Haptic visualization of molecular model

Aleš Křenek, Martin Černohorský,
Zdeněk Kabeláč
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno
Czech Republic
e–mail: {ljocha,kabi}@fi.muni.cz, cerno@ics.muni.cz

Abstract

A haptic display allows human operators to manipulate and feel virtual environments. Visual and auditory feedback alone cannot enable a person to interact with the computers as naturally as he would interact with a real environment.
In this paper we present our current results on creating virtual models of results of quantum chemistry computation, particularly electron density field utilizing a haptic device. We mainly focus on mapping quantities with chemical meaning to force field. And we show how they are used together with the point interface haptic display PHANT*o*M^{TM}.

Additional Keywords: haptic, PHANT*o*M, chemistry, molecule, visualization

1 Introduction

Nowadays we have to solve various problems arising in the graphical visualization. We are able to visualize complicated processes on graphical displays. But it is often difficult to present data graphically so that user can easily understand them. The possibility to touch data, manipulate them, feel the pressure and resistance, preventing the user to move to restricted areas and navigating him in the 3D space might help to percieve properties of data.

We joined our haptical research group with the chemistry staff to develop new system for visualization of chemistry computation.

In recent years computational chemistry became at least an equal partner to traditional experimental chemistry. The entire area is too wide to speak about it generally, however, two common characteristics of the produced data can be observed:

- The amount of the data is huge.
- There are relatively numerous features which should be taken in account simultaneously when analyzing data.

The first statement implies the need of visualization and many tools and packages for this have been developed indeed. However, in most cases this software uses rather conservative techniques only. Visualization is usually limited to static or animated images (sometimes in stereo) of simple molecular models — balls for atoms and stick for bonds, ribbons depicting backbone chains in more complicated structures, or isosurfaces of certain quantities.

The second statement in the list above suggests using other facilities, not only pure visualization, which can bring impression of more features while preserving an acceptable complexity of the image itself.

Many of the quantities examined in computational chemistry are either some sort of energy or are related to energy quite closely. For example, an energetic barrier has to be overcome to initiate a reaction which converts certain reactants to products. Another example, electron density, is discussed in detail further in this article (see [2] etc. for more examples).

The relation to energy allows to map a lot of quantities to mechanical force. In traditional physics an integration of a force along a path equals to work and vice versa the force is proportional to a gradient of the energy. Therefore mapping certain quantities to force is fairly intuitive as far as we accept the traversal from the real microscopic world into virtual macroscopic one.

Building virtual models of molecules and their properties brings another benefit. Human knowledge and intuition is still powerful compared with automated computational methods. Computers have to perform exhaustive blind search across the entire state space of the particular problem. The user, according to his expert knowledge and intuition, could guide the machine from overall but roughly approximating model to refine it selectively in appropriate areas instead. Computation resources can be utilized more efficiently then. Among other results our research should add this efficiency to exhaustive computations carried by the program CICADA [6] which become unaffordable on larger molecules.

2 Overview

In this article we focus on modeling *electron density* with haptic hardware.

First we touch briefly the quantity itself, discuss creation of the force field according to this quantity, and finish with a description of visualization.

2.1 Electron density

Knowledge about electron properties of a molecule is a key condition for predicting chemical reactivity or biological activity of chemical compounds. Although the electrons represent only small fraction of the total mass of the molecule, they occupy most of the volume of the molecule and thus determine its behaviour. The overwhelming majority of methods used in quantum mechanics for calculating electron distribution in a molecule are based on the assumption that the atom nuclei are stationary charged points, and the electron density is a function of the nuclei configuration. This assumption is known as Born-Oppenheimer approximation.

