https://doi.org/10.2352/ISSN.2470-1173.2020.17.3DMP-034


# Variable Precision Depth Encoding for 3D Range Geometry Compression

***Matthew G. Finley and Tyler Bell****

***Department of Electrical and Computer Engineering, University of Iowa; Iowa City, Iowa 52242, USA***
* ***tyler-bell@uiowa.edu***

## Abstract

*This paper presents a novel method for accurately encoding 3D range geometry within the color channels of a 2D RGB image that allows the encoding frequency—and therefore the encoding precision—to be uniquely determined for each coordinate. The proposed method can thus be used to balance between encoding precision and file size by encoding geometry along a normal distribution; encoding more precisely where the density of data is high and less precisely where the density is low. Alternative distributions may be followed to produce encodings optimized for specific applications. In general, the nature of the proposed encoding method is such that the precision of each point can be freely controlled or derived from an arbitrary distribution, ideally enabling this method for use within a wide range of applications.*

## Introduction

Modern 3D range imaging techniques have made available 3D scanning devices that can capture high-resolution, high-accuracy 3D data at real-time to kHz speeds. As the resolution, accuracy, and acquisition speed of 3D scanning devices has increased, so have their potential applications in a variety of areas, such as medicine, manufacturing, homeland security, and forensics. One aspect that may impede the adoption and practical usage of such devices, however, is the amount of data they can generate.

Conventionally, one method of storing 3D range data is within 3D mesh formats such as OBJ, STL, or PLY. These formats represent a mesh by storing 3D coordinate data along with information specifying how the coordinates should be connected. In addition to storing geometry, a mesh format may also specify additional information, such as surface normals or texture coordinates. Although providing a generic method of storage for 3D geometries, mesh file formats typically need large amounts of data to be stored. Given this, methods for compressing 3D range data has been explored.

One such method is representing 3D range geometry within the color channels of a conventional 2D image. In the last decade there have been many methods proposed to use principles of phase-shifting to encode floating point 3D geometry within 8-bit color channels of a 2D image [1, 2, 3, 4, 5, 6, 7, 8]. Once in a 2D format the encoded 3D geometry can be compressed with traditional 2D image compression techniques, such as PNG.

In each of the above encoding methods, a user-defined encoding constant is selected that defines how many encoding periods with which to encode the 3D geometry. This constant is ultimately used to determine the encoding frequency of the geometry. In general, as the number of encoding periods increase, the encoding frequency also increases. As a result, the encoding precision will typically increase, but it brings along larger file sizes in the compressed 2D image. Conversely, a smaller number of encoding periods can be used to achieve smaller compressed file sizes, but this may lead to greater loss of precision.

This paper presents a novel method that allows for a variable encoding frequency to be used to encode 3D range geometry within a 2D image. The proposed method thus allows for the encoding precision to be determined, per-coordinate, as a function of the depth data being encoded. For example, the method can be used to encode geometry along a normal distribution, dynamically increasing the encoding frequency—and thus encoding precision—in areas of high coordinate density and dynamically decreasing the encoding frequency—and thus file size—in areas of lower coordinate density. Based on the desired tradeoff between file size and encoding precision, alternative distributions may be used, ideally enabling this method to be employed within a wide range of applications. The remainder of this paper will describe the principles of the proposed encoding method and will report on its experimental performance.

## Principle

### *Multiwavelength Depth Encoding*

In the multiwavelength depth encoding method [5], principles of phase-shifting are used to encode a floating-point depth map, $Z$, into the three color channels of a regular 2D image. This encoding approach is mathematically described as

$$I_1(i,j) = \frac{1}{2} + \frac{1}{2}\sin\left(2\pi \times \frac{Z(i,j)}{P}\right), \tag{1}$$

$$I_2(i,j) = \frac{1}{2} + \frac{1}{2}\cos\left(2\pi \times \frac{Z(i,j)}{P}\right), \tag{2}$$

$$I_3(i,j) = \frac{Z(i,j)}{\text{Range}(Z)}. \tag{3}$$

In Eqs. (1)-(2), the depth map $Z$ is encoded within two continuous functions at a frequency determined by a user-defined *fringe width*, $P$. Physically, $P$ can be described as the depth distance, in the range of $Z$, that is encoded within each period of Eq. (1) and Eq. (2). For example, to encode a depth map $Z$ into four equal periods, $P$ can be defined as $P = \text{Range}(Z)/n$ where $Range()$ returns the valid depth range of $Z$ and where $n = 4$. As

previously mentioned, when the encoding frequency increases, the encoding precision also increases (along with the resulting file sizes). Mathematically it can be seen that since $P$ is a constant value, applied to the encoding of each depth value $Z(i,j)$, the encoding precision is fixed for the entire depth map.

