# Large-Scale 3D Point Cloud Compression Using Adaptive Radial Distance Prediction in Hybrid Coordinate Domains

Jae-Kyun Ahn, Kyu-Yul Lee, Jae-Young Sim, *Member, IEEE*, and Chang-Su Kim, *Senior Member, IEEE*

***Abstract*—An adaptive range image coding algorithm for the geometry compression of large-scale 3D point clouds (LS3DPCs) is proposed in this work. A terrestrial laser scanner generates an LS3DPC by measuring the radial distances of objects in a real world scene, which can be mapped into a range image. In general, the range image exhibits different characteristics from an ordinary luminance or color image, and thus the conventional image coding techniques are not suitable for the range image coding. We propose a hybrid range image coding algorithm, which predicts the radial distance of each pixel using previously encoded neighbors adaptively in one of three coordinate domains: range image domain, height image domain, and 3D domain. We first partition an input range image into blocks of various sizes. For each block, we apply multiple prediction modes in the three domains and compute their rate-distortion costs. Then, we perform the prediction of all pixels using the optimal mode and encode the resulting prediction residuals. Experimental results show that the proposed algorithm provides significantly better compression performance on various range images than the conventional image or video coding techniques.**

***Index Terms*—Large-scale 3D point clouds (LS3DPC), point cloud compression, radial distance prediction, range image compression, terrestrial laser scanner.**

## I. Introduction

THREE-DIMENSIONAL (3D) geometry data can offer realistic depth feeling when combined with two-dimensional images or videos. With the advances of 3D data acquisition devices, real world objects and scenes can be captured as 3D models in real-time, which facilitates realistic tele-presence applications [1], [2]. A point cloud, which consists of points distributed in a 3D space, can represent a real world scene faithfully, and thus can be used for multi-user interaction of 3D geometry, due to their inherent nature of structural simplicity. In particular, a point cloud with a huge number of points can render detailed 3D environments accurately, enabling more immersive visual experiences in many applications, such as virtual reality and augmented reality [3], [4].

A highly detailed point cloud can be captured with a small-scale laser scanner [5], which samples points on the surface of a single object. On the other hand, a large-scale terrestrial laser scanner based on the light detection and ranging (LIDAR) can generate a point cloud for a real-world scene including many objects, facilitating many applications of geometry signal processing. Such point clouds are often combined with images and videos together to be used in interactive and immersive visual applications, for example, 3D navigation in virtual environment [4], realistic 3D city modeling [6], and interactive mobile applications [7]. Also, terrestrial scanners can be attached to cars or airplanes to capture accurate digital elevation models for large-scale urban environments [8]–[10]. Surface elevation variations can be monitored by terrestrial scanners as well [11]. In addition, rock or soil surfaces can be represented with millions of points for the analysis of the surface characteristics and shapes [12], as well as for the classification of complex natural scenes [13]. Furthermore, terrestrial scanners are also applicable to biophysical studies, such as the estimation of leaf areas and vegetation structures [14].

A point cloud is often composed of a huge number of points to capture highly detailed geometric information, which makes point cloud compression algorithms essential for efficient storage and transmission. Over the past decade, many algorithms for point cloud compression have been proposed. Gumhold *et al.* [15] predictively encoded the position of each point from the previously coded neighbors in a spanning tree structure. Waschbüsch *et al.* [16] contracted a pair of points to their center position recursively to build up a multiresolution hierarchy. Then, they predicted point positions in a local coordinate system using a lifting scheme. Kalaiah and Varshney [17] partitioned a point cloud hierarchically, and applied the principal component analysis to compress point positions in each partition. Ochotta and Saupe [18] decomposed a point cloud into several surface patches and resampled points on each patch. They encoded the resampled height field using a shape-adaptive wavelet transform. Fleishman *et al.* [19] encoded a point cloud at several levels of detail based on the moving least squares method. Moreover, several techniques have employed octree structures to encode unstructured point

Manuscript received April 16, 2014; revised August 20, 2014; accepted November 06, 2014. Date of publication November 13, 2014; date of current version March 18, 2015. This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2013R1A1A2011920), and in part by the Global Frontier R&D Program on Human-centered Interaction for Coexistence funded by the NRF grant funded by the Korean Government (MSIP) (2011-0031648). The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Gene Cheung.

J.-K. Ahn and C.-S. Kim are with the School of Electrical Engineering, Korea University, Seoul 136-713, Korea (e-mail: demian@korea.ac.kr; changsukim@korea.ac.kr).

K.-Y. Lee and J.-Y. Sim are with the School of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology, Ulsan 689-798, Korea (e-mail: ever1135@unist.ac.kr; jysim@unist.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2014.2370752

clouds. Botsch *et al.* [20] subdivided a non-empty cell of an octree into eight child cells recursively and encoded the nonuniform tree structure. Schnabel and Klein [21] predicted the numbers and indices of non-empty child cells by intersecting locally approximated surfaces. Huang *et al.* [22] used normal vectors to estimate the indices of non-empty cells. Also, in [23], [24], locally approximated plane parameters were encoded instead of point coordinates based on octree structures.

On the other hand, relatively structured point clouds can be obtained as depth maps by analyzing multi-view videos or using depth sensors, which are often employed in video plus depth applications. They can be efficiently compressed by depth map coding techniques. Merkle *et al.* [25] treated a sequence of depth maps as a monochromatic video sequence and encoded it using a conventional video codec. Maitre and Do [26] applied a shape-adaptive wavelet transform for depth map coding. Merkle *et al.* [27] partitioned a depth map into multiple blocks and estimated depth values in each block using a modeling function based on the rate-distortion optimization. Ruiz-Hidalgo *et al.* [28] segmented a depth map into homogeneous regions of different shapes, and encoded depth values in each segment using a shape adaptive transform coder. Gao *et al.* [29] projected the geometry of a 3D object onto a set of optimally selected image planes and encoded the multiple depth images using a conventional multi-view video codec.

The aforementioned algorithms focus on the compression of point clouds, which are captured by small-scale laser scanners [15]–[24] or represented as depth maps [25]–[29]. In comparison, relatively little effort has been made to compress point clouds captured by large-scale terrestrial laser scanners, despite their versatility in immersive 3D applications. From now on, we call point clouds obtained by terrestrial laser scanners as large-scale 3D point clouds (LS3DPCs). LS3DPCs have different characteristics from small-scale point clouds. For example, a typical small-scale cloud is obtained by sampling the surface of a small 3D object, which can be modeled by smooth surfaces. On the contrary, an LS3DPC covers a wide field of view in a real-world scene and includes many objects in general. Hence smooth surfaces may not approximate the entire set of points faithfully. Also, a typical small-scale cloud contains no connectivity information, unlike meshes, which makes it difficult to identify the neighboring points for predicting and encoding each point. However, point positions in an LS3DPC can be described by either 3D coordinates or a scalar field on uniform grids, and this hybrid representation can be exploited to improve the compression performance of LS3DPCs.

In this paper, we propose a novel geometry compression algorithm for LS3DPCs. Since an LS3DPC can be converted into a range image, in which the horizontal and vertical coordinates of pixels correspond to the azimuthal and polar angles of 3D points, we encode the radial distances in the range image instead of the 3D coordinates of the point cloud. We also introduce an alternative representation of range image, called height image, which stores the height values of 3D points that are uniquely generated from a given range image. Then, we can encode the radial distances in the height image domain as well. In practice, we first divide an input range image adaptively into blocks of various sizes. For each block, we encode the pixels in a raster scan order, which are predicted from the previously encoded neighbors. More specifically, we design twelve prediction modes, which are performed in different coordinate domains of the range image, the height image, and the 3D space. Then, for each block, we select the optimal mode that demands the minimum rate-distortion cost. We predict all pixels in the block using the selected mode and transmit the prediction residuals. Experimental results demonstrate that the proposed algorithm achieves significantly higher coding gains on diverse LS3DPCs models than the conventional image and video coding techniques.

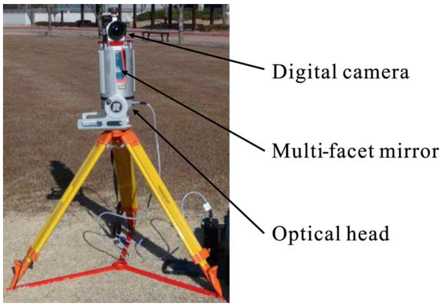


Fig. 1. A terrestrial laser scanner, RIEGL VZ-400 [30].

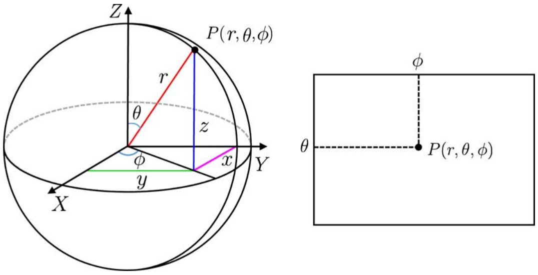


Fig. 2. Two coordinate systems for 3D point geometry. Each point in the spherical coordinate system corresponds to each pixel in the range image.

The rest of this paper is organized as follows. Section II describes how to acquire LS3DPCs and discusses their characteristics. Sections III and IV propose the radial distance prediction modes and the range image compression algorithm, respectively. Section V presents and discusses the experimental results. Finally, Section VI concludes the paper.

