Kinect-Like Depth Data Compression

Jingjing Fu, Dan Miao, Weiren Yu, Shiqi Wang, Yan Lu, and Shipeng Li

***Abstract*—Unlike traditional RGB video, Kinect-like depth is characterized by its large variation range and instability. As a result, traditional video compression algorithms cannot be directly applied to Kinect-like depth compression with respect to coding efficiency. In this paper, we propose a lossy Kinect-like depth compression framework based on the existing codecs, aiming to enhance the coding efficiency while preserving the depth features for further applications. In the proposed framework, the Kinect-like depth is reformed first by divisive normalized bilateral filter (DNBL) to suppress the depth noises caused by disparity normalization, and then block-level depth padding is implemented for invalid depth region compensation in collaboration with mask coding to eliminate the sharp variation caused by depth measurement failures. Before the traditional video coding, the inter-frame correlation of reformed depth is explored by proposed 2D+T prediction, in which depth volume is developed to simulate 3D volume to generate pseudo 3D prediction reference for depth uniqueness detection. The unique depth region, called active region is fed into the video encoder for traditional intra and inter prediction with residual coding, while the inactive region is skipped during depth coding. The experimental results demonstrate that our compression scheme can save 55%–85% in terms of bit cost and reduce coding complexity by 20%–65% in comparison with the traditional video compression algorithms. The visual quality of the 3D reconstruction is also improved after employing our compression scheme.**

***Index Terms*—2D+T prediction, denoising, depth volume, Kinect-like depth, lossy compression, padding.**

I. Introduction

In the past few decades, the rapid development of sensor technology offers consumers powerful tools for perceiving and recording the real world. This evolution is especially evident for image sensors, which can capture 2D optical images and represent them in the form of digital signals. As the main component of digital multimedia data, numerous techniques have been proposed to process the image and video data for efficient compression and transmission [1], [2], and related standards [3]–[5] have been established by the multimedia industry. As technology has advanced, the 2D description of the real world has become insufficient to meet the increasing sensory requirements. Various devices have been invented as an attempt to access the 3D information of the physical word, including time-of-light (TOF) camera [6], stereo camera [7], laser scanner [8], and structured light camera [9]. These sensors all measure the distance from the camera to the target object surface, namely depth, by utilizing light wave properties, but their working principles are distinct from each other. For example, the TOF measures the distance by detecting the light wave phase shift after the reflection from the object surface, while the stereo camera generates a disparity map by stereo matching between the left and right view images. In general, depth sensors are not as popular as traditional image sensors due to their high cost and bulky size.

The launch of Kinect [10] facilitates the possibility of depth capture in real-time at a low cost for consumers and has achieves strong commercial success through Xbox immersive gaming, in which real-time depth information is used to assist skeletal tracking in conjunction with the texture information from the RGB sensor of the Kinect. Besides meeting the industry success, Kinect has also drawn the attention of researchers for its potential to aid in a variety of applications, such as object tracking, dynamic 3D reconstruction, activity recognition, and so on. Some improvements [11], [12] have already been achieved by using Kinect. More specially, Weise [11] built up a real-time low-cost character animation system that avoided the substantial manual post-processing by using Kinect as the acquisition device instead of the complex acquisition system. In Shotton's work [12], quick and accurate recognition of human pose can be realized using the single depth image captured from Kinect.

All these depth related applications are developed locally, since the raw depth data size is too large to be transmitted through the network. A high computing capacity is therefore required for the local data processing, and the depth data can hardly be shared with other machines in real-time. In order to transmit depth data through the network, we came up with a remote sensor system shown in Fig. 1, in which the depth data is captured and compressed by the local processor, then transmitted to the server side and reconstructed for further processing. With rich database and computing resource available at the server side, more depth-related applications can be leveraged. Depth compression is of great importance, because the compression's efficiency and complexity directly affect system latency and the effectiveness of the reconstructed depth data.

Similar to RGB texture, depth maps are composed of tremendous pixels, which are organized on a regular, 2D sampling grid, but each depth pixel value represents the distance from the camera to the target object surface rather than the color information. The compression techniques for the RGB texture sequence have been studied for years and a number of mature solutions are available that achieve acceptable compression perfor-

Manuscript received June 07, 2012; revised September 17, 2012; accepted November 11, 2012. Date of publication February 15, 2013; date of current version September 13, 2013. This work was done while D. Miao, W. Yu, and S. Wang were with Microsoft Research Asia as research interns. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ebroul Izquierdo.

J. Fu, Y. Lu, and S. Li are with the Media Computing Group, Microsoft Research Asia, Beijing, China (e-mail: jifu@microsoft.com; yanlu@microsoft.com; spli@microsoft.com).

D. Miao is with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China (e-mail: miaodan@mail.ustc.edu.cn).

W. Yu is with the School of Computer Science and Engineering, Beihang University, Beijing, China (e-mail: yuweiren@act.buaa.edu.cn).

S. Wang is with the Institute of Digital Media, Peking University, Beijing, China(e-mail: sqwang@jdl.ac.cn).

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/TMM.2013.2247584

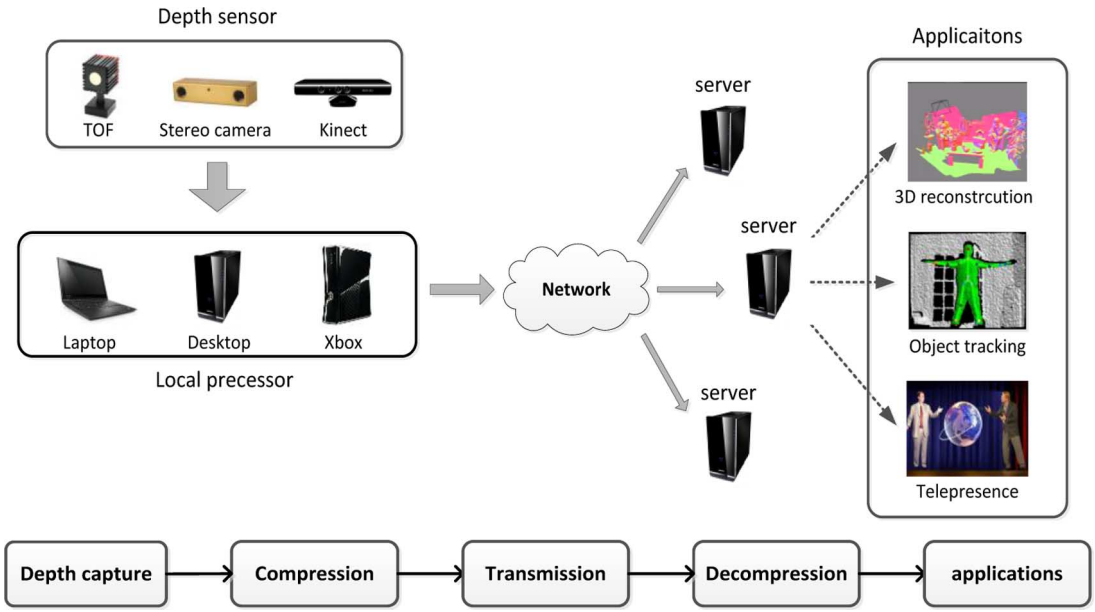


Fig. 1. Up: our proposed system architecture; bottom: block diagram of our system.

mance, such as JPEG [3], MPEG-2 [4], H.264/AVC [5], and so on. However, Kinect-like depth compression has seldom been investigated.

In this paper, we propose a novel depth compression framework based on the existing codec for Kinect-like depth. Instead of directly encoding the raw depth data, we introduce necessary depth preprocessing to stabilize the depth values in the spatial domain, and suppress the step-shaped artifacts with the inherent depth features preserved. The block-level depth padding is then implemented for invalid depth region compensation in collaboration with mask coding, so that considerable bits are saved during the block-based transform. The processed depth is fed into the depth encoding module, in which the depth is first predicted by 2D+T depth reference and then the recognized active regions are passed to a conventional video coding module. In this way, more coding resource is allocated to the important depth information, and the compression ratio is increased without a drop in data effectiveness. The experimental results shows that the bit rate is reduced by 55–85% of the original bitrate, and the complexity is reduced by 20–65%. Comparisons of the render results show that the render view of our algorithm is visually better than that of the original coding scheme.

The rest of paper is organized as follows. In Section II, some related research is presented. By investigating the generation principle of Kinect-like depth, the depth measurement error is modeled in Section III. Section IV describes the proposed framework for depth compression. The depth reformation algorithm is introduced in detail in Section V, including depth denoising and padding techniques. In Section VI, depth coding with 2D+T prediction is explained. The experimental results are provided to verify the performance of our coding system in Section VII. Finally, Section VIII concludes the paper.

II. Related Work

As Kinect-like depth is a kind of range data in terms of its physical meaning, it can be converted to point cloud for sequential predictive compression [13] or geometry compression based on octree structure [14]. But these point cloud compression algorithms focus on the static scanned point cloud, and cannot handle the replicate geometry information among depth frames. Recently, Kammerl [15] proposed point cloud compression for the point cloud library, in which real-time spatial changes are detected based on XOR comparison of octree structure to remove the redundant 3D points and both the position information and color information of the residue points are entropy encoded.

