# *Multi-Perspective stereoscopy from light fields*

**Massachusetts Institute of Technology**

# Multi-Perspective Stereoscopy from Light Fields

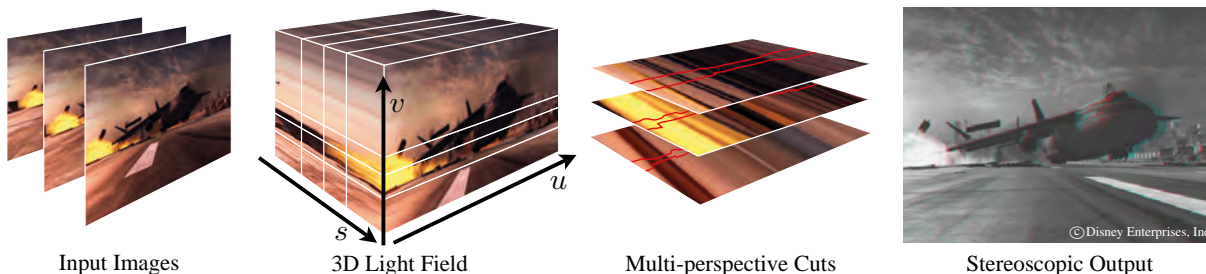Changil Kim[1,2]  Alexander Hornung[2]  Simon Heinzle[2]  Wojciech Matusik[2,3]  Markus Gross[1,2]

[1]ETH Zurich  [2]Disney Research Zurich  [3]MIT CSAIL

**Figure 1:** *We propose a framework for flexible stereoscopic disparity manipulation and content post-production. Our method computes multi-perspective stereoscopic output images from a 3D light field that satisfy arbitrary prescribed disparity constraints. We achieve this by computing piecewise continuous cuts (shown in red) through the light field that enable per-pixel disparity control. In this particular example we employed gradient domain processing to emphasize the depth of the airplane while suppressing disparities in the rest of the scene.*

## Abstract

This paper addresses stereoscopic view generation from a light field. We present a framework that allows for the generation of stereoscopic image pairs with per-pixel control over disparity, based on multi-perspective imaging from light fields. The proposed framework is novel and useful for stereoscopic image processing and post-production. The stereoscopic images are computed as piecewise continuous cuts through a light field, minimizing an energy reflecting prescribed parameters such as depth budget, maximum disparity gradient, desired stereoscopic baseline, and so on. As demonstrated in our results, this technique can be used for efficient and flexible stereoscopic post-processing, such as reducing excessive disparity while preserving perceived depth, or retargeting of already captured scenes to various view settings. Moreover, we generalize our method to multiple cuts, which is highly useful for content creation in the context of multi-view autostereoscopic displays. We present several results on computer-generated content as well as live-action content.

**Keywords:** stereoscopy, light field, multi-perspective imaging, autostereoscopic display, post-production

**Links:** ◆DL 🗋PDF

## 1 Introduction

Three-dimensional stereoscopic television, movies, and games have been gaining more and more popularity both within the entertainment industry and among consumers. An ever increasing amount of content is being created, distribution channels including live-broadcast are being developed, and stereoscopic monitors and TV sets are being sold in all major electronic stores. With novel genera-tions of autostereoscopic and multi-view autostereoscopic displays even glasses-free solutions become available to the consumer.

However, the task of creating convincing yet perceptually pleasing stereoscopic content remains difficult. This is mainly because post-processing tools for stereo are still underdeveloped, and one often has to resort to traditional monoscopic tools and workflows, which are generally ill-suited for stereo-specific issues [Mendiburu 2009]. This situation creates an opportunity to rethink the whole post-processing pipeline for stereoscopic content creation and editing. In the past the computer graphics community has greatly contributed to the development of novel tools for image and video processing. One particular example in the context of this work is the recent progress on light field capture and processing, which enables *post-acquisition* content modification such as depth-of-field, focus, or viewpoint changes. A variety of prototypes for light field acquisition have been developed [Adelson and Wang 1992; Yang et al. 2002; Ng et al. 2005; Wilburn et al. 2005; Georgiev et al. 2006; Veeraraghavan et al. 2007] such that we can expect plenoptic cameras to become available in the near future. However, the concept of post-acquisition control and editing is missing in stereoscopic post-processing.

The main cue responsible for stereoscopic scene perception is binocular parallax (or binocular disparity) and therefore tools for its manipulation are extremely important. One of the most common methods for controlling the amount of binocular parallax is based on setting the baseline, or the inter-axial distance, of two cameras prior to acquisition. However, the range of admissible baselines is quite limited since most scenes exhibit more disparity than humans can tolerate when viewing the content on a stereoscopic display. Reducing baseline decreases the amount of binocular disparity; but it also causes scene elements to be overly flat. The second, more sophisticated approach to disparity control requires remapping image disparities (or remapping the depth of scene elements), and then re-synthesizing new images. This approach has considerable disadvantages as well; for content captured with stereoscopic camera rigs, it typically requires accurate disparity computation and hole filling of scene elements that become visible in the re-synthesized views. For computer-generated images, changing the depth of the underlying scene elements is generally not an option, because changing the 3D geometry compromises the scene composition, lighting calculations, visual effects, etc [Neuman 2010].

In this paper we propose a novel concept for stereoscopic post-production to resolve these issues. The main contribution is a

framework for creating stereoscopic images, with accurate and flexible control over the resulting image disparities. Our framework is based on the concept of 3D light fields, assembled from a dense set of perspective images. While each perspective image corresponds to a planar cut through a light field, our approach defines each stereoscopic image pair as general cuts through this data structure, i.e., each image is assembled from potentially many perspective images. We show how such multi-perspective cuts can be employed to compute stereoscopic output images that satisfy an arbitrary set of goal disparities. These goal disparities can be defined either automatically by a disparity remapping operator or manually by the user for artistic control and effects. The actual multi-perspective cuts are computed on a light field, using energy minimization based on graph-cut optimization to compute each multi-perspective output image. In our results we present a number of different operators including global linear and nonlinear operators, but also local operators based on nonlinear disparity gradient compression.

This basic framework also allows for a number of practically relevant extensions. For example, we show how our method can be used to drive multi-view autostereoscopic (automultiscopic) displays more flexibly and efficiently by computing multiple cuts through a light field. Moreover, for computer-generated content we show that a full rendering of the input light field is not necessary, since the cut computation is performed on a 3D disparity volume corresponding to the light field. Hence, only those light rays have to be rendered, which are actually required to form the stereoscopic output images.

In summary, our proposed concept and formulation provides a novel, general framework that leverages the power and flexibility of light fields for stereoscopic content processing and optimization.

## 2 Related Work

Our work draws from research on multi-perspective images and disparity remapping for stereoscopic images.

**Multi-perspective images:** In the history of art multi-perspective imaging has been used by painters and artists as a fundamental stylistic tool. Similar methods have later been employed by animators in movie production, e.g., for drawing backgrounds for 2D cell animation [Thomas and Johnston 1995]. The computer graphics and computer vision community further studied the geometry and applications of multi-perspective imaging (a good overview is presented by Yu et al.[2010]). Wood et al. [1997] describe a first computer-assisted method to compute multi-perspective panoramas from a collection of perspective images. In the recent years many other types of multi-perspective cameras and corresponding images have been introduced: pushbroom cameras [Hartley and Gupta 1997] and related multiple-center-of-projection images [Rademacher and Bishop 1998], cross slit cameras [Pajdla 2002; Zomet et al. 2003], or general linear cameras [Yu and McMillan 2004]. In our work we do not assume any particular camera model. Instead the (multiple) perspectives of our images are optimized subject to prescribed stereoscopic disparity constraints.

The two most related publications to ours are the works by Seitz [2001] and Peleg et al. [2001]. Seitz [2001] analyzes the space of all possible image types (including multi-perspective images) that provide depth cues due to binocular parallax. His work provides a theoretical basis for our discussion of stereoscopic constraints and light field parameterization in the following section. Peleg et al. [2001] provide a framework to construct multi-perspective omnidirectional stereoscopic images. They show how to dynamically adapt the baseline to modify scene parallax by a local selection scheme for image columns. Our work is inspired by these ideas and

extends them to a more general, global optimization of the output views with respect to arbitrary, per-pixel disparity constraints. We provide further discussion in Section 4.4.4.

