

# 分子骨格操作に伴う分子軌道変化の等値面 リアルタイム描画システム

中 貴俊 <sup>a\*</sup>, 山本 茂義 <sup>b</sup>, 秦野 やす世 <sup>c</sup>, 遠藤 守 <sup>c</sup>, 山田 雅之 <sup>c</sup>, 宮崎 慎也 <sup>c</sup>

<sup>a</sup> 中京大学大学院情報科学研究科, 〒 470-0393 愛知県豊田市貝津町床立 101

<sup>b</sup> 中京大学教養部, 〒 466-8666 愛知県名古屋市昭和区八事本町 101-2

<sup>c</sup> 中京大学情報理工学部, 〒 470-0393 愛知県豊田市貝津町床立 101

\*e-mail: naka@om.sist.chukyo-u.ac.jp

(Received: December 14, 2006; Accepted for publication: May 8, 2007; Published on Web: July 6, 2007)

核座標の変化による分子軌道の変化をリアルタイム観察できる分子軌道可視化システム MOOTIC を開発した。このシステムでは、テクスチャマッピングの混合処理による高速なレンダリング方法を適用することによって、変化する分子軌道をリアルタイムに表示できる。あらかじめ、核座標が変化したときの一連の任意数の分子軌道データ（ボクセルデータ）をキーフレームとして構成しておき、次にキーフレーム間の中間状態をテクスチャマッピングの内挿を高速に処理して出来る画像群として表示させることで、一組のキーフレームから中間フレームをリアルタイムに表示するものである。描画を高速化するために、新しい手法を開発した。Microsoft Windows 環境下でのシステム適用例として、水素分子の  $2\sigma_g$  仮想軌道の内部節領域の表示、核間距離の変化による一酸化炭素の分子軌道の変化、レチナルプロトン化シッフ塩基中の二重結合の回転に伴う分子軌道の変化を示す。MOOTIC システムには、高速化等値面表示法のほかに、各種の 3D 表示方法を取り入れており、これらの表示法を即座に切り替えて表示することができる。本システムは分子軌道の特徴付けに重要な役割を持つ内部節領域の数や形状を観察し、解析することに有効である。

キーワード: 分子軌道, 等値面, サイエントフィック・ビジュアライゼーション, ボリュームレンダリング, リアルタイム CG

## 1 はじめに

分子スペクトルや化学反応の研究において、分子軌道の描画は必須の研究手段となっており、PDS や市販ソフトウェア [1] が普及している。分子軌道の可視化は、分子軌道の分類を容易にし、分子スペクトルの解析に貢献する。我々は最近の論文 [2] で、特に節領域（または節面）を数えることが分子軌道の特徴づけに役立つことを示した。たとえば、エチレンの Rydberg 軌道を、その節領域数によって分類することができることを示した。また、節領域計数は CASSCF 計算を行う際の active 軌道の選択にも有効である。

節領域の把握には分子軌道の可視化が重要であるが、内部節領域の可視化の手法については、これまであまり議論されてこなかった。たとえば、水素分子の  $1\sigma_g$  占有軌道は内部節面を持たないが、 $2\sigma_g$  仮想軌道は節面を持ち、節領域計数により両者は容易に区別できる。分子軌道の可視化には、従来、断面の等高線や鳥瞰図による表現、等値面のワイヤフレームによる 3D 表示等 [3–5] が用いられてきた。しかしながら、 $2\sigma_g$  は内部節領域ゆえに、既存の手法では描画が困難であり、新しい描画方法が求められる。内部節領域の可視化が今回のプログラムの開発の動機のひとつである。

一方、化学反応の解析には、核座標の変化に伴う分

子軌道の変化を知ることが重要である。この変化を表示するためにこれまで用いられてきた方法は、いくつかの核座標で分子軌道をあらかじめ描画してファイルに蓄積しておき、これらをまとめて動画にするものである。しかし、この場合、リアルタイムに核座標を変化させることはできない。分子軌道変化のリアルタイム表示も本論文で扱うプログラム開発の目的のひとつである。

以上の2つの目的を共に実現するため、我々は分子軌道描画プログラムシステム MOOTIC (Molecular Orbital Observation Tool with Iso-surface and Cloud) と名づけるプログラムを開発した。リアルタイム表示には、描画の高速化が欠かせないが、我々は新しい高速化アルゴリズムを開発し、この MOOTIC プログラムに実装したので、これについても述べる。