The principal task of these methods is to solve the wave equation in the form

$$\widehat{\mathbf{H}}\Psi = E\Psi \qquad (1)$$

where $\widehat{\mathbf{H}}$ is the Hamiltonian which describes the particle in terms of forces affecting the electron, namely the electric field caused by the atomic nuclei and the other electrons eventually, Ψ is the wave function and E is the electron energy. The expression $\Psi\Psi^*$ (product of Ψ and its complex conjugate) is interpreted as a probability of occurrence of the electron; in other words—the *electron density.*

Because these calculations require unacceptable computing resources there were introduced several other approximations. In this paper we used the semiempirical AM1 method [4] as implemented in MOPAC [9]. Semiempirical methods are taking into account only the electrons of the valence layer of the atoms constituting the molecule and all underlying electrons along with the nucleus are treated as so called *core.* With this semiempirical method it is possible to calculate properties of majority of small to middle-sized organic molecules. More about quantum mechanics methods can be found in: [10, 11, 5, 1].

2.2 Generating force field

We start with the following intuitive requirements

- The higher is the density the “harder” should the virtual molecule “feel”.
- While delivering a constant magnitude force the user should be able to track an isosurface of the density.

The first requirement determines a magnitude of the force at any point — it should be directly proportional to the density at that point. From the second one we can derive the direction of the force.

A perception of touching a surface at a point is actually a perception of a resistance (or force) in a direction normal to the surface at that point[1]. Therefore an illusion of a surface in a virtual environment can be created by a force which is normal to the surface at all its points.

Figure 1 illustrates the principle for isosurfaces (or isolines as the figure is two dimensional only) of example values 1 and 2 of the density function.

We extend the principle from isolated isosurfaces to continuous space by considering an isosurface at any point in the space

[1] Imagine touching a rubber ball with a spike of a pencil.

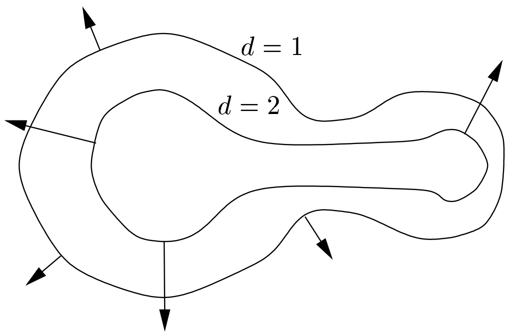

Figure 1: Isosurfaces and forces

and computing the force according to that surface. This approach is correct because any point is located on some isosurface as long as the density is defined at that point. On the other hand, if the density function and its partial derivatives are continuous (these properties can be assumed) the isosurface is determined uniquely at all points of nonzero gradient of the density function. Stationary points (those with zero gradient) require special handling and are discussed bellow.

We shall summarize the previous statements into a formula used for computing the force at any non-stationary point $\boldsymbol{x}$

$$F(\boldsymbol{x}) = -d(\boldsymbol{x})\frac{\nabla d(\boldsymbol{x})}{|\nabla d(\boldsymbol{x})|} \tag{2}$$

where d is the density function defined as product of the wave function and its complex conjugate $d(\boldsymbol{x}) = \Psi(\boldsymbol{x})\Psi(\boldsymbol{x})^*$, $\nabla d(\boldsymbol{x})$ is its gradient vector, and $|\nabla d(\boldsymbol{x})|$ its Euclidean norm. As the expression $\frac{\nabla d(\boldsymbol{x})}{|\nabla d(\boldsymbol{x})|}$ is a normalized vector in the direction of the gradient the force computed in this way satisfies the requirements — its magnitude is proportional to the value of the density function and its direction is given by the negative gradient. The force is normal to an isosurface and pointing “outside” (to the area of lower density) then.

2.3 Visualization

We began this article declaring that other devices like force feedback can deliver more information in situation when just graphical visualization would become too complicated. However, a vice versa statement is true as well. Our experiments show that for a user with no special training, it is extremely difficult to create a mental image of the examined virtual object from single point force perception only[2]. Understanding rapidly improves when visual perception is added.

The connection between the force field delivered by the physical device and the visualization is realized statically by coordinate alignment and dynamically by depicting current position of the device in the image. We use a little sphere of distinct colour in our implementation but one can imagine another highlight in the visualization (e.g. a light source in a volume rendered image) or even realistic reconstruction of the user’s hand and the device itself.

