Research on Image Storage Based on 3D Point Cloud

Xiangning Ma, Jinghao Xu, Cailing Wu, Wanpeng Duan

Computer Science Institute, Xiamen University of Science and Technology, Fujian, China

ABSTRACT

3D technology has been widely used in the fields of architecture, medical image, cultural relic protection, film production and 3D game. The polygon mesh is the most commonly used method for expressing the 3D model, it connects the points on the surface of the 3D model into polygons as a unit, which can express complex surfaces and provide strong adaptability. Especially in the use of triangular mesh the most widely used. But the triangular grid display, simplified, progressive transmission algorithm is not suitable for large-scale data sets. The triangular mesh data file described in this paper with the boundary ball data file conversion system will represent and gradually compress these triangular meshes. This representation is not only compact, but also can be quickly calculated, its biggest feature is the high compression ratio, saving a lot of time and space, which makes it suitable for large-scale data sets. Its execution program can serve large-scale 3D data processing projects. This article has demonstrated that this system can be converted to include a large order of magnitude polygon model.

KEYWORDS: point cloud; polygon grid; boundary ball

Citation: Xiangning Ma, et al. Research on Image Storage Based on 3D Point Cloud. (2017); 1(1): 1–10.

*Correspondence to: Wanpeng Duan, Computer Science Institute, Xiamen University of Science and Technology, Fujian, China. dwpcom@gmail.com

1. Introduction

Three-dimensional data as a new media data cause people's attention, in the process of gradual integration with the traditional application process, three-dimensional data also contributed to the development of new areas of interdisciplinary cross. This design can analyze the application prospect of three-dimensional data from the following aspects:

The 3D model records the precise geometric information of the surface of the object, which contributes to the morphological analysis and geometric measurement of the model. Based on the real three-dimensional data of the geometric analysis, its credibility is much higher than the two-dimensional information analysis. Examples are currently being applied: at different stages of human evolution, changes in the shape of the skull; debris flow on the soil and so on.

Three-dimensional technology can record all-round information of the scene. Compared with the previous two-dimensional image, the three-dimensional data allows the user to observe things from all angles, getting more accurate information. For example, through three-dimensional scanning to record geological exploration, archaeological excavation of the various stages of the scene (because these scenes are often non-reproducible), can be analyzed from different angles of its topography; in some accident scene, three-dimensional data can make people from each point of view to observe the scene, to avoid the two-dimensional images due to the impact of the shadow caused by the problem [1].

Three-dimensional technology can be used to enhance the reality of virtual reality applications to build a realistic scene environment. For example, in the ongoing digital museum project, the three-dimensional model can truly reproduce the shape of the object does not require people to visit the scene. Therefore, three-dimensional modeling can’t only play a major role in cultural relics display, culture and education, it also can protect the original cultural relics. More importantly, the three-dimensional data that records accurate geometric information are of value and contribute to the restoration of artifacts. In addition, the three-dimensional modeling of historical sites can make the people to roam through virtual scenes as if they are in the real environment, from all angles to observe and appreciate these historical treasures, but also for these historical sites to preserve a complete, real large-scale three-dimensional point cloud.
model of the interactive drawing of the data records, once the accidental damage can also be based on these real data to repair and improve. In addition, in the online shopping, film entertainment, online games, remote surgery, culture and education and many other aspects of the involvement of three-dimensional data also contributed to the development of the application [1]. Three-dimensional technology to be applied and the development of the basic premise is the first need to obtain real-world objects in the real three-dimensional information.

