# Mapping Perceived Depth to Regions of Interest in Stereoscopic Images

Nick Holliman

Department of Computer Science, University of Durham, Durham, United Kingdom

## ABSTRACT

The usable perceived depth range of a stereoscopic 3D display is limited by human factors considerations to a defined range around the screen plane. There is therefore a need in stereoscopic image creation to map depth from the scene to a target display without exceeding these limits. Recent image capture methods provide precise control over this depth mapping but map a single range of scene depth as a whole and are unable to give preferential stereoscopic representation to a particular region of interest in the scene.

A new approach to stereoscopic image creation is described that allows a defined region of interest in scene depth to have an improved perceived depth representation compared to other regions of the scene. For example in a game this may be the region of depth around a game character, or in a scientific visualization the region around a particular feature of interest.

To realise this approach we present a novel algorithm for stereoscopic image capture and describe an implementation for the widely used ray-tracing package POV-Ray. Results demonstrate how this approach provides content creators with improved control over perceived depth representation in stereoscopic images.

**Keywords:** Graphics Systems, Human Factors, Rendering, Virtual Reality, Stereoscopic, 3D display

## 1. INTRODUCTION

The properties of stereoscopic displays frequently present a challenge for content creators with many scenes containing a larger depth range than a 3D display can present to the viewer. For example, trials using a high quality desktop auto-stereoscopic 3D display by Sharp Corporation[1] demonstrated that the range of available perceived depth range could be as little as $[-50mm, +60mm]$ in-front and behind the display surface. These limits are generally understood to result from a combination of factors including the display's level of inter-channel crosstalk,[2] the quality of the stereoscopic images[3] and the ability of the viewer to fuse screen disparity into a single image.

Many methods have been proposed to control how scene depth is captured in a stereoscopic image. The problem can be considered a mapping of scene depth onto the available perceived depth range on the target stereoscopic display as shown in figure 1(a). Recent solutions are able to guarantee this mapping has certain properties, for example that the perceived depth will not exceed pre-defined ranges and that the perceived depth remains constant when there is head movement. This has resulted in benefits for content creators who can now define the scene depth range they want to capture while being sure that this will be reproduced as expected on the target display. Experience generating images using one of these methods[1] was positive, it allowed novice 3D display users to have repeatable control over the depth mapping from scene to displayed image. As a result content production using computer graphics and digital photography no longer required repeated trial and error to produce an image with defined stereoscopic depth properties.

However, current solutions to stereoscopic image capture have drawbacks due to the fact that they map a single range of scene depth onto the target display depth range as a whole. If the scene depth range is larger than the available display depth range this will result in compression of perceived depth, as in figure 1(a). This has the consquence that a region of interest in scene depth cannot be guaranteed a good representation in perceived depth in the displayed image.

Further author information: (Send correspondence to Nick Holliman.)
Email: n.s.holliman@durham.ac.uk, Telephone: +44 191 334 1761, Web: http://www.durham.ac.uk/n.s.holliman/

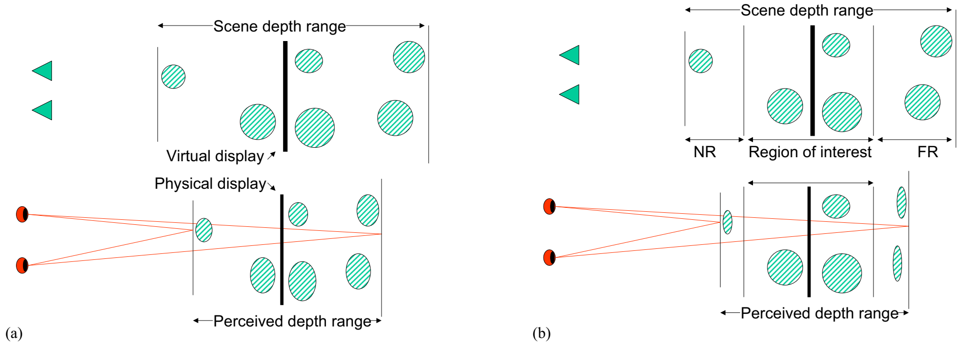

**Figure 1.** Stereoscopic image capture maps a scene depth range to a display perceived depth range: (a) Existing methods map a defined range of scene depth to the usable display perceived depth as a whole. (b) The new algorithm maps depth differently in three regions, near region(NR), region of interest (ROI) and far region(FR) allowing the available stereoscopic depth range to be split differently between the three regions, for example, as shown the ROI may be given the best stereo depth representation.

Our new approach to depth mapping allows the content creator to define different mappings from real scene depth to perceived display depth for three different ranges; a region of interest (ROI) in the scene and a near region (NR) and far region (FR) in the scene, as shown in figure 1(b). All three regions can be mapped to a different proportion of the available perceived depth range on the target 3D display. The result is that the image creator can choose to give better stereoscopic representation to the ROI compared to the near and far regions.

