Molecular models superposition and visualization

Aleš Křenek
Masaryk University
Faculty of Informatics
Burešova 20
Brno
Czech republic

E-mail: `ljocha@fi.muni.cz`

Abstract

A visualization system of molecular behaviour is presented in this paper. The system was designed for very large input data sets. An essential problem of the visualization is finding the best match of two structures. A solution (the *superposition* transformation) is discussed in detail.

1 Introduction

Software systems on molecular modelling produce extremely large amounts of primary results in these days. The visualization becomes almost the only way essential features in the examined chemical system can be observed and concentrated on in the further processing when so much data describe the system and its behaviour.

Recently a package of a new approach in conformational[1] analysis was developed. Besides the quantity the output data differ from those of other chemical software as not a continuous process but only discrete states are described (see section 2).

The available software systems (XMOL, BIOSYM), however, are not able to deal with such amounts of data in general (strictly speaking, the data they have not produced themselves), e.g. XMOL requires a single huge file describing all the visualization steps and tries to load it all into memory. That is obviously impossible without an excessive hardware support.

Because of the way the calculations are done, the output configurations are located and oriented in a virtually random position in the cartesian coordinate space (see figure 1). It is not easy to find out that the top part of the molecule doesn't change too much and the bottom one rotates a bit. Having been faced with a sequence of such images running at 15 ones per second (the required speed of the animation) the user couldn't get any profit of the visualization.

Therefore the *superposition* is required. The user chooses a substructure which is supposed not to change the relative positions of its atoms too much. For each configuration a transformation is calculated which, having been applied on all the atom coordinates, causes the chosen substructure to be fixed in almost the same place. However, the available visualization systems support only a very trivial superposition (e.g. XMOL fixes the first atom only not taking account of an orientation of the rest of the molecule).

From the above discussion the requirements on the developed system emerge. It has to be able to deal with large data, calculate a sophisticated superposition, and animate the simulated behaviour of the molecule at a sufficient speed in terms of a sequence of images like those in the figure 1. In addition, the system should be as independent on hardware as possible and runable even on a relatively cheap one.

[1] It is not the subject of this paper to discuss chemistry. When involved in conformational behaviour atoms of a molecule change their position only, the bonds are not affected.

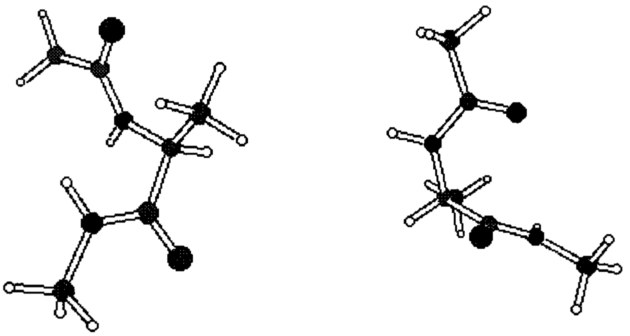

Figure 1: Random orientation

2 Calculations and the data

A crutial property of a molecule is its potential energy. Displayed in a multi-dimensional diagram where the energy depends on distances between atoms and angles between bonds (*internal coordinates*) it forms a *potential energy surface* (PES).

The package of chemical calculation consists of two main parts. The first one, *Cicada* [6], calculates all the possible behaviour of an examined molecule. This is independent on environmental conditions (temperature etc.). Viewing the calculation on PES, Cicada makes the molecule travel from a given starting local minimum along a bottom of a valley by rotating a part of the molecule along a chosen bond. Two points must be emphasized

- Changing the shape of the molecule is initiated by a distortion of the model only with almost no dependency on real behaviour. Positions of atoms are recalculated then in order to meet energy constraints but the initial distortion is preserved.
- The process does not model a real one yet, it is only a *possibility* of such behaviour and all those possibilities are examined.

Unlike standard molecular behaviour analysing software only local minima (the conformers) and saddle points (transition states) are saved. This causes the resulting output data to be "very discrete"—both the real time elapsed between neighbouring configurations and the changes of structure are rather big. A result of Cicada's calculation is a graph where vertices represent possible shapes of the molecule and edges are possible transitions between them.

Once Cicada is finished another program, *Combine* [5], simulates a single possible behaviour of the molecule under certain conditions. Stochastic methods are involved here. The result is a sequence of states formely calculated by Cicada.

In a real application the PES graph can be composed of upto 100,000 distinct configurations and the simulation can involve more than 1,000,000 steps.

