https://doi.org/10.2352/ISSN.2470-1173.2017.18.COLOR-048


# 3D Halftoning*

***Ruiyi Mao*** [a]***, Utpal Sarkar*** [b]***, Robert Ulichney*** [c] ***and Jan P. Allebach*** [a]

[a] ***School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906, U.S.A***
[b] ***HP Inc., Large Format Division, Barcelona, 08174, Spain***
[c] ***HP Labs, Cambridge, MA 02142, U.S.A.***

## Abstract

*3D printing is becoming increasingly popular around the world today. By utilizing 3D printing technology, customized products can be manufactured much more quickly and efficiently with much less cost. However, 3D printing still suffers from low quality surface reproduction compared with 2D printing. One effective approach to improve it is to develop an advanced halftoning algorithm for 3D printing. In this paper, a novel 3D DBS (Direct Binary Search) halftoning algorithm that can cooperate with current 3D printing technology is proposed which can generate high quality surface reproduction.*

## Introduction

The emerging 3D printing technology enables human to create complicated 3D objects and structures with different kinds of materials more economically and efficiently than the traditional manufacturing process. However, currently more attention has been paid on the development of 3D printing process, material and structure rather than the 3D object surface image reproduction. With the growing need for high surface printing quality, improving the image reproduction quality on 3D surface has become more and more important in 3D printing area.

A number of 3D printing techniques that are able to do 3D color printing have been developed for some time. One technique uses powder and binder to generate 3D objects. It spreads thin layers of powders across a platform and the print head selectively deposits color binders onto the powder. And finally the redundant powders will be removed. Another technology utilizes layer-laminating system to do 3D color printing. It uses normal ink jet printing technology to print a slice of the 3D model on a piece of standard copy paper and then stack them up. A cutting blade traces the outline of the model to remove the excess paper. However, most of them suffer from low resolution and rough surface. Recently the Multi-Jet technology from companies like HP Inc. has been revealed to public. This technology enables micro manipulation of a single voxel, which can produce much smoother and higher resolution surface with high quality image reproduction. 3D printers equipped with this technology will have more potential in generating high quality and complex appearance 3D objects.

By utilizing Multi-Jet technology which enables voxel-wise printing, digital halftoning will be able to play a core role in improving 3D surface reproduction quality. In general, digital halftoning algorithms can be classified into three categories: screening, error diffusion and search-based methods. Screening only requires point-to-point memoryless comparison with a threshold to generate a halftone image with binary values. Error diffusion method includes point-to-point comparison also and requires a neighborhood processing with a little bit memory. Search-based method directly searches for the best arrangement of binary pattern for the halftone image by minimizing the error metric of perceived error. Search based methods are usually iterative which requires more computation than screening and error diffusion methods which require only one pass of the entire image and no or very small memory. However, search-based methods yield better halftone quality than screening and error diffusion at the cost of higher computation. Among several reported search-based methods, DBS which was first reported in [1] yields very good halftone quality and is currently one of the most widely used search-based methods.

In fact, several 3D digital halftoning algorithms have been reported recently based on the 2D halftoning algorithms of screening and error diffusion. For example, the monochrome 3D dispersed-dot screening digital halftoning algorithm is reported in [2]. It generated a 3D volume threshold matrix to halftone a 3D object. Some other works like Brunton et al.[3] and Zhou et al. [4] applied error diffusion to layer based 3D object halftoning for color and monochrome 3D halftoning respectively.

*Research supported by HP Inc., Palo Alto, CA.

However, the search-based methods which yields the best halftone quality in 2D printing still have not been applied to 3D printing. In this work, DBS algorithm is redeveloped and applied to monochrome 3D surface halftoning. Regarding DBS algorithm, a lot of works have been done within 2D halftoning area, for example [5][6][7][8]. Our work is the first trying to adapt it to 3D and is based on the numerous works done in 2D printing area mentioned above.

## Methodology

### A. 3D Visual Model

In this paper, we use $(x) = (x,y,z)^T \in \mathbb{R}^3$ to represent continuous 3D spatial coordinates and $[m] = [m,n,o]^T \in \mathbb{Z}^3$ to represent discrete 3D spatial coordinates. We use $f[m]$ to denote the original discrete 3D continuous-tone image and $g[m]$ to denote the original discrete 3D halftone image. For monochrome halftone, $g[m] = 0$ or 1.

The continuous 3D HVS (Human Visual System) filer or PSF (point spread function) which will be explained in detail later is denoted by $p_{hvs}(x)$. The printer dot profile function which embodies the printer model is denoted by $p_{dot}(x)$. And usually $\tilde{p}(x) = p_{dot}(x) * p_{hvs}(x)$ is used to represent the total effects of both the printer and HVS models. However, since 3D printers tend to be high-resolution and have consistent size voxels, the effect from printer model can be ignored so that $\tilde{p}(x) \approx p_{hvs}(x)$. In this report, the approximation will be used and HVS filter will be referred as $\tilde{p}(x)$.

