グラフィクスとCAD　91－9
（1998．8．28）

UNCLET 表現による三次元高速ブラウジング

長澤 幹夫

超高速ネットワーク・コンピュータ技術研究所 (UNCL)

〒 185-8601 東京都国分寺市東恋が窪 1-280

E-mail: `m-nagasa@crl.hitachi.co.jp`

`http://www.iijnet.or.jp/uncl`

インターネット環境での三次元シーンデータの高速転送と高速レンダリングに有効な三次元要素モデル UNCLET を提案する。UNCLET は、三次元サーフェスデータおよびボリュームデータの標本点に対して、単位矩形の表面積を絶対値とした法線ベクトルと、表面内の基準方向に長さ成分を持つ接線ベクトル、および体積要素の高さ方向ベクトルの三本のベクトルから構成される。ベクトル要素を離散符号化することにより、描画表示時の冗長度を削減しデータ圧縮する。頂点接続を持たずに三次元形状を表示する量子化法線ベクトルでは、輪郭を平滑処理していたが、UNCLET は正確な長さと体積情報を持っており、テクスチャーマッピングを用いた高速ラスター表示だけでなく、プロッタ式の高解像度線画表示にも対応可能であり、CAD/CAM データとしても利用できる。また、時系列データの転送表示において、複数の時系列ベクトルを、ベジェ曲線の制御点を端点とするベジェ制御ベクトルに変換して固定長セル単位で転送表示するで、データ損失に耐えうる遠隔レンダリングが可能となる。
キーワード: データ圧縮, 解像度変換, 非構造格子, ベクトル可視化, ボリュームレンダリング.

Fast Browsing of 3-D Data Using UNCLET Representaion

Mikio Nagasawa

Ultra-high Speed Network and Computer Technology Laboratories (UNCL)

1-280, Higashi-Koigakubo, Kokubunji, Tokyo 185-8601 Japan

E-mail: `m-nagasa@crl.hitachi.co.jp`

`http://www.iijnet.or.jp/uncl`

UNCLET is a new vector primitive that represents any 3-D geometrical objects. The quantized vector structures of UNCLET are efficient for transmitting and rendering 3-D scenes on the Internet. In the 3-D vector represented animation, a moving sequence can be described compactly by the group of Bézier control vectors.
Keywords: Compression; Multiresolution Modeling; Irregular Grids; Vector Visualization; Volume Rendering.

1 Unified Vector Modeling

For remote visualizations with massive 3-D data, an effective compression and a fast rendering algorithm are required. There are several models that represent 3-D objects such as lines, NURBS, polygons, voxels, particles, and QNV.[1] The functionality for progressive transmission and Level-of-Detail control would be key features that the upcoming 3-D models must confront with. For that purpose, we have developed a new vector-based modeling that measures the characteristics of 3-D features: line length, surface area, and volume quantity. The traditional representations generally use 3-D geometrical models. While, UNCLET (UNiversal CloudLET) representation encodes 3-D objects as a set of vectors. Each vector represents line information, surface information and volume information. Line vector length is proportional to the tangential line length. Normal vector length is proportional to its surface area. Height vector length is used to calculate the volume of 3-D object element.

UNCLET representation can be converted from a set of polygons. The normal direction and the area of each polygon will be mapped to Normal vector $\boldsymbol{N}$. In other words, $\boldsymbol{N}$ is a cross product determined by the selected two edges of a triangle polygon. We can choose one edge of a polygon as tangential Line vector $\boldsymbol{L}$. The magnitude for Normal vector is assigned in accordance with the coverage area of corresponding polygon.

UNCLET could be interpreted as an approximation of the fan-shaped primitive with the opening angle Ψ at the selected polygon vertex. The key point is that we have to neglect some of polygon edges. With this assumption, the variety of polygons can be unified as a vector set with a unified data structure (see Fig.1).