Depending on actual requirements two basic approaches to the visualization can be taken:

- *Tight coupling* of the visualization and the force field. The image is derived from the same data as the force field (or at least these data are included in the image). Our implementation is an example of this approach. We show an isosurface of chosen low threshold value of the electron density. Moreover, we clip the force field with this surface making the force be zero outside. In effect, the user feels touching the molecular model in the same instant as the visualized device intersects with the model in the image.

- *Loose coupling.* The image comes from different quantities than the force field. We could use some simple schematic models like balls and sticks, a simple surface (van der Waals [2] or its more general form—Connolly surface [3]), or choose visualization based on different quantities (partial charge, hydrophility, etc.). We have not carried out experiments with this visualization yet, but we believe it will prove useful.

3 Implementation

The implementation of our experimental system consists of three more or less independent components:

1. Force field and model calculation

[2]We have actually carried out our first experiments with this data with no visualization available.

2. Graphical visualization
3. Haptical visualization

The actual data calculation is not listed here as it is performed by standalone third party quantum mechanics software [4, 9]. Obviously, our system is dependent on data formats of that software. However, this dependency is strictly located in a single library with well-defined API (initialization, calculation of the density at a particular point etc.). This library reads the required files and calculates some derived quantities as well while keeping the rest of the system independent on actual data source.

3.1 Force field calculation

This program takes output files of quantum mechanics software and according to the formula (2) it produces a force field in form of a grid file. In that file a force is stored for each point in discrete 3D regular grid of the given resolution.

3.1.1 Gradient computation

Partial derivatives of the density function are not generally available in neither analytic nor procedural form. Therefore we approximate the derivatives with central differences on the grid. Let i, j, k be a position in the grid. The derivative approximations are calculated in this way:

$$\left.\frac{\partial d}{\partial x}\right|_{(i,j,k)} = \frac{d(i+1,j,k) - d(i-1,j,k)}{2l_x} \qquad (3)$$

where $d(i, j, k)$ is the density at a particular grid point (i, j, k) and l_x is the distance of two successive grid points in the x-direction and the other partial derivatives are approximated in a similar way.

3.1.2 Valid and forbidden areas

Although a force can be computed in the above described way at any non–stationary point in the grid, from the chemical point of view it makes sense to do so in an “outer” area only. In the used semiempirical calculation only valence orbitals are considered. Those electron orbitals form shells of various shapes. The density is high in those shells, it decreases in the outside direction, and the shell’s interiors are almost “empty” (zero density).

Our simple model approximately reflect force interaction of the molecule with a negatively charged particle. It is valid only as long as the particle is approaching the molecule from outside. If its energy were high enough to reach a valence shell it would interact with the molecule, cause principal changes in its shape, form or break bonds etc. Computing (and even approximating) such an interaction in real time is currently impossible.

So it is necessary to separate the outer valid area and inner forbidden ones. Fig. 2 shows an example of density field restricted on two dimensions only. Part (a) represents

