Visualization of molecular orbital metamorphosis according to interactive molecular skeletal transformation

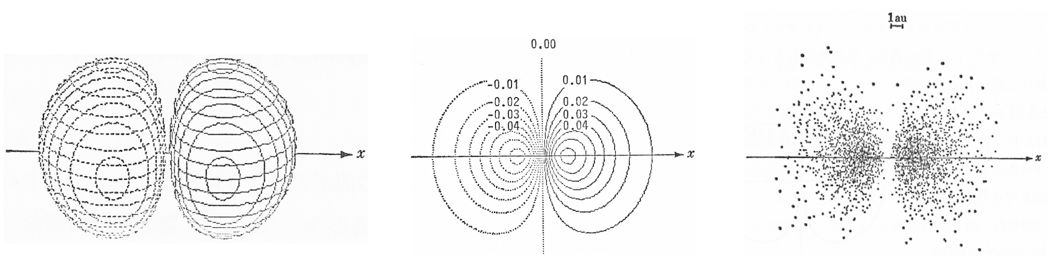
Takatoshi NAKA*, Mamoru ENDO**, Masashi YAMADA**, Shinya MIYAZAKI**, Yasuyo HATANO **, and Shigeyoshi YAMAMOTO***
** Graduate School of Computer and Cognitive Sciences, Chukyo University*
*** School of Computer and Cognitive Sciences, Chukyo University*
101 Tokodachi, Kaizu-cho, Toyota-city 470-0393, Japan
****Faculty of Liberal Arts, Chukyo University*
101-2 Yagotohonmachi, Showa, Nagoya, Aichi, 466-8666, Japan

Abstract. This paper realizes visualization of the metamorphosis of molecular orbital according to interaction of molecular frame distortion. Chemists are given a great help in analyzing the mechanism of chemical reaction. Molecular orbital is represented by a voxel data set that is obtained by solving differential equations numerically, and is transparently rendered as a cloud. To display continuous results according to interaction, a novel method is introduced. Key volume data sets are sampled according to the bending of the molecular skeleton and a pair of key volume data sets is blended in the rendering process.

1. Introduction

A molecular orbital or electron density is a physical quantity which serves as the foundation for describing an electron's behavior. It is important to embody them for analyzing the mechanism of chemical reactions. Also, it is useful for chemists to see the metamorphosis of an electronic state in order to understand its chemical reaction.

With the increment of the quantum number, it is difficult to understand its electron's appearance from an orbital function formula. Comparatively simple structures can be illustrated by two dimensional diagrams as used in textbooks of quantum chemistry. For example, contour surfaces which have equivalent value on them are drawn as 3D wire-



(a) (b) (c)

Figure 1. The $2p_x$ atomic orbitals of a hydrogen atom is expressed by several kinds of diagrams above, (a) three dimensional wire-framed contour surface (b) two dimensional contour lines on an intersectional plane (c) density expression by dots.

frame models. Contour lines are drawn on an intersectional plane, and density value is expressed by the degree of concentrated dots as shown in Figure 1 [1]. These can be a great help when we are trying to understand the outline of electron's behavior, but molecular orbitals in general, are too complex to be expressed by these illustration diagrams. Transparent volume rendering technique in computer graphics gives a solution to display three dimensional value distributions.

We realize cloud's metamorphosis in real-time, which can be controlled by direct manipulation of molecular bonds. Volume data sets are generated when a part of a molecular bond is bended. The mixed state between a pair of volume data is properly rendered in real-time. This generates an electron state in any position along a certain path and realizes seamless display.

Volume data is calculated with Gaussian 98, a molecular orbital calculation software [2]. A volume data set can store only one molecular orbital when molecular bonds are fixed. To realize manipulation more freely, large amounts of volume data have to be used. Volume data is drawn as a set of piled texture images for real-time processing. We have investigated the rendering performance with typical graphic cards.