We aim to retain the scene outside the region of interest, rather than clip it or disguise it by blurring, because in many applications it is important to see context around the region of interest. For example in a game, or other interactive application, it can be important to see into the distance to check what is coming, but it is not important to have this represented using high resolution perceived stereo depth, particularly when other depth cues are available.[4] Applications that will benefit from the new algorithm are primarily those that are computer graphics based and include computer games, medical, scientific and information visualisation.

## 2. BACKGROUND

The benefits provided by stereoscopic displays are well known[3, 5–7] and include; depth perception relative to a display surface, perception of structure in visually complex scenes, improved perception of surface curvature, improved motion judgement and improved perception of surface material type. These benefits have been to shown to be helpful in providing users a better understanding or appreciation of visual information in a range of studies in different application domains.[4, 8–10]

An important step in obtaining these benefits is to use high quality 3D displays, with good basic 2D image quality (brightness, contrast, high resolution, full colour, moving images) as well as good 3D image quality (image channels well aligned spatially, in brightness and contrast and with low inter-channel crosstalk).[3, 8] Such displays are becoming available for desktop and laptop PC systems from companies including Dimension Technologies, Stereographics, SeeReal and Sharp Corporation.[3]

For the user the quality of the stereoscopic image pair is at least as important as the quality of the display it is presented on. The image pair needs to be captured with aligned cameras (or rectified to appear as if it has been) and to have minimal differences in luminance, contrast, focus and magnification. Synthetic computer graphics systems are better than optical photographic systems in this respect as the imaging process is not affected by physical alignment differences between the two capture channels. For all stereoscopic image capture methods a key requirement is to control the amount of perceived depth seen when the image is displayed on the target 3D display as this directly affects the viewer's ability to perceive a comfortable fused 3D image.

## 2.1. The Geometry of Perceived Depth

Depth perception in planar stereoscopic images has been widely studied and the geometry of stereoscopic depth perception is well known.[3, 11–13] The image disparity captured when a stereo image pair is created becomes physical screen disparity when the stereo pair is displayed on an electronic 3D display. The screen disparity is detected by the retina and interpreted by the brain as a perceived depth in-front or behind the screen plane, as shown in figure 2.

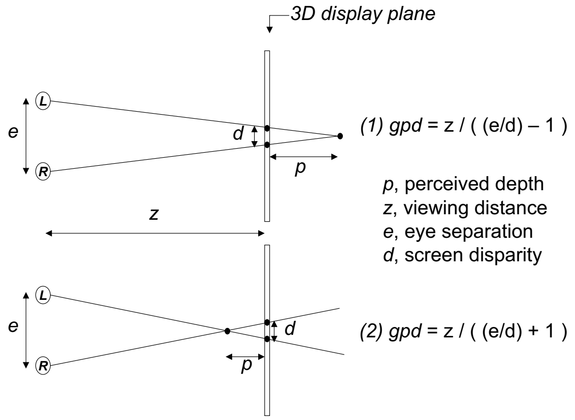

**Figure 2.** Geometric perceived depth for positive, (1), and negative, (2), screen disparity.

While a viewer's actual perception of depth resulting from a given screen disparity is important we adopt the common approximation of geometric perceived depth *gpd*.[1, 3, 12, 13] This is calculated, as shown in figure 2, from the value of screen disparity the viewer perceives and it is currently thought to provide a good working approximation to the viewer's actual perceived depth.

As noted earlier the range of *gpd* should be limited if the viewer is to comfortably fuse a stereo image pair.[1] In this paper we adopt a target *gpd* range around the display plane of $[-55mm, +55mm]$ as a conservative, but useful, range for users with normal stereoscopic vision, viewing a desktop stereoscopic display.

## 2.2. Current Approaches to Stereoscopic Image Generation

The generation of high quality stereoscopic images has received substantial interest and recent methods have achieved new quality levels. Recent reviews of the literature can be found in several publications[1, 3, 14, 15] and here we briefly review two approaches, analysing one in more detail.

Wartell[14, 15] studied the distortions in *gpd* due to different assumptions about the link between eye separation and camera placement when generating images for head tracked stereoscopic displays. Methods to remove part or all of these distortions by pre-distorting the scene using a defined transformation matrix were presented. These methods worked well but the best was complex and only easily applied to the generation of computer graphics images. In addition these methods considered the scene as a whole and were unable to give preferential treatment to a particular region of interest in the scene.

Jones et al[1] presented a method capable of controlling *gpd* applicable to content creation using computer graphics or digital cameras. The algorithm maps scene depth to perceived display depth as illustrated in figure 1(a). Near and far limits of a single range of scene depth (top) are identified and the algorithm maps these to be within the defined *gpd* range of the target 3D display (bottom). The problem of choosing a stereo camera separation is removed from the user as it is automatically calculated, along with other parameters, to achieve the required mapping between scene depth and perceived display depth.