The physical information of a visual object can be transformed to 2D or 3D meshes, and the object's transformation is compressed in the form of mesh animation in MPEG-4. For specific 2D meshes (e.g., face), MPEG-4 defines a complete set of animation parameters to describe the face animation, and the parameters are compressed by temporal prediction [16]. In Frame-based animated mesh compression (FAMC) [17], the dynamic 3D mesh is compressed by combining a model-based motion-compensation strategy with transform/predictive coding of residual errors. The depth map can also be represented by triangular mesh for mesh compression [18], [19]. Grewatsch *et al.* [20] presented a mesh-based coding scheme that compressed the 3D depth information using the MPEG-4 3DMC coder. In Chai's work [21], an adaptation triangular mesh generation algorithm is introduced for a depth map coding scheme, where a complicated tree structure has to be maintained frame by frame. In the mesh-based depth schemes, extracting the mesh from each raw depth frame costs additional computing and accordingly coding complexity increases.

From another point of view, a depth map can be regarded as a grey image of high dynamic range. One straightforward approach for compressing depth map sequences is to encode them using conventional image/video compression algorithms [22]. Grewatsch and Muller [23] investigate and evaluate several depth coding algorithms, and demonstrate that the standard H.264/AVC codec outperforms the mesh-based coding when compressing a sequence of depth maps. Considering that depth

data contains smooth areas partitioned by sharp edges, with very limited texture, some depth compression algorithms [24]–[27] are proposed as an attempt to achieve more efficient depth compression based on the existing image/video coding framework, such as platelet based coding [24], [25], edge based coding [26], [27]. In the platelet based coding schemes, depth frame is segmented into regions with different sizes, and each region is encoded by utilizing the homogeneousness of the depth map. For example, Marvon *et al.* [24] segmented the depth using a quadtree decomposition for depth coding, where each block is modeled by one of three pre-defined piecewise linear functions. Milani and Calvagno [25] partitioned the depth map based on graph-based image segmentation. The generated regions associated with the average depth value and the residual frame is encoded by a standard H.264/AVC Intra coder. In an attempt to reduce the bit cost caused by the non-zero high frequency transform coefficients in the edge block, Shen *et al.* [26] proposed edge-adaptive transform, which avoids filtering cross edges by encoding edge positions explicitly. In [27], shape an adaptive wavelets are employed to ensure that the support for the wavelet lies in the same region separated by edges.

In recent years, the data format including the 2D multiview videos and corresponding depth is proposed to benefit the intermediate view synthesis in the 3D video applications. The color video is considered as the side information to assist the depth compression [28]. By using the structure similarity between depth map and corresponding video, Liu introduced a new in-loop filter to suppress the coding artifact and a new intra coding mode to reconstruct depth map with sparse representations of depth blocks.

However, these depth compression schemes cannot be adopted to perform Kinect-like depth compression directly for two primary reasons. First, these schemes are designed to generate depth data of high quality. In contrast to ideal depth, Kinect-like depth data is characterized by its noise and instability, causing the temporal and spatial correlation to be destroyed to some degree. Moreover, Kinect depth data has a high dynamic range that is different from the traditional 8-bit depth data. The coding schemes implemented based on 8-bit image/video codecs cannot be applied to Kinect-like depth compression. In order to encode the high dynamic range depth data in real-time, Mehrotra *et al.* [29] proposed a near lossless depth compression to encode the scaling reciprocals of the depth values pixel by pixel. Although significant data size reduction is achieved, the strong relationships among the neighboring frames are barely considered in the algorithms.

III. Error Model of Kinect-Like Depth

Due to the distinct generation principle, Kinect-like depth data possesses special characteristics distinguished from the traditional image data. In order to achieve high efficiency depth coding, we investigate generation principle [30] of Kinect-like depth and derive a theoretical model for depth measurement error to assist compression framework design.

A. Depth Generation Principle

As a kind of structured light camera, Kinect depth is derived from the distortion between the projected infrared light pattern and the received one. To be more specific, the infrared projector of the Kinect emits pseudo-random light pattern through a diffractive mask, so that each speckle in the pattern can be distinguished from the others. With the observed light pattern by infrared sensors, depth value is derived by triangulation between the observation and the reference light pattern that is obtained by capturing a plane at a known distance beforehand and hard coded in the memory of the sensor. Once a speckle is projected on an object whose distance to the sensor is different from that of the reference plane, the speckle's position in the received image will be shifted along the direction of the baseline between the projector P and the perspective center of the infrared camera S. These shifts are measured for all speckles by a simple image correlation procedure, which yields a disparity map.

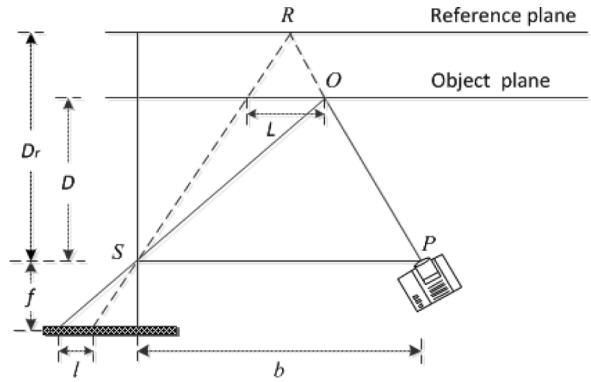


Fig. 2. Schematic representation of depth-disparity relation.

Fig. 2 illustrates the relation between the distance D of an object point to the IR sensor and the distance D_r of a reference plane. To simplify the model, we assume that the origin of the depth coordinate system is located at the perspective center of the IR sensor. According to the similarity of the triangles, we have

$$\frac{L}{b} = \frac{(D_r - D)}{D_r} \tag{1}$$

$$\frac{l}{f} = \frac{L}{D} \tag{2}$$

where f is the focal length of the IR sensor; l is the relative shift length (disparity); b is the base length. After combining (1) and (2), the depth is calculated as follows

$$D = \frac{D_r}{1 + D_r \cdot \frac{l}{(f \cdot b)}} \tag{3}$$

B. Kinect-Like Depth's Characteristics

Kinect-like depth is derived from the infrared light disparity map. The ideal disparity map is characterized by continuity and uniqueness. That is, the disparity varies continuously within object surfaces and the disparity at a fixed coordinate has a unique value. Unfortunately, due to the constraint of the speckle's granularity and the instability of the received pattern, the generated disparity map suffers from step-shaped fluctuation and inconsistent measurements over time. Furthermore, the disparity absence caused by imaging failure introduces depth information

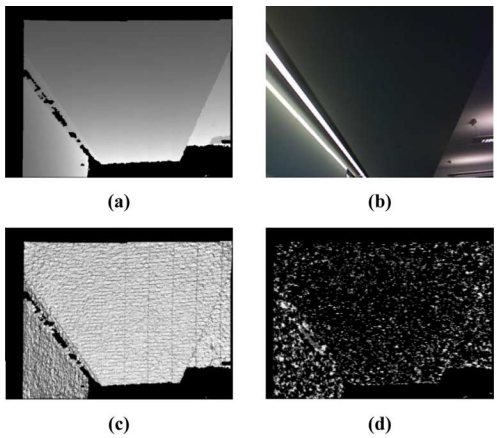


Fig. 3. Example of Kinect depth image and its corresponding RGB image (a) depth map after calibration (b) RGB image (c) normal map of the Kinect depth (d) neighboring depth frames difference.

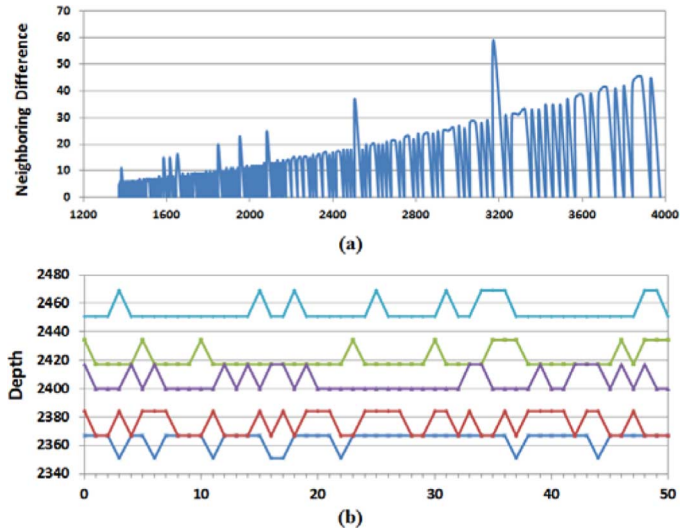


Fig. 4. Illustration of spatial and temporal depth characteristics. Top: plot of the neighboring depth difference; the horizontal axis represents depth value. Bottom: depth variation with the time; the horizontal axis represents frame number.

loss on the mirror surfaces, occlusion regions, and out-of-range surfaces.

A typical example is illustrated in Fig. 3. The RGB images and the depth images are captured simultaneously from a fixed Kinect camera by the Kinect Windows SDK [31]. The captured Kinect depth has a large dynamic range from 0 to 4000, and each pixel value is represented by a 12-bit integer. For display, the depth map is scaled down to 8-bit grey level representation (see Fig. 3(a)). Since it is difficult to observe the depth variation through a grey depth image, we generate a corresponding normal map to simulate the 3D rendering scene of the depth map. From the generated normal map (see Fig. 3(c)), it can be observed that noticeable step-shaped fluctuations exist on the roof surface, and the depth information is lost along the intersection between the roof and the wall. The difference between the neighboring depth frame always has significant values (see Fig. 3(d)), even if the distance is unchanged over time.