2. Rendering of Molecular Orbital

2.1 Creating volume data

An electron's behavior is explained by an orbital function's physical quantity, and is defined as a continuous function of special coordinates. The whole electronic structure of a molecule is expressed by a combination of these functions.

Values of an orbital function can be positive or negative, and to the second power, this equals probabilities of electron's existence (electron density) at that position

If the value of electron density is higher and proportionally makes transparency lower, the distribution of electron density can be expressed like cloud-like figures. The higher value of electron density means that the cloud is heavy, because it is hard to penetrate light and makes its own color visible.

The value of orbital function or electron density is continuous in reality, but here we use voxel-format data constructed by discrete values which are obtained by constant interval sampling. We adopt CUBE file which is standard in theoretical chemistry and can be generated by Gaussian 98 [2].

2.2 Outline of rendering procedure

The ray casting method is a fundamental way to render transparent volume [3]. It integrates intensity values considering transparent elements along a ray which goes through the pixel and eye position. Although, it makes exact results, procedure is out of graphics acceleration.

A combination of alpha blending and texture-mapping is fundamentally the only way to render transparent volume in real-time at present. Although, this method makes poorer results in some cases, it is not a significant problem for objects when intensity values are continuous such as a cloud.

Electron density values are basically used as alpha blending values in transparent rendering, but they should be modulated to make suitable results because the standard 32bit color environment is still not enough for transparent rendering. Alpha value $\alpha(x,y,z)$ is derived from electron density $\gamma(x,y,z)$ at every point (x,y,z) in the cubic lattice of volume data as follows.

$$\alpha(x,y,z) = k_1 \cdot \gamma(x,y,z)^{k_2} \quad (1)$$

Here, regulated modulation parameters k_1 and k_2 are constant coefficient and constant powers, respectively. They are interactively and continuously changed in real-time response.

2.3 Rendering Method for Wide Use

To render interpolated results between a pair of volume data in real time, the function of three dimensional texturing and programmable shader is now available. However, it limits graphics hardware executable in real time. Therefore, for wide use, we developed a simple way only with ordinal two dimensional texture mapping function.

Figure 2 illustrates generation of multiplex texture planes from voxel data and selection among them. Three kinds of texture plane sets are generated parallel to voxel faces. The best one in which the angle between the normal vector and the view-line vector is the smallest, is selected for each rendering.

If the size of voxel data is Nx Ny Nz, and the direction of the planes are parallel to the xy plane, the number of planes equal Nz, and the alpha value of pixel (x, y) in texture plane i $\alpha_i(x, y)$ stores the following expression.

$$\alpha_i(x, y) = \alpha(x, y, i) \quad (i = 1, 2, \cdots N_z) \qquad (2)$$

2.4 Drawing molecular skeletons

Many kinds of atoms are identified by independent color spheres in rendering. CUBE format has no bonding information, but if the distance between any neighboring atom is less than the sum of its atomic radius, the bond between them is drawn.

