https://doi.org/10.2352/ISSN.2470-1173.2019.14.COLOR-101

3D Tone-Dependent Fast Error Diffusion

Adam Michals, Jiayin Liu, Altyngul Jumabayeva, Zhi Li, Jan P. Allebach; Purdue University; West Lafayette, IN

Abstract

As 3D printing becomes more prevalent, more attention is being paid to its ability to adequately reproduce the appearance of surfaces. Research into methods to accurately represent grayscale images through halftoning is well-developed in 2D printing, but work in the halftoning of surfaces in 3D printing is less developed. The halftoning method of tone-dependent fast error diffusion has been shown in 2D printing to be an effective means of achieving both high image quality and computational efficiency, making it an ideal algorithm to run in printing units where computational power is restricted. This work seeks to adapt tone-dependent fast error diffusion to halftone the surfaces of three-dimensional objects.

Here, the ideal tone-dependent error diffusion parameters will be calculated for an image. Then, a surface traversal mechanism will be implemented to navigate the surface of a three-dimensional object while error diffusion is applied to halftone it. The expected result is an algorithm that can halftone the surface of an object with quality approaching that of iterative methods, with a fraction of the processing that they require. Iterative methods currently produce the highest halftoning quality for 3D surfaces, but their use is limited due to the amount of computation they entail. The development of a tone-dependent fast error diffusion algorithm allows 3D halftoning to represent a continuous-tone surface with comparably high quality, but the computation it requires is more appropriate for standard printers; thus, it improves on the quality of surface halftoning that most printing units can produce.

Introduction

In 2D halftoning, there are three main categories of algorithms: screening, search-based methods, and error diffusion. Screening methods use a simple threshold to convert gray pixels to black or white, which requires little computation but is limited in output quality. Search-based methods generally produce high quality images [1], but as is mentioned in [1], this quality comes at the cost of intensive computation because search-based methods like DBS are often iterative. Early error diffusion work such as that of Floyd and Steinberg [2] produced images with better quality than screening methods with less intensive computation than was the case in search-based methods, but several artifacts were generated that subtracted from the quality of the image. Since then, work on error diffusion has introduced ways to diminish the effects of these artifacts, and much of that has come in the form of modifying the characteristics of error diffusion throughout the process, such as varying weights, thresholds, and diffusion filter sizes, as well as varying scanning orders like Peano [3] and serpentine scanning [4], as opposed to standard raster scan.

Work using variable weights includes that of Wong [5], which used the LMS algorithm [6] to optimize the weights throughout the error diffusion process. Dithering the weight matrix to reduce artifacts has also been shown by Ulichney [7], [4] to be effective.

Variable thresholds have been used as well. Miller, Sullivan, and Pios used visual feedback to adjust the error diffusion thresholds [8]. Knox and Eschbach used threshold modulation [9]. Billotet-Hoffmann and Bryngdahl used a threshold matrix rather than a single threshold value to hinder artifacts' ability to form in any particular direction.

Work in varying diffusion filter sizes includes that of Eschbach, who used a more spread-out filter for extreme highlight and shadow regions and a tighter filter for midtones [10].

While Brunton et al. [11] and Zhou et al. [12] have adapted error diffusion to 3D, the same range of variable parameters has not been seen yet in 3D error diffusion. Other works by Lin et al. [13], Lou and Stucki [14], Kuipers et al. [15], and Cho et al. [16] consider 3D halftoning in relation to current 3D printing technologies.

This work looks at adapting error diffusion with variable weights and thresholds to 3D in a voxel-based printing format, which is described by Mao et al. in [17]. As tone-dependent error diffusion in 2D approached the quality of DBS-halftoned images [18], this work seeks to use the tone-dependent methods described in 2D works to develop a 3D error diffusion algorithm with high quality results.

Error Diffusion

Here, halftoning is considered for converting grayscale images to black and white pixels or voxels for printing purposes. Thus, absorptance is used to describe gray levels. An absorptance of 1 corresponds to black, the desired tone of ink applied to a page; an absorptance of 0 corresponds to white, the desired tone of paper where no ink has been applied. These definitions of absorptance are to extended to the colorants used in halftoning the surface of a 3D object.

In 2D error diffusion, a gray pixel is first converted to black or white based on a threshold gray level. Then, in order to preserve the local tone, the error between the original and halftoned pixels is subtracted from surrounding pixels that have yet to be halftoned.

Here, the input 2D image will be denoted by $x[m,n]$, the image as updated with diffused error by $u[m,n]$, the output halftoned image by $y[m,n]$, and the threshold applied to pixels by $t[m,n]$. So,

$$y[m,n] = \begin{cases} 1, & \text{if } u[m,n] \geq t[m,n] \\ 0, & \text{else} \end{cases} \tag{1}$$

The error $e[m,n]$ in binarizing a pixel is considered as

$$e[m,n] = y[m,n] - u[m,n] \tag{2}$$

Then, using weighting matrix $w[k,l]$, a pixel is updated as

$$u[m+k,n+l] \leftarrow u[m+k,n+l] - w[k,l] * e[m,n] \tag{3}$$

In error diffusion that scans pixels in raster order, error can be diffused according to the weights shown below in figure 1. Raster scan order would visit pixels from left to right, then from top to bottom.