The discrete 3D HVS filter is represented by $\tilde{p}[m] = \tilde{p}(Xm)$, where $X$ is the periodicity matrix whose columns comprise a basis for the lattice [9]. So the perceived 3D halftone image through human visual system is denoted by $\tilde{g}(x)$ and is calculated by convolving the original halftone image $g[m]$ with the 3D HVS filter $\tilde{p}(x)$ as

$$\tilde{g}(x) = \sum_m g[m]\tilde{p}(x - Xm) \tag{1}$$

and the discrete perceived 3D halftone image is given by

$$\tilde{g}[m] = g[m] * \tilde{p}[m] = \sum_n g[n]\tilde{p}[m-n] \tag{2}$$

And the discrete perceived 3D original continuous-tone image is given by

$$\tilde{f}[m] = f[m] * \tilde{p}[m] = \sum_n f[n]\tilde{p}[m-n] \tag{3}$$

### B. 3D HVS Filter

A HVS filter is an HVS model which captures the low-pass characteristic of the human visual system. It is the most important aspect used in the halftoning process to obtain the perception of images.

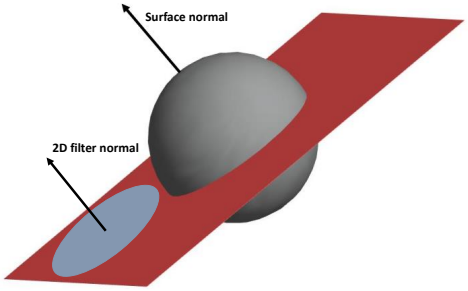
2D HVS filter models the human visual system and assumes the eye sight always is normal to the 2D plane. If a 2D HVS filter is used in the 3D halftoning process, it has to be guaranteed that the 2D HVS filter plane is always perpendicular to the surface normal of the 3D object. However for 3D object, its surface orientation and curvature may keeps changing so that its surface normal varies. As a result, it would be highly computational to apply a 2D HVS filter to 3D object's halftoning since we have to keep adjusting the orientation of the 2D HVS filter all the time. In order to efficiently obtain the perceptual image in 3D space, a 3D HVS filter is established.

According to the HVS model, a spacial 2D HVS filter is an isotropic symmetric function on a 2D plane with its maximal value at the origin. All points on the filter plane with the same distance from origin have the same value and these points form one of the concentric circles centered at the origin. Since the 2D HVS filter is symmetric and isotropic, the 3D HVS filter can be simply built by revolving the 2D circular filter plane around any of its diameters and duplicate its value on any spatial coordinate it goes through. By doing this, a 3D spherical HVS filter is built and any cross section going through its center is a duplicate of the original 2D HVS filter. Similar as 2D filter, a spacial 3D HVS filter is an isotropic symmetric function in a 3D space with its maximal value at the origin and all the points on any sphere centered at the origin have the same value which is inverse proportional to its distance from the origin.

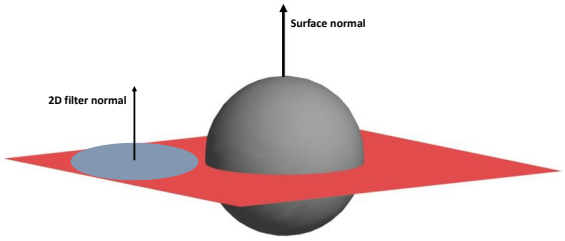
According to the sensitivity function of human visual system, the effective supporting area of a normal HVS filter is usually hundredth to even thousandth of one square inch according to normal viewing distance and printer resolution. As a result when an HVS filter is applied to the surface of a 3D object, the part of the surface covered by the filter supporting area can be approximated as a flat plane.

With the property of the 3D HVS filter discussed above, the correct HVS response will always be obtained as long as the origin of the 3D HVS filter is applied on the 3D surface, since any cross section of the 3D filter is exactly a duplicate of a 2D HVS filter with its normal perpendicular to the cross section which is the 3D surface plane. By doing this, accurate and efficient HVS response can be obtained without modification of the HVS filter according to the 3D surface orientation. As shown in Fig. 1, a circular 2D HVS filter and a spherical 3D HVS filter are applied on 3D surfaces with different ori-

entation. It can be seen that the 2D HVS filter need to be adjusted for different surface in order to make sure its normal is parallel to the surface normal. And it is obvious that regardless of the surface orientation, as long as the 3D filter center is on the surface, it is equivalent as applying a correctly adjusted 2D filter whose normal is perpendicular to the surface plane, since the cross section containing the filter center is a duplicate of the original 2D HVS filter.




(a) 2D and 3D HVS filter applied on an horizontal plane.




(b) 2D and 3D HVS filter applied on an inclined plane.

