

# 3D Halftoning based on Iterative Method Controlling Dot Placement

**Fereshteh Abedini, Sasan Gooran, and Daniel Nyström; Media and Information Technology Division, Department of Science and Technology, Linköping University; Norrköping, Sweden.**

## Abstract

*Realistic appearance reproduction is of great importance in 3D printing's applications. Halftoning as a necessary process in printing has a great impact on creating visually pleasant appearance. In this article, we study the aspects of adapting and applying Iterative Method Controlling Dot Placement (IMCDP) to halftone three-dimensional surfaces. Our main goal is to extend the 2D algorithm to a 3D halftoning approach with minor modifications. The results show high-quality reproduction for all gray tones. The 3D halftoning algorithm is not only free of undesirable artifacts, it also produces fully symmetric and well-formed halftone structures even in highlight and shadow regions.*

## Introduction

2D printing and reproducing images in two-dimensional domain have been well-studied for several decades. Many technologies and algorithms have been developed to improve the quality of print productions.

In printing applications, halftoning is a necessary process and plays a significant role in the quality of the image reproduction. Halftoning algorithms can be divided into three main categories: thresholding and table halftoning, error diffusion, and iterative methods. In thresholding, pixels are converted to black or white pixels based on a simple comparison with a threshold. Thresholding is a simple point-by-point halftoning method but it does not always result in satisfying halftone quality. Error diffusion which first was introduced in [1] produces halftoned images with higher quality than thresholding, but still generates different artifacts. Many studies have been conducted to reduce these artifacts and improve the quality of halftoned images using error diffusion.

While thresholding and the original error diffusion suffer from low halftone quality and several artifacts, the third category of halftoning methods, iterative halftoning, obtain the highest quality at the cost of computational complexity. Direct Binary Search (DBS) [2] and Iterative Method Controlling Dot Placement (IMCDP) [3] fall into this category.

Over the past few years, 3D printing has received a growing attention from industries and researchers. However, compared to 2D printing, 3D printing and 3D surface reproduction still require more development and progress. Numerous technologies and studies have emerged which paved the way toward creating complex 3D surfaces and reproducing realistic appearance. Among them, voxel-based printing technology, which enables full control over every single voxel, provides the possibility to create smoother surfaces with higher resolution. Voxel-level control of 3D prints enables high-quality surface and image reproduction in three-dimensional domain.

Several studies have been conducted to take advantage of the extensive knowledge and research in 2D printing in order to adapt 2D halftoning algorithms to 3D domain. Lou and Stucki

were among the first scientists who adapted 2D dithering and error diffusion to 3D domain [4]. In [5], 3D dithering has been applied to material composition as a halftoning approach. Error diffusion has been adapted to halftone layer based 3D surfaces in [6]. Brunton, et al. proposed a surface traversal algorithm to visit every voxel of a three-dimensional object. Then, they adapted 2D error diffusion to a 3D halftoning method, producing full color with voxel-based printing technology, compatible with translucent printing materials [7]. In [8], DBS algorithm is applied to 3D printing for the first time and redeveloped to halftone monochrome 3D surfaces. Michals, et al. proposed a three-dimensional extension of 2D tone dependent fast error diffusion. Their method seeks to achieve the halftone quality of iterative halftoning methods by using the concepts of DBS and error diffusion halftoning [9].

While 3D error diffusion halftoning suffers from several artifacts, the iterative methods can produce higher halftoning quality in 3D domain. It has been reported that 3D error diffusion generates radial and comb artifacts [9]. By adapting 2D search-based halftoning algorithms to 3D shapes, the quality of the halftones has been shown to increase significantly [8-9]. Improving halftoning algorithms and decreasing the artifacts is a substantial step to produce visually pleasant structures and high-quality appearance reproduction, which still needs more improvement in three-dimensional printing applications.

The 2D IMCDP halftoning method is a color channel dependent halftoning method which has been proved not only to produce higher quality halftones, but also to reduce the amount of ink needed to reproduce colors [3]. Later, the authors used predetermined and image-independent threshold matrices to increase the IMCDP speed by making it a point-by-point halftoning process [10]. Due to the high-quality halftoning results achieved by 2D IMCDP and the possibility to enhance the performance by making it a point-by-point halftoning algorithm, the aspects of extending the 2D IMCDP to 3D domain and adaptation to halftone three-dimensional surfaces are studied in this paper.