2. Ply File Data Structure Analysis

2.1. Ply data storage format

In general, to get the better visual effects and rapid display, three-dimensional surface model after the reconstruction also need to be smooth and simplified follow-up processing. However, the classical surface model reconstruction algorithm (such as the MC algorithm) can only get the spatial information of the reconstructed grid data, and the topology information necessary for grid optimization (the connection between each triangle, edge and vertex) cannot be extracted during the model rebuild process. However, if the grid data obtained from the reconstruction is directly extracted from the topology information, it will consume a lot of computing resources and time. Geng et al. [36] proposed an algorithm for the generation and organization of triangular patches based on the MC algorithm, which greatly shortened the time and space of triangular mesh data generation and organization. The main idea of this algorithm is to allocate a Buffer to store the number of vertices on each Cube edge in the current layer when constructing a layer of triangular mesh, which greatly accelerates the positioning process of the vertices. Lu et al. proposed a method of hierarchically reconstructing and simplifying data, and realized the three-dimensional reconstruction of large data in the case of fewer resources. Soetebier et al. also proposed a method of dealing with oversized progressive meshes. If the three-dimensional reconstruction of massive fault data can be used in parallel processing of multiple computers, it can greatly improve the speed of reconstruction and subsequent simplification. The parallelization method requires the data to be divided into several segments consecutively, implementing reconstruction or organization of a segment or segments on different compute nodes, and then all the resulting grid data are combined to form the result.

2.2. Ply file format

Ply data file format ply Animator Pro creates a graphical file format that contains information about a series of points used to describe polygons:

- Vertex list
- Patch list
- List of other elements

The head is a series of text lines ending in a carriage return, used to describe the remainder of the file. The header contains a description of each element type, including the element name (such as ‘edge’), how much this element is in the project, and a list of different attributes associated with the element. The head also shows that the file is binary or ASCII. Behind the head is a list of elements for each element type, appearing in the order described in the head [7].

The right side is a complete ASCII description of the cube. The only difference between the head of the same project’s binary version is to replace the word ‘ascii’ with the word ‘binary_little_endian’ or ‘binary_big_endian’. The annotations in braces are not part of the file, and they are annotations to this example. The comments in the file are usually in the ‘comment’ at the beginning of the keyword.

```
ply
format ascii 1.0 {ascii / binary, format version}
comment made by anonymous {comment Keyword description, like other lines}
comment this file is a cube
element vertex 8 {Define ‘vertex’ element with 8 in the file}
property float32 x {vertex containing floating point coordinate ‘x’}
property float32 y {y coordinate is also a vertex attribute}
property float32 z {z is also coordinates}

element face 6 {in the file there are six ‘face’ (dough)}
property list uint8 int32 vertex_index {‘vertex_indices’ is a column of integers}
```
This example illustrates the basic composition of the head. Each part of the header is an ASCII string that begins with a keyword beginning with a carriage return. Even the beginning and ending of the head ('ply' and 'end_header') are in this form. Because the character 'ply' is the magic number of the file, it must be the first four characters of the file. Followed by the beginning of the file header is the keyword 'format' and a specific ASCII or binary format, followed by a version number. And then the following is a description of each element in the polygon file, there are many attributes in each element description. The general elements are described in the following format:

    element <element name> <number in file>
    property <data type> <attribute name -1>
    property <data type> <attribute name -2>
    property <data type> <attribute name -3>
    ...

The attribute is listed after the 'element' line, which contains both the data type of the attribute and the order in which the attribute appears in each element. A property can have three data types: scalar, string, and list. Attributes may have a list of scalar data types as follows:

    Name Type Bytes
    ---------------------------
    int8 character 1
    uint8 non-negative character 1
    int16 short integer 2
    uint16 non-negative short2
    int32 integer 4
    uint32 non-negative integer 4
    float32 single precision floating point 4
    float64 double precision floating point 8

These byte counts are important and cannot be modified in the implementation process to make these files portable. There is a special format for defining attributes using list data types:

    property list <numeric type> <numeric type> <attribute name>
One example of this format is in the above cube file:

```
property list uint8 int32 vertex_index
```

This means that the attribute 'vertex_index' first contains a non-negative character, which contains the number of indexes in the attribute, followed by a list containing many integers. Each integer in this side list is a vertex index [7].

### 2.3. Ply data grid compression

The mesh simplification method reduces the number of vertices and triangles in the triangular mesh, reduces the amount of data stored and transfers, and can recover the necessary details in the drawing process through the method of grid reconstruction and multiresolution modeling. The three-dimensional grid compression method takes another approach, that is, through the encoding / decoding process to reduce the geometry of the grid model, topology information storage capacity, and not directly on the grid model to reduce the details of the information.