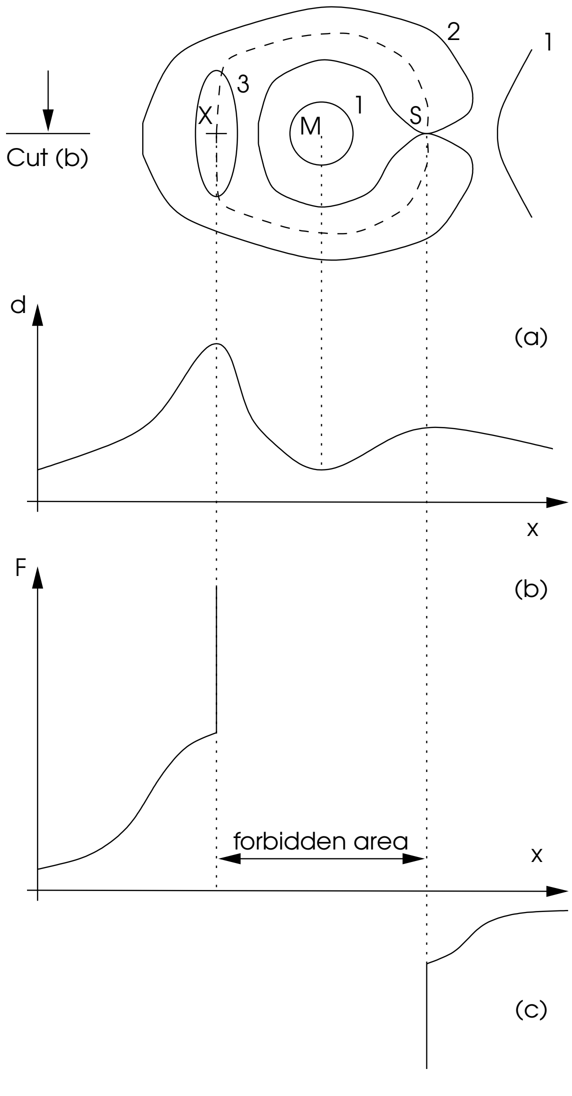

Figure 2: Valid and forbidden areas

a contour map of the density with depicted example values corresponding to particular contour lines, (b) is the graph of the density values along a horizontal line going through the centre of (a). Stationary points (see later – local maximum, local minimum, and saddle point) are labeled with X, M, and S.

The extent of the forbidden area in horizontal direction is shown in the figure, its border is drawn with a dashed curve in the contour map. (This curve becomes a surface in 3D obviously.) Intuition suggests to define the forbidden area border as a “ridge” closing “valleys” of lower values of the density. Exact specification of the forbidden area border in terms of analytical properties of the density function which would respect chemical significance and its computation, are topics of our current research.

3.1.3 Force adjustments

As shown in fig. 2 (c) the force magnitude should increase at the border of the forbidden area to potential infinity (physical limit of the hardware device) so that the user was not able to reach the forbidden area. In this way the forbidden area in the force field forms a hard “core” representing inner orbitals which are essentially inert and do not contribute to chemical properties of the molecule.

According to experimental experience the best results are achieved with further logarithmic transformation of the force magnitude. Furthermore we can substitute zero force in areas of density below some threshold (probably the same as used for visualization) in order to create a better illusion of a strictly bounded object.

3.1.4 Stationary points

The general method of computing a force from the density field is given by formula (2). It is valid for non–stationary points only. Therefore we have to introduce a classification of the stationary points of a function of three variables and handle each class separately. Fortunately, the concept of forbidden areas together with assumptions of continuity of the density function and its zero value in infinity reduces those singular cases significantly. As usual, we classify the stationary points according to eigenvalues of the Hessian matrix of the density function:

- *Local minimum.* All eigenvalues positive. Local minima always falls into forbidden areas so no handling is required.
- *Local maximum.* All eigenvalues negative. If the local maximum falls at a forbidden area border, in fact, this point looses its *maximum* property as we increase the force here to infinity. So the direction of the force should be normal to the border of the forbidden area here.

  If the maximum is reached in a standalone peak (very unlikely anyway) the finite precision grid most likely misses it. Otherwise we can choose an arbitrary direction in this point (introducing a nonsignificant error only). The resulting force field behaves in an expected way—the device is repelled from the maximum in all directions.
- *Saddle point.* Either one or two eigenvalues negative, the others are positive. Saddle points of first or second order always fall at a forbidden area border so similarly to local maximum no special handling is required.
- *Another stationary point.* One or more eigenvalues are zero. Force direction can be interpolated from surrounding of the stationary point as it is neither extremum nor saddle. It is sufficient to take the central difference of larger radius in expression (3).