過去に、我々はリアルタイム CG の技法におけるテクスチャマッピングを利用したボリュームレンダリング手法を用いて、多原子分子の分子骨格中の特定の原子を、あるパスに沿って移動させた際の分子軌道変化のリアルタイムでの表示を実現し報告した [6, 7]。特に、ボリュームデータを構成する多層テクスチャ面を、二つのボリュームに関して交互にレンダリングすることにより、OpenGL [8] の基本的なテクスチャマッピング関数のみを用いて両ボリュームデータの補間状態の表示を可能とした。この方法は GPU (Graphics Processing Unit) が基本的なテクスチャマッピングの機能を有してさえいればリアルタイムでの表示が可能であり、通常の PC での利用が可能となる点で有利であった。この表示方法では輪郭が明確でないという難点があるが、MOOTIC では等値面表示と自由な表示切替によって克服している。

厳密な等値面表示においては、Marching Cubes 法や四面体メッシュを用いた方法が一般的に用いられているが、これらの手法は基本的に単一のボリュームデータを対象としたものであるため、本研究で目的とする分子軌道変化のリアルタイムな等値面表示への適用は容易ではない。

そこで本研究では、ボリュームレンダリングにおいて表示する等値面をテクスチャ画像として生成し、テクスチャマッピングを利用したボリュームレンダリング法により、補間状態の等値面のリアルタイム表示を実現する方法を取った。

内部節領域の把握には、新しく開発した上記の等値面表示に加え、前回報告した雲表示 [6, 7]、あるいは Marching Cubes 法による面表示などの描画方法を

切り替えて表示できるようにすることが有効である。MOOTIC プログラムでは、これらの描画方法を円滑に切り替えることが可能になっており、利用者の目的に最適な方法で分子軌道を観察できる。

## 2 分子軌道描画システム MOOTIC

本章では、分子軌道可視化システム MOOTIC について、システムで使用する分子軌道データ、本研究で開発した描画処理手法、システムの概要の順で説明する。

### 2.1 使用データとキーボリューム

分子軌道は、空間中の関数として定義される量であるが、実際には、非経験的分子軌道プログラムにより空間中の格子点におけるサンプル値 (ボリュームデータ) を求めることになる。ここでは、分子軌道は主に Gaussian 98 [9] プログラムにより求め、ボリュームデータは cube 形式でファイルに格納したものを利用した。

本研究では対話操作により分子骨格を変形させ、その状態の分子軌道の電子状態を円滑に観察することを目的としている。しかしながら、分子骨格が変化すれば分子軌道は再計算する必要があり、さらにあらかじめ用意できる軌道データの数は有限である。そこで、本システムでは、分子骨格の動きを単一のパス上に限定し、そのパス上の代表となる離散点のみをあらかじめ軌道データとして作成しておき (キーボリューム)、この作成されたキーボリュームを利用して分子軌道を可視化する。軌道データが存在するキー位置以外においては、本研究で提案する近隣2つのキーボリュームデータのみを用いた補間表示法によりリアルタイムな対話操作を実現している。

### 2.2 分子軌道の描画手法 (補間状態描画手法)

本描画システムでは核座標の変化における分子軌道の変化をリアルタイムで可視化する手法として、電子雲により分子軌道全体の状態の変化を可視化する手法と等値面によりある値における分子軌道の状態の変化を可視化する手法を備えている。両手法ともテクスチャマッピングの混合処理を利用した描画手法であり、前者の電子雲での表示については、前回の報告 [6, 7] による手法を用いて本システムに導入した。さらに、

本研究では新たに後者の手法を考案し本システムに導入している。

以下では新たに考案した等値面の高速表示による分子軌道の状態の変化を可視化する手法について述べる。

ポリウムデータの等値面表示には、前述の通り Marching Cubes 法や四面体メッシュを用いた方法などが考案されており、これらのさらなる研究も進められている [10-12]。さらに、PC の性能の進歩に伴ってリアルタイム処理可能なケースは広がりつつあるものの、これらの手法は基本的に単一のポリウムデータを対象としたものである。これに対し、本研究で目的とする等値面表示を実現するためには、複数のポリウムの補間状態を表示する必要がある。従来の手法では補間状態のポリウムデータを生成するための計算時間はポリウムデータの解像度の 3 乗に比例するため、高解像度データに対してリアルタイム処理が著しく制限される。よって、補間状態の等値面をポリウムデータを求めることなくリアルタイムで表示するためには、異なる等値面の補間状態を求めるアルゴリズムを確立する必要があるが、実現は容易ではない。