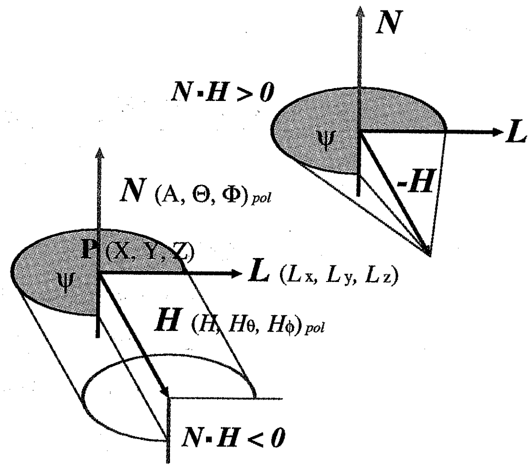

Fig 1: UNCLET vector attributions. The vector position $P(x, y, z)$ and the tangential Line vector $\boldsymbol{L}$ are specified in Cartesian coordinates, while the surface Normal vector $\boldsymbol{N}$ and the volumetric Height vector $\boldsymbol{H}$ are in polar coordinates. The volume classifications of *Cylinder-shape* and *Cone-shape* are labeled by the sign of $\boldsymbol{H}$.

In order to get UNCLET vectors from polygons, one can easily start from the uniform triangulation of surface with some relaxation algorithms.[2] A vertex position of a given polygon corresponds to the UNCLET position $P(x, y, z)$. The length of the cross product vector $\boldsymbol{N}$ is the area, and the direction is orthogonal to the polygon surface. The introduction of Line vector $\boldsymbol{L}$ is to get distinct line features in a rendered image. Defining the length of $\boldsymbol{H}$ with the value of the volume divided by the surface area, we can also apply UNCLET formats to model the volumetric objects.

There is another advantage of UNCLET representation that the rotation matrix which produces the 3-D configuration from the primitives located at the origin is described simply by the vector components as follows:

$$R^t = \begin{pmatrix} l_x & n_y l_z - n_z l_y & n_x \\ l_y & n_z l_x - n_x l_z & n_y \\ l_z & n_x l_y - n_y l_x & n_z \end{pmatrix} \quad . \qquad (1)$$

where

$(n_x, n_y, n_z) \equiv (\cos\Phi\sin\Theta, \sin\Phi\sin\Theta, \cos\Theta)$ and $(l_x, l_y, l_z) \equiv (L_x, L_y, L_z)/|L|$.

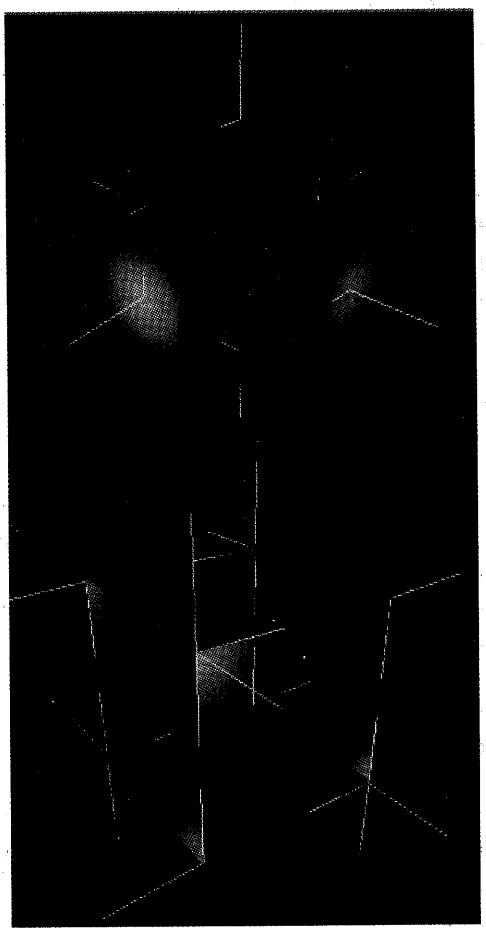

Fig 2: UNCLET representations of cubic surfaces: single model with Ψ=360 (upper) and double model with Ψ=90 (lower). Semi-transparent fan-shaped modelings are applied.

2 UNCLET Rendering

Each UNCLET is assumed to have the Gaussian opacity profile mapped with a fan-shaped texture. The opaque area is calculated by the length of Normal vector $\boldsymbol{N}$. The opacity value around the vertex position $P(\boldsymbol{x})$ is calculated as

$$\alpha(\boldsymbol{u}, \boldsymbol{x}) = \frac{1}{(\pi A)^{3/2}} \exp\left(-|\boldsymbol{u} - \boldsymbol{x}|^2/A\right) \quad , \qquad (2)$$

where $\boldsymbol{u}$ is the texture pixel position.