	*	$w[0,1]$
$w[1,-1]$	$w[1,0]$	$w[1,1]$

Figure 1. *Error Diffusion filter for raster scan.*

Here, the blank space is the previous pixel, * is the current pixel, and the remaining weight terms are associated with unvisited pixels.

For example, in Floyd-Steinberg error diffusion [2], the weights $w[0,1], w[1,-1], w[1,0]$, and $w[1,1]$ are respectively $\frac{7}{16}, \frac{3}{16}, \frac{5}{16}$, and $\frac{1}{16}$. So, for raster scan, the pixel to the right of the pixel under operation would receive $\frac{7}{16}$ of the error from binarizing the pixel under operation, the pixel below the pixel under operation would receive $\frac{5}{16}$ of the error, and so on.

Serpentine scan order visits pixels from the top row of the image to the bottom row, and for successive rows alternates between moving from left to right and moving from right to left. For moving from left to right, the above filter orientation is used; for moving from right to left, the filter above is mirrored to diffuse error toward the next pixel and the next row to be visited.

Tone-Dependent Error Diffusion

In tone-dependent error diffusion, the thresholds and weights vary with the gray level of the input pixel, denoted by a. So, the thresholds and weights will become $t[m,n;a]$ and $w[k,l;a]$, respectively.

In [18], $t_u(a)$ and $t_l(a)$ are defined as the upper and lower thresholds for binarization at a given gray level. Gray pixels above the upper threshold become black, those below the lower threshold become white, and the halftone result for those in the middle is chosen from a 0.5 absorptance patch halftoned by DBS, $p[m,n;0.5]$. That is,

$$y[m,n] = \begin{cases} 1, & \text{if } u[m,n] \geq t_u(x[m,n]), \\ 0, & \text{if } u[m,n] \leq t_l(x[m,n]), \\ p[m,n;0.5], & \text{else} \end{cases} \tag{4}$$

In adapting tone-dependent error diffusion to 3D, this work will keep the upper and lower thresholds but change the halftone output for gray levels between the thresholds. Projecting a 2D DBS patch onto a 3D surface can distort its perceived tone, and while work has been done in creating 3D DBS programs [17], there is no 3D surface that can be considered representative of all others.

Thus, the halftone output for gray levels between the two thresholds will simply alternate between black and white. This mimics the average 0.5 absorptance of the DBS-halftoned patch but has a more reliable effect across a range of arbitrary 3D surfaces.

The work in [18] describes the process for optimizing the thresholds and weights for each gray level for each of raster scanning, serpentine scanning, and two-row serpentine scanning. Here, the 2D serpentine scan parameters will be used, as described in the section "Error Diffusion in 3D" later.

3D Analogs

The three-dimensional equivalent of a pixel is a voxel, which is a cube rather than a square. Objects in 3D will be composed of a number of these cubic voxels, much as a 2D image is composed of pixels. However, while all pixels in a 2D image are relevant, not all voxels will be. A cuboid workspace is defined around the 3D object of interest, so the empty space around the object is composed of what will be called exterior voxels. These voxels are used in calculations but are not operated on, since they have no tone.

As seen in figures 2 and 3, any 3D object will have some form of shell of surface voxels (red voxels in the figures) that encloses interior voxels (cyan). In the 3D halftoning considered in this work, the goal is to accurately represent the surface of a 3D object using black and white voxels. So, as with the exterior voxels, interior voxels are used in calculations but never actually operated on, since their tone is irrelevant. All halftoning is performed only on the surface voxels of the object.

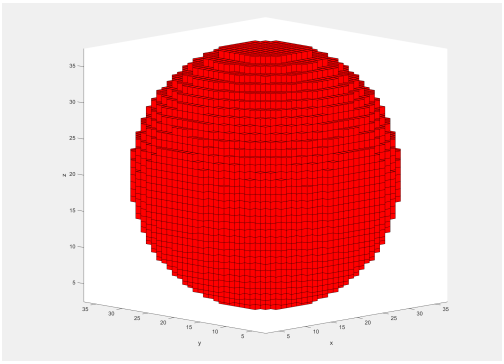


Figure 2. *A sphere, where red voxels are surface voxels and the white voxels represent empty space.*

For a 2D image, a number of standard scanning methods exist to guide the order in which pixels are visited. Of these, serpentine scan has been shown to be effective [4] in reducing the artifacts commonly associated with error diffusion. As will be described in the Surface Traversal section that follows, serpentine scan is mimicked by taking a 3D object in slices and rotating around the slices in alternating clockwise and counterclockwise directions.

