IPT-EGVE Symposium (2007), pp. 1–9
B. Fröhlich, R. Blach, and R. van Liere (Editors)

Three Extensions to Subtractive Crosstalk Reduction

F. A. Smit[1] and R. van Liere[1] and B. Fröhlich[2]

[1]Center for Mathematics and Computer Science (CWI), Amsterdam
[2]Bauhaus Universität Weimar

Abstract

Stereo displays suffer from crosstalk, an effect that reduces or even inhibits the viewer's ability to correctly perceive depth. Previous software crosstalk reduction methods reduce the amount of visible crosstalk by subtracting intensity from the displayed pixels. However, as these methods operate on each RGB color channel independently, they fail to reduce crosstalk adequately when the intensity of a color channel is too low to subtract from. What is needed is a crosstalk reduction algorithm that can be applied more generally than previous methods, for example for scenes with colored textures, various shades in background color and many colored objects causing crosstalk onto eachother.
In this paper, three extensions are introduced for software crosstalk reduction. First, we propose a reduction method that operates in the CIELAB color space and reduces crosstalk for all color channels simultaneously. Second, we introduce a geometry based reduction method that operates on fused 3D pixels. Finally, a run-time optimization is introduced that avoids the need to process each pixel. In this way, we show an improved real-time software crosstalk reduction method, applicable to a wider range of scenes, providing better quality and more flexibility.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism Virtual Reality I.3.3 [Computer Graphics]: Picture/Image Generation Display Algorithms

1. Introduction

Stereoscopic display systems allow the user to see three dimensional images in virtual environments. For active stereo with CRT monitors, Liquid Crystal Shutter (LCS) glasses are used in combination with a frame sequential display of left and right images. When the left image is displayed the right eye cell of the LCS glasses goes opaque and the left eye cell becomes clear, and vice versa.

Stereo systems suffer from a disturbing effect called crosstalk or ghosting that reduces, or even inhibits, the viewer's ability to correctly perceive depth [YS90]. Crosstalk occurs when one eye receives a stimulus which was intended for the other eye. Three main sources of crosstalk can be identified: phosphor afterglow, LCS leakage and LCS timing [WT02]. The effect of crosstalk is that it produces a visible shadow on the image. This is most noticeable at high contrast boundaries with large disparities.

To enhance depth perception we want to eliminate or reduce the effect of crosstalk. One way to achieve this is to reduce the effect of crosstalk in software by processing and adjusting the image frames that are to be displayed. The governing idea is to subtract an amount of intensity from each pixel in the displayed image to compensate for the leakage of intensity from the previous video frame. We call this method *subtractive crosstalk reduction*.

A pre-condition to subtractive crosstalk reduction is that the displayed pixels have enough initial intensity to subtract from. All previous subtractive methods (eg. [LW94] [KLD00] [KFNN03] [SvLF07]) operate in the RGB color space, and on each of the red, green and blue color channels entirely independently. When the estimated amount of leakage for one of the three color channels is larger than the desired display intensity for that color channel, the best those subtractive reduction methods can do is to set the corresponding color channel to zero. Therefore, previous subtractive reduction methods are unable to reduce crosstalk between regions consisting of different colors, for example a green object on a red background. The pixel regions where this is the case are said to be uncorrectable.

We propose an optimized crosstalk reduction algorithm that can be applied more generally than previous methods, for example for scenes with colored textures, various shades in background color and many colored objects causing crosstalk onto eachother. In this paper, our contribution is threefold:

- CIELAB color space reduction for uncorrectable regions. In the cases where standard subtractive crosstalk reduction fails we convert the pixel from RGB to CIELAB color space and try to find a perceptually closer match by altering all color channels simultaneously instead of simply setting a color channel to zero. In this way, we are able to reduce some amount of visible crosstalk in uncorrectable regions where previous subtractive methods could not.
- Geometry based reduction for uncorrectable regions. For a 3D object the apparent brightness of a point is caused by a combination of the fused pixels in the left and right eye. Instead of only subtracting intensity from the corresponding pixel with the same screen coordinates, we can also subtract intensity from the corresponding fused pixel at a distance depending on the 3D point's disparity.
- A run-time performance optimization for subtractive crosstalk reduction. Previous reduction algorithms have to be performed for each pixel. However, by changing the assumption of what constitutes a crosstalk-free image, we will show that a quick exit test is sufficient for the majority of pixels. This method reduces the visible shadows due to crosstalk, but does not reduce the overall increase in brightness.

The effect of these contributions is an improved real-time software crosstalk reduction method, applicable to a wider range of scenes, providing better quality and more flexibility.

