International Journal of CAD/CAM Vol. 7, No. 1, pp. 1~10 (2007)

INTERNATIONAL JOURNAL OF CAD/CAM
www.ijcc.org

3D Model Compression For Collaborative Design

Jun Liu*, Qifu Wang, Zhengdong Huang, Liping Chen and Yunhua Liu

CAD Center, School of Mechanical Science & Engineering, HuaZhong University of Science and Technology, Wuhan, Hubei, 430074, P.R. China

Abstract – The compression of CAD models is a key technology for realizing Internet-based collaborative product development because big model sizes often prohibit us to achieve a rapid product information transmission. Although there exist some algorithms for compressing discrete CAD models, original precise CAD models are focused on in this paper. Here, the characteristics of hierarchical structures in CAD models and the distribution of their redundant data are exploited for developing a novel data encoding method. In the method, different encoding rules are applied to different types of data. Geometric data is a major concern for reducing model sizes. For geometric data, the control points of B-spline curves and surfaces are compressed with the second-order predictions in a local coordinate system. Based on analysis to the distortion induced by quantization, an efficient method for computation of the distortion is provided. The results indicate that the data size of CAD models can be decreased efficiently after compressed with the proposed method.

Keywords **: Collaborative product development, model compression, hierarchical structure, B-spline, second-order prediction**

1. Introduction

With rapid development of economic globalization and Internet Technology, manufacture enterprises have extended from traditional regional cooperation mode to global collaboration mode. In the course of product development, a great deal of CAD product models are exchanged and shared between cooperators via network transmission. With products becoming increasingly complex, increased products data have brought a heavy burden to storage and transmission, resulting in drastic efficiency reduction of cooperated product development under distributed network environment. To compress product model data for fast network transmission has become a bottleneck technology that needs an urgent solution [1].

Until now, little work has been done on the study of compression and transmission algorithms for precise CAD product models. Although many researchers have made studies to the compression of 3D model and presented a lot of efficient algorithms [2-6], most of them are for approximate polygon mesh models, unsuitable for 3D CAD models created with accurate mathematical representations. Recently, research to the accurate CAD model transmission has gained more and more attentions [7-9]. These studies focus on the incremental transmission of boundary representation (BREP) CAD model without covering the analysis of structural redundant data and pertinent compression method.

Compression methods for polygon mesh model have been studied for a long time and have a vast literature. Taubin [2], Touma [3] and Bajaj [4] have each developed an efficient coding strategy of single-resolution mesh compression. The basic idea of single resolution compression is to traverse each triangle or vertex of the mesh and represent the results into a symbol sequence, which is then coded. In this way, the connectivity of polygon can be compressed. Using the encoding order of traversal, the position of a vertex can be predicted by the positions of previously encoded vertices. The difference between the actual position and the predicted position is encoded as an integer. Other attributes such as normal, color, etc. can be compressed in the same way. To ease the contradiction between limited bandwidth and fast transmission of huge models, an algorithm for progressive transmission is proposed. The idea of the algorithm is to transmit a group of basic meshes to remote clients first, then, provides clients with a gradually refined model and as a result, the model is increasingly refined to approximate the original model eventually. Hoppe [5] introduced Progressive Mesh (PM) which transforms the original mesh models into simpler base meshes and a sequence of detail records that represents how to gradually restore base meshes to the original model. Taubin [6] improved the PM algorithm and developed Progressive Forest Split (PFS) scheme.

Due to the inherent limitation of polygon mesh model in representation of product model, some researchers turn to the study of accurate geometry model compression. Martin [10] proposed a method for compressing floating-point coordinates with predictive coding in a completely lossless manner. The predicted and the actual floating-point values are broken up into sign, exponent, and mantissa and their corrections are compressed separately with context-based arithmetic coding. Diego [11] presented a method for compressing NURBS 3D models with a small and controllable loss. The method employs a DPCM coder with parallelogram

*Corresponding author:
Tel: +86-(0)27-8754-7405
Fax: +82-(0)27-8754-7405
E-mail: liujun_hust@126.com

predictors and a uniform scalar quantizer, followed by entropy coding. Wang [12] simplified complex CAD model by removing Non-geometric data and fitting B-spline surfaces with sample points less than original.

In summary, compression and fast transmission of 3D models are highlights of this field. However, polygon based 3D models are approximation to the accurate CAD models and the corresponding compression algorithms are unsuitable for accurate CAD product models. The compression of accurate models remains to be a research issue that needs to be addressed. In fact, the internal structures of accurate CAD models are highly similar. It is crucial to efficient compression and fast transmission of 3D models to study the features of model structures and the law of the redundancy information distribution.

This paper analyzes the structure and redundancy information of CAD product models and presents an encoding scheme for the features data in hierarchical structure of CAD models. The distribution of free-form curves and surfaces in the geometric data are analyzed, and a second-order prediction algorithm under local coordinate system is proposed to compress control points. The prediction errors are quantized and entropy coded. The distortion introduced by quantization is also analyzed, and a method to compute quantization distortion of curves and surfaces rapidly from quantization errors of control points is provided.

The remainder of this paper is organized as follows. In Section 2 we give a brief overview of hierarchical representation of CAD models. After this, Section 3 introduces the compression of feature layer and geometry layer, which is focused on the predictive encoding of free-form curves and surfaces in geometric data. Then, Section 4 describes system implementation and Section 5 gives compression results. Finally, the paper is finished with a summary to our contributions in Section 6.

2. Hierarchical structure of CAD models

A complete product model contains geometric data, feature information, assembly information, product attributes, manufacture information, etc. Product information is usually abstracted as attributes layer, geometry layer, topology layer and assembly layer. Figure 1 shows the structure of traditional CAD model.