3.1.5 File formats and sizes

A single grid point force is now stored as 6 bytes ($3\times$ short integer) which is above the actual resolution of the physical device. In our experiments we use grid size $200 \times 260 \times 85$. thus we work with approximately 26MB files.

For the force field calculation is the most exhaustive part of our system in terms of computation it is done only once for each molecular model and is performed offline. The calculation time is dependent on the grid size and, with our method, proportional to the complexity of the molecule as well. For example it takes about 5 minutes on MIPS R10000 for the previously mentioned grid size on a simple testing molecule of 1,3-butadiene 2-nitrile.

3.2 Graphical visualization

In the first stage of the project we mostly concentrate on the force field generation so accompanying visualization has been kept simple. We use the generic Visualization Toolkit (VTK) [8] for this purpose.

First an isosurface of some low density threshold value is created with the marching cubes algorithm [7] implemented in the VTK. This computation is performed offline together with the computation of the force field and stored in a VTK file.

During interaction with the user the visualization subsystem is continuously receiving data about the current position of the device. The electron density value is recomputed in that position (the computation of the density in a single point is affordable) and compared with the threshold value. When greater we conclude the device has reached the virtual object and highlight the area of collision with a colour mapped to an actual density at the location of the device.

This visualization is fairly simple, however, it serves its main purpose to deliver an overall notion of the object which is examined with the force feedback device otherwise. Anyway, the visualization should become a subject of further improvements.

3.3 Haptic visualization

A haptic display differs from the graphical display in several ways. The graphical display has to render large environment at refresh rate about 25Hz. With slightly higher refresh rate we could achieve a bit smoother move, but there is no reason to create more frames as of the limitation of the visual perception.

The haptic display needs to show only small portion of the whole scene (as we just move single point representing our finger in the virtual world), but everything has to be calculated with quite higher refresh rates – the acceptable minimum is around 1kHz and it looks that the refresh rate above 5kHz won't give us any more details.

One common thing for both displays is that any longer delay between two display updates (temporal slowdown in update frequency) is annoying for the user.

For this kind of task starts to be important precise timing. We overlooked already developed applications and we found most of them were written in DOS. This system gives the user the best control on what is happing in the computer, as this operating system does practically nothing.

We wanted more general solution, as we planed to work with the net, large memory areas, multitasking, 3D displays and as we mentioned earlier we wanted some support for real–time tasks.

With the haptic device itself we have received sources of drivers for Windows NT and DOS. Both of them failed to pass our check list. We selected Linux and applied Real Time patch [14][3]. This Linux extension was developed to control robotic devices and thus it is very appropriate to control our PHANT*o*M[4] device.

Because we are working with "robotic" device we have to precisely sample position from our haptic device and update forces based upon the received results. For this task we've developed special driver and we are able to update position with high precision. We use 2.5kHz (400μs) refresh rate and the maximum delay between two updates is around 450μs running RT–Linux on common PC (200MMX).

Our first experiments were oriented to display simple geometrical models like balls and walls and we had to deal with several problems. Some of them we've solved with our implementation of "proxy" position[5] and can be found in [12]. This solution was using the FDHs, presented in the work [13], for the fastest collision detection.

The task for our chemistry data is different, thus we had to create support for the force–fields. In the current state of development we are using only precalculated force–fields we have described earlier.

Algorithm 1 haptic software loop

```
loop
  locate position in the VR space
  detect collisions
  calculate a reaction force vector
  apply that force
  inform other tasks
end loop
```

In our case the part which detects collisions and calculates a reaction vector is made with this fast and simple algorithm.