2. Related Work

Woods and Tan [WT02] studied the various causes and characteristics of crosstalk. They showed that most CRT display devices use phosphors with very similar characteristics, such as spectral response and decay times. They state that phosphor afterglow and the LCS shutter glasses are about equal contributors to crosstalk. Therefore, the crosstalk problem can not be solved solely by using fast-phosphor display hardware and software crosstalk reduction remains necessary.

In the past, a number of subtractive crosstalk reduction methods have been proposed:

- To the best of our knowledge, the first subtractive crosstalk reduction algorithm was proposed by Lipscomb and Wooten [LW94]. The display area is divided into horizontal bands to accommodate for the non-linearity of crosstalk intensity over the display. Each of the bands undergoes subtractive crosstalk reduction according to a specifically constructed function. There is no user-calibration for the reduction model. To allow for some amount of reduction in uncorrectable regions the display intensity is artificially increased. This global increase of intensity causes a significant loss of contrast.
- Konrad et al. [KLD00] proposed a subtractive reduction method that is user calibrated. However, they assume crosstalk is constant over the display area, which causes the algorithm to fail for the bottom part of the display. Optionally, a contrast reducing mapping of display intensity from $[0,1]$ to $[\alpha,1]$ is used to allow for some reduction in uncorrectable regions.
- Klimenko et al. [KFNN03] implemented a real-time subtractive crosstalk reduction method for passive stereo systems. The method is based on the one proposed by Lipscomb and Wooten [LW94], and therefore suffers from the same disadvantages with respect to calibration and the inability to perform reduction in uncorrectable regions without a significant loss of contrast.
- A user-calibrated, subtractive reduction method that is not constant over the display area was proposed by Smit et al. [SvLF07]. Since crosstalk is not assumed to be constant over the display area, this method succeeds in reducing crosstalk over the entire display. However, due to the subtractive nature of the reduction algorithm it suffers from the same disadvantages for uncorrectable regions as mentioned earlier.

All of these methods operate in the RGB color space, and reduce crosstalk for each of the red, green and blue color channels independently. Therefore, all of them fail to reduce crosstalk adequately for regions consisting of different colors. Our method operates in the CIELAB color space and reduces crosstalk for all color channels simultaneously. In this way, it provides better reduction for these otherwise uncorrectable regions.

A related approach in the context of anaglyphic stereo images was shown by Sanders and McAllister [SM03]. The desired left and right eye pixel colors are converted to CIELAB space in order to find the single color to be displayed in the anaglyph. In this way, an attempt is made to minimize the perceptual difference between the desired colors and the observed colors through the color filter glasses. Altough this method is similar to ours in the use of the CIELAB space to find a perceptually close match, it is used in a different context.

To evaluate the perceptual quality of crosstalk reduction, Smit et al. [SvLF07] used the Visible Differences Predictor (VDP) by Daly [Dal93]. The VDP takes two images as input and produces a per-pixel probability map of perceivable difference. The algorithm operates using a frequency domain weighting with human contrast sensitivity function, followed by a series of detection mechanisms based on the human visual system. Digital photographs taken through the LCS shutter glasses are compared in this manner to evaluate

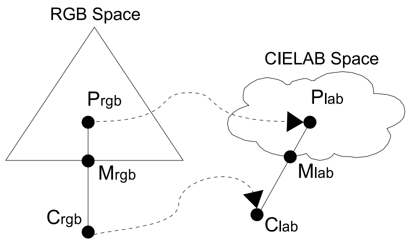

Figure 1: *P_{rgb} represents an RGB pixel to be displayed. After RGB crosstalk reduction has been applied the desired display pixel C_{rgb} is found to be outside the displayable RGB space. RGB reduction methods now display the closest match M_{rgb} in RGB space. However, with CIELAB reduction we first convert P_{rgb} and C_{rgb} to CIELAB space, giving P_{lab} and C_{lab}. Now it is often possible to find a better perceptual match M_{lab} than M_{rgb} is to C_{rgb}.*

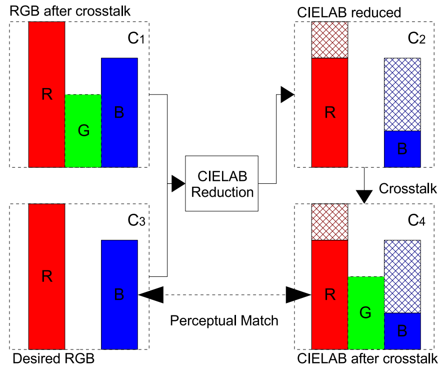

Figure 2: *The desired display pixel is shown as C_3, while the perceived pixel after green crosstalk is added is shown as C_1. Previous subtractive reduction methods would simply display C_3 as the green channel is zero and can't be subtracted from. This results in a perceived pixel C_1. However, CIELAB reduction tries to find display values (C_2) that, after crosstalk is added (C_4), are perceptually closer to the desired pixel C_3 than C_1 would be.*

the perceptual quality of crosstalk reduction. In this paper we will follow the same approach.