Cicada works strictly with internal coordinates and only the configurations which are saved are converted into the Carthesian space. This is the reason of scattering the resulting structures in random position and orientation.

3 Superposition

We have already mentioned the problem of finding the best match of two corresponding substructures and the importance of its solving. In this section we shall focus on the methods.

Only a brief version is presented, the original ideas come from [2] and [3], the full construction including the underlying theory can be found in [4]. As only a substructure of the original molecule is important now we shall restrict our attention to that substructure in all considerations. The emerging transformation is, of course, applied on the entire molecule again.

The first task is a formal specification. What does it mean "the structures match fairly well on each other"? One must assign a number to each pair of structures which would express the intuitive "level of matching". A ***residual*** can be defined in many ways but a sum of squared distances of atoms is used in general

$$\boldsymbol{RS}(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{n} |\boldsymbol{x}_i - \boldsymbol{y}_i|^2 \tag{1}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ stand for the substructures being matched, $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ are their atoms considered as points in 3-dimensional space and n is the number of those atoms. This is good enough to express the intuitive requirements and easy to be delt with as well.

The ***superposition*** is a transformation Q (chosen from some specified set) such that the equation $\boldsymbol{RS}(Q(\boldsymbol{x}),\boldsymbol{y})$ is minimal. We tried [1] transformations of the form

$$Q(\boldsymbol{x}) = R\boldsymbol{x} + \boldsymbol{t}$$

where R is a matrix and $\boldsymbol{t}$ a vector. Such transformations are quite easy to minimize (system of linear equations) but unfortunately they cause an unacceptable distortion of the structure.

That's why only transformations which preserve distances must be considered. First the centroids of both the structures are shifted to the origin. The superposition is a minimizing orthogonal transformation then. A general condition of orthogonality is expressed by $RR^T = E$ which yields six more nonlinear equations. Such a system is solvable by numeric methods but an implementation is rather complicated. Another great problem of this approach are singularities and, regarding the finite precision of floating point operations, nearly singular states which must be handled.

The involved orthogonal transformation must be parametrized in some more sophisticated way then. A standard Eulerian angle approach is presented in [8], however, it is somewhat clumsy. On contrary, the *quaternion* parametrization and the emerging minimization is a very nice application of a part of algebra[2] which is generally considered not to be useful at all.

In the following for $q \in \boldsymbol{H}$ (where $\boldsymbol{H}$ is the 4-dimensional space of quaternions) the notation $\bar{q}$ stands for a conjugate defined in a similar way as in the case of complex numbers. Treating the point $(x_1, x_2, x_3) \in \boldsymbol{R}^3$ as a quaternion $\boldsymbol{x} = (0, x_1, x_2, x_3) \in \boldsymbol{H}$ and given a normalized quaternion q it can be proven that the mapping defined by $\boldsymbol{x} \mapsto q\boldsymbol{x}\bar{q}$ is an orthogonal transformation of $\boldsymbol{R}^3$. Vice versa, given an ortoghonal transformation Q a corresponding normalized quaternion q exists.

In terms of this transformation the residual can be expressed (using some rules of quaternion algebra [7])

$$\begin{aligned}\boldsymbol{RS}(Q(\boldsymbol{x}),\boldsymbol{y}) = \sum_{i=1}^{n} |q\boldsymbol{x}_i\bar{q} - \boldsymbol{y}_i|^2 = \\ \sum_{i=1}^{n}(q\boldsymbol{x}_i\bar{q} - \boldsymbol{y}_i)(\overline{q\boldsymbol{x}_i\bar{q}} - \overline{\boldsymbol{y}_i}) = \\ \sum_{i=1}^{n}(q\boldsymbol{x}_i\bar{q}q\overline{\boldsymbol{x}_i}\,\bar{q} + \boldsymbol{y}_i\overline{\boldsymbol{y}_i} - \boldsymbol{y}_i q\overline{\boldsymbol{x}_i}\,\bar{q} - q\boldsymbol{x}_i\bar{q}\,\overline{\boldsymbol{y}_i})\end{aligned} \tag{2}$$

[2] For details on quaternion algebra see [7].