The most commonly used method of grid compression is to split the triangular mesh model into a continuous triangular band, so that the repetition of the triangle vertex is greatly reduced and the storage space. This method has a lot of application and development. In addition, the parallel quadrilateral prediction method can also reduce the storage of vertex coordinates (i.e., geometric information of the grid). Another kind of grid compression method encodes the geometric information of the grid by means of coordinate quantization, which is a damaged compression method. In the recent years, there is some ways the original grid is transformed into a semi-regular mesh by re-building the network, and then the geometric information of the grid is compressed by some conventional signal processing tools such as wavelet analysis so that the grid in the geometric and topological aspects of the compression are detrimental. In recent years the rise of the 'point-based (point-based)' approach can also be seen as a means of grid compression. It completely abandoned the topological connection in the grid model, only save the geometric information part, thus greatly reducing the amount of data, but the effect of the model is also a corresponding loss. The grid compression method, while reducing the amount of data, does not simplify the details of the model and is therefore very useful in data transmission; but it also fails to provide a multi-resolution model with different LOD and there are many ways to apply a grid coding algorithm in the large data volume model simplified out-of-core method, and achieved good results [8].

### 2.4. Point cloud data and three-dimensional grid (ply) data conversion

Imageware uses NURBS technology, software is powerful, and easy to use, low hardware requirements, can run on a variety of platforms: UNIX workstations, PC, the operating system can be UNIX, NT, Windows95 and the other platforms. Imageware due to advanced engineering in reverse engineering, the product is occupied by a large market share, software revenue is growing at an annual rate of 47%. Surfacer is the main product of Imageware, which is mainly used for reverse engineering, the process of dealing with data follows the point-curve-surface principle, the process is simple and clear, and the software is easy to use. The process is as follows:

#### 2.4.1 Point process

Read lattice data. Surfacer can receive almost all the coordinate measurement data, but also can receive other formats into stl, vda and so on. Align the separated lattices together (if needed). Sometimes because the parts shape complex, a scan can’t get all the data, or some parts cannot be a scan to complete, which requires moving or rotating parts, which will get a lot of separate dot matrix. Surfacer can use the special point information such as cylindrical surface, sphere, plane and so on to precisely align the lattice. Judging the lattice, removing the noise point (i.e. measuring the error point). Due to the limitations of measuring tools and measurement methods, there are sometimes noise points, Surfacer has a lot of tools to judge the lattice and remove the noise points to ensure the accuracy of the results.

#### 2.4.2 Curve creation process

Determine and decide which type of curve to generate. The curve can be precisely through the lattice, or it can be very smooth (capture the lattice represents the main shape of the curve), or between the two to create a curve. To create a curve as needed, you can change the number of control points to adjust the curve. The control point is increased, the shape of the fit is good, the control point is reduced, the curve is more smooth. Diagnose and modify curves. The curvature of the curve can be judged by the curvature of the curve, it is possible to check the consistency of the curve with the lattice and to change the continuity (connection, tangency, curvature) of the curve and other curves. Surfacer provides a lot of tools to adjust and modify the curve.

#### 2.4.3 Surface creation process

Decided to generate the kind of surface. As with the curve, you can consider generating more accurate surfaces, smoother surfaces (such as class 1 surfaces), or both. According the product design needs to decide to create a surface.
There are many ways to create a surface, you can use the dot matrix directly generated surface (Fit free form), you can use the curve through the skin, sweeping, four border lines and other methods to generate the surface, it can also combine the lattice and curve information to create a surface. But also through other, such as the corner, over the bridge and another provincial capital surface. Diagnose and modify surfaces. Compare the curvature of the surface with the lattice, check the smoothness of the surface and the continuity of the other surfaces, and can be modified, for example, you can align the surface with the lattice, adjust the surface control point to make the surface smoother, or the surface is reconstructed and processed. "With Surfacer, you can do more design cycles in less time, it can reduce design time by 50 percent" said Chris Chatburn, a design engineer at Triumph Motorcycles in the UK. [9].