3. Methods

3.1. CIELAB Color Space Reduction

Subtractive reduction methods try to eliminate the visible crosstalk by estimating the amount of intensity leakage between left and right frames and subtracting this amount from the displayed intensities. In this way, the crosstalk will cancel out against the darkened regions rendering it invisible. The estimation and subtraction procedure is performed in the RGB color space, for each of the red, green and blue color channels independently.

Whenever the estimated crosstalk for any of these channels is larger than the desired display intensity for that channel, the method is unable to eliminate all of the crosstalk completely due to the otherwise resulting negative pixels values. For these uncorrectable regions, all previous reduction methods simply clamp the respective color channels to zero.

Our method is focused on providing better crosstalk reduction in uncorrectable regions. First, we start by performing normal crosstalk reduction and isolate the uncorrectable regions where negative pixel values would result. It is known from the calibration tables how much crosstalk can be eliminated given the desired display intensity. Whenever the display intensity of a pixel is not sufficient for any color channel to perform complete reduction, we mark the pixel as uncorrectable and apply our extended algorithm.

Instead of setting the uncorrectable color channel to zero and leaving the others unaffected we try to modify all color channels to get a perceptually closer match to the desired intensity when crosstalk is added implicitly. To do this, we convert the RGB values of pixel $P^{rgb}_{desired}$ to the perceptually uniform CIELAB color space giving $P^{lab}_{desired}$. The conversion is based on the assumption of sRGB phosphors and a D65 illuminant. Also, we estimate the amount of uncorrected crosstalk and add this to the pixel's RGB values giving $P^{rgb}_{crosstalk}$, after which these resulting values are also converted to CIELAB color space. This gives us two pixels in CIELAB space: $P^{lab}_{desired}$ corresponding to the desired pixel color, and $P^{lab}_{crosstalk}$ corresponding to the pixel after crosstalk is added. This idea is illustrated in Figure 1.

Next, we can estimate the uncorrectable increase in lightness by looking at the L-channels of $P^{lab}_{crosstalk}$ and $P^{lab}_{desired}$. Once this is known, we can subtract a similar amount of lightness from $P^{lab}_{desired}$ in such a way that after the implicit addition of crosstalk the pixel will be perceptually closer to the desired intensity than had we only set one color channel to zero. Thus, by altering all color channels via CIELAB space our method is able to find a perceptually closer best-match when one or more color channels are found to be uncorrectable. This procedure is shown schematically in Figure 2.

After the correct CIELAB pixel values are found, they are converted back to normal RGB space and replace the original pixel values. When the resulting pixel is darker than before it will also cause less crosstalk and this has to be compensated for accordingly. Therefore, the normal crosstalk reduction algorithm is performed again to eliminate any artifacts that might otherwise result from changed color channels in the pixel.

All of the above methods can be implemented entirely on

GPU hardware in the pixel shader. The procedure to map pixels between RGB and CIELAB space is costly, however this need only be done for uncorrectable pixels. Therefore, the complete, extended algorithm still runs in real-time on modern video hardware (eg. the NVidia G80 series).

3.2. Geometry Based Reduction

Previous subtractive crosstalk reduction methods are purely 2D pixel based, that is for every pixel the corresponding pixel with the same 2D coordinates is looked up in the display texture for the other eye to perform crosstalk reduction. However, when a user looks at a three dimensional object on a stereo display, pixels with different coordinates are fused together depending on the disparity, which in turn depends on the depth of the 3D point.

The governing idea of geometry based reduction is to subtract a small amount of intensity from the corresponding fused pixel in uncorrectable regions. This pixel will have different coordinates based on the disparity. Now when the user fuses these two pixels into a 3D image the combined observed intensity will be slightly lower, allowing us to reduce for some additional crosstalk.

The first step is finding the coordinates of the corresponding fused pixel. To do this, we need to know the actual depth of the current pixel. This information is present in the depth buffer in the form of the Z-component of a projected homogeneous vector. However, what we need is the actual depth from the camera so we can determine the disparity. The camera projection matrix contains all the information required to invert the projection and calculate the actual pixel depth P_{depth} from the depth buffer alone. This results in the expression:

$$P_{depth} = \frac{C_{far} \cdot C_{near}}{C_{far} - (P_{z/w} \cdot (C_{far} - C_{near}))}$$

where C_{far} and C_{near} are the camera far and near plane respectively, and $P_{z/w}$ is the pixel depth buffer value.