The outcome of using a fixed encoding frequency is that all data within the depth map is encoded at the same precision. In order to more precisely compress specific regions of interest within a depth map $Z$, the encoding frequency can be increased, but this results in larger file sizes since regions of less interest must also be encoded at the increased precision. Practically, it may be beneficial for the encoding frequency to be defined independently for each $Z(i,j)$, thus allowing the encoding precision of each $Z(i,j)$ to be uniquely determined. The next section will describe the principle of this paper's proposed variable precision encoding.

### Variable Precision Encoding

This paper presents a novel method for the variable precision encoding of floating point 3D range geometry into the 8-bit color channels of a traditional 2D image. As previously established in Eqs. (1)-(2), the parameter $P$ is a fixed, user-defined fringe width that determines the overall frequency, and thus precision, of the encoding. In the proposed method, however, this parameter will now be defined as $P(i,j)$ so that it can be independently determined for each depth coordinate, $Z(i,j)$. It should be noted that $P(i,j)$ may be generated however necessary to meet a user's needs. For example, in order to encode with highest precision where the density of 3D coordinates is high, this paper chose to generate $P(i,j)$ using a normal distribution centered around the highest concentration of data points via the following method.

First, a normal distribution through the depth range of $Z$ is defined as

$$N(z) = \text{Norm}(z \mid \mu, \sigma^2), \tag{4}$$

where $\mu$ and $\sigma$ are the mean and standard deviation of the depth values of the $Z$ to be encoded and *Norm* is a function that outputs the value of the normal probability density function (PDF), based on those parameters, for a given depth value, $z$. This PDF can be mapped to the geometry to be encoded by determining the output of $N(z)$ for each depth value, $Z(i,j)$. However, as the mapping will need to be regenerated at the time of decoding, values of $Z$ cannot be used in $N(z)$ directly as $Z$ is not available to the decoding process (as it is the information being decoded). Although $Z$ is not known at the time of decoding, the encoding $I_3$, stored in one of the 2D image's color channels using Eq. (3), will be available to the decoding process. Since $I_3$ is simply an encoding storing the normalized depth map, which can be rescaled to be equivalent to $Z$, the primary difference between $I_3$ and the depth map $Z$ comes when $I_3$ is stored in a conventional image format. For example, when $I_3$ is stored in a PNG image, it will be scaled and quantized into 8 bits per pixel. This loss of precision means that $Z$ cannot be directly recovered from the quantized depth map; however, a rescaled $I_3$ may still be used to approximate the depth map, $Z$. Given this, and the fact that $I_3$ is known to both the encoding and decoding processes, an approximate mapping of the PDF to the geometry to be encoded is given by determining $N(z)$ for each coordinate in the quantized, normalized depth map, $I_3'$. Here, $I_3'$ is simply $I_3$ scaled and quantized to 8 bits per pixel since the PNG image format is used for the experiments in this paper. Mathematically, the approximate mapping of the distribution to the geometry to be encoded can be described as

$$D(i,j) = N(I_3'(i,j)). \tag{5}$$

This distribution map can then be normalized such that $\overline{D}$ is $D$ scaled between 0 and 1. The normalized distribution map can then be scaled between two user-defined constants $n_{min}$ and $n_{max}$ that will ultimately determine the minimum and maximum number of periods, and thus the minimum and maximum precision, with which to encode the depth data. The scaled distribution map can be described as

$$D_s(i,j) = \overline{D}(i,j) \times (n_{max} - n_{min}) + n_{min}. \tag{6}$$

The per-pixel fringe width, $P(i,j)$, can be determined from $D_s$ via

$$P(i,j) = \text{Range}(Z)/D_s(i,j). \tag{7}$$

This variable fringe width can then be used to generate variable, high-frequency encodings of the 3D geometry via

$$I_1(i,j) = \frac{1}{2} + \frac{1}{2}\sin\left(2\pi \times \frac{Z(i,j)}{P(i,j)}\right), \tag{8}$$

$$I_2(i,j) = \frac{1}{2} + \frac{1}{2}\cos\left(2\pi \times \frac{Z(i,j)}{P(i,j)}\right). \tag{9}$$

Finally, the encodings $I_1$, $I_2$, and $I_3$, generated from Eqs. (8), (9), and (3), respectively, can be placed into the color channels of a regular 2D RGB image and compressed with traditional image compression techniques, such as PNG.