そこで本研究では、テクスチャマッピングを利用したポリウムレンダリングに対して、等値面においても電子雲表示と同様のリアルタイム性を維持する補間手法を実現した。この手法により表現された等値面を以下では等値面(雲)と呼ぶこととする。

今回のテクスチャ画像において表示される等値面(雲)は一般的なポリウムレンダリングとは異なり等値面(一定値のみ)の表現であるため、生成されたテクスチャ画像面における分子軌道は等値面的一部分であり透過度( $\alpha$ )の値はすべて一定(不透明)と考えることができる。そこで背面から  $i$  番目のテクスチャ画像の画素値を  $d_i$ 、 $i-1$  番目の画素値を  $d_{i-1}$ 、近隣の 2 つのキーポリウム  $A, B$  の画素値をそれぞれ  $A, B$  とし、 $A$  および  $B$  の内分比が  $k: k-1$  であるとき、背面との透過度をととして考えると、画素値  $d_i$  は以下のような式で求められる。

$$d_i = d_{i-1}(1 - \alpha) + \alpha((1 - k)A + kB) \quad (1)$$

これは「 $A$  および  $B$  のテクスチャ画像を  $k: k-1$  の比で混合処理したもの」と「背面から  $i-1$  番目まで透過処理したもの」とを透過処理することで実現できるが、通常ポリウムレンダリングは背面から順に描画する必要がある。しかしながら、OpenGL[8]の基本的な混合処理においては現在描画されている領域と新しい描画オブジェクトとを任意のブレンド比で混合処理

Table 1. Constants (arguments) for the glBlendFunc function

Constants	(fR, fG, fB, fA)
GL_ZERO	(0, 0, 0, 0)
GL_ONE	(1, 1, 1, 1)
GL_SRC_COLOR	(Rs/kR, Gs/kG, Bs/kB, As/kA)
GL_SRC_ALPHA	(As/kA, As/kA, As/kA, As/kA)

することしかできない。実際、OpenGL[8]のサブルーチン glBlendFunc() では引数を以下のように与えて使用する。

```
void glBlendFunc(GLenum sfactor, GLenum dfactor);
```

ここで sfactor には新しい色の、dfactor には現在の色の計算方法を指定する。具体的な定数値の一部を Table 1 に示す。

このため、この処理を OpenGL の基本的な混合処理のみを用いて行うには、あらかじめ  $A$  および  $B$  のテクスチャ画像を  $k: k-1$  の比で混合処理した画像情報を用意しておく必要があり、リアルタイム処理は不可能である。

ここで、 $\frac{\alpha(1-k)}{1-\alpha k} = \alpha'$ 、 $\alpha k = k'$  とおくと式 (1) は次のように変形できる。

$$d_i = (d_{i-1}(1 - \alpha') + \alpha'A)(1 - k') + k'B \quad (2)$$

これは背面から  $i-1$  番目まで透過処理をしたものに  $A$  のテクスチャ画像を  $\alpha': \alpha'-1$  の比で混合処理し、さらに  $B$  のテクスチャ画像を  $k': k'-1$  の比で混合処理することを表している。

上記式 (2) の処理手法は、OpenGL の基本的な混合処理のみを順に用いて実現できるため、幅広い GPU で高速処理が可能である。

以下の Figure 1 に本手法による補間表示結果例を示す。

また、等値面での表示結果を比較対象としてあわせて示す。いずれも 2 つのキーポリウムデータによる内挿状態の表示であるが、等値面(雲)表示においてはテクスチャの混合処理により実現した表示であり、等値面表示においては、分子軌道データであるキーポリウムを内挿した線形補間による計算結果を基にして表示したものである。