It's difficult to analyze the information redundancy and transmit model incrementally, due to the tight coupling of layer data in traditional model structure. Therefore, the model data are reorganized and a B-rep based hierarchical representation structure is set up.

As Figure 2 shows, the bidirectional relationships between layer data are decoupled in hierarchical structure, and translated to a structure of single direction list. The process of reorganization is as follows. First, Hierarchical Data Buffer (HDB) is introduced to store the model data of every layer. Five HDBs are provided correspond to the number of layers. Second, all nodes of the layer data are tagged. Finally, the directed graph of nodes is traversed and every node is wrote to the HDB according to it's layer. After finishing the traverse of graph, the model data are decomposed and every layer data are stored in the HDBs. In the hierarchical structure, the relationships between nodes are maintained by the tag. The node of topper layer records the tag of lower layer node, then the structure of model data can be described as a single direction list. The detailed layer information is described as follows:

The attribute layer, which can be described as $ATT = \{\alpha_1, \alpha_2, ..., \alpha_m\}$ is a set of attributes, including not only part attributes but also feature and geometric attributes. The attribute data describe name, identification, engineering attribute, physical attribute and color, texture, etc.

The geometric layer records the geometric shape of model accurately. It can be described as $GEOM = \{PT, CUV, SURF\}$, where PT denotes the set of points, CUV denotes the set of curves, and $SURF$ denotes the set of surfaces. The points are recorded with coordinates. The analytic curves and surfaces are expressed in the form of equation. The free-form curves and surfaces are defined with knots and control

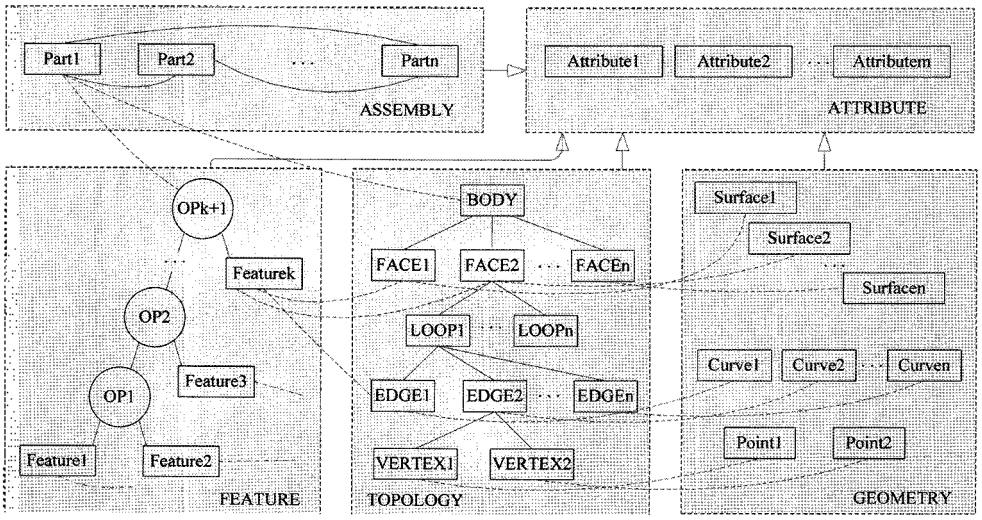


Fig. 1. CAD model structure

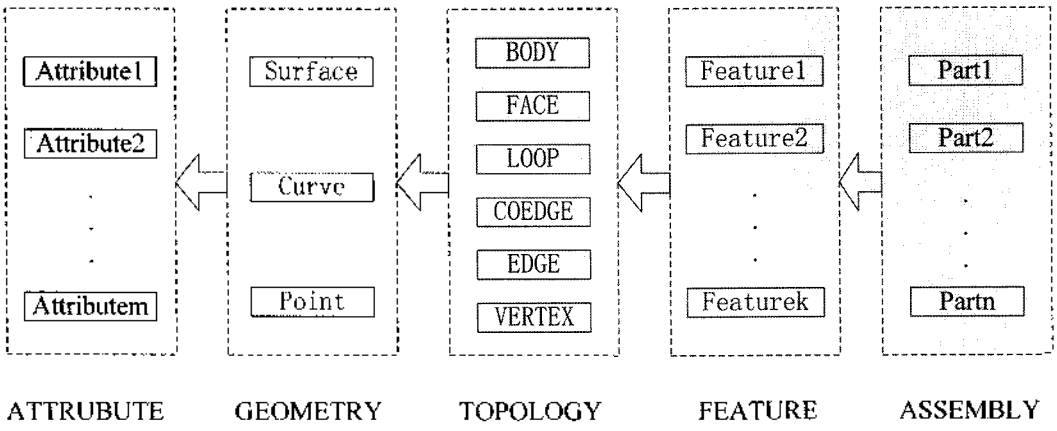


Fig. 2. Hierarchical structure of CAD models

points.

The topology layer records the relationships of geometric element by B-rep. It is represented as $TOP = \{top \mid < geom_i, geom_j > geom_i, geom_j \subset GEOM, i \neq j\}$, where *geom* denotes the element in geometric layer.

The feature layer is represented as $FET = \{F,G\}$, where $F = \{f_1, f_2, ..., f_k\}$ denotes the set of feature nodes, $G = \{g \mid g = (f_i,f_j), f_i, f_u \in F, i \neq j\}$ denotes the constraint graph of features, which describing the constraint relations between feature nodes.

The assembly layer is described as $ASM = \{P,C\}$, where $P = \{p_1, p_2, ..., p_n\}$ denotes the set of parts, represents the assembly relations.

Organized in such a hierarchical structure, CAD models can be analyzed, and a compressive algorithm for hierarchy information can be developed. Moreover, the hierarchical structure is convenient for incremental transmission.