Surface Traversal

This work draws on the 3D traversal method for error diffusion developed by Brunton, Arikan, and Urban [11]. Their work breaks up the 3D surface into units more familiar for halftoning algorithms. An object represented by a 3D array of voxels can be considered in one-voxel-thick slices in the X-Y plane, moving

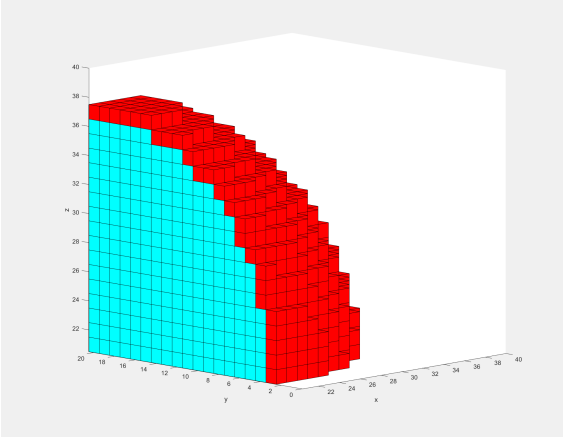


Figure 3. *One eighth of the sphere in figure 2, where cyan voxels represent the interior voxels of the shape*

up the object in the +z direction as shown in figure 4. Traversal moves from the bottom up since this is the order in which a 3D printer would build an object [19]. Each slice will have a region of interior voxels enclosed by surface voxels, as shown in figure 5.

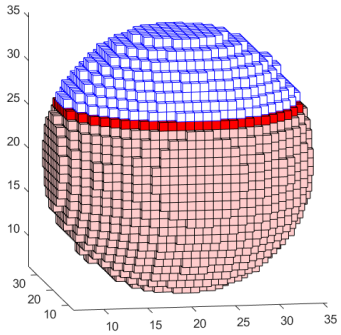


Figure 4. *A sphere taken in slices moving in the +z direction. The pink voxels belong to slices already visited, the red voxels are the slice currently being worked on, and the white/blue voxels are in slices that have yet to be visited.*

Now, the surface voxels in each slice will form some manner of closed loop around a line parallel to the z-axis, as shown in figure 6. These voxels can be visited by moving either clockwise or counterclockwise around that central line parallel to the z-axis.

As can be seen in figure 7 with a segment of the top three slices of a sphere, the ring of surface voxels within a slice will be multiple voxels wide when the surface is nearly horizontal. When this is the case, the traversal described above (in which the surface voxel ring is moved around in a rotational fashion) will have to either move radially inward or outward. Using figure 7, the shown segment of slice $z = 35$ has surface voxels with $y = 14$ and $y = 15$. Considering that the slices are visited in order of increasing z coordinate and error will be diffused toward adjacent unvisited voxels, then the surface voxels in $\{(y,z) : y = 16, z = 36\}$ should be receiving diffused error from the surface voxels in $\{(y,z) : y = 15, z = 35\}$. This would make sense, since the voxels

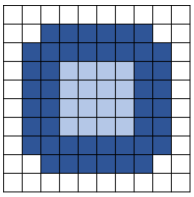


Figure 5. *An example of a slice, with surface voxels in dark blue, interior*

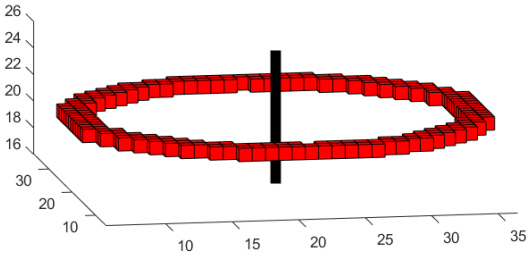


Figure 6. *The surface voxels in a slice.*

in $\{(y,z) : y = 15, z = 35\}$ are closer to those in slice $z = 36$ than are those in $\{(y,z) : y = 14, z = 35\}$. So, to summarize, the voxels at $y = 15$ will receive error from those at $y = 14$, then those at $y = 16$ will receive error from those at $y = 15$, and so on.