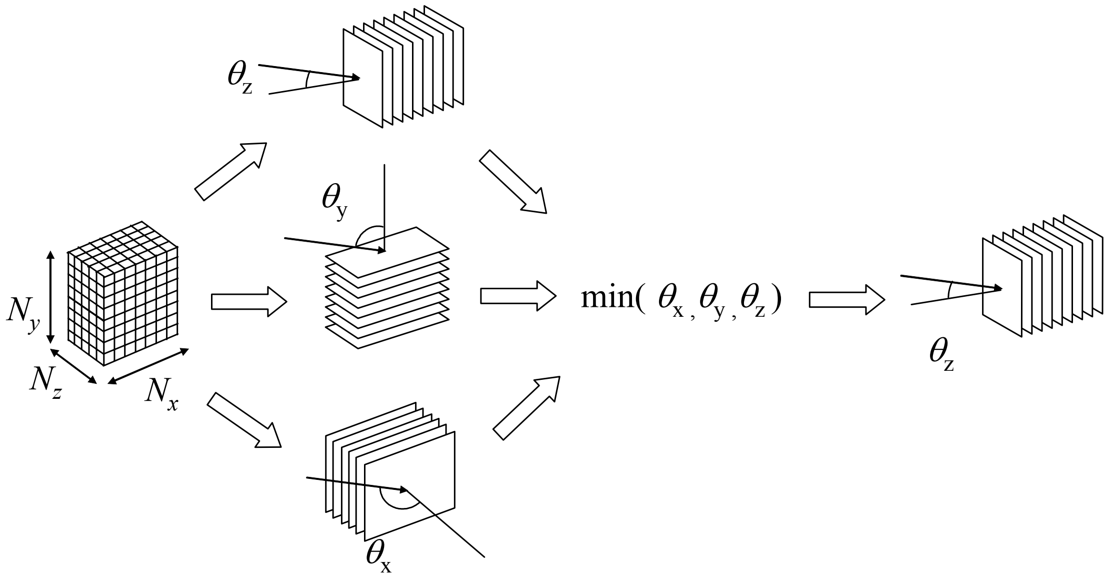


Figure 2. Selection among three directional texture plane sets in the rendering process. Only one texture plane set is selected in the rendering procedure. The selected one has the smallest angle between the normal vector face and the view-line vector.

3. Molecular Orbital Cloud Metamorphosis

The molecular orbital can be drastically metamorphosed by transforming molecular skeletons intentionally. It becomes an effective way for chemists to confirm or predict chemical reaction phenomena. Volume data of a molecular orbital has to be recalculated once the molecular skeleton is modified, but it takes much computation time for real-time processing. To animate it interactively, we limit molecular skeleton's transformation along a certain path and generate the interpolated positions state by interpolating some key volume data.

3.1 Volume data generation based on interpolation

Figure 3 shows key volume skeleton states and interpolated ones when a part of the skeleton rotates in an identical plane. Interpolated volume is generated as interpolation between the neighboring two key volumes. To realize it exactly, transparent texture images have to be interpolated for each texture image in a pile, but again it takes too much computation time for real-time processing. Therefore, we approximate the interpolation only by a combination of fundamental alpha blending functions.

Expression 3 gives alpha value α_i from alpha values α_{Ai} and α_{Bi} which are the values of voxel data A and B. It is applied in every pixel in every image.

$$\alpha_i = (1-k)\alpha_{Ai} + \alpha_{Bi} \qquad (3)$$

Here, the interpolation ratio between them is *1 – k : k*. And expression 4 is a recurrent formula to obtain integrated pixel density d_i from the background to image *i* that is derived from d_{i-1}.

$$d_i = d_{i-1}(1-\alpha_i) + \alpha_i \qquad (4)$$

They are combined into the following expressions.

$$d_i = d_{i-1}(1-((1-k)\alpha_{Ai} + k\alpha_{Bi})) + (1-k)\alpha_{Ai} + k\alpha_{Bi} \qquad (5)$$

It is impossible to construct the above procedure within a combination of fundamental alpha blending functions

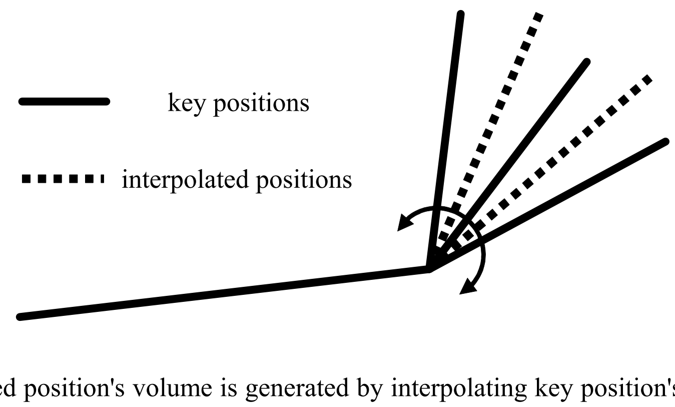


Figure 3. Interpolated position's volume is generated by interpolating key position's volume. The volume data on the key position is calculated beforehand.