***Figure 1.*** *Applying a 2D circular HVS filter (blue circle) and 3D spherical HVS filter (gray sphere) to 3D surface (red plane) with different surface normal (black arrow).*

An HVS filter is based on the contrast sensitivity function of the human visual system. For this case which is currently monochrome halftoning, only the luminance channel is used. A work done by Kim and Allebach [10] compared four different HVS models and came up with the conclusion that Näsänen's model [11] yielded the best halftone quality over the other three when incorporated with DBS algorithm. So in this work the luminance contrast sensitivity function based on Näsänen's model is used. The luminance spatial frequency response $H(\bar{u},\bar{v})$ of the original 2D Näsänen's model is given [11] by

$$H(\bar{u},\bar{v}) = a\Gamma^b \exp\left(-\frac{\sqrt{\bar{u}^2+\bar{v}^2}}{c\ln\Gamma+d}\right) \tag{4}$$

where $a = 131.6$, $b = 0.3188$, $c = 0.525$, $d = 3.91$. $\Gamma$ is the average luminance of the light reflected from the print in $cd/m^2$. $(\bar{u},\bar{v})$ is the spatial frequency coordinates in cycles/degree subtended at the retina.

By taking the inverse 2D continuous-space Fourier transform, the 2D spacial filter $\bar{h}(\bar{x},\bar{y})$ is obtained and the units of $(\bar{x},\bar{y})$ are degrees subtended at the retina. In order to convert the units on retina to printed page, it is discovered that a length $x$ inches on the paper viewed at a distance $D$ inches will subtend the angle $\bar{x}$ degrees as

$$\bar{x} = \frac{180}{\pi}\tan^{-1}\left(\frac{x}{D}\right) \approx \frac{180}{\pi D}x, \qquad for \quad x << D \tag{5}$$

And the continuous Näsänen's model HVS filter is obtained as

$$\tilde{p}(x,y) = h(x,y) = a\Gamma^b \frac{2\pi k}{\left(k^2+4\pi^2(x^2+y^2)\right)^{3/2}} \tag{6}$$

and the discrete Näsänen's model HVS filter is

$$\begin{aligned}\tilde{p}[m,n] = h[m,n] &= h\left(\frac{m}{R},\frac{n}{R}\right)\\ &= a\Gamma^b \frac{2\pi k}{\left(k^2+\frac{4\pi^2}{R^2}(m^2+n^2)\right)^{3/2}}\end{aligned} \tag{7}$$

where

| | |
|---|---|
| $a$ | $= 131.6$ |
| $b$ | $= 0.3188$ |
| $c$ | $= 0.525$ |
| $d$ | $= 3.91$ |
| $k$ | $= \frac{(\pi D)/180}{c\ln\Gamma+d}$ |
| $\Gamma$ | average luminance of the light reflected from the print in $cd/m^2$ |
| $D$ | viewing distance |
| $R$ | printer resolution |

It should be mentioned that $S = RD$ is called Scale Factor and is commonly used in describing the human visual system. In this work, the printer resolution $R$ and viewing distance $D$ are set to be 300 dpi and 10 inch, which yield a scale factor S of 3000.

As discussed above, the 3D Näsänen's model filter is built by revolving the 2D circular filter plane around any of its diameters and duplicate its value on any spatial coordinate it goes through so that any cross section going

through its center is a duplicate of a 2D Näsänen's model filter. By doing this the continuous 3D Näsänen's model filter is

$$h(x,y,z) = a\Gamma^b \frac{2\pi k}{\left(k^2 + 4\pi^2(x^2+y^2+z^2)\right)^{3/2}} \tag{8}$$

The discrete 3D Näsänen's model filter is obtained as

$$\begin{aligned} h[m] &= h[m,n,o] = h\left(\frac{m}{R}, \frac{n}{R}, \frac{o}{R}\right) \\ &= a\Gamma^b \frac{2\pi k}{\left(k^2 + \frac{4\pi^2}{R^2}(m^2+n^2+o^2)\right)^{3/2}} \end{aligned} \tag{9}$$

Since the filter function has no boundary, the spherical truncated discrete Näsänen's mode filter in the 3D spatial domain is given by

$$\begin{aligned} \tilde{p}[m] &= \tilde{p}[m,n,o] = h_t[m,n,o] \\ &= \begin{cases} h[m,n,o], & \sqrt{m^2+n^2+o^2} \le N \\ h[m,n,o](N+1-\sqrt{m^2+n^2+o^2}), & \\ \qquad N < \sqrt{m^2+n^2} \le N+1 & \\ 0, & N+1 < \sqrt{m^2+n^2+o^2} \end{cases} \end{aligned}$$