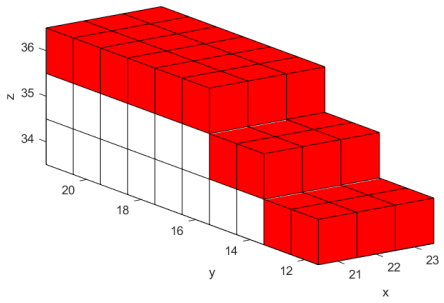


Figure 7. *A cutout section of the top three slices of a sphere; the ring of an object's surface voxels in a slice can be multiple voxels wide in nearly horizontal surfaces.*

To generalize, the innermost surface voxels of a slice will be closest to the next slice whenever the surface is facing upward. The opposite is true for a downward-facing surface. This can be handled by first computing the normal vector to the surface at each surface voxel, denoted by $n[m,n,p]$; voxels whose normal vectors have positive z components are part of an upward-facing section of the surface. Then, the distance from each voxel within that slice to the exterior of the slice, d_s, can be computed. Using slice $z = 35$ within figure 7 as an example, the voxel with $(x,y) = (22,14)$ has a distance of 1 from the exterior, $(x,y) = (22,15)$ has distance 2, $(x,y) = (22,16)$ has distance 3, and so on. If the traversal is moving counterclockwise within a slice, then the next voxel to be visited should satisfy the following:

- adjacent to the current voxel
- in the counterclockwise direction

- $\begin{cases} min(d_s), & \text{for upward-facing surface} \\ max(d_s), & \text{for downward-facing surface} \end{cases}$

Each loop of surface voxels within a slice is treated almost like a row of pixels that has been coiled up. So, in order to adapt serpentine scanning to 3D, the traversal will alternate between clockwise and counterclockwise directions when moving from one slice to the next slice, similar to the manner in which 2D serpentine scanning alternates between scanning rows of pixels from left to right and right to left.

The first voxel to be visited within a slice will be the one that has received the most diffused error the from the previous slice. Then, when the traversal mechanism is determining the next voxel to visit, it first considers only the surface voxels in the same slice in a 3x3 neighborhood around the voxel currently under operation (i.e., adjacent voxels within the same slice). From this neighborhood, surface voxels are split into those in the same direction of rotation as had been used previously and those in the opposite direction of rotation; then, within each group, the innermost or outermost voxel is given priority to be visited next, as according to the description above. Ideally, the traversal would continue in the same direction of rotation to a neighboring voxel. If this is not possible, then the traversal will consider next the prioritized voxel in the opposite direction of rotation. If this direction also yields no unvisited surface voxels, then the traversal jumps to the nearest unvisited surface voxel within the slice and starts again.

Error Diffusion in 3D

As described previously, the 3D object whose surface is being halftoned is considered in slices moving up the object. The first non-empty slice encountered will be composed entirely of surface voxels and exterior voxels. As described in [11], these surfaces are best halftoned as if they were 2D images in order to avoid start-up artifacts. The same is the case for the last non-empty slice of an object. So, the first and last non-empty slices encountered in an object are considered two-dimensional and halftoned with a serpentine scan, rather than allowing the 3D traversal mechanism to spin around the slice as would be the case for other slices.

When the 3D traversal method is used, error diffusion is applied by defining the error diffusion filter in the tangent plane to the surface at the voxel under operation. In order to orient the filter properly, two vectors are used. To find the vector that defines the vertical direction of the filter, $\vec{v}[m,n,p]$, first $\vec{v_n}[m,n,p]$ is computed as the vector from the voxel under operation toward the next voxel that is to be traversed. Then,

$$\vec{v}[m,n,p] = \vec{n}[m,n,p] \times \pm\vec{v_n}[m,n,p] \tag{5}$$

If traversal is moving counterclockwise, then positive $\vec{v_n}[m,n,p]$ is used in equation 5. For clockwise traversal, negative $\vec{v_n}[m,n,p]$ is used to ensure that $\vec{v}[m,n,p]$ has a positive z component.

Then, the vector that defines the horizontal direction of the filter, $\vec{u}[m,n,p]$, is defined as

$$\vec{u}[m,n,p] = \vec{u}[m,n,p] \times \pm\vec{n}[m,n,p] \tag{6}$$

For traversal that is moving counterclockwise, positive $\vec{n}[m,n,p]$ is used in equation 6. For clockwise traversal, negative $\vec{n}[m,n,p]$ is used to ensure that $\vec{u}[m,n,p]$ is perpendicular to both the surface and $\vec{v}[m,n,p]$ while being oriented toward the next voxel to be traversed.

An example of the 2D error diffusion filter being applied tangentially to the surface is shown in figure 9, where the multicolored rectangle represents the 2x3 error diffusion filter. While 2D error diffusion generally traverses an image from top to bottom, this work moves from the lowest slice to the highest slice. As a result, the error diffusion filter will have to be modified as shown below such that one component is still directed toward the next voxel to be traversed and the other three point toward voxels that have yet to be halftoned, either in a higher slice or within the current slice and closer to the next slice.