Figure 1 に示したように両結果において相違ない結果が得られた。

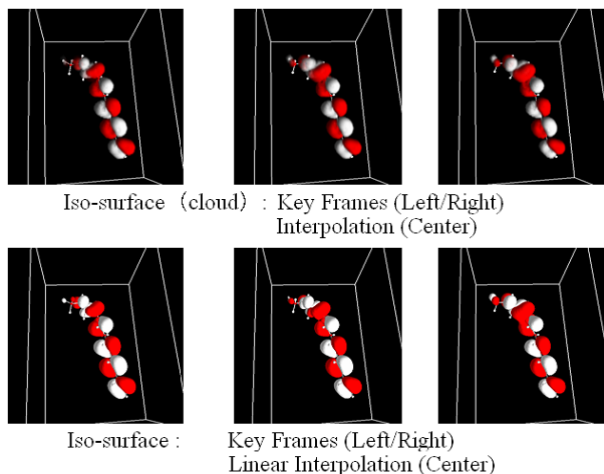


Figure 1. Comparison of the present interpolation method (upper) and the previous iso-surface interpolation method (lower).

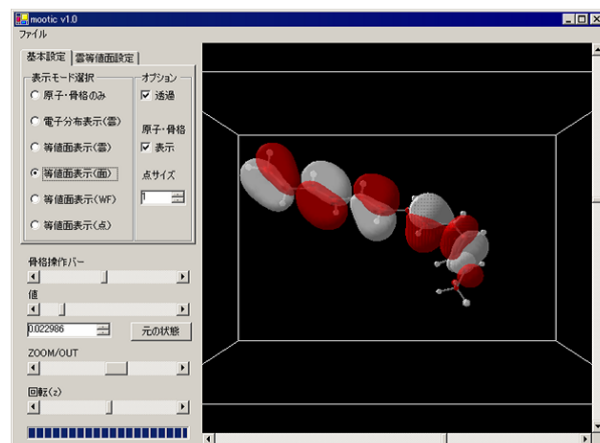


Figure 2. The console of MOOTIC (Windows version)

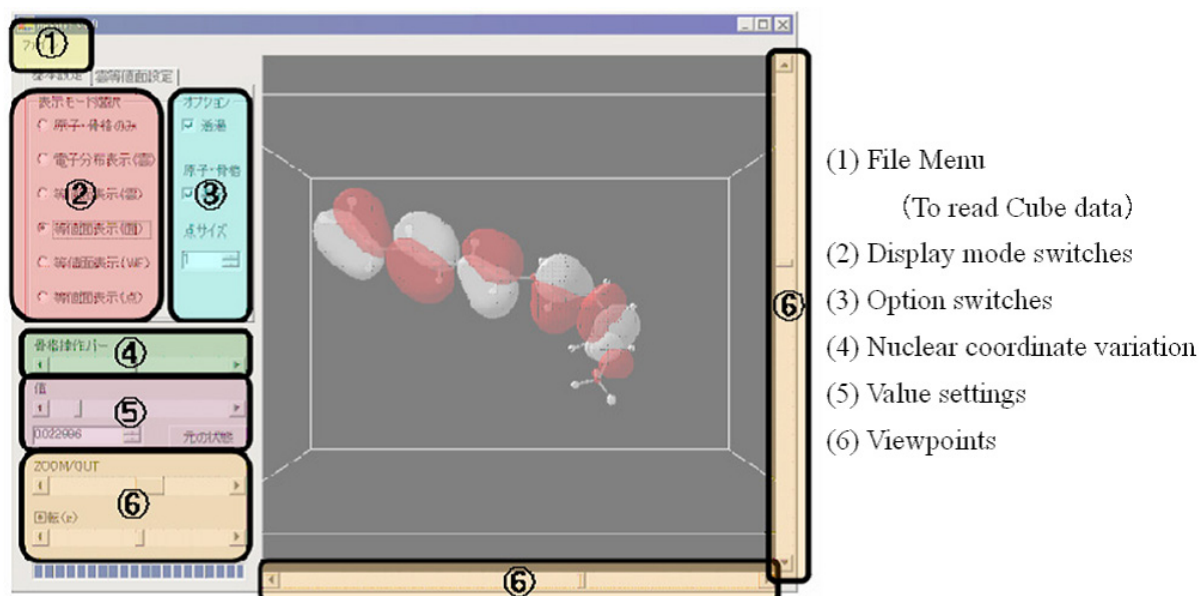


Figure 3. Menu items of MOOTIC

## 2.3 システムの概要

本研究で我々は、前節において記述した等値面(雲)の表示アルゴリズムを搭載した、核変化に伴うMOの変化を可視化するためのプログラムMOOTICを開発した。MOOTICでは、等値面(雲)による表示を含め内部領域の把握など観察目的に適した5種類の表示を自由に切り替えることができ、視点もマウスを利用し、自由に分かりやすく変更することができる (Figure 2)。