In order to analyze the spatial and temporal variations of the depth numerically, we choose the depth data along a vertical line on the roof surface. The variation is investigated using the neighboring depth difference (see Fig. 4(a)). As the ideal depth increases linearly, the depth difference should be uniform. However, the real depth difference fluctuates in a strange way—the neighboring difference keeps increasing with the depth value. Even if a static scene is captured by a fixed Kinect, the depth values at a fixed coordinates change from time to time (see Fig. 4(b)). The inherent spatial continuity of the depth map is destroyed by the inaccurate disparity measurement and disparity detection failure, whereas the temporal consistency is broken by the random interference of the speckle correlation detection. Given that the texture video codec is developed upon the spatial and temporal correlation exploration, the imperfect Kinect depth data is more difficult to compress in comparison to the texture data.

C. Kinect Depth Error Modeling

In this subsection, the Kinect-like depth error is analyzed and modeled to benefit depth compression. There are several possible reasons for the depth measurement error: a) Light condition interference. Under strong light, laser speckles appear in low contrast in the infrared image, which may give rise to false disparity detection and speckle recognition failure. b) Imaging geometry. When the depth is out of the measurement range or the surface's tangent plane parallels the ray casting direction, depth information is lost due to the absence of reflected light speckle. c) Disparity normalization. The disparity is normalized during the depth measurement, and sequentially the normalization error is added to the generated depth, which results in step-shaped depth fluctuation.

As depth is derived from disparity, we formulate an inaccurate disparity in terms of the main error reasons.

$$\hat{l} = M \cdot (l + r_d + r_n) \tag{4}$$

Let l be the truth disparity map, and $\hat{l}$ the disparity generated for Kinect depth deviation. The disparity mask is represented by M, indicating whether the disparity value is valid at that position. r_d is the disparity error introduced by the light pattern misidentification. The raw disparity length is normalized during depth measurement, such that $\hat{l}$ can be substituted for $(ml^* + n)$, with l^* the normalized disparity and m, n the parameter of normalized disparity. r_n is the normalization error caused by disparity round-off and is equal to $-n$, with $r_n \in [-l^*, 0]$.

The relationship between Kinect depth $\widetilde{D}$ and true depth D can be formulated as follows,

$$\widetilde{D} = M \cdot (D + e) \tag{5}$$

where e is the depth measurement error caused by degradation of the disparity. In a region with valid depth values, the error between the true depth and the output depth is

$$e = \widetilde{D} - D = \frac{r_d + r_n}{f \cdot b} D\widetilde{D} \tag{6}$$

The depth error can be decomposed to identification error e_d and normalization error e_n in terms of their origins. Assuming $\widetilde{D} - D \ll \widetilde{D}$, the true depth D can be replaced by $\widetilde{D}$ in (6) for approximation. Since the focal length and base length are both

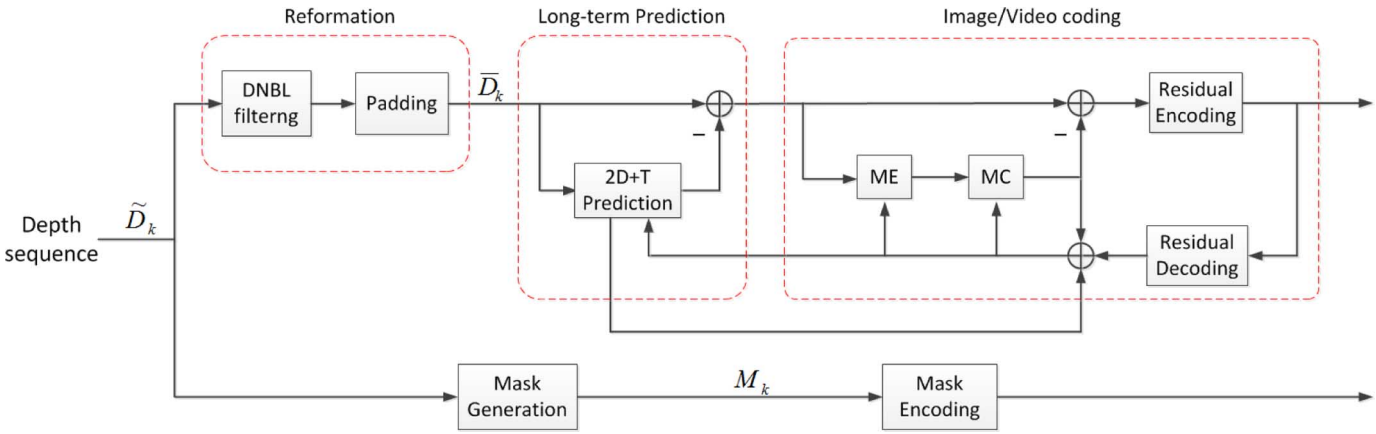


Fig. 5. The framework of our proposed depth compression scheme.

constant for Kinect, C_0 is used to represent the constant factor $1/fb$.

$$e = e_d + e_n \tag{7}$$

with $e_d = C_0 \widetilde{D}^2 r_d$ and $e_n = C_0 \widetilde{D}^2 r_n$. The levels of error e_d and e_n are proportional to the square of the corresponding depth value. This inference is verified by the Fig. 4(a), since the neighboring depth difference of the smooth region implies the upper bound of the normalization error. The difference between neighboring frames is partially introduced by the normalization, but the underlying cause is the of time-variant identification error.

IV. Our Proposed Depth Compression Framework

Considering the special characteristics of Kinect depth, we propose a novel compression framework, aiming to enhance the coding efficiency while preserving the inherent depth features. In our framework, the original depth is reformed under an error bound constraint to rebuild the spatial correlation for efficient intra coding. The accumulated depth data is used to judge the uniqueness of the input depth content. If the content can be reconstructed from the previous depth frames using volumetric integration, it will be skipped during the depth coding. In contrast, once the content is recognized as new emerging content, it will be compressed at a high quality to maintain as many details as possible. In this way, the computing and bitrate resource can be better allocated among different depth contents.

The specific implementation of our framework is depicted in Fig. 5. First, the input depth frame $\widetilde{D}_k$ is filtered by a divisive normalized bilateral filter (DNBL) in the spatial domain in an attempt to reduce the normalization error e_n. The filtered depth frame is divided into blocks, and the invalid depth region within the block is padded by its spatially neighboring depth. Since the disparity map is modified during reformation, the original binary mask M_k is encoded by *JBIG* [32] and transmitted to the decoder for depth recovery. After reformation, the depth becomes much friendlier to block based compression.

Before normal video encoding, the reformed depth frame is predicted following the error tolerant rule with the depth reference generated by long-term depth volumetric integration. As a result, the input depth frame is segmented into inactive and active regions, representing stable surfaces and unstable surfaces, respectively. The inactive regions are skipped during coding, while the active regions are fed into the video encoder for traditional intra and inter prediction and residual coding. Given that the region segmentation may change according to the depth content frame by frame, the depth reference inside the video encoder must be completely maintained and the synchronizing must be updated with 2D+T prediction reference in the inactive regions. As a loop, the reconstructed depth of the video encoder is fed back to the 2D+T prediction module for depth volume updating.

The novelty of our proposed framework can be summarized in three aspects. First, we analyze the Kinect-like depth's characteristics in-depth based on its generation principle and model the depth measurement error. Secondly, in contrast to the conventional video compression, the original depth is preprocessed before compression to avoid unnecessary coding cost because of the depth measurement error and all the preprocessing algorithms are tailored for the Kinect-like depth. Last but not least, by identifying the uniqueness of the input depth content through 2D+T prediction, more bitrate is allocated to the critical depth contents, where the historical depth information is inadequate for depth reconstruction.

V. Kinect-Like Depth Reformation

The Kinect-like depth is reformed in two steps to rebuild the spatial correlation for efficient intra coding. First, depth is filtered by divisive normalized bilateral filtering to eliminate depth normalization errors; second, the depth is padded for depth hole compensation.

A. Divisive Normalized Bilateral Filtering

To attenuate signal fluctuation and enhance spatial correlation, the input depth frame is first filtered to reduce normalization error. Since disparity normalization error is a random variable with uniform distribution in a limited range, the absolute value of the depth normalization error is no large than the bound $C_0 l^* \widetilde{D}^2$. With this priori knowledge, we can distinguish edges generated by normalization from the inherent depth edges and then deal with normalization error reduction as a denoising problem.