**Disparity remapping:** The motivation and need to remap disparities in order to optimize stereoscopic content for display on different output devices or according to user preferences has been shown in perceptual research [Woods et al. 1993; Held and Banks 2008; Didyk et al. 2011]. A number of technical approaches for disparity remapping applied to standard stereoscopic content have been proposed. Jones et al. [2001] analyze the scene depth range and adjust the stereoscopic camera baseline to a given disparity budget. Feldman et al. [2003] present a system that uses nonlinear depth-scaling for transmitting a 3D scene to be rendered from multiple views. Holliman [2004] describes a system that compresses the scene depth for stereoscopic displays by identifying a region of interest and compressing it differently compared to the rest of the scene. Koppal et al. [2011] provide a detailed discussion on optimal stereo and also provide basic post-processing tools. The main focus of their work is, however, shot planning during capture. Ward et al. [2011] proposed a system for 2D-to-3D conversion. The conversion relies on image warping and requires manual interaction. Kim et al. [2008] discuss how to perform non-linear depth remapping for multi-view autostereoscopic displays. Zwicker et al. [2006] present a remapping and a prefiltering framework for automultiscopic displays that adapts an input light field to the display capabilities.

All these works, however, are restricted in the type of disparity remapping operators they support. In particular they do not provide a solution for detailed control of disparity in real world images. The only method allowing for more complex, nonlinear and local disparity remapping has been presented by Lang et al. [2010]. But since their method uses smooth 2D warping of a stereoscopic image pair based on sparse image correspondences, it is likely to bend salient scene structures such as straight lines. Furthermore, other visually relevant cues such as disocclusions cannot be handled by this method. It is therefore restricted with respect to the amount of remapping that can be achieved without producing noticeable visual distortions and it does not allow for per-pixel control over disparities. Moreover, it cannot easily generalize to more than two input views. Our approach inherently benefits from a richer scene information, and is fundamentally different from the aforementioned methods: it selects actual light rays from an input light field in order to achieve per-pixel disparity control, instead of using image deformations or inpainting.

## 3 Image Generation from a Light Field

In this paper we are interested in generating image pairs for stereoscopic viewing, with accurate control over the corresponding space of binocular disparities, such as range or gradients. More specifically, the images we want to generate should satisfy the stereo constraint [Seitz 2001], i.e., they should feature *horizontal parallax* only, without any vertical displacement of scene points between the images. Seitz showed that, in order to satisfy this stereo constraint, the images have to be constructed from a very specific three-parameter family of light rays. This observation is important to the design of our algorithm; instead of having to process full 4D or higher-dimensional light fields [Adelson and Bergen 1991; Levoy and Hanrahan 1996; Gortler et al. 1996], we can focus our discussion on image generation from a 3D light field without loss of generality. In practice, typical examples of setups for 3D light field acquisition are a camera mounted to a linear stage, a linear camera array, or corresponding renderings of a virtual scene (see Figure 2a). For now we assume that the light field has been created with such a setup.

**(a)** *Scene space*  **(b)** *Ray space*
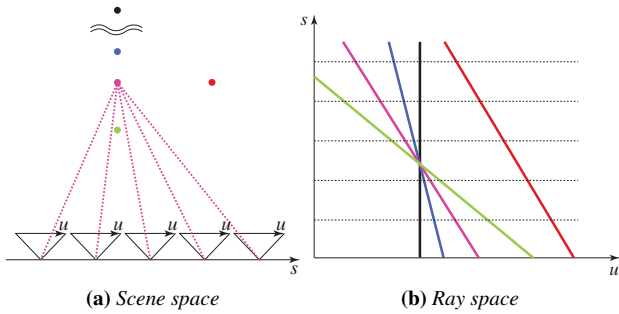
**Figure 2:** *Light field parameterization. (a) A 2D illustration of a scene and the corresponding imaging setup to generate a light field. (b) The corresponding 2D light field or epipolar-plane image (EPI). Each point in ray space corresponds to a ray in the light field. Scene points seen in multiple images become EPI lines in ray space (see Figure 1 or 3). The slope of each line is proportional to the distance of the corresponding scene point. For points at infinity (black point) the line becomes vertical.*

Let $\mathcal{L} : \mathbb{R}^3 \to \mathbb{R}^3$ be a 3D light field, created from a set of standard perspective RGB images. Each light ray $\mathcal{L}(u, v, s)$ is parameterized by three parameters; parameter $s$ denotes the 1D positional degree of freedom of the ray origin, whereas parameters $(u, v)$ represent the ray direction. Assuming uniform sampling of the ray space with respect to these parameters, Figure 2b illustrates a 2D light field corresponding to Figure 2a. Figure 1 shows an example of an actual 3D light field in the form of an *EPI volume* [Gortler et al. 1996], which can be intuitively interpreted as a stack consisting of the 2D input images. Since the capture process naturally results in a discrete set of rays, the parameters $u$, $v$, and $s$ will from now on be implicitly treated as integers. Therefore, $s$ can be regarded as an index to one of the input images, while $(u, v)$ indexes a pixel in image $I_s$, i.e., $\mathcal{L}(u, v, s) = I_s(u, v)$. For simplicity, our discussion will be based on this discretized view of the ray space; (re-)sampling has been addressed in previous works [Chai et al. 2000].

A 2D view that is not necessarily perspective can be generated from a 3D light field $\mathcal{L}$ by selecting a 2D subset of rays. As a simple example, a planar $u$-$v$-slice or 2D *cut* at a particular parameter position $s$ extracts the original standard perspective input image $I_s$ (see Figure 3a). Cuts with varying parameter $s$ yield images with varying centers of projection. For instance, a $v$-$s$-cut with constant parameter $u$ results in a so called pushbroom panorama, which corresponds to a sensor with a single pixel column and a linearly varying position of the camera center [Yu et al. 2010]. A $u$-$s$-cut represents a single EPI, i.e., a 2D stack of the same scanline across all images, also illustrated in Figure 3. However, there is no restriction to planar cuts. In principle, any 2D subset of rays can be used to generate an image, although a certain ray coherence is required in order to produce "meaningful" images. In the context of stereoscopic image generation, curved, piecewise continuous cuts result in multi-perspective views of a scene, as shown in Figure 3b. As shown by Seitz [2001] and Peleg et al. [2001], multi-perspective images can be fused stereoscopically, as long as they feature horizontal parallax only. This observation is the justification for our algorithm that allows the generation of multi-perspective stereoscopic image pairs with controlled disparity by computing corresponding cuts through a light field.

In order to convert a light field cut into an actual image, one has to sample the rays lying on the cut surface. This requires a parameterization of the possibly discontinuous cut which, in general, is a highly difficult problem (related to the field of surface parameterization). However, this problem is further complicated in the context of multiple simultaneous cuts for stereoscopic image generation,
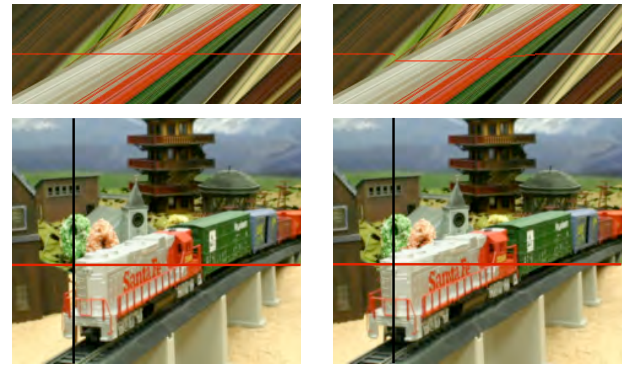


**(a)** *Planar cut and the resulting single-perspective image*  **(b)** *Nonplanar cut and the resulting multi-perspective image*

**Figure 3:** *Illustration of a planar, single perspective cut and a non-planar, multi-perspective cut through a EPI volume. The red line in the bottom images indicates the scanline of the EPI. For easier comparison the black line highlights the same column in both images. Note how the images are identical except for the train front.*



**Figure 4:** *A 2D EPI of a light field, showing two planar $u$-$v$-cuts. The horizontal offset $c$ changes the convergence plane of the stereoscopic image pair. The bottom image shows the corresponding stereoscopic image pair generated from $I_s$ and $I_{s'}$.*

since we have to take additional constraints into account. Assume, for example, a straight and a curved cut (as in Figure 3) represent a stereoscopic image pair. When sampling the rays along both cuts, any difference in the step size along the $u$-axis between the two cuts will have an impact on the horizontal parallax between corresponding scene points in the two images and, of course, also result in different image widths. Similarly, a differing parameterization and sampling along the $v$-axis will result in vertical parallax, which is undesirable for any stereoscopic image pair. A simple parameterization and sampling strategy, which naturally avoids these issues and does not introduce additional distortion in the output view, is a regular sampling of the cut surface along the $u$- and $v$-axis.