このプログラムは OpenGL[8] を基礎としており、可搬性が高い点も特長のひとつである。現在、Windows、Linux 環境下で稼働している。

開発言語は C++、描画には Windows、Linux 共に OpenGL を使い、Windows 版での GUI 操作部分については Microsoft .NET FrameWork[13] を用いて開発した。

以下では MOOTIC で実現されている表示法と現在 (Ver.1.0) における対話操作手法の詳細について述べる。

### 2.3.1 電子分布表示(雲)

透過処理を利用して、電子分布を雲のように表現した表示法である。この方法については既に報告している [6, 7]。多原子分子の分子骨格中の特定の原子を、あるパスに沿って移動させた際の分子軌道変化をリアルタイムで表示できる。

### 2.3.2 等値面表示(雲)

この方法は本研究で開発した新しい描画方法である。電子分布表示(雲)が分子軌道全体を表現した表示であるのに対し、等値面表示(雲)は任意の等値面を表示する。一般に、ボリュームデータの等値面表示は面の構成処理に計算時間がかかるが、等値面表示(雲)はテクスチャマッピングを利用したボリュームレンダリング手法を利用しているので、核変化に伴う MO の変化をリアルタイムで描画できる。ただし、前処理に若干時間を要する。

### 2.3.3 等値面表示(面, ワイヤフレーム, 点)

上記の等値面表示(雲)とは異なり、与えられた軌道データから等値面を計算し描画する。そのため、等値面表示(面)、等値面表示(ワイヤフレーム)は等値抽出点から構成処理に時間がかかるためリアルタイムではない。しかしながら、正確な結果を提供する。また、等値面表示(点)は等値抽出点からの構成処理時間を必要としないため、簡易的ではあるが等値面を高速に提供することが可能である。

### 2.3.4 操作手順概要(Windows版)

ファイルメニューにより読み込むボリュームデータを選択する。現時点で1次元的に複数のデータを一度に読み込むことが可能となっている(Figure 3(1))。読み込んだデータは左に表示されたモード切替ボタン(Figure 3(2))により前述した表示モードの切り替えが可能となっている。また、オプションチェックボックスでは等値面(面)の半透明や分子骨格の表示の有無なども可能となっている(Figure 3(3))。

本研究で提案した分子骨格による分子軌道変化の様子は、雲表示と等値面(雲)、そして面構成を必要としない点による描画でのみ利用可能で、描画モード選択ボタンの下にある分子骨格操作バーにより分子骨格

を操作でき、対話的にその状態の分子軌道が表示される(Figure 3(4))。分子骨格操作バーの下には等値面の値を切り替える操作バーがあり、これは等値面(面, ワイヤフレーム, 点)表示といったテクスチャを使用していない表示法に適應される(Figure 3(5))。

また、視点は Window 右, 下などに配置されたスクロールバーにより自由に切り替えることができる(Figure 3(6))。

なお本プログラムは公開されており、次の URL からダウンロードできる

(<http://www.om.sist.chukyo-u.ac.jp/main/research/molecule/>)

## 3 分子軌道描画への適用

本章では、2章で述べた方法を実際に分子軌道法で得られた cube データに適用した例について述べる。

### 3.1 水素分子 $2\sigma_g$ 軌道

まず、水素分子(核間距離 1.401 au)の  $2\sigma_g$  仮想軌道を見てみよう。基底関数は cc-pV6Z (g, h 除外)、計算レベルは RHF(Restricted Hartree-Fock)、プログラムは GAMESS [14] である。cube データのメッシュは  $81 \times 81 \times 81$  である。この軌道は 2s AO から成り、内部に節面を持つ。

節構造 [2] は、分子軌道の分類や Rydberg 軌道の対称性の同定において重要な情報を提供する。しかしながら、内部節面を可視化できる分子軌道描画ソフトウェアは多くない。たとえば、市販の分子軌道描画プログラム Chem3D [1] では、「ワイヤメッシュ」表示により内側の節面を見ることができ、もともと内部節面の可視化を目的として作られておらず、あまり分かりやすい表示ではない。