In the present work, we propose a three-dimensional extension of IMCDP. In the following section, we give a brief description of 2D IMCDP, then our proposed method and how the 2D IMCDP is extended to a 3D halftoning method is elaborated. Subsequently, we present and discuss the results of the proposed 3D IMCDP, where we also provide a comparison of the proposed method with 3D error diffusion. Finally, a conclusion of this article is given.

## 2D IMCDP

Iterative Method Controlling Dot Placement (IMCDP) halftoning was presented in [3]. In this method, the halftoning problem is defined as placing a predetermined number of dots in a blank image the same size as the original. IMCDP places dots

in the initially empty image in a fashion that the resulting halftoned image resembles the original.

It is assumed that the original continuous-tone image is scaled between 0 and 1, and is going to be converted to a binary image where 0 and 1 represent white and black, respectively. The absorbance is used to describe gray tone levels. An absorbance of 1 corresponds to black tone of ink on the paper and absorbance of 0 corresponds to white tone where no ink is applied on the paper.

The original image and the halftoned image should have the same average values over similar tonal regions because the perception of lightness/darkness is in direct proportion with the number of black dots in the halftoned image. As a consequence, the number of black dots in a tonal region in the halftoned image should equal the sum of the pixel values of the corresponding region in the original image. Considering this, the total number of dots to be placed in different tonal regions of the halftoned image is known in advance. Assume that the size of the original image is  $n \times n$  pixels, and pixels in the original image are holding an average gray tone value of  $p$  ( $0 \leq p \leq 1$ ). Then, the total number of dots which should be placed in the halftoned image would be  $pn^2$ .

IMCDP starts with a blank image, the same size as the original, and continues with searching over the whole original image for the pixel holding the maximum value, i.e. the darkest pixel. Once this pixel position is found, a "1" is placed in the corresponding position in the halftoned image, representing the darkest pixel. The impact of the placed dot is fed-back to the original image by a filter. As the human eye acts as a low-pass filter, it is reasonable to compare low-pass filtered versions of the original image and the halftoned image in each iteration. The low-pass filtered version of the original image is subtracted from the low-pass filtered version of the halftoned image in each iteration. Figure 1 shows the block diagram of the 2D IMCDP algorithm.

Experiments in [3] showed that a Gaussian filter with a standard deviation of 1.3 truncated to  $11 \times 11$  pixels leads to desired halftone results in most cases. Applying the filter to the  $11 \times 11$  neighboring pixels, centered at the pixel holding the maximum value, decreases the values of the neighborhood based on the distance from the center, hence reducing the probability of finding the next maximum in this area.

Next, the pixel with the second largest value is found in the modified reference image and the second dot is placed at its corresponding position in the halftoned image. In each iteration, the pixel with the largest value is replaced with a "1" in the target halftoned image and its effect is fed-back to the halftoning process. The process continues until all the predetermined number of black dots are placed and the halftoned image is created.

Since the algorithm is designed to search for the maximum pixel value at each iteration, a very small amount of noise is added to the original image before halftoning to avoid structures in areas with constant tone values.

IMCDP reproduces most of the gray tones quite well, but in highlight and shadow regions, i.e. in regions with gray tones lighter than 0.04 and darker than 0.96, the dots are not distributed as homogeneously as expected [3, 10]. The reason is that the  $11 \times 11$  filter is not big enough for these regions. Assume that the average distance between two dots, i.e. the principal wavelength, is  $a$ , and the average pixel value is  $p$ . To have a homogeneously halftoned image of size  $n \times n$  pixel, we then have:

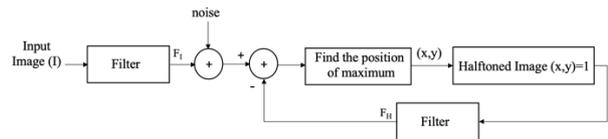


Figure 1. Block diagram of 2D IMCDP.

$$a^2 pn^2 = n^2 \quad (1)$$

Therefore, the average distance between the dots is:

$$a = \sqrt{1/p} \quad (2)$$

Consequently, the size of the filter would be  $(2a+1 \times 2a+1)$  [3]. In the regions with gray tones lighter than 0.04 and darker than 0.96, a filter with varying size is used and the principal wavelength corresponding to the gray tones decides the size of the filter in these regions [10].