The following algorithm combines these basic ideas to compute multiple synchronized cuts through a light field in order to produce multi-perspective images with specific stereoscopic properties.

## 4 Stereoscopic Light Field Cuts

In order to introduce the terms and definitions used for our algorithm, we will first consider the generation of a standard perspective stereoscopic image pair. As discussed in the previous section, one can extract a perspective view from a light field $\mathcal{L}$ by fixing parameter $s$ and sampling the rays on the corresponding $u$-$v$-plane, effectively selecting the input image $I_s$. As illustrated in Figure 2a, different parameters $s$ represent input images captured at different,

**(a)** *Normalized disparity* $\mathcal{D}$      **(b)** *True disparity* $\mathcal{T}_s$
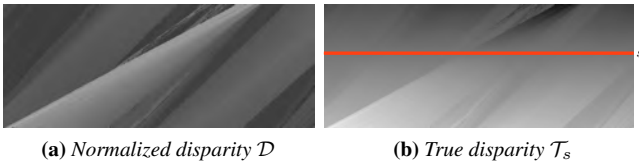
**Figure 5:** *(a) A 2D $u$-$s$-slice of the normalized disparity volume $\mathcal{D}$. (b) A 2D $u$-$s$-slice of the true image disparity volume $\mathcal{T}_s$ with respect to a reference view $I_s$.*

linearly translated camera positions. Correspondingly, the difference $\Delta(s', s) = s' - s$ is proportional to the camera baseline between two images $I_{s'}$ and $I_s$, i.e., $b = \Delta(s', s)/\gamma$, where $\gamma$ depends on the distance between the cameras, the field-of-view, and the horizontal image resolution. Hence, a stereoscopic image pair with baseline $b$ can be generated by picking a reference view $I_s$, and selecting the second view at $s' = s + \gamma b$, corresponding to two parallel $u$-$v$-cuts through $\mathcal{L}$. The convergence $c$ for such a stereoscopic image pair can be modified by shifting $I_{s'}$ horizontally with respect to $I_s$ (see Figure 4).

In order to create a stereoscopic image pair from a 3D light field $\mathcal{L}$ with constraints on the space of disparities, we define a corresponding 3D disparity volume $\mathcal{D} : \mathbb{R}^3 \to \mathbb{R}^+$ that stores the scaled reciprocal of the distance from the origin of each ray in $\mathcal{L}$ to its corresponding scene point (Figure 5a). $\mathcal{D}$ can be interpreted as a *normalized disparity*, such that the image disparity of a pixel $\mathbf{p}$ in $I_{s'}$ to a reference image $I_s$ is defined as
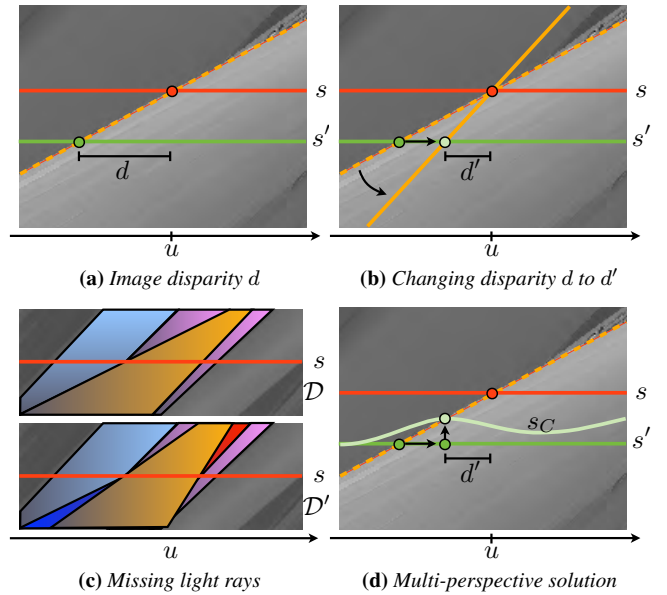
$$\mathcal{T}_s(\mathbf{p}, s') = \Delta(s', s)\mathcal{D}(\mathbf{p}, s'), \tag{1}$$

where we use $\mathbf{p}$ as shorthand notation for the coordinate pair $(u, v)$. Using this equation, $\mathcal{D}$ can be created from standard depth or disparity maps; we provide details on the computation in our results in Section 5. We call $\mathcal{T}_s$ the *true* disparity volume for a particular view $I_s$, as illustrated in Figure 5b.

Given a reference view $I_s$ and the true disparities $\mathcal{T}_s$ it is straightforward to formulate a simple procedure that finds a second view $I_{s'}$ such that $\mathcal{T}_s(*, *, s')$ does not exceed a certain disparity range. However, the only means for controlling disparity is the distance $\Delta(s', s)$ between the planar cuts. In the following we will describe how to compute nonplanar, multi-perspective views that satisfy more general, content-dependent disparity constraints.

## 4.1 Goal-based Multi-perspective Cuts

Consider Figure 6a, showing a normalized disparity volume $\mathcal{D}$ and planar cuts for two images $I_s$ and $I_{s'}$. According to Eq. (1) the horizontal parallax or image space disparity $d$ of a pixel in $I_s$ to the corresponding pixel in $I_{s'}$ can be computed as $\mathcal{T}_s(\mathbf{p}', s') = \Delta(s, s')\mathcal{D}(\mathbf{p}, s)$. Now assume we want to create a modified stereoscopic image pair that features a different depth impression only for the particular scene point seen at $I_s(\mathbf{p})$. As argued in the previous section, changing $\Delta(s, s')$ globally does not allow for such a local changes. An alternative solution is to keep $s$ and $s'$ fixed, and update the actual scene depth $\mathcal{D}(\mathbf{p}, s)$ instead by deforming the actual geometry of the scene. The problem with this approach is that modifying the depth of a scene implies changes to the complete underlying light field, since changing the depth of a scene point influences the slope of the corresponding line in ray space (see Figure 2 and Figure 6b). An example for the consequences is illustrated in Figure 6c: reducing the disparity of the frontmost, orange region results in missing light rays in regions further in the back of the scene (depicted in red and blue). The corresponding rays have not been captured in the original light field. Completing those regions would require complex resampling and hole-filling operations on the light field.



**(a)** *Image disparity $d$*      **(b)** *Changing disparity $d$ to $d'$*

**(c)** *Missing light rays*      **(d)** *Multi-perspective solution*

**Figure 6:** *Multi-perspective light field cuts for changing stereoscopic disparity. (a) Given two images $I_s$ and $I_{s'}$ with image disparity $d$ at pixel $u$. (b) Modification of the disparity $d$ to $d'$ effectively amounts to changing the scene depth (see also Figure 2), and, hence, the slope of the corresponding lines in the EPI volume. (c) Changing depth, in this example of the orange region, results in different (dis-)occlusion patterns, with missing information in the light field (red and blue region). (d) We propose to compute a cut $s_C$ instead, whose corresponding multi-perspective image $I_{s_C}$ effectively results in the same change of disparity from $d$ to $d'$.*

Instead of modifying the image distance $\Delta(s, s')$ or the scene depth $\mathcal{D}$, our algorithm computes a nonplanar cut $s_C : \mathbb{R}^2 \to \mathbb{R}$ through the light field, which maps rays $\mathbf{p}$ to parameters $s$ in order to meet a given set of goal disparity constraints. This idea is illustrated in Figure 6d: given the reference image $I_s$, a second view satisfying the disparity constraint for pixel $\mathbf{p}$ can be generated from a cut $s_C$ that intersects the EPI line corresponding to $I_s(\mathbf{p})$ at parameter position $u + d'$. Intuitively, the cut $s_C$ picks for each pixel $I_s(\mathbf{p})$ a pixel from *some* input image, such that the desired disparity constraints are fulfilled. As each input image shows a different perspective of the scene, the cut produces a multi-perspective output image $I_C$ that, together with the reference view $I_s$, forms a stereoscopic image pair where we effectively control the camera baseline for each pixel individually.