## II. Large-Scale 3D Point Clouds

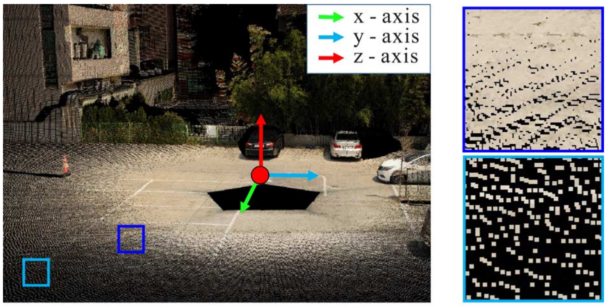
### A. Data Acquisition

We capture LS3DPCs for indoor and outdoor scenes using a terrestrial laser scanner, RIEGL VZ-400 [30], which is composed of an optical scanning module and a photometric camera, as shown in Fig. 1. The scanner represents the 3D geometry of points using the spherical coordinate system, by measuring the radial distances from the scanner center to objects in uniformly distributed angular directions. Fig. 2 shows a point $\mathbf{P}$ in a 3D space represented by spherical coordinates $(r, \theta, \phi)$, where $r$, $\theta$, and $\phi$ denote the radial distance, polar angle, and azimuthal angle, respectively. The corresponding Cartesian coordinates $(x, y, z)$ is computed from the spherical coordinates,

$$x = r \sin\theta \cos\phi, \tag{1}$$

$$y = r \sin\theta \sin\phi, \tag{2}$$

$$z = r \cos\theta. \tag{3}$$



(a) (b)

Fig. 3. Characteristics of LS3DPCs: (a) a captured large-scale outdoor scene and (b) the comparison of different sampling densities according to radial distances.

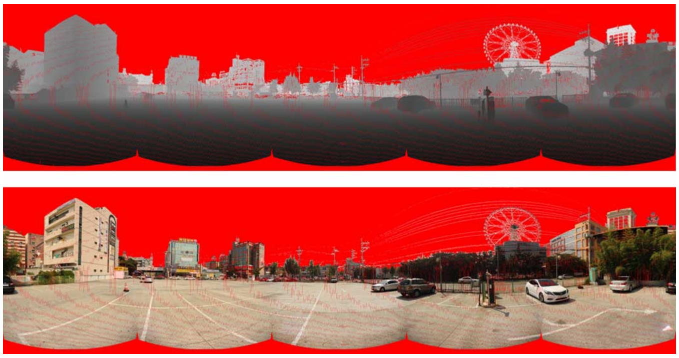


Fig. 4. The range image in the log-scale (top) and the color image (bottom) of the "Downtown" model.

The radial distance $r$ is measured from the scanner to a target object, by transmitting a pulsed laser beam toward a uniformly sampled grid position on the unit sphere and then recording the time of flight of the reflected beam:

$$r = \frac{\Delta T \times \mu}{2} \tag{4}$$

where $\Delta T$ is the time difference between the emitted and received laser beams, and $\mu$ is the light speed.

The scanner covers 100° vertical and 360° horizontal fields of view. Specifically, the vertical line scan is performed using a fast rotating multi-facet mirror, which deflects laser beams toward uniformly distributed radial directions of polar angles from 30° to 130° along the vertical direction. The angle between two consecutive laser beams is 0.08°, yielding 1,250 points per each vertical scan line. The horizontal scan is performed by rotating the optical head over the 360° azimuthal range. The angular resolution is set to 0.08°, which results in 4,500 vertical scan lines. Therefore, for each scan, we have 5,625,000 sampled points in total. When operating in the high speed mode, measurement distances range from 1.5 meters to 350 meters and the scanning speed is 122,000 points per second, which spends about 47 seconds for a single scan.

As a photometric sensing module, a Nikon D700 digital camera is mounted on top of the optical scanning module, which captures several color images for a scanned scene during the horizontal rotation. Each image has a spatial resolution of 4,256 × 2,832. We assign a color value to each point by calibrating the geometry data and the color images using the RIEGL RiSCAN PRO software [31], as a post-processing step. Consequently, we obtain an LS3DPC, in which a single point has 6-tupled attributes: radial distance, polar angle, azimuthal angle, red, green, and blue color values. However, in this work, we focus on the range image compression and do not consider the compression of the color values. Note that those red, green, and blue values can be represented by a single color image, which can be encoded by any conventional image or video coding techniques.

### B. *Characteristics of Large-Scale 3D Point Clouds*

LS3DPCs have several characteristics. First, we can capture large-scale 3D scenes with long distance ranges, *e.g.,* hundreds of meters, as well as wide fields of view, *e.g.,* a horizontal scan angle of 360°. Thus, an LS3DPC model usually includes various 3D objects with different sizes, such as buildings, trees, and cars, as shown in Fig. 3(a).

Second, points are sampled along uniformly distributed radial directions, and thus the sampling density depends on the distances of the points from the scanner. As compared in Fig. 3(b), the points on the objects close to the scanner are densely sampled, while those far from the scanner are sparsely sampled, since the adjacent laser beams spread out as they travel farther from the scanner.

Third, an LS3DPC model can be represented by a rectangular range image. As explained in Section II-A, the laser beam directions are uniformly sampled on the unit sphere in terms of both polar and azimuthal angles. Therefore, each point is associated with integer multiples of a fixed angular step size, which can serve as the pixel coordinates in a rectangular image. As shown in Fig. 2, we map those integers associated with the azimuthal and polar angles to the horizontal and vertical pixel coordinates of a range image, respectively, where each pixel value describes the radial distance of a 3D point in a floating-point number. Fig. 4 shows the range image, which is reconstructed from the LS3DPC model in Fig. 3, where red pixels are not assigned valid radial distances by the scanner. It is worth pointing out that we can exploit the regular structure of a range image to efficiently compress the geometry data of an LS3DPC, by encoding the single radial distance instead of the three coordinates of each point.

Fourth, we can represent the color information of an LS3DPC model using a single color image as well. Fig. 4 shows a pair of range and color images. Note that some points, whose polar angles are close to 30° or 130°, are not assigned color values due to the mismatch of fields of views between the scanner and the camera. Especially, due to the limited horizontal field of view of the camera, a polygonal hole occurs when a colored point cloud is acquired. We take five color images during the horizontal scan, and thus a pentagon hole is observed in Fig. 3(a), which yields rippling boundaries of valid pixels in both range and color images in Fig. 4.

## III. Radial Distance Prediction in Hybrid Coordinate Domains

We propose a geometry data compression algorithm for LS3DPCs, which encodes radial distances in range images. Since range images of LS3DPCs have different characteristics from ordinary color images, the conventional algorithms based on block-wise transform and prediction may not achieve high coding gains. Specifically, adjacent pixels in a range image often exhibit significantly different radial distances. Moreover, as shown in red in Fig. 4, a range image may have empty pixels where radial distances are not defined due to the reflection characteristics of target objects and the limited vertical field of view, for example, in the windows of buildings and cars, around the scanning center, and the sky regions. We call these empty pixels as holes. In addition, a range image stores pixel values in floating-point numbers, which cannot be directly encoded by the conventional image coders.

Considering these characteristics, we propose a hybrid prediction algorithm for radial distances in three domains: range image, height image, and 3D space. In practice, we divide an input range image into blocks of various sizes from $8 \times 8$ to $32 \times 32$. For each block, we perform the pixel-wise prediction using twelve prediction modes, which are carried out in one of the three domains. Then, we select the optimal mode and predict the pixels in the block using the selected mode. Let us describe the proposed multiple prediction modes for radial distances subsequently.

### *A. Prediction Modes in the Range Image Domain*

The prediction of pixel values in a range image can be regarded as the extrapolation of a radial distance field. Given a current pixel $\mathbf{p}$, we design a predictor $\hat{r}(\mathbf{p})$ that estimates the original radial distance $r(\mathbf{p})$ using previously encoded distances.

*1) Nearest Neighbor Modes:* We first design a simple predictor $\hat{r}(\mathbf{p})$ by employing the 8-connected neighbors to $\mathbf{p}$. Since we encode pixels in the raster scan order, the four pixels among the 8-connected neighbors are previously encoded and thus available for the prediction. They are the left, upper, upper-left, and upper-right pixels to $\mathbf{p}$, denoted by $\mathbf{p}_{\mathrm{l}}$, $\mathbf{p}_{\mathrm{u}}$, $\mathbf{p}_{\mathrm{ul}}$, and $\mathbf{p}_{\mathrm{ur}}$, respectively. Thus, we define the four nearest neighbor (NN) prediction modes accordingly – horizontal, vertical, diagonal down-right, and diagonal down-left modes – by setting the predictor as

$$\hat{r}(\mathbf{p}) = \begin{cases} \tilde{r}(\mathbf{p}_{\mathrm{l}}), & \text{horizontal,} \\ \tilde{r}(\mathbf{p}_{\mathrm{u}}), & \text{vertical,} \\ \tilde{r}(\mathbf{p}_{\mathrm{ul}}), & \text{diagonal down-right,} \\ \tilde{r}(\mathbf{p}_{\mathrm{ur}}), & \text{diagonal down-left,} \end{cases} \tag{5}$$

where $\tilde{r}(\cdot)$ denotes the reconstructed radial distance of $r(\cdot)$. Fig. 5 illustrates the four NN prediction modes. Blue and white pixels represent encoded and unencoded pixels, respectively. A red pixel is the current pixel to be encoded, and a green pixel denotes the NN used for the prediction. Note that some pixels may have holes as their NNs. In such a case, we employ the next nearest pixel among the previously encoded neighbors as a predictor, regardless of the prediction directions. If there are several pixels with the same distance, we select a pixel by testing the distances along the horizontal direction and the vertical direction in order.

*2) Radial Basis Function Mode:* Radial basis functions (RBFs) are widely used to interpolate or extrapolate a function value from scattered data points in applications, such as implicit surface approximation and image inpainting [32]. In this work, we consider a radial distance as a function value, and approximate $r(\mathbf{p})$ at pixel $\mathbf{p}$ as a weighted sum of RBFs centered at $N$ previously encoded neighbors $\mathbf{p}_i$'s within the $3 \times 3$ window around $\mathbf{p}$. Specifically, we design the RBF predictor as

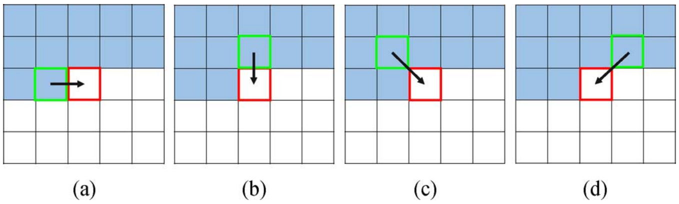


Fig. 5. Four nearest neighbor (NN) prediction modes: (a) horizontal, (b) vertical, (c) diagonal down-right, and (d) diagonal down-left modes. A red pixel is predicted from a green pixel. Blue regions and white regions correspond to encoded and unencoded pixels, respectively.

$$\hat{r}(\mathbf{p}) = \sum_{i=1}^{N} \lambda_i \Phi(\|\mathbf{p} - \mathbf{p}_i\|) + K \tag{6}$$

where $\Phi(\cdot)$ is an RBF, $\lambda_i$ is a weight, and $K$ is a constant. We tested various functions, including thin-plate spline, Gaussian, and multi-quadric functions, as $\Phi(\cdot)$ and adopted the power function, which provided good performance. The power function is given by

$$\Phi(x) = x^{\beta}. \tag{7}$$

We set $\beta$ to 1.3 empirically. In order to find the optimal weights of $\lambda_i$'s and the optimal constant of $K$ in (6), we solve the following linear equations [32] with an additional constraint $\sum_{i=1}^{N} \lambda_i = 0$.