We apply a modified volume rendering algorithm to this UNCLET rendering.[3] The rendering is achieved by superposing these semi-

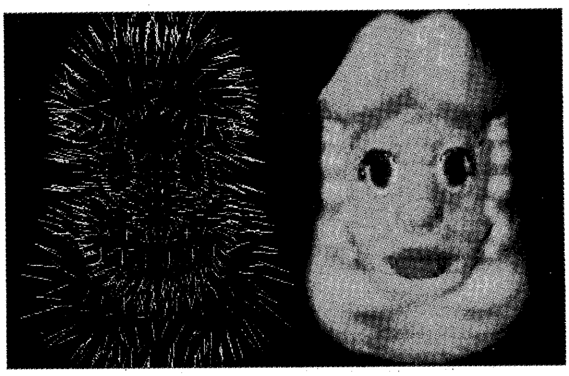

Fig 3: QNV rendering only with Normal vectors. Line drawing of vectors $\boldsymbol{N}$ (left). Texture mapping on orthogonal disks (right).

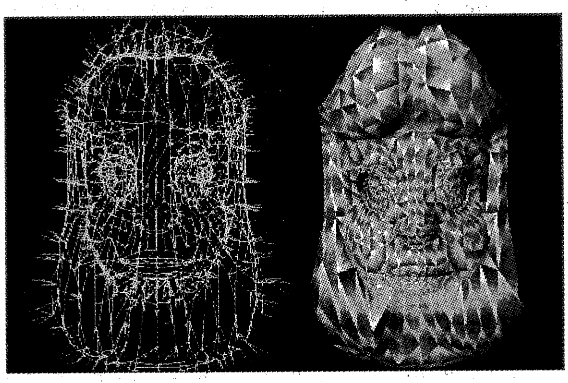

Fig 4: Line drawing of UNCLET vectors: $\boldsymbol{N}$ and $\boldsymbol{L}$ (left). UNCLET rendering with fan-shaped textures (right).

transparent fan-shaped textures instead of direct ray casting calculations. As in Fig.2, the fast and high quality rendering becomes possible. Anti-aliasing treatment is not necessary due to the smooth profile of the filtering function. In QNV rendering, the edge drawing is not expected and the neglect of some polygon edges causes sparse areas as in Fig.3. While, in UNCLET rendering, the wireframe rendering of surface data is possible by drawing Line vector $\boldsymbol{L}$. Rendering algorithm is able to control the sparse region by adjusting the effective area of opacity profile.

For a given 3-D primitive, UNCLET representation can conserve the integrated area and volume independent of the rendering algorithm. An irregular distribution of sampling points often causes error in the interpolations. But, with the smooth spatial profile of Eq.(2), robust rendering of 3-D data is possible for UNCLET.

Figure 3 and Fig.4 show the comparison of QNV and UNCLET representation for 3-D surface data. In UNCLET rendering, the distinct boundary feature appears at each sampling edge. While in QNV rendering that uses only Normal vectors, the result shows the diffuse surface without outline. The high resolution parts are similar in both cases.

It is possible to accelerate the UNCLET rendering by making use of texture mapping hardwares.[4] Most of the graphics libraries have texture map functions that specify color and transparency. For including semi-transparent surface effects, we use square polygon patches as surface elements perpendicular to Normal vectors. Fan-shaped transparent textures with any opening angle Ψ are mapped on them.

3 Effective 3-D Communication

If the independence of 3-D primitives is good enough, the model can be rendered progressively on a remote screen. Compared with the ordinary polygon data transfer that requires lossless transmission, the vector representation is robust

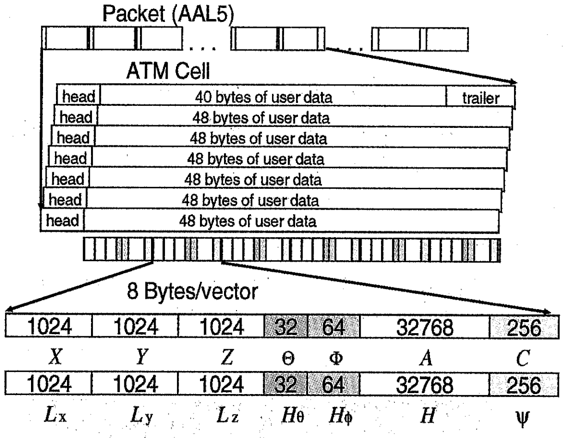

Fig 5: The elements of UNCLET vectors are discretized in a constant bit length to be efficient payloads on ATM cells.

against transmission errors. The polygons or finite elements data usually have the adjacency list of vertices. Thus, it is crucial that the network protocol guarantees lossless transmission. The UNCLET vector representation gives us a complementary information even with a lack of data. As the number of transferred vectors increases, the rendered image becomes clear more and more.