We define the set of goal disparities as a 2D map $\mathcal{G} : \mathbb{R}^2 \to \mathbb{R}$ that, for each pixel of the *output* view $I_C$, defines the desired disparity with respect to the reference view $I_s$ as follows. Assume that the disparity of pixel $u$ in $I_s$ to the multi-perspective image $I_C$ should be $d'$, as shown in Figure 6d. This implies that the value of the goal disparity map at position $u + d'$ has to be set to $\mathcal{G}(u + d', v) = d'$. More generally speaking, let $\phi : \mathbb{R} \to \mathbb{R}$ be a disparity mapping function that defines how to map the normalized disparity $\mathcal{D}$ to a new disparity range. In order to create a corresponding stereoscopic image pair, the goal disparity map then is defined as

$$\mathcal{G}(u + \phi(\mathcal{D}(u, v, s)), v) = \phi(\mathcal{D}(u, v, s)). \tag{2}$$

$\mathcal{G}$ can be constructed by iterating over all pixels $u$ in the reference image $I_s$. The construction of $\mathcal{G}$ is neither surjective nor injective due to occlusions and disocclusions in the scene. Intuitively, one cannot define disparity constraints for scene elements that are not visible in $I_s$. Hence these regions remain undefined in $\mathcal{G}$ (see Figure 7). However, in practice, these monocular regions span only a small number of pixels, hence we can compute a plausible out-

**(a)** *Effects of disocclusion and undefined goal disparity*



**(b)** *Effects of occlusions and competing goal disparities*
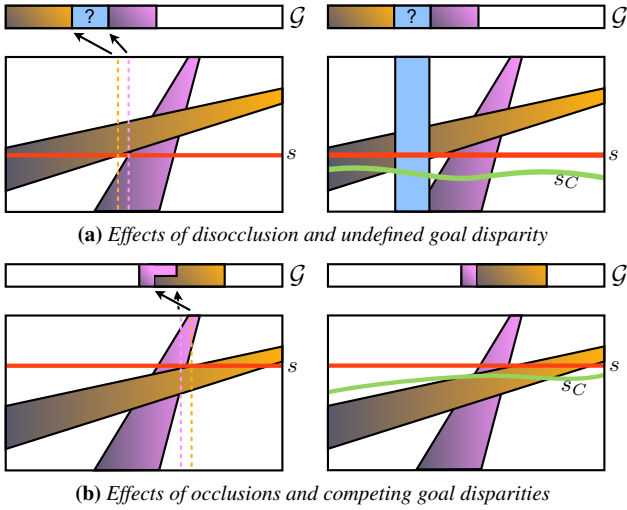
**Figure 7:** *Illustration of the effects of disocclusions and occlusions. (a) Since only depth of scene elements visible in the reference image $I_s$ is known, the construction of $\mathcal{G}$ by forward mapping of disparities $\phi(\mathcal{D}(*, *, s))$ (see Eq. (2)) is not surjective. This can lead to undefined segments in $\mathcal{G}$, illustrated in blue on the left. Intuitively, disparity constraints cannot be defined for regions that are occluded in $I_s$, but visible in an output view $I_C$. Since these regions generally span only a small number of pixels, a reasonable choice is to impose a smoothness prior on the cut $s_C$. This ensures that the output image shows an undistorted, standard perspective view of all undefined areas, illustrated in blue on the right. (b) Similarly, due to visible regions in $I_s$ that will be occluded in other views, the construction of $\mathcal{G}$ is not injective. Differently remapped disparities of close and distant objects compete for the same range in $\mathcal{G}$ (overlapping orange and pink region). In this case, we store the disparity constraints for the object closer to the camera (right).*

put view by imposing certain smoothness criteria on $s_C$, which are described in the following section.

Now recall that the true disparity volume $\mathcal{T}_s(u, v, s')$ represents the actual disparity of a point $(u, v, s')$ with respect to $I_s$; correspondingly, the difference volume $\mathcal{T}_s(u, v, s') - \mathcal{G}(u, v)$ then represents the deviation of a pixel's disparity from the desired goal disparity. The underlying idea of our algorithm for generating the output image $I_C$ is to find a cut $s_C$ that passes close to the zero set of this difference volume (see Figure 8). The following sections describe how the problem of finding $s_C$ can be formulated as an energy minimization problem.

### 4.2 Formulation as an Energy Minimization Problem

With the discretization of the light field described in Section 3, the energy measuring the deviation of a 2D cut $s_C$ can be expressed as

$$E_{\mathrm{d}}(s_C) = \sum_{\mathbf{p}} |\mathcal{T}_s(\mathbf{p}, s_C(\mathbf{p})) - \mathcal{G}(\mathbf{p})|. \qquad (3)$$

While a cut computed from this data term alone closely follows the prescribed goal disparities, it does not enforce any coherence between neighboring output rays/pixels and therefore can lead to visual artifacts in noisy or ambiguous estimates of $\mathcal{T}_s$. These artifacts are particularly noticeable in highly textured regions or at depth discontinuities.

Therefore we design an additional content-adaptive smoothness term according to the following observations:

- In the proximity of visually salient parts of an image, such as depth discontinuities and highly textured regions, we would like
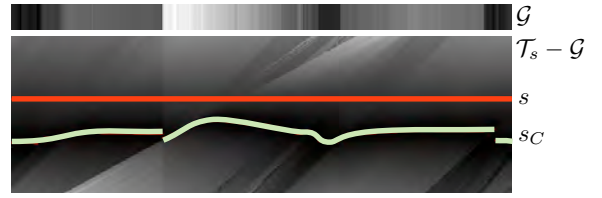


**Figure 8:** *Goal disparity. The upper image shows a 1D slice of the 2D goal disparity map $\mathcal{G}$. The difference volume $\mathcal{T}_s - \mathcal{G}$, shown as an unsigned function in this figure, then represents the deviation of each point in the light field from the desired disparity. Our algorithm computes a cut $s_C$ that passes close to the zero set of this volume. The resulting image $I_C$ and $I_s$ then form a multi-perspective stereoscopic image pair with the desired goal disparities.*

to enforce a higher smoothness to increase the coherence of the rays selected by $s_C$. In particular, we would like to assign a higher saliency to scene elements close to the camera and cut through more distant regions.

- In visually less salient, homogeneous and continuous regions, smoothness constraints can be relaxed in order to increase the flexibility of the cut to perform multi-perspective view transitions in the light field.

These properties are formulated in the following energy for measuring the smoothness of a cut $s_C$:

$$
\begin{aligned}
E_{\mathrm{s}}(s_C) = &\sum_{(\mathbf{p}, \mathbf{q}) \in N_u} |s_C(\mathbf{p}) - s_C(\mathbf{q})|\, p_u(*) + \\
&\sum_{(\mathbf{p}, \mathbf{q}) \in N_v} |s_C(\mathbf{p}) - s_C(\mathbf{q})|\, p_v(*), \text{with} \qquad (4)
\end{aligned}
$$

$$p_u(*) = \min\left(p_{\max}, |\partial_s \mathcal{D}(*)| + \lambda \mathcal{D}(*) + \kappa |\partial_s \mathcal{L}(*)|\right), \text{and}$$
$$p_v(*) = \min\left(p_{\max}, |\partial_s \mathcal{D}(*)| + \lambda \mathcal{D}(*) + \kappa |\partial_u \mathcal{L}(*)|\right),$$

where $N_u$ and $N_v$ are the sets of all neighboring pixels along the $u$-axis and $v$-axis, respectively. $(*)$ stands for $(\mathbf{p}, s_C(\mathbf{p}))$. The term $|s_C(\mathbf{p}) - s_C(\mathbf{q})|$ penalizes variation of the cut $s_C$ along the $s$-axis, i.e., view transitions. This penalty is weighted by the content-adaptive terms $p_u(*)$ and $p_v(*)$, respectively.

For both axes, the weighted terms depend on the depth discontinuities $\partial_s \mathcal{D}$ and the absolute normalized disparity $\mathcal{D}$. Intuitively, for scene elements very close to the viewer, even view transitions to an adjacent view may introduce noticeable disparity jumps. Increasing smoothness for nearby regions and strong depth discontinuities effectively moves view transitions to the background. Note that these concepts can be easily generalized to other types of image saliency, for example to encourage view transitions in less salient regions.