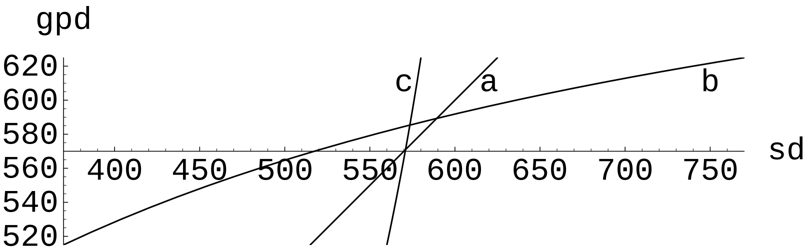

**Figure 3.** The result of applying the Jones[1] single range algorithm. In the orthoscopic case (a) the *gpd* directly reproduces scene depth *sd*. In case (b) scene depth is compressed into the available *gpd* range and in case (c) scene depth is expanded into the available *gpd* range.

The result of applying the Jones algorithm to map different scene depth ranges to the same target *gpd* is illustrated in figure 3. This uses target display parameters similar to the Sharp RD3D display and assumes a nominal viewing distance of 570mm[16] and a target *gpd* of [-55,+55] around the screen plane. In figure 3(a) the scene depth matches the *gpd* target and there is a one-to-one correspondence between the two. We can expect the viewer will see minimal difference between perceived depth in the original scene and on the display as long as the *gpd* target is kept within Panum's fusional area for the target display. In figure 3(b) the scene depth range captured is greater than the *gpd* target and there is predicted a compression of the scene depth; here the viewer is likely to perceive discrete objects to be closer together on the display than in the original scene. The opposite is predicted in figure 3(c) where the scene depth is smaller than the *gpd* target and there is an expansion of scene depth; here the viewer is likely to perceive discrete objects further apart on the display than in the original scene. Previous studies and our experience suggest these effects are apparent and we note in almost all cases when creating 3D content the scene depth is either compressed or expanded when viewed on a 3D display.

Considering the case in figure 3(b) where scene depth is compressed; an original range of $[370, 770]$ is compressed to a *gpd* of just $[515, 625]$. In this case a particular region of interest in the scene is at risk of receiving relatively little stereo representation in the displayed 3D image. For example, a region of interest in the scene of $[515, 625]$ will be represented to the viewer by a *gpd* range of $[569, 597]$ a depth compression by a factor of approximately fourteen.

In summary the guarantee by the Jones[1] algorithm not to exceed pre-defined ranges of *gpd* ensures that the viewer never sees excessive depth on the display treating the scene as a whole. If there is a specific region of interest in a scene then it cannot be given better stereoscopic representation than other regions of the scene. In addition if the total scene depth range is varying such as in an animation or real time scene navigation then the region of interest will not have a constant *gpd* representation.

## 3. A THREE-REGION DEPTH MAPPING ALGORITHM

We present a new algorithm that uses a three-region depth mapping to overcome the limitations of previous single region depth mapping algorithms and as a result provide new control over the way scene depth is mapped to *gpd*.

We consider two distinct geometries; the geometry defining the relationship between the viewer and the display and the geometry of the scene and camera. The three-region algorithm maps the defined regions, NR, ROI and FR in scene depth onto corresponding defined ranges of *gpd*. This mapping is required to meet the constraint that points on the region boundaries are projected to coincident positions, and hence depth, in the image pair whichever region they are considered to belong to. The algorithm implements perspective projection as a piecewise continuous function of scene depth and uses a different perspective projection (different stereo cameras) to capture each of the three regions.

### 3.1. Display Geometry

The geometry of display viewing is illustrated in figure 4. The viewer's half eye separation is given by $e$, the screen plane is at distance $z$ from the viewer and the half screen width is $w$. The total working geometric perceived