Figure 1 illustrates the proposed variable-frequency encoding process on an ideal hemisphere with a radius of 256 mm. Figure 1a is the 3D geometry to be encoded. Figure 1b is the 2D depth map representation. Figure 1c is the normal probability density function N(z) described in Eq. (4). Figure 1d is the result of mapping the PDF onto a quantized version of the depth map in Fig. 1b and then scaling between $n_{min}$ and $n_{max}$ via the process described in Eq. (5) and Eq. (6), respectively. Figures 1e-1g are the encodings $I_1$, $I_2$, and $I_3$ given by Eqs. (8), (9), and (3), respectively. Figure 1h is the $512 \times 512$ output image when $I_1$ is stored in the red channel, $I_2$ is stored in the green channel, and $I_3$ is stored in the blue channel of a compressed PNG image.

### Variable Precision Decoding

As discussed in the previous section, before the encoded geometry $Z$ can be recovered it is first necessary to regenerate the distribution map and the variable fringe width $P(i,j)$ for each point of geometry. Due to the scaling and 8-bit quantization of $I_3$ when it was stored within a channel of the output image, it can be noted that the $I_3$ accessible to the decoding process is equivalent to the $I_3'$ that was used to generate the approximate distribution mapping during the encoding process. Given this, the normal distribution through the depth range of $Z$ can be computed via Eq. (4), and then a distribution map can be approximated via the quantized depth map $I_3$ via

$$D'(i,j) = N(I_3(i,j)) = N(I_3'(i,j)) = D(i,j). \tag{10}$$

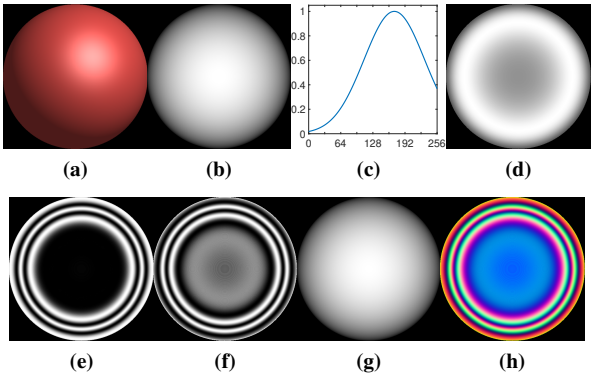



**Figure 1.** The proposed variable precision depth encoding process. (a) original 3D geometry to be encoded, in this case an ideal hemisphere with a radius of 256 mm; (b) 2D depth map representing (a); (c) assumed normal distribution given the mean and standard deviation of data in (b); (d) scaled distribution map, $D_s$, derived from (b) and (c); (e)-(g) $I_1$-$I_3$ encodings of (b) given by Eqs. (8), (9), and (3), respectively; (h) encoded color image which is the output of the proposed method.

The distribution map $D'$ can then be scaled and used to recover $P(i,j)$ following Eqs. (6) and (7), respectfully.

With $P(i,j)$ recovered, $Z$ can begin to be decoded by finding the wrapped phase of the two higher frequency encodings by

$$\phi_{HF}(i,j) = \tan^{-1}\left(\frac{I_1(i,j)-0.5}{I_2(i,j)-0.5}\right). \tag{11}$$

The wrapped phase of the single lower-frequency encoding can be directly scaled from $I_3$ to lie between $-\pi$ and $\pi$ as

$$\phi_{LF}(i,j) = I_3(i,j)\times 2\pi - \pi. \tag{12}$$

The continuous, low-frequency encoding stored in $I_3(i,j)$ can be used to determine the number of $2\pi$ discontinuities that must be subtracted or added to each pixel of the discontinuous, high-frequency wrapped phase, $\phi_{HF}(i,j)$, by computing a stair map via

$$K(i,j) = \text{Round}\left(\frac{\phi_{LF}(i,j)\times \text{Range}(Z)/P(i,j) - \phi_{HF}(i,j)}{2\pi}\right). \tag{13}$$

This stair map, $K(i,j)$ can then be used to unwrap $\phi_{HF}(i,j)$ via

$$\Phi(i,j) = \phi_{HF}(i,j) + 2\pi\times K(i,j). \tag{14}$$

Finally, the encoded depth map $Z$ can be recovered from the unwrapped phase as

$$Z(i,j) = \frac{\Phi(i,j)\times P(i,j)}{2\pi}. \tag{15}$$

Figure 2 illustrates the proposed variable precision depth decoding process. Figure 2a shows the wrapped phase $\phi_{HF}$ given by Eq. (11). Figure 2b shows the scaled distribution map, $D_s$, recovered from $I_3$. Figure 2c shows the recovered stair map, $K$. Figure 2d is the recovered unwrapped phase map, $\Phi$, as solved by Eq. (14). Finally, Fig. 2e and Fig. 2f show the recovered depth map $Z$ and its 3D rendering, respectively.