$w[1,-1]$	$w[1,0]$	$w[1,1]$
	*	$w[0,1]$

Figure 8. *Error Diffusion filter for moving up an object in 3D.*

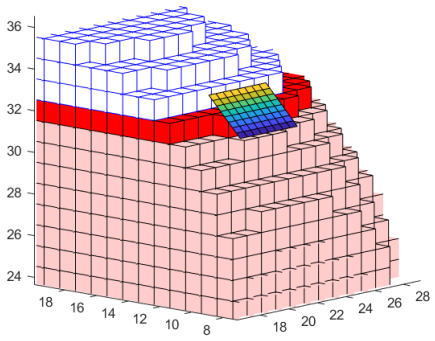


Figure 9. *The error diffusion filter is created in the tangent plane to the surface at the voxel under operation, then projected onto the surface. Here, the dark red voxels make up the slice being halftoned, and the filter is being applied at one of the outer voxels in that slice.*

In adapting tone-dependent error diffusion to 3D, this work uses the weight and threshold parameters for serpentine scanning that were calculated in [18]. That work optimized parameters by considering patches halftoned by DBS to be ideal, then either comparing the DBS patches' 2D DFTs with those of patches halftoned with varying error diffusion parameters or using the DBS cost function; the method used in [20] was similar, but compared the power series of the DBS and error diffusion outputs instead.

In an object printed or rendered with sufficiently high voxel resolution, the surface in a small region can be estimated by its tangent plane with a satisfactory degree of accuracy. Thus, for a surface treated as two-dimensional in a small region, the parameters for 2D error diffusion will be appropriate. Since the surface traversal method used here mimics serpentine scanning by alternating between moving clockwise and counterclockwise around a slice, the 2D serpentine parameters were chosen.

When traversal is forced to stop and restart at a different voxel or if the last voxel in a slice is under operation, then there may not be a next voxel adjacent that can receive the diffused error that is generated. When this happens, the error is only diffused with the three upper components of the error diffusion filter.

Results

Shown in figure 10 is a cube halftoned with the TDFED algorithm, and in figure 11 is the same cube halftoned with 3D Floyd-Steinberg error diffusion. The Floyd-Steinberg cube shows more pronounced artifacts in the 0.25, 0.5, and 0.75 absorptance regions, which can also be seen in figure 16.

Figure 12 shows a sphere halftoned with the TDFED algorithm, and figure 13 shows the same sphere halftoned with 3D Floyd-Steinberg error diffusion. The Floyd-Steinberg sphere shows some radial artifacts on regular lines of constant longitude that are not seen in the TDFED one, and the Floyd-Steinberg sphere shows comb patterns on lines of constant latitude. Both of these artifacts can be seen more clearly in figures 14 and 15, which show one-eighth cutouts of both spheres.

Conclusion

Compared with the outputs generated by 3D Floyd-Steinberg error diffusion, the 3D TDFED algorithm produces images with notably fewer artifacts. Looking at figures 10, 11, , and the artifacts that are most apparent in Floyd-Steinberg are the stripes in the cube around each quarter, which the tone-dependent cube avoids. In figures 12, 13, 14, and 15, the radial artifacts on lines of constant longitude and the comb artifacts on lines of constant latitude are the most evident flaws in the Floyd-Steinberg examples that the tone-dependent examples improve on.

Acknowledgments

We would like to thank Pingshan Li and Yafei Mao for their work with tone-dependent error diffusion and the algorithm introduced in [18], as this work uses the optimized weight and threshold parameters that they calculated for serpentine scanning.

Also, we would like to thank Ruiyi Mao for his development of the MATLAB program used here to display the results of 3D halftoning and to create illustrative figures.

References

[1] M. Analoui and J. Allebach, "Model based halftoning using direct binary search," in *Proceedings of the SPIE - The International Society for Optical Engineering*, vol. 1666, USA, 1992//, pp. 96 – 108.

[2] R. Floyd and L. Steinberg, "An adaptive algorithm for spatial greyscale," *Proceedings of the S.I.D.*, vol. 17, no. 2, pp. 75 – 7, 1976//, adaptive algorithm;spatial greyscale;error diffusion;picture element;.

[3] Witten and Neal, "Using peano curves for bilevel display of continuous-tone images," *IEEE Computer Graphics and Applications*, vol. 2, no. 3, pp. 47–52, May 1982.

[4] R. Ulichney, *Digital Halftoning*. Cambridge, MA, USA: MIT Press, 1987.

[5] P. W. Wong, "Adaptive error diffusion and its application in multiresolution rendering," *IEEE Transactions on Image Processing*, vol. 5, no. 7, pp. 1184 – 96, 1996/07/.

[6] B. Widrow and S. Stearns, *Adaptive signal processing*, Englewood Cliffs, NJ, USA, 1985//.

[7] R. Ulichney, "Dithering with blue noise," *Proceedings of the IEEE*, vol. 76, no. 1, pp. 56 – 79, 1988/01/.

[8] J. Sullivan, R. Miller, and G. Pios, "Image halftoning using a visual model in error diffusion," *Journal of the Optical Society of America A (Optics and Image Science)*, vol. 10, no. 8, pp. 1714 – 24, 1993/08/.

