHI R.: Bilateral filtering for gray and color images. In *Proc. of ICCV* (1998), pp. 839–846. 5

[WLF*11] WANG O., LANG M., FREI M., HORNUNG A., SMOLIC A., GROSS M.: Stereobrush: interactive 2d to 3d conversion using discontinuous warps. In *Proc. SBIM* (2011), ACM, pp. 47–54. 2

[WZL*16] WANG M., ZHANG X.-J., LIANG J.-B., ZHANG S.-H., MARTIN R. R.: Comfort-driven disparity adjustment for stereoscopic video. *Computational Visual Media 2*, 1 (2016), 3–17. 2