3. Compression of CAD models

The data of the feature layer and geometric layer weighs heavily in a product model. Therefore, compression of a complex product model can be achieved by compressing the data in both the layers just mentioned. Any complex CAD model is formed from a series of simple features through Boolean operation and local operation. A few feature operations such as sweeping, rounding, etc. are frequently used in CAD modeling. Therefore, the feature data can be entropy coded based on statistic law. On the other hand, the geometric information in a CAD model is expressed by a uniform representation based on B-spline, and stored in a form of tensor matrix. Some feature operations such as approximation, offsetting, etc. often generate surfaces containing a great deal of control points. Based on analysis of the distribution law of these control points, geometric prediction can be used to compress it. The compression algorithms for the feature layer and geometric layer are respectively described below.

3.1 Feature layer compression

The feature node can be represented as: Feature = {ID, Type, Name, Parameter, Attribute, Topology_ID, Operator}

Where ID denotes the feature identification, which is an integer value; Type, Name and Attribute are textual values. Parameters are float-point values. Topology_ID records the ID of related topology element. The data of feature nodes can be compressed according to their data type respectively.

For numerical values in feature node such as ID, Topology_ID and Parameter, incremental compression scheme is applied.

Name and Attribute record the feature information for design and manufacturing, and the terms of which can be normalized or even standardized. Taking advantage of this characteristic, Name and Attribute can be encoded by dictionary encoding [13]. A dictionary encoding assigns an integer to each new word in Names and Attributes, and stores the mapping from codes to strings in a dictionary. The pseudo-code is shown below:

```
Initialize Dictionary;
e ← the first text;
while (TRUE) do
    Input the next text N;
    if N not exist then
    Output Index of e;
Break;
end if
if eN exist in Dictionary then
    e ← eN;
else
    Output Index of e;
    Add eN to Dictionary;
e ← N;
end if
end while
```

At the beginning of encoding, the dictionary is empty, which is built up dynamically in the course of coding. Therefore, the compression rate is low in the beginning. To overcome this disadvantage, the frequently appearing Names and Attributes in features are obtained by statistics before coding and the compression rate will be improved by adding these Names and Attributes to dictionary when initialization. Similarly, other textual data can be compressed with the same algorithm.

3.2 Geometry layer compression

The shape of CAD model is described by geometry layer which is composed of points, curves, surfaces, etc. The analytic curves or surfaces are expressed in the form of equation with only a small amount of information, and free-form curves or surfaces are computed from control polygon, which often contains a great number of control points resulting in a drastic increase of model data. For example, the offset surfaces and transition surfaces in a CAD model are often approximated with B-spline surfaces, where a great amount of control points are often generated for higher precision. As shown in Figure 3, the size of model data is 29 K before fillet operation, which drastically increases to 538 K after filleting and approximating with B-spline. The approximate surface contains 344 × 25 = 8,600 control points. Therefore, the geometric data can be compressed through encoding the control points of B-spline curves and surfaces.

B-splines are defined with knots and control points. The knots and control points usually show certain regularity as well as some redundancy. Therefore, they can be compressed with predictive coding. Parallelogram rule is commonly used in predictive coding of polygon mesh model, i.e. assuming that the adjacent two triangles form a parallelogram. However, the prediction accuracy of parallelogram rule for control points in B-spline is low. When analyzing the B-spline in a CAD model, we can find that the curve or surface curvatures are normally continuous with very few cusps. The trend of control polygon is basically the same as that of curves and surfaces. The predictive accuracy can be improved with the help of continuity of curvature.

According to the above analyses, a second-order prediction algorithm in local coordinate is presented for control polygon. The procedure of the algorithm includes the following steps. First, local coordinate systems (LCS) are set up on each control point. And the control points are predicted linearly with first-order predictor in LCS to obtain the first-order prediction errors which describe the trend of curvature. Second, the first-order prediction errors are predicted linearly to obtain the second-order prediction errors. Finally the second-order prediction errors are transmitted to quantizer and entropy coder. The points after quantization are fed back to the first-order predictor. The LCS is set up to eliminate the influence of the world coordinate system. Predict under such a local coordinate system of each point, the values of prediction errors can be reduced and the distribution can be more centralized. The quantized point is fed back to the first-order prediction of next control point to avoid the propagation of the quantization error. The knot vector is decomposed into break vector and multiplicity map. The multiplicity map is entropy coded without prior processing. For the break vector, prediction and uniform scalar quantization are used prior to entropy coding.

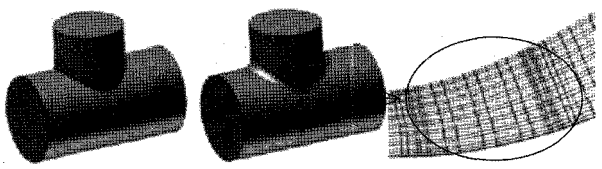


Fig. 3. Control polygon of free-form surface.

3.2.1 Predictive coding of control points

The predictive algorithms of curve and surface control points are described below respectively.