Figure 2 presents the results of 2D IMCDP halftoning for different constants gray tones in patches of size  $120 \times 120$  pixels. A relatively low resolution of 100 dpi is used to ensure that dots and the halftone structure are clearly visible. A Gaussian filter with a standard deviation of 1.3 truncated to  $11 \times 11$  pixels is applied to the patches. As shown, IMCDP reproduces these gray tones well and the dots are homogeneously distributed over every patch.

IMCDP reproduces well-formed halftoned images by operating on the whole original image iteratively. But, iterative halftoning approaches usually suffer from high computational cost and slow performance when the input image is large [10]. In [10], predetermined image-independent threshold matrices are introduced to make IMCDP a point-by-point method resulting in a faster approach. Generating in-built thresholding matrices makes the halftoning process quick. Moreover, it enables the user to have control over the halftone structure, cluster dot size, shape, and alignment [10].

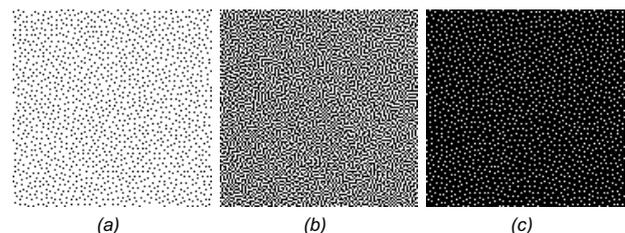


Figure 2. Samples of patches halftoned by 2D IMCDP at different absorbance: (a): 0.1, (b): 0.5, (c): 0.9.

### 3D IMCDP

Halftoning 2D images using IMCDP has been proposed in [3] and briefly described in the previous section. As 2D IMCDP has proven to produce high quality halftones and further been extended to a point-by-point halftoning process [10], adapting it to 3D domain would be valuable and the main focus of this work is to evaluate this adaptation. In adapting 2D IMCDP to 3D domain, we will keep the original pipeline of the algorithm, but to adapt it to 3D shapes, a modified method is used for applying the Gaussian filter. We refer to this extension of 2D IMCDP as 3D IMCDP halftoning. Our main concern about 3D IMCDP is to find an approach which requires slight changes to adapt 2D IMCDP to 3D. Furthermore, the 3D extension of IMCDP should

result in the same output as 2D IMCDP in regions where the 3D shape mimics a 2D surface.

The input to the three-dimensional reproduction process is a 3D shape. The 3D shape is first sliced horizontally, parallel with the x-y plane, and then voxelized. During voxelization, a regular grid of voxels is created. Voxels are divided into three types: exterior, surface, and interior voxels [7]. In the present work, we only consider surface voxels. Each voxel is defined with an array of elements holding the cartesian coordinates of its center. Surface voxels are assigned with gray tone values. The algorithm takes the coordinates and gray tone value of the voxels as input and produces the halftoned data for each coordinate as output.

After voxelization and assigning tonal values to the surface voxels, the 3D IMCDP halftoning starts by finding the position of the voxel holding the maximum value. Then, according to the original 2D IMCDP, a Gaussian filter should be applied to the neighboring voxels. In a 3D shape, this means, the Gaussian filter is subtracted from the surface voxels within a box of size  $m \times m \times m$  centered at the voxel with the maximum value (O). This subset of neighboring voxels is denoted by N and all surface voxels of the 3D shape which fall into this area are called neighbors ( $V_i \in N$ ). In 2D IMCDP, an  $m \times m$  filter covers exactly  $m \times m$  number of pixels. However, due to the geometrical characteristics of a 3D shape, the total number of surface voxels which are within the neighbors' subset and are used in calculations, varies with its location on the shape. Figure 3 shows examples of neighbors ( $V_i$ ) at different locations of a sphere.

