

# Compression of 3D Objects with Multistage Color-Depth Panoramic Maps

Chang-Ming Tsai <sup>1,2</sup>, Wen-Yan Chang <sup>1</sup>, Chu-Song Chen <sup>1</sup>, Gregory Y. Tang <sup>2</sup>

<sup>1</sup> Institute of Information Science, Academia Sinica, Nankang 115, Taipei, Taiwan.

<sup>2</sup> Department of Computer Science and Information Engineering, National Taiwan University.

*Email:* song@iis.sinica.edu.tw

## Abstract

We propose a new representation method, the multistage color-depth panoramic map (or panomap), for compressing 3D graphic objects. The idea of the proposed method is to transform a 3D graphic object, including both the shape and color information, into a single image. With a pseudo cylinder (or sphere), the color/depth information of a 3D graphic object can be recorded completely via a main panomap combined with a set of residual panomaps. Then, an index sequence is constructed to index non-redundant blocks in the residual panomaps. A rectangular image, the compact multistage panomap (CMP), is formed based on the panomaps and index sequence. Existing image compression techniques can be applied for compressing the CMP structure, which can achieve a highly efficient representation due to its regularity.

**Keywords:** data compression; 3D data compression; cylindrical image; spherical image; multistage color-depth panoramic map (panomap); main panomap; residual panomap; compact multistage panomap (CMP).

## 1. Introduction

As the growing needs for transmitting 3D graphic models over internets, compression of them becomes more and more important due to the large amount of data. In particular, for some interactive applications such as virtual museum or virtual mall, compact representation of 3D objects is a central issue for overcoming the bottleneck of efficient transmission over internets.

Generally, there are two types of representations of 3D objects: polygonal type and voxel type. In the past, most of the 3D compression methods focus on how to simplify the mesh of a 3D object by reducing the number of polygons for the polygonal type of data [2][4][7]. In principle, the simplification is usually based on the generation of a coarse mesh by collapsing the edges or removing nonessential vertices of objects. To compress voxel type of data, various transformations, e.g. Fourier, Laplacian and Wavelet [6][9], can be used. Even though the size of a voxel-based representation can be reduced, it is still a heavy load for applications associated with internet transmissions.

In fact, accompany with a rapid progress in 3D imaging technologies, tools for acquisition of 3D objects such as laser scanners [1][3], contour-based 3D modeler [5], and feature-based 3D modeler [10], have been widely applied for computer graphics, virtual reality, or augmented reality. In these cases, we care not only shapes but also textures of the surfaces of 3D objects. Since the raw data captured with a 3D scanner is in essence a set of points, it is easy to be recorded via voxel-type representation. In particular, only the points on the surface of the object have the associated color (or texture) values; that is, only the surface points are required to be considered. However, most existing voxel-based compression methods were developed to compress not only the surface but the interior points, and thus are not efficient for the data where only surface information has to be concerned.

Polygon-mesh structure can also be constructed for the scanned 3D points. However, since a polygon mesh structure contains irregular connections among vertices, it is difficult to be compressed efficiently and effectively. In addition, associations between the vertices and the texture maps are also irregular, which increases the difficulty to find a compact representation. Although neighbor vertices (i.e., vertices connected by an edge) exhibit redundancies either in their positions or colors, the redundancies would diminish when the mesh is simplified.

In this paper, we propose a new representation method for 3D graphic objects, the