1) Predictive coding of curves

Given control point vector $d_i (i = 0, 1, ..., n)$, degree k, knot vector $U = [u_0, u_1, ..., u_{n+k+1}]$ the B-spline curve equation is is defined as $p(u) = \sum_{i=0}^{n} d_i N_{i,k}(u)$, where $N_{i,k}(u)$ denotes the ith B-spline function. Figure 4 shows the B-spline curve prediction, where Figure 4(a) shows the control polygon of curve, Figure 4(b) shows the first-order prediction of previous control point, Figure 4(c) shows the prediction of current control point. $(x-y-z)$ denotes the world coordinate system. d_i is the current coding control point. d_{i-1}, d_{i-2}, d_{i-3}, d_{i-4} denote the actual control points. The coded points fed back to first-order predictor are denoted by $\hat{d}_{i-1}$, $\hat{d}_{i-2}$, $\hat{d}_{i-3}$. The prediction algorithm is as follows:

1 Construction of the local coordinate systems

The local coordinate system $(x_i - y_i - z_i)$ of current coding point is set up with $\hat{d}_{i-2}$ as the origin, $(\hat{d}_{i-2}, \hat{d}_{i-1})$ as axis x, the plane formed of the points $\hat{d}_{i-1}$, $\hat{d}_{i-2}$, $\hat{d}_{i-3}$ as xoy plane. The unit vectors of the axes in $(x_i - y_i - z_i)$ are expressed as follows:

$$OX_i = \frac{\hat{d}_{i-1}, \hat{d}_{i-2}}{|\hat{d}_{i-1}, \hat{d}_{i-2}|}, \quad OZ_i = \frac{(\hat{d}_{i-1}, \hat{d}_{i-2}) \times (\hat{d}_{i-3}, \hat{d}_{i-2})}{|(\hat{d}_{i-1}, \hat{d}_{i-2}) \times (\hat{d}_{i-3}, \hat{d}_{i-2})|},$$
$$OY_i = OZ \times OX_i$$

Similarly, the local coordinate system $(x_{i-1} - y_{i-1} - z_{i-1})$ of previous control point is set up. The unit vectors of the axes are OX_{i-1}, OY_{i-1}, OZ_{i-1}.

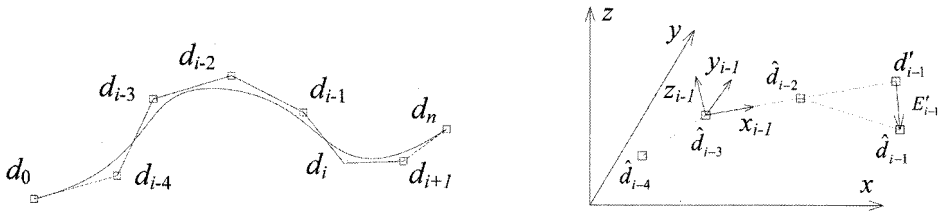
2 First-order prediction of control points.

As Figure 4(b) shows, the previous control point is predicted using parallelogram rule and the predictive point is $d'_{i-1} = 2\hat{d}_{i-2} - \hat{d}_{i-3}$ in world coordinate system with the first-order prediction error $E_{i-1} = \hat{d}_{i-1} - d'_{i-1}$. Transforming E_{i-1} to local coordinate system $(x_{i-1} - y_{i-1} - z_{i-1})$, E'_{i-1} is obtained and its coordinates are $E'_{i-1}.x = (E_{i-1} - \hat{d}_{i-3})^* OX_{i-1}$, $E'_{i-1}.y = (E_{i-1} - \hat{d}_{i-3})^* OY_{i-1}$, $E'_{i-1}.z = (E_{i-1} - \hat{d}_{i-3})^* OZ_{i-1}$, where "*" denotes dot product operation.

Similarly, the first-order prediction error of the current coding control point is computed and transformed to LCS. The first-order prediction point is $d'_i = 2\hat{d}_{i-1} - \hat{d}_{i-2}$. The first-order prediction error is $E_i = \hat{d}_i - d'_i$. E'_i is obtained by transforming E_i to $(x_{i-1} - y_{i-1} - z_{i-1})$.

3 Second-order prediction and quantization

As Figure 4(c) shows, a vector λ_{i-1} is obtained by mapping E'_{i-1} from LCS $(x_{i-1} - y_{i-1} - z_{i-1})$ to LCS $\lambda_0[num\_u]$, that is, the lengths of λ_{i-1} and E'_{i-1} are equal, and the angles between



(a) Curve control points (b) First-order prediction for previous control point

(c) Prediction for current control point

Fig. 4. Prediction of B-spline curve control point.

the vectors and their respective LCS axes are also equal. Predicted in $(x_{i-1}-y_{i-1}-z_{i-1})$, the second-order prediction error E_i'' is obtained using $E_i''=E_i'-\lambda_{i-1}$. By quantizing E_i'', we obtain the quantized prediction error $\hat{E}_i$, which is sent to entropy coder. The detailed algorithms of quantization and entropy coding will be described in Section 3.2.3.

4 Feedback. Transforming vector $\hat{E}_i+\lambda_{i-1}$ to world coordinate system, we obtain $\tilde{E}_i$. The quantized control point is computed using $d_i=2d_{i-1}-d_{i-2}+\tilde{E}_i$. d_i is fed back to the predictor for the prediction of next control point to avoid the propagation of the quantization error.

Repeat the above steps until all control points have been coded.

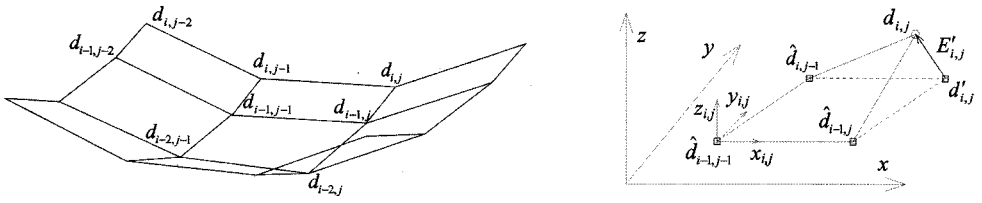
2) Predictive coding of surfaces

The prediction of B-spline surface is similar to that of curve. The primary difference is that the surface prediction uses the adjacent multiple first-order errors together to second-order predict. Figure 5 shows the B-spline surface prediction.