As an edge-preserving filter, the bilateral filter [33] can provide a weighted average of nearby pixels as the filtered result, with two Gaussian kernels defining the weight: a domain filter

kernel and a range filter kernel. The domain filter kernel is used to describe the geometric closeness between two pixels, while the range filter kernel is used to measure the photometric similarity. The 2D Gaussian kernel is denoted as $G_\sigma(x)$ and its formula is

$$G_\sigma(x) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{8}$$

Assume p is the position of the pixel that is to be filtered and its pixel value is $I(p)$. The local neighborhood set of p is denoted as $S(p)$, which may have influence on $I(p)$. The bilateral filter can be formulated as follows:

$$\mathrm{BF}[I]_p = \frac{1}{W_p} \sum_{q \in S(p)} G_{\sigma_s}(\|p-q\|) G_{\sigma_r}(\|I(p)-I(q)\|) I(q) \tag{9}$$

where $\mathrm{BF}[I]_p$ is the filtered result at the position p and W_p is the normalization factor.

$$W_p = \sum_{q \in S(p)} G_{\sigma_s}(\|p-q\|) G_{\sigma_r}(\|I(p)-I(q)\|) \tag{10}$$

σ_s and σ_r are the filtering parameters, G_{σ_s} is a spatial Gaussian that decreases the influence of distant pixels, and G_{σ_r} is a range Gaussian that decreases the influence of pixels with a large intensity difference. The weight of each pixel q in the set $S(p)$ is determined by its position and value difference form the current pixel, and the filtering could efficiently smooth the image while preserving the edge. As we analyzed in the Section III, the level of depth error is proportional to the square of the corresponding depth values. Therefore, if the depth is directly processed by bilateral filtering with a uniform range filter kernel without considering variation on error level, the region with small depth values will be over smoothed, and the normalization error in the region with large depth values will be preserved as depth edges. In order to preserve the true depth edges and suppress the fake ones, we apply an adaptive scaling technique to the pixel difference, called divisive normalized bilateral filtering (DNBL) [34]. In DNBL, range Gaussian kernel G_{σ_r} in bilateral filter is replaced by divisive normalized range Gaussian kernel G'_{σ_r}, which is defined as follows:

$$G'_{\sigma_r}(\|I(p)-I(q)\|) = G'_{\sigma_r}\left(\left\|\frac{I(p)-I(q)}{\Theta(I(p))}\right\|\right) \tag{11}$$

where $\Theta(I(p))$ is the normalization error bound of $I(p)$ and equals to $C_0 l^* I(p)^2$. The divisive normalization is equivalent to the operation that the difference between two pixels is normalized in terms of their depth range.

Fig. 6(a) shows a typical case of a depth normalization error. The true depth smoothly varies with the x-coordinates value. As shown in Fig. 6(b), after disparity normalization, the depth becomes step-shaped and its step size increases with the depth value. The denoised results of the example are given in Fig. 6(c). Although the denoised depth is different from the original Kinect depth, it may approach the ground truth in comparison with the original one. Meanwhile, the elimination of the stair-wise jump can greatly reduce the bit cost on non-zero high-frequency coefficients.

The depth variation trend can be represented by its normal map. For the plane surface (e.g., Fig. 6(d)), the normal of

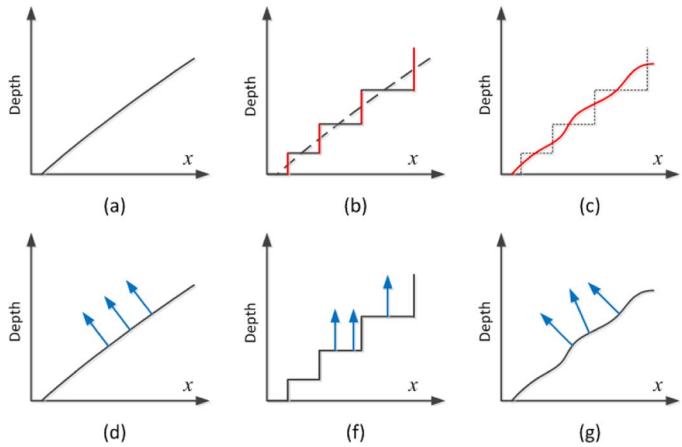


Fig. 6. Illustration of the step-sharped depth and its filtering results.

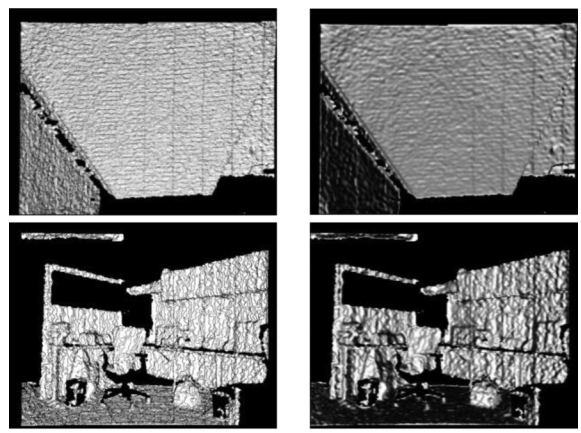


Fig. 7. Normal maps comparison between the original depth (left) and their filtering results (right), the depth sequences from the top to bottom: "*Roof*" and "*Cubicle*".

different vertices should be perpendicular to the surface and the same to each other. For Kinect depth, due to the stair-wise variation, most of the vertex normals are perpendicular to the axis. In Fig. 6(f), we can hardly detect the slope of the surface by its normal. After the filtering, the vertex normal undergoes smooth changes and approaches the ground truth, see Fig. 6(g). To verify our inference, Fig. 7 shows the normal maps of the original and the filtered depth for comparison. The left window of the video is the normal map of the raw Kinect depth sequence, while the right one shows the normal map of the filtered depth sequence. We observed that the normal map of the filtered depth is more reasonable compared with that of Kinect depth. We can easily discover the surface features and recognize different objects.

B. Depth Padding

Depth information loss occurs frequently during Kinect capture. In the depth map, the invalid depth region H is evaluated as zero, and the corresponding mask value is designed to be zero.

$$M(i,j) = \begin{cases} 1, & (i,j) \notin H \\ 0, & (i,j) \in H \end{cases}. \tag{12}$$

These irregular invalid depth regions break the continuousness of the depth map and produce sharp variation on depth values

at their boundaries. Considering that most of the compression schemes are block-based, we only need to repair the invalid depth region within a block. Therefore, we introduce block based padding. The filtered depth frame is divided into blocks, and the invalid depth region in the block is padded by its spatially neighboring depth. The sharp variation within the depth block is eliminated, and thus leading to a reduction in the coding bits. The padding value of the pixel in the invalid regions is calculated as follows:

$$\bar{D}_k(u,v) = \frac{\sum_{(i,j)\in Blk} M_k(i,j)\widehat{D}_k(i,j)}{\sum_{(i,j)\in Blk} M_k(i,j)}, \quad (u,v) \in H \tag{13}$$

$\bar{D}_k(u,v)$ is the padded depth value at position (u,v), and $\widehat{D}_k(i,j)$ denotes the filtered depth value at (i,j). For the depth hole located in the smooth surface, it is possible that the padded values are close to the true depth. But for considerable depth holes along the object boundaries, the padded values may be quite different from the truth. It is necessary to transmit the mask to the decoder for correction. Therefore, the original binary mask M_k is generated from the raw depth map and encoded using *JBIG*.

VI. Depth Coding With 2D+T Prediction

In the traditional video coding, reconstructed depth maps are utilized as short-term references that are sensitive to depth noise and may lead to large-scale residuals in the unstable depth sequence. But they can accurately predict the moving object's depth. Based on this fact, we propose 2D+T prediction to exploit the long-term inter frame correlations as compensation for conventional 2D depth prediction. The references for 2D+T prediction are generated using volumetric integration.

A. Reference Generation With Volumetric Integration

A 3D surface can be generated using tremendous range data accumulation and the random noises in the range data can be suppressed during the surface reconstruction. As a typical representation of range data, Kinect-like depth can be utilized to reconstruct the surface of a 3D object. Among the various surface reconstruction techniques, volumetric integration [35] is widely applied for surface reconstruction of range data.

The combination rules of volumetric integration can be described with the following equation

$$D(x) = \frac{\sum w_i(x) d_i(x)}{W(x)} \tag{14}$$

with $W(x) = \sum w_i(x) \cdot d_i(x)$ is the assigned distance of each point x to the i-th range surface along the line of sight of the sensor, and $w_i(x)$ is the weight function depending on the angle between vertex normal and the viewing direction, denoted as θ_i. The continuous implicit function $D(x)$ is presented on a dictate voxel grid, and the isosurface is extracted corresponding to $D(x) = 0$. Fig. 8 illustrates the process of isosurface extraction. The range images captured by two time slots are R_t and R_{t-1}. For R_t, the corresponding signed distance is $d_t(x) = x - R_t$ and the weight is $w_i(x) = \cos\theta_t$. In terms of the combination rule depicted in (14), when $x = (R_{t-1}\cos\theta_{t-1} + R_t \cos\theta_t)/2$, the isosurface is obtained.

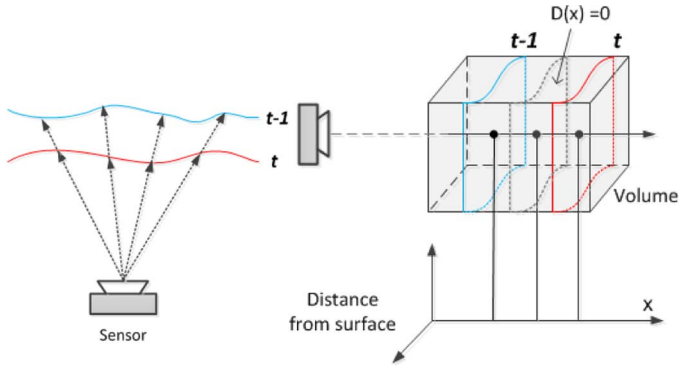


Fig. 8. A simple example of volumetric integration.