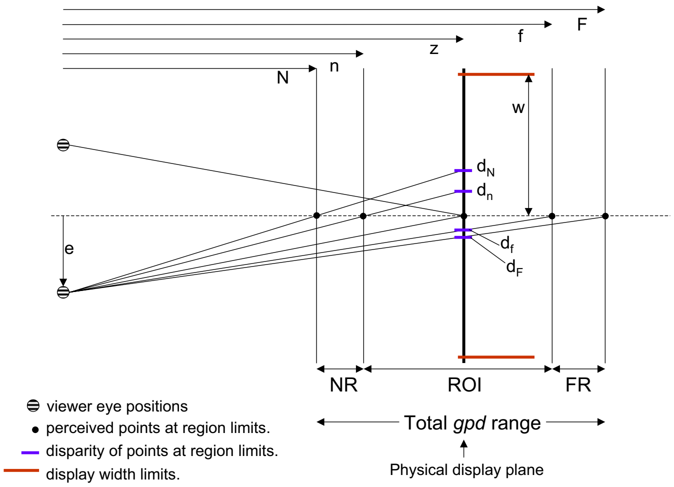

**Figure 4.** The display viewing geometry showing total *gpd* range and the split into near, ROI and far regions.

depth range is between the planes at distances $N$ and $F$ from the viewer. The total *gpd* range is divided into a near range, NR $[N, n]$, a region of interest range, ROI $[n, f]$, and a far range, FR $[f, F]$, by planes defined at distances $n$ and $f$ from the viewer.

The half screen disparities of points lying on the display viewing centre line for the planes at distances $N$, $n$, $f$, $F$ are given by $d_N$, $d_n$, $d_f$ and $d_F$ respectively. Note, in each case just the half disparity for one view is shown with the matching half from the other view omitted for clarity.

## 3.2. Scene Geometry

The geometry of the scene and camera is illustrated in figure 5. We assume that the image creator has positioned a single camera that frames the required view of the scene. The total depth range in the scene we are asked to capture is $[N', F']$ and this is divided into the near $[N', n']$, ROI $[n', f']$ and far $[f', F']$ regions by the planes at distances $n'$ and $f'$ from the viewer. These regions will be mapped to the defined ranges of *gpd* on the target display.

In single region methods a single stereo camera separation $a'$ is calculated to position the camera to take the left and right images. In the three-region approach we need to find three camera separations one each for the NR, ROI and FR regions. The calculations to determine these are described in the following three sections.

## 3.3. Region of Interest Mapping

The ROI mapping uses essentially the same approach as the Jones algorithm,[1, 17] although it differs in some of the measurements used for clarity.

The aim is to map the region of depth in the scene defined as the ROI onto the matching region of *gpd* identified for the target display. As in the original method the display is represented in the scene by a virtual display plane. This allows us to consider the mapping of scene depth onto disparities in the virtual display plane separately from the physical disparities on the target display.

Considering the geometry of the display space in figure 4 we can derive by similar triangles that the following relationships hold:

$$d_n = \frac{e(z-n)}{n} \tag{1}$$

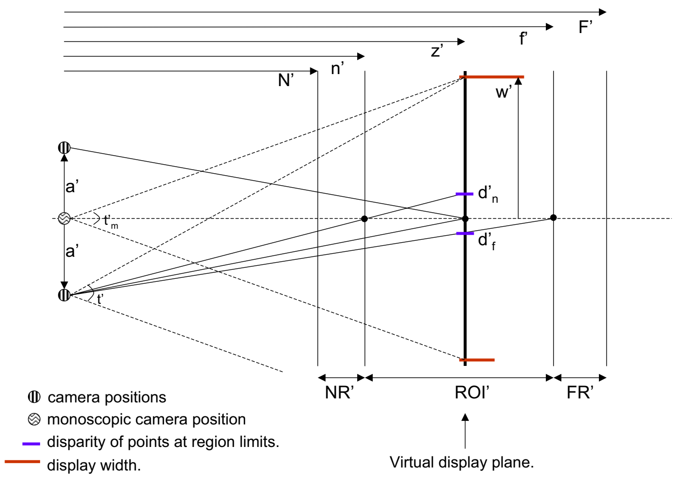

**Figure 5.** The scene geometry showing defined near, ROI and far regions of scene depth and the monoscopic camera position.

$$d_f = \frac{e(f-z)}{f} \tag{2}$$

Considering the geometry in the scene in figure 5 we can derive by similar triangles that the following relationships hold:

$$d'_n = \frac{a'(z'-n')}{n'} \tag{3}$$

$$d'_f = \frac{a'(f'-z')}{f'} \tag{4}$$

The link between these quantities is that the ratio of the disparities remains the same between the display and the virtual display in the scene.[1] Intuitively this is the case because the link between the scene and the display is the captured image which is the same except for a scale factor between the virtual display in scene space and the physical display. The ratio is:

$$r = \frac{d_n}{d_f} = \frac{d'_n}{d'_f} \tag{5}$$

As we are given $e, n, f$ and $z$ we can simply calculate the value of $r$. However, we do not know the distance to the virtual screen plane $z'$ and of particular interest the half camera separation $a'$. We can derive expressions for these by substituting (3) and (4) into (5) as below:

$$r = \frac{f'(z'-n')}{n'(f'-z')} \tag{6}$$

which we can rearrange to give an expression for $z'$:

$$z' = \frac{f'n' + f'n'r}{f' + n'r} \tag{7}$$

We can now find the virtual screen width $w'$, using $t'_m$ the monoscopic camera field of view.