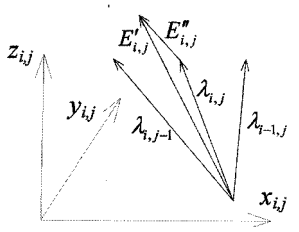
Figure 5(a) shows the control polygon of B-spline surface. $d_{i,j}$ is called the current coding control point. Coded and quantized the actual control points $d_{i-1,j}$, $d_{i,j-1}$, $d_{i-1,j-1}$, the coded control points $\hat{d}_{i-1,j}$, $\hat{d}_{i,j-1}$, $\hat{d}_{i-1,j-1}$, are obtained, which are fed back to the first-order predictor. The quadrangle formed of points $d_{i,j}$, $\hat{d}_{i-1,j}$, $\hat{d}_{i,j-1}$, $\hat{d}_{i-1,j-1}$ is called the current coding quadrangle. Detailed algorithm is as follows:

1 Construction of the local coordinate systems

The local coordinate systems of previous row coded quadrangle, previous column coded quadrangle and current coding quadrangle are set up respectively. The coded quadrangle of the previous row is $\hat{d}_{i-1,j}$, $\hat{d}_{i-1,j-1}$, $\hat{d}_{i-2,j-1}$, ε. Using $\hat{d}_{i-2,j-1}$ as the origin, $d_{i-2,j-1}d_{i-2,j}$ as axis x, the plane formed of triangle $\hat{d}_{i-1,j-1}$, $\hat{d}_{i-2,j-1}$, $\hat{d}_{i-2,j}$ as xoy plane, local coordinate



(a) B-spline surface control polygon (b) First-order prediction of current coding control point



(c) Second-order prediction

Fig. 5. Prediction of B-spline surface

system σ is set up. Similarly, the local coordinate system $(x_{i,j-1}-y_{i,j-1}-z_{i,j-1})$ of previous column and $(x_{i,j}-y_{i,j}-z_{i,j})$ of current point are set up.

2 First-order prediction of control points.

Figure 5(b) shows the first-order prediction of current coding control point. The first-order prediction error is $E_{i,j}=d_{i,j}-d'_{i,j}$ where $d'_{i,j}=\hat{d}_{i,j-1}+\hat{d}_{i-1,j}-\hat{d}_{i-1,j-1}$ is the point of parallelogram prediction in world coordinate system $(x-y-z)$. $E'_{i,j}$ is obtained by transforming $E_{i,j}$ to local coordinate system $(x_{i,j}y_{i,j}z_{i,j})$. Similarly, the first-order prediction errors $E'_{i-1,j}$ and of pre-row and pre-column are obtained respectively.

3 Second-order prediction and quantization

As Figure 5(c) shows, similarly to that of curve, two vectors $\lambda_{i-1,j}$ and $\lambda_{i,j-1}$ are obtained by mapping $E'_{i-1,j}$ and $E'_{i,j-1}$ to the local coordinate system $(x_{i,j}-y_{i,j}-z_{i,j})$. Then the second-order prediction error in LCS $(x_{i,j}-y_{i,j}-z_{i,j})$ is, $E''_{i,j}=E'_{i,j}-\lambda_{i,j}$, where $\lambda_{i,j}=(\lambda_{i,j-1}+\lambda_{i-1,j})/2$. $E''_{i,j}$ is quantized and entropy coded.

4 Feedback. To avoid the propagation of quantization errors, the quantized coordinates of current coding point are computed and fed back to the first-order predictor for the prediction of next control point.

Since the prediction of one point need the points of the previous two rows and two columns, the boundaries of surfaces are handled in a different way. The coordinates of $d_{0,0}$ are recorded directly. The first row $\{d_{0,j}\}$ and first column $\{d_{i,0}\}$ are predicted using the method of curve. The row $\{d_{1,j}\}$ and column $\{d_{j,1}\}$ are predicted by $\{d_{0,j}\}$ and $\{d_{j,1}\}$ using the parallelogram rule. All other points can be handled using the above second-order predictive algorithm.

3.2.2 Knot vectors coding

The knot vector $U=[u, u, ..., t_l]$ is a non-decreasing sequence. It is decomposed into the break vector $t=[t_0, t_1, ..., t_l]$ and the multiplicity map $r=[r_0, r_1, ..., r_l]$. The break vector contains the values of the knot vector, but where multiple knots appear only once, while the multiplicity map expresses the multiplicity of each knot, minus one. The sum of all multiplicity is $\sum_{i=0}^{l} r_i=n+k+2$.

The multiplicity map is entropy coded without prior processing. For the break vector prediction is used prior to entropy coding. The current coding knot is t_i. The predicted value for t_i is $t_i'=2\hat{t}_{i-1}-\hat{t}_{i-2}$ where $\hat{t}_{i-1}$ and $\hat{t}_{i-2}$ are coded knots. Then we obtain the prediction error $\zeta_i=t_i'-t_i$. ζ_i is quantized and entropy coded. The quantized current knot is fed back to predict the next knot.

3.2.3 Quantization of second-order prediction error

The quantization is synchronous with prediction. The purpose of quantization is to map the prediction errors in the form of floating-point numbers to an integer interval. The length of the interval is determined by quantization parameter. The quantization index is $Q_i=\langle \varepsilon_i/\Delta \rangle$, where ε_i is the prediction error, Δ is a given quantization parameter and $\langle\ \rangle$ denotes the rounding operator. The value of the quantization parameter Δ determines the magnitude of Q_i. In the meanwhile quantization has also resulted in quantization error, which is $\Delta/2$ for the maximum. A larger quantization parameter leads to a small value of Q_i but higher quantization error. In contrary, a smaller quantization parameter cause a lower quantization error but a larger value of Q_i, that is, the distortion is small. The quantization parameter Δ can be determined by the allowed distortion together with the compression rate to be achieved.