[9] K. Knox and R. Eschbach, "Threshold modulation in error diffusion," *Journal of Electronic Imaging*, vol. 2, no. 3, pp. 185 – 92, 1993/07/, error diffusion;threshold modulation;theoretical analysis;spatial modulation;equivalent input image;standard error diffusion algorithm;high-pass-filtered version;threshold spatial modulation;edge enhancement;noise;. [Online]. Available: http://dx.doi.org/10.1117/12.148736

[10] R. Eschbach, "Reduction of artifacts in error diffusion by means of input-dependent weights," *Journal of Electronic Imaging*, vol. 2, no. 4, pp. 352 – 8, 1993/10/.

[11] A. Brunton, C. A. Arikan, and P. Urban, "Pushing the limits of 3d color printing: Error diffusion with translucent materials," *ACM Transactions on Graphics*, vol. 35, no. 1, 2015.

[12] C. Zhou and Y. Chen, "Three-dimensional digital halftoning for layered manufacturing based on droplets," in *Transactions of the North American Manufacturing Research Institution of SME*, vol. 37, Greenville, SC, United states, 2009, pp. 175 – 182.

[13] C.-I. Lin, Y.-P. Sie, T.-H. Lin, and P.-L. Sun, "Slicing and halftoning algorithm for high quality color 3d printing," in *Proceedings of the International Display Workshops*, vol. 2, Otsu, Japan, 2015, pp. 786 – 789.

[14] Q. Lou and P. Stucki, "Fundamentals of 3d halftoning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1375, Saint-Malo, France, 1998, pp. 224 – 239.

[15] T. Kuipers, E. Doubrovski, and J. Verlinden, "3d hatching: Linear halftoning for dual extrusion fused deposition modeling," in *Proceedings - SCF 2017: ACM Symposium on Computational Fabrication*, Cambridge, MA, United states, 2017, pp. ACM Special Interest Group on Computer Graphics and Interactive Techniques (ACM SIGGRAPH) –.

[16] W. Cho, E. Sachs, N. Patrikalakis, and D. Troxel, "A dithering algorithm for local composition control with three-dimensional printing," *Computer Aided Design*, vol. 35, no. 9, pp. 851 – 67, 2003/08/.

[17] R. Mao, U. Sarkar, R. Ulichney, and J. P. Allebach, "3d halftoning," in *IS and T International Symposium on Electronic Imaging Science and Technology*, vol. 2017-January, Burlingame, CA, United states, 2017, pp. 147 – 155.

[18] P. Li and J. Allebach, "Tone-dependent error diffusion," *IEEE Transactions on Image Processing*, vol. 13, no. 2, pp. 201 – 15, 2004/02/.

[19] C.-I. Lin, Y.-P. Sie, T.-H. Lin, and P.-L. Sun, "Slicing and halftoning algorithm for high quality color 3d printing," in *Proceedings of the International Display Workshops*, vol. 2, Otsu, Japan, 2015, pp. 786 – 789.

[20] T. chiun Chang and J. Allebach, "Memory efficient error diffusion," in *Proceedings of the SPIE - The International Society for Optical Engineering*, vol. 5008, USA, 2003//, pp. 525 – 36.

Author Biography

Adam Michals is an undergraduate in electrical engineering at Purdue University. His research interests include halftoning and other image processing.

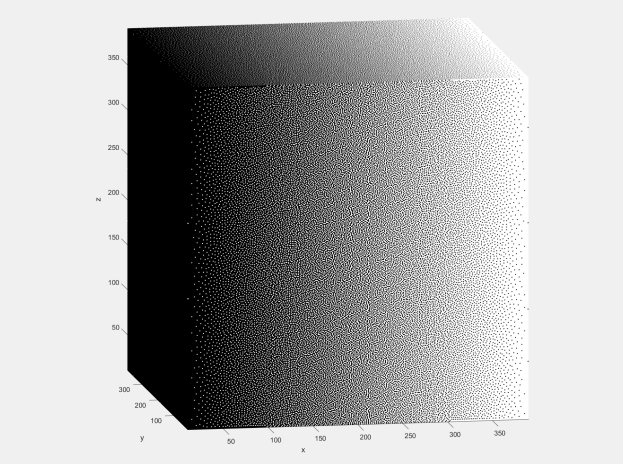


Figure 10. *TDFED-halftoned cube of dimensions 380x380x380 that changes from 1 absorptance to 0 absorptance in the positive x direction.*