Our current setup for interactive manipulation with the molecule is composed from SGI Indigo R4400 which is being used for

[3] Most of todays *RT* systems deliver ms precision – with this patch we have μs precision

[4] Personal Haptic iNTerface Mechanism

[5] Other implementation can be found in [15]

Algorithm 2 Phant*o*m calculation function

receive actual position
pick the right cube in the force–field
make fast trilinear interpolation
send calculated forces

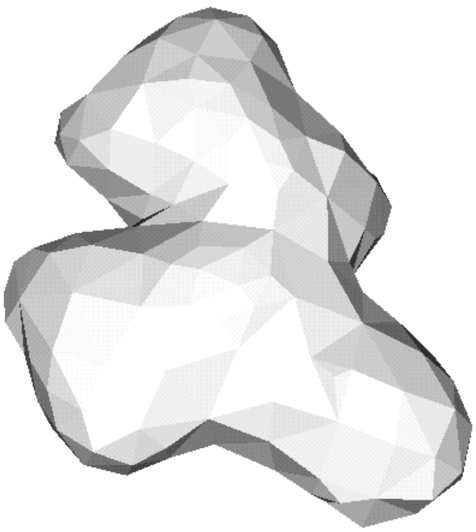

Figure 3: Visualized molecule with VTK

the graphical visualization and PC Pentium 200MHz MMX controlling the haptic device. Figure 3 presents visualized molecule with VTK and fig. 4 shows our haptic device.

4 Conclusion

Our model is now based upon the use of one PHAT*o*M which has only three degrees of freedom[6]. This can be modeled as a point or frictionless sphere in virtual environment and only those forces at the tip of the tool can be accurately simulated with point force interaction.

In the future we want to improve physical model of the whole environment. Also we will try to join two PHANT*o*Ms to be able to simulate torsion move.

5 Acknowledgement

This project is supported by GAČR with the grant #201/98/K041. We are also grateful to Jiří Sochor and Petr Konečný for their help.

[6]we are also able to detect six degrees of freedom, but we can't react on the finger rotation

Figure 4: Our PHANT*o*M device

October 1998

References

[1] *Basis Sets for ab Initio Molecular Orbital Calculations and Intermolecular Reactions*. In: K. B. Lipkowitz, D. B. Boyd: Reviews in Computational Chemistry, VCH Publishers, Inc., New York, 1990.

[2] George M. Bodner and Harry L. Pardue. *Chemistry, An Experimental Science*. John Wiley & Sons, Inc., 1989, 1995.

[3] Michael L. Connolly. Solvent – accessible surfaces of proteins and nucleic acids. *Science*, 221:709–713, 1983.

[4] M. J. S. Dewar et al. *J. Am. Chem. Soc.*, 107:3902, 1985.

[5] W. J. Hehre, L. Radom, P. R. Schleyer, and J. A. Pople. *Ab Initio Molecular Orbital Theory*. Wiley, New York, 1986.

[6] Jaroslav Koča. – computer program CICADA — travelling along conformational PES. *J. Mol. Struct. (Theochem)*, 308:13–24, 1994.

[7] Lorensen and Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM Comp. Graphics*, (21):163–170, 1987.

[8] Will Schroeder, Ken Martin, and Bill Lorensen. *The Visualization Toolkit.* Prentice Hall, Inc., 1996.

[9] J. J. P. Stewart. Mopac. F. J. Seiler Res. Lab., Colo. Springs, QCPE # 455.

[10] J. J. P. Stewart. *Semiempirical Molecular Orbital Methods.* In: K. B. Lipkowitz, D. B. Boyd: Reviews in Computational Chemistry, VCH Publishers, Inc., New York, 1990.

[11] M. C. Zerner. Semiempirical molecular orbital methods. In D. B. Boyd K. B. Lipkowitz, editor, *Reviews in Computational Chemistry*, volume 2, pages 313–366. VCH Publishers, Inc., New York, 1991.

[12] Petr Kučera. Master thesis, *Force — Feedback in Virtual Reality (in czech)* 1998

[13] Petr Konečný. Master thesis, *Bounding Volumes in Computer Graphics* 1998

[14] Real Time linux.
http://www.rtlinux.org

[15] Diego C. Ruspini, Krasimir Kolarov and Oussama Khatib *The Haptic Display of Complex Graphical Environments* SIGGRAPH 1997

[16] Thomas Harold Massie *Virtual Touch Through Point Interaction* 1993