

3D-Enhancement: Reducing Cross-talk in Auto-stereoscopic Displays

Assaf Zomet, HumanEyes Technologies

cto@humaneyes.com

Joint work with Daniel Lau

Introduction

Auto-stereoscopic displays provide 3D visualization without requiring the user to wear glasses or other devices. Of special interest are auto-stereoscopic displays that use optics such as lenticular lenses and parallax barrier to direct a set of views to different angles. This type of displays is by far the most common in print as well as in digital displays such as computer monitors, mobile devices and recently also television sets.

Auto-stereoscopic displays are inherently limited by the range of depth they can present clearly. Figure 1 illustrates typical blurriness in auto-stereoscopic displays: Objects in depth appear blurry, with the extent of blur growing with depth. The result is degradation of the quality of the displayed image and the quality of overall depth perception.

We show that this depth-dependent blur can be substantially reduced using image processing, hence producing crisp-sharp images. Our enhancement technique is computationally efficient and does not require any understanding of the displayed content. Hence it is applicable to complex content including thin 3D structures, transparent objects, reflections and specularities. We have applied our method successfully to lenticular printing as well as to auto-stereoscopic monitors. A similar approach was applied successfully to reduce ghosting in lenticular prints of flip effects.



Figure 1

Depth dependent-blur in auto-stereoscopic displays. The further an object is from the display plane, the blurrier it appears on screen.

Reducing Cross-talk in Auto-stereoscopic 3D Displays

We explain our method by demonstrating it on lenticular displays.

Figure 2 shows a 1D illustration of the optics of lenticular lenses. A lenticular sheet consists of a set of elongated lenticular lenses (lenticules). The views for display are combined to create an interlaced image that is attached to the flat side of the lenticular sheet, for example by printing directly to the sheet, by laminating a printed media to sheet, or by attaching a digital display to the sheet. In an ideal optics, a viewer's eye sees one of the views by seeing via each lenticule a single column of the interlaced image corresponding to a column from the view.

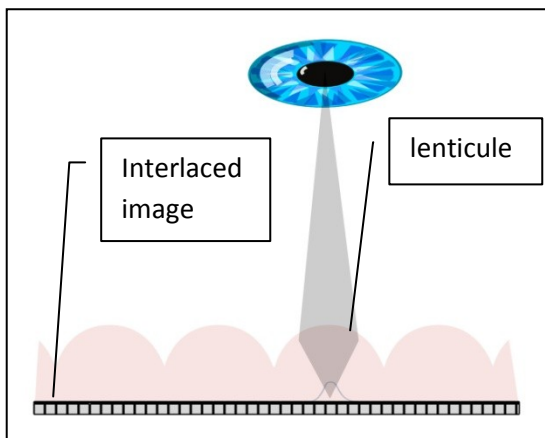


Figure 2

An ideal lenticular display/print. An interlaced image combines columns from different views. Each lenticular lens (lenticule) focuses the light coming from a column in the interlaced image to the eye of a viewer. In practice, imperfect optics results in cross-talk between the viewed images.

In practice, however, imperfections in the optics result in cross-talk: Rather than seeing a single column corresponding to a single view, the viewer will often see a mix of the views. This cross talk is mathematically modeled as:

$$U_i(x, y, c) = \sum_{j=1}^K A(j, i) V_j(x, y, c)$$

Equation 1

Where x, y denote spatial coordinates and c denotes a color channel, typically R,G,B.

In other words, when providing for display the views V_1, \dots, V_k , instead of ideally seeing views V_1, \dots, V_k , the viewers see views U_1, \dots, U_k which are mixtures of V_1, \dots, V_k , according to a mixing matrix A .

The basic idea of our method is to provide views W_1, \dots, W_k for display, given by:

$$\begin{bmatrix} W_1(x, y, c) \\ \vdots \\ W_k(x, y, c) \end{bmatrix} = A^{-1} \begin{bmatrix} V_1(x, y, c) \\ \vdots \\ V_k(x, y, c) \end{bmatrix}$$

Equation 2

where the operator A^{-1} denotes an approximate inverse to matrix A . The approximate inverse operator A^{-1} is designed to minimize the difference between the original views V_1, \dots, V_k and what's actually seen by the user according to the model, minimizing:

$$\left\| \begin{bmatrix} V_1(x, y, c) \\ \vdots \\ V_k(x, y, c) \end{bmatrix} - A \begin{bmatrix} W_1(x, y, c) \\ \vdots \\ W_k(x, y, c) \end{bmatrix} \right\| = \left\| \begin{bmatrix} V_1(x, y, c) \\ \vdots \\ V_k(x, y, c) \end{bmatrix} - AA^{-1} \begin{bmatrix} V_1(x, y, c) \\ \vdots \\ V_k(x, y, c) \end{bmatrix} \right\|$$

Equation 3

The design of approximate inverse operator A^{-1} is application specific, depending on the display constraints as well as expected inaccuracies in the model as expressed in Equation 2. One obvious display constraint is that the views W_1, \dots, W_k are limited to the valid values range, e.g. within the range 0..255 in 8-bit displays. A straight-forward solution is to define A^{-1} as the pseudo-inverse of matrix A , and clip the result to the valid range. To calibrate the matrix A , we assume with some displays that A is a Toeplitz matrix implementing a convolution with, for example, a 2-parameters kernel and test different kernel parameter sets to find the most visually-pleasing enhancement results.

We found, however, that with most 3D displays, setting A^{-1} to be a simple unsharp-mask operator followed by clipping the values to the range provides an effective result.