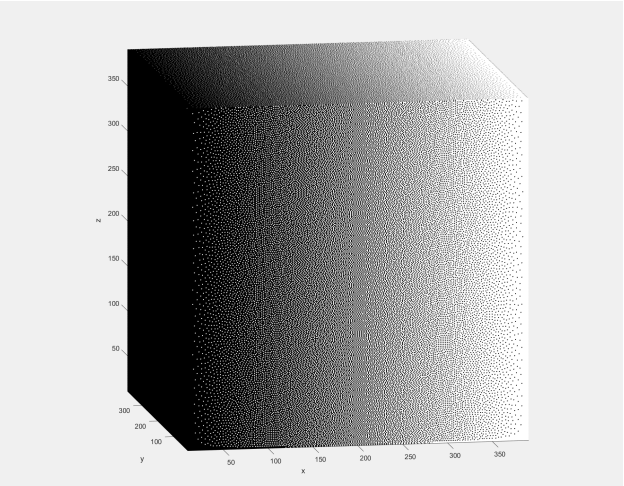


Figure 11. *FSED-halftoned cube of dimensions 380x380x380 that changes from 1 absorptance to 0 absorptance in the positive x direction.*

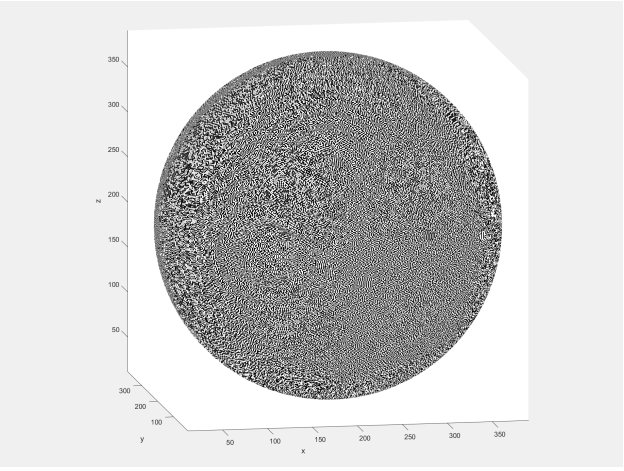


Figure 12. *TDFED-halftoned sphere of radius 190 with a constant input tone of 0.5 absorptance.*

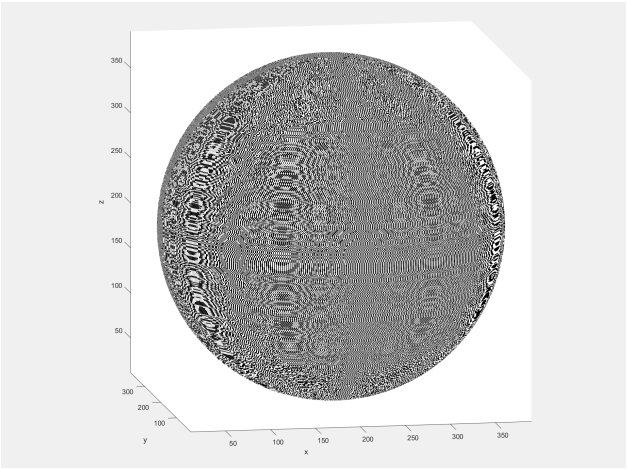


Figure 13. *FSED-halftoned sphere of radius 190 with a constant input tone of 0.5 absorptance.*

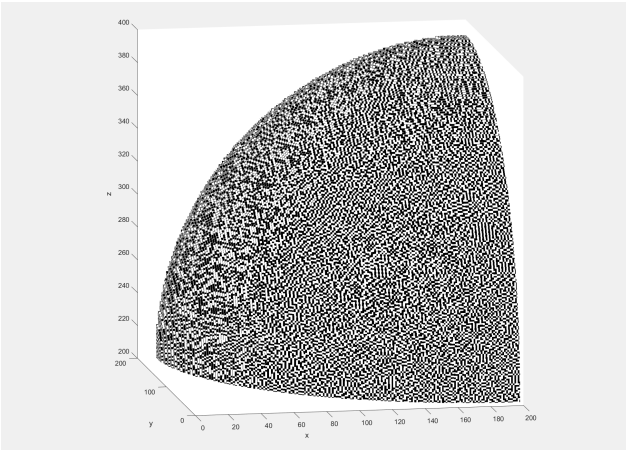


Figure 14. *One-eighth cutout of a TDFED-halftoned sphere of radius 190 with a constant input tone of 0.5 absorptance.*