As a B-spline surface is calculated from control points, the quantization errors of control points will definitely lead to the errors of surface. Given a surface f, the decoded surface is denoted as f'. The distortion between the coded and original surfaces is measured by means of the Hausdorff distance. However, evaluating the Hausdorff distance directly on the surface is an extremely difficult task. Furthermore, it is only a measure of the shortest distance between two surfaces, which is unable to reflect the movements of equal parameter points on surface. The other method is to compute the distances between the equal parameter points on f and f' as the quantization distortion of a surface. More specifically, the method is to divide the parameters in direction u and v into n and m equal parts respectively. The quantization distortion is calculated by

$$l=\frac{\left(\sum_{i=1}^{n}\sum_{j=1}^{m}(f(u_i,v_j)-f'(u_i,v_j))^2\right)^{1/2}}{n\times m}$$

This method is also a computationally expensive operation. In this paper, the following method is used: using curve as an example, the matrix form of a curve is $P=ND$, where P is a vector of the points to be calculated on the curve, N is the basis function, D is the matrix of control points. After the quantization distortion ΔD of control points is introduced, the new points on the curve are $P_{new}=N(D+\Delta D)$, then the quantization distortion for the curve is $\Delta P=N\Delta D$. Similarly, the surface distortion can be computed.

The prediction errors are entropy coded after quantization. As the prediction errors are not completely uncorrelated, QM arithmetic coder [14] is used to eliminate the correlation between prediction errors to further improve the coding efficiency.

The process flow of the second-order predictive coding of the surfaces control points is shown in Figure 6.

3.2.4 Decoding of control points

After compression, the CAD models can be stored and transmitted. The original models can be restored by decoding. Decoding is a reverse process of the prediction step in coding. The construction of local coordinate system and first-order prediction are similar to that in the coding. Using surface as an example, the decoding steps are described below:

(1) Calculate the first-order prediction error $\lambda_0[num\_u]$ in the first row of u direction under the local coordinate system;

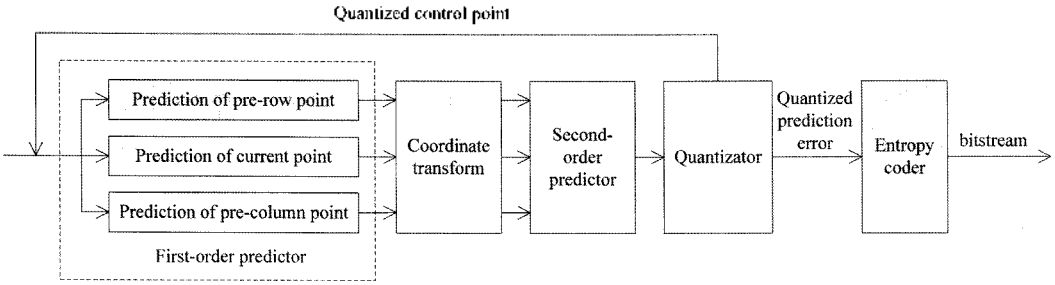


Fig. 6. Process flow of second-order predictive coding algorithm.

(2) Construct the local coordinate system of the previous column and calculate the first-order prediction error λ_{ij-1} in the local coordinate system;

(3) Set the previous row first-order prediction error $\lambda_{i-1,j}$ to $\lambda_0[j]$;

(4) Calculate the current first-order prediction error with $\lambda_{i,j}=\hat{E}_{i,j}+(\lambda_{i,j-1}+\lambda_{i-(1,j)})/2$, where is the second-order prediction error of current control point. Set $\lambda_0[j]=\lambda_{i,j}$;

(5) Construct the local coordinate system of current decoding point and compute the parallelogram prediction point $d'_{i,j}$. Then the decoded point is $d_{i,j}=d'_{i,j}-\lambda_{i,j}$;

(6) Repeat steps 2~5 to obtain the decoded values of all control points.

The original B-spline surface can be restored after decoding the knot vector and control points respectively.

4. System implementation

The above algorithms are implemented in the prototype system, which provides such functions as remote collaborative browse, check, comment, assembly, edit of animation and explosion, etc.

Figure 7 shows the modules and principles of the prototype system. The product models designed in different CAD systems are imported into the prototype system through the interface developed with API provided by CAD systems. The compressed accurate models are obtained by analyzing the redundancies, coding the feature data and free-form curves and surfaces and entropy coding. Then the compressed accurate models can be transmitted incrementally through network and be used in such fields as collaborative development, mass customization and model retrieval.