These depth-based terms are sufficient for controlling smoothness of the cut. Optionally, for the $u$-axis, we can take the change of radiance between different input images $I_s$ into account, while for $v$ we penalize jumps of the cut in the proximity of vertical image edges. Finally, the maximum penalty $p_{\max}$ ensures that the cut can be discontinuous, similar to the concept of robust nonlinear error functions. In our discrete setting, the partial derivatives are computed via forward differences. The above constants are only necessary for bringing all terms to a similar scale, but not critical to the quality of the results. For the results in this paper we used $\lambda = 0.5, \kappa = 1$, and $p_{\max} = 3$. The final energy is then defined as

$$E(s_C) = E_{\mathrm{d}}(s_C) + k E_{\mathrm{s}}(s_C), \qquad (5)$$

with $k = 25$. One additional interpretation of the smoothness term is that an increased value of $k$ leads to "flattened" cuts, i.e., output images closer to a standard perspective image. We believe that this is a notable property, since higher smoothness does not compromise image quality, but simply falls back to the original input images.

## 4.3 Optimization via Graph Min-Cuts

The minimization of Eq. (5) can be solved using graph cut optimization [Boykov et al. 2001; Boykov and Kolmogorov 2004]. We employ the standard procedure for binary s-t-cuts.

- For $n$ input images of dimension $w \times h$ we construct a 3D regular graph of size $w \times h \times (n + 1)$.
- A ray at position $(u, v, s')$ is associated with a directional graph edge between the corresponding two nodes along the $s$-axis, and the edge weight is chosen as $|\mathcal{T}_s(u, v, s') - \mathcal{G}(u, v)|$.
- Bi-directional edges between neighboring nodes along the $u$-axis and $v$-axis are weighted with the corresponding smoothness values $kp_u$ and $kp_v$, respectively.
- Boundary nodes corresponding to parameters $s = 0$ and $s = n$ are connected to the source and sink of the graph, respectively, with infinite weights.

The min-cut of this graph then yields the desired cut surface $s_C$ that minimizes Eq. (5).

We explored various conceptual modifications of this algorithm and the energies. Most notably, we also experimented with additional penalty edges for enforcing $C^0$ continuity [Rubinstein et al. 2008]. However, we found that piecewise continuous cuts provide more flexibility due to the support for sudden view transitions. Other algorithms for minimizing this energy would be applicable as well. An alternative formulation could be based on multi-labeling via $\alpha$-expansion [Boykov et al. 2001], where each label is associated with a particular $u$-$v$-slice along the $s$-axis of the EPI volume. While such an approach reduces the size of the graph, it has certain restrictions regarding the optimality of the result. In practice, however, we found the binary s-t-cut to produce reliable results.

## 4.4 Extensions of the Basic Algorithm

There exist several useful extensions of our basic algorithm which we briefly describe next.

### 4.4.1 N-View Stereo from Multiple Cuts

Instead of creating a stereoscopic pair consisting of a standard perspective image $I_s$ and a multi-perspective image $I_C$, the algorithm can be easily extended to create two multi-perspective cuts. For example, two goal disparity maps $\mathcal{G}_L$ and $\mathcal{G}_R$ can be defined as $\mathcal{G}_L(u - \frac{1}{2}\phi(\mathcal{D}(u, v, s)), v) = -\frac{1}{2}\phi(\mathcal{D}(u, v, s))$ and $\mathcal{G}_R(u + \frac{1}{2}\phi(\mathcal{D}(u, v, s)), v) = \frac{1}{2}\phi(\mathcal{D}(u, v, s))$, where the goal disparities are evenly distributed to both views and the reference view is centered between the two corresponding cuts. More than two views can be handled in a similar manner. As we will discuss in Figure 14 and show in our supplemental video, this multi-cut approach is particularly interesting for content generation for multiview autostereoscopic displays.

While defining a goal disparity map for each view separately provides high flexibility, many application scenarios such as multiview autostereoscopic displays often require a simple linear change of disparity between views. This can be exploited for an efficient, interpolation based algorithm to generate multiple views, given just the reference view $s$ and one multi-perspective cut $s_C$. Suppose $s_C$ has been computed from a mapping function $\phi(\mathcal{D}(u, v, s))$, and that the two views $s$ and $s_C$ should be converted into $n$ views with linearly interpolated disparities. From Eq. (2) we can conclude that the goal disparities of view $k \in [0, n]$ are given as $\mathcal{G}(u + \frac{k}{n}\phi(\mathcal{D}(u, v, s)), v) = \frac{k}{n}\phi(\mathcal{D}(u, v, s))$, meaning that a cut $s_C^k$ will contain the interpolated points of all EPI lines connecting corresponding points of $s$ and $s_C$.
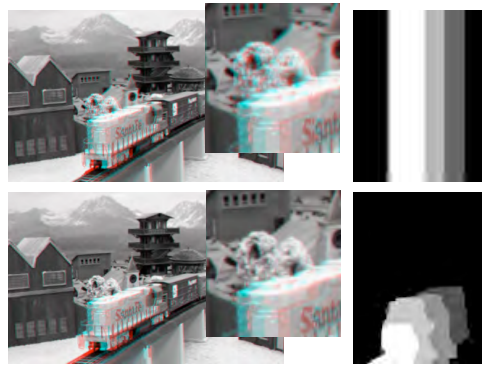


***Figure 9:*** *Comparison to Peleg et al. [2001]. Since the method of Peleg et al. supports only column-wise disparity control, it is not possible to achieve truly localized effects as with our per-pixel control over the disparity space.*
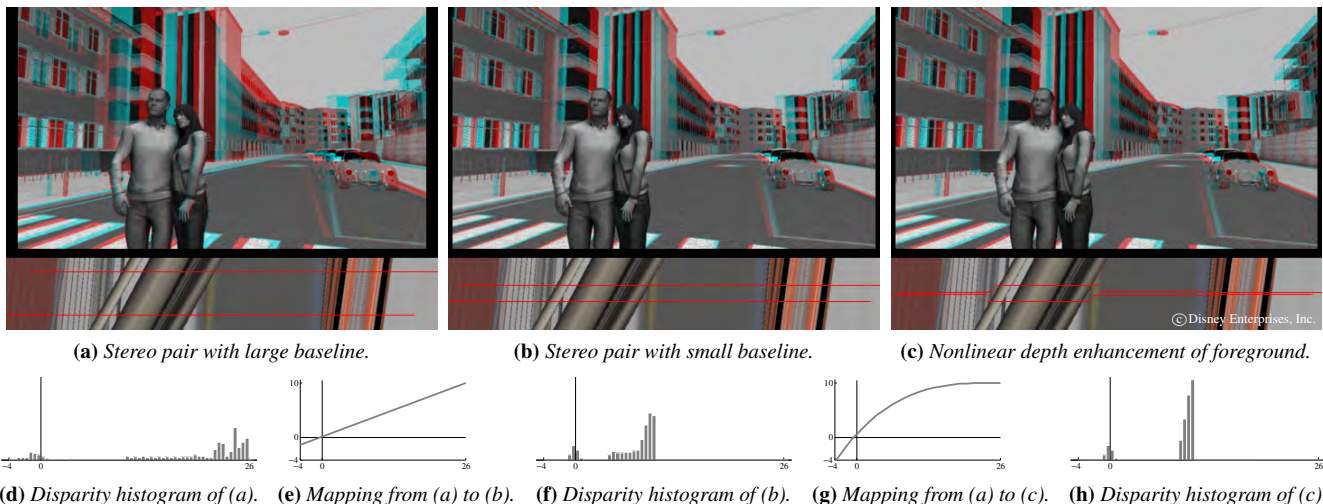
### 4.4.2 Stereoscopic Video Processing

In order to process video it is generally advisable to enforce a certain continuity between two cuts at consecutive time steps. One solution would be to enforce temporal smoothness by adding a temporal dimension to the graph structure. Each time step then has its own 3D subgraph, and corresponding nodes of subgraphs from consecutive time steps are connected via additional edges. Using a multi-label approach instead of binary labeling, the graph dimension could be reduced to 3D again. The disadvantage of this approach is that it has to process the whole 4D spatio-temporal light field volume at once.

Our solution uses an exponentially decaying influence of previous time steps on the data and smoothness terms for the current time step. Let $e_t$ denote the edge weight for a given time step $t$ according to Eq. (3) and Eq. (4). During the update of the graph structure from time $t - 1$ to $t$, we set the temporally averaged edge weight $e'_t = \alpha e_t + (1 - \alpha)e_{t-1}$ for any edge. However, the temporal evolution of a light field is quite coherent in general. For all examples included in the supplemental video a weight $\alpha = 0.9$ has been used.