$$w' = z' tan \left( \frac{t'_m}{2} \right) \tag{8}$$

From which we get the scaling, $s$, for disparities from the target display to the virtual screen.

$$s = \frac{w'}{w} \tag{9}$$

As we are given the target disparity ranges on the physical screen we can now calculate the disparity ranges on the virtual screen. Then by rearranging (3) and substituting $d'_n = sd_n$ we can find $a'$:

$$a' = \frac{sd_n n'}{z' - n'} \tag{10}$$

We now have almost all the information needed to calculate the left and right camera positions and generate the left and right partial images for the ROI region with the exception of the new field of view:

$$t' = 2arctan \left( \frac{w' + a'}{z'} \right) \tag{11}$$

This is the field of view for a camera with a symmetric frustum and hence we must clip a proportion of pixels from the left and right edges of the left and right partial images respectively. This ensures that points projecting zero disparity onto the virtual screen plane will also have zero disparity when displayed on the physical screen plane. The proportion of pixels to crop is given by:

$$c' = \frac{a'}{w' + a'} \tag{12}$$

### 3.4. Near Region Mapping

For the near region (NR) we map the scene depth in $[N', n']$ to the *gpd* $[N, n]$ using the same image plane used for the ROI mapping, as shown in figure 6.

We need to ensure that points on the plane at $n'$ map to the same position in the final image whether they are mapped by the ROI step or the NR step. We can consider this to be a constraint that the field width of the ROI camera and the NR camera be the same in the plane at distance $n'$ from the camera location. This will result in a piecewise continuous representation of stereoscopic depth which meets at region boundaries but may not be smoothly continuous.

For the NR mapping we calculate a new half camera separation $a''$, a symmetric field of view and the associated image cropping. Additionally we need to calculate an offset adjustment $o''$ to shift the NR disparity range to be continuous with the disparity range for the ROI region.

We first consider the disparity on the virtual screen plane of a point on the planes at the near region limits of $N'$ and $n'$ when projected from a camera at our new near region camera position $a''$.

The virtual screen disparity of a point on the far limit of the near region at $n'$ is given by:

$$d''_n = \frac{a''(z' - n')}{n'} \tag{13}$$

The virtual screen disparity of a point on the near limit of the near region at $N'$ is given by:

$$d''_N = \frac{a''(z' - N')}{N'} \tag{14}$$

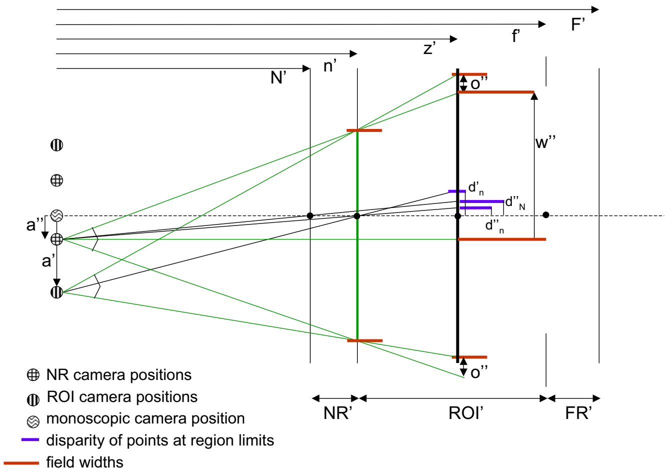

**Figure 6.** The scene geometry showing the variables related to the near region image generation.

We note that $d''_N - d''_n = s(d_N - d_n)$ since we define $[d_N, d_n]$ to be the target disparity range for the NR depth and hence, using equations(13) and (14), we find $a''$ to be:

$$a'' = \frac{s(d_N - d_n)}{\left(\frac{z'-N'}{N'}\right) - \left(\frac{z'-n'}{n'}\right)} \tag{15}$$

We now find the offset correction to the disparity on the virtual screen so that the near region disparity is continuous with the ROI region disparity in the virtual screen plane:

$$o'' = d'_n - d''_n \tag{16}$$

where $d''_n$ is given by (13). Equation (16) may be derived by inspection or by considering application of the intercept theorem[18] to the relevant geometry.

The field of view for the NR camera can now be calculated if we know the half field width $w''$ in the virtual screen plane which can be found as below:

$$w'' = w' - o'' + a'',\ o'' < a'' \qquad w'' = w' - a'' + o'',\ o'' >= a'' \tag{17}$$

The symmetric field of view for the left and right NR cameras is then:

$$t'' = 2atan\left(\frac{w''}{z'}\right) \tag{18}$$

There is a need to crop a proportion of pixels, $c''$, from the resulting images where

$$c'' = \frac{a'' - o''}{w''},\ o'' < a'' \qquad c'' = \frac{o'' - a''}{w''},\ o'' >= a'' \tag{19}$$

if $o'' < a''$ then we crop pixels from the left of the left image and the right of the right image while if $o'' >= a''$ then we crop pixels from the right of the left image and left of the right image.