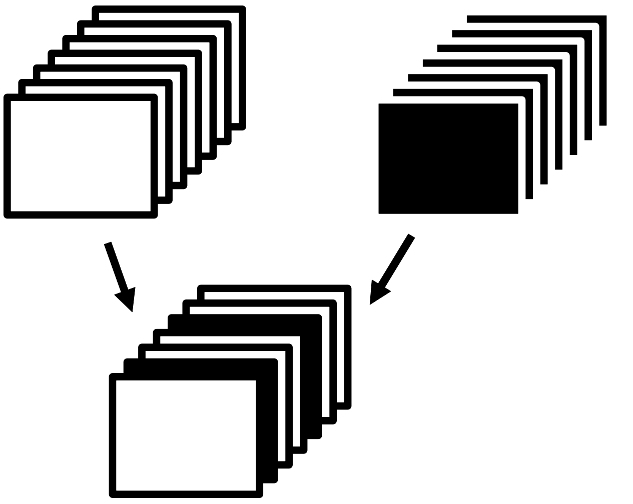


Figure 4 . The interpolated state is generated by sequentially alternatively rendering two key position's volume data sets.

3.2 Approximation in rendering interpolated states

If $\alpha_{Ai} = \alpha_{Bi}$ is supposed, alpha values in each pixel is close between neighboring key volumes, and less error is considered, expression (5) is rewritten as follows.

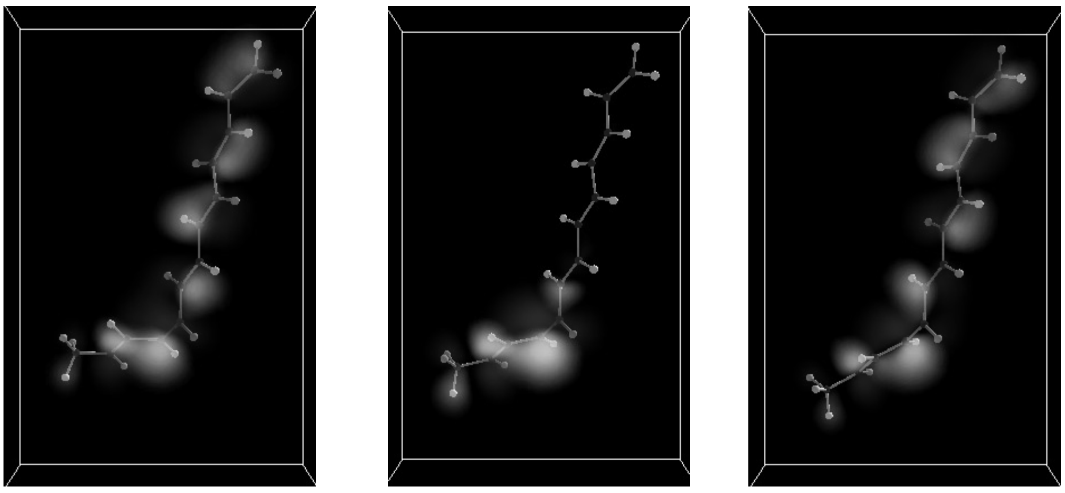
$$d_i = d_{i-1}(1-\alpha_{Ai}) + (1-k)\alpha_{Ai} + k\alpha_{Bi} \quad (k < 0.5) \qquad (6)$$

$$d_i = d_{i-1}(1-\alpha_{Bi}) + (1-k)\alpha_{Ai} + k\alpha_{Bi} \quad (k > 0.5) \qquad (7)$$

The right side consists of three simple terms which are products of pixel density by alpha values and by constant value *1 – k* and *k*. They can be performed by three sequences of alpha blending functions. Figure 4 shows an outline of these procedures.