Next, we need to determine the disparity P_{disp} corresponding to the found pixel depth P_{depth}. This is given by:

$$P_{disp} = \frac{C_{eyesep} \cdot (C_{focal} - P_{depth})}{P_{depth} \cdot C_{width}}$$

where C_{eyesep} and C_{focal} are the camera eye separation and focal plane distance, and C_{width} is the width of the display area at the focal plane. The division by C_{width} maps the focal plane coordinates into a $[0,1]$ range on the near plane. The disparity depends only on the pixel depth and is independent from its position. This is illustrated in Figure 3. Note that P_{disp} switches sign when $P_{depth} > C_{focal}$, so both crossed and uncrossed disparity are handled correctly.

Once the disparity is known, we can find the 2D coordinates of the fused pixel by a simple addition. We then determine the actual depth for this fused pixel as before by making use of the depth buffer. If the found depth values are not equal we run into a case of occlusion between the two eyes and the algorithm takes an early exit. If the depth values match we have found the pixel that is fused with the current pixel for 3D viewing.

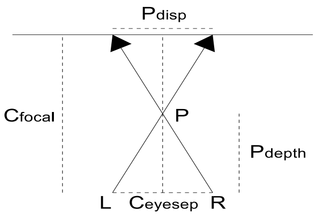

Figure 3: *P represents a 3D point at depth P_{depth} viewed in stereo with left eye L and right eye R. The parameters C_{focal} for focal length, and C_{eyesep} for eye separation are constants defined by the 3D camera. The disparity on the focal plane between the two projections is given by P_{disp}. An expression for P_{disp} follows quickly from the equality $\frac{C_{eyesep}}{P_{depth}} = \frac{P_{disp}}{C_{focal} - P_{depth}}$ due to similar triangles. Note that this equality holds regardless of the position of P.*

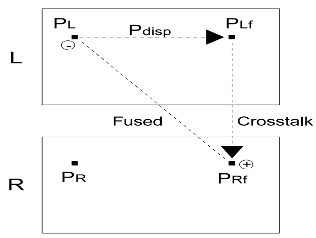

Figure 4: *P_L represents the current pixel to be rendered for the left eye. Given the disparity P_{disp} we can find the pixel P_{Rf} in the right eye that this pixel will be fused with. P_{Lf} has the same coordinates as P_{Rf} and increases the intensity of P_{Rf} due to crosstalk. Since the user will fuse together P_L and P_{Rf} we can reduce the intensity of P_L to compensate.*

Next, we need to subtract intensity from the fused pixel. However, current video hardware is limited by an inability to perform random writes. Therefore, we can not find the corresponding fused pixel for the current pixel and lower the fused pixels intensity, but rather we must work in reverse. First, for the current pixel, we find the corresponding fused pixel in the other eye and determine if that pixel receives uncorrectable crosstalk. If so, we can lower the intensity of the current pixel to compensate. This is illustrated in Figure 4.

The algorithm can be implemented entirely on GPU hardware, and can operate independently of the application geometry as a post-processing step, as long as a proper depth buffer is provided.

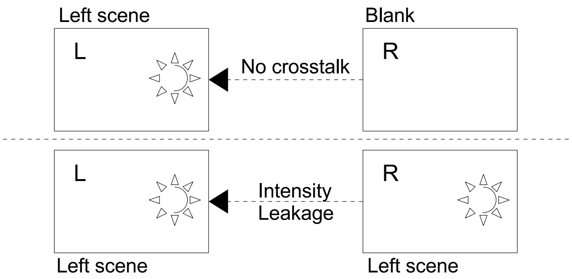

Figure 5: *(Top) Previous methods assume the left scene is crosstalk-free when the right eye is kept blank. This is called on-off reduction. (Bottom) Alternatively, there will be no visible shadows when the right eye displays the exact same scene as the left eye. Since left and right pixels with the same coordinates have equal values, there is only an increase in overall brightness. The benefit is that pixels with equal left and right values need not be reduced in intensity. This is called on-on reduction.*

3.3. Run-time Optimized Reduction

Current crosstalk reduction algorithms must be performed for each pixel in the display and are complex and time consuming. However, for a large number of pixels there is no disturbing crosstalk present in the form of visible shadows. Therefore, we would like a way to quickly determine whether a pixel shows disturbing crosstalk or not, and only process it when this is the case. From the calibration data we can determine at runtime which pixels cause visible shadows, however there is a problem when we only process these pixels.