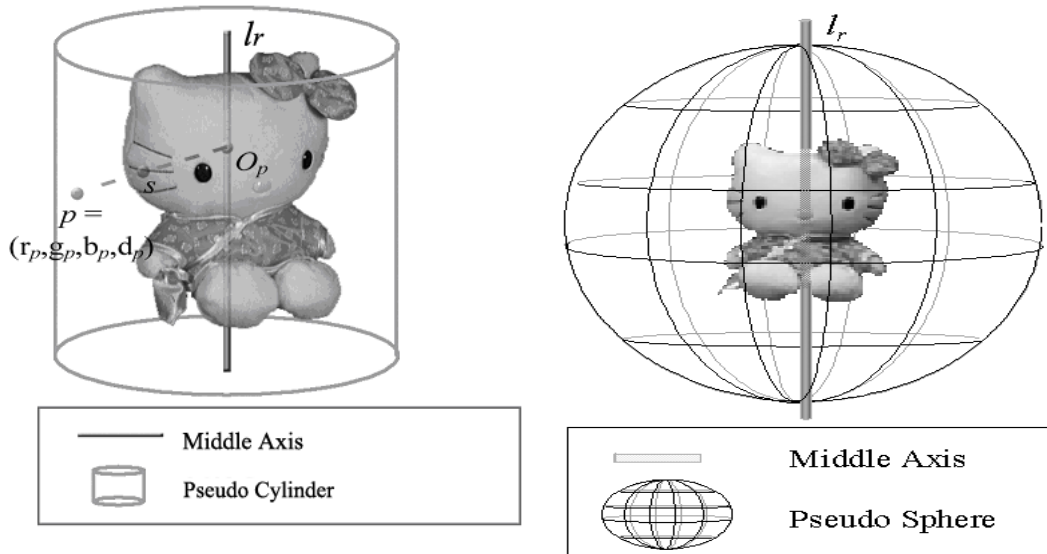
*multistage color-depth panoramic map* (or in brief, *panomap*). It represents both the shape and color information for a 3D graphic object in a regular way. Moreover, the association between the 3D points and the colors are recorded inherently when this regular representation is used, and thus the point-color association information is not required to be further stored. Standard compression techniques (such as JPEG, JPEG2000, or PNG) can be easily applied for compressing the panomap structure, which can achieve a highly-efficient representation due to the regularity of the proposed format. In the following, Section 2 will introduce the multistage panomap, and some conclusions will be given in Section 3.

## 2. Multistage Color-Depth Panomap

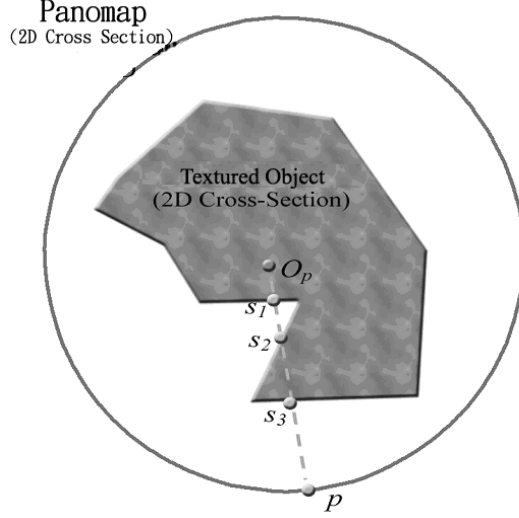
### 2.1 Single-Stage Color-Depth Panomap

Given a 3D textured object,  $V$ , we can define a pseudo cylinder (or sphere) to cover it as illustrated in Fig. 1. A cylindrical (or spherical) image can be regarded as being recoded in the cylinder (or spherical) surface that surrounds  $V$ . The cylindrical (or spherical) image is termed as the *color-depth panomap* that contains both the color and depth information. Without loss of generality, we focus on the cylindrical representation in the following, while our method can be applied for both types of representations.

In essence, each of the pixels in the cylindrical image is a 4-tuple,  $(r, g, b, d)$ , and is assigned in the following way. Given a pixel  $p=(r_p, g_p, b_p, d_p)$  in the cylindrical image,



**Figure 1.** Generation of the color-depth panomap of an object. The panomap of an object is essentially a cylindrical or spherical image that surrounds the textured object.



**Figure 2.** Illustration of the concept of panomap. The pixel  $p$ , which is on the cylindrical image, has multiple corresponding intersection points. In this case,  $s_1$ ,  $s_2$ , and  $s_3$  are these intersection points. For the single-stage (or main) panomap, the surface point which is nearest to  $p$  (in this case,  $s_3$ ) is chosen. For the first-stage residual panomap, the point which is nearest to  $s_3$  but not equal to  $s_3$  (in this case,  $s_2$ ) is chosen.

the line  $\overline{pO_p}$  is constructed, where  $O_p$  lies on the middle axis,  $l_r$ , of the cylinder, and  $\overline{pO_p}$  is perpendicular to  $l_r$ . Let the intersection point of **Surface(V)** and  $\overline{pO_p}$  be  $s$ . If there are many points,  $s_1, s_2, \dots, s_n$ , lying on the intersection of **Surface(V)** and  $\overline{pO_p}$ , as shown in Fig. 2, the point nearest to  $p$  is chosen to be  $s$ . Then,  $r_p, g_p, b_p$  are assigned as the three components of the color of  $s$ , and  $d_p$  is set to be the distance between  $p$  and  $s$ ,  $\|\overline{ps}\|$ . In particular, the cylinder selected for this representation has to satisfy that the object is fully contained in its interior, and thus  $d_p$  is always larger than zero. In our work, the single-stage panomap is also referred to as the *main panomap* of an object.