We now have the new camera parameters and image adjustments we need to render the NR partial images for the left and right views. While the projection of the NR and ROI regions will differ in order to map depth differently from scene depth to available *gpd* for each region we have ensured the depth effect will be piecewise continuous at the region boundary.

### 3.5. Far Region Mapping

For the far region (FR) we need to map the scene depth in $[f', F']$ to the *gpd* range $[f, F]$ rendering onto the same image plane used for the ROI mapping, as shown in figure 7. The method is directly analogous to the NR algorithm described above and is derived in a similar manner.

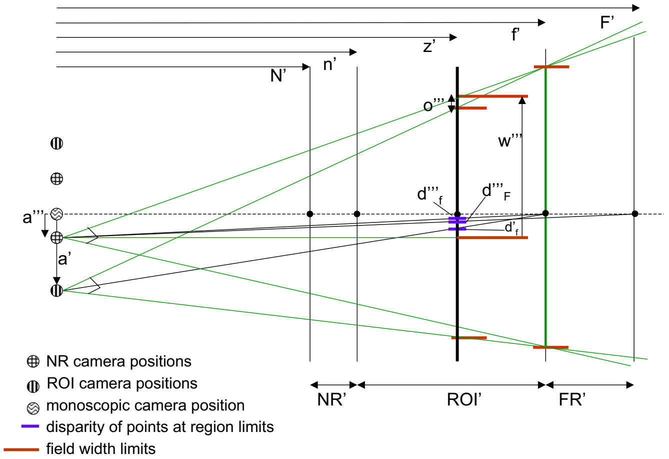

**Figure 7.** The scene geometry showing the parameters related to far region image generation.

As for the NR mapping we need to ensure that points on the plane at $f'$ map to the same position in the final image whether they are mapped by the ROI step or the FR step. We can consider this as a constraint that the FR and ROI cameras have the same field width in the plane $f'$.

We need to calculate a new camera separation $a'''$ that will map $[f', F']$ to $[f, F]$, calculate the symmetric field of view, associated cropping and finally calculate the offset needed to adjust the disparity range of the far region to be piecewise continuous with that of the far limit of the ROI disparity range.

We first determine the disparity on the virtual screen plane of a point on each of the planes at the far region limits of $f'$ and $F'$ when projected from a camera at our new far region camera position $a'''$ as below.

The virtual screen disparity of a point on the near limit of the far region at $f'$ is given by:

$$d_f''' = \frac{a'''(f' - z')}{f'} \tag{20}$$

The virtual screen disparity of a point on the far limit of the far region on the plane $F'$ is given by:

$$d_F''' = \frac{a'''(F' - z')}{F'} \tag{21}$$

We note that $d_F''' - d_f''' = s(d_F - d_f)$ by definition, since we are given $[d_f, d_F]$ as the target disparity range. Hence from (20) and (21) we find $a'''$ to be:

$$a''' = \frac{s(d_F - d_f)}{\left(\frac{F'-z'}{F'}\right) - \left(\frac{f'-z'}{f'}\right)} \tag{22}$$

We need to calculate the correction to the disparity on the virtual screen so that the far region is continuous with the ROI:

$$o''' = d'_f - d'''_f \tag{23}$$

Where $d'''_f$ is given by (20). Equation (23) may be derived by inspection or by application of the intercept theorem[18] to the relevant geometry.

The field of view for the FR camera location can be calculated if we know the field width $w'''$

$$w''' = w' + o''' + a''' \tag{24}$$

The symmetric field of view for the FR camera is then

$$t''' = 2atan\left(\frac{w'''}{z'}\right) \tag{25}$$

There is then the need to adjust the zero disparity plane by cropping a proportion of pixels from the left of the left image and the right of the right image as below:

$$c''' = \frac{a''' + o'''}{w'''} \tag{26}$$

We now have the new camera parameters and image adjustments we need to render the FR partial images for the left and right views. While the projection of the FR and ROI regions will differ in order to map depth differently from scene to display in each region we have ensured the depth effect will be piecewise continuous at the region boundary.

## 4. IMPLEMENTATION

The three-region mapping algorithm described above has been implemented using the POV-Ray ray tracing system to generate images of test scenes and application examples.

To use the POV-ray scripts the user defines a monoscopic camera position and field of view to frame the view they wish to generate. They provide details of the target display including the proportion of the total available *gpd* to be used for the FR, ROI and NR regions. In addition the total scene depth and the proportions to be mapped to the FR, ROI and NR regions are specified.