The Gaussian kernel in 3D domain has the following form:

$$G(x_i, y_i, z_i) = K e^{-\frac{R(x_i, y_i, z_i)^2}{2\sigma^2}} \quad (3)$$

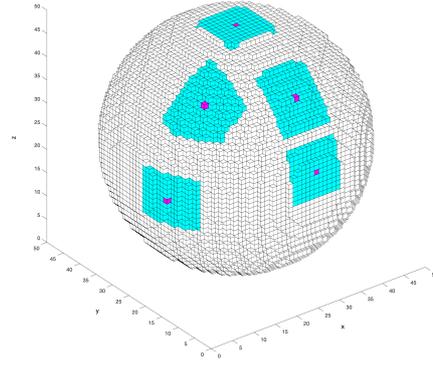
K is the normalization constant and comes from the fact that the sum of the filter elements equals to 1. R is the three-dimensional distance of each filter element from the central point (O) and is calculated based on Equation 4:

$$R(x_i, y_i, z_i) = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} \quad (4)$$

$(x_0, y_0, z_0)$  is the coordinate of the voxel with maximum value, located at the central point of surface voxels (O), and  $(x_i, y_i, z_i)$  is the coordinate of a neighboring voxel within the neighbors' subset ( $V_i \in N_i$ ).

As demonstrated in Figure 1, after finding the voxel with maximum value, a 1 (black dot) is placed at the location of the voxel with the maximum value on the empty shape. Next, the filter element corresponding to each neighboring voxel is computed using equation 3. Then, the filter elements are subtracted from the corresponding neighboring voxels' values within the filter area surrounding the maximum value voxel. The halftoning continues with finding the next maximum until the predetermined number of dots are placed and the final halftoned 3D shape is achieved.

Halftoning 3D surfaces based on Equation 3 (referred to as Gauss method in this paper) reproduces gray tones quite well. However, according to our experiments, circular artifacts might occur in some tonal regions. Figure 4 illustrates the problem of circular effects. This problem is derived from the geometrical characteristic of the voxelized 3D shape and the Gaussian filter

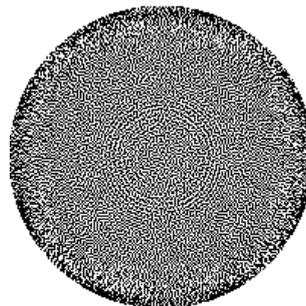


**Figure 3.** The surface voxels within a box of size  $m \times m \times m$  on different positions of a sphere with the radius of 25. In this case,  $m=11$  and the total number of voxels covered by the 3D filter varies in the range of [97, 130].

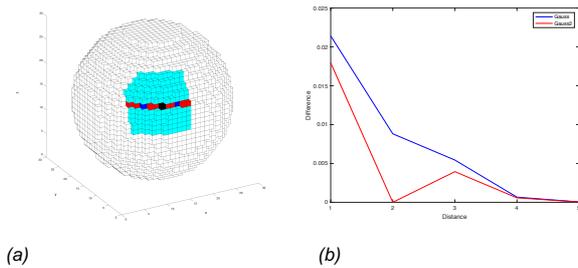
used.

Consider the situation shown in Figure 5a. A Gaussian filter of size  $11 \times 11$  is applied on a neighborhood centered at the black voxel located at  $(x_0, y_0, z_0) = (10, 2, 20)$ . Two neighboring voxels located at  $(x_1, y_1, z_1) = (13, 2, 20)$  and  $(x_2, y_2, z_2) = (7, 4, 20)$  are colored in blue. These two voxels' Euclidean distances to the central voxel according to Equation 4 are  $\sqrt{9}$  and  $\sqrt{13}$ , respectively. Using Equation 3, the normalized filter elements for these two voxels would be 0.0078 and 0.0024, respectively. As one can see, these two neighboring voxels are both located three-voxel away from the centered voxel. Comparing with the weights in a 2D filter, these two voxels should receive similar weights. However, due to the geometrical characteristic of the 3D shape, their three-dimensional distances are different and these two voxels will not receive similar weights. Calculating the filter elements for these two neighboring voxels using the Gauss method for a 3D shape results in values which differ a lot. This big difference between filter elements causes circular patterns. Hence, decreasing the difference will reduce the circular effects.

To improve the algorithm and remove the circular patterns, we propose to change the way of assigning weights and we proceed as follows: First, the voxels in the specified neighborhood ( $V_i \in N$ ) are sorted based on their three-dimensional distance from the central voxel (O), in ascending order. Assuming that the size of the filter is  $m \times m$ , if the total number of voxels in the filter is greater than  $m \times m$ , only the first  $m \times m$  voxels are included. This means that voxels which are very far from the central voxel are ignored. This would not cause any error since the calculated weights for these elements are close to zero. If the number of total voxels is less than  $m \times m$ , only the



**Figure 4.** Circular effects in the 2D x-y view (top view) of a sphere with radius of 100 and absorptance of 0.5, halftoned by Gauss method.