**Property 1:** If an object is convex, then it can be completely represented via the main panomap.

Although most part of the surface of a 3D object can be recorded, the above representation suffers from the case that some points are missed when the shape is concave along any of its cross sections, as shown in Fig. 2. Consequently, it can only be used to represent a limited class of textured objects that each of which contains a single layer of surface. To improve the representation ability, the multistage panomap is proposed as described in the next section.

## 2.2 Multistage Panomap

**Definition 1 [first-stage residual panomap]:** Given a textured object  $V$ , its *first-stage residual panomap* is also a cylindrical image in which each pixel is a 4-tuple,  $(r, g, b, d)$ . Let the line  $\overline{pO_p}$  and the intersection points,  $s_1, s_2, \dots, s_n$  be determined as before, and denote  $\underline{s}$  to be the nearest point to  $p$  among  $s_1, s_2, \dots, s_n$  (i.e.,  $\underline{s}$  is the point recorded in the main panomap described above). If  $n=1$  (i.e., there is only a single intersection point), the 4-tuple associated with  $p$ ,  $(r_p, g_p, b_p, d_p)$ , is set to be  $(0,0,0,0)$ . Otherwise, consider the point  $s_i$  that is the nearest to  $\underline{s}$  but not equal to  $\underline{s}$ , among  $s_1, s_2, \dots, s_n$ . Then,  $r_p, g_p$ , and  $b_p$  are set to be the three primitive colors of  $s_i$ , and  $d_p$  is set to be  $\|\overline{ps_i}\|$ , respectively.

An illustration of the first-stage panomap is also shown in Fig. 2. It is obvious that more complex surfaces (here, surfaces with two layers) can be represented by combining the main and the first-stage residual panomaps.

Similarly, the second-stage residual panomap of an object can be constructed as well by setting the pixels associated with  $n=1$  or  $2$  to be  $(0,0,0,0)$ , and keeping on recording the next remaining points. Repeat the above procedures, a textured object can be represented with the main panomap accompanied with  $K$  residual panomaps ( $K \geq 0$ ), until that every pixel of the  $(K+1)^{th}$ -stage residual panomap is  $(0,0,0,0)$ . The union of the main and all the  $K$  residual maps is called the *multistage panomap* of an object. A textured 3D object, whether concave or convex, can thus be completely recorded.

**Property 2:** Associated with each 3D object, there is a unique multistage panomap that completely represents this object (if a cylinder is given).

The depth and color of an object can be reconstructed from its multistage panomap layer by layer. The first layer of the object surface can be recovered from the main panomap; then, other  $k$  layers can be obtained in turn by subsequently resolving the  $k$  residual panomaps. Notice that in each residual panomap, pixels with the value  $(0,0,0,0)$  are not required to be used for the computation of the object surface, which leads to a compact representation for multistage panomaps as introduced in the following.

## 2.3 Compact Representation for Multistage Panomaps

In practice, the  $(0,0,0,0)$  pixels of a residual panomap can be ignored during the surface reconstruction process. Moreover, when  $k$  is getting larger, more dummy pixels with value  $(0,0,0,0)$  will be produced in the  $k^{th}$ -stage residual panomap. By taking advantage of these properties, a compact representation of the residual maps can be achieved by using an index sequence as introduced below.

**Definition 2 [index table of the first-stage residual map]:** Consider an object  $V$  and its first-stage residual panomap  $R_v^{(1)}$ . Suppose the width and height of  $R_v^{(1)}$  are  $w$  and  $h$ , respectively. Let us divide  $R_v^{(1)}$  as a set of non-overlapping  $8 \times 8$  blocks (the choice of  $8 \times 8$  is because that it coincides with many image compression standards, and both  $w$  and  $h$  are assumed to be multiples of 8), and thus  $(w/8) \times (h/8)$  blocks are generated, which can be indexed by  $(i,j)$ , where  $0 \leq i \leq (w/8)-1$  and  $0 \leq j \leq (h/8)-1$ . Then, these blocks are scanned from a row-major order. If all the pixels contained in the  $(i,j)$ th block are  $(0,0,0,0)$ , then  $Table^{(1)}(i \times (h/8) + j) = 0$ ; otherwise,  $Table^{(1)}(i \times (h/8) + j) = 1$ . The Boolean function,  $Table^{(1)}(\bullet)$ , is then referred to as the *index table* of the first-stage residual panomap.