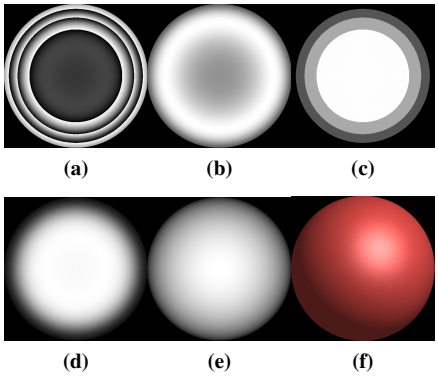



**Figure 2.** The proposed variable precision depth decoding process. (a) wrapped phase $\phi_{HF}$ recovered from the encodings $I_1$ and $I_2$; (b) the scaled distribution map, $D_s$, recovered from the quantized depth map, $I_3$; (c) recovered stair map, $K$, as solved by Eq. (13); (d) unwrapped phase $\Phi$ from (a) and (c) solved by Eq. (14); (e) recovered depth map $Z$; (f) recovered 3D geometry.

## Experiments

The proposed variable precision depth encoding method was evaluated with several experiments. In the first set of experiments an ideal hemisphere with a radius of 256 mm was used as the 3D geometry to be compressed into a $512\times 512$ 2D image, as shown in Fig. 1h. As discussed in the Principle section, the encodings $I_1$, $I_2$, and $I_3$—given by Eqs. (8), (9), and (3)—are stored into the red, green, and blue channels of an output 2D image. For all experiments, traditional PNG compression is applied to the 2D images to further reduce file sizes of the encodings.

Figure 3 shows the results of the proposed variable precision encoding when it followed the normal distribution of the ideal hemisphere's depth data. In this experiment, $[n_{min}, n_{max}] = [2,6]$ meaning that each point $Z(i,j)$ in $Z$ was encoded at some precision between two periods and six periods, depending on each point's location in the distribution of data per Eq. (4) and Eq. (5). In Fig. 3, Fig. 3a is the original ideal hemisphere (with a radius of 256 mm), and Fig. 3b is the recovered geometry when Fig. 3a was encoded with the proposed method with $[n_{min}, n_{max}] = [2,6]$. Figure 3c shows the absolute error map (in mm) between the orig-

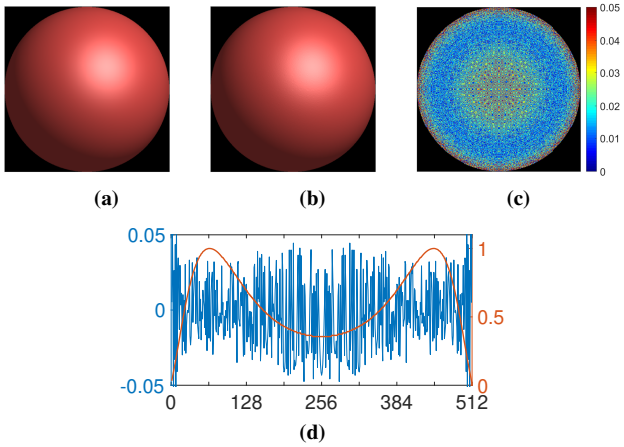



**Figure 3.** Encoding an ideal hemisphere with a radius of 256 mm and $[n_{min}, n_{max}] = [2,6]$. (a) original geometry; (b) recovered geometry; (c) absolute error map, in mm, between (a) and (b); (d) error overlaid with distribution map, $\overline{D}$, for the 256th row.

inal and recovered depth maps. Lastly, Fig. 3d plots the error of the 256th row along with the 256th row of the normalized distribution map, $\overline{D}$. It can be seen in Fig. 3d that there is an inverse relationship between the reconstruction error and the normalized distribution. As the normalized distribution increases, the number of periods used to produce the encoding will increase toward $n_{max}$, and thus the error will decrease. Conversely, when the normalized distribution decreases, the number of periods used to encode that data will decrease toward $n_{min}$, and thus the error will increase. This relationship can also be seen visually between Fig. 2b and Fig. 3c: where there is a greater amount of data in the distribution (as shown by the regions of higher intensity) in Fig. 2b there is a lower amount of error in Fig. 3c, and vice versa.