3.3 Example of implementation

We chose protonated Schiff base of retinal (PSBR) for an example of implementation [4, 5]. PSBR molecules are included in rhodopsin which is a kind of chromoprotein in



HOMO ($\theta = -70°$) HOMO ($\theta = -90°$) HOMO ($\theta = -110°$)

Figure 5 . Signs that a Electronic state of HOMO orbit changes near 90 degrees

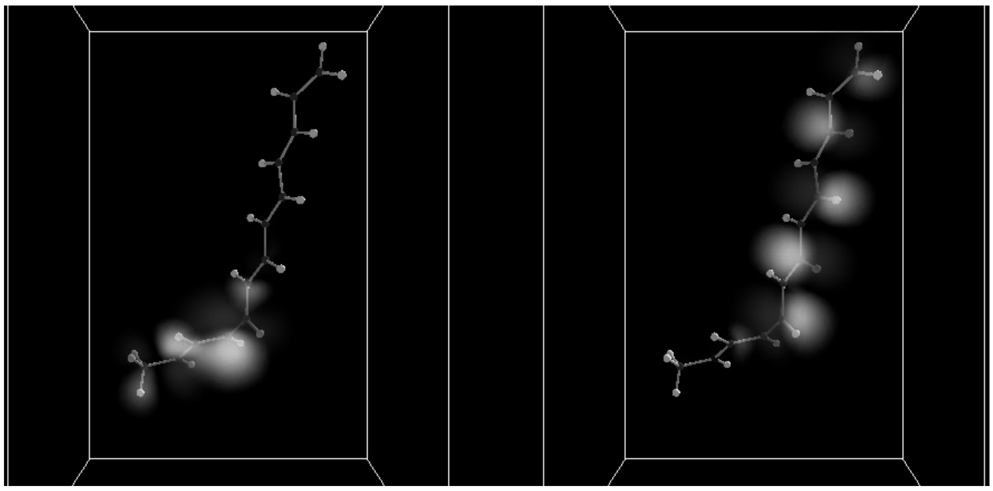


Figure 6. Two kinds of molecular orbitals of PSBR rendered as a cloud

amphiblestrodes in our eyes that makes us feel object's colors. To study PSBR is important for vision analysis.

Fifteen key volumes are generated by using Gaussian 98 in which the dihedral angle of a certain bond (C13=C14) is at -180, -150, -120, -110, -105, -100, -95, -90, -85, -80, -75 -70, -60, -30, and 0 (degrees). Angle intervals are in closer around -90 degrees in which molecular orbital changes drastically (Figure 5). White clouds express the negative of a molecular orbital's value. More than two kinds of molecular orbital are displayed at the same time. Figure 6 shows a pair of orbitals, HOMO and LUMO. Their movements are seamless and connected during interactive operation.

4. Performance Experiments

To realize freer operation or higher resolution display, it is necessary to increase the number of key volumes. The capacity of high-speed video memory in graphics cards can be an index for the performance, but recent graphics cards have a novel memory administration function such as AGP (accelerated graphics port) that enables using main memory as texture memory by high-speed memory transitions. Therefore, we measure real performance to recent existing standard graphics cards.

4.1 Memory quantity necessary for texture images

If the size of 128×128 and 4byte (32bit) color texture image is used without any compression, and the number of key volume is K, then the quantity necessary for texture images is calculated by the following formula.

$$128^2 \times (Nx + Ny + Nz) \times 4 \times K \text{ (byte)} \qquad (8)$$

In relation to the case in section 3.3, it becomes about 18.3×K (MB) = 275MB (K=15).

4.2 Performance results

Rendering performance is measured in the following two kinds of hardware environments that are standard ones including low price editions of popular graphics cards.