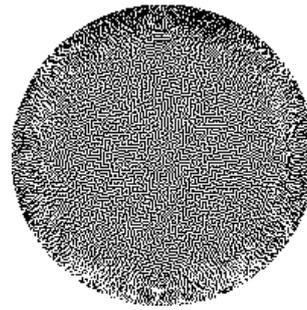


**Figure 5.** (a): The Gaussian filter applied on a sphere with the radius of 15. (b): comparison of Gauss and Gauss2 approach in assigning filter elements to the voxels.

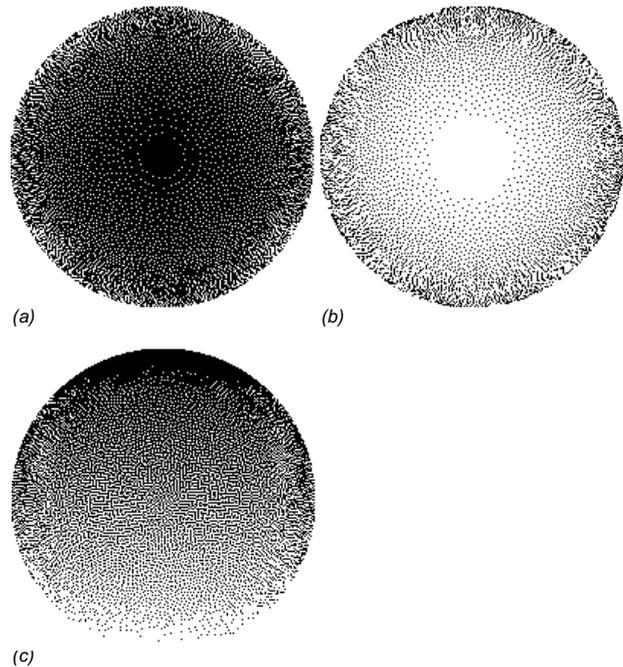
available voxels are included in the calculations. Next, a two-dimensional Gaussian function is defined, which consists of  $m \times m$  weights. Weights are sorted in descending order. In the next step, the largest weight is assigned to the closest voxel to the under-operation voxel and continuing so, the smallest weight is assigned to the voxel which has the maximum distance from the central voxel. Now, the 3D Euclidean distances still decide on assigning the weights from a 2D Gaussian filter to the corresponding voxels, but they are not directly used for calculating the weights. The improved method will be denoted as Gauss2 in this paper. Figure 5b shows how symmetrical the filters created by Gauss and Gauss2 methods for the ten colored voxels in Figure 5a are. Considering this row of voxels in Figure 5a, there are five voxels to the left and five voxels to the right of the central voxel. The absolute value of the difference between the weights being assigned to the right voxel and the weight being assigned to its counterpart to the left is calculated. The difference for each two voxels equally far from the central voxel shows how symmetric the filter is. The smaller the difference, the more similar the corresponding weights. According to this graph, the difference between filter elements calculated for neighboring voxels, which are located at the same distance from the central point on the surface, are smaller when the improved method, Gauss2, is used. This illustrates that Gauss2 outperforms Gauss in generating identical weights for the equally important voxels. Figure 6 shows the projection of a sphere with radius 100 and a constant absorptance of 50% halftoned using Gauss2. According to this figure, Gauss2 works well and comparing it with the same sphere halftoned with Gauss in Figure 4, verifies that circular effects are removed.

To evaluate the performance of our improved algorithm for different absorptances, we applied Gauss2 to a sphere with initial voxel values of different absorptances defined as a ramp. The voxels in the first slice are assigned with the lightest tone and as the shape winds up toward the top of the sphere, the voxel values gradually increase. This means that the voxels in the last slice are assigned with the darkest tone. All voxels in the same slice hold the same absorptance. Figure 7 shows three 2D projections of the halftoned sphere. The algorithm performs quite well for most of the absorptances, but according to Figure 7a and 7b, dot placement in highlight and shadow regions, i.e. regions with very low and very high absorptance is not satisfying.