The UNCLET vector elements have the discretized attributions. The record length can be formatted in accordance with ATM cell size, quantizing the vector lengths and directions as in Fig.5. The record length of each UNCLET is $8\times2=16$ Bytes. The spatial resolution is 10 bit each and the direction of vectors are quantized in 6 degree resolution. With these formats, we implemented the quantized vector encoder, decoder and renderer that communicate on TCP/IP and Native-ATM network. In our network system of ATM-OC3 with 155Mbps bandwidth, a typical 3-D communication of 0.6 Mvectors/sec was achieved.

4 Bézier Vector Animation

In the field of numerical analysis or digital drawings, there are many smoothing methods that use the Bézier curves.[5] UNCLET vector representation is also easy to take advantage of this smoothing algorithm. If there is a series of motion vectors $\boldsymbol{V}(t)$, that is, UNCLET animation as in Fig.6, the trajectory of vector starting points and that of ending points can be fitted by two Bézier curves in space-time. The Bézier curve of starting points and ending points are reproduced using two control vectors $\boldsymbol{P}$ and $\boldsymbol{B}$. Given motion vectors are fitted with the Bézier curves in order to decode the 3-D motion at a time-step t.

A unit Bézier vector packet $[\boldsymbol{P},\boldsymbol{B},\boldsymbol{V},t]$ is composed of three quantized UNCLET vector components: $\boldsymbol{V}$ and the corresponding control vectors, $\boldsymbol{P}$, $\boldsymbol{B}$ in addition to time-step tag t.

$$
\begin{aligned}
\boldsymbol{P} &= (P_x, P_y, P_z) - (P_x', P_y', P_z') \\
\boldsymbol{B} &= (B_x, B_y, B_z) - (B_x', B_y', B_z') \\
\boldsymbol{V} &= (V_x, V_y, V_z) - (V_x', V_y', V_z')
\end{aligned} \quad . \qquad (3)
$$

The interpolated vector motion in the time interval of $[0,T]$ is calculated according to the expression:

$$
\begin{aligned}
v(t) = (&\alpha_x t^3 + \beta_x t^2 + \gamma_x t + V_x(t=0), \\
&\alpha_y t^3 + \beta_y t^2 + \gamma_y t + V_y(t=0), \\
&\alpha_z t^3 + \beta_z t^2 + \gamma_z t + V_x(t=0)) \\
- (&\alpha_x' t^3 + \beta_x' t^2 + \gamma_x' t + V_x'(t=0), \\
&\alpha_y' t^3 + \beta_y' t^2 + \gamma_y' t + V_y'(t=0), \\
&\alpha_z' t^3 + \beta_z' t^2 + \gamma_z' t + V_x'(t=0))
\end{aligned} \quad , \qquad (4)
$$

The coefficients are specified using the components of Bézier control vectors.

$$
\begin{aligned}
\gamma_{x,y,z} &= 3(P_{x,y,z} - V_{x,y,z}(t=0)) \\
\beta_{x,y,z} &= 3(B_{x,y,z} - P_{x,y,z}) - \gamma_{x,y,z} \\
\alpha_{x,y,z} &= V_{x,y,z}(T) - V_{x,y,z}(0) - \beta_{x,y,z} - \gamma_{x,y,z}
\end{aligned} \qquad (5)
$$

In this representation of 3-D animation, a moving sequence can be described compactly by the group of Bézier control vectors. In Bézier vector representation, smooth motion data can get a higher compression ratio. If the shape