Given the above information a script is run to analyse the depth range in the scene and calculate the stereoscopic camera parameters for the three regions. Note, that when rendering the NR, ROI and FR regions we only use the scene geometry within each region so the script also calculates appropriate clipping volumes. The resulting scripts are then run to generate three partial images for both the left and the right views.

Finally the partial images are merged in depth order from far to near resulting in the stereo image pair which can then be displayed on the target 3D display.

## 5. RESULTS

The POV-ray results for a simple test scene are shown in figure 8. This shows the three regions rendered as partial images and the stereo pair resulting from merging the partial images in depth order.

A comparison of the results of the new three-region mapping algorithm with an existing single region algorithm is shown in figure 9. The upper graph in figure 9 shows how the single range mapping compresses the region of interest along with the rest of the scene as a whole into the available *gpd*. The lower graph shows how the three-region algorithm is able to distribute available *gpd* preferentially to the region of interest, in this case the ROI is given a one-to-one representation at the expense of further compression of *gpd* in the near and far regions.

As well as preferentially allocating available perceived depth to the region of interest the new algorithm has the benefit of being able to fix the *gpd* of the ROI even if the total depth range in the scene is changing. This seems likely to be of significant benefit when, for example, moving a game character around a scene with varying depth range. The depth representation of the game character can now be fixed where with the previous single range mapping the character's depth representation will vary depending on the total scene depth visible from the current viewpoint.

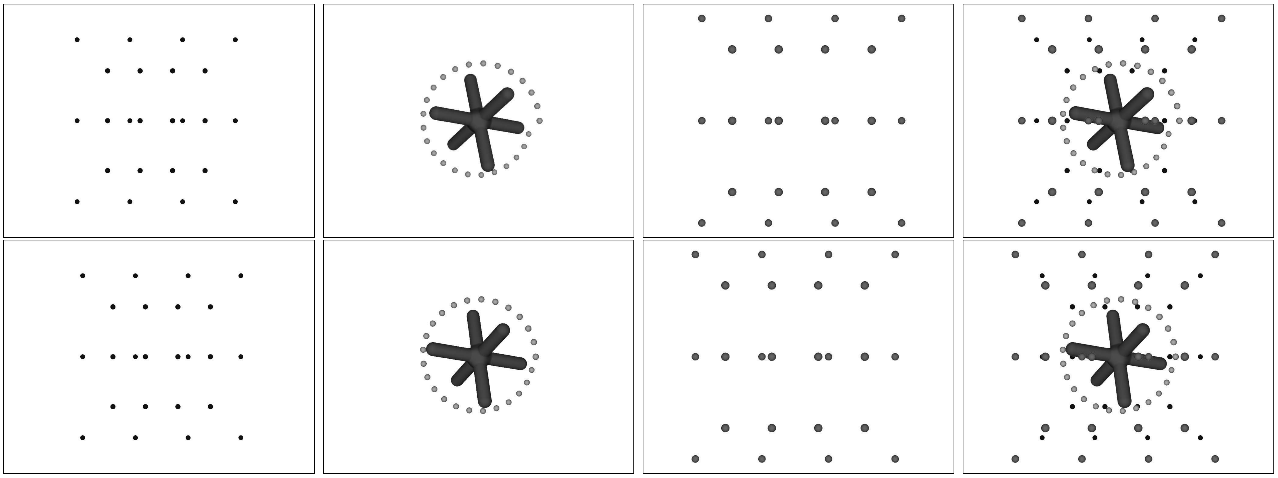

**Figure 8.** The partial images from left to right for the FR, ROI and NR regions are combined in depth order to form the stereo pair for this view shown at the far left. The original images are available on the author's web site.

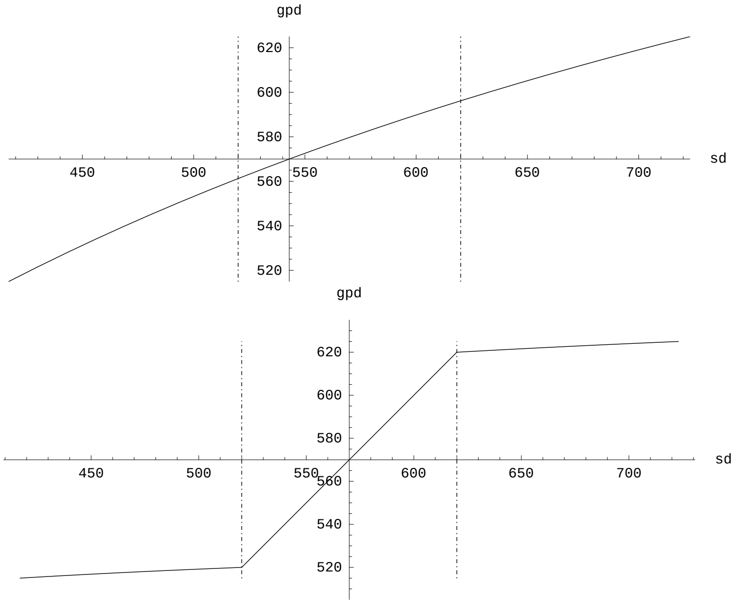

**Figure 9.** The result of applying an existing single-region mapping (upper) and the new three-region algorithm (lower). The scene depth, *sd*, along the abscissa is plotted against the geometric perceived depth, *gpd*, along the ordinate. The vertical dashed lines indicate the boundary of the region of interest.