An image is assumed to be crosstalk-free when the image is displayed as normal for one eye and completely black for the other eye. In this way, no leakage of light from one eye to the other is possible, as nothing is displayed for the other eye. The crosstalk reduction calibration tables are constructed according to this assumption. This also means that even when only a constant background color is displayed for both eyes, there will be a non-linear increase in brightness over the entire display area.

Crosstalk reduction algorithms try to reduce this increase in brightness for every pixel, in effect darkening the entire image to compensate for global crosstalk. When only the pixels that cause visible shadows are processed, those will appear darker than the unprocessed background due to the overall increase in brightness. Therefore, crosstalk reduction must be performed for each pixel, even when the pixel values in both eyes are equal.

A way to solve this is to change our assumption of a crosstalk-free image. The idea is to still eliminate the crosstalk shadows, but to allow a global increase in brightness. The previous assumption was that the left eye image is crosstalk-free when the right eye image is black. We call this *on-off reduction*. However, we can also make the assumption that the left eye image is crosstalk-free when the right eye image is equal for every pixel. This we call *on-on reduction*. For on-off reduction the calibration procedure uses a reference image where the left eye is displayed as normal and the right eye is kept blank. However, when calibrating an on-on reduction method the left eye is displayed as normal and the right eye also displays the same left image. This is illustrated in Figure 5.

4. Results

In this section we describe experimental results and compare various crosstalk reduction methods. The methods in this paper have been applied to a desktop virtual environment that consists of a head tracked user behind a 22 inch Iiyama Vision Master Pro 450 CRT monitor using active stereo. The display resolution is 1280x1024 pixels at a 120Hz refresh rate (2x 60Hz).

As the basis for our extensions we use the crosstalk reduction algorithm proposed by Smit et al. [SvLF07]. This algorithm is implemented on the GPU using frame buffer objects and calibration textures. For every pair of left and right pixels the amount of crosstalk is determined from the calibration data and reduced for accordingly. Our extensions are implemented on top of this on the GPU. In this way, interactive frame rates are achieved at a 1280x1024 resolution using an NVidia Quadro FX 3450 graphics board. Further implementational details can be found in [SvLF07].

As crosstalk is only visible to an external viewer we acquired result data by taking photographs of the display through the shutters. In this way, a Canon A510 digital camera was fixed in front of activated NuVision LCS glasses, taking photographs of the CRT display. This approach is equivalent to the one used by Smit et al. [SvLF07]. All photographs are taken through the left eye of the shutter glasses, and are compared to a crosstalk-free reference photo where the right eye is kept blank (also see Figure 5).

4.1. CIELAB Reduction Results

For the evaluation of the CIELAB reduction compared to subtractive RGB reduction we use a scene that shows a very general case of color combinations. Six shaded wire cubes with combinations of primary colors are positioned in front of the focal plane. A checkerboard of primary colors is located behind the focal plane. Again we took three photographs of this scene: one without reduction, one with RGB subtractive reduction and one with CIELAB reduction, and compared those to a fourth crosstalk-free reference photo. This is shown in the top row of Figure 6.

The bottom row of Figure 6 shows the output of the Visible Differences Predictor (VDP) [Dal93] comparisons. The