From the above definition,  $Table^{(1)}$  contains  $(w/8) \times (h/8)$  entries. One possible way is to define the  $k^{th}$ -stage residual panomap similarly for every  $k$ . However, such a straightforward extension is waste representation for  $Table^{(k)}(\bullet)$  because the following property:

**Property 3:** If a pixel is  $(0,0,0,0)$  in the  $k^{th}$ -stage residual panomap, then it is also  $(0,0,0,0)$  in the  $(k+1)^{th}$ -stage residual panomap.

From the above property, the definition of  $Table^{(k)}$  for  $k > 1$  is given as follows:

**Definition 3 [index tables of the other-stage residual panomaps]:** Let  $R_v^{(k)}$  be the  $k^{th}$ -stage residual panomap, which is divided into a set of non-overlapping  $8 \times 8$  blocks, and indexed by  $(i,j)$  where  $0 \leq i \leq (w/8)-1$  and  $0 \leq j \leq (h/8)-1$ . Assume that there are totally  $n_k$   $8 \times 8$  blocks whose pixels are not all  $(0,0,0,0)$ , in  $R_v^{(k)}$ . Then, these blocks are called the *non-redundant* blocks, and are scanned from a row-major order. Denote  $(I^{(k)}(n), J^{(k)}(n))$  to be the index of the  $n^{th}$  non-redundant block scanned ( $0 \leq n \leq n_k-1$ ). Then,  $Table^{(k)}$  is a function with  $n_{k-1}$  entries, and is recursively defined as

$$Table^{(k)}(n) = \begin{cases} 1, & \text{if the } (I^{(k-1)}(n), J^{(k-1)}(n))\text{th block in } R_v^{(k)} \text{ is non-redundant;} \\ 0, & \text{otherwise;} \end{cases}$$

for all  $0 \leq n \leq n_{k-1}-1$  and  $k > 1$ .

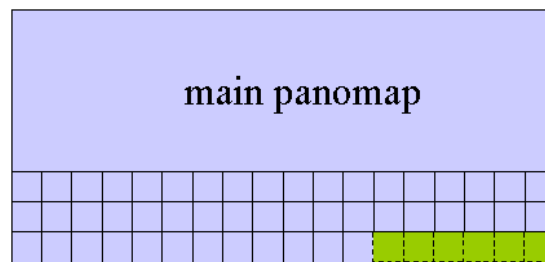
By joining the Boolean function values contained in  $Table^{(1)}(\bullet)$ ,  $Table^{(2)}(\bullet)$ , ..., and  $Table^{(K)}(\bullet)$  in turn, a unique sequence can be obtained, which is referred to as the *index sequence* associated with the residual maps. To be clear, how to generate the index sequence is explained with an example. Table 1 shows an example of the index tables of an object whose main panomap is a  $16 \times 16$  image. From Table 1, it reveals that there are totally 3 residual panomaps of this object. In addition, there are three, two, and one

non-redundant entries in the first-, 2<sup>nd</sup>-, and 3<sup>rd</sup>-stage panomaps, respectively. Then, the index sequence of this object is recorded to be 110110110, i.e., it is generated from a column-major order of the Boolean values of Table 1. The index sequence can be used to identify the positions of non-redundant blocks in each residual map. Notice that the size of an index sequence is usually very small because it summarizes every 8×8 block as a single bit. For example, consider a 360×240 image that has (360/8)×(240/8)=1350 entries in the index table of its first-stage panomap. Assume that, in average, half of the entries are associated with non-redundant blocks in each stage, then the length of the index sequence is  $1350+0.5\times 1350+0.5^2\times 1350+\dots=2700(\text{bits})$ , which takes only  $2700/8=337.5(\text{bytes})$ .