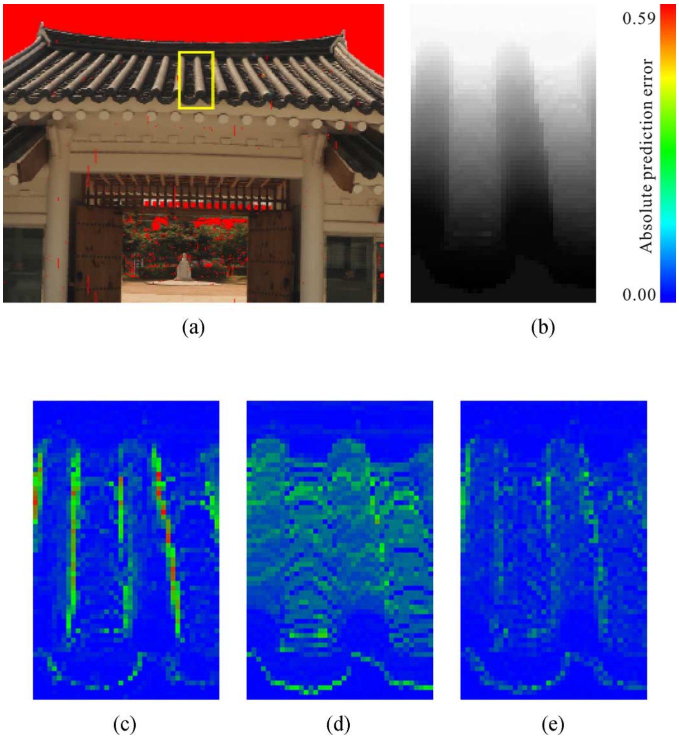
$$\begin{bmatrix} \tilde{r}(\mathbf{p}_1) \\ \tilde{r}(\mathbf{p}_2) \\ \vdots \\ \tilde{r}(\mathbf{p}_N) \\ 0 \end{bmatrix} = \begin{bmatrix} \Phi(\|\mathbf{p}_1 - \mathbf{p}_1\|) & \dots & \Phi(\|\mathbf{p}_1 - \mathbf{p}_N\|) & 1 \\ \Phi(\|\mathbf{p}_2 - \mathbf{p}_1\|) & \dots & \Phi(\|\mathbf{p}_2 - \mathbf{p}_N\|) & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \Phi(\|\mathbf{p}_N - \mathbf{p}_1\|) & \dots & \Phi(\|\mathbf{p}_N - \mathbf{p}_N\|) & 1 \\ 1 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \\ K \end{bmatrix}. \tag{8}$$

Note that we use the reconstructed radial distances $\tilde{r}(\mathbf{p}_i)$'s of the previously encoded neighbors to form the linear equations, so that the decoder can determine the same optimal parameters as well.

Fig. 6 shows the prediction errors of the RBF mode, compared with those of the horizontal and vertical NN modes. The yellow region in Fig. 6(a) captures highly curved surfaces, whose radial distances are shown in Fig. 6(b). While the NN predictors are inaccurate in this region as shown in Fig. 6(c) and (d), the RBF predictor provides relatively small prediction errors in Fig. 6(e).

### *B. Prediction Modes in the Height Image Domain*

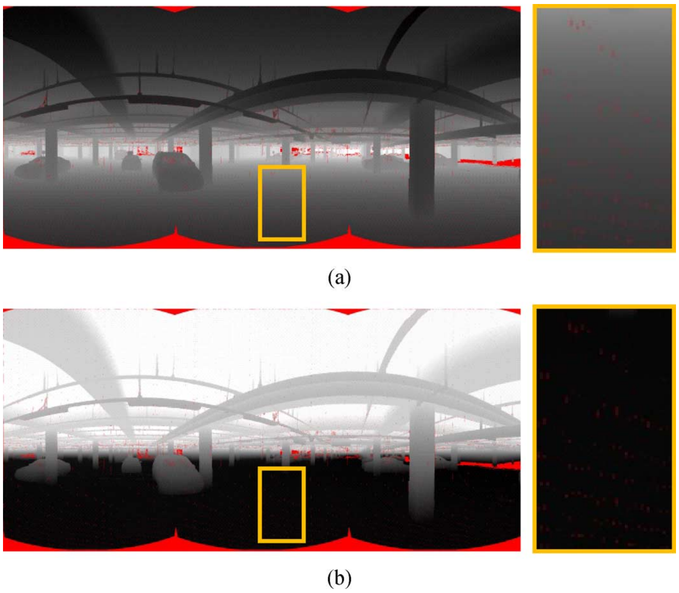
In certain regions of range images, even though neighboring pixels have significantly different radial distances, they may be highly correlated in other coordinate systems. For example, a planar region such as a ceiling or a ground plane is often



(a) (b)

(c) (d) (e)

Fig. 6. Radial distance prediction using the RBF mode. (a) The "Memorial Hall" model. (b) The radial distances corresponding to the yellow rectangle in (a). The resulting prediction errors of the (c) horizontal, (d) vertical, and (e) RBF modes, respectively.



(a)

(b)

Fig. 7. Comparison of radial distances and heights in the "Parking Lot" model: (a) the range image and (b) the height image. Highlighted regions within the yellow rectangles are magnified for easier comparison.

perpendicular to the $z$-axis in the Cartesian coordinate system, since a terrestrial scanner is usually held upright when capturing scenes. Therefore, pixels in such a region have almost the same height values in $z$-coordinates. In order to exploit such correlation and improve the compression performance, we propose a novel image domain, called height image. It has the horizontal and vertical coordinates of pixels as the azimuthal and polar angles of 3D points as the range image does, but it stores a height value of a 3D point at each pixel. The height image can be uniquely obtained from a given range image using the relation in (3) between the spherical coordinate system and the Cartesian coordinate system. Fig. 7 shows the range image and the corresponding height image for the "Parking Lot" model, which are separately normalized for visualization. We observe that the pixel values within the highlighted region on the ground plane are almost constant in the height image, while those in the range image vary according to the distances from the scanner.

We exploit this property to improve the prediction performance for radial distances. We first obtain height values $z(\mathbf{p})$'s from radial distances $r(\mathbf{p})$'s using the relation in (3). Then, instead of predicting $r(\mathbf{p})$ directly, we predict $z(\mathbf{p})$ in the height domain. We apply the four NN prediction modes and the RBF prediction mode in Section III-A to height values. In the NN modes, the height predictor $\hat{z}(\mathbf{p})$ is given by

$$\hat{z}(\mathbf{p}) = \begin{cases} \tilde{z}(\mathbf{p}_\mathrm{l}), & \text{horizontal,} \\ \tilde{z}(\mathbf{p}_\mathrm{u}), & \text{vertical,} \\ \tilde{z}(\mathbf{p}_\mathrm{ul}), & \text{diagonal down-right,} \\ \tilde{z}(\mathbf{p}_\mathrm{ur}), & \text{diagonal down-left,} \end{cases} \tag{9}$$

where $\tilde{z}(\cdot)$ is the reconstructed value of $z(\cdot)$. In the RBF mode, we have

$$\hat{z}(\mathbf{p}) = \sum_{i=1}^{N} \lambda_i \Phi(\|\mathbf{p} - \mathbf{p}_i\|) + K \tag{10}$$

where the optimal parameters of $\lambda_i$'s and $K$ are also obtained by solving the linear equations in (8) after replacing $\tilde{r}(\mathbf{p}_i)$'s with $\tilde{z}(\mathbf{p}_i)$'s. Consequently, from (3), the radial distance predictors of the five modes in the height domain are given by

$$\hat{r}(\mathbf{p}) = \frac{\hat{z}(\mathbf{p})}{\cos\theta_{\mathbf{p}}}, \tag{11}$$

where $\theta_{\mathbf{p}}$ is the polar angle at $\mathbf{p}$, which varies according to the pixel position. Note that, when $\theta_{\mathbf{p}}$ is close to 90°, a height prediction error $|z(\mathbf{p}) - \hat{z}(\mathbf{p})|$ is amplified into a large radial distance prediction error $|r(\mathbf{p}) - \hat{r}(\mathbf{p})|$.

### C. Pixel-Wise Plane Prediction Mode in the 3D Domain

We also exploit the correlation in 3D point coordinates to improve the coding gains for range images. From a radial distance of each pixel, we can reconstruct the Cartesian coordinates $(x, y, z)$ of the corresponding 3D point via (1), (2), and (3). Then, we can generate an implicit surface in the 3D space, which approximates the distribution of 3D points of previously encoded pixels. We then can associate a ray for a current pixel, starting from the projection center of the scanner, and predict the radial distance of the pixel by measuring the distance to the intersection point of the ray and the implicit surface. However, an LS3DPC model may include various kinds of complicated objects, which are not well approximated by smooth surfaces. Furthermore, the implicit surface modeling requires high computational complexity in general. Also, it is not trivial to find the intersection point between the ray and the implicit surface, and there may be multiple intersections. Therefore, instead of the implicit surface modeling, we approximate the local distribution of 3D points simply by a plane and adopt the plane prediction mode. This mode exploits the spatial correlation of 3D

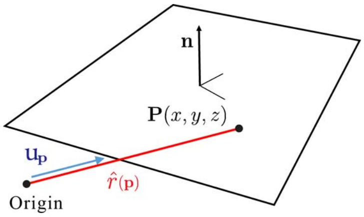


Fig. 8. Plane-based prediction of a radial distance.