電子分布表示では、Figure 4(a)にあるように、外側の白色部分の内側に赤い部分が透けて見える。これにより内部節面を認識できる。しかし、雲表示であるがゆえに輪郭が明確にはならない点が難点である。

等値面表示(雲)は、核の動きに追従できる点の特長であり、内部節面の可視化においても、透過度を適應することによりある程度把握することができる(Figure 4(b))。これに比べ、等値面表示(面)(Figure 4(c))では内部節面がはっきり表示されている。

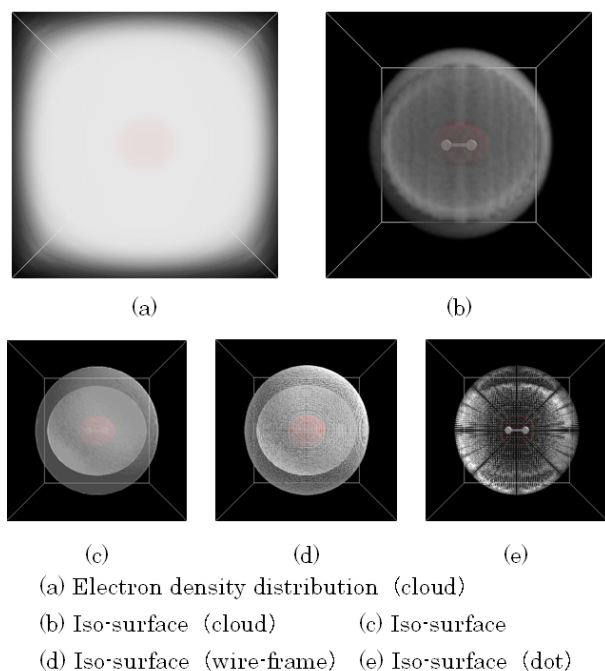


Figure 4. The  $2\sigma_g$  orbital of the hydrogen molecule

等値面表示(ワイヤフレーム)(Figure 4(d))や等値面表示(点)(Figure 4(e))でも認識できるが、等値面表示(面)が最も明確な図を与える。各表示方法には一長一短がある。内部節面の可視化には、電子分布表示(雲)や等値面表示(点)で大まかに節構造を探索しておいてから、等値面表示(面)に切り替えて詳細を調べるやり方が有効である。

### 3.2 CO分子 $5\sigma$ 軌道

基底関数は aug-cc-pVDZ, 計算レベルは RHF, プログラムは Gaussian 03[15] である。メッシュの数は  $81 \times 81 \times 81$  である。核間距離, 2 au ~ 5 au の 14 点の cube データを使っている。CO 分子の双極子モーメントは観測値が +0.112 Debye で,  $C-O^+$  に対応する。しかし, RHF レベルでは電荷の符号が逆に出ることが知られている。この逆転は,  $5\sigma$  の電子分布が炭素原子側に偏り, しかも, 軌道が酸素原子の方向に向いて横方向に張り出していることによる [16]。

このような解析に注目して  $5\sigma$  軌道の形状を見てみる。本表示システムでは, 核間距離を連続的に変化させたときの軌道の変化の様子を映像化することができるが, 紙面ではスナップショットを示す。平衡核間距離 (2.132 au) における  $5\sigma$  軌道の電子分布表示 (Figure 5(a)), 等値面表示(ワイヤフレーム) (Figure 5(b)) を見てみよう。