In our approach, the non-redundant blocks are recorded by appending them to the tail of the main panomap in turn. That is, associated with each “1” in the index sequence, an 8×8 non-redundant block is appended in a row-major order. Since our purpose is to tile these blocks to form a single image, some further 8×8 blocks whose pixels are all (0,0,0,0) are used to fill the empty sites such that a rectangular image can be made, as shown in Fig. 3. This rectangular image is referred to as a *compact multistage panomap* (CMP) of the object. By combining both the CMP and the index sequence of an object, the depth and color information can be completely recorded.

**Table 1.** An example of the index tables

	Index table of the first-stage map (4 entries)	Index table of the 2 <sup>nd</sup> -stage map (3 entries)	Index table of the 3 <sup>rd</sup> -stage map (2 entries)
(0,0)	$Table^{(1)}(1)=1$	$Table^{(2)}(1)=1$	$Table^{(3)}(1)=1$
(0,1)	$Table^{(1)}(2)=1$	$Table^{(2)}(2)=0$	NONE
(1,0)	$Table^{(1)}(3)=0$	NONE	NONE
(1,1)	$Table^{(1)}(4)=1$	$Table^{(2)}(3)=1$	$Table^{(3)}(2)=0$



**Figure 3.** Illustration of the CMP. 8×8 redundant blocks are appended to the tail of the main panomap in turn, while some empty blocks (i.e. blocks with all its pixels being (0,0,0,0)) shown with dashed lines are used to allow a rectangular image to be made.

## 2.4 Compression of CMP

A CMP is actually a map in which each pixel contains four channels, red (R), green (G), blue (B) and depth (D). If a 3D object is opaque, the D channel of its CMP can be normalized to fit in the range from 0 to 255, and then be saved into the alpha channel of the image. Hence, a CMP can be compactly represented as a single image. Figs. 4(a) and 4(b) show a 3D object (the “Summer Kitty Cat”) in accompany with its colors from different view points, which was acquired with a 3D reconstruction method combining both the contour and stereo information [8]. Fig. 4(c) shows the CMP associated with this textured 3D object, while the left part is associated with the color and the right part is associated with the depth. The resolution of this CMP is  $360 \times 192$ , which takes 276.48 KB (kilo bytes) without further compression. Since a CMP completely preserves the depth and color information of an object if the index sequence is given (in this case, the index sequence takes less than 0.14KB, as shown in Table 2), it can be used to reconstruct views of the objects from different positions and orientations. Fig. 4(d) shows some of the reconstructed views with the CMP, which reveals that a very compact representation can be used to record each view of a 3D object. Fig. 5 shows another example (the “China Kitty Cat”) about the CMP and the reconstructed images.

The CMP can be further compressed to achieve more efficient representation. Since a CMP is an image, standard image-compression methods can be applied to it. Some compression strategies investigated in our work are discussed below.

1. Using a lossless compression method, such as PNG, to compress the CMP.
2. Using a lossy compression method, such as JPEG, to compress the CMP.
3. Split the CMP to be a color part and a depth part, (as shown in Fig. 4(c) or 5(b)). Compressing the color part with a lossy method and the depth part with a lossless one.

Table 2 summaries the compression efficiencies and the associated PSNRs, where the PSNRs are measured by comparing the reconstructed images of the decompressed CMPs with the original images shown in Figs. 4(a) or 5(a). Fig. 6 and Fig. 7 show the CMPs compressed with the strategies 2 and 3 of the Summer Kitty Cat and China Kitty Cat, respectively. From the compression results, it reveals that good image quality can be achieved with low bit rates.

As for the compression ratio, consider that the voxel space of the original data of “Summer Kitty Cat” and “China Kitty Cat” are  $126 \times 159 \times 117 = 2343.97\text{K}$  and  $126 \times 110 \times 110 = 1524.6\text{K}$ , respectively. Each voxel requires three bytes to record its color, and thus 7031.9KB and 4573.8KB are required, respectively. If the third strategy



is chosen, 47.14KB and 40.13KB are required for both experiments, respectively. The compression ratios are thus 149 and 114 times, respectively.

### 3. Conclusions

In this paper, we have proposed a new representation, *multistage panomap*, for 3D textured objects. Characteristics of our approach include

1. A compact but complete representation for 3D graphic objects, which combines a 2D image, CMP, and an index sequence, is proposed.
2. The proposed compact representation is appropriate for standard image compression techniques.
3. The developed approach is also suitable for progressive transmission. For example, the main panomap usually captures most parts of the object surface, and thus can be transmitted first. Then, other residual blocks can be transmitted later in turn. We will investigate this point in the future.
4. Experimental results have shown that our approach has great potential to be applied for low bit-rate transmission of 3D graphic objects.