TABLE I
RADIAL DISTANCE PREDICTION MODES, WHERE 'H' INDICATES THE PREDICTION IN THE HEIGHT IMAGE DOMAIN.

| Index | Prediction mode | Index | Prediction mode |
|---|---|---|---|
| 0 | Horizontal | 6 | Diagonal down-right |
| 1 | Horizontal (H) | 7 | Diagonal down-right (H) |
| 2 | Vertical | 8 | RBF |
| 3 | Vertical (H) | 9 | RBF (H) |
| 4 | Diagonal down-left | 10 | Pixel-wise plane prediction |
| 5 | Diagonal down-left (H) | 11 | Block-wise plane prediction |

coordinates and is effective especially for artificial structures of planar shapes, *e.g.*, buildings and walls.

A plane in the Cartesian coordinate system can be expressed as

$$ax + by + cz = d, \tag{12}$$

where $(a, b, c)$ represents the normal vector $\mathbf{n}$ and $d$ is the distance from the origin to the plane. Let $\mathbf{P} = (x, y, z)$ denote a 3D point corresponding to pixel $\mathbf{p}$ in a range image. As shown in Fig. 8, we define the unit directional vector $\mathbf{u_p}$ from the origin to $\mathbf{P}$, where

$$\mathbf{P} = r(\mathbf{p})\mathbf{u_p}. \tag{13}$$

If $\mathbf{P}$ lies on the plane, then

$$\mathbf{P} \cdot \mathbf{n} = d. \tag{14}$$

Therefore, we determine the predicted radial distance $\hat{r}(\mathbf{p})$ as the distance from the origin to the approximate plane along the direction $\mathbf{u_p}$, given by

$$\hat{r}(\mathbf{p}) = \frac{d}{\mathbf{u_p} \cdot \mathbf{n}}. \tag{15}$$

We determine the normal vector $\mathbf{n}^*$ and the distance $d^*$ of the optimal plane, which best approximates the 3D positions of previously encoded neighboring pixels, using the weighted least squares method:

$$(\mathbf{n}^*, d^*) = \arg\min_{(\mathbf{n},d)} \sum_{\mathbf{p}_i \in S_\mathbf{p}} w_i \|\tilde{\mathbf{P}}_i \cdot \mathbf{n} - d\|^2 \tag{16}$$

where $S_\mathbf{p}$ is the set of the previously encoded neighbors within the $7 \times 7$ window centered at $\mathbf{p}$, and $\tilde{\mathbf{P}}_i$ is the reconstructed 3D position of $\mathbf{p}_i$. Also, the weight $w_i$ is set as

$$w_i = \frac{\frac{1}{\|\mathbf{p}-\mathbf{p}_i\|}}{\sum\limits_{\mathbf{p}_j \in S_\mathbf{p}} \frac{1}{\|\mathbf{p}-\mathbf{p}_j\|}}. \tag{17}$$

Note that we use the reconstructed 3D positions $\tilde{\mathbf{P}}_i$'s in (16), so that the decoder also can estimate the approximate plane without requiring any side information. Moreover, we selectively take the local plane predictor $\hat{r}(\mathbf{p})$ in (15), only when the normalized fitting error $(1/|S_\mathbf{p}|) \sum_{\mathbf{p}_i \in S_\mathbf{p}} w_i \|\tilde{\mathbf{P}}_i \cdot \mathbf{n} - d\|^2$ is less than a threshold $\epsilon$. We empirically set $\epsilon = 0.005$ in this work.

### D. Block-Wise Plane Prediction Mode in the 3D Domain

Even when neighboring points are sampled from a flat surface, the pixel-wise plane prediction may be inaccurate with a small number of available pixels or at low bit-rates. Hence we also propose a block-wise planar prediction mode, which directly estimates a single plane for a block and predicts the radial distances of all pixels in the block using the estimated plane. We find the optimal plane parameters $\mathbf{n}_B^*$ and $d_B^*$ for block $B$ by minimizing the prediction errors of the radial distances,

$$(\mathbf{n}_B^*, d_B^*) = \arg\min_{(\mathbf{n},d)} \sum_{\mathbf{p}_i \in B} \left| r(\mathbf{p}_i) - \frac{d}{\mathbf{u}_{\mathbf{p}_i} \cdot \mathbf{n}} \right|. \tag{18}$$

Note that we do not find the plane that best fits the distribution of the 3D points corresponding to the pixels in the block, but find the optimal plane that minimizes the prediction errors of the radial distances in order to improve the compression performance. In practice, we employ the random sample consensus (RANSAC) method [33]. The encoder randomly selects three pixels in a given block and then estimates the plane for the three corresponding points. Then, the encoder computes the absolute sum of prediction errors in (18) with respect to this plane. We repeat this procedure 100 times, and select the best plane to provide the minimum sum of prediction errors. Using the optimal plane parameters, we predict the radial distances via (15). Note that, in this mode, we should transmit the plane parameters to the decoder as side information.

## IV. RANGE IMAGE COMPRESSION

We divide a range image into blocks of various sizes from $8 \times 8$ to $32 \times 32$. For each block, we select the best prediction mode and predict all pixels in the block using the selected mode. We encode and transmit the index of the prediction mode and the resulting prediction residuals.

### A. Prediction Mode Selection

For each block, we determine the optimal prediction mode using the Lagrangian optimization technique [34], [35]. Specifically, by employing each prediction mode in Table I, we predict the radial distances $r(\mathbf{p})$'s in the block and quantize the resulting prediction residuals $(r(\mathbf{p}) - \hat{r}(\mathbf{p}))$'s with a given step size

$$\Delta_q = \frac{M}{2^q}, \tag{19}$$

where $M$ is the difference between the maximum and the minimum radial distances of an input LS3DPC model and $q$ is the quantizer resolution. We apply the context-based entropy encoding to the quantized prediction residuals, as well as to the index $k$ of the prediction mode, and measure the required rate $R_k$. We also compute the distortion $D_k$ as the sum of the squared differences (SSD) between the original and reconstructed positions of the 3D points, corresponding to the pixels in the block. Then, we compute the rate-distortion cost $J_k$ for the $k$th prediction mode as

$$J_k = D_k + \lambda R_k, \tag{20}$$

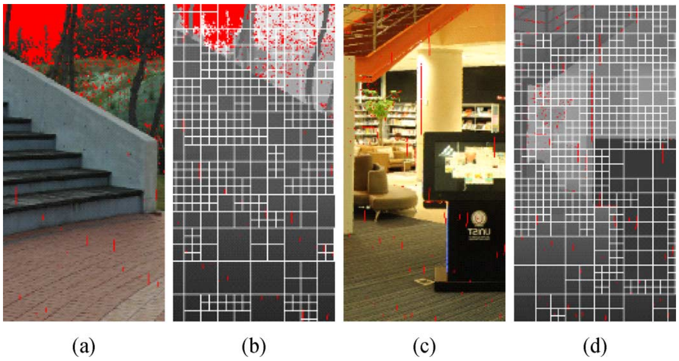


Fig. 9. Adaptive block partitioning of range images: (a) the color image and (b) the range image of the “Amphitheatre” model, and (c) the color image and (d) the range image of the “Library” model, at the quantization step size of $\Delta_{14}$. The biggest and the smallest block sizes are 32 × 32 and 8 × 8, respectively.

where $\lambda$ is a Lagrange multiplier. As in [35], the Lagrange multiplier is set according to the quantization step size as

$$\lambda = \alpha \Delta_q^2, \tag{21}$$

where $\alpha$ is set to 0.1 in this work. By comparing the costs $J_k$'s for all prediction modes, we choose the best mode that yields the minimum cost.

Moreover, we adaptively determine variable block sizes. We first divide an input image into 32 × 32 blocks, and find the optimal prediction mode and the corresponding rate-distortion cost for each block. Then, we subdivide each 32 × 32 block into four 16 × 16 blocks, and find the optimal prediction modes and the rate-distortion costs for the smaller blocks, respectively. In general, as the block size becomes smaller, smaller prediction residuals are generated but more bits are required to encode more indices of the prediction modes and the flag bits for the block partitioning. When the sum of the costs for the four 16 × 16 blocks is lower than the cost of the single 32 × 32 block, we subdivide the 32 × 32 block into the four sub-blocks. We iteratively perform this block partitioning until the size of a sub-block becomes 8 × 8. Fig. 9 shows the partitioning results of the “Amphitheatre” and “Library” models. Whereas homogeneous regions are divided into 32 × 32 or 16 × 16 blocks, complicated regions are decomposed into 8 × 8 blocks.

### B. Context-Based Entropy Coding

We first encode flag bits to specify adaptively partitioned blocks. Then, for each block, we encode the prediction mode index. In the case of the block-wise plane prediction mode, we additionally encode plane parameters as side information. To reduce the overhead for the side information, we find an approximating plane for the pixels in the top row and the left column adjacent to the current block, which is then used to predict the optimal plane for the block. The prediction residuals are then quantized and encoded. Since the plane parameters affect the prediction of all radial distances in the block, we quantize them with a sufficiently fine step size of 0.0001, regardless of the quantization step size for radial distances. We also predictively encode the maximum bit depth to describe the magnitude of the prediction residuals in the block, using those of the upper block and the left block. Then, we encode each bit plane of the quantized prediction residuals including the sign bits in the raster scan order, by employing the bit-plane coder in [36], [37]. All those symbols are encoded by a context-based arithmetic coder [38].