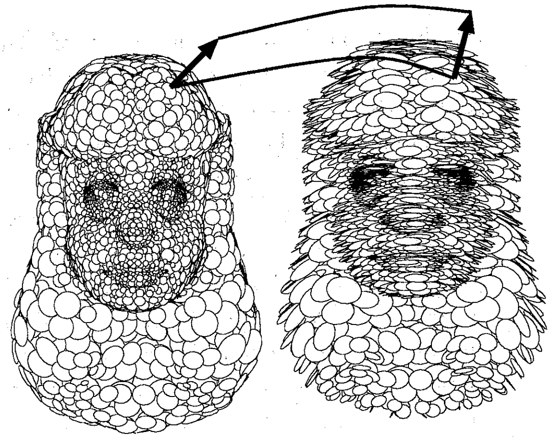

Fig 6: UNCLET vector morphing motion.

of vectors changes considerably along the time axis, we have to choose another new set of Bézier vectors. In Fig.7, two sets of Bézier vectors are used to represent the abrupt change of motions at t=10.

The redundancy of the 3-D motion is different at each 3-D element. In the Bézier vector transfer for UNCLET animations, the proper time interval could be assigned to each UNCLET individually. The slowly changing UNCLET is represented with the small number of Bézier vector sets, that means, the long time interval

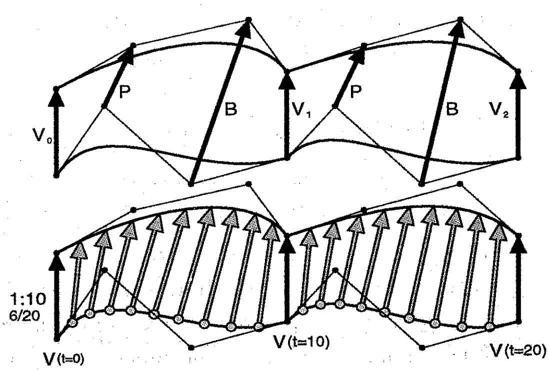

Fig 7: Bézier vector representation of 3-D motions. Spatiotemporal interpolations of 20 time-steps motion are driven with two sets of Bézier vector packet, resulting 6/20 compression ratio.

can be decoded with a single set of Bézier vectors. Because the Bézier interpolation can get the UNCLET state at an arbitrary time-step, a decoding of synchronized animation for plural UNCLET vectors is also possible.

Some explicit time marching algorithms in scientific simulation use the small time-steps for keeping accuracy of numerical integration. The change is also too small to make a dynamical animation. We can roughly estimate that the motion in ten time-steps in simulation is good for one time-step in animation. Therefore, the sparse sampling of simulation could make an animation data compact. The smooth interpolations from these sparse sampled data can reproduce and have little numerical error of the original 3-D simulation data.

Bézier control vectors are represented in quantized format similar to the UNCLET vectors. Each normal and tangential vector components, $\boldsymbol{N}$ and $\boldsymbol{L}$, could have the corresponding Bézier control vectors, respectively. The data format and communication protocol for Bézier vectors could be the same as for UNCLET transfer. The effectiveness for ATM network mentioned above is still valid. Even in the network with the narrow bandwidth, the small number of Bézier vector transmitted from the sender can reproduce the large number of UNCLET vectors in a synchronized way of decoding.

5 Concluding Remarks

The Bézier interpolation of UNCLET vectors means not only the vectors themselves, but also that of the surface and volume data represented by UNCLET. In this way, the unified representation of 3-D models leads to the unified 3-D animation format. The independency and generality of UNCLET vector elements make it possible to apply this representation to 3-D flow visualizations in CFD or the 3-D motion of finite elements in Computational Mechanics. The vector representations, not polygons, are very natural and easy to apply for those scientific applications.

Acknowledgements

The author would like to thank Mr.N.Seoka and Mr.K.Akagi for assisting in the implementation of UNCLET browser.

References

1. See, [www.iijnet.or.jp/uncl/GhostSpace/PSB98paper/psb98.html] ()
2. G. Turk, *Computer Graphics* **26**, 55-64 (1992)
3. M. Nagasawa and K. Kuwahara, *Scientific Visualization of Physical Phenomena* (ed. N.M.Patrikalakis, Springer-Verlag) 589-605 (1991)
4. K. Akeley, *Computer Graphics Proceedings* (ACM SIGGRAPH) **27**, 109-116 (1993)
5. J. D. Foler and A. VanDam, *Fundamentals of Interactive Computer Graphics,* (Addison-Wesley) (1982)