In the following sections, we will discuss the the mean squared error between the perceived halftoned image and continuous-tone image only depends on $\tilde{p}(x)$ through its autocorrelation function

$$c_{\tilde{p}\tilde{p}}(x) = \int \tilde{p}(y)\tilde{p}(y+x)dy \tag{10}$$

By evaluating the function at points on the printer lattice, we get

$$c_{\tilde{p}\tilde{p}}[m] = \sum_n \tilde{p}[n]\tilde{p}[n+m] = \tilde{p}[m] ** \tilde{p}[-m] = c_{\tilde{p}\tilde{p}}(Xm) \tag{11}$$

### C. 3D Search Heuristic

DBS halftoning algorithm seeks to minimize the mean-squared error between perceptually filtered continuous-tone original image and halftoned binary image. In 2D DBS, this is done by scanning through the image pixel-by-pixel. At each pixel, the trials of changing the state of the pixel (toggle), or swapping it with its neighbors in an $M$ by $M$ neighborhood that have different state from it are implemented. By evaluating the effect of each trial change, the one that decreases the mean-squared error the most is kept. One single iteration of the algorithm consists of a visit to every pixel in the image and the algorithm terminates when no change is made during an entire iteration. Typically it requires ten or so iterations, regardless of the size of the image. Through experimental results, the larger the swapping neighborhood is, the better quality halftone it will yield at the expense of more computation. In order to make the algorithm more efficient, an idea of processing the pixels in non-overlapping cells in 2D DBS halftoning is presented in [12]. The purpose of doing this is to eliminate changes that yield only a small decrease in error but take the same computational cost as more effective changes. In [12], it states that it is found $4 \times 5$ was an optimum cell size.

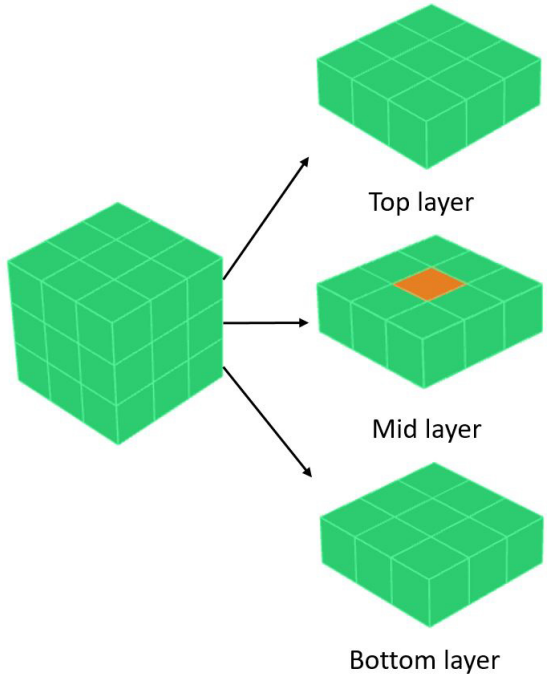



***Figure 2.*** *An example of 3D search heuristic. The neighborhood size is* $3 \times 3 \times 3$. *The red voxel in the mid layer is the voxel currently under processing and all the green voxels are its neighbor voxels. For swap trials, the voxel under processing will be swapped with the neighbor voxels in green if the green voxels are on-surface voxels and hold different state with it.*

3D DBS search heuristic is developed based on 2D DBS search heuristic. For 3D DBS halftoning algorithm, firstly a 3D working space is defined. And each voxel inside the working space has three different status: on surface, inside the object and outside the object. 3D DBS algorithm scans through all the voxels in the working space in raster order. Since the algorithm only does surface

halftoning, at each voxel, if the voxel is on the surface of the 3D object, the algorithm does the trial of toggling this voxel, or swapping it with its on-surface neighbor voxels with different state in an $M \times M \times M$ neighboring space. By only swapping with on-surface voxels, the on-surface halftoning is realized and the object shape will keep constant. An example of 3D search heuristic is shown in Fig. 2. And by evaluating each trial change, the one which decreases the mean-squared error the most will be kept. To accelerate the halftoning process, we can build an on-surface voxel address look-up table in the first iteration, and scan the voxels with address in the table for the following iterations instead of scanning all the voxels in the working space. For more efficient 3D DBS halftoning algorithm, the concept of cells in 2D DBS halftoning is extended to 3D DBS halftoning by introducing the concept of blocks. A block is a 3D volume and the 3D working space is divided by the non-overlapping 3D blocks. In each block only one change can be made within one iteration in order to eliminate inefficient changes. In this work, the block size is set to be $5 \times 5 \times 5$ which yields good halftone result. In the following sections, how to efficiently calculate and evaluate the changes of mean-squared error with different trials will be discussed.