2.5. Ply data files in different models in the storage mode analysis

2.5.1 Storage structure of different models

Speaking of byte sorting problems, will inevitably involve two CPU factions. That's Motorola's PowerPC series CPU and Intel's x86 series of CPUs. PowerPC series uses big endian way to store data, while the x86 series uses little endian way to store data. ARM also supports big and little; the actual application usually use little endian. In fact, big endian refers to the low address stored in the most significant byte (MSB), while the little endian is the low address to store the least significant byte (LSB). The text may be more abstract, the following image to illustrate. Such as the number 0x12345678 in two different byte order CPU storage order as follows:

**Big Endian**

A high-bit Byte in Word is placed in memory at the low address of the Word area

<table>
<thead>
<tr>
<th>Low address</th>
<th>high address</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - +</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>34</td>
</tr>
</tbody>
</table>

**Little Endian**

A low-bit Byte in Word is placed in memory at the low address of the Word area

<table>
<thead>
<tr>
<th>Low address</th>
<th>high address</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - + - +</td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>56</td>
</tr>
</tbody>
</table>

From the above two figures, using big endian way to store data is in line with human thinking habits. It must be noted that the length of a Word is 16 bits, and the length of a Byte is 8 bits. If a number exceeds the length of a Word, you must first divide the Word into several parts, and then each section (that is, each Word inside) by Big-Endian or Little-Endian different operations to deal with bytes.

2.5.2 Storage structure conversion algorithm design

The design of the conversion algorithm is in fact the cycle of the data on the transposition of the address, the following is the main part of the algorithm, with the template will enhance the usefulness of the function

```c
F3D_INLINE T ConvertEndian (T t)
{
    T tResult = 0;
    for (int I = 0; I <sizeof (T); ++ I)
    {
        tResult <<= 8;
        tResult = (t & 0xFF);
        t >>= 8;
    }
    return tResult;
}
```

The code above will provide a convenient function implementation for byte sorting conversion.
Research on Image Storage Based on 3D Point Cloud

2.6 Experimental results analysis

In dealing with point cloud data, the first point to the original point cloud data for noise reduction. This process is also used by other processing software. For those little or almost no noise point cloud model, you can use imageware on the point cloud data processing STL data format, and then use the 3D Object converter STL data processing into ply data files, this method is simple but the point of the quality cloud data model is higher. Imageware for the large amount of data ply file display is very good, but it will take up a lot of time and space. If the number of sampling points is large, the number of polygons obtained from the sampling points is also large (if the sampling points are connected to triangles, the number of triangles is about twice the number of sampling points). For example, Stanford University's digital Michelangelo project produced the St. Matthew model with more than $1.86 \times 10^8$ triangles [1]. If you want to store the St. Matthew model on the hard disk, you will need more than 6G of space, for such a huge amount of data, read from the hard drive to spend several minutes, the need for more time on the network transmission. This also produces ideas that want to simplify the amount of data. A new approach will be introduced later to solve this problem.

3. Qsplat file structure analysis and transformation algorithm design

3.1. Qsplat data structure and quantization algorithm

The triangular mesh is generated by adjusting and integrating the scanned point cloud data, which can be processed by the imageware software described earlier. On this basis, place a sphere on each node, which is large enough to be in contact with the adjacent sphere. Qsplat is a hierarchical structure by coloring the borders of different regions. [19].

Qsplat uses the hierarchical structure of the borders as a visual elimination technique, multi-level control and drawing. Each node in the tree contains the center of the sphere and the radius, the width of the pan and the pan, and the color (optional). Although the algorithm requires only one algorithm to generate the hierarchical structure of the boundary sphere from the triangular mesh, Qsplat can also generate such hierarchies from polygons, voxels, or point cloud models. This hierarchy is constructed as a preprocessor and is written to disk [19].