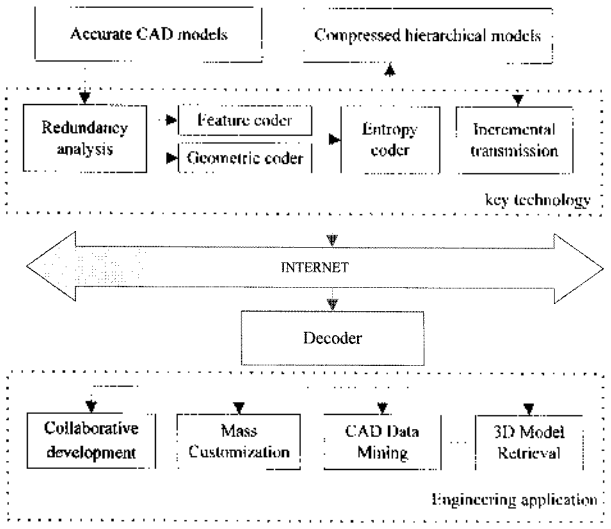


Fig. 7. System framework for prototype system

5. Experimental results

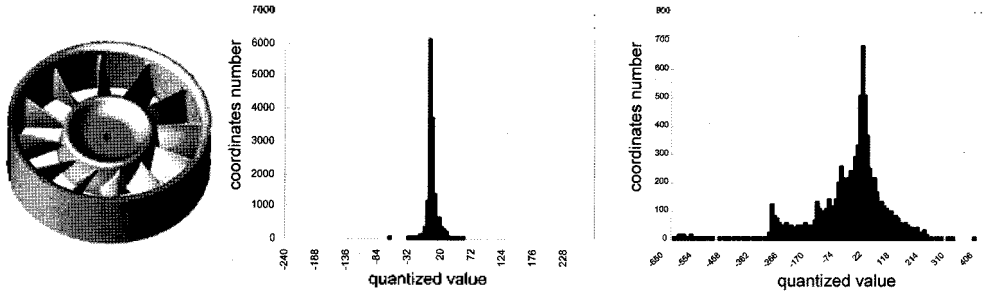
Figure 8 shows the histograms of the quantized prediction error for the control points of the turbine model, calculated over 16698 values, with the quantization parameter being 10^{-4}. As it can be seen, the distribution of the proposed second-order prediction error is more skewed than that of the parallelogram. The results of the both algorithms satisfy the Laplace distribution. The mean square deviation σ of prediction error ε computed from proposed scheme is 25.023, in comparison to that from parallelogram prediction being 133.609, where the σ is computed by $\sigma=\sqrt{E[(\varepsilon-E\varepsilon)^2]}$. Therefore the proposed algorithm has more concentrated error distribution, smaller mean square deviation and higher compressive efficiency.

Figure 9 shows the rendered models after decoding. The Cylinder and Car body model come from UG, the Gear and Handle from ACIS, and the Phone from SolidWorks.

Table 1 shows the data sizes of the feature. It is clearly seen that the feature data can be compressed more effectively by applying differential encoding according to the data type. The structure of features has a big effect on the compression ratio. The implementation result is better for those models contain similar internal structures.

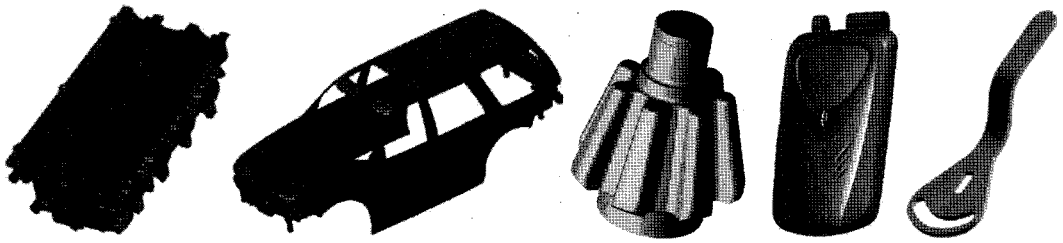
Table 2 illustrates the compression performance of B-spline surfaces of our predictive scheme in comparison to standard gzip compression and general parallelogram prediction at different quantization levels. The files are stored in binary form, and bits/coordinate denotes the number of bits per coordinate.

The experimental results verify that the proposed scheme achieves higher compression rates. The larger the quantizing parameter is, the smaller the data sizes and the higher the quantization distortion will be. Different CAD systems generate B-spline in different method, which results in the compression ratio varying greatly. As there are a lot of ruled surfaces in the gear model, the compression ratio is high; as the curvature



(a) Turbine model (b) second-order prediction (c) parallelogram prediction

Fig. 8. Histogram of quantized control point prediction errors for the turbine model



(a) Cylinder (b) Car body (c) Gear (d) Phone (e) Handle

Fig. 9. Decoded models

Table 1 Compressed size of feature data.

	Uncompressed feature data (KB)	Compressed with proposed method (KB)	Compressed with Gzip (KB)
Cylinder	16053.72	2624.54	4051.28
Car body	3950.35	856.07	1143.21
Gear	153.21	31.76	54.83
Phone	337.21	76.06	101.74
Handle	34.01	9.58	10.26

Table 2 Compressed result of B-spline surface.

		Cylinder	Car body	Gear	Phone	Handle
	Uncompressed (KB)	1682.44	5379.26	240.12	33.56	15.04
	Number of control points	71784	229515	10245	1432	642
	Gzip (KB)	539.32	2059.28	144.28	18.8	8.58
Quantized parameter 10^{-6}	Second-order prediction (KB)	142.69	471.89	30.04	8.24	6.72
	bits/coordinate	5.43	5.61	8.01	15.72	28.58
	Average quantized distortion	0.37×10^{-5}	0.75×10^{-5}	0.8×10^{-5}	0.7×10^{-5}	0.43×10^{-5}
	Parallelogram prediction(KB)	220.31	768.36	58.91	13.52	7.32
Quantized paramcter 10^{-4}	Second-order prediction (KB)	65.05	116.3	8.38	4.63	4.53
	bits/coordinate	2.47	1.38	2.24	8.85	19.27
	Average quantized distortion	0.63×10^{-3}	0.55×10^{-3}	0.74×10^{-3}	0.37×10^{-3}	0.32×10^{-3}
	Parallelogram prediction(KB)	173.05	286.82	17.61	7.75	5.63
Quantized parameter 10^{-3}	Second-order prediction (KB)	22.64	58.16	2.39	3.18	3.03
	bits/coordinate	0.86	0.69	0.64	6.07	12.89
	Average quantized distortion	0.36×10^{-1}	0.41×10^{-1}	0.16×10^{-1}	0.53×10^{-1}	0.21×10^{-2}
	Parallelogram prediction(KB)	83.56	135.63	10.78	5.02	4.71

of the surfaces changes a lot in the handle model, the compression ratio is low.