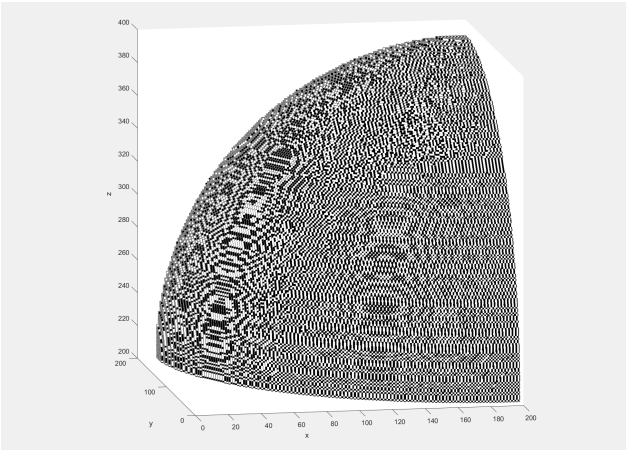
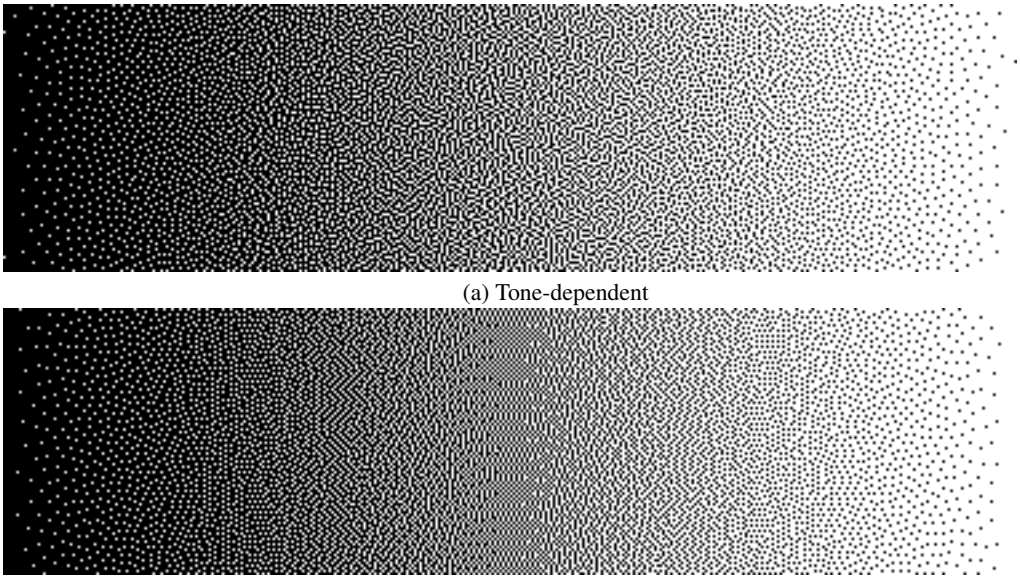


Figure 15. *One-eighth cutout of a FSED-halftoned sphere of radius 190 with a constant input tone of 0.5 absorptance.*



(a) Tone-dependent

(b) Floyd-Steinberg

Figure 16. *Cube faces*

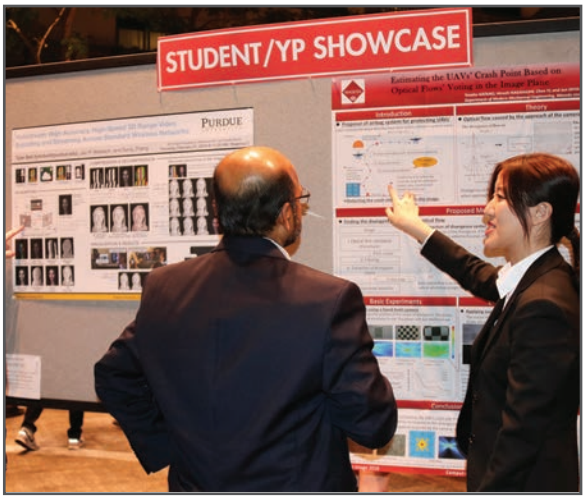
JOIN US AT THE NEXT EI!

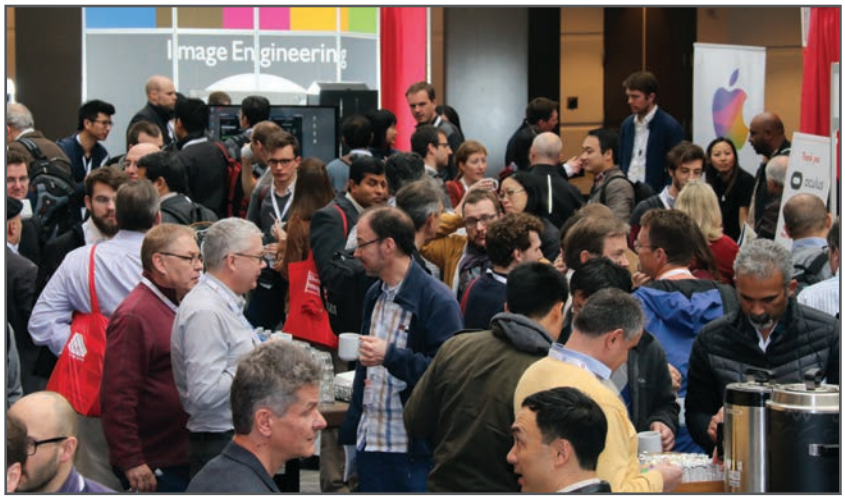
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org