### D. Error Metric

As discussed above, DBS needs to evaluate the error changes of the toggle or swap trials in order to determine which trial should be accepted. So it is necessary to introduce the error metric for 3D DBS algorithm. As stated above, $f[m]$ is used to denote the original discrete space continuous-tone 3D image, where $[m] = [m,n,o]^T \in \mathbb{Z}^3$. Each voxel of $f[m]$ on the 3D object surface takes an absorptance value between 0 to 1 for monochrome case. And the voxels of halftone image $g[m]$ on 3D object surface take values of 0 (white) or 1 (black). The goal of DBS is to compute a halftone image $g[m]$ that have a minimal measure of a perceptually filtered error image. The original error between the original continuous-tone image and the halftoned image is given by

$$e[m] = g[m] - f[m] \tag{12}$$

and the continuous perceived error is

$$\tilde{e}(x) = \sum_m e[m]\tilde{p}(x - Xm) \tag{13}$$

where $(x) = (x,y,z)^T \in \mathbb{R}^3$.

Then the error metric needs to be minimized by DBS algorithm is the total squared perceived error which is given by

$$E = \int |\tilde{e}(x)|^2 dx \tag{14}$$

However it is time consuming to calculate $E$ directly for each trial. In the next section, an efficient way to evaluate the error metric change of each trial will be discussed.

### E. Efficient Evaluation of Halftone Trial Changes

In order to reduce computational complexity, an efficient way to evaluating the effect of halftone trial changes is given in [1]. In this work, it is extended to 3D space. Same as above, we use $(x) = (x,y,z)^T \in \mathbb{R}^3$ to represent continuous 3D spacial coordinates and $[m] = [m,n,o]^T \in \mathbb{Z}^3$ to represent discrete 3D spatial coordinates. First an additional correlation function is defined

$$c_{\tilde{p}\tilde{e}}(x) = \int \tilde{p}(y)\tilde{e}(y+x)dy \tag{15}$$

By substituting (13) into (14), $E$ can be expressed as

$$\begin{aligned} E &= \int |\tilde{e}(x)|^2 dx \\ &= \sum_m \sum_n e[m]e[n]\tilde{p}(x - Xm)\tilde{p}(x - Xn) \\ &= \sum_m \sum_n e[m]e[n]c_{\tilde{p}\tilde{p}}[m-n] \end{aligned} \tag{16}$$

Substituting (13) into (15), we get

$$c_{\tilde{p}\tilde{e}}[m] = c_{\tilde{p}\tilde{e}}(Xm) = \sum_n e[n]c_{\tilde{p}\tilde{p}}[n-m] \tag{17}$$

Usually the $c_{\tilde{p}\tilde{e}}$ matrix is initialized through the above equation at the beginning of the DBS algorithm. For 2D DBS, the $c_{\tilde{p}\tilde{e}}$ matrix should includes the $c_{\tilde{p}\tilde{e}}$ value for all pixels of the image. However, for 3D DBS halftoning algorithm, an on-surface voxel address look-up table is built and only the $c_{\tilde{p}\tilde{e}}$ values for on-surface voxels in the look-up table need to be calculated in order to reduce the computation. The reason for this will be discussed later. The total squared perceived error can be expressed as

$$E = \sum_m e[m]c_{\tilde{p}\tilde{e}}[m] \tag{18}$$

However, it is still painful to calculate the total error after each trial. It is better to consider the effect of toggle and swap in a more efficient way. First, let's consider of a halftone voxel toggle trial at index $m_0$. With this toggle,

$g[m]$, $e[m]$, and $c_{\tilde{p}\tilde{e}}[m]$ will change as follows:

$$g'[m] = g[m] + a_0\delta[m-m_0] \tag{19}$$
$$e'[m] = e[m] + a_0\delta[m-m_0] \tag{20}$$
$$c'_{\tilde{p}\tilde{e}}[m] = c_{\tilde{p}\tilde{e}}[m] + a_0 c_{\tilde{p}\tilde{p}}[m-m_0] \tag{21}$$

where the parameters with prime represents its new value after the toggle. $a_0 = -1$ if $g[m_0]$ is changed from 1 to 0, and $a_0 = 1$ if $g[m_0]$ is changed from 0 to 1. $\delta[m] = 1$ if $m = 0$ and 0 otherwise.

By substituting (20) and (21) into (18), the changing in error caused by a toggle at one voxel is given by

$$\begin{aligned}\Delta E_0 &= \int |\tilde{e}'(x)|^2 dx - \int |\tilde{e}(x)|^2 dx \\ &= \sum_m (e'[m]c'_{\tilde{p}\tilde{e}}[m] - e[m]c_{\tilde{p}\tilde{e}}[m]) \\ &= \sum_m (e[m] + a_0\delta[m-m_0])(c_{\tilde{p}\tilde{e}}[m] \\ &\quad + a_0 c_{\tilde{p}\tilde{p}}[m-m_0]) - e[m]c_{\tilde{p}\tilde{e}}[m] \\ &= a_0^2 c_{\tilde{p}\tilde{p}}[0] + 2a_0 c_{\tilde{p}\tilde{e}}[m_0]\end{aligned} \tag{22}$$

It should be noted that $c_{\tilde{p}\tilde{p}}[m]$ and $c_{\tilde{p}\tilde{e}}[m]$ can both be stored as look-up tables (LUTS). If the change of toggle reduce the total error, $\Delta E_0 < 0$. To determine whether to accept the toggle or not, all the possible trials including the trial of swap should be evaluated.