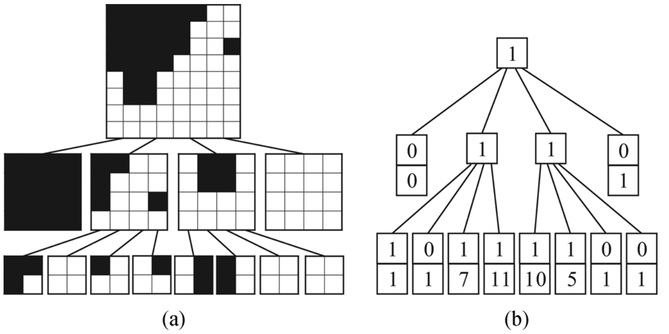


Fig. 10. An example of binary map coding: (a) quad-tree decomposition and (b) encoded symbols.

More specifically, we employ different contexts according to the bit types to be encoded. We use the same *header context* to encode the header information of all blocks, but use a different set of contexts for each block to encode the prediction residuals. For each block, we first use the *sign context* to encode the sign bits of the prediction residuals. We also use the *head context* to encode the magnitude bits of each residual from the most significant bitplane to the bitplane where the first ‘1’ bit occurs. After encoding the first ‘1’ bit, we use the *tail context* to encode the remaining bitplanes. Notice that the head context is associated with high probabilities of ‘0’ bits, since all bits should be ‘0’ except for the first ‘1’ bits. In contrast, ‘0’ and ‘1’ bits are almost equally likely to appear for the tail context.

### C. Binary Map Coding

Note that a range image may include hole pixels, to which valid radial distances are not assigned. Therefore, we define a binary map to indicate whether each pixel in a range image is valid or not. We first define the smallest binary square map to represent the range image. When all binary values within the map are identical, we encode the flag bit ‘0’ and the binary value. Otherwise, we subdivide the map into the four smaller blocks using the quad-tree decomposition [39]. Then, we iteratively perform the encoding and the subdivision until all pixels in a sub-block have the same binary value or the block size becomes 2 × 2. When the pixels in a 2 × 2 block do not have the same binary value, we encode the four binary values directly. We also use the context-based arithmetic coder [38] to encode the binary map. We employ three context models for the flag bits for the quad-tree structure, the binary value of each homogeneous block, and the four bit symbols of each 2 × 2 block, respectively. Fig. 10 shows an example of the binary map decomposition and the encoded symbols.

## V. Experimental Results

We evaluate the performance of the proposed geometry compression algorithm for LS3DPCs using diverse test models: eight outdoor scenes and four indoor scenes. They are captured by an LIDAR-based terrestrial range scanner, RIEGL VZ-400 [30]. Fig. 11 shows rendered views of the colored test models, as well as the corresponding range and color images. Table II summarizes the properties of these test models. The

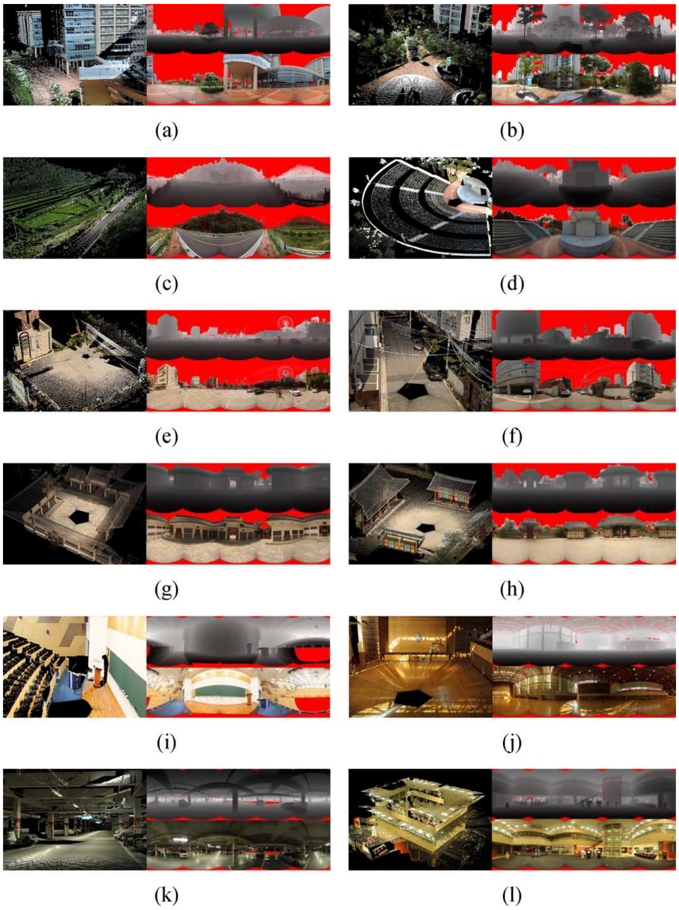


Fig. 11. Twelve test LS3DPC models: (a) "Campus," (b) "Apartments," (c) "Country," (d) "Amphitheatre," (e) "Downtown," (f) "Street," (g) "Memorial Hall," (h) "Korean House," (i) "Classroom," (j) "Gymnasium," (k) "Parking Lot," and (l) "Library." For each model, a 3D rendering (left), the range image (top), and the color image (bottom) are shown.

TABLE II
PROPERTIES OF TEST LS3DPC MODELS.

| Test model | Scene type | Number of points | Maximum range (m) | Range image resolution |
|---|---|---|---|---|
| Campus | Outdoor | 3,194,231 | 347 | 4,000×1,120 |
| Apartments | Outdoor | 3,499,066 | 221 | 4,000×1,120 |
| Country | Outdoor | 2,063,542 | 349 | 4,000×768 |
| Amphitheatre | Outdoor | 2,514,012 | 350 | 4,000×832 |
| Downtown | Outdoor | 2,398,256 | 350 | 4,000×992 |
| Street | Outdoor | 3,171,315 | 294 | 4,000×1,120 |
| Memorial Hall | Outdoor | 2,512,360 | 274 | 4,000×736 |
| Korean House | Outdoor | 2,345,658 | 233 | 4,000×736 |
| Classroom | Indoor | 3,846,112 | 18 | 4,000×1,120 |
| Gymnasium | Indoor | 4,153,589 | 36 | 4,000×1,120 |
| Parking Lot | Indoor | 4,299,233 | 91 | 4,000×1,120 |
| Library | Indoor | 4,266,200 | 99 | 4,000×1,120 |

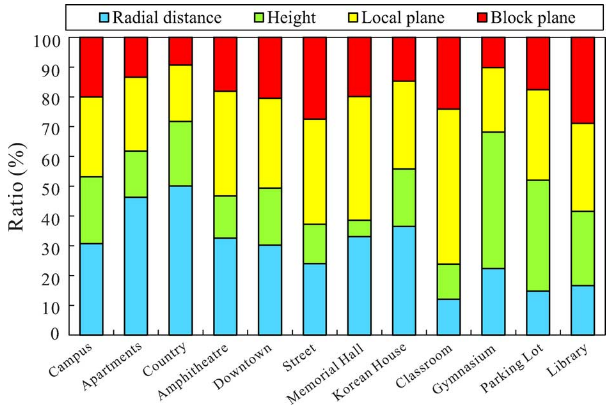


Fig. 12. Selection ratios of the prediction modes, when the quantization step size is $\Delta_{12}$.

TABLE III
SELECTION RATIOS OF THE PREDICTION MODES AT VARIOUS QUANTIZATION STEP SIZES.

| Quantization step size | Range image domain | Height image domain | Pixel-wise plane prediction | Block-wise plane prediction |
|---|---|---|---|---|
| $\Delta_8$ | 28.66 | 25.20 | 33.32 | 12.82 |
| $\Delta_9$ | 28.55 | 23.42 | 33.66 | 14.37 |
| $\Delta_{10}$ | 28.28 | 22.19 | 33.08 | 16.45 |
| $\Delta_{11}$ | 28.17 | 21.68 | 31.15 | 19.00 |
| $\Delta_{12}$ | 29.10 | 20.91 | 31.31 | 18.68 |
| $\Delta_{13}$ | 30.20 | 21.05 | 28.89 | 18.86 |
| $\Delta_{14}$ | 32.16 | 23.54 | 29.14 | 15.16 |
| Average | 29.30 | 22.57 | 31.65 | 16.48 |

TABLE IV
SELECTION RATIOS (%) OF THE THREE BLOCK SIZES AT VARIOUS QUANTIZATION STEP SIZES.

| Quantization step size | 32 × 32 | 16 × 16 | 8 × 8 |
|---|---|---|---|
| $\Delta_8$ | 54.96 | 24.43 | 20.61 |
| $\Delta_9$ | 50.52 | 25.52 | 23.96 |
| $\Delta_{10}$ | 46.73 | 27.04 | 26.23 |
| $\Delta_{11}$ | 42.78 | 29.26 | 27.96 |
| $\Delta_{12}$ | 38.74 | 31.84 | 29.42 |
| $\Delta_{13}$ | 35.75 | 34.20 | 30.05 |
| $\Delta_{14}$ | 32.65 | 36.64 | 30.71 |
| Average | 43.16 | 29.85 | 26.99 |

number of points ranges from two millions to 4.3 millions. The measured radial distances are up to hundreds of meters for outdoor scenes and tens of meters for indoor scenes. We resample initially obtained point clouds at 0.09° intervals over 360° and 100° fields of azimuthal and polar angles, respectively. Therefore, the horizontal resolution of each range image is 4,000. However, we retain the points within valid polar angles only, which results in different vertical resolutions for the five outdoor scenes: "Country," "Amphitheatre," "Downtown," "Memorial Hall," and "Korean House." In addition, we increase the vertical resolutions to the nearest multiples of 32, since the largest block size is 32 × 32 in the proposed algorithm.