### 4.4.3 Deferred Rendering for Computer-Generated Content

Our method is particularly interesting for computer-generated content such as 3D animation movies. Implementing multi-perspective camera models into the CG rendering pipeline to meet the expectations of a director regarding control and flexibility is often a difficult problem [Neuman 2010]. Warping the 3D geometry instead is not an alternative, since this does not allow for arbitrary complex disparity constraints without compromising the scene composition, lighting calculations, or visual effects. Our method shifts the effort from the artist towards automatic computations: the well-established CG pipeline for modeling, shading, and cameras remains unaltered, and stereography becomes a simple post-process.

However, given the significant rendering time, the generation of the complete light field of a complex scene is not often feasible. To deal with this, deferred rendering could be applied; since our algorithm works well with depth data only (the normalized disparity volume $\mathcal{D}$), it is sufficient to render only the depth maps of the input views. This is typically several orders of magnitude faster than rendering fully shaded color images. Even lower resolution proxy geometry could be used instead of the highly tessellated subdivision surfaces often used in rendering. Once the required set of input views is known from the cut $s_C$, those images or just the required light rays can be rendered and combined. These considerations render our method a highly practical solution.

**(a)** *Stereo pair with large baseline.* **(b)** *Stereo pair with small baseline.* **(c)** *Nonlinear depth enhancement of foreground.*



**(d)** *Disparity histogram of (a).* **(e)** *Mapping from (a) to (b).* **(f)** *Disparity histogram of (b).* **(g)** *Mapping from (a) to (c).* **(h)** *Disparity histogram of (c).*

***Figure 10:*** *Nonlinear disparity remapping. (a) shows a standard stereo pair with a large baseline where the foreground provides a good impression of depth. The background disparities, however, are quite large and can lead to ghosting artifacts or even the inability to fuse, when viewed on a larger screen. (b) Decreasing the baseline reduces the problems with respect to the background, but also considerably reduces the depth between the foreground and the background. (c) With a nonlinear disparity mapping function we can enhance the depth impression of the foreground, while keeping the maximum disparities in the background bounded as in (b). Compare the disparity distribution (h) to that of the small baseline stereo (f). (d) (f) and (h) show the disparity distributions of respective stereo pairs, and (e) and (g) show the disparity remapping functions. Observe that the depth between the foreground and the background in (d) is preserved in (h), while it is not in (f).*

### 4.4.4 Different Light Field Parameterizations

Section 3 made certain assumptions about the acquisition and parameterization of a light field, and the required sampling scheme to generate an image from a given cut $s_C$. We also assumed that the reference view is a standard perspective view, and that correspondingly our desired output view should be as-perspective-as-possible as well, while satisfying our prescribed goal disparity constraints. For this scenario we argued that a regular sampling along the $u$-$v$-dimension is the most natural choice. In other applications scenarios, however, it could be desirable to produce other forms of stereoscopic images, such as omnistereo panoramas as discussed by Peleg et al. [2001], or stereo pushbroom panoramas and cyclographs as discussed by Seitz [2001]. For these types of images the light field parameterization and image cut have to be reconsidered.

As mentioned in Section 3, a stereo pushbroom panorama simply corresponds to a $v$-$s$-cut instead of a $u$-$v$-cut. This insight renders handling of stereoscopic pushbroom images straightforward; one has to swap the dimensions $u$ and $s$ in our formulation, and then apply the algorithm as is. For omnistereo panoramas and cyclographs, the 3D light fields are constructed with a rotating camera at a certain offset orthogonal to the rotation axis, yielding a $u$-$v$-$\alpha$ volume. Both above mentioned works show that planar $v$-$\alpha$ slices can be used to produce stereoscopic panoramas. Peleg et al. [2001] also show an algorithm for adaptively changing the camera baseline for each image column. Our concept of multi-perspective, piecewise smooth cuts with a global optimization scheme generalizes these ideas to *per-pixel* control over the baseline (see Figure 9 for a comparison).

## 5 Results

In this section we present results that are generated using our algorithm, given a prescribed set of disparity constraints. Please refer to the supplemental material for the video examples. All results are presented as gray-scale, red-cyan anaglyph images ( , red left). This not only allows for seeing the images stereoscopically in
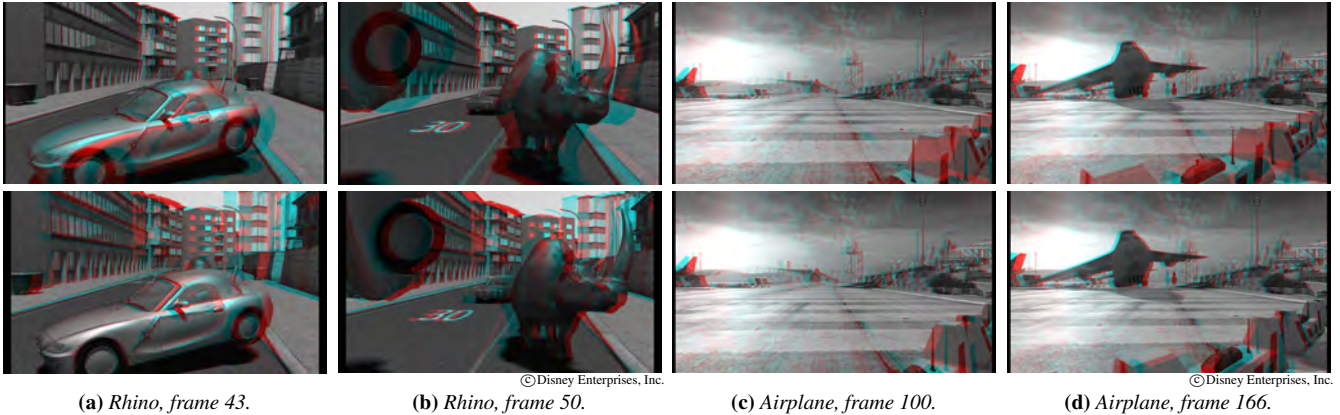
3D, but also to quickly assess the horizontal parallax between images without glasses. We show results for computer-generated light fields as well as for real-world images, some of which are taken from UCSD/MERL Light Field Repository[1]. As high frame rate light field cameras are not yet available, we captured stop motion videos to demonstrate our method on live-action footage. For our results on computer-generated light fields, the normalized disparity volume $\mathcal{D}$ has been constructed from the z-buffer of the input images. For the real-world examples we used the method of Kang and Szeliski [2004], but in principle any method suitable for depth reconstruction from multiple images can be employed. We first provide a set of examples demonstrating the application of different disparity mapping operators $\phi$. In our experiments, $\phi$ is defined on the normalized disparity, which is converted to pixel disparities by taking the physical depth budget, screen size and viewer position into account. For all results, we computed both output views.

**Linear remapping:** The most straightforward example is a linear remapping of the disparity range, which corresponds to changing the camera baseline between two standard perspective views. In this case our method simply produces the expected planar cuts (e.g., Figure 10b). However, in this context a notable property of our method is that it eliminates the quite abstract and unintuitive concept of the camera baseline. Instead, one can directly specify the desired goal disparity range of the output images. This is the preferred method in actual production environments [Neuman 2010].
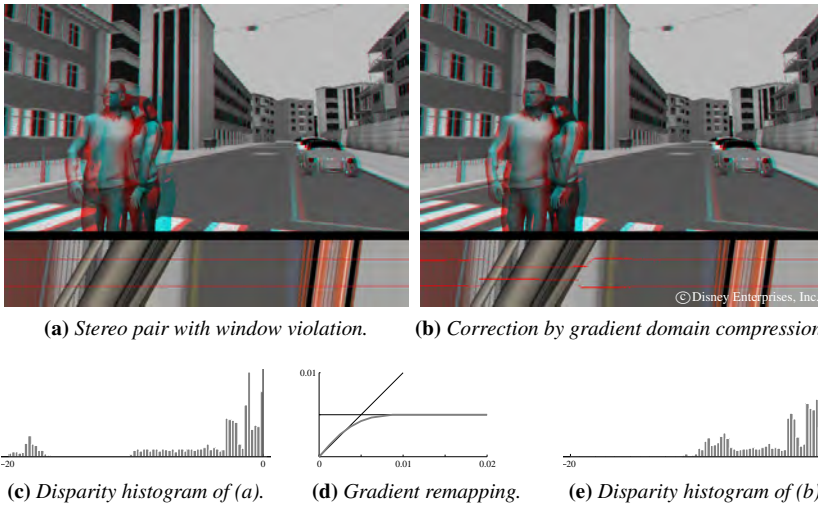
**Nonlinear and gradient-based disparity remapping:** The strengths of our method are revealed for application scenarios where nonlinear changes of the disparity space are required. In principle, arbitrary remapping functions $\phi$ can be applied to construct the desired goal disparity volume, and even constant disparities are possible. For example, $\phi$ could be any of the nonlinear disparity mapping operators introduced by Lang et al. [2010] for display adaptation, stereoscopic error correction, or artistic effects. These functions can act globally on the complete domain as well as locally by remapping disparity gradients. For the gradient based remapping, we compute the gradient field in

---

[1] http://graphics.ucsd.edu/datasets/lfarchive/

**(a)** *Rhino, frame 43.*     **(b)** *Rhino, frame 50.*     **(c)** *Airplane, frame 100.*     **(d)** *Airplane, frame 166.*

©Disney Enterprises, Inc.