For a swap trial, the two halftone voxels with different binary states will exchange their current states. The changed error $\Delta E$ caused by the swapping of two voxels at coordinate $m_0$ and $m_1$ is calculated by

$$\Delta E = \Delta E_0 + a_1^2 c_{\tilde{p}\tilde{p}}[0] + 2a_1^2 c'_{\tilde{p}\tilde{e}}[m_1] \tag{23}$$

where $m_0$ is the same voxel as the above toggle trial and $\Delta E_0$ is the trial effect of toggling the voxel $m_0$ in (22). $c'_{\tilde{p}\tilde{e}}[m_1]$ is the cross-correlation after $m_0$ is toggled. $a_1 = -1$ if $g[m_1]$ is changed from 1 to 0 through the swap, and $a_1 = 1$ if $g[m_1]$ is changed from 0 to 1. By substituting (21) and (22) into (24), we get

$$\begin{aligned}\Delta E =& (a_0^2 + a_1^2)c_{\tilde{p}\tilde{p}}[0] + 2a_0 c_{\tilde{p}\tilde{e}}[m_0] \\ &+ 2a_1 c_{\tilde{p}\tilde{e}}[m_1] + 2a_0 a_1 c_{\tilde{p}\tilde{p}}[m_1 - m_0]\end{aligned} \tag{24}$$

The equation (24) is a generalized equation that can efficiently calculate the change in error for both toggle and swap. For toggle, $a_1$ is set to 0 and for swap $a_1$ is set accordingly.

After evaluating of all possible trials of toggle and swaps, the trial that reduce the total error the most for the current voxel under processing will be kept. After one change is accepted, $g[m]$ and $c_{\tilde{p}\tilde{e}}[m]$ need to be updated to prepare for the next change. The equations to update $g[m]$ and $c_{\tilde{p}\tilde{e}}[m]$ is given as

$$g'[m] = g[m] + a_0\delta[m-m_0] + a_1\delta[m-m_1] \tag{25}$$
$$c'_{\tilde{p}\tilde{e}}[m] = c_{\tilde{p}\tilde{e}}[m] + a_0 c_{\tilde{p}\tilde{p}}[m-m_0] + a_1 c_{\tilde{p}\tilde{p}}[m-m_1] \tag{26}$$

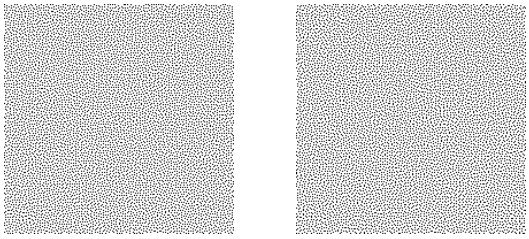
It should be noted that these are also generalized equations for both toggle and swap. If the accepted change is toggle, $a_1$ is set to 0. If the accepted change is swap, we should set $a_1$ value accordingly as stated above. And also, from equation (24), it can be observed that the calculation of $\Delta E$ only relies on $c_{\tilde{p}\tilde{p}}$ and $c_{\tilde{p}\tilde{e}}$ values of the current voxel and the voxel to be swapped with. Since these voxels are all on-surface voxels, it means only the $c_{\tilde{p}\tilde{p}}$ and $c_{\tilde{p}\tilde{e}}$ values for on-surface voxels are needed for trial changes evaluation. As a result, for $c_{\tilde{p}\tilde{e}}$ matrix initialization and update, only the values of on-surface voxels which is in the on-surface LUT are required to be calculated. This can greatly reduce the computation of this algorithm since $c_{\tilde{p}\tilde{e}}$ matrix initialization is highly time consuming. To summarize, with on-surface LUT, equation (24), (25) and (26), the computational efficiency can be largely improved.

## Experimental Results

It is reasonable to compare the halftone quality of 3D DBS halftoning algorithm with original 2D DBS halftoning algorithm regarding halftoning a plane, since for halftoning a plane, 3D DBS should have similar halftone quality comparing with 2D DBS. As shown in Fig. 3, the halftones of constant tone patches are compared with 2D and 3D DBS algorithm. For 2D DBS, the input are 2D constant tone images. For 3D DBS, the input are one-vexel-width flat constant tone square planes in a 3D working space. In Fig. 4, the halftone quality of a ramp is compared between 2D and 3D DBS the same way as above. It is observed that, the two algorithms yield very similar halftone quality for flat planes.