The results above verify that the proposed variable frequency encoding can indeed be used to more precisely store geometry in more dense regions, while reducing the precision of the encoding in less dense regions, of a depth map. Figure 4 presents the results of another experiment in which the variable frequency encoding between $[n_{min}, n_{max}] = [2,6]$ is compared to fixed $P$ encodings when $n = 2$ and $n = 6$. The first row of Fig. 4 shows the distribution of the parameter $n$ through the depth range of the same ideal hemisphere shown previously. It can be seen that in Fig. 4a and Fig. 4c the distribution is uniform through the depth range, but Fig. 4b shows the values of $n$—between $[n_{min}, n_{max}] = [2,6]$—along the scaled normal distribution which was determined by using the mean ($\mu = 170.7$) and standard deviation ($\sigma = 60.3$) of the hemisphere's depth values. The second row of Fig. 4 provides the absolute error map of the reconstruction error, when compared to the original geometry, for each distribution in the first row. The third row gives the reconstruction error values along the 256th row of the recovered geometry for each distribution in the first row. As can be seen, the proposed variable precision encoding method's reconstruction error more closely represents error from the uniformly distributed encoding where $n = 6$.

In addition to encoding at precisions similar to $n = 6$, the benefit of this method can be seen when the resulting file sizes are addressed. Table 1a provides the RMS error of the reconstructed hemisphere as well as the file size, in kilobytes, of the encoded output image when stored in the PNG format. Due to encoding the hemisphere along a normal distribution, the proposed method is able to achieve reconstruction accuracies similar to the uniformly distributed case where $n = 6$ without an equivalent increase in file size. To better illustrate the relationship between reconstruction accuracy and file size, Table 1b provides the reduction in RMS error per kilobyte compared to the uniformly distributed case when $n = 2$. This table highlights the fact that the proposed method of variable precision encoding with $[n_{min}, n_{max}] = [2,6]$ is able to achieve 50% greater reduction in error per kilobyte of file size when compared to the case where the hemisphere was encoded with a uniform distribution with $n = 6$.

A final experiment highlights that the proposed method allows $P(i, j)$ to be determined independently for each pixel. One benefit of this is that $P(i, j)$ can thus be completely controlled or derived from an arbitrary distribution map, $D$. So far, the variable precision encoding has followed a normal distribution which has allowed the dense data regions to be more precisely encoded (toward $n_{max}$) and sparse data regions to be less precisely encoded (toward $n_{min}$). While this encoding heuristic may be desired in some applications, it may not be suitable for them all.

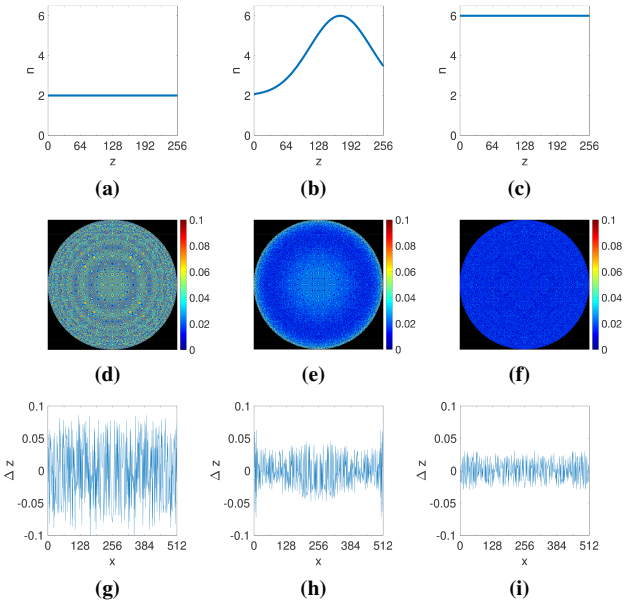



**Figure 4.** Encoding an ideal hemisphere with proposed method, following a normal distribution between $[n_{min}, n_{max}] = [2,6]$, compared to uniformly distributed methods when $n = 2$ and $n = 6$. The first column shows a uniform encoding when $n = 2$, the second column shows the proposed method, and the third column shows $n = 6$. (a)-(c) distribution of the parameter $n$ through the depth range; (d)-(f) absolute error maps versus the original hemisphere; and (g)-(i) reconstruction error values for the 256th row.