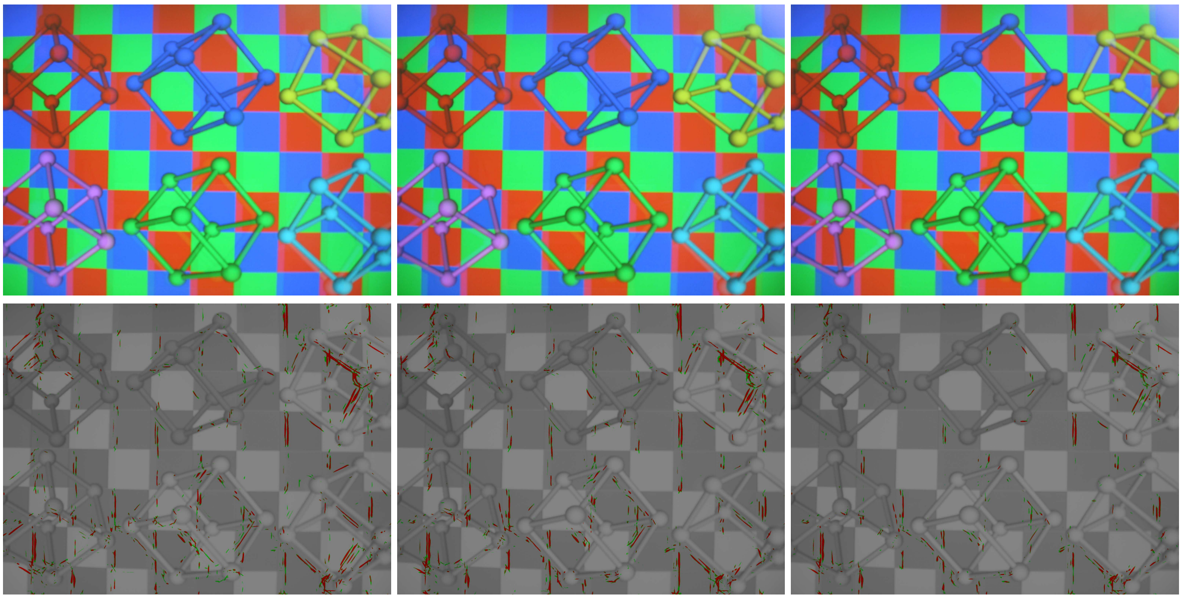

Figure 6: *(Top) From left to right photographs with no crosstalk reduction, normal subtractive crosstalk reduction and CIELAB reduction are shown. (Bottom) The VDP difference outputs between the reference and corresponding images on top.*

	VDP % Different	Average ΔE
Unreduced	3.5	274.4
RGB reduced	3.35	216.7
CIELAB reduced	2.12	178.5
CIELAB vs RGB	36.7%	17.6%

Table 1: *The first column shows the percentage of perceptually different pixels according to the VDP. The second column shows the average ΔE for the pixels found perceptually different with CIELAB reduction. Finally, the bottom row shows the percentage of improvement between CIELAB and RGB reduction.*

green pixels indicate a perceptual difference with a probability over 75%, while the red pixels are perceptually different with a probability over 95%. It can be seen that RGB reduction performs only slightly better than no reduction at all due to the many uncorrectable regions. However, CIELAB reduction performs much better, reducing the amount of perceptually different pixels according to the VDP by as much as 36.7% compared to RGB reduction. This data is shown in Table 1.

Even when the VDP marks a pixel as perceptually different, there is still the question of how much difference we perceive. Therefore, for every pixel that was marked perceptually different by the VDP, we measured the difference between this pixel and the reference pixel in terms of CIELAB ΔE units. The ΔE is a standard measure of perceptual color difference. It is based on the distance between two points in the uniform CIELAB space, thereby providing a quantitative measure of the difference between two colors. As we are interested in the improvement of reduction, we plotted the percentual reduction of the average ΔE per scanline compared to normal crosstalk in Figure 7. Only pixels that were found perceptually different by the VDP are included in these calculations. Figure 7 and Table 1 show that even though perceptually different pixels remain after crosstalk reduction, the shadows are less noticeable in terms of ΔE differences. We also see that CIELAB reduction offers a 17.6% improvement over standard RGB reduction. Also, while the global improvement is 17.6%, for some local cases CIELAB reduction is shown to perform much better.

These results show that CIELAB reduction is able to reduce crosstalk in much more general cases than RGB reduction can. Also, even in cases where the crosstalk is still perceptually noticeable, it is much less noticeable than with RGB reduction. When the noticeability of the crosstalk shadows falls below a certain threshold, the users are not bothered by them as much and depth perception possibly remains unaffected.

4.2. On-on Reduction Results

To evaluate the quality of on-on reduction compared to on-off reduction we used a test scene with a uniform grey background. Three photographs with normal crosstalk, on-off reduction and on-on reduction were taken and compared to a fourth crosstalk free reference photo, as shown in Figure 8 and 9. As can be seen from Figure 8 the overall brightness

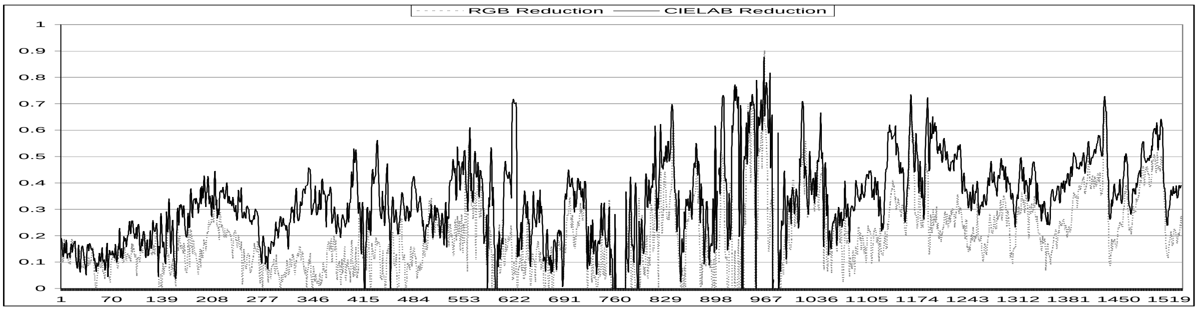

Figure 7: *This plot shows the percentage of improvement for RGB and CIELAB reduction over no reduction, for the average ΔE values per scanline for pixels that are found to be perceptually different after CIELAB reduction.*