## *A. Usage of Prediction Modes and Block Sizes*

We classify the twelve prediction modes for radial distances into the four types: the range image domain (modes 0, 2, 4, 6, 8), the height image domain (modes 1, 3, 5, 7, 9), the pixel-wise plane (mode 10), and the block-wise plane (mode 11). Then, we analyze the number of pixels predicted by each type. Fig. 12 shows the ratios of the four types on the test models, when the prediction residuals are quantized with a step size of $\Delta_{12}$. We observe that relatively larger portions of pixels are predicted in the height image domain or by the pixel-wise plane mode in the cases of the indoor models "Classroom," "Gymnasium," "Parking Lot," and "Library." It is because these indoor

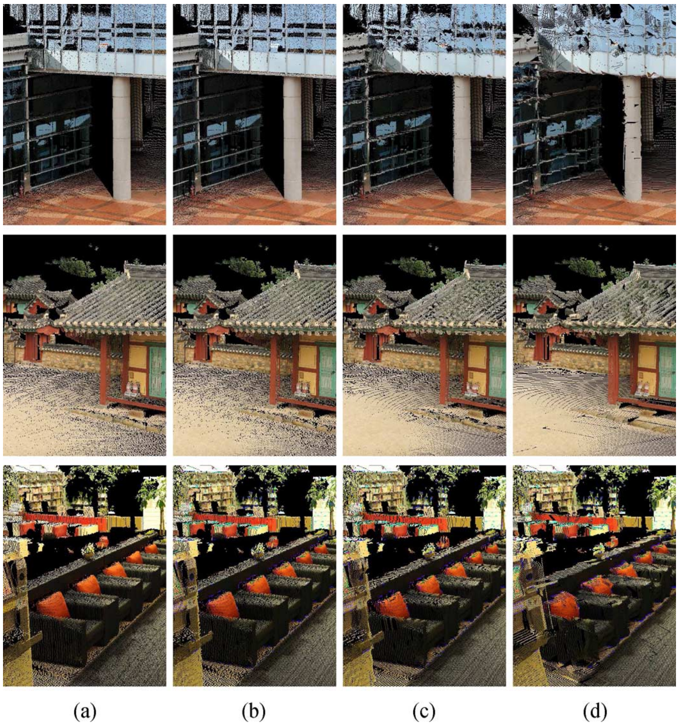


Fig. 13. Reconstructed LS3DPC models of "Campus," "Korean House," and "Library," when different quantization step sizes are used for the prediction residuals: (a) original, (b) reconstruction at $\Delta_{12}$, (c) reconstruction at $\Delta_{10}$, and (d) reconstruction at $\Delta_8$.

scenes contain many planar structures in general, including ceilings and walls. Table III also summarizes the selection ratios of the four types at various quantization step sizes.

As mentioned in Section IV-A, the proposed algorithm partitions an input range image into multiple blocks with various sizes by using the rate-distortion optimization. In Table IV, we analyze the selection ratios of the three block sizes, 32 × 32, 16 × 16, and 8 × 8, at various quantization step sizes. As the quantization step size increases, 32 × 32 blocks are less frequently selected while 16 × 16 and 8 × 8 blocks are more frequently selected. This is because, at low bit-rates, the encoder tends to use bigger blocks to save flag bits for the block partitioning. In contrast, at high bit-rates, image details are more faithfully encoded by employing smaller blocks.

### B. Range Image Compression

We evaluate the proposed algorithm at various bit-rates by changing the quantization step sizes for prediction residuals of radial distances from $\Delta_8$ to $\Delta_{14}$. We measure the distortion as the root mean squared (RMS) error between original 3D points and the reconstructed ones in the Cartesian coordinate system. Note that we regard the original 3D points without any post-processing as ground truth data, which correspond to the range image including hole pixels. Fig. 13 shows original LS3DPC models and their reconstructed ones at three quantization step sizes $\Delta_8$, $\Delta_{10}$, and $\Delta_{12}$. Whereas the reconstructed models are severely distorted at $\Delta_8$ and $\Delta_{10}$, the reconstructions at $\Delta_{12}$ yield similar qualities to the originals. In other words, since an LS3DPC model captures a wide range of radial distances, it can be stored faithfully only at relatively small step sizes (or equivalently relatively high bit-rates). For example, in the outdoor scene of the "Campus" model, the difference between the maximum and the minimum radial distances is 345m. In such a case, the maximum quantization error becomes 0.67 m at $\Delta_8$, which results in severe distortions of the reconstructed geometry, as shown in Fig. 13(d).

We compare the rate-distortion curves of the proposed range image compression algorithm with those of the conventional image and video compression techniques: JPEG2000 [40], JPEG-LS [41], H.264/AVC [42], and High Efficiency Video Coding (HEVC) [43]. While radial distances are recorded in floating-point numbers, the conventional techniques can encode only integer pixel values. Hence, to use the conventional techniques, we pre-quantize pixel values of range images with a given quantization step size $\Delta_q$, and apply the conventional techniques in lossless coding modes. We control the bit-rates for the conventional techniques by varying the pre-quantization step sizes from $\Delta_8$ to $\Delta_{14}$. Moreover, since holes in range images degrade the performance of the conventional techniques, we fill in the holes using the nearest valid pixels as a pre-processing step. For the conventional techniques as well, we transmit the header information and the binary map in the same way as the proposed algorithm. For JPEG2000 and JPEG-LS, we use the reference softwares [44] and [45], respectively. For H.264/AVC and HEVC, we use the latest software releases [46] and [47], by adopting the Fidelity Range Extensions (FRExt) high 4:4:4 profile and the main profile, respectively. For HEVC, we use the color encoder, instead of the depth encoder, and employ pre-quantized range images as the luma components and set the chroma components to 0. In addition, we compare the performance of the lossy intra coding of H.264/AVC and HEVC. In the lossy intra modes, we pre-quantize range images with a fine quantization step size of $\Delta_{13}$, and control the bit-rates by changing the quantization parameter (QP) for DCT coefficients from 1 to 51.

Fig. 14 compares the resultant rate-distortion curves on the test LS3DPC models. We see that the lossy intra modes of H.264/AVC and HEVC achieve better performance than the lossless coding in general, except at high bit-rates. Among the lossless modes, H.264 yields the best performance. However, the proposed algorithm always provides better rate-distortion curves than all conventional techniques on all test models. It is worth pointing out that, in the conventional image and video coding techniques, the pre-quantization of range images degrades the rate-distortion performance in general. On the contrary, the proposed algorithm attempts to optimize the rate-distortion performance by directly encoding prediction residuals without the pre-quantization.

In Table V, we compare the bit-rates of the proposed algorithm and the conventional techniques in the lossless modes. In this test, we use the same quantization step size of $\Delta_{12}$ for the pre-quantization in the conventional techniques and for the quantization of prediction residuals in the proposed algorithm. Therefore, the differences in the distortions of the reconstructed models are negligible. In fact, the proposed algorithm always provides lower distortions then the conventional techniques in this test. We see that the proposed algorithm requires significantly lower bit rates than the conventional techniques. On average, the proposed algorithm requires 1.54 bpp, whereas

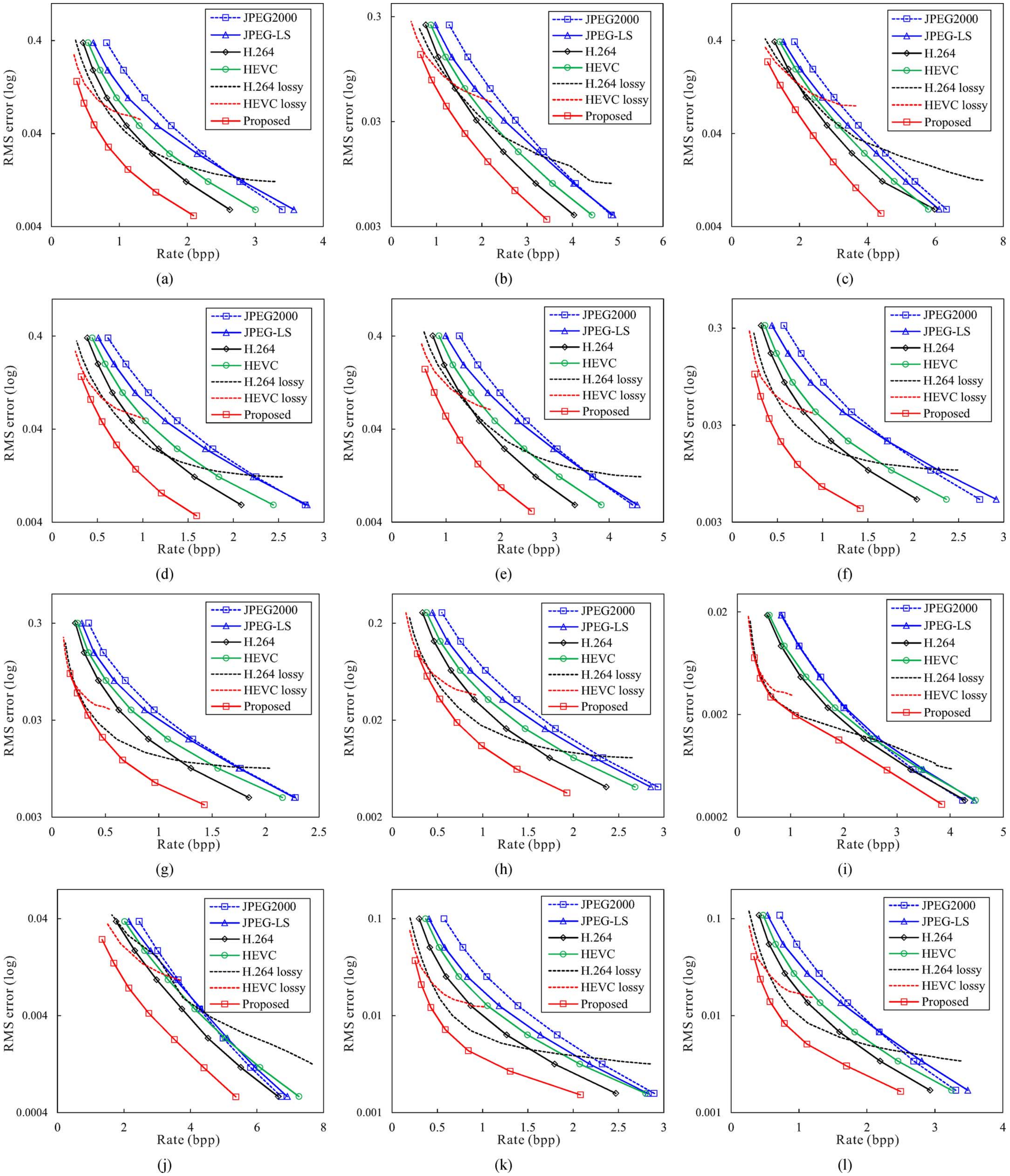


Fig. 14. Comparison of the rate-distortion curves of the proposed algorithm and the conventional techniques on (a) “Campus,” (b) “Apartments,” (c) “Country,” (d) “Amphitheatre,” (e) “Downtown,” (f) “Street,” (g) “Memorial Hall,” (h) “Korean House,” (i) “Classroom,” (j) “Gymnasium,” (k) “Parking Lot,” (k) “Parking Lot,” and (l) “Library.”.