Although the integrated surface can provide a stable reference for depth prediction, the volumetric representation of range images requires a large amount of memory and computation. So we propose depth volume to simulate the 3D volume under the assumption that the angle between the viewing direction and the range surface is unchanged within a short period of time. In this way, the range of isosurface can easily be generated by taking the average of related range data. If this derivation extends to multiple range images, the range of the isosurface should be the average range of each range images.

$$x_0 = \frac{1}{n}\sum R_i, \text{ with } \quad D(x_0) = 0 \tag{15}$$

where n is the number of range images. There is a strong temporal correlation among the neighboring depth frames, which offer support to the previous assumption. If the depth data varies too much to satisfy the assumption, the data will not be loaded to depth volume for depth reference generation.

Listing 1 Depth reference generation

1: **for** each pixel in the k-th reconstructed depth D_k^*
2: **if** $|D_k^*(pos) - D_{k-1}^*(pos)| < \alpha \cdot \Theta(D_k^*(pos))$ **then**
3: $Cnt(pos) \leftarrow Cnt(pos) + 1$
4: $DV(pos, Cnt(pos)) \leftarrow D_k^*(pos)$
5: **else**
6: $Cnt(pos) \leftarrow 1$
7: $DV(pos, Cnt(pos)) \leftarrow D_k^*(pos)$
8: **if** $Cnt(pos) \geq Cnt\_threshold(pos)$
9: $Dr_k(pos) \leftarrow Average\ of\ DV(pos)$

The pseudocode Listing 1 illustrates the main steps of reference generation. Depth volume (DV) is a three-dimensional depth buffer, in which each voxel represents the historical depth value at a certain position (pos). First, a comparison is implemented between the current reconstructed depth D_k^* and the previous reconstructed depth D_{k-1}^* to evaluate the activity of each

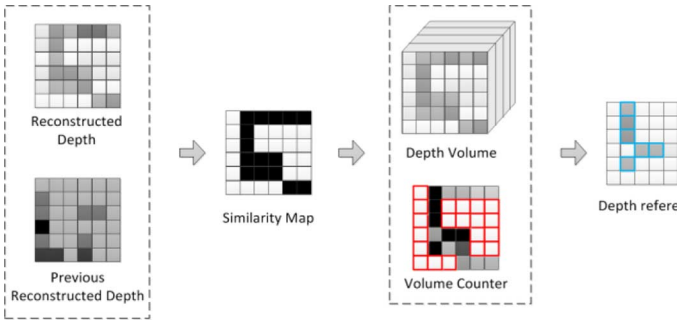


Fig. 9. Generation process of the depth reference.

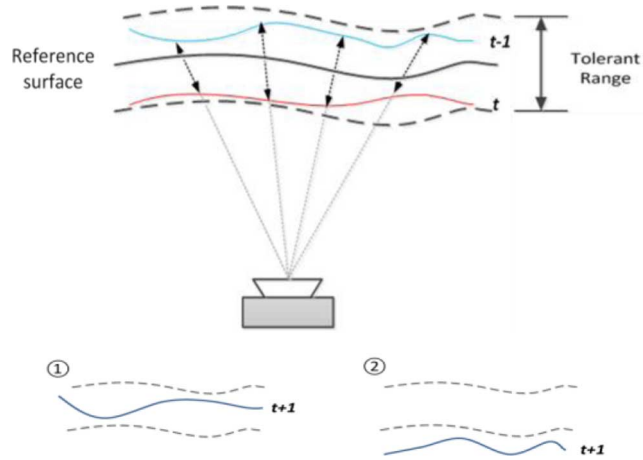


Fig. 10. Error tolerant rule during 2D+T predication.

depth pixel. If the pixel difference is smaller than the product of the constant α and the normalization error bound $\Theta(\cdot)$ mentioned in the Section V, the pixel of the current reconstructed depth will be regarded as an inactive pixel, whose value will be loaded into the depth volume and stored behind the pervious recorded depth value. The corresponding counter $Cnt(pos)$ is increased by one. Otherwise, the pixel is taken as an active pixel, whose value will be recorded as the fresh value located on the front of the array. Meanwhile, the counter at that position is reset to one. If the number of the similar depth values at a position is large enough, the average of the historical depth value is load loaded to k-th depth reference Dr_k.The constant α is proportional to the intensity of light inference.

Fig. 9 shows an example of depth reference generation. After checking the neighboring reconstructed depth difference, a similarity map is produced to denote whether the pixel is compatible with previous depth records. In the similarity map, the inactive pixels are black, while the active pixels are white. The counter of the active pixel is reset to zero (squares with red boundaries), and the counters of the rest positions are increased by one. For these positions (squares with blue boundaries), if the counter is large or equal to the depth of the depth volume, the pixel values at these positions in the depth reference are updated in terms of the depth volume.

B. Video Coding With 2D+T Prediction

According to the causes of the disparity error, both the normalization error and the identification error are distributed in a random manner. These random errors tend to be eliminated during depth accumulation, and the generated depth reference can be regarded as a reliable reference approaching the ground truth. Therefore, if the difference between input depth and the reference is smaller than the disparity error bound $\Theta(\cdot)$, the difference is more likely caused by measurement error rather than a new emerging surface.

Based on the above analysis, we assign a tolerant range (see Fig. 10) to the reliable reference surface with respect to the depth error model deduced in Section III. If the new depth is located in the tolerant region, the surface can be represented by the accumulated historical depth information and will be skipped during traditional 2D coding. If the newly coming surface is out of range, the surface is regarded as a moving surface and will be passed to the next step for traditional 2D prediction. The prediction is implemented block by block, which is equivalent to the motion detection of each subsurface. Therefore, our approach is more flexible than the traditional object-based segmentation methods, and more than one moving object can be distinguished during the prediction process. The coding bitrate and complexity is greatly reduced due to the inactive region removal.

Our prediction results of the depth sequence for "*Roof*" and "*Player*" are shown in Fig. 11. In the depth sequence, a man was playing with a Kinect, and he moved and jumped according to the instructions from the Xbox. The depth reference grew over time because more and more stable depth data was loaded to the depth volume. It was observed that the 2D+T depth reference contained both the player's earlier motion information and static background information. The new emerging depth could be easily recognized referring to the depth reference and recorded as a residue of prediction. In the residue, the static background is automatically removed from the depth except for some unstable boundary blocks. After traditional 2D prediction only a small range residual remain.

VII. Experimental Results

Our compression framework can be integrated with the start-of-the-art image/video coding schemes, and the coding scheme can be adaptively chosen in terms of the system requirements for coding efficiency and complexity. In order to evaluate our depth compression scheme, we have carried out a series of experiments on the depth sequences captured by the Kinect. The objective and subjective comparisons between the results obtained by different coding scheme are implemented mainly based on three aspects: 1) coding efficiency; 2) computational complexity; 3) 3D reconstructed results. Four depth sequences given in Fig. 12 are used for testing, all of which are captured at 30 fps, with a resolution of 640×480. The description of testing sequences is listed in Table I.

A. Coding Efficiency and Computation Complexity

Considering that the pixel's bit-depth supported by HEVC [36] is no larger than ten, we employ the H.264 reference software (JMkta) [37] in our depth compression framework. This software can be directly applied for high dynamic (up to

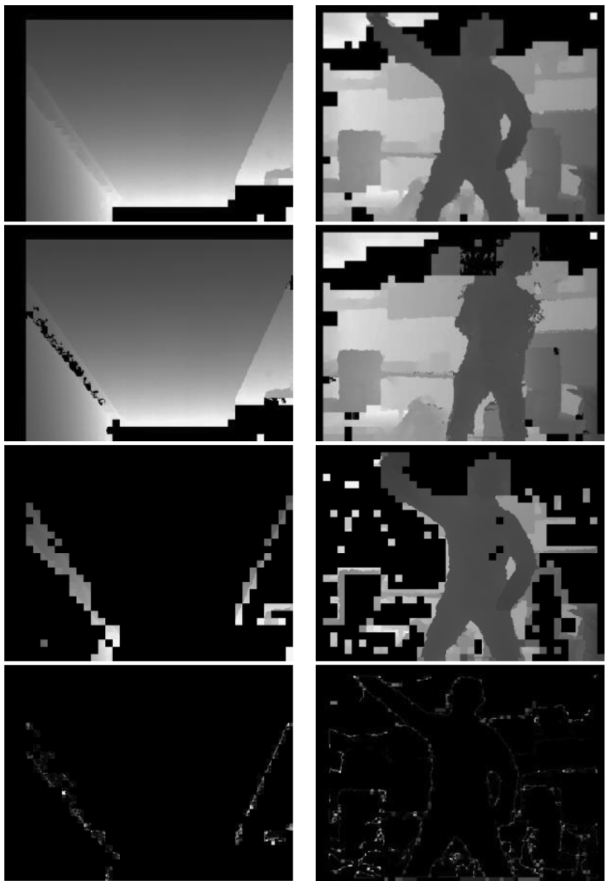


Fig. 11. The intermediate prediction results of depth sequence "*Roof*" (left) and "*Player*" (right). The depths from the top to bottom: padded depth, 2D+T depth reference, residue of 2D+T prediction, residual after traditional inter predication.