Each node in the tree contains: the position and radius of the ball, the normal vector at each point, the width of the pan pan, the color value (optional), each node with 4 bytes (no color information) or 6 Byte (when there is color information), as shown in Figure 1:

![Figure 3-1 Qsplat node layout](image)

(A) The tree is stored in a width-first manner (in the order indicated by the red line). (B) The connection of the parent node to the child node is established by a pointer from a group of parent nodes to the first child node. If all the parent nodes of the parent node are leaf nodes, then these pointers do not appear. All the pointers are 32-bit (c) a quantized node occupies 48 bits (no color if 32 bits)

3.1.1 Qs sphere position radius and quantization algorithm

The center and radius of each sphere are encoded on the boundary sphere hierarchy relative to their parent nodes. To save the space, the value of these measures set at 13 bits. That is, the radius range of the sphere is 1/13 to 13/13 relative to its parent node, and the offset of the center of the sphere relative to the center of its parent node (each X, Y, Z) is its parent node diameter of the multiple of 1/13. Quantify the top-down, so that the location and range of the child's nodal ball are entered relative to the quantized position of the parent node; thus, the quantization error does not go to the grid Figure 3-4 Qs sphere position radius spread. To ensure that the quantification process does not cause any voids, the quantized radius always converges to the value representing the nearest point to ensure that the quantized sphere is very close to the real sphere.
The position and radius of each node are quickly decoded at the time of drawing. For this reason, the data structure is not only compact on disk, but also requires less memory space at the time of drawing than the solution before decompressing the drawing [19].

**QS sphere quantization algorithm**

pcx is the parent sphere X coordinate, pr is the radius of the parent sphere, cx is the sub-sphere X coordinate, dx is the relative coordinate value, N is 13, INVALID is 0xff ff

\[
\begin{align*}
    dx &= cx - pcx; dy = cy - pcy; dz = cz - pcz; \\
    X1 &= \text{min} (\text{max} (\text{int} (\text{floor} ((rdx - 0.5f * rr) * N)), 0), N-1) \\
    Y1 &= \text{min} (\text{max} (\text{int} (\text{floor} ((rdy - 0.5f * rr) * N)), 0), N-1) \\
    Z1 &= \text{min} (\text{max} (\text{int} (\text{floor} ((rdz - 0.5f * rr) * N)), 0), N-1) \\
    X2 &= \text{min} (\text{max} (\text{int} (\text{ceil} ((rdx + 0.5f * rr) * N)), 1), N) \\
    Y2 &= \text{min} (\text{max} (\text{int} (\text{ceil} ((rdy + 0.5f * rr) * N)), 1), N) \\
    Z2 &= \text{min} (\text{max} (\text{int} (\text{ceil} ((rdz + 0.5f * rr) * N)), 1), N) \\
    \text{Maxdiff} &= \text{max} (\text{max} (X2 - X1, Y2 - Y1), Z2 - Z1)
\end{align*}
\]

3.1.2 QS sphere number and quantization algorithm

Each node occupies 14 bits when the number of pixels is stored. Can be represented by the corresponding number of points on the hexagonal square 52*52 lattice on the following (number of numbers), and want to sample the normal space on the more inconsistent bending. A lookup table is used to decode the number that can be represented when drawing. In practice, only 52 * 52 * 6 = 16224 different numbers are used to generate products that are not visible in the dark parts of the background, but some striped products are visible around high-intensity mirrors in a wide area of low curvature. By moving to each pan number to do the relative to the parent of the pan-number progressive.

Figure 3-2 QS sphere pan quantization

Coding to eliminate these products is possible and to achieve a better compression. But this will increase the complexity of the internal algorithm loop calculation, resulting in a trade-off between space and time. Unlike the range of nodes, the number of spaces is bounded, so a fixed quantization table can satisfy the number of arbitrary number of coding. Thus, in this case, the design chooses a quantization of the number of fixed numbers, which requires only a lookup table at run time. As the processing speed of the processor increases, the design predicts progressive quantization scheme will become more promising.