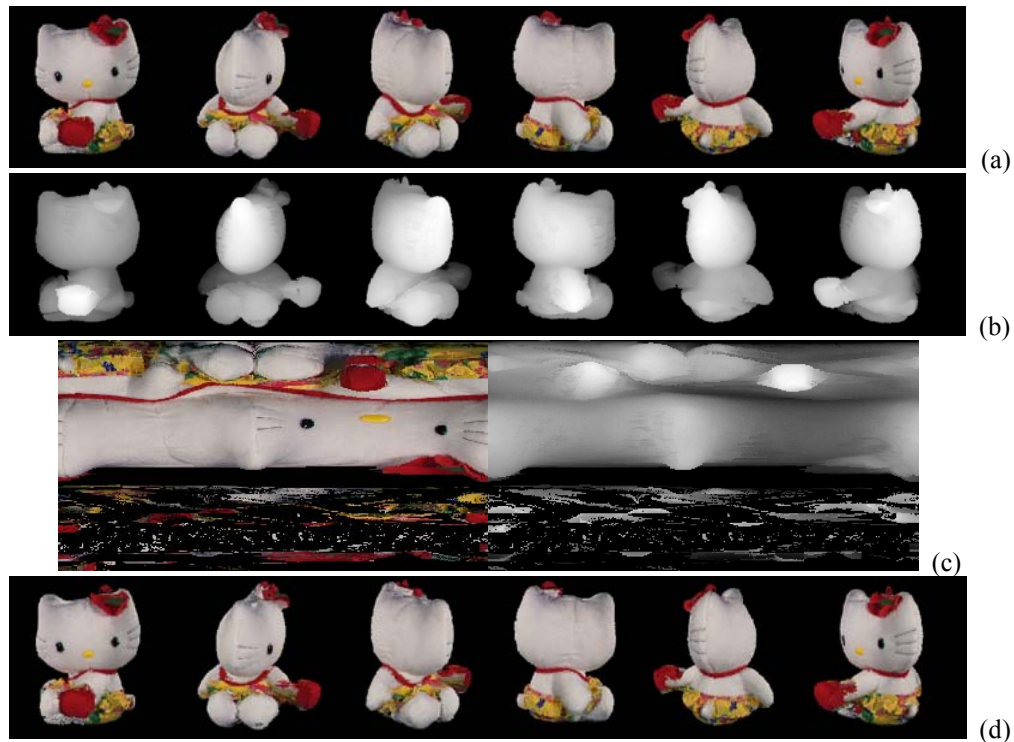
**Table 2:** Comparison of different compression for the complete panomap

	Size of the file (kB)	Size of the file (kB)
	Summer Kitty Cat	China Kitty Cat
CMP (uncompressed)	276.48	241.92
Supplementary Files*	0.14	0.13
Compressing the CMP with PNG	110	101
RGB: JPEG format, depth: PNG format	47	40
PSNR (RGB: JPEG format, depth: PNG format)	34.01 db	33.98 db
Compressing the CMP with JPEG	28.5	25
PSNR (Compressing the CMP with JPEG)	32.00 db	31.26 db

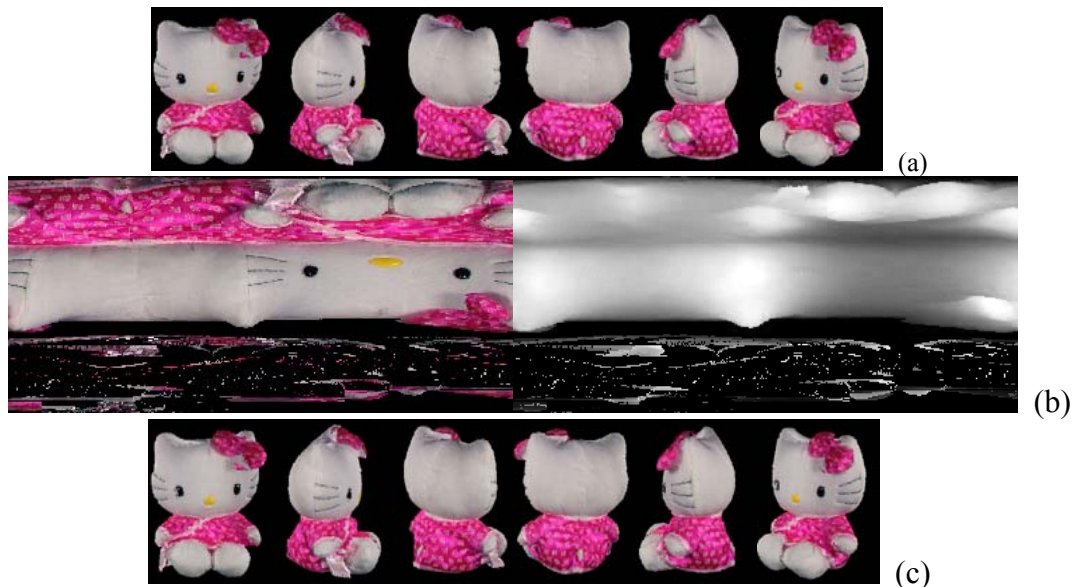
\* Supplementary files include the index sequence and the 2D resolution of the main panomap.