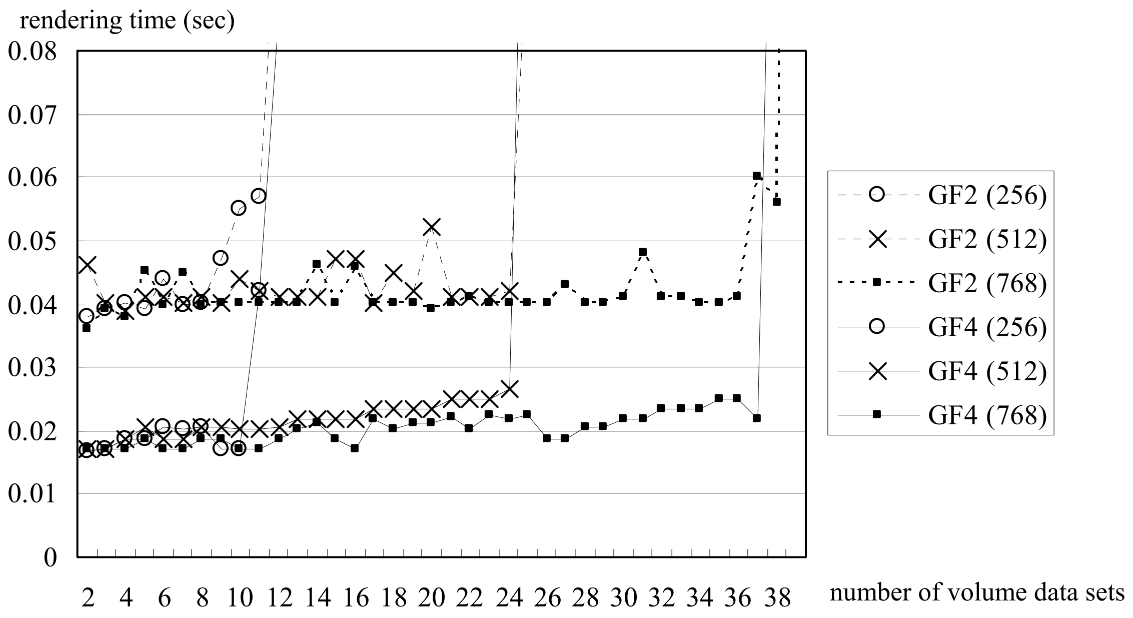


Figure 7. Rendering time depending on the number of volume data sets.

(1)CPU : Pentium III 600MHz
memory band-width : 1.06GB/sec (PC133)
graphics chip : GeForce2 MX(32MB)

(2)CPU : Athlon XP 1900++(1.6GHz)
memory band-width : 2.1GB/sec (PC2100)
graphics chip : GeForce4 MX440(64MB)

Figure 7 shows the experimental results of them. It indicates fundamental dependence in the amount of main memory. Although, temporary increments of rendering time can be seen in some cases that are considered being due to memory swapping, rendering time is stable as well as all the texture data being stored in the main memory. Fifteen fps and thirty fps drawings are guaranteed in case (1) and (2), respectively. In the case of 3.2, in which 275MB is necessary for texture memory, 512MB is enough for main memory for real-time processing.

5. Conclusions

In this paper, we developed a way to display molecular orbital metamorphoses according to interactive transformation of molecular skeletons. Interpolation between a pair of transparent volumes is realized within real-time rendering. It is believed that our system will serve as an effective tool for chemical analysis.

Acknowledgments

We would like to thank Professor Teruo Fukumura of Chukyo University for his encouragement and assistance. Special thanks to Marvin Lenoar, a great English teacher and for English assistance on this project. This research was supported in part by the grant-in-aid for Private University High-Tech Research Center and grant-in-aid for Young Scientists from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] T. Helgaker, P.Jorgensen, and J. Olsen: Molecular Electronic-Structure Theory, John Wiley & Sons, 2000.

[2] M. J. Frisch et al.: Gaussian 98 (Revision A.5), Gaussian, Inc, 1998.

[3] S. Handa, T. Takada: Visual Computing: Integrating Computer Graphics with Computer Vision, pp.313-328, 1992.

[4] T. Kobayashi, T. Saito and H. Ohtani, Nature, 414, pp531-534, 2001.

[5] S. Yamamoto, H.Wasada, and T. Kakitani, Theochem, 451, pp. 151-162, 1998.