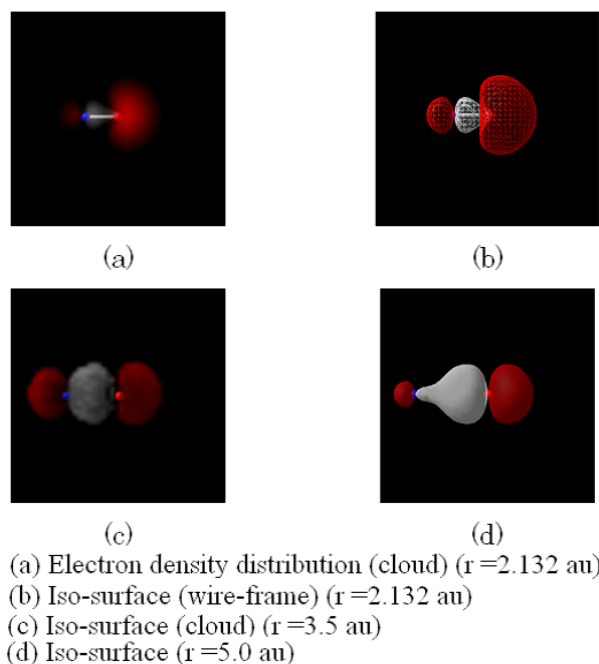


Figure 5. The  $5\sigma$  orbital of the CO molecule

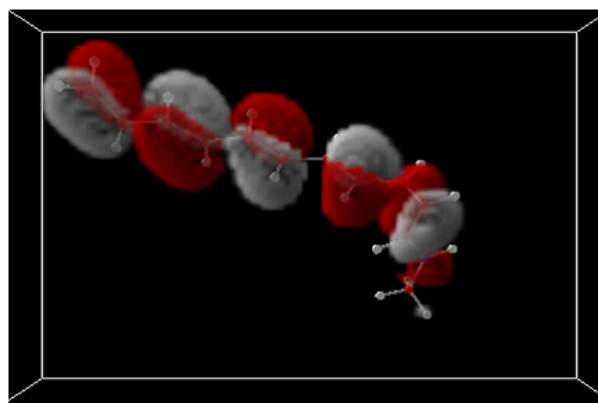


Figure 6. The HOMO of protonated Schiff base of retinal (PSBR)

図では C を原点に固定し, その左側 ( $-z$  方向) に O が配置されている。C の右側で大きく膨れた領域は, C の位置では窪んでいることが, Figure 3(b) の等値面表示(ワイヤフレーム)から鮮明に見てとれる。RHF による計算結果では, CO 分子の  $5\sigma$  が O ではなくて C 側に偏った電子分布を示す。

核間距離を 3.5 au, 5.0 au と延ばした場合の軌道を各々 Figure 5(c), Figure 5(d) に示す。Figure 5(c) は等値面表示(雲)で 2 つの原子の間に大きな電子分布があり, C と O の外側にほぼ対称的に分布している。Figure 5(d) 等値面表示(面)では, やはり C の両側に大きく電子が分布することが描かれる。

このように核間距離を変えながら，同時に表示方法を高速に切り替えることができる本システムの機能は，軌道の特徴をつかむ上で有効である．Figure 5(b),(c),(d)の等値面表示では 0.05 の値の面を描いている．本システムを使うことにより，この値を連続的に変化させて軌道の内部構造を把握できる．

### 3.3 レチナール プロトン化シッフ塩基 HOMO

最後に，色素タンパク質バクテリオロドプシンの発色団であるレチナールのプロトン化シッフ塩基 (protonated Schiff base of retinal, PSBR) の HOMO を取り上げる [17]．等値面表示(雲)したものが Figure 6 である．基底関数は 6-31g，使用プログラムは Gaussian 94[18] である．cube データのメッシュは  $82 \times 71 \times 12$  である．

バクテリオロドプシンは光エネルギーを利用したプロトンポンプとして働く．レチナール分子は all-trans の構造をとるが，568 nm の光励起により異性化反応が引き起こされ， $C_{13}=C_{14}$  結合が回転して 13-cis になる．等値面表示(雲)では， $C_{13}=C_{14}$  結合の回転に伴う HOMO の変化をリアルタイムで見ることができる．紙面では動画を見せることができないが，回転するにつれて，この結合より左側の部分に電子が局所化する様子が見える．12 個の核配置におけるボクセルデータを用いて，2 章で説明した内挿法により，円滑に MO の変化を表示することができる．

等値面表示(雲)は核配置の変化にともなう MO の変化を追跡するのに適した表示方法である．

## 4 むすび

本研究ではボクセルデータとして与えられた分子軌道データに基づき，分子骨格の変形に伴い分子軌道が変化する様子を，5 種類の表示によりリアルタイムで表示する分子軌道描画プログラム MOOTIC を開発した．分子軌道計算結果の可視化解析ツールとして，分子の電子状態の特徴を把握するのに役立つことが期待できる．