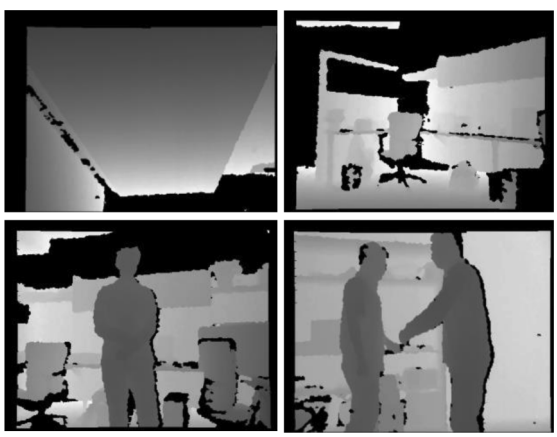


Fig. 12. The first frame of each test depth sequence used in the paper, upleft: "*Roof*", upright: "*Cubicle*", bottomleft: "*Player*", bottomright: "*MovCam&People*".

TABLE I
DESCRIPTION OF TEST SEQUENCES

Sequence	**Discription**
Roof	Fixed Kinect captures simple scene
Cubicle	Fixed Kinect captures complex scene
Player	Fixed Kinect captures moving objects
MovCam&People	Moving Kinect captures the multiple moving objects

14bits) depth coding. In the experiments, the counter thresholds in depth volume are uniformly set to five. Actually, the counter threshold of each position can be defined respectively in accordance to the sensitivity requirement of the depth content. For simplicity, the constant α is set to one and does not change with the coding parameters, since we do not measure the variation of the light influence in this work.

In an attempt to evaluate the function of reformation in the proposed coding framework, the coding performance of the depth encoder with/without reformation is compared, where 2D+T prediction is disabled. The plots in Fig. 13 shows the rate-distortion performance of different sequences compressed at quantization parameters (QP) equal to $[0, -5, -10, -15]$. Notice that the QP in *JMkta* can be a negative value, when pixel value has a dynamic range of more than 256. The peak signal-to-noise ratio (PSNR) is utilized to measure the quality of the reconstructed depth.

$$PSNR = 10 \cdot \log_{10}\left(\frac{Peak^2}{MSE}\right) \tag{16}$$

where *MSE* denotes the mean square error of between the reference depth and the reconstructed depth, and *Peak* is the peak value equal to $(2^{12} - 1)$ due to the large dynamic data range. The PSNR of the codec with reformation (*kta_reform*) is calculated by referring the reformed depth instead of original depth.

We can observe that the depth coding performance is dramatically improved after depth reformation for all the depth sequences in Fig. 13. At the same distortion level, the bitrate is reduced to 25%–50% of the original bitrate. The remarkable coding gain mainly stems from two factors: one is that the DNBL filtering recovers the broken spatial correlation in the depth map; the other is that the efficient padding and mask representation greatly reduce the bit cost of the non-zero high frequency transform coefficients caused by irregular depth holes. The reformed depth sequence is friendlier for traditional video coding than the original one.

When combining reformation and 2D+T prediction, the rate-distortion of our approach is denoted as "proposed." The quality of the active region and the bitrate of the different coding schemes are compared in Table II. The active regions are composed of the blocks remained after 2D+T prediction, and the corresponding PSNR value is denoted as "$PSNR_{active}$". "*Bitrate*" is the whole depth frame bit cost in the coding scheme. As seen in the table, our codec performs the best on both depth quality and bitrate. It saves 55%–85% of bit cost for the Kinect depth sequence. By skipping inactive regions during depth encoding, the coding bit cost of the whole frame is reduced to that of the active depth region. The reduction ratio on bitrate and complexity is determined by the depth contents. The greater the size of inactive blocks, the larger the bitrate and complexity reduces. To verify our analysis, the block ratios of the inactive region are listed in Table III. The inactive blocks ratio of the sequence "*Roof*" is 93.59%, the largest among the four sequences, and over 85% bitrate reduction is achieved at the high bitrate. In contrast, the inactive block ratio of "*MovCam&People*" is 38.11%, the smallest among the four sequences, since the capture content keeps changing with

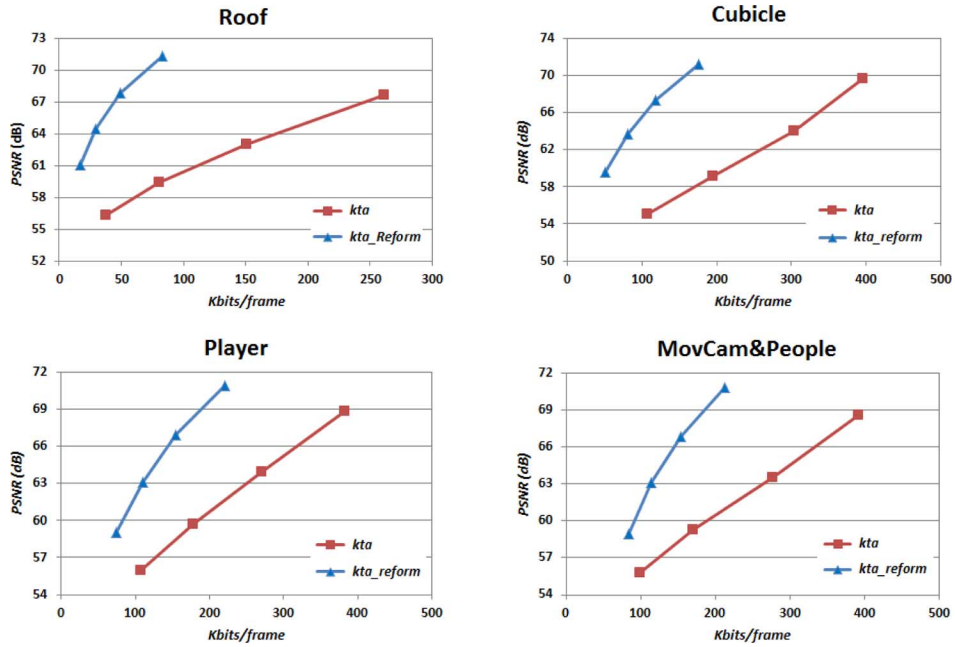


Fig. 13. Comparisons on rate-distortion performance of depth encoder with/without depth reformation.

TABLE II
RATE-DISTORTION COMPARISON OF SCHEMES BASED ON VIDEO CODECS

Codec setting		*"Roof"*		*"Cubicle"*		*"Player"*		*"MovCam&People"*	
		PSNR$_{active}$ *(dB)*	*Bitrate (Mbps)*	*PSNR*$_{active}$ *(dB)*	*Bitrate (Mbps)*	*PSNR*$_{active}$ *(dB)*	*Bitrate (Mbps)*	*PSNR*$_{active}$ *(dB)*	*Bitrate (Mbps)*
QP = -15	*kta*	66.47	7.64	68.25	11.61	67.35	11.21	68.61	11.47
	kta_reform	69.10	2.09	68.91	5.14	69.49	6.45	**70.80**	6.20
	proposed	**69.23**	**1.16**	**69.01**	**2.24**	**69.53**	**4.59**	70.24	**4.94**
QP = -10	*kta*	61.66	4.42	62.65	8.89	62.80	7.91	63.49	8.11
	kta_reform	64.94	1.98	64.63	3.46	65.34	4.53	**66.85**	4.49
	proposed	**65.08**	**0.99**	**64.76**	**1.89**	**65.40**	**3.45**	66.30	**3.57**
QP = -5	*kta*	57.55	2.35	57.80	5.71	58.96	5.20	59.26	5.00
	kta_reform	60.97	1.88	60.54	2.35	61.32	3.22	**63.09**	3.35
	proposed	**61.11**	**0.85**	**60.68**	**1.48**	**61.40**	**2.63**	62.53	**2.64**
QP = 0	*kta*	53.65	1.01	53.43	3.12	54.81	3.16	55.80	2.90
	kta_reform	56.74	1.75	56.05	1.48	56.99	2.17	**58.95**	2.44
	proposed	**56.78**	**0.74**	**56.22**	**1.20**	**57.07**	**1.96**	58.41	**1.88**

TABLE III
BLOCK RATIO OF INACTIVE REGION DURING 2D + T PREDICTION

Sequence	*"Roof"*	*"Cubicle"*	*"Player"*	*"MovCam&People"*
Block Ratio	93.59%	88.98%	61.44%	38.11%

Kinect's movement. As a result, the bitrate saving is less than 60% and the encoding time saving is around 20%.

From Table II, we can observe that the quality of the reconstructed active regions is improved in comparison to the compression without 2D+T prediction for the first three sequences. It is caused by depth quality improvement in the inactive regions, which provide reliable reference for the intra and inter prediction of active regions. For the sequence "*MovCam&People*", the Kinect moves during the capture, it is difficult to generate reliable reference surface by volume integration. The quality of the active region degrades a little in comparison with that of *kta*. In general, the details of the active region are preserved as the *kta* does, and the noise of the inactive region is suppressed with the inherent depth features being preserved.

Besides the coding efficiency, our scheme also shows the advantage for the coding complexity. The computation complexity results of each scheme are tabulated in Table IV. With a priori knowledge gained by 2D+T prediction, there is no need to search for the optimal mode to remove inactive blocks and then encode the remaining. As a result, a noticeable reduction in complexity is achieved and the encoding time is reduced by 20%–65%. Similar to the bitrate, the complexity reduction ratio is proportional to the ratio of inactive regions.