The first two terms of the final expression are the norms of $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ respectively (as $q\overline{q} = 1$). They are independent on q at all. The others are conjugates to each other therefore the residual is minimal iff the expression

$$\sum_{i=1}^{n} Re(y_i q \overline{x_i}\, \overline{q}) \tag{3}$$

is maximal. After another rather technical transformation we arrive at the result that having fixed the values of $\boldsymbol{x}_i, \boldsymbol{y}_i$, the expression (3) is a *quadratic form* mapping $\boldsymbol{H} \to \boldsymbol{R}$.

The *Principal Axes Theorem* [7] states that a given quadratic form $f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}A\boldsymbol{x}$ can be expressed as

$$f(\boldsymbol{x}) = \lambda_1 y_1^2 + \lambda_2 y_2^2 + \lambda_3 y_3^2 + \lambda_4 y_4^2 \tag{4}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are eigenvalues of the matrix A and y_1, y_2, y_3, y_4 are coordinates of $\boldsymbol{x}$ in an orthonormal basis consisting of eigenvectors of A. In order to minimize the residual the expression $f(\boldsymbol{q})$ must be maximized in terms of normalized $\boldsymbol{q} \in \boldsymbol{H}$. It can be easily seen that a normalized eigenvector corresponding to the largest eigenvalue should be taken as the maximizing quaternion.

Algorithms on eigenvector/value calculations are fairly well known, a smart one is a part of [2]. That algorithm arrives at an acceptable approximation in less than 20 iteration steps.

Finally we shall sum up all the necessary computation.

1. Shift centroids of the structures to the origin.
2. Using the shifted atom coordinates caculate the quadratic form matrix.
3. Find the largest eigenvalue.
4. Take the corresponding eigenvector and calculate a matching orthogonal trasnsformation.
5. Compose the two translations (1) and the orthogonal transformation (4) in order to get the superposition.

In all the calculation only basic floating-point operations (addition, multiplication and division) are used so the described algorithm is particulary fast.

4 Visualization

The graphic representation of the transformed molecules is fairly straightforward. A 3D model of the molecule is created where little balls representing atoms are connected with sticks—bonds. A perspective projection of such model forms a single frame of the animation. As the molecules have been already superposed the animation is fairly smooth.

In order to meet the low-cost hardware requirement and the speed simultaneously, only very simple *models* (the way of a final look of the molecule image) can be used. Currently only two-coloured sticks as bonds and optionaly one colour filled circles as atoms are supported. An example can be seen in fig. 1. However, the system can be easily extended by more sophisticated ones if a hardware rendering is available.

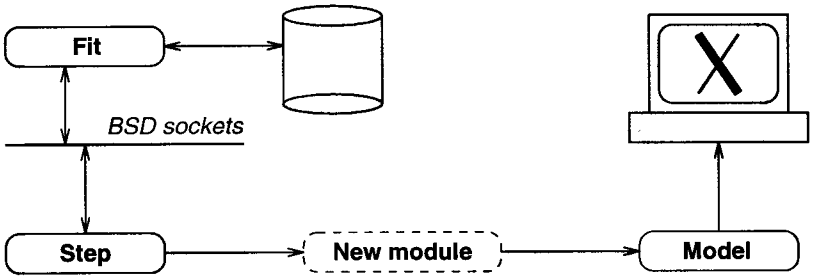

Figure 2: System design

For the same reasons the used hidden-line algorithm is the simplest one—back-to-front painter's algorithm where the elements (balls and sticks) are sorted by the depth of their centroids. That algorithm is fast enough, requires no additional memory but does not give a correct result when the bonds hide one another in a cycle. Fortunately, this hardly happens. Again a hardware support can be utilized if required.

Choosing the best substructure to be superposed is not a simple task. This cannot be probably determined unambiguously in a general case and the selection may even differ during different examinations of the same data so this choice is entirely left to the user.

5 The system

According to the required independence on hardware and software, UNIX, X11R5, and C were chosen as a software platform. The visualization system was ported and tested on the following OS's and HW

- NetBSD 1.0, XFree 3.1, PC-486. The the developement was mostly done on this system.
- SunOS 5.3, SUN Sparc ELC and Sparc 10
- IRIX 5.2, SGI Indy XL
- IRIX 6.0, SGI POWER Challenge (remote X-server, 32-bit executables)

The system was tested in two UNIX clones (BSD and SVR4) so it is likely to be portable to any similar operating system.

A rough design is displayed in the figure 2. The system is more or less based on a STREAMS-like structure. There are some modules connected with pipes. Messages of several types are passed through the pipes. If a module understands an incoming message the message is processed. It may be either consumed or modified and passed on. New messages may be generated as well. On the other hand, an unknown message is passed on unchanged. The following types of messages appear in the system.