本研究の一部は文部省私立大学ハイテク・リサーチ・センター補助金による財政的支援を受けた．

## 参考文献

- [1] Chem3D : Cambridge Soft Corporation
- [2] Y. Hatano, S. Yamamoto, and H. Tatewaki, *J. Comput. Chem.*, **26**, 325-333 (2005).
- [3] 鐸木啓三, 菊池修, 電子の軌道, 共立出版 (1984).
- [4] 神沼二真, 鈴木勇, 分子を描く, 啓学出版 (1988).
- [5] T. Helgaker, P. Jorgensen, and J. Olsen, *Molecular Electronic-Structure Theory*, John Wiley and Sons (2000).
- [6] 中貴俊, 山本茂義, 秦野やす世, 山田雅之, 宮崎慎也, *J. Comput. Chem. Jpn.*, **1**, 135-142 (2002).
- [7] 中貴俊, 山田雅之, 宮崎慎也, 秦野やす世, 山本茂義, 電子情報通信学会 MVE 研究会, 2002-6, 東京, Jun.2002.
- [8] J. Neider, T. Davis, and M. Woo, *OpenGL Programming Guide*, Addison-Wesley (1993), p.478.
- [9] M. J. Frisch *et al.*, *Gaussian 98 (Revision A.5)*, Gaussian Inc. (1998).
- [10] T. Itoh, and K. Koyamada, *IEEE Transactions on Visualization and Computer Graphics*, **1**, 319-327 (1995).
- [11] P. Cignoni, P. Marino, C. Montani, E. Puppo, and R. Scopigno, *IEEE Transactions on Visualization and Computer Graphics*, **3**, 158-170 (1997).
- [12] S. Grimm, S. Bruckner, A. Kanitsar, and E. Groller: Memory Efficient Acceleration Structures and Techniques for CPU-based Vol.
- [13] Microsoft .NET FrameWork  
<http://www.microsoft.com/japan/msdn/netframework/>
- [14] M. W. Schmidt *et al.*, GAMESS 2004, *J. Comput. Chem.*, **14**, 1347-1363 (1993).
- [15] M. J. Frisch *et al.*, *Gaussian 03, Revision C.02.*, Gaussian Inc., Wallingford CT (2004).
- [16] 藤永茂, 入門分子軌道法, 講談社 (1990), pp.117-121.

[17] S. Yamamoto, H. Wasada, and T. Kakitani, *Theochem*, **451**, 151-162 (1998).

[18] M. J. Frisch *et al.*, *Gaussian 94, Revision C.3*, Gaussian Inc., Pittsburgh PA (1995).

## Real-Time Volume Rendering of Molecular Orbital by Iso-Surface Texture Mapping

Naka TAKATOSHI<sup>a\*</sup>, Shigeyoshi YAMAMOTO<sup>b</sup>, Yasuyo HATANO<sup>c</sup>, Mamoru ENDO<sup>c</sup>,  
Masashi YAMADA<sup>c</sup> and Shinya MIYAZAKI<sup>c</sup>

<sup>a</sup>Graduate School of Computer and Cognitive Sciences, Chukyo University  
101 Tokodachi, Kaizu-cho, Toyota, Aichi, 470-0393 Japan

<sup>b</sup>Faculty of Liberal Arts, Chukyo University  
101-2 Yagotohonmachi, Showa, Nagoya, Aichi, 466-8666 Japan

<sup>c</sup>School of Information Sciences and Technology, Chukyo University  
101 Tokodachi, Kaizu-cho, Toyota, Aichi, 470-0393 Japan

\*e-mail: naka@om.sist.chukyo-u.ac.jp

We have developed the visualization system MOOTIC that enables real-time observation of changes of molecular orbitals along the reaction coordinate or the nuclear displacement coordinate. Variation of the iso-surface of a molecular orbital is also visualized. In our system, a fast rendering method is implemented to display the iso-surface by interpolating a pair of voxel data sets similarly to the keyframe animation, which is an image generation method to blend a pair of key frame images into the interpolated frame. In advance, a series of voxel data sets are constructed by changing the nuclear coordinate, called the key position. We give examples of visualization on a Microsoft Windows platform. They are inner nodal regions of the  $2\sigma_g$  virtual orbital of the hydrogen molecule and changes of molecular orbitals occur through bond elongation (carbon monoxide) / bond rotation (protonated Schiff base of retinal). The number of inner nodal regions is useful to characterize molecular orbitals.

**Keywords:** Molecular orbital, Electron density cloud, Scientific visualization, Volume rendering, Real-time computer graphics