We compare the proposed scheme with near-lossless Kinect-like depth compression scheme developed by Mehro-tra *et al.* [29]. In Mehrotra's scheme, the scaling reciprocal of each depth

TABLE IV
COMPLEXITY COMPARISON OF SCHEMES BASED ON VIDEO CODECS

Codec	*"Roof"*		*"Cubicle"*		*"Player"*		*"MovCam&People"*	
	Enc. (ms)	*ΔEnctime*	*Enc. (ms)*	*ΔEnctime*	*Enc. (ms)*	*ΔEnctime*	*Enc. (ms)*	*ΔEnctime*
kta	9592	——	10113	——	10007	——	11209	——
kta_reform	9724	-1.38%	9870	2.40%	9902	1.05%	11003	1.84%
proposed	**3349**	**65.09%**	**3751**	**62.91%**	**6071**	**39.33%**	**8954**	**20.12%**

TABLE V
BITRATE COMPARISON WITH ONE KINECT DEPTH COMPRESSION SCHEME

Codec setting		*"Roof"*		*"Cubicle"*		*"Player"*		*"MovCam&People"*	
		Bitrate (Mbps)	*Compression Ratio*	*Bitrate (Mbps)*	*Compression Ratio*	*Bitrate (Mbps)*	*Compression Ratio*	*Bitrate (Mbps)*	*Compression Ratio*
Factor = 1.5	***Mehrotra's***	14.58	9.64	14.17	9.92	19.37	7.26	19.06	7.38
	proposed	**1.14**	**123.84**	**2.19**	**64.09**	**4.32**	**32.58**	**4.26**	**33.03**
Factor = 0.8	***Mehrotra's***	11.58	12.15	11.76	11.95	16.36	8.59	15.87	8.86
	proposed	**0.81**	**173.31**	**1.32**	**106.77**	**2.24**	**62.81**	**1.81**	**77.86**

value is encoded pixel by pixel with low complexity, and the near-lossless compression can be achieved if the scaling factor is large enough. Conversely, the smaller scaling factor will introduce larger coding distortion and cost less bitrate correspondingly. The scaling factor of the reference scheme is set as 1.5 and 0.8, respectively. For each factor, we compare the bitrate cost of the compressed depth sequences with the similar MSE value, which is calculated in the active regions generated by our coding scheme. The comparison results are shown in the Table V. Our coding scheme can save 73%–93% bitrate comparing with the reference one. The significant coding gain is achieved, because the temporal redundancy is greatly reduced by the 2D+T prediction.

B. 3D Reconstruction Comparisons

In this subsection, we apply normal maps and rendering mesh for 3D reconstruction results comparison. The normal maps of the original depth, reformed depth and the reconstructed depth of *kta* and our scheme are shown in Fig. 14. After reformation, the step-shaped depth fluctuation is suppressed in the smooth region, e.g., the back of the chair and the wall, as well as player's arms and legs. Meanwhile, the inherent depth discontinuity is preserved, such as the small objects on the table in the "*Cubicle*", the clothing wrinkles and the player's cheek in "*Player*." The compression QP is set to -10 to generate high quality reconstructed depth. In the reconstructed results of *kta*, the original depth noises are retained and new noisy pixels emerge around the boundaries adjacent with depth holes, such as the region behind the player's left shoulder and the region above the player's head in "*Player*", as well as the wall boundary regions on the upper-left part of "*Cubicle*". The results of our scheme show that the object boundaries are clearly reconstructed without emerging noisy pixel. For the inactive regions where the blocks are skipped in the 2D+T prediction, like the wall and the chair back, the surface looks even smoother than that of the reformed depth due to the temporal filtering during the depth reference generation in depth volume. In contrast, the details of the active regions are maintained by traditional video compression, like the wrinkles on the player's clothes in "*Player*" and the surfaces of the small items on the table in "Cubicle".

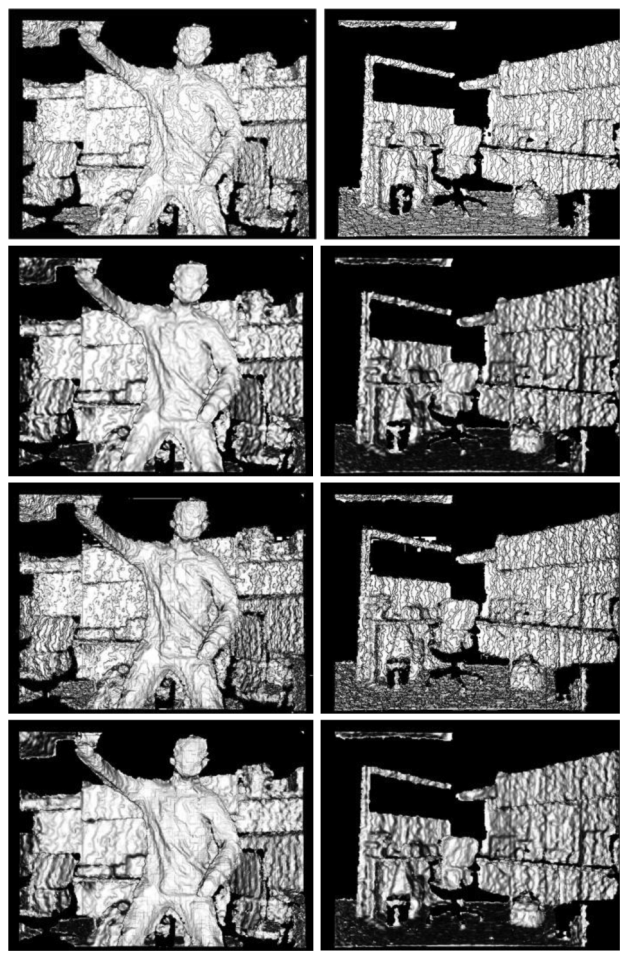


Fig. 14. Normal maps of the final result of depth coding for the 63-th frame of "*Player*"(left) and the 6-th frame of "*Cubicle*"(right) at $\mathrm{QP} = -10$. The images from the top to bottom: normal map of the original depth, normal map of the reformed depth, normal map of the reconstructed by *kta*, and the reconstructed depth by our approach.

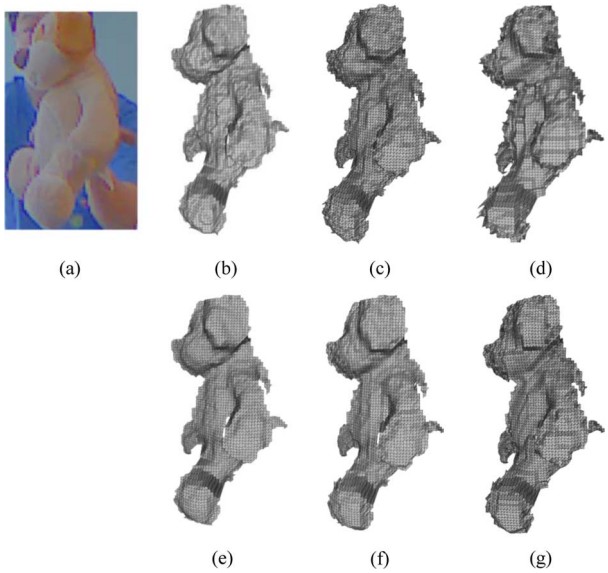


Fig. 15. Comparison on the 3D reconstruction mesh. (a) the clipped texture (b) mesh rendered from the raw depth (c) mesh of the compressed depth by *kta* $(\mathrm{QP} = -15)$ (d) mesh of the compressed depth by *kta* $(\mathrm{QP} = -10)$ (e) mesh rendered from the reformed depth (f) mesh of the compressed depth by our approach $(\mathrm{QP} = -15)$ (g) mesh of the compressed depth by our approach $(\mathrm{QP} = -10)$.

Another comparison results on 3D rendering view is give in Fig. 15. The left RGB image is the texture of a toy that is clipped from a Kinect texture frame, which has been calibrated with the depth by windows Kinect SDK. The top-left image is the 3D reconstruction result of the original depth, and the bottom is the result of the reformed depth. The right images show the render results of the reconstructed depth after compression by *kta* or our scheme. The 3D render results of our algorithm are much better than that of *kta*, especially for compression with large QP. The blocking artifacts caused by spectral information loss are notable in the results of *kta*, and the mesh boundary become jagged after compression, see Fig. 15(c) and (d). In contrast, our results preserve the original depth features with fewer artifacts in Fig. 15(f) and (g).

VIII. Conclusions and Future Work

Based on the special characteristics of Kinect-like depth data, we propose a novel depth compression framework in which the depth is reformed first to suppress the depth spatial noises and then predicted using a long-term reference to detect the uniqueness of depth contents for better bit allocation. In the reformation technologies, the spatial DNBL filtering fully utilizes the depth measurement error model to distinguish the inherent depth edges from the normalization error, and the depth padding rebuild the depth continuity inner depth block for efficient block-based coding. More importantly, we distinguish the new emerging depth contents by 2D+T prediction, and accordingly assign more coding resources to them. Our approach can save more than 55% bitrate with a remarkable reduction in the coding complexity at the same time.

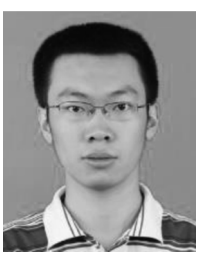
As we indicated earlier, the primary goal of this paper is to develop a depth coding framework to achieve high efficiency depth coding with low-complexity and better serve for the sequential depth-related applications. Therefore, there is a good deal research that is still on-going or that we intend to conduct in the near future. First of all, considering the physical meaning of the depth, we intend to define new metrics for depth distortion measurement in accordance with distinct requirements of sequential applications, with which the designed coding framework can adapt to the applications much better. Secondly, the parameter settings of the proposed coding scheme should adjust to time-variant light influence. Finally, we intend to reduce the computing complexity to satisfy the requirements of real-time depth data transmission.