Topology. By topology we mean the description of atoms and bonds not regarding their absolute position. When a conformational behaviour is examined, a message of this type appears only once—when a new molecule is read in. In the case of reactions the importance of these messages grows.

Geometry. This is just a set of atom coordinates. Both topology and geometry messages are generated directly from the input data, passed through the system and finaly utilized for creating images in the X-server.

Request. Due to the client-server architecture described bellow the former messages must be asked for.

In addition, a few other types exist which are used for dealing with startup, new molecule, end-of-data etc. As the modules can be distributed over a heterogenous network, the standard **xdr** encoding is strictly used for all the communication.

Sometimes an asynchronous or reverse direction communication is required as well. As there are only little data transmitted a standard X communication via *properties* is used.

The modules displayed in the figure 2 are standalone UNIX executables. We shall focus on their specific functions now.

Fit. This is a data server dealing with the huge file structure produced by the molecular modeling programs. It is implemented as a standard server accepting connections at an internet-domain socket. Serving an incoming request the module loads the required configuration, calculates its superposition and keeps the resulting (already transformed) structure in a buffer as long as possible. There is no need to keep the raw, not superposed data and a significant speedup is achieved then.

Step. The sequence of images which are displayed finally is actualy generated here. By controlling this module the user can start, stop, rewind, and step forward and backward the sequence produced by the molecular modeling programs.

Model. Actual images of molecules are created from the incoming *topology* and *geometry* messages and displayed in a window. In fact, each *geometry* causes a new image to be generated. The properties of a view (e.g. the eye position, perspective ratio) can be changed interactively. An interface for starting up the other modules is included as well.

Each of the modules makes a separate connection to the X-server in order to provide a user interface (filenames input, buttons controlling the sequence of images etc.).

If required a new module can be inserted into the pipeline between STEP and MODEL. It should read its standard input for incoming messages, write the standard output and follow the described STREAMS-like discipline. In this way any additional data processing can be included.

6 Conclusions

A new visualization system which fills in a gap in an existing software was designed. Unlike similar systems it can deal with quite large data sets and animate long sequences of images. A distribution over network can be utilized as well.

The superposition transformation is calculated in a sophisticated way. This is essential to keep the structures which are scattered by primary calculations in a fixed location.

Determined by the nature of the calculation the visualized data are only a very large grain discrete approximation of a real continuous process. However, the experience of using the system shows that it is sufficient to give the user an impression of the real behaviour.

The system is quite independent on hardware, it was tested on several different platforms. The primary design is open, the system can be extended by new modules of data processing, interfaces of new input formats, and new looks of the output in an easy way. Though the system was designed for a particular problem originally, a modification for another one should be straightforward then. Anyway, the ideas or even parts of the code can be used anywhere if a sequence of scattered graph-like structures appear, should it be a completely different application area than chemistry.

Currently a group of people uses the system experimentally. As soon as an acceptable release is finished it will be available in public-domain.

Anyway, the nice utilization of the quaternion algebra would be a piece of good news for Sir William R. Hamilton.

Acknowledgement

This work has been supported by the Grant Agency of the Czech Republic, grant no. 203/94/0522. Special thanks to Luděk Matyska, Jaroslav Koča, and Jan Slovák for discussions on the topic.

References

[1] Jiří Czernek, *Molecular Fitter vs. 1*, unpublished results, 1994.

[2] David J. Heisterberg, *Quatfit*, unpublished results, `ftp://kekule.osc.edu/pub/chemistry/software/SOURCES/C/quaternion_molfit`, 1990.

[3] Simon K. Kearsley, *An Algorithm for the Simultaneous Superposition of a Structural Series*, J. Comp. Chem. **11** (1990), 1187–1192.

[4] Aleš Křenek, *PES Travelling Visualization*, Master's thesis, Masaryk University, Faculty of Informatics, 1995.

[5] Jaroslav Koča, *Computer Simulation of Conformational Movement Based on Interconversion Phenomenon*, J. Mol. Struct. (Theochem), in press, 1995.

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

[7] S. Mac Lane, G. Birkhoff, *Algebra*, ALFA, 1973.

[8] Zong Jie Liu, Roland van Rapenbusch, *A Fast System for the Best Matching of Two Sets of Atomic Coordinates*, Computers Chem. **13** (1989), 5–23.