As a matter of fact, highlights and shadows are more sensitive to changes in gray tones and there is no control over the number of dot placements in these critical regions. To solve the problem and to have a more precise control over dot placement in highlights and shadows, we define control regions for these areas. We choose ten control regions in highlight areas, i.e. areas



**Figure 6.** 2D x-y view (top view) of a sphere with radius of 100 and absorptance of 0.5, halftoned by Gauss 2 method.

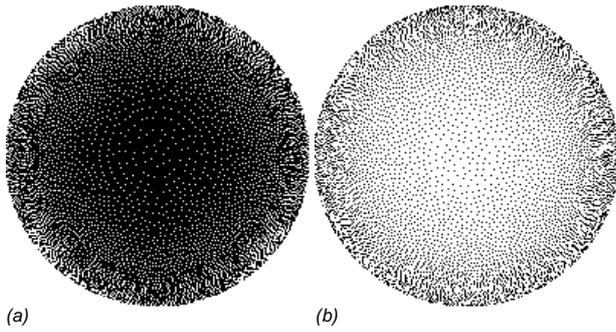


**Figure 7.** 2D views of a sphere with radius of 100 with ramp gray tone input of  $0.01 \leq \text{absorptance} \leq 0.99$ , halftoned by Gauss2 method. (a): x-y top view, (b): x-y bottom view, (c): x-z view.

with absorptance less than 0.04 and ten control regions in shadows, i.e. areas with absorptance greater than 0.96 [3]. The total number of dots to be placed in each control region is determined in advance based on the corresponding regional absorptance. Hence, in these regions, the algorithm will continue until the predetermined number of black dots is placed in each area. As can be seen in Figure 8, after adding the control regions, dot placement for all the gray tones works properly.

In halftoning, it is very important to have a symmetric dot placement in highlights and shadows. Thereby, the dot placement should be symmetric for both sides of the mid-tone level (50%) in a ramp of gray tones. For instance, the black dots at 30% should have the same structure as the white dots at 70%.

According to the 2D projection of the halftoned sphere in highlight and shadow regions in Figure 8, white dots and black dots are not distributed symmetrically. That means our approach cannot generate symmetric halftone structure in all gray tone regions, we proceed as follows: In each iteration, first, the location of the voxel with maximum value, in light tone regions ( $\text{absorptance} \leq 0.5$ ) and the voxel with minimum value, in dark



**Figure 8.** 2D views of a sphere with radius of 100 and initial voxel values as a ramp, halftoned by Gauss2 with control regions method. (a): x-y top view, (b): x-y bottom view.

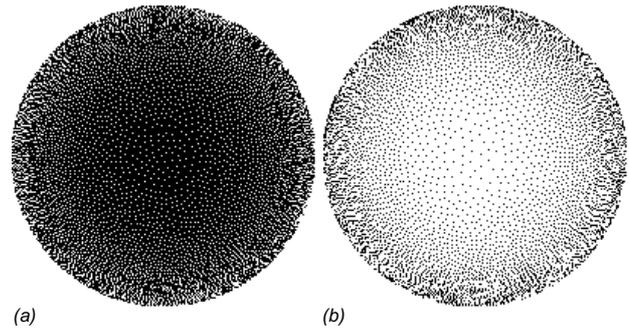
tone regions (absorbance  $> 0.5$ ), are found. Then, a 1 is placed at the maximum voxel position and a 0 is placed at the minimum voxel position in the initial blank 3D shape. To ensure that the voxels at these positions would not be found as the maximum or the minimum in the next iterations, a large negative number is placed at the position of the maximum voxel value and a large positive number is placed at the position of the minimum voxel value. Then, the algorithm continues with calculating the filter elements and the feedback process, i.e. subtracting (or adding) the filter elements from (or to) the neighboring voxels. The process continues until the predetermined number of 1s and 0s are placed in light and dark tone regions, respectively. As illustrated in Figure 9, when the algorithm terminates, a symmetric structure is achieved in both sides of the gray tone ramp.

After this additional algorithm improvement, the 3D IMCDP produces symmetric and homogeneous halftone structure for every gray tone.

## Results and Discussion

To illustrate the performance of the proposed 3D IMCDP halftoning, the algorithm is applied to a voxelized sphere. The simulations are performed in MATLAB. Figure 10 shows a sphere of radius 150 halftoned with our proposed 3D IMCDP algorithm for a ramp gray tone input with absorbance values between 0.01 and 0.99. Based on our experiments, a Gaussian filter of size  $11 \times 11$  with standard deviation within the range of  $1.3 < \sigma < 1.7$  results in visually pleasant outputs with minimal artifacts. For mid-tone gray tone, a standard deviation of 1.3 is applied and for highlight and shadow regions, it gradually increased to 1.7. According to Figure 9 and 10, the algorithm reproduces the gray tones well, dots are placed homogeneously, and halftone structure is symmetric.