**Pan Numerical Quantification Algorithm**

```
7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
+---------- +---------- +
| Number of Pan (Normal) | C |
+---------- +---------- +
Byte 0 | Byte 1
```

The pan number is divided into A, U, V three values are as follows:

\[
A = \frac{N}{2704}, U = (N / 52) \% 52, V = N% 52
\]
A value determines the allocation of pan number axes:

A = 0 Positive X U = Y axis V = Z axis
1 Negative X
2 Positive Y U = Z axis V = X axis
3 Negative Y
4 Positive Z U = X axis V = Y axis
5 Negative Z

3.1.3 QS sphere color and quantization algorithm

The color currently occupies 16 bits, R, G, and B are 5, 6 and 5 respectively. When doing the number of operations, the color increase of the code can save storage space, but in the run time to spend more memory.

3.2. Qsplat data compression algorithm

The highly compressed data structure is a major feature of the QSplat method in which the position and radius of the ball of each node are encoded relative to the parent node and are discretized into 13 discrete values the radius of the ball of a node is a value between 1/13 and 13/13 of the radius of the ball of the parent node, the distance of the center of the sphere relative to the center of the parent node is a multiple of 1/13 of the radius of the parent node. The normal vector of each node is also represented by discretization, and each of the faces centered on that point is divided into grid points on a 52x52 grid cube to approximate the direction of the vector. The width of the conical number is expressed by only 2 digits, and the four discrete values represent the case where the sinusoidal value of the half-apex of the conical number is 1/16, 4/16, 9/16 and 16/16, respectively. The color values are discretized to 16 bits. After the above compression, the maximum number of bytes occupied by each node is reduced, and the error is controlled within an acceptable range. The QSplat in memory and memory data representation is the same, do not have to decompress the first work, just transfer the data directly to the memory can be, and thus further speed up the drawing speed.

Since QSplat is primarily intended for rapid display of large data volumes, the required data files for each model are still large, even though the data is transferred remotely through the network, a considerable compression rate is used. It is likely that data transmission cannot keep up with the display speed of the situation. To solve the problem, the Streaming QSplat [18] method modifies the QSplat method. Since QSplat can be aborted at any level in the recursive process, the modification of the Streaming QSplat method takes a very simple and straightforward approach - to abort recursion once the required data has not yet been passed. So, that you can first show a rough image, and other data after the gradual refinement.

4. The programming and implementation of the conversion between ply data and QSplat data

4.1. Data conversion system algorithm design

4.1.1 QS tree node structure algorithm design

Through the understanding of QSplat data structure and tree structure, we know that the key of QSplat structure lies in the design of tree node. When designing the conversion system, I pay attention to Qtree_node design. In the software, I define Qtree_node as follows:

```c
struct QTree_Node {
    point pos; // position of tree node
    float r; // the radius of the tree node
    vec norm; // The pan value of the tree node
    float normcone; // the cone pan value of the tree node
    union {
        QTree_Node * child [4]; // child node
    }
}
```
int refcount; // reference count
int remap; // remapping
short col_tmp[3]; // child node color
} m;
}
color col; // tree node color
static poolAlloc memPool; // into memory
void operator delete (void * p, size_t n) {memPool.free (p, n);} // release memory

4.1.2 Ply file into Qsplat file main flow chart design

In front of the PLY file has been introduced, in the design of the conversion algorithm, first we should design and open the PLY file algorithm, from the above PLY file structure is easy to design the process. The second step is for the PLY data and QS data interface design; the interface is read the data processing. The third part is to calculate the size of the QS tree to be generated for it to set the space. The fourth is to initialize the QS tree, such as ball initialization, color, and the establishment of QS tree structure. The last step is to write a file output to the QS tree storage structure.

4.1.3 Read the ply file algorithm design

When reading a PLY file, you must know the structure of the PLY file, which is described in Chapter 2. The first step in this design is to determine the input file based on some of the keyword characteristics of the PLY file header. The second step is to design the key, that is, the data in the PLY file is extracted and stored. Here is divided into two blocks, one is the number of leaves and leaves the information stored. The other is the number of faces and the surface of the information storage. The third step is that some finishing work, such as reading some of the failure to deal with the function.