**Table 1.** Performance of the proposed method compared to uniformly distributed methods when PNG was used to encode an ideal hemisphere, with a radius of 256 mm, into a $512 \times 512$ image. (a) RMS error and file size; (b) reduction in error per additional kilobyte of file size necessary to store the encoded image, as compared to the uniform case where $n = 2$.

**(a)**

| | **n = 2** | **n = [2,6]** | **n = 6** |
|---|---|---|---|
| **RMS Error** | 0.0467 mm | 0.0218 mm | 0.0157 mm |
| **File Size** | 166.6 KB | 219.9 KB | 255.5 KB |

**(b)**

| | **n = 2** | **n = [2,6]** | **n = 6** |
|---|---|---|---|
| **Reduction in Error / KB** | – | 0.4680 $\frac{\mu m}{KB}$ | 0.3109 $\frac{\mu m}{KB}$ |

For example, consider 3D geometry that has a significant amount of data points at a largely uniform location in the depth range, such that the distribution's mean ($\mu$) also falls in this region. In a normal distribution was followed, points in this region would be more precisely encoded as they would have larger values within the normalized distribution map, $\overline{D}$. Regions of variation away from the densely populated depth location would therefore be encoded at a lower precision. Instead, if the regions of variations were deemed regions of interest, it may be desirable to better encode these areas. One simple solution could be to use an inverted normal distribution where $D_i = 1.0 - \overline{D}$. Figure 5 details an experiment that highlights such a scenario.