6. Conclusion

This paper analyzes the characteristics of the hierarchical structure in accurate CAD models, develops a hierarchy information compression algorithm and resolves effectively the bottleneck problem of complicated model transmission existing in collaborative design. For feature information in feature layer, a new coding algorithm is given according to the data type. For free-form curves and surfaces in geometric

layer, a second-order prediction algorithm in local coordinate system is developed, by which the error distribution is more centralized than that of general parallelogram prediction, the mean square deviation is less, the compression ratio is higher and the quantized distortion is smaller and more controllable. Several experimental results are given to verify the effectiveness of the proposed algorithm.

7. Acknowledgements

The authors gratefully acknowledge the financial support from the National Science Foundation of China (NSFC) under grant No. 60673030.

References

[1] Les A, Piegl. (2005), Ten challenges in computer-aided design. Computer-Aided Des 37(4), 461-470.

[2] Taubin, G. and Rossignac, J. (1998), Geometric compression through topological surgery. *ACM Transactions on Graphics*, **17**(2), 84-115.

[3] Touma, C. and Gotsman, C. (1998), Triangle mesh compression. In: Proceedings of the Graphics Interface'98, Vancouver, 1998.

[4] Bajaj, C., Pascucci, V. and Zhuang, G. (1999), Single resolution compression of arbitrary triangular meshes with properties. In: Proceedings of the Data Compression Conference, 1999, 247-256

[5] Hoppe, H. (1996), Progressive Meshes. In: Proc. of SIGGRAPH 96',1996

[6] Taubin, G., Gueziec, A., Horn, W. P. and Lazarus, F. (1998), Progressive Forest Split. In Proceeding of SIGGRAPH'98, 1998, 123-132.

[7] Li, W. D., Ong, S. K. and Fuh, J. Y. H. (2004), Feature-based design in a distributed and collaborative environment. *Computer Aided Des*, **36**(9), 775-797.

[8] Wu, D., Bhargava, S. and Sarma, R. (2000), Solid model streaming as a basis for a distributed design environment. In: *Proceedings of ASME Design Engineering Technical Conferences*, Baltimore, Maryland, DETC2000/DAC-14250

[9] Li, J., Gao, S. M. and Zhou, X. (2003), Direct incremental transmission of boundary representation. In: Proceedings of ACM Solid Modeling'03 Conference, New York, *ACM Press*, 298-303

[10] Martin, I., Peter, L. and Jack, S. (2005), Lossless compression of predicted floating-point geometry. *Computer Aided Des*, **37**(8), 869-877

[11] Diego, S. C. and Touradj, E. (2002), Coding of 3D virtual objects with NURBS. *Signal Processing*, **82**(11), 1581-1593

[12] Wang, Q. F., Yang, L., Huang, Y. b. and Wang, B. X. (2006). Model Simplification in Collaborative Design. *Journal of Computer-Aided Design & Computer Graphics*, **18**(1), 108-113

[13] Jacob, Z. and Abraham, L. (1977), A universal algorithm for sequence data compression. *IEEE Trans. on Information Theory*, **23**(3), 337-343

[14] Chrysafis, C. and Ortega, A. (1997), Efficient context-based entropy coding for lossy wavelet image compression. In: *Proceedings of IEEE Data Compression Conference*, Snowbird, Utah, 1997.241-250.

Jun Liu Jun Liu received his BSc and MSc degrees at Huazhong University of Science and Technology in 1999 and 2002, respectively. Now he is a Ph.D candidate at the CAD center of Huazhong University of Science and Technology. His works concentrate on computer graphics, and CAD/PDM system & integration.

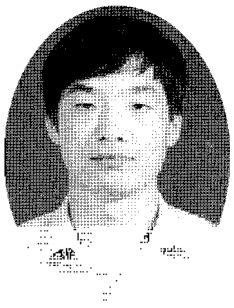
Zhengdong Huang Zhengdong Huang is a professor in the CAD center at Huazhong University of Science and Technology (HUST), China. He worked as a post-doctor in the ERC for Reconfigurable Machining Systems at the University of Michigan, Ann Arbor, from 1998 to 2002. He received his BS degree in Computational Mathematics from Wuhan University, his MS degree in Applied Mathematics from Zhejiang University and his PhD degree in Mechanical Engineering from HUST. His research interests include CAD/CAM, computer aided process planning, computer graphics and its applications in manufacturing.

Yunhua Liu Yunhua Liu received his Master Degree and Ph.D Degree from Huazhong University of Science and Technology in 1997 and 2001 respectively. He is currently an assistant professor in the CAD Center of Huazhong University of Science and Technology. His research interests include feature recognition, collaborative design and CAD/CAM integration.

Qifu Wang Qifu Wang is a professor in the CAD center at Huazhong University of Science and Technology. He received his Master Degree and Ph.D Degree from Huazhong University of Science and Technology in 1990 and 1993 respectively. His research interests include CAD/CAM/CAPP/PDM, compute graphics and intelligent design.

Liping Chen Liping Chen received his PhD degree from HUST in 1993. Now he is a professor in the CAD Center of HUST. His research interests include: spatial geometric constraint solving, multi-knowledge design optimizing, multi-body system kinematics & dynamics analyzing, and multi-domain physical system modeling and analyzing. He has published nearly one hundred research papers, and two monographs.

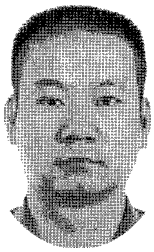
Jun Liu



Qifu Wang

Zhengdong Huang

Liping Chen



Yunhua Liu