**Figure 11:** *More examples for nonlinear disparity gradient remapping in order to reduce the overall disparity range, while preserving the perception of depth discontinuities and local depth variations. The first row shows the stereo pairs with two perspective images and a fixed baseline, while the second row shows the depth remapped versions. In particular, for the airplane scene the disparity gradient of the image's upper half was intensified, and the gradient of the lower half was attenuated. See the supplemental video for the full sequences.*



**(a)** *Stereo pair with window violation.*     **(b)** *Correction by gradient domain compression.*

©Disney Enterprises, Inc.

**(c)** *Disparity histogram of (a).*    **(d)** *Gradient remapping.*    **(e)** *Disparity histogram of (b).*

**Figure 12:** *Gradient domain compression. (a) A typical example for a stereo pair where the partially cropped couple features strong negative disparity, resulting in a so called window violation [Mendiburu 2009]. Changing the convergence would increase the background disparities, potentially leading to the same problems as in Figure 10a. (b) With our method, we can resolve this stereoscopic issue by gradient domain compression of strong negative disparities. This effectively pushes the couple closer to the screen while keeping the background disparities unchanged. (c) and (e) show the disparity distribution of (a) and (b), respectively. Note that in this example the empty space between the foreground and the background in (c) is squeezed in (e). (d) shows the gradient remapping function, which effectively compresses strong disparity gradient.*

both $x$ and $y$ directions and process the gradients non-uniformly using the gradient magnitude, e.g., to supress big disparity jumps. We then reconstruct the height field by integration of the gradients using a Poisson solver [Agrawal and Raskar 2007], and use this reconstructed height field to set a final goal disparity.

Figure 1, 10, 11, 12, and 14 show various examples for applications of nonlinear disparity remapping and gradient-based disparity processing. The respectively used mapping function $\phi$ and the histograms of disparities before and after applying our method are shown as well. Figure 10 and 12 show typical issues arising in the production of stereoscopic content, and how they can be resolved using our method. In Figure 1 we use gradient domain remapping to increase the dramatic appearance of the scene by emphasizing depth gradients. Figure 11 shows a similar example where we reduce the overall disparity range while preserving the depth perception around depth discontinuities and local depth variations. See the supplemental video for the complete sequence.

**Artistic control:** In addition to the automatic mappings described above our method allows for concise manual control of disparities, which is an important requirement in any stereoscopic production environment. Users can directly modify the depth map $\mathcal{D}$ at the reference view $s$, e.g., using painting tools. The goal disparities are then set using the modified depth map. This allows for interesting artistic effects as well as fine-scale correction of the stereoscopic

impression of a scene. Figure 9 and 13 show examples for manual control over disparity.

**Multi-view autostereoscopic displays:** A further interesting application domain is multi-view autostereoscopic displays. Similarly to stereoscopic displays, these displays have a limited depth budget. Thus, it is usually necessary to prefilter and remap an input light field to the available spatio-angular display bandwidth in order to avoid inter-perspective aliasing [Zwicker et al. 2006]. We can obtain properly remapped data to drive a particular automultiscopic display by computing multiple cuts through a light field. In Figure 14 we show an example for an 8-view autostereoscopic display from Alioscopy. More results are shown in the supplemental video.

**Performance:** The computationally intensive steps of our method are the graph construction and the min-cut computation. The required time for the graph construction depends on the size of the underlying light field, while the optimization depends additionally on the complexity of the disparity constraints. The timings below have been measured for our MATLAB prototype on a machine with an Intel i7 2.8Ghz CPU and 6GB memory. For example, the complete graph construction for a single video frame consisting of 50 images with $640 \times 480$ resolution used for Figure 1 takes about 2.5 seconds. The min-cut computation with goal disparities computed by the gradient compression takes about 30 seconds. Overall per-frame computation times for all presented results ranged from 10

| (a) Original stereo. | (b) Resulting stereo output. | (c) Disparity maps and histograms. |
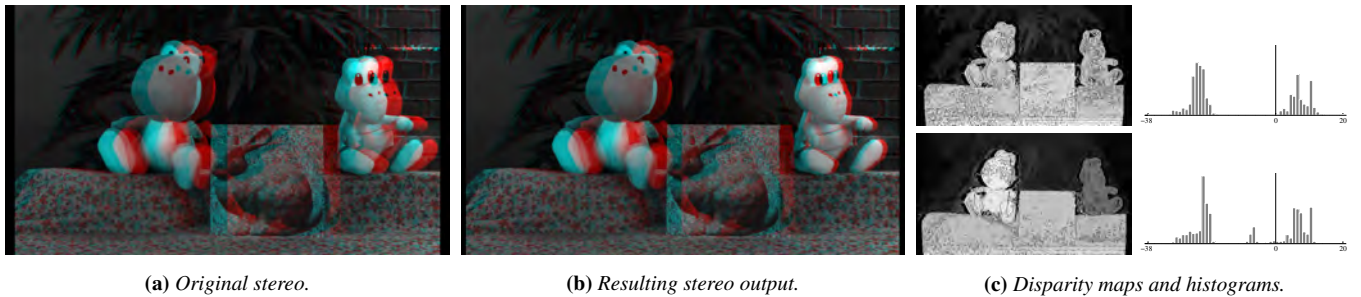
**Figure 13:** *Artistic control over depth. (a) (b) We manually masked the two toys which are approximately at the same depth in the original stereo, and then defined different goal disparities for those regions. The resulting stereo output of our algorithm creates a different depth sensation for the two toys, even though they are placed at the same distance. (c) shows the actual disparity maps and disparity distributions of the two stereoscopic image pairs. The two image pairs exhibit significantly different disparities in the area of two toys. Also note the new peak in the disparity histogram of the output stereo which corresponds to the toy on the right.*



(a) *Eight views optimized for a multi-view autostereoscopic display.*



(b) *Photographs of these views shown on a 8-view autostereoscopic display.*

**Figure 14:** *Multiple view generation. (a) Multi-perspective 8-view stereo, optimized with respect to the disparity range of an Alioscopy 8-view autostereoscopic display. (b) Unoptimized content easily leads to considerable ghosting artifacts (left column). Our method can automatically compute n-view stereo images that are designed to meet the disparity requirements of the output device and at the same time enhance perceived depth (right column).*

seconds to about 1.5 minutes for the more complex airplane and elephant data sets. In general, the more the goal disparities differ from the standard perspective of the input images the more processing time is required. The memory requirements of the employed graph cut algorithm for a light field of size $s \times u \times v$ are $115 \cdot s \cdot u \cdot v$ bytes. We expect significant speed and memory improvements for more efficient graph-cut implementations.

**Limitations and discussion:** Our current method has some limitations. First, very complex disparity constraints, which can only be resolved by many view transitions of the cut, can result in slight stretching or bending of image content. However, we do not observe any noticeable artifacts in our results, since our smoothness term has been designed to hide such transitions in visually less salient background regions. Second, our method is very reliable when using dense light field sampling and accurate depth estimates. With a lower number of input images and less accurate depth map quality the cut may become less smooth and potentially cut through foreground regions. Fortunately, these effects can be compensated by setting $\lambda$ and $\kappa$ in Eq. (4) higher to strengthen the depth and the radiance gradient based smoothness. By doing so, view transitions are less likely to happen inside well textured, foreground objects. With higher smoothness the output images are composed of standard perspective image segments which can also be interpreted as "multi-perspective image stitching."