References

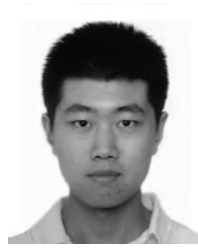
[1] A. M. Tekalp, *Digital Video Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1995.
[2] A. Bovik, *Handbook of Image and Video Processing*. New York, NY, USA: Elsevier Academic Press, 2000.
[3] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, pp. 30–44, 1991.
[4] Generic Coding of Moving Pictures and Associated Audio (MPEG-2), Part 2: Video, ISO, 1995, ISO/IEC JTC 1/SC 29/WG 11/13 818-2.
[5] Advanced Video Coding (AVC), ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 Part 10), JVT, 2004.
[6] S. B. Gokturk, H. Yalcin, and C. Bamji, "A time-of-flight depth sensor-system description, issues and solutions," in *Proc. CVPR*, 2004.
[7] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: Theory and experiment," in *Proc. Pobotics and Automation*, 1991, pp. 1088–1095.
[8] G. F. Marshall, *Handbook of Optical and Laser Scanning*. New York, NY, USA: Marcel Dekker, 2004.
[9] G. Frankowski, M. Chen, and T. Huth, "Real-time 3D shape measurement with digital stripe projection by texas instruments micromirror devices DMDTM," in *Proc. SPIE*, 2000, vol. 3958, pp. 90–106.
[10] Microsoft Kinect. [Online]. Available: http://www.xbox.com/kinect.
[11] T. Weise, S. Bouaziz, H. Li, and M. Pauly, "Realtime performance-based facial animation," in *Proc. ACM SIGGRAPH*, 2011.
[12] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proc. CVPR, 2011*, Nov. 2011, pp. 1297–1304.
[13] S. Gumhold, Z. Karni, M. Isenburg, and H. Seidel, "Predictive point-cloud compression," in *Proc. SIGGRAPH '05, ACM SIGGRAPH 2005 sketch*, Aug. 2005.
[14] R. Schnabel and R. Klein, "Octree-based point-cloud compression," in *Proc. Symp. Point-Based Graphics 2006*, Jul. 2006, pp. 147–156, Eurographics press.
[15] J. Kammerl, Development and Evaluation of Point Cloud Compression for the Point Cloud Library. [Online]. Available: http://www.willow-garage.com/blog/2011/06/01/compressing-point-clouds-point-cloud-library-pcl.
[16] A. T. Murat and O. Jorn, Face and 2-D Mesh Animation in MPEG-4.
[17] K. Mamou, T. Zaharia, and F. Preteux, FAMC: The MPEG-4 Standard for Animated Mesh Compression.
[18] C. Touma and C. Gotsman, "Triangle mesh compression," in *Proc. Graphics Interface. 1998*, , 1998, pp. 26–34.
[19] Z. Karni and C. Gotsman, "Spectral compression of mesh geometry," in *Proc. 27th Annu. Conf. Computer Graphics and Interactive Techniques*, Jul. 2000, pp. 279–286.
[20] S. Grewatsch and E. Muller, "Fast mesh-based coding of depth map sequences for efficient 3D video reproduction using openGL," *Visualiz., Imag., Image Process.*, pp. 66–480, 2005.
[21] B. B. Chai, S. Sethuraman, H. S. Sawhney, and P. Hatrack, "Depth map compression for real-time view-based rendering," *Pattern Recognit. Lett., Elsevier*, pp. 755–766, May 2004.
[22] C. Fehn, "Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D TV," in *Proc. SPIE, Conf. Stereoscopic Displays and Virtual Reality Systems XI*, 2004, vol. 5291, pp. 93–104.
[23] S. Grewatsch and E. Muller, "Evaluation of motion compensation and coding strategies for compression of depth map sequences," in *Proc. 49th SPIE's Annual Meeting*, 2004.

[24] Y. Morvan, D. Farin, and P. H. N. de With, "Depth-image compression based on an R-D optimized quadtree decomposition for the transmission of multiview images," in *Proc. IEEE ICIP*, Sep. 2007.
[25] S. Milani and G. Calvagno, "A depth image coder based on progressive silhouettes," *IEEE Signal Process. Lett.*, vol. 17, no. 8, pp. 711–714, 2010.
[26] G. Shen, W. Kim, S. Narang, A. Orterga, J. Lee, and H. Wey, "Edge-adaptive transform for efficient depth map coding," in *Proc. Picture Coding Symp. (PCS)*, Nagoya, Japan, Dec. 2010.
[27] M. Maitre and M. Do, "Depth and depth-color coding using shape-adaptive wavelets," *J. Vis. Commun. Imag. R.*, vol. 21, no. 5–6, pp. 513–522, 2010.
[28] S. Liu, P. Lai, D. Tian, and C. W. Chen, "New depth coding techniques with utilization of corresponding video," *IEEE Trans. Broadcasting*, vol. 57, no. 2, pp. 551–561, 2011.
[29] S. Mehrotra, Z.-Y. Zhang, Q. Cai, C. Zhang, and P. Chou, "Low-Complexity, near-lossless coding of depth maps from Kinect-like depth cameras," in *Proc. MMSP, IEEE*, Oct. 2011.
[30] K. Khoshelham, "Accuracy analysis of Kinect depth data," in *Proc. ISPRS WORKSHOP 2011*, Calgary, AB, Canada.
[31] [Online]. Available: http://research.microsoft.com/en-us/um/redmond/projects /Kinectsdk/ (SDK).
[32] JBIG. [Online]. Available: http://ww.jpeg.org/jbig/index.html.
[33] C. Tomaso and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. ICCV*, 1998, pp. 839–846.
[34] J. Fu, S. Wang, Y. Lu, S. Li, and W. Zeng, "Kinect-like depth denoising," in *Proc. ISCAS*, 2012, pp. 512–515.
[35] B. Curless and M. Kevoy, "A volumetric method for building complex models form range images," in *Proc. ACM SIGGRAPH 96*, 1996, pp. 303–312.
[36] HEVC [Online]. Available: http://hevc.kw.bbc.co.uk/trac/browser/branches.
[37] Jm14.2kta1.0. [Online]. Available: http://iphome.hhi.de/suehring/tml/.

Jingjing Fu received the B.Sc. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 2005 and the Ph.D. degree in electronic and computer engineering from Hong Kong University of Science and Technology, Kowloon, Hong Kong, in 2010. She is currently an associate researcher at the Media Computing Group, Microsoft Research Asia. Her areas of interest include image/video coding, 3-D image/video processing and multimedia system.



Dan Miao received the B.E. degree from University of Science and Technology of China (USTC) in 2009. He is currently working toward the Ph.D. degree in Department of Electrical Engineering, USTC. He has been a research intern in Microsoft Research Asia since 2010. Currently, his research interests focus on image and video coding, 3D video compression, rendering and transmission.



Weiren Yu received the B.E. degree from the School of Advanced Engineering at Beihang University (formerly known as Beijing University of Aeronautics and Astronautics) in 2011. He is currently a Ph.D. candidate in the Department of Computer Science, Beihang University since 2011. His research interests include video streaming and transmission, cloud mobile gaming, user experience study and data mining.

Shiqi Wang received the B.S. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2008. He is currently working toward the Ph.D. degree in computer science at Peking University, Beijing, China. He was a Vising Student with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada from 2010 to 2011. From April 2011 to August 2011, he was with Microsoft Research Asia, Beijing, as an Intern. His current research interests include video compression, image and video quality assessment and multi-view video coding.

Yan Lu received his Ph.D. degree in computer science from Harbin Institute of Technology, China, in 2003. He joined Microsoft Research Asia in 2004, where he is now a lead researcher. From 2001 to 2004, he was affiliated with the Joint R&D Lab (JDL) for advanced computing and communication, Chinese Academy of Sciences, China, working on the AVS standard. From 1999 to 2000, he was also with the City University of Hong Kong as a research assistant. His research interests include image and video coding, multimedia system and mobile computing. He is the co-inventor of 9 granted US patents and over 20 filed patent applications. In 2007, he received the IS&T/SPIE VCIP Best Paper Award.

Shipeng Li received his B.S. and M.S. in Electrical Engineering from the University of Science and Technology of China, Hefei, China in 1988 and 1991, respectively. He received his Ph.D. in Electrical Engineering from Lehigh University, Bethlehem, PA, USA in 1996. He was a faculty member in Department of Electronic Engineering and Information Science at University of Science and Technology of China in 1991–1992.

Dr. Shipeng Li joined Microsoft Research Asia (MSRA) in May 1999. He is now a Principal Researcher and Research Manager of the Media Computing group. He also serves as the Research Area Manager coordinating the multimedia research activities at MSRA. His research interests include multimedia processing, analysis, coding, streaming, networking and communications; digital right management; advertisement; user intent mining; eHealth; etc. From October 1996 to May 1999, Dr. Li was with Multimedia Technology Laboratory at Sarnoff Corporation as a Member of Technical Staff. Dr. Li has been actively involved in research and development in broad multimedia areas. He has authored and co-authored 6 books/book chapters and 250+ referred journal and conference papers. He holds 120+ granted US patents.