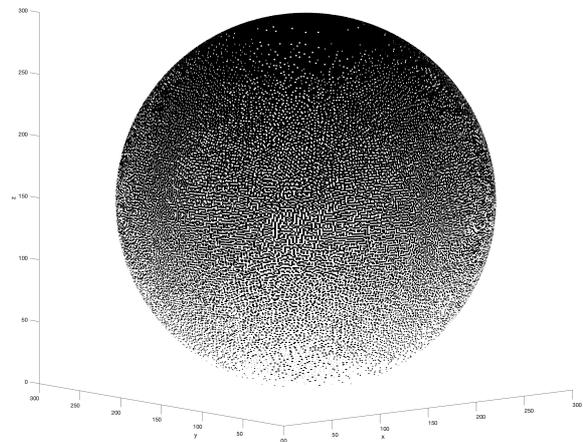
The results of the 3D extension of the IMCDP is also compared with the 3D version of Floyd-Steinberg error diffusion. Implementation of the 3D error diffusion has been done by following the traversal algorithm proposed in [7]. Figure 11 illustrates a half-sphere of radius 150 which is halftoned using 3D IMCDP and 3D Floyd-Steinberg error diffusion. Results demonstrate that error diffusion causes more visible artifacts on specific regular longitude and latitude, which are clearly removed using 3D IMCDP algorithm. Moreover, the halftone structure generated by 3D IMCDP is apparently more homogeneous. The radial artifacts visible in Floyd-Steinberg error diffusion halftoning in Figure 11 are in accordance with what the authors have reported about their observation of artifacts in 3D error diffusion in [9].



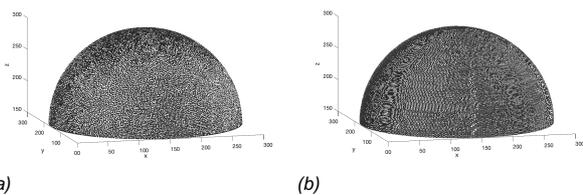
**Figure 9.** 2D views of a sphere with radius of 100 and initial voxel values as a ramp, halftoned by Gauss2 method after adding control regions and finding a maximum and a minimum in each iteration. (a): x-y top view, (b): x-y bottom view.

Finally, to evaluate the performance of the proposed algorithm, an image is mapped by planar mapping on the sphere and then halftoned. As can be seen in Figure 12, the algorithm reproduces all the gray tones very well.

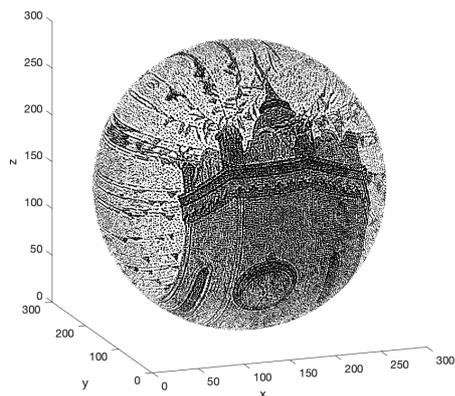
In order to test whether the 3D IMCDP is performing equivalent to the 2D IMCDP, the halftoned 3D shape is projected on the x-z plane. It is expected that the halftoning structures of this projection results in an almost similar structure as 2D IMCDP halftoning. Figure 13 presents this projection. Disregarding the outer artifacts, which are derived from projecting the 3D shape on a 2D plane, we observe that the 3D extension of IMCDP produces different gray tones well and the dot placement is homogeneous. Halftone structure and quality obtained by applying 3D IMCDP is quite similar to what is resulted from 2D IMCDP halftoning in Figure 2.



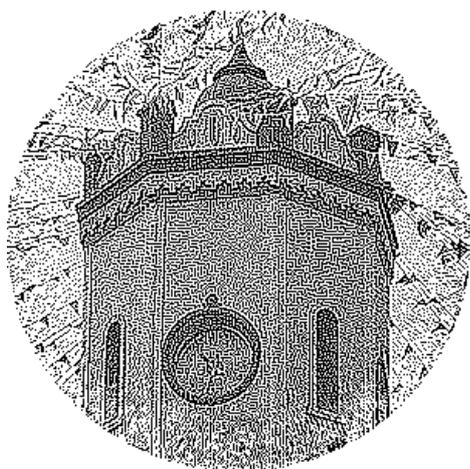
**Figure 10.** 3D IMCDP halftoned sphere of radius 150 with ramp gray tone input of  $0.01 \leq \text{absorbance} \leq 0.99$ .