## References

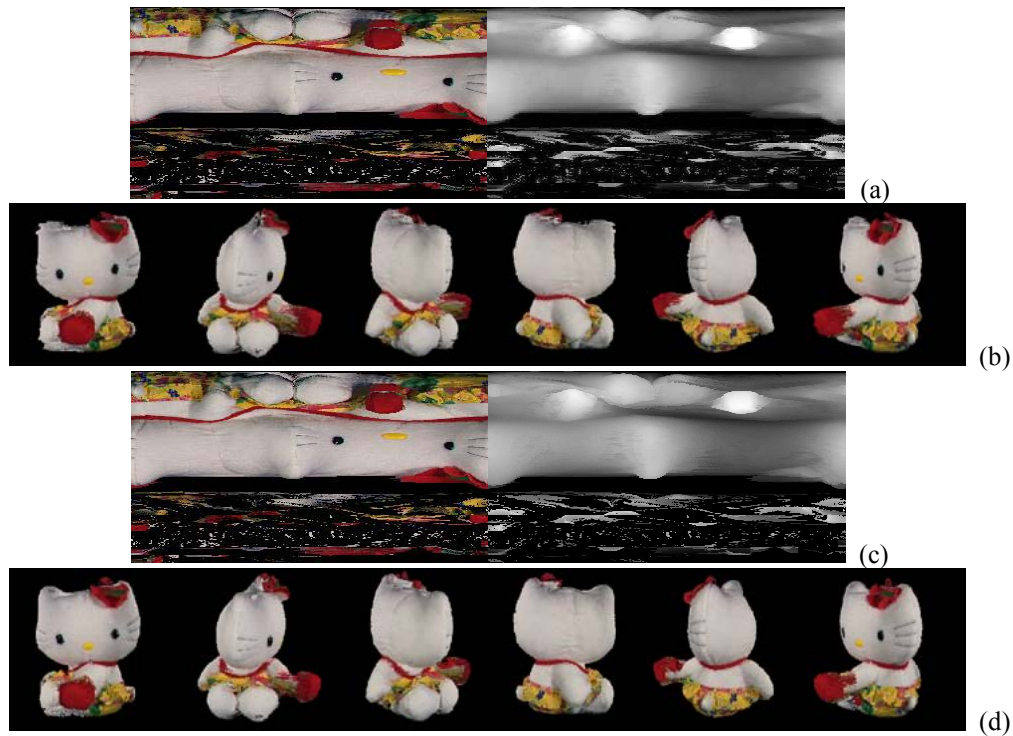
- [1] C. S. Chen, Y. P. Hung, and J. B. Cheng, "RANSAC-based DARCES: A New Approach to Fast Automatic Registration of Partially-Overlapping Range Images," *IEEE Trans. PAMI*, Vol. 21, No. 11, pp. 1229-1234, 1999.
- [2] H. Hoppe, "Progressive meshes," *Proc. SIGGRAPH'96*, 1996.
- [3] D. Langer, M. Mettenleiter, F. Härtl and C. Fröhlich, "Imaging laser scanners for 3-D modeling and surveying applications," *Proc. IEEE Int. Conf. on Robotics & Automation*, April 2000.
- [4] J. Li and C. C. J. Kuo, "Progressive coding of 3-D graphic models," *Proc. of IEEE*, vol. 86, no. 6, June 1998.
- [5] Y. Matsumoto, H. Terasaki, K. Sugimoto, T. Arakawa, "A Portable Three-dimensional Digitizer," *Proc. of Int. Conf. on Recent Advances in 3-D Digital Imaging and Modeling*, pp. 197-204, 1997.
- [6] S. Muraki, "Volume data and wavelet transforms," *IEEE Computer Graphics and Applications*, vol.13 no.4, 1993.
- [7] P. V. Sander, J. Snyder, S. J. Gortler and H. Hoppe, "Texture mapping progressive meshes," *Proc. SIGGRAPH'01*, 2001.
- [8] C. M. Tsai and C. S. Chen, "Finding Minimal Photo Hull for Image-based Rendering by Carving Space with Progressively Stricter Consistent Criterion," *Proc. MVA2000*, Tokyo, Japan, 2000.
- [9] C. K. Yang, "Integration of volume compression and visualization: A survey," *State University of New York at Stony Brook, Research Proficiency Exam Report*, ([http://www.ecsl.cs.sunysb.edu/compression\\_domainVR/compression\\_domainVR.html](http://www.ecsl.cs.sunysb.edu/compression_domainVR/compression_domainVR.html)), September 2000.
- [10] <http://www.photomodeler.com/>