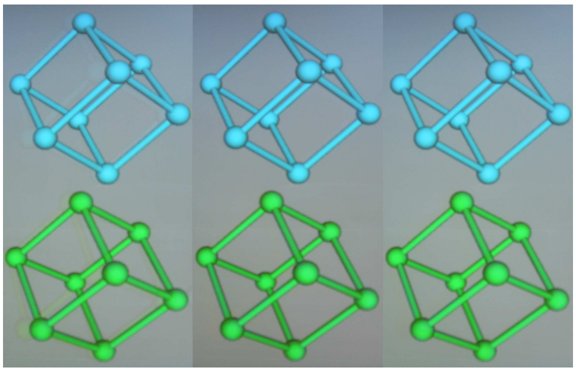

Figure 8: *From left to right photographs with no crosstalk reduction, on-off reduction and on-on reduction are shown. There is a noticeable increase in brightness for on-on reduction compared to on-off. The brightness for on-on is similar to the crosstalk image.*

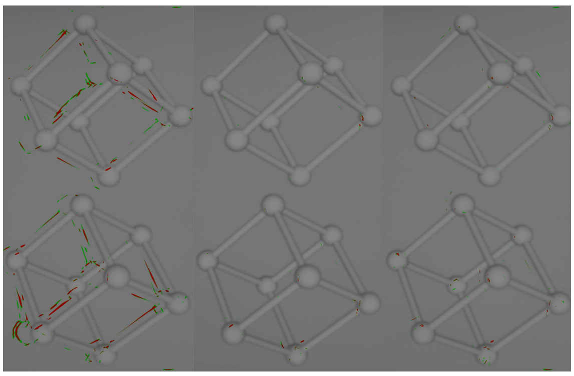

Figure 9: *Shown from left to right are the VDP outputs between the reference and no crosstalk reduction, on-off reduction and on-on reduction. It can be seen that both on-off and on-on remove the visible shadows which are present in the crosstalk image. Also, the increased brightness for on-on reduction is not perceptually disturbing.*

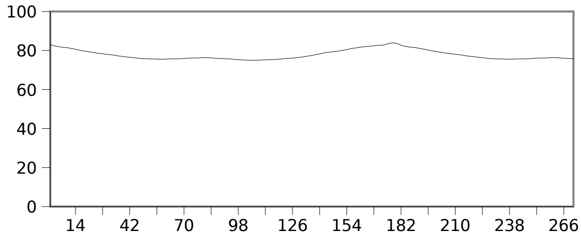

Figure 10: *For each frame of an animation sequence, this plot shows the percentage of pixels that do not require processing when using on-on reduction. The scene consisted of six wire cubes on a grey background, similar to Figure 8.*

for the on-on reduction is higher than for on-off reduction as expected. The brightness of the on-on reduction is similar to the crosstalk photo, however no visible shadows are present.

Figure 9 shows the VDP output of the comparisons to the reference photo as before. For the unreduced scene on the left the crosstalk shadows are clearly visible. Both the on-off and the on-on reduction algorithms manage to almost completely eliminate all of the crosstalk shadows. This shows that the on-off and on-on reduction methods perform equally well in removing visible crosstalk shadows. While the on-on reduction method causes an increased brightness, it still removes all crosstalk shadows that affect depth perception. The benefit of on-on reduction is that many pixels did not need to be processed, while on-off reduction had to be performed for each pixel. In this way, we avoid processing 77.8% of the pixels on average in our test scene, as is shown in Figure 10.

5. Discussion

As was shown earlier, crosstalk reduction in the CIELAB color space provides an improvement over subtractive reduction in the RGB color space by finding perceptually closer matching colors. This method might be improved upon by using different perceptual color spaces, such as CIECAM and different color matching procedures. As the CRT phosphor spectral responses are known, and reasonable models of the human visual system exist, it might be possible to find even better perceptual matches. However, finding a perceptually closest match under various constraints is closely related to gamut mapping, which is as of yet an open research topic. Second, even if we are able to find a theoretically better match this might not result in a large perceivable difference in practice.