The PLY function is according the feature definition of the PLY data and the QS data
read_ply (infilename, numleaves, leaves, numfaces, faces, havecolor, comments)

The parameters in this function
Infilename is the name of the input file,
Numleaves is the number of leaves, and the number of vertices is stored in it
Numfaces is the number of faces, faces are face data
Havecolor for possession of the color, comments for the comment is the information in the PLY header read the program flow diagram of the ply file function

5. Conclusion

Qsplat data processing is much greater in rate than ply data processing. Qsplat system in the new data structure, efficient compression of the data and to achieve the level of LOD control and based on the normal clustering of the visibility removed. In the pretreatment phase, the sampling points are stored in the hierarchical sphere. This data structure allows the use of hardware to accelerate the drawing, in real time to draw hundreds of millions of points of the large model, can make the hardware equipment is poor and large amount of data when drawing a better graphics. So, Qsplat three-dimensional data processing is very promising in the future.

This article has done the following work for ply data and Qsplat data conversion:

1. The development of the information, the design is using imageware software to extract the point cloud data into three-dimensional grid data, and then reconstruct the surface and get. In this understanding of the ply data storage structure, file structure.

2. Qsplat data structure and quantization algorithm design, as Qsplat data structure is a tree node, so the key is to understand the node, the node is the tree leaves, it is Qsplat the most basic unit, carrying Qsplat data. The Qsplat data compression is based on the data structure, the proposed quantization algorithm for data compression is very effective.
3. The implementation of the data and Qsplat data conversion, which includes the design of the Qsplat tree node, the design of the ply data file, the algorithm design of the Qsplat tree, and the final output algorithm. The design compares the advantages and disadvantages of the two data graphs.

This article describes the conversion of ply files and Qsplat files. The future work schedule is as follows:

1. This design is only valid for ply file conversion made up of triangular meshes, and the conversion of ply files for other polygons such as quadrilaterals is not supported. Therefore, in the future, we should study the conversion algorithm of various grid ply files.
2. The experimental graphics are black and white, lack of real effect. So, the future work will focus on the color of the ply data files and Qsplat data file conversion research, making the conversion effect is more real.

References

1. Meng Fang, Large-scale three-dimensional point cloud data interactive drawing research, Peking University doctoral degree thesis, 2005: 1-117
2. Yang Zhenyu, Based on the point of the geometry of the modeling technology, master's degree thesis, 2004: 1-86
3. Wei Tao, Scanning density of different three-dimensional depth of data registration and scene reconstruction, Peking University undergraduate thesis, 2004: 1-46
5. FENG Jie, Simplification of large-scale three-dimensional grid model and viewpoint-based LOD control, Peking University doctoral dissertation, 2004: 1-119
7. Li Xin, Based on the external mass of the triangular grid model compression, Peking University [Undergraduate thesis], 2005: 1-48
8. Zhao Xinfang, Triangular mesh cutting algorithm research, Huazhong University of Science and Technology [Master's degree thesis], 2006: 1-46
10. Lu Ji-xiang, Li Dehua, From point cloud to surface modeling issues, Wuhan University of Technology, 2004, 26 (5): 84-86
13. Mario Botsch, Andreas Wiratanaya and Leif Kobbelt, Efficient High Quality Rendering of
16. Baoquan Chen and Minh Xuan Nguyen, POP: A Hybrid and Polygon Rendering System for
18. Shachar Fleishman, Daniel Cohen-Or, Marc Alexa, et al., Progressive Point Set Surfaces,
21. Leif Kobbelt and Mario Botsch, A Survey of Point-Based Techniques in Computer Graphics,
23. Aravind Kaliah and Amitabh Varshney, Modeling and Rendering of Points with Local
25. Szymon Rusinkiewicz, Marc Levoy, QSplat: A Multiresolution Point Rendering System for Large Meshes, Stanford University, 2000,343-352