In order to visualize 3D DBS halftoning result, a lightweight 3D voxel visualization tool is developed in MATLAB. Fig. 5, Fig. 6 and Fig. 7 show three representative halftoned 3D objects with absorptance 0.5. Fig. 5 is a halftoned cube with absorptance 0.5 on its surface. Fig. 6 shows part of a halftoned sphere with absorptance 0.5 on its surface. It should be noted that the voxel under this viewing angle is not square and is more like a hexagon, which is explained in Fig. 7. In Fig. 7, a halftoned sphere with absorptance 0.5 is shown and three

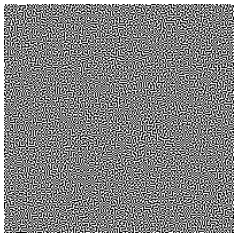
typical halftone patterns on different part of the sphere are shown on the right. In this three patterns, it is clear to see that they have different visual shapes of voxels, which is square, rectangle and hexagon. This is because of the different viewing angle of the cubic voxel which is shown right next to them. From these figures, it can be observed that the 3D DBS algorithm yields reasonable 3D halftones with good quality.



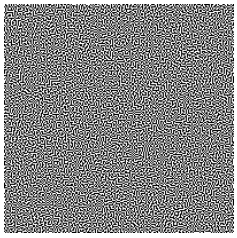
(a) 2D DBS halftone with absorptance 0.2.



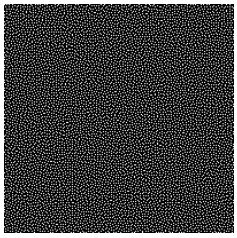
(b) 3D DBS halftone with absorptance 0.2.



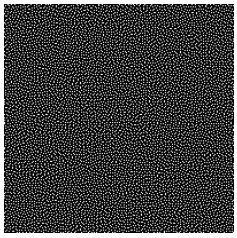
(c) 2D DBS halftone with absorptance 0.5.



(d) 3D DBS halftone with absorptance 0.5.

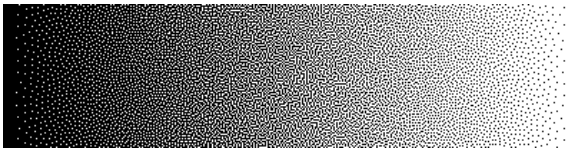


(e) 2D DBS halftone with absorptance 0.8.

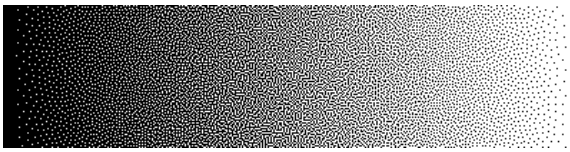


(f) 3D DBS halftone with absorptance 0.8

***Figure 3.*** *Flat square planes halftoned by 2D and 3D DBS*



(a) 2D DBS halftoned ramp.



(b) 3D DBS halftoned ramp.

***Figure 4.*** *Flat ramp plane halftoned by 2D and 3D DBS.*

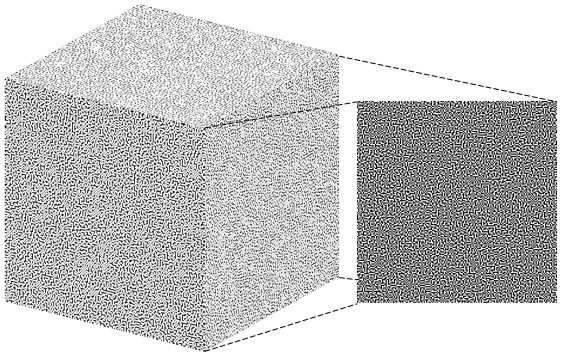


***Figure 5.*** *3D DBS halftoned cube with absorptance 0.5.*

## Conclusion

In this work, a novel 3D DBS halftoning algorithm is developed. This algorithm does not require any preprocess of the 3D original continuous tone object, for example slicing it or splitting it. And also, it doesn't requires any 3D to 2D mapping in order to implement the halftoning algorithm. Instead it directly halftone the surface of 3D objects with 3D human visual system quickly and efficiently. The 3D DBS halftoning algorithm can yield very reasonable and high-quality halftones which is almost the same as 2D DBS does in 2D printing. By only halftone the surface of the object and the use of on-surface look-up table, the computational cost is greatly reduced.