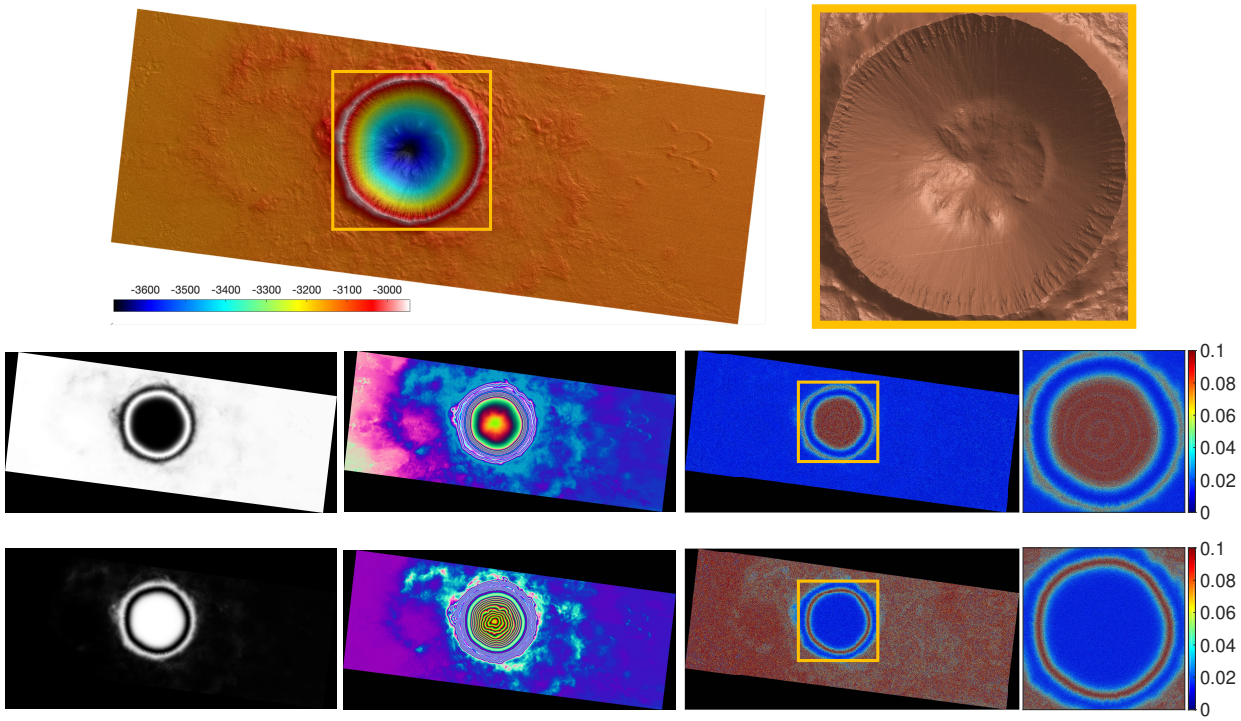



**Figure 5.** Applying the proposed variable precision encoding method to a digital terrain model (DTM) captured by HiRISE: Mars Reconnaissance Orbiter's High Resolution Imaging Science Experiment (NASA/JPL/University of Arizona) [9]. (Row 1) 3D rendering of the DTM "Crater" [10]. The data has a resolution of $14,607 \times 7037$ with a depth range of approximately 734 meters. The right hand side shows a shaded rendering of the $3,300 \times 3,300$ region of interest around the crater. (Row 2) Encoding the DTM with the proposed method between $[n_{min}, n_{max}] = [2, 18]$ following a normal distribution. (Row 3) Encoding the DTM with the proposed method between $[n_{min}, n_{max}] = [2, 18]$ following an inverted normal distribution. In Row 2 and Row 3, the first column shows the distribution map of each encoding; the second column is the output image of each encoding; and the third column is the absolute error map, in meters, for each encoding, with the right hand side showing a magnified view of the error around the crater region of interest.

In Fig. 5 the data encoded is a 411.2 MB digital terrain model (DTM) that was captured by the Mars Reconnaissance Orbiter's High Resolution Science Experiment (HiRISE) [9]. Specifically the $14,607 \times 7,037$ DTM "Crater" [10], with a depth range of approximately 734 meters, was encoded into a 2D image. The first row of Fig. 5 shows a 3D rendering of the full-resolution DTM. The right hand side shows a shaded rendering of the $3,300 \times 3,300$ region of interest around the crater. The second row of Fig. 5 shows the results when the proposed method was used to encode the DTM between $[n_{min}, n_{max}] = [2, 18]$ following a normal distribution of its data. The third row shows results with the same encoding parameters except the distribution was inverted, thus encoding the regions of variance away from the mean with a higher precision. In the second and third rows, the first column shows the distribution map for each encoding. The second column shows the encoded image produced by each encoding. The third column show the absolute error map (in meters) for each encoding, where the right hand side shows a magnified error map for the region of interest around the crater.

As row two of Fig. 5 shows, encoding along a normal distribution results in a low error rate for most of the points. The entire DTM can be reconstructed from a 46.9 MB PNG image (8.77:1 compression ratio) with an RMS error of 0.033 m. However, as the encoding's normal distribution is heavily influenced by the significant number of points near the ground plane, the region of interest around the crater is encoded with less precision. The reconstruction RMS error in this $3,300 \times 3,300$ region thus increases to 0.076 m. If the encoding followed an inverted normal distribution, for example, this region of interest can be more precisely encoded as shown in row three of Fig. 5. For the entire DTM, the RMS error will increase to 0.106 m, however, in the region of interest the RMS error decreases to 0.049 m. Further, since a fewer number of points are being encoded at a higher precision, the encoded image only requires 27 MB to store (15.25:1 compression ratio). This experiment highlights that the proposed method enables the derivation of $P(i, j)$ to be customized, and potentially optimized, for the specific data being encoded.

## Summary

This paper presented a novel method that allows for a variable encoding frequency to be used to encode 3D range geometry within a 2D image. The proposed method allows the encoding frequency to be determined on a point-by-point basis, thus allowing the encoding precision to be independently determined for each coordinate. To more precisely encode regions in the depth range that are dense, while less precisely encoding regions that are sparse, the encoding may follow a normal distribution. In contrast, to more precisely encode regions of interest that may have a higher variance, an inverted normal distribution can be used. In general, the nature of the proposed method is such that the encoding precision of each point can be freely controlled or derived by following an arbitrary distribution, thus enabling this method to be employed within a wide range of applications.

## References


[1] Nikolaus Karpinsky and Song Zhang. Composite phase-shifting algorithm for three-dimensional shape compression. *Optical Engineering*, 49(6):1 – 6, 2010.

[2] Song Zhang. Three-dimensional range data compression using computer graphics rendering pipeline. *Appl. Opt.*, 51(18):4058–4064, 2012.

[3] Zhiling Hou, Xianyu Su, and Qican Zhang. Virtual structured-light coding for three-dimensional shape data compression. *Optics and Lasers in Engineering*, 50(6):844 – 849, 2012.

[4] Pan Ou and Song Zhang. Natural method for three-dimensional range data compression. *Appl. Opt.*, 52(9):1857–1863, 2013.

[5] Tyler Bell and Song Zhang. Multiwavelength depth encoding method for 3d range geometry compression. *Appl. Opt.*, 54(36):10684–10691, 2015.