The parameters used for the color matching procedure in CIELAB space were found through direct experimentation. It would be desirable to implement a user calibration routine for this, so the method can easily be adapted to different hardware and users. However, as the user calibration tables from the original method are used as a basis, this is not immediately necessary.

Even with CIELAB crosstalk reduction, it is not possible to completely eliminate visible crosstalk shadows in all cases. For example, when the background is black it is impossible to perform any reduction. However, this is an inherent problem to software crosstalk reduction and not a specific fault of our method. In many cases, for example scenes with colored textures, various shades in background color and many colored objects causing crosstalk onto eachother, our method provides an improvement over classic subtractive reduction methods. Even in the cases where we are unable to completely eliminate visible crosstalk, the method still provides a better, less noticeable alternative to previous methods. In practice we see that most, if not all, visible crosstalk can be eliminated when using colored backgrounds.

As the geometry based reduction is an effect that requires both eyes to fuse a 3D image, it was impossible to experimentally show and quantify this effect using a photographic camera. The fused intensity can be slightly reduced, allowing for more crosstalk reduction. However, the effect of displaying a different intensity in either eye causes some amount of jitter to be seen. Also, the darkening might be visible when the user is focussing at a different depth plane.

The on-on reduction method provides an optimized alternative to on-off reduction. The benefit of using on-on reduction is that whenever the left and right pixel values are found to be nearly equal, no reduction has to be performed and the algorithm can take an early exit by simply comparing pixel values. A side effect is that the overall brightness is increased, however this is not perceptually disturbing and the disturbing crosstalk shadows are still eliminated as before. Furthermore, depth perception is affected by the visible shadows, not the increase in brightness. Finally, the method provides a way to compare the effects of crosstalk shadows only, without introducing a difference in contrast. The on-on reduced scene will have the same apparent brightness as the non-reduced scene, however no crosstalk shadows will be visible.

6. Conclusion

We have introduced three extensions to subtractive crosstalk reduction. First, we have shown a reduction method that operates in the CIELAB color space and is therefore able to reduce crosstalk in uncorrectable regions where subtractive RGB methods failed. This resulted in a 36.7% improvement in reduction over previous methods. Also, in cases where the crosstalk was still perceivable, it was shown to be 17.6% less noticeable in terms of ΔE units. Second, we have shown a geometric reduction approach where the fusion of 3D pixels is taken into account to perform extra reduction. Finally, we have shown an optimized reduction method that prevents having to execute the reduction algorithm for each pixel, improving run-time performance. All of these methods were implemented entirely on the GPU hardware and run in real-time.

In many cases, for example scenes with colored textures, various shades in background color and many colored objects causing crosstalk onto eachother, these methods provide an improvement over classic subtractive RGB reduction methods. In this way, we have shown an improved real-time software crosstalk reduction method, applicable to a wider range of scenes, providing better quality and more flexibility.

References

[Dal93] DALY S.: The visible differences predictor: an algorithm for the assessment of image fidelity. In *Digital images and human vision*. 1993, pp. 179–206.

[KFNN03] KLIMENKO S., FROLOV P., NIKITINA L., NIKITIN I.: Crosstalk reduction in passive stereo-projection systems. In *Eurographics* (2003), pp. 235–240.

[KLD00] KONRAD J., LACOTTE B., DUBOIS E.: Cancellation of image crosstalk in time-sequential displays of stereoscopic video. In *IEEE Transactions on Image Processing, Vol. 9 No. 5* (2000), pp. 897–908.

[LW94] LIPSCOMB J. S., WOOTEN W. L.: Reducing crosstalk between stereoscopic views. In *Proc. SPIE Vol. 2177* (1994), pp. 92–96.

[SM03] SANDERS W., MCALLISTER D. F.: Producing anaglyphs from synthetic images. In *Proc. SPIE Vol. 5006* (2003), pp. 348–358.

[SvLF07] SMIT F., VAN LIERE R., FROEHLICH B.: Non-uniform crosstalk reduction for dynamic scenes. In *Proc. IEEE Virtual Reality, March 2007* (2007), pp. 139–146.

[WT02] WOODS A. J., TAN S. S.: Characteristic sources of ghosting in time-sequential stereoscopic video displays. In *Proc. SPIE Vol. 4660* (2002), pp. 66–77.

[YS90] YEH Y.-Y., SILVERSTEIN L. D.: Limits of fusion and depth judgment in stereoscopic color displays. *Hum. Factors 32*, 1 (1990), 45–60.