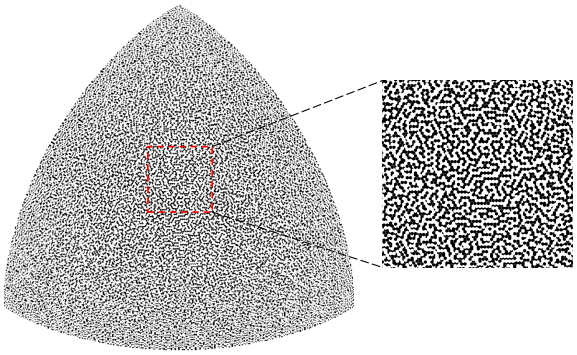


***Figure 6.*** *Part of a 3D DBS halftoned sphere with absorptance 0.5.*

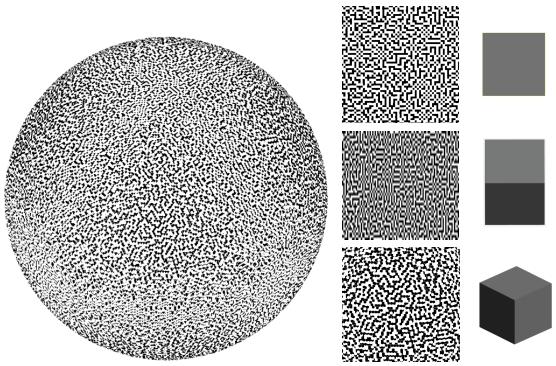


***Figure 7.*** *3D DBS halftoned sphere with absorptance 0.5 and two typical halftone patterns in 3D halftoned object caused by different viewing angle.*

## References


[1] M. Analoui and J. P. Allebach, "Model-based halftoning using direct binary search," in SPIE/IS&T 1992 Symposium on Electronic Imaging: Science and Technology, pp. 96-108, International Society for Optics and Photonics, 1992.

[2] W. Cho, E. M. Sachs, N. M. Patrikalakis, and D. E. Troxel, "A dithering algorithm for local composition control with three-dimensional printing," Computer-aided design, vol. 35, no. 9, pp. 851-867, 2003.

[3] A. Brunton, C. A. Arikan, and P. Urban, "Pushing the limits of 3d color printing: Error diffusion with translucent materials," ACM Transactions on Graphics (TOG), vol. 35, no. 1, p. 4, 2015.

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

[5] D. J. Lieberman and J. P. Allebach, "A dual interpretation for direct binary search and its implications for tone reproduction and texture quality," IEEE Transactions on Image Processing, vol. 9, no. 11, pp. 1950-1963, 2000.

[6] F. A. Baqai and J. P. Allebach, "Halftoning via direct binary search using analytical and stochastic printer models," IEEE Transactions on Image Processing, vol. 12, no. 1, pp. 1-15, 2003.

[7] J. H. Lee and J. P. Allebach, "Colorant-based direct binary search halftoning," Journal of Electronic Imaging, vol. 11, no. 4, pp. 517-527, 2002.

[8] D. J. Lieberman and J. P. Allebach, "Efficient model based halftoning using direct binary search," in Image Processing, 1997. Proceedings., International Conference on, vol. 1, pp. 775-778, IEEE, 1997.

[9] T. S. Rao, G. R. Arce, and J. P. Allebach, "Analysis of ordered dither for arbitrary sampling lattices and screen periodicities," IEEE transactions on acoustics, speech, and signal processing, vol. 38, no. 11, pp. 1981-2000, 1990.

[10] S. H. Kim and J. P. Allebach, "Impact of hvs models on model-based halftoning," IEEE Transactions on Image Processing, vol. 11, no. 3, pp. 258-269, 2002.

[11] R. Nasanen, "Visibility of halftone dot textures," IEEE transactions on systems, man, and cybernetics, vol. 14, no. 6, pp. 920-924, 1984.

[12] J. P. Allebach, DBS: retrospective and future directions," in Photonics West 2001-Electronic Imaging, pp. 358-376, International Society for Optics and Photonics, 2000.


## Author Biography

***Ruiyi Mao*** *received his B.Eng degree in Electrical and Computer Engineering from Huazhong University of Science and Technology, China, and University of Birmingham, UK, in May 2012. He is currently pursuing his Ph.D. degree in Electrical and Computer Engineering at Purdue University. His current research interest includes digital halftoning, digital image processing and computer vision.*

***Utpal Sarkar*** *received his M.Sc. in Mathematics from the University of Utrecht, the Netherlands. He works at the Large Format Division of HP in Barcelona on image processing and printing pipelines for 2D and 3D printing.*

***Robert Ulichney*** *is a Distinguished Technologist with HP Labs focusing on systems for high capacity data embedding, and structures for variable density 3D printing. He received a Ph.D. from MIT in electrical engineering and computer science. Before joining HP he was with Digital Equipment Corp for several years then with Compaqs Cambridge Research Lab. His publications are available at ulichney.com.*

***Jan P. Allebach*** *is Hewlett-Packard Distinguished Professor of Electrical and Computer Engineering at Purdue University. Allebach is a Fellow of the IEEE, the National Academy of Inventors, the Society for Imaging Science and Technology (IST), and SPIE. He was named Electronic Imaging Scientist of the Year by IS&T and SPIE, and was named Honorary Member of IST, the highest award that IST bestows. He has received the IEEE Daniel E. Noble Award, and is a member of the National Academy of Engineering. He currently serves as an IEEE Signal Processing Society Distinguished Lecturer (2016-2017).*