**Figure 11.** Comparison of error diffusion and IMCDP halftoning on a half-sphere of radius 150 with a constant input gray tone of 0.5 absorbance. (a): 3D IMCDP, (b): 3D error diffusion.



**Figure 12.** Result of halftoning an image mapped on to the sphere with radius of 150.



**Figure 13.** Projection of the sphere of Figure 12 on x-z plane.

## Conclusion

In this paper, an extension of IMCDP halftoning method has been proposed, allowing IMCDP to halftone 3D shapes. Due to the fact that highlight and shadows are more sensitive to changes in gray tones, we added control regions to the algorithm to place dots more accurately in these critical regions. In order to generate symmetric halftone structures, the concept of finding maximum and minimum voxel values in each iteration is introduced. As a result, the proposed algorithm produces homogenous and symmetric halftone structures for every gray tone. The 3D halftoning results in high-quality halftones closely resembling the results of 2D IMCDP.

## Acknowledgments

This work has received funding from the European Union's Horizon 2020 programme under the grant No. 814158.

## References

- [1] R. W. Floyd and L.S. Steinberg, An adaptive algorithm for spatial gray-scale, Proc. Soc. Inf. Disp, pg. 75. (1976).
- [2] M. Analoui and J. P. Allebach, Model-based halftoning using direct binary search, Proc. Human vision, visual processing, and digital display III, pg. 96. (1992).
- [3] S. Gooran, "Dependent color halftoning: Better quality with less ink," J. Imaging Science and Technology. 48, 4 (2004).
- [4] Q. Lou and P. Stucki, Fundamentals of 3D halftoning, Proc. International Conference on Raster Imaging and Digital Typography, pg. 224. (1998).
- [5] W. Cho, E. M. Sachs, N. M. Patrikalakis, and D. E. Troxel, "A dithering algorithm for local composition control with three-dimensional printing," J. Computer-Aided Design. 35, 9 (2003).
- [6] C. Zhou and Y. Chen, "Three-dimensional digital halftoning for layered manufacturing based on droplets," J. Transactions of the North American Manufacturing Research Institution of SME. 37 (2009).
- [7] A. Brunton, C. A. Arıkan, and P. Urban, "Pushing the Limits of 3D Color Printing: Error Diffusion with Translucent Materials," J. ACM Trans. Graph. 35, 1 (2015).
- [8] R. Mao, U. Sarkar, R. Ulichney, and J. P. Allebach, "3D Halftoning," J. Electronic Imaging. 2017, 18 (2017).
- [9] A. Michals, J. Liu, A. Jumabayeva, Z. Li, and J. P. Allebach, "3D Tone-Dependent Fast Error Diffusion," J. Electronic Imaging. 2019, 4 (2019).
- [10] S. Gooran and B. Kruse, "High-speed first- and second-order frequency modulated halftoning," J. Electronic Imaging. 24, 2 (2015).

## Author Biography

*Fereshteh Abedini received her M.Sc. in Digital Electronics Engineering from Amirkabir University of Technology, Iran (2015). Currently, she is pursuing her Ph.D. studies at Media and Information Technology division, Linköping University, Sweden. Her research interest includes digital halftoning, digital image processing, image reproduction, 3D halftoning and appearance reproduction.*

*Sasan Gooran received his M.Sc. in Computer Science and Engineering (1994) and his Ph.D. in Media Technology (2001), both from Linköping University, Sweden. He is currently an associate professor at Media and Information Technology division, Linköping University. His research has mainly focused on digital imaging, digital halftoning, color and image reproduction, 3D halftoning and appearance printing. He has taught courses in, among others, Graphic technology, Computer graphics, Image processing, Numerical methods, Transform theory and Mathematics.*

*Daniel Nyström received his Ph.D. in Media Technology from Linköping university (2009). He is currently an assistant professor at Media and Information Technology division, Linköping University. His research interests include multispectral imaging and spectral reproduction, paper optics, color prediction and color separation models in halftone reproduction, color science, and appearance reproduction.*