## 6. CONCLUSIONS

This paper has presented a new approach to capturing scene depth in stereoscopic images solving problems that existing methods have representing depth in regions of interest. The new approach allows content creators to distribute available perceived depth preferentially to regions of interest and keep this mapping constant even if total scene depth is changing. We presented a new three-region algorithm to achieve this and have demonstrated results from analysis of the algorithm and a multi-pass rendering implementation using the POV-ray ray tracing system.

This approach will be important in animated content such as computer games where a game character moves around a scene and the total depth in view of the cameras is changing. It also has benefits for applications where depth judgement is critical, such as scientific and medical imaging, where an identified region of interest can be

”locked-down” to maintain a defined perceived depth representation.

For future work the current approach can be implemented easily for real time graphics in the standard graphics pipeline. We are also extending the algorithm from three to multiple-region mapping and in the limit continuously variable mapping of scene depth to display depth. Finally we are interested to investigate how to combine the new algorithm with other methods for controlling perceived depth, for example blurring disparities outside the region of interest.

## ACKNOWLEDGMENTS

Thanks are due to the Department of Computer Science at the University of Durham for purchasing the initial 3D display hardware for this work.

## REFERENCES

1. G. Jones, D. Lee, N. Holliman, and D. Ezra, “Controlling perceived depth in stereoscopic images,” in *Stereoscopic Displays and Virtual Reality Systems VIII*, *Proceedings of SPIE* **4297A**, 2001.
2. Y. Yeh and L. Silverstein, “Limits of fusion and depth judgements in stereoscopic color displays,” *Human Factors* **1**(32), 1990.
3. N. Holliman, “3D Display Systems.” to appear; Handbook of Opto-electronics, IOP Press, Spring 2004, ISBN 0-7503-0646-7.
4. C. Ware, *Information visualization, perception for design*, Morgan Kaufmann, 1999. ISBN 0-1-55860-511-8.
5. L. Lipton, “Stereographics Developers Handbook.” Stereographics Corporation, 1997. http://www.stereographics.com/support/downloads_support/handbook.pdf.
6. D. Diner and D. Fender, *Human engineering in stereoscopic viewing devices*, Plenum Press, 1993. ISBN 0-306-44667-7.
7. D. McAllister, ed., *Stereo computer graphics and other true 3D technologies*, Princeton University Press, 1993. ISBN 0-691-08741-5.
8. R. Sand and A. Chiari, eds., *Stereoscopic Television: Standards, Technology and Signal Processing*, European Commision, Directorate General XIII-B, Brussels, 1998.
9. C. Ware and G. Franck, “Evaluating stereo and motion cues for visualizing information nets in three dimensions,” Tech. Rep. TR 94-082, University of New Brunswick, 1984.
10. R. Hubbold and D. Hancock, “Stereo display of nested 3d volume data using automatic tunnelling,” in *Stereoscopic Displays and Virtual Reality Systems VI*, *Proceedings of SPIE* **3639**, 1999.
11. H. Helmholtz, *Treatise on physiological optics*, Thoemmes Press, 1867. 1924 edition, reprinted 2000.
12. A. Woods, T. Docherty, and R. Koch, “Image distortions in stereoscopic video systems,” *Proceedings of SPIE* **1915**, 1993.
13. L. Hodges and E. Davies, “Geometric considerations for stereoscopic virtual environments,” *Presence* **2**(1), 1993.
14. Z. Wartell, *Stereoscopic Head-Tracked Displays: Analysis and Development of Display Algorithms.* PhD thesis, Georgia Institute of Technology, 2001.
15. Z. Wartell, L. Hodges, and W. Ribarsky, “An analytic comparison of alpha-false eye separation, image scaling and image shifting in stereoscopic displays,” *IEEE Transactions on Visualization and Computer Graphics* **8**(2), pp. 129–143, 2002.
16. *PC-RD3D Series Operation Manual*, Sharp Corporation, Japan, 2003.
17. G.R.Jones, D.Lee, and N.S.Holliman, “Method of producing a stereoscopic image.” EP 1089573, April 2001.
18. J. Harris and H. Stocker, *Handbook of mathematics and computational science*, Springer-Verlag, 1998. ISBN 0-387-94746-9.