Representing unsharp mask operator using a matrix M :

$$M(i, j) = s\delta(i, j) - aG\left(\frac{i-j}{\sigma}\right)$$

Equation 4

Here G is a blur kernel, e.g. standard Gaussian filter with zero mean and std=1. The parameters a, σ are calibrated by testing for the most visually appealing enhancement result. Parameter 's' is set so as to have sum of 1 on the rows of the matrix.

Note that unlike conventional unsharp mask as used e.g. in Adobe® Photoshop®, this unsharp mask is applied across views and not within views. The whole process is illustrated in Figure 3. Example views are shown in Figure 4.

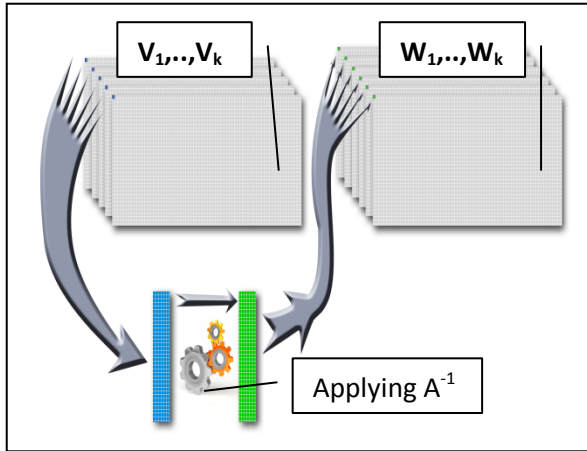


Figure 3

3D Enhancement: An inverse operator A^{-1} such as an unsharp mask filter is applied to matching pixels in the original views $V_{1,..,V_k}$, denoted in blue. The result, denoted in green, is placed in matching pixels in the result views $W_{1,..,W_k}$



Figure 4

3D enhancement: An example of a view W_j . The inverse operator introduces inverse-ghost elements in the images, the size and magnitude of which depending on the depth of the object

Inverse ghost of the bride's face

Deghosting: Reducing Ghosting of Lenticular Flip Effect

The reader may be familiar with printed cards the change from one image to another when tilted. This flip effect is typically implemented similarly to 3D auto-stereoscopic displays by creating an interlaced image combining two (or more) images and attaching it e.g. to a lenticular sheet. With ideal optics, each image should be seen clearly. In practice, however, in many cases a viewer will see a mix of the two images, as if a "ghost" of one image appears on top of the other image, as illustrated in Figure 5.



Figure 5

Ghosting: In flip effect, ideally a viewer should see at an angle one of the images clearly (left). In practice, due to cross-talk, the viewer will often see a "ghost" of one image over the other (right).

We found that using the method of 3D displays, as described for example in Equation 4 is in-effective with flip effect. In particular, it produces artifacts of inverse-ghosting, a ghost of the intensity-inverse of one view on top of the other view. This is a result of two factors: First, the cross-talk in front lit displays such as lenticular prints is a function of illumination homogeneity. In other words, the model defined in Equation 1 is not only view-dependent (A is shift-variant), but it also changes as a function of illumination environment. Second, this inaccuracy in the model leads to artifacts that are visible especially in flip effects. To this end, we have developed an illumination insensitive inverse operator for flips. This operator inserts separator views between the original views, the separator views including an inverse of the original views. A flip of two views A, B with intensity range 0..1, a separator S will be computed for example as follows:

$$S(x, y, c) = \alpha(1 - A(x, y, c)) + (1 - \alpha)B(x, y, c)$$

Equation 5

This is illustrated visually in Figure 6:

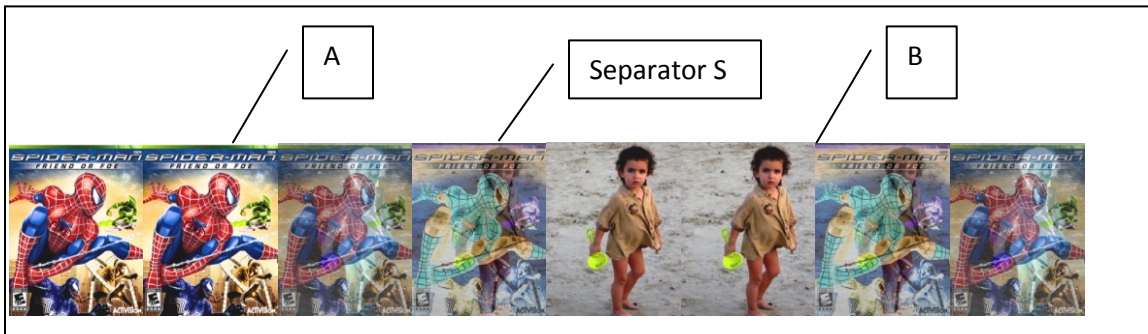


Figure 6

De-ghosting: The ghosting artifact is substantially reduced by adding separator views depicting an inverse version of the second image. For example, separator S above includes an inverse version of image A.

Using the illumination insensitive operator reduces ghosting substantially, but pushes the viewed colors to a grayish tint. To overcome this effect, we pre-process the views to boost their colors such that the colors in the views of the final print match the colors in the original views. Such color correction is less applicable in image areas with fully-saturated colors, leaving us with a tradeoff in these areas between reducing ghosting and maintaining color fidelity.

In order to calibrate for the amount of deghosting (parameter α) and the extent of color correction, we print a test flip with different pairs of parameters and choose the pair that provides best color fidelity and minimal ghosting.