[6] Yajun Wang, Lianxin Zhang, Sheng Yang, and Fang Ji. Two-channel high-accuracy holoimage technique for three-dimensional data compression. *Optics and Lasers in Engineering*, 85:48 – 52, 2016.

[7] Tyler Bell, Bogdan Vlahov, Jan P. Allebach, and Song Zhang. Three-dimensional range geometry compression via phase encoding. *Appl. Opt.*, 56(33):9285–9292, 2017.

[8] Matthew G. Finley and Tyler Bell. Two-channel depth encoding for 3d range geometry compression. *Appl. Opt.*, 58(25), 2019. (Accepted, In Production).

[9] Alfred S. McEwen, Eric M. Eliason, James W. Bergstrom, Nathan T. Bridges, Candice J. Hansen, W. Alan Delamere, John A. Grant, Virginia C. Gulick, Kenneth E. Herkenhoff, Laszlo Keszthelyi, Randolph L. Kirk, Michael T. Mellon, Steven W. Squyres, Nicolas Thomas, and Catherine M. Weitz. Mars reconnaissance orbiter's high resolution imaging science experiment (hirise). *Journal of Geophysical Research: Planets*, 112(E5), 2007.

[10] NASA/JPL/University of Arizona. Digital Terrain Map: Crater. `https://www.uahirise.org/dtm/dtm.php?ID=ESP_027065_2405`.


## Author Biography

*Matthew G. Finley is a graduate student pursuing an M.S. in Electrical and Computer Engineering at the University of Iowa. Matthew received his B.S. from the University of Iowa in 2018. His current research interests include system optimization and parameter selection for optimal 3D image encoding.*

*Prof. Tyler Bell is an Assistant Professor of Electrical and Computer Engineering at the University of Iowa. He leads the Holo Reality Lab and is a faculty member of the Public Digital Arts (PDA) cluster. Tyler received his Ph.D. from Purdue University in 2018. His current research interests include high-quality 3D video communications; high-speed, high-resolution 3D imaging; virtual reality, augmented reality; human computer interaction; and multimedia on mobile devices.*

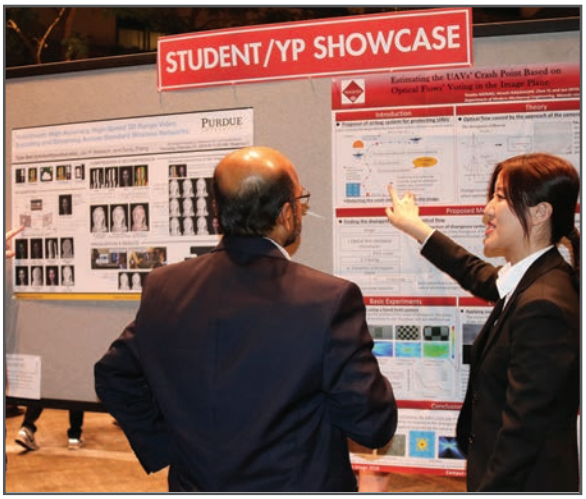
# JOIN US AT THE NEXT EI!

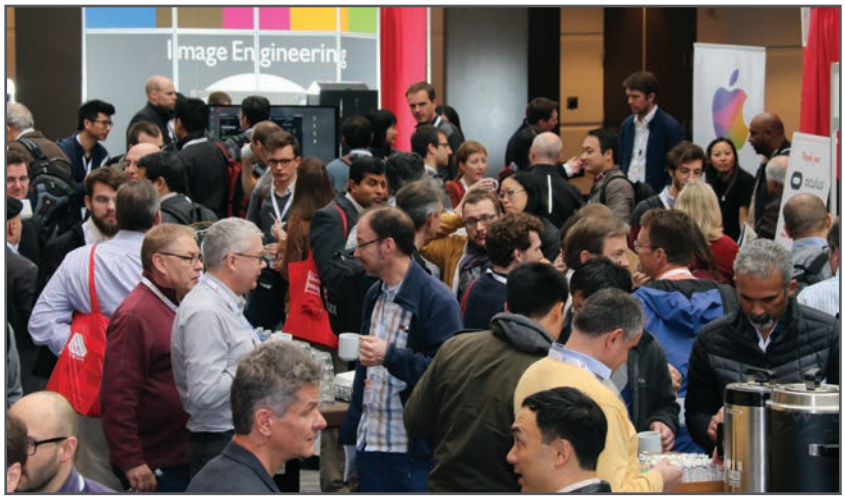
IS&T International Symposium on

# 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