Finally, even though it is often very challenging to compute high quality depths for real world images, the proposed algorithm is quite robust and generally produces high quality output stereoscopic images. The rationale behind this is that for those regions where accurate cut computation is required to deal with the high frequency texture or the depth discontinuity, depth computation also becomes reliable for the very same reason. On the other hand, for those regions where the depth computation often fails, such as texture-less regions with uniform color, the caused inaccuracy of the cut and thereby undesirable ray selection are not very noticeable as these regions are not visually salient.

# 6 Conclusion

We have presented a general, multi-perspective framework for computing stereoscopic images from a light field, which satisfy a prescribed set of per-pixel goal disparity constraints. The core idea is to compute piecewise continuous cuts through the light field that minimize an energy derived from the goal disparities. We have demonstrated that our method is an effective and practical solution to key issues arising in today's stereoscopic content generation and post-production, and we believe that it will be an even more important tool for upcoming plenoptic camera systems.

The presented framework provides a multitude of future research opportunities. For example, the current energy formulation strives at finding a cut that follows the goal disparity constraints as closely as possible without introducing visual artifacts. However, it could be valuable to extend this formulation with more sophisticated insights about stereoscopic perception, visual saliency, or temporal coherence. Moreover, our image generation selects pixels from the original input views and does not explicitly handle potential resampling issues. In this context, gradient-based image reconstruction, gradient-domain cuts, sub-pixel resolution techniques, and more sophisticated methods for up-sampling of light fields would be interesting to investigate. Finally, our solution to multiple cut generation defines goal disparities with respect to the reference view. To define disparity constraints for regions occluded in the reference view, this formulation could be extended to pairwise constraints between neighboring views.

On a more general level we would like to further investigate how our method relates to previous works such as Peleg et al. [2001] or Lang et al. [2010]. For instance, the stereoscopic image warping by Lang et al. could in principle be explained as planar cuts with a deformed light field or an adaptive $u$-$v$-parameterization. We believe that a formulation of these techniques within our framework would lead to further interesting insights on stereoscopic imaging.

## Acknowledgements

## References

ADELSON, E., AND BERGEN, J. 1991. The plenoptic function and the elements of early vision. *Computational Models of Visual Processing*, 3–20.

ADELSON, E. H., AND WANG, J. 1992. Single lens stereo with a plenoptic camera. *IEEE PAMI 14*, 2, 99–106.

AGRAWAL, A., AND RASKAR, R. 2007. Gradient domain manipulation techniques in vision and graphics. In *ICCV Courses*.

BOYKOV, Y., AND KOLMOGOROV, V. 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE PAMI 26*, 9, 1124–1137.

BOYKOV, Y., VEKSLER, O., AND ZABIH, R. 2001. Fast approximate energy minimization via graph cuts. *IEEE PAMI 23*, 11, 1222–1239.

CHAI, J., CHAN, S.-C., SHUM, H.-Y., AND TONG, X. 2000. Plenoptic sampling. In *SIGGRAPH*, 307–318.

DIDYK, P., RITSCHEL, T., EISEMANN, E., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2011. A perceptual model for disparity. In *SIGGRAPH*, 96:1–96:10.

FELDMANN, I., SCHREER, O., AND KAUFF, P. 2003. Nonlinear depth scaling for immersive video applications. In *WIAMIS*.

GEORGIEV, T., ZHENG, C., NAYAR, S., CURLESS, B., SALESIN, D., AND INTWALA, C. 2006. Spatio-angular resolution tradeoffs in integral photography. In *EGSR*, 263–272.

GORTLER, S. J., GRZESZCZUK, R., SZELISKI, R., AND COHEN, M. F. 1996. The Lumigraph. In *SIGGRAPH*, 43–54.

HARTLEY, R. I., AND GUPTA, R. 1997. Linear pushbroom cameras. *IEEE PAMI 19*, 9, 963–975.

HELD, R. T., AND BANKS, M. S. 2008. Misperceptions in stereoscopic displays: A vision science perspective. In *APGV*, 23–32.

HOLLIMAN, N. 2004. Mapping perceived depth to regions of interest in stereoscopic images. In *SPIE*, vol. 5291, 117–128.

JONES, G., LEE, D., HOLLIMAN, N., AND EZRA, D. 2001. Controlling perceived depth in stereoscopic images. In *SPIE*, vol. 4297, 42–53.

KANG, S. B., AND SZELISKI, R. 2004. Extracting view-dependent depth maps from a collection of images. *IJCV 58*, 2, 139–163.

KIM, M., LEE, S., CHOI, C., UM, G.-M., HUR, N., AND KIM, J. 2008. Depth scaling of multiview images for automultiscopic 3D monitors. In *3DTV-CON*, 181–184.

KOPPAL, S. J., ZITNICK, C. L., COHEN, M. F., KANG, S. B., RESSLER, B., AND COLBURN, A. 2011. A viewer-centric editor for 3D movies. *IEEE CG&A 31*, 1, 20–35.

LANG, M., HORNUNG, A., WANG, O., POULAKOS, S., SMOLIC, A., AND GROSS, M. 2010. Nonlinear disparity mapping for stereoscopic 3D. In *SIGGRAPH*, 75:1–75:10.

LEVOY, M., AND HANRAHAN, P. 1996. Light field rendering. In *SIGGRAPH*, 31–42.

MENDIBURU, B. 2009. *3D Movie Making: Stereoscopic Digital Cinema from Script to Screen*. Focal Press.

NEUMAN, R., 2010. Personal Communication with Robert Neuman, Chief Stereographer, Disney Animation Studios.

NG, R., LEVOY, M., BRÉDIF, M., DUVAL, G., HOROWITZ, M., AND HANRAHAN, P. 2005. Light field photography with a hand-held plenoptic camera. Technical Report CSTR 2005-02, Stanford University.

PAJDLA, T. 2002. Geometry of two-slit camera. Research Report CTU-CMP-2002-02, Czech Technical University.

PELEG, S., BEN-EZRA, M., AND PRITCH, Y. 2001. Omnistereo: Panoramic stereo imaging. *IEEE PAMI 23*, 3, 279–290.

RADEMACHER, P., AND BISHOP, G. 1998. Multiple-center-of-projection images. In *SIGGRAPH*, 199–206.

RUBINSTEIN, M., SHAMIR, A., AND AVIDAN, S. 2008. Improved seam carving for video retargeting. In *SIGGRAPH*, 16:1–16:9.

SEITZ, S. 2001. The space of all stereo images. In *IEEE ICCV*, vol. 1, 26–33.

THOMAS, F., AND JOHNSTON, O. 1995. *The Illusion of Life: Disney Animation*. Hyperion.

VEERARAGHAVAN, A., RASKAR, R., AGRAWAL, A. K., MOHAN, A., AND TUMBLIN, J. 2007. Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. In *SIGGRAPH*, 69:1–69:12.

WARD, B., KANG, S. B., AND BENNETT, E. P. 2011. Depth director: A system for adding depth to movies. *IEEE CG&A 31*, 1, 36–48.

WILBURN, B., JOSHI, N., VAISH, V., TALVALA, E.-V., ANTUNEZ, E., BARTH, A., ADAMS, A., HOROWITZ, M., AND LEVOY, M. 2005. High performance imaging using large camera arrays. In *SIGGRAPH*, 765–776.

WOOD, D. N., FINKELSTEIN, A., HUGHES, J. F., THAYER, C. E., AND SALESIN, D. H. 1997. Multiperspective panoramas for cel animation. In *SIGGRAPH*, 243–250.

WOODS, A., DOCHERTY, T., AND KOCH, R. 1993. Image distortions in stereoscopic video systems. In *SPIE*, vol. 1915, 36–48.

YANG, J. C., EVERETT, M., BUEHLER, C., AND MCMILLAN, L. 2002. A real-time distributed light field camera. In *EGWR*, 77–86.

YU, J., AND MCMILLAN, L. 2004. General linear cameras. In *ECCV*, 14–27.

YU, J., MCMILLAN, L., AND STURM, P. 2010. Multi-perspective modelling, rendering and imaging. *Computer Graphics Forum 29*, 1, 227–246.

ZOMET, A., FELDMAN, D., PELEG, S., AND WEINSHALL, D. 2003. Mosaicing new views: The crossed-slits projection. *IEEE PAMI 25*, 6, 741–754.

ZWICKER, M., MATUSIK, W., DURAND, F., AND PFISTER, H. 2006. Antialiasing for automultiscopic 3D displays. In *EGSR*.