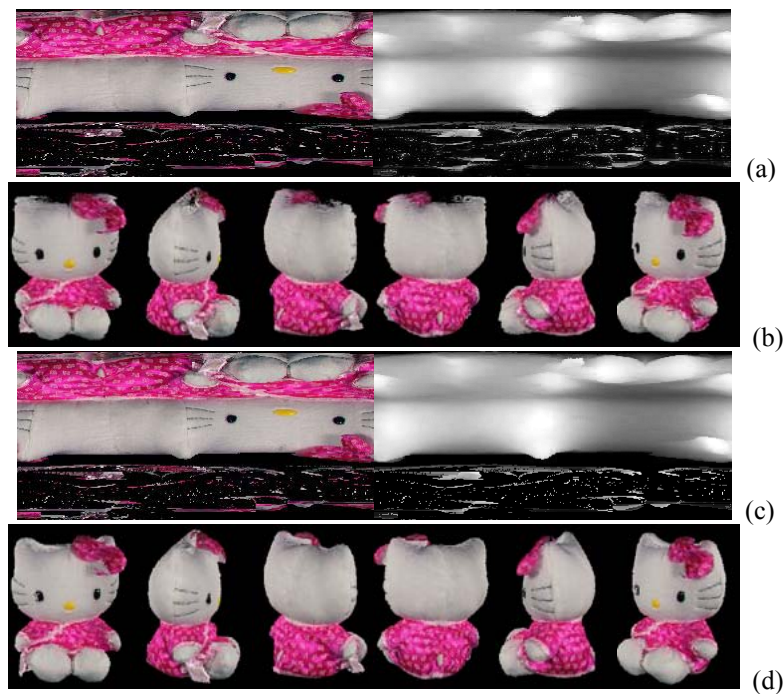
**Figure 4.** A textured object namely, the Summer Kitty Cat. (a) The color part of it. (b) The depth part of it. (c) The CMP of "Summer Kitty Cat". The left side shows its color channels, and the right side is its depth channel. There are 4 layers of the residual panomaps in this case, and each of them is composed of non-redundant  $8 \times 8$  blocks that are appended to the bottom of the main panomap. (d) Some reconstructed views of "Summer Kitty Cat" by using the CMP.



**Figure 5.** The CMP of "China Kitty Cat". (a) The color part of it. (b) The CMP of "China Kitty Cat". The left side shows its color channels, and the right side is its depth channel. (c) Some reconstructed views of "China Kitty Cat" by using the CMP.



**Figure 6.** Lossy compression of "Summer Kitty Cat". (a) Both the color and the depth channel are compressed using JPEG file format. (b) Some reconstructed views of (a). (c) The color channels are compressed using JPEG file format, and the depth channel is compressed using PNG file format. (d) Some reconstructed views of (c).



**Figure 7.** Lossy compression of "China Kitty Cat". (a) Both the color and the depth channel are compressed using JPEG file format. (b) Some reconstructed views of (a). (c) The color channels are compressed using JPEG file format, and the depth channel is compressed using PNG file format. (d) Some reconstructed views of (c).