JPEG2000, JPEG-LS, H.264/AVC, and HEVC require 2.61 bpp, 2.55 bpp, 1.98 bpp, and 2.25 bpp, respectively.

In Table VI, we also compare the encoding times of the proposed algorithm and the conventional techniques in the lossless modes at the quantization step size of $\Delta_{12}$. We measure the execution times using a PC with a 3.20 GHz Intel Core i5-4460 Processor and 4 GB RAM. While JPEG2000 and JPEG-LS spend around 20 seconds, H.264/AVC and HEVC consume 40~60 seconds. On the other hand, the proposed algorithm requires 91.46 seconds on average, since it employs complicated prediction

TABLE V
COMPARISON OF THE REQUIRED BIT-RATES OF THE PROPOSED ALGORITHM AND THE CONVENTIONAL JPEG2000 (J2K), JPEG-LS (JLS), H.264/AVC, AND HEVC. THE UNIT OF BIT-RATES IS BITS PER POINT (BPP). THE QUANTIZATION STEP SIZE FOR PREDICTION RESIDUALS IN THE PROPOSED ALGORITHM AND THE PRE-QUANTIZATION STEP SIZE IN THE CONVENTIONAL TECHNIQUES ARE BOTH FIXED TO $\Delta_{12}$. THUS, THE DISTORTIONS OF THE PROPOSED ALGORITHM ARE ALWAYS SMALLER THAN THOSE OF THE CONVENTIONAL TECHNIQUES.

| Test model | J2K | JLS | H.264 | HEVC | Proposed |
|---|---|---|---|---|---|
| Campus | 2.23 | 2.14 | 1.48 | 1.74 | 1.13 |
| Apartments | 3.36 | 3.25 | 2.48 | 2.81 | 2.13 |
| Country | 4.53 | 4.27 | 3.54 | 3.91 | 3.00 |
| Amphitheatre | 1.77 | 1.69 | 1.18 | 1.38 | 0.92 |
| Downtown | 3.05 | 2.98 | 2.07 | 2.44 | 1.59 |
| Street | 1.72 | 1.71 | 1.09 | 1.28 | 0.72 |
| Memorial Hall | 1.32 | 1.28 | 0.90 | 1.08 | 0.66 |
| Korean House | 1.80 | 1.69 | 1.26 | 1.47 | 0.99 |
| Classroom | 2.57 | 2.66 | 2.38 | 2.55 | 1.91 |
| Gymnasium | 4.99 | 5.12 | 4.53 | 5.04 | 3.52 |
| Parking Lot | 1.82 | 1.64 | 1.27 | 1.50 | 0.84 |
| Library | 2.18 | 2.19 | 1.60 | 1.82 | 1.12 |
| Average | 2.61 | 2.55 | 1.98 | 2.25 | 1.54 |

TABLE VI
COMPARISON OF THE ENCODING TIMES (IN SECONDS) OF THE PROPOSED ALGORITHM AND THE CONVENTIONAL JPEG2000 (J2K) [44], JPEG-LS (JLS) [45], H.264/AVC [46], AND HEVC [47].

| Test model | J2K | JLS | H.264 | HEVC | Proposed |
|---|---|---|---|---|---|
| Campus | 57.10 | 56.78 | 97.54 | 79.75 | 87.93 |
| Apartments | 45.11 | 44.55 | 89.82 | 67.78 | 104.15 |
| Country | 18.31 | 17.85 | 48.13 | 33.77 | 62.67 |
| Amphitheatre | 17.89 | 17.35 | 39.27 | 32.88 | 67.71 |
| Downtown | 35.23 | 34.80 | 65.53 | 53.66 | 67.62 |
| Street | 25.07 | 24.90 | 54.31 | 45.92 | 85.21 |
| Memorial Hall | 8.93 | 8.69 | 28.81 | 22.84 | 67.33 |
| Korean House | 12.60 | 12.31 | 33.84 | 26.40 | 64.10 |
| Classroom | 17.32 | 16.94 | 69.52 | 41.99 | 115.20 |
| Gymnasium | 11.54 | 10.68 | 78.08 | 37.99 | 134.59 |
| Parking Lot | 10.00 | 9.51 | 49.76 | 32.64 | 118.61 |
| Library | 10.03 | 9.40 | 52.19 | 33.28 | 122.41 |
| Average | 22.43 | 21.98 | 58.90 | 42.41 | 91.46 |

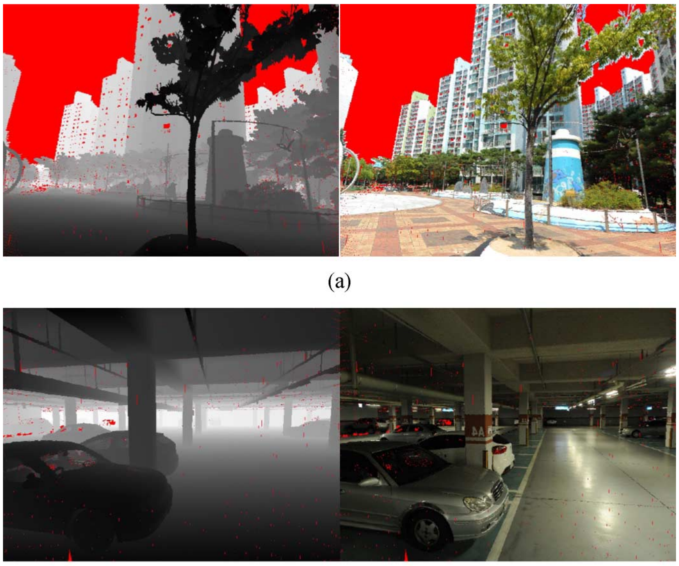


Fig. 15. Log-scaled depth images and the corresponding color images. (a) “Apartments” and (b) “Parking Lot.”

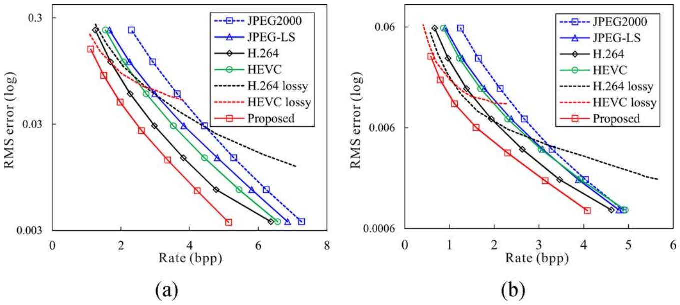


Fig. 16. Comparison of the rate-distortion curves of the proposed algorithm and the conventional techniques on the depth images in Fig. 15 . (a) “Apartments” and (b) “Parking Lot.”

modes, especially the RBF mode and the local and block plane prediction modes.

### C. Depth Image Compression

Finally, we apply the proposed algorithm to encode depth images. We generate a depth image from a captured LS3DPC, by projecting the 3D points onto an image plane of a virtual depth camera, which is located at the scanner center. Fig. 15 shows the depth images obtained from the “Apartments” and “Parking Lot” models by setting the image size to 640 × 480 and the focal length of the camera to 400 pixels. The proposed algorithm can encode those depth images in the same way as the range image compression, by replacing the roles of the azimuthal and polar angles at each pixel with the horizontal and vertical coordinates. Fig. 16 compares the rate-distortion curves of the proposed algorithm with those of the conventional image and video coding techniques on the depth images in Fig. 15. We see that the proposed algorithm also yields a higher coding gain on these depth images than the conventional techniques.

## VI. CONCLUSIONS

We proposed a geometry compression algorithm for LS3DPCs to encode radial distances in a range image. For efficient compression, we designed twelve prediction modes for radial distances, which are performed adaptively in one of the three domains: the range image domain, the height image domain, and the 3D domain. The proposed algorithm first partitions a range image into blocks of various sizes from 8 × 8 to 32 × 32. It then finds the optimal prediction mode for each block by comparing the rate-distortion costs of the twelve modes. Then, the proposed algorithm applies the optimal mode to all pixels in the block and encodes the prediction residuals using a context-based entropy coder. Experimental results demonstrated that the proposed algorithm provides a much higher compression gain than the conventional image and video coding techniques.

Further research issues include the analysis of the correlation between range and color images, the joint compression of range and color images, the progressive compression of range images, the compression of overlapped range images obtained from multiple scans of the same scene, and the efficient handling of holes in range images.

## References

[1] C. Zhang, Q. Cai, P. A. Chou, Z. Zhang, and R. Martin-Brualla, "Viewport: A distributed, immersive teleconferencing system with infrared dot pattern," *IEEE Multimedia*, vol. 20, no. 1, pp. 17–27, Jan.-Mar. 2013.

[2] D. S. Alexiadis, D. Zarpalas, and P. Daras, "Real-time, full 3-D reconstruction of moving foreground objects from multiple consumer depth cameras," *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 339–358, Feb. 2013.

[3] M. Waschbüsch, S. Würmlin, D. Cotting, and M. Gross, "Point-sampled 3D video of real-world scenes," *Signal Process.: Image Commun..*, vol. 22, no. 2, pp. 203–216, Feb. 2007.

[4] W. Guan, S. You, and U. Neumann, "Recognition-driven 3D navigation in large-scale virtual environments," in *Proc. IEEE Virtual Reality*, Mar. 2011, pp. 71–74.

[5] Cyberware 3030 MS Scanner Cyberware [Online]. Available: http://cyberware.com/products/scanners/ms.html

[6] T. Pylvänäinen, J. Berclaz, T. Korah, V. Hedau, M. Aanjaneya, and R. Grzeszczuk, "3D city modeling from street-level data for augmented reality applications," in *Proc. Int. Conf. 3D Imaging, Modeling, Process., Visualizat. Transmiss.*, Oct. 2012, pp. 238–245.

[7] T. Korah and Y.-T. Tsai, "Urban canvas: Unfreezing street-view imagery with semantically compressed LIDAR pointclouds," in *Proc. IEEE Int. Symp. Mixed Augment. Real.*, Oct. 2011, pp. 271–272.

[8] F. Lafarge and C. Mallet, "Creating large-scale city models from 3D-point clouds: A robust approach with hybrid representation," *Int. J. Comput. Vis.*, vol. 99, no. 1, pp. 69–85, 2012.

[9] G. Vosselman, P. Kessels, and B. Gorte, "The utilisation of airborne laser scanning for mapping," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 6, no. 3-4, pp. 177–186, Mar. 2005.

[10] C. Früh and A. Zakhor, "An automated method for large-scale, ground-based city model acquisition," *Int. J. Comput. Vis.*, vol. 60, no. 1, pp. 5–24, 2004.

[11] M. Jaboyedoff, T. Oppikofer, A. Abellán, M.-H. Derron, A. Loye, R. Metzger, and A. Pedrazzini, "Use of LIDAR in landslide investigations: A review," *Nat. Hazards*, vol. 61, no. 1, pp. 5–28, 2012.

[12] S. Slob and R. Hack, "3D terrestrial laser scanning as a new field measurement and monitoring technique," *Lecture Notes in Earth Sci.*, vol. 104, pp. 179–189, 2004.

[13] N. Brodu and D. Lague, "3D terrestrial LIDAR data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology," *ISPRS J. Photogramm. Remote Sens.*, vol. 68, pp. 121–134, Mar. 2012.

[14] G. Zheng and L. M. Moskal, "Computational-geometry-based retrieval of effective leaf area index using terrestrial laser scanning," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 10, pp. 3958–3969, Oct. 2012.

[15] S. Gumhold, Z. Karni, M. Isenburg, and H.-P. Seidel, "Predictive point-cloud compression," in *Proc. ACM SIGGRAPH Sketches*, 2005.

[16] M. Waschbüsch, M. Gross, F. Eberhard, E. Lamboray, and S. Würmlin, "Progressive compression of point-sampled models," in *Proc. Eurograph. Symp. Point-Based Graph.*, 2004, pp. 95–102.

[17] A. Kalaiah and A. Varshney, "Statistical geometry representation for efficient transmission and rendering," *ACM Trans. Graph.*, vol. 24, no. 2, pp. 348–373, 2005.

[18] T. Ochotta and D. Saupe, "Compression of point-based 3D models by shape-adaptive wavelet coding of multi-height fields," in *Proc. Eurograph. Symp. Point-Based Graph.*, 2004, pp. 103–112.

[19] S. Fleishman, D. Cohen-Or, M. Alexa, and C. T. Silva, "Progressive point set surfaces," *ACM Trans. Graph.*, vol. 22, no. 4, pp. 997–1011, Oct. 2003.

[20] M. Botsch, A. Wiratanaya, and L. Kobbelt, "Efficient high quality rendering of point sampled geometry," in *Proc. Eurograph. Workshop Rendering*, 2002, pp. 53–64.

[21] R. Schnabel and R. Klein, "Octree-based point-cloud compression," in *Proc. Eurographics Symp. Point-Based Graphics*, 2006, pp. 111–120.

[22] Y. Huang, J. Peng, C.-C. J. Kuo, and M. Gopi, "Octree-based progressive geometry coding of point clouds," in *Proc. Eurograph. Symp. Point-Based Graph.*, 2006, pp. 103–110.

[23] S.-B. Park and S.-U. Lee, "Multiscale representation and compression of 3-D point data," *IEEE Trans. Multimedia*, vol. 11, no. 1, pp. 177–182, Jan. 2009.

[24] J. Smith, G. Petrova, and S. Schaefer, "Progressive encoding and compression of surfaces generated from point cloud data," *Comput. Graph.*, vol. 36, no. 5, pp. 341–348, Aug. 2012.

[25] P. Merkle, A. Smolic, K. Müller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *Proc. IEEE ICIP*, Oct. 2007, pp. 201–204.

[26] M. Maitre and M. N. Do, "Joint encoding of the depth image based representation using shape-adaptive waveletes," in *Proc. IEEE ICIP*, Oct. 2008, pp. 1768–1771.

[27] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Müller, P. de With, and T. Wiegand, "The effects of multiview depth video compression on multiview rendering," *Signal Process.: Image Comm.*, vol. 24, no. 1-2, pp. 73–88, Jan. 2009.

[28] J. Ruiz-Hidalgo, J. R. Morros, P. Aflaki, F. Calderero, and F. Marqués, "Multiview depth coding based on combined color/depth segmentation," *J. Vis. Commun. Image Rep.*, vol. 23, no. 1, pp. 42–52, Jan. 2012.

[29] Y. Gao, G. Cheung, T. Maugey, P. Frossard, and J. Liang, "3D geometry representation using multiview coding of image tiles," in *Proc. IEEE ICASSP*, May 2014, pp. 6198–6202.

[30] RIEGL VZ-400 3D Terrestrial Laser Scanner, RIEGL. [Online]. Available: http://www.riegl.com/nc/products/terrestrial-scanning/produktdetail/product/scanner/5/

[31] RIEGL RiSCAN PRO Software for 3D Terrestrial Laser Scanner, RIEGL. [Online]. Available: http://www.riegl.com/index.php?id=221

[32] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, "Reconstruction and representation of 3D objects with radial basis functions," in *Proc. ACM SIGGRAPH*, 2001, pp. 67–76.

[33] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[34] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

[35] T. Wiegand and B. Girod, "Lagrange multiplier selection in hybrid video coder control," in *Proc. IEEE ICIP*, Oct. 2001, pp. 542–545.

[36] J.-K. Ahn, D.-Y. Lee, M. Ahn, J. D. K. Kim, C. Kim, and C.-S. Kim, "Progressive compression of 3D triangular meshes using topology-based Karhunen-Loève transform," in *Proc. IEEE ICIP*, Sep. 2010, pp. 3417–3420.

[37] J.-K. Ahn, Y. J. Koh, and C.-S. Kim, "Efficient fine-granular scalable coding of 3D mesh sequences," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 485–497, Apr. 2013.

[38] A. Moffat, R. Neal, and I. H. Witten, "Arithmetic coding revisited," in *Proc. IEEE Data Compress. Conf.*, Mar. 1995, pp. 202–211.

[39] E. Shusterman and M. Feder, "Image compression via improved quadtree decomposition algorithms," *IEEE Trans. Image Process.*, vol. 3, no. 2, pp. 207–215, Mar. 1994.

[40] *Information Technology – JPEG 2000 Image Coding System – Part 1: Core Coding System*, ISO/IEC 15444-1, 2000.

[41] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.

[42] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Rec. H. 264 and ISO/IEC 14496-10 AVC, Mar. 2005.

[43] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

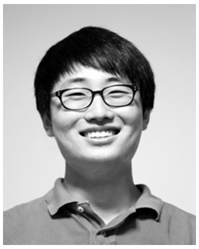
[44] JasPer 1.9. [Online]. Available: http://www.ece.uvic.ca/~frodo/jasper/

[45] JPEG-LS Reference Encoder, HP Laboratories [Online]. Available: http://www.hpl.hp.com/loco/locodown.htm

[46] *H.264/AVC Latest Reference Software*, (JM 18.5) [Online]. Available: http://iphome.hhi.de/suehring/tml/download/

[47] *HEVC Latest Reference Software*, (HM 12.0) [Online]. Available: http://hevc.kw.bbc.co.uk/git/w/jctvc-hm.git

**Jae-Kyun Ahn** received B.S., M.S., and Ph.D. degrees in electrical engineering from Korea University, Seoul, Korea, in 2007, 2009, and 2014, respectively. His research topics include video segmentation, 3-D mesh compression, and 3-D multimedia processing.



**Kyu-Yul Lee** received the B.S. degree in electrical and computer engineering from Ulsan National Institute of Science and Technology, Ulsan, Korea, in 2013, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. His research interests include correspondence matching and machine learning including deep learning.

**Jae-Young Sim** (S'02–M'06) received the B.S. degree in electrical engineering and the M.S. and Ph.D. degrees in electrical engineering and computer science from Seoul National University, Seoul, Korea, in 1999, 2001, and 2005, respectively. From 2005 to 2009, he was a Research Staff Member, Samsung Advanced Institute of Technology, Samsung Electronics Company, Ltd., Yongin, Korea. In 2009, he joined the School of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology, Ulsan, Korea, where he is now an Associate Professor. His research interests include image and 3-D visual signal processing, multimedia data compression, and computer vision.

**Chang-Su Kim** (S'95–M'01–SM'05) received the Ph.D. degree in electrical engineering from Seoul National University with a Distinguished Dissertation Award in 2000. From 2000 to 2001, he was a Visiting Scholar with the Signal and Image Processing Institute, University of Southern California, Los Angeles. From 2001 to 2003, he coordinated the 3D Data Compression Group in National Research Laboratory for 3D Visual Information Processing in SNU. From 2003 and 2005, he was an Assistant Professor in the Department of Information Engineering, Chinese University of Hong Kong. In September 2005, he joined the School of Electrical Engineering, Korea University, where he is now a Professor. His research topics include image processing and computer vision. In 2009, he received the IEEK/IEEE Joint Award for Young IT Engineer of the Year. In 2014, he received the Best Paper Award from *Journal of Visual Communication and Image Representation* (JVCI). He has published more than 210 technical papers in international journals and conferences. Dr. Kim is an Editorial Board Member of JVCI and an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING.