3D VISUALIZATION AND VIRTUAL EXPLORATION OF GENOMIC SEQUENCES

J Hérisson, N Férey, P-E Gros, O Magneau and R Gherbi

Université Paris-Sud XI, LIMSI-CNRS, BP 133 F-91403 ORSAY CEDEX, France
E-mail: {herisson,gros,ferey,magneau,gherbi}@limsi.fr

ABSTRACT

In this paper, we address the potential offered by Virtual Reality for 3D modeling and immersive visualization of large genomic sequences. The representation of the 3D structure of DNA allows biologists to observe and analyze genomes in an interactive way at different levels. We developed a powerful software platform that provides a new point of view for sequences analysis: ADN-Viewer. Nevertheless, a classical eukaryotic chromosome of 40 million base pairs requires about 6 Gbytes of 3D data. In order to manage these huge amounts of data in real-time, we designed various scene management algorithms and immersive human-computer interaction for user-friendly data exploration.

Keywords: Virtual Reality, Scientific Visualization, 3D Modeling, Human-Computer Interaction, Bioinformatics.

1. INTRODUCTION

Most biologists work on textual DNA files (i.e. sequence of the four known nucleotides A, C, G and T) that are limited to the linear representation of DNA. Such linearity offers only a local and partial view of these molecules (each letter in the sequence represents a nucleotide). However, within a cell, the DNA is a supercoiled double-helix, resulting in a complex 3D trajectory in space. The 3D structure of such complex conformations is very important in many essential biological mechanisms, such as replication, transcription or regulating. Indeed, two genes located very far away within a textual DNA sequence might be very close together in space because of 3D DNA curvature. This spatial proximity is very important in biological mechanisms such a co-regulation one. Thanks to 3D conformation models, we are able to construct the 3D trajectory of a DNA molecule given its textual sequence. Some software as DNATools (SEQTools, 2001) represent 3D DNA trajectory but they are limited up to several hundreds of nucleotides. Others, as RasMol (University of Massachusetts, 1995) or Swiss-PDB Viewer (Guex, 1996), display very well data contained in PDB (RCSB, n.d.) files and do not construct any trajectory. In fact, a DNA sequence contains a huge number of nucleotides: bacterial or archaeal chromosomes contain between 1 and 10 million nucleotides, while eukaryotic ones (animals, plants, humans, and so on) usually contain several million up to several hundred million nucleotides. In this context, the visualization of a whole chromosome requires a lot of graphic and memory resources. Our main aim was to design a powerful and *user-friendly* visualization software tool that would render whole molecules in real-time, taking into account such data mass.

The 3D information, from model construction, could either be visualized or processed further. The 3D visualization, contrary to the textual one, offers a global view of the molecules and is achieved through software called *ADN-Viewer* (Gherbi & Hérisson, 2002) developed at LIMSI-CNRS. Currently, *ADN-Viewer* can load and visualize multiple sequences of tens of million of nucleotides (depending on memory size). It is possible for the user to identify various DNA zones that have either compact or relaxed properties. *ADN-Viewer* is also linked to a genomic database containing data on all currently sequenced and annotated living organisms. This database gives us additional information about gene names, gene locations and other genetic objects within a chromosome. In addition, *ADN-Viewer* is also available within an immersive virtual environment, which provides stereoscopic visualization on large screens as well as a *user-friendly* 3D interface that uses powerful navigation techniques such as gesture and speech recognition, and so on. Such a multimodal human-computer interface offers intuitive use and is very suitable when processing and managing huge amounts of data, as in the case of complete chromosomes.

In this paper, we will first describe some elementary concepts about genomic data, in particular DNA sequences (Section 2). Then, a review of the various computer-based representations will be proposed in Section 3. Section 4 will describe our virtual modeling of DNA and its multiple representations. In this section, we will also propose some new algorithms that deal with scene management that handle and visualize, at different levels, large chromosomes in real-time. Section 5 will present content-based exploration techniques for annotated DNA, in particular gene information. Finally, in Section 6, we will show that a large immersive environment can be very suitable for complex data analysis in the context of scientific visualization.

2. DNA *IN VIVO*

DNA molecules are composed of four elementary molecules (or nucleotides), represented by four letters: A standing for *Adenine*, C for *Cytosine*, G for *Guanine* and T for *Thymine*. Moreover, DNA has a double-helix structure and is formed of two complementary strands (Watson strand and Crick strand) (Watson & Crick, 1953): when a *A* nucleotide is on one strand, a *T* nucleotide must be on the other strand, and when a *C* nucleotide is on one strand, a G nucleotide must be on the other one. Two complementary nucleotides are linked by hydrogen bonds and thus form a base pair (bp) or a plate. A textual DNA sequence represents only one of the two strands.

3. DNA *IN SILICO*

3.1 3D Engine

The 3D engine of our *ADN-Viewer* software takes both textual DNA sequences and the 3D conformation model as input, and it then outputs the 3D co-ordinates of each nucleotide. The 3D conformation model, established by Shpigelman, Trifonov & Bolshoy (1993), provides, for each dinucleotide (i.e. each succession of 2 nucleotides in the textual sequence), three angular values and a raise translation (Table 1). The first plate (base pair of two nucleotides) is placed at the origin of the graphical scene. By scanning the textual sequence in a linear way, *ADN-Viewer* computes the position and orientation of one plate, by applying Eq. (1) on the previous plate’s position and orientation (Figure 1).

Table 1. Rotation angles of the 3D conformation model (in degrees). Each dinucleotide has 3 angle values associated with it. This set of values has been established from several experimental studies.

Dinucleotide	Twist (Ω)	Wedge (σ)	Direction (δ)
AA	35.62	7.2	-154
AC	34.4	1.1	143
AG	27.7	8.4	2
AT	31.5	2.6	0
CA	34.5	3.5	-64
CC	33.67	2.1	-57
CG	29.8	6.7	0
CT	27.7	8.4	-2
GA	36.9	5.3	120
GC	40	5	180
GG	33.67	2.1	57
GT	34.4	1.1	-143
TA	36	0.9	0
TC	36.9	5.3	-120

TG	34.5	3.5	64
TT	35.62	7.2	154

$$M_{xy} = T(-\frac{h}{2}) * R_z(\frac{\Omega}{2}) * Q(\sigma, \delta - 90) * R_z(\frac{\Omega}{2}) * T(-\frac{h}{2}) \qquad (1)$$

This equation defines a predictive algorithm for 3D DNA sequences. *T* is the vertical translation along *Z*-axis, *h* is the value of this translation (here *h*=3.39 Å). R_n is the rotation around *N*-axis and $Q(\alpha, \beta) = R_z(-\beta)*R_x(-\alpha)*R_z(\beta)$.

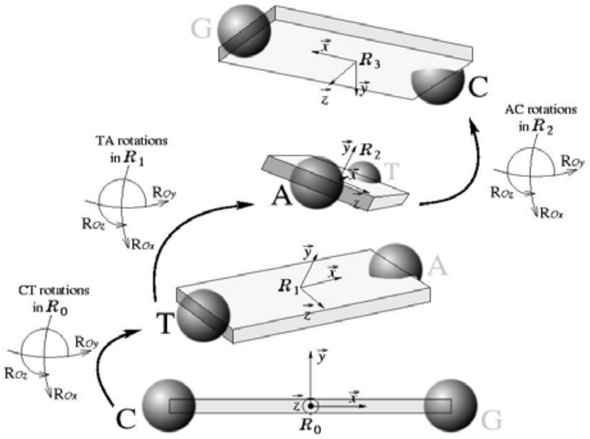

Figure 1. Construction of the DNA's 3D trajectory.

3.2 Data Storage

For performance reasons, we have to store the 3D chromosome trajectory resulting from the previous computation. Besides, in order to perform quantitative processing (compactness, curvature…) on these trajectories, we need to know the spatial position and orientation for each plate. We use a 4x4 transformation matrix to represent orientation and translation information. It means that if we store a transformation matrix (4x4 double precision floating point) for each plate, a sequence of 10 Mbp needs $10e^6$ bp * 4*4 * 8 bytes = 1.28 GB of memory. Nowadays, this is possible if we use PC workstations, with 2 to 4 Gb of memory. But this is not sufficient for bigger DNA sequences: for instance 100 Mbp represent about 13GB of 3D data. Indeed, in order to be biologically pertinent, we have to keep high data resolution. An initial solution to overcoming such a problem consists of storing, for each plate, a 3D point (3 double precision floating points) for position information, and a quaternion (Hamilton, 1843) (4 double precision floating points) for the orientation one. With this kind of data structure, a sequence of 10 Mbp only needs $10e^6$ bp * (4*8 bytes + 3*8bytes) = 560 Mb of memory. Thus, on a computer that has 1 Gb of memory, we can load a sequence of about 20 Mbp. Presently, *ADN-Viewer* can load about *n*/50 Mbp, where *n* is the memory size.

4. DNA *IN VIRTUO*

ADN-Viewer offers 3 types of 3D DNA sequence modeling. The sequence may be visualized as a whole, partially (for instance a gene) or at a very local scale by displaying all atoms and atomic links of the considered nucleotides. The programming language used is C++, the GUI is managed by Qt (Trolltech, n.d.), while graphics are processed by pure OpenGL® code. *ADN-Viewer* can be processed on Linux, UNIX (IRIX®) and Windows® platforms.

4.1 Visualization at the Chromosome Scale

As we have already said, visualization of a whole chromosome offers us a global point of view of the DNA molecule. The user can identify different areas, some of them show characteristics of compactness while others exhibit more relaxed ones (Figure 2).

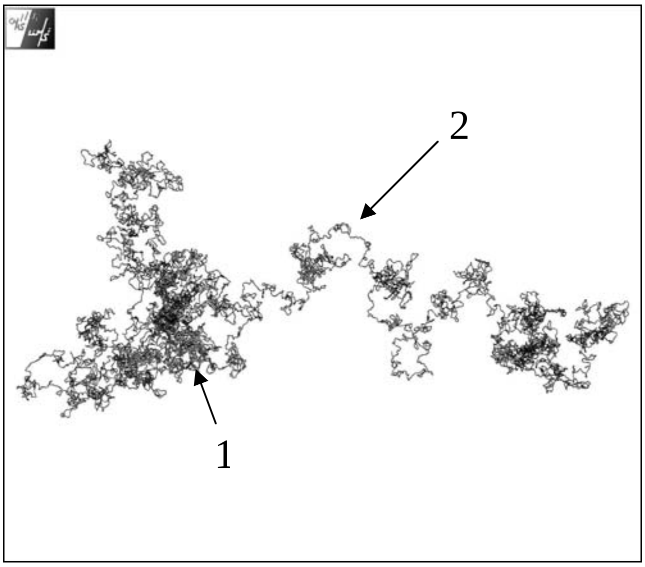

Figure 2. Genomic visualization of *S. cerevisiae chrIII* (~300 Kbp). Area 1 is compacted DNA while area 2 is relaxed DNA.

In this genomic modeling, we only display the segments linking successive plates. Nevertheless, with a chromosome of 10 Mbp, 10 million segments per frame have to be displayed. In addition, about 15 frames per second (fps) are necessary for visual fluidity, even though it is well-known that a 25 fps rate is better. Consequently, 10.10^6 Mbp * 25 fps = 250.10^6 segments per second have to be displayed. It is not possible to attain such a performance, even with powerful graphic cards. We will now describe how we have tackled this problem by proposing a new filtering algorithm.

4.1.1 Scene management

When a scene contains a lot of objects, it is difficult to carry out real-time rendering. In order to limit the amount of displayed data, the idea is to display a lot of object detail when the latter is close to the user, and to display a minimum of information when the object is far from him or hidden by another object. To implement these mechanisms, tree structures are often used. In fact, two kinds of scenes exist: hierarchical ones (for instance urban scenes) and non-hierarchical ones (for instance point clouds). For hierarchical scenes, tree leaves are atomic (basic) objects, nodes are subobjects and the root is the complete object. For non-hierarchical scenes that contain several objects, tree leaves represent objects and nodes represent geometric transformations or effects (lights and so on); in non-hierarchical scenes containing only one object (such as point clouds) tree leaves and nodes represent squares of scene volume. Tree structures are very useful because each object or scene square can be managed independently and are very powerful because we can eliminate the graphical rendering of entire subtrees. For non-hierarchical scenes, particular algorithms divide the scene into squares to form structures such as an octree (Wilhelms & Gelder, 1990), BSP tree or K-D tree and so on. Such algorithms are very efficient but are unfortunately very expensive computing-wise.

A scene that contains a 3D DNA sequence is different from those mentioned above. It is more akin to a points cloud rather than to an urban scene. Nevertheless, while the data are less dense than in an isosurface, the performance gain between the computational time of scene tree and rendering time is strongly reduced. On the other hand, a 3D DNA sequence is a linear strand that constitutes a volume. This means that each 3D point of a DNA sequence is only connected to the previous point and to the next one. So it is only connected to its linear next door neighbors whereas in an isosurface, each point is connected to its spatial neighbors to define a little triangle or a quad. In order to design a powerful scene management algorithm, the idea was to use the wire-like structure of a 3D DNA sequence to filter displayed data.

Uniform filtering provides poor rendering performances because of the DNA sequence's 3D topology. Consequently, we have to apply non-uniform filtering. The main idea is to detect a focused area "on the fly" during the filtering adaptation. For each point displayed, the filter selects which point to display next by performing this linear function: $d/750$, where d is the distance between the user and the current point, and where the constant 750 has been determined experimentally. In this way, the closer an area is to the user, the more points to display it will contain; and the further away an area is from the user, the fewer points to display it will contain. At each user movement, the filtering computes a sample step before displaying the chromosome. The gain is difficult to evaluate because it depends if the user is close to a densely focused area or if he is far from any part of the chromosome. From experience, no visual distortions were observed and all DNA sequences were rendered in real-time, even 20 Mbp molecules.

4.1.2 Optical distortions

Due to the projection of an 3D object onto a 2D screen, some undesired optical effects may appear. By applying a fog effect, depth perception is augmented and the user has a better view of the chromosome's volume. However, even with a fog effect, the exact perception of a 3D geometrical feature of a DNA sequence is very difficult, especially because of the 2D projection of a 3D object. To overcome such an optical effect caused by 2D projection, the *ADN-Viewer* has a multiview interface to enable the DNA sequence to the observer from different angles of view (front perspective view, and front, left-side and under orthogonal ones). In Section 6, a more interesting solution based on stereoscopic visualization onto large screens is described.

4.2 Visualization at the Gene Scale

It is possible for the user to select a part of a chromosome (such as gene) in order to observe the double-helix of this part (Figure 3).

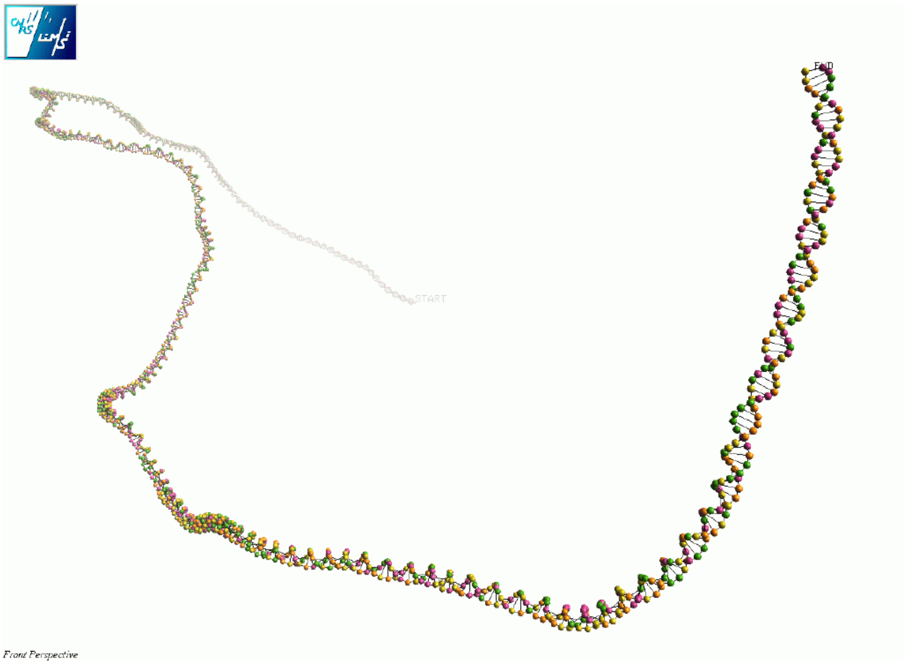

Figure 3. Visualization of a gene (~500 bp) with genic modeling. Each nucleotide is represented by a colored sphere.

The non-uniform filtering described in Section 4.1.1 is again applied here to select which plate will be displayed. In addition, fog effect also increases the 3D perception of the gene structure.

4.3 Visualization at the Atomic Scale

Nucleotides are composed of atoms that are not visible in the previous modeling. However, the user can select a small part of a gene and observe its atomic structure (Figure 4).

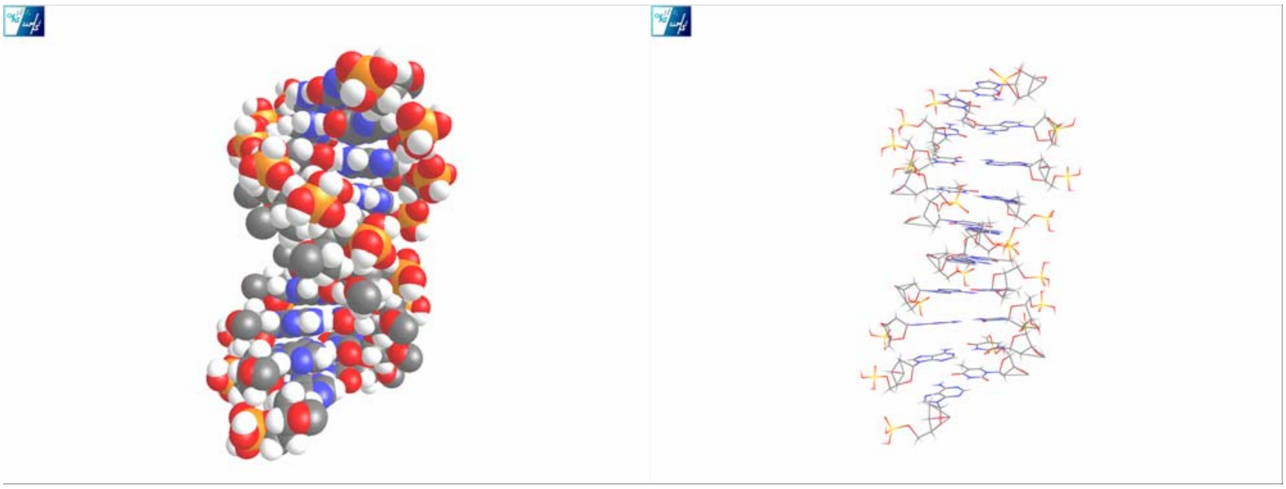

Figure 4. On the left, the full nucleic representation of part of a gene (10 bp). On the right, its wired-like nucleic representation.

Each DNA nucleotide is composed of several linked atoms, and the 3D co-ordinates of each atom are known exactly (from X ray experiments). From these biophysical experiments, complete 3D information is obtained and stored in a standard PDB file, which contains the 3D co-ordinates of every atom. Thus, we have the position and orientation of each nucleotide, geometric transformations are then applied and the plate is displayed.

Even if we only observe small sequences (10 up to 20 bp) in the full nucleic modeling, there are many geometric objects to display. One plate alone has about 70 spheres intersecting with one another. For good modeling, spheres are constructed from at least 500 triangles which therefore represents about 35000 triangles per plate. With a 10 bp sequence for example, we have to display 35000 triangles * 10 bp * 25 fps = 8.75e^6 triangles per second. Again, it is very difficult to obtain satisfactory performances with current graphic cards. Consequently, as spheres intersect with one another, we have to eliminate the hidden triangles. Because the positions, centers, radius and atomic links (we use chemical structure of a plate) are all known, we can apply the algorithm described below and illustrated by Figure 5:

```
All triangles are initially considered as hidden
for each plate $p$
  for each sphere $S_0$
    for each sphere $S_1$ connected to $S$ by an atomic link
      for each triangle $t$ of sphere $S_0$
        for all vertices $v$ of triangle $t$
          if $d(v, C_1) > R_1$ then the triangle $t$ is visible
```

where C_i and R_i are respectively the center and the radius of sphere i, and $d(P_0, P_1)$ is the Euclidian distance between the 3D points P_0 and P_1.

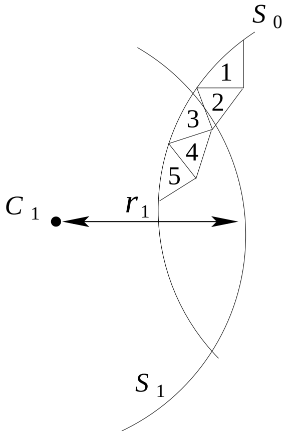

Figure 5. Depiction of hidden triangles of sphere S_0. In this case, triangles 1, 2 and 3 will be visible because all or some of their vertices are outside sphere S_1. Triangles 4 and 5 will be hidden because all of their vertices are inside sphere S_1.

All hidden triangles are deleted from the data structure, thanks to information indicating which triangle is visible and which is not. Because only four kinds of plate (A-T, T-A, C-G and GC) exist, the algorithm is thus applied in the initialisation phase. Hence, we obtain about a 30 percent reduction of computed data, which represents approximately over $5e^6$ triangles per second to display a 10 bp sequence.

Because our topological database is static, we can optimize the rendering. With recent *n*Vidia® graphic cards, we can use OpenGL® (Neider, Davis & Woo, 2001) extensions (**glLockArraysEXT** and **glDrawElements**) that store 3D data directly in video memory and compile structures to optimize the render time. With these extensions, on a scene that contains 35,000 triangles, we can obtain a gain of 30 per cent in comparison with previous performances.

5. GENOMIC CONTENT-BASED EXPLORATION

A trajectory is the basic information about a DNA sequence. However, a chromosome can be modeled according to its collection of genes and intergenic areas. Moreover, several biological signals, patterns or motifs constitute the functional content of these chromosomes. We will describe how we represent this type of information by offering the user genomic content-based exploration.

5.1 Interfacing the ADN-Viewer with a Genomic Database

We have developed a genomic database called *GenoMEDIA* (Gros, Férey, Hérisson & Gherbi, 2004) that stores annotated DNA sequences containing information about DNA sequences and gene positions and names. Thanks to SQL queries, the *ADN-Viewer* can download any requested annotated sequence in order to augment the visualization (Figure 6). Such augmented representation is very useful for biologists when performing supplementary genomic studies. For example, two distant genes in a textual DNA sequence could be close together in 3D space, due to the DNA folding up.

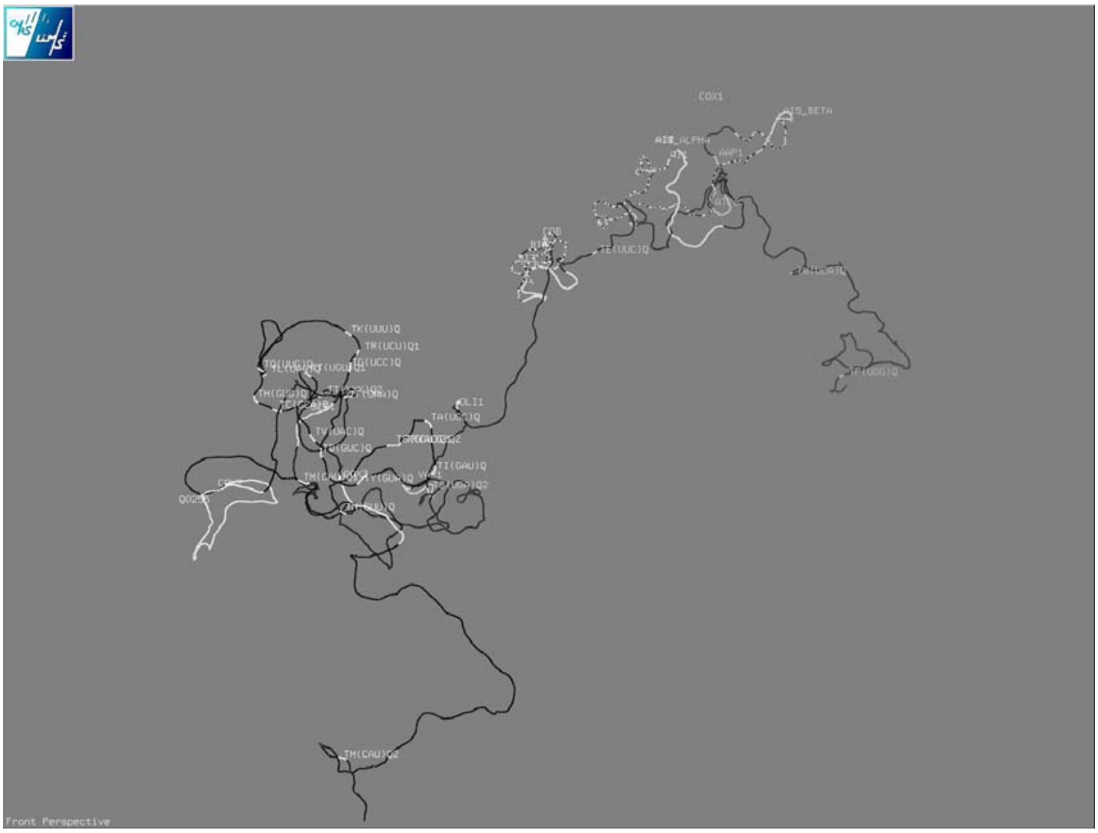

Figure 6. 3D Visualization of *Scerevisiae chrMT* (~50 Kbp). Genes are displayed in white and intergenic areas are in black.

The filtering process described in Section 4.1.1 does not take into account the gene positions within the chromosome. In particular, important information about where a gene starts and stops could be not displayed. Genes are displayed in a different color than intergenic areas. So filtering provides a graduation of color at gene positions, which can distract the user when he wants to observe exact gene positions during genomic visualization. To overcome this problem, we must force the display of the first and last point of each gene. In addition, with this representation, the user can designate and select any gene by mapping a mouse pointing in 3D space (Figure 7).

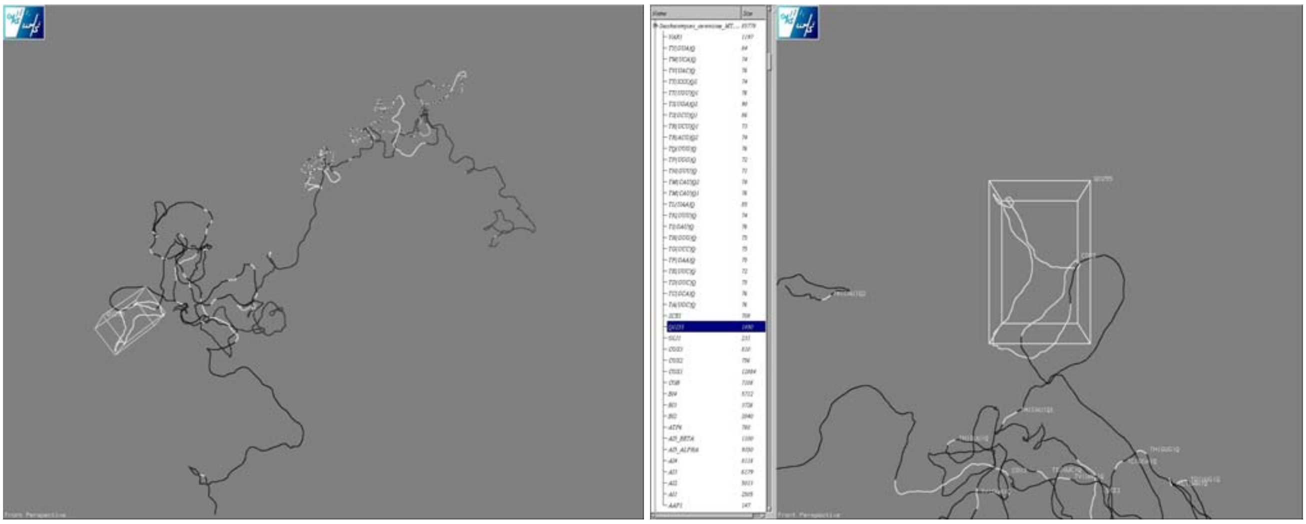

Figure 7. Gene designation by mouse pointing (on the left) and selection by mouse clicking (on the right).

On the request of biologists, the *ADN-Viewer* now offers the possibility of displaying several chromosomes at the same time. This functionality is useful when comparing all the chromosomes of a single eukaryotic genome (17 for *Saccharomyces cerevisiae*), or several chromosomes from different organisms. Each chromosome has its own modelview matrix and can thus be manipulated in an independent manner from the others and display additional information. This multi-chromosome visualization also allows a bioinformatic comparison of chromosomes in terms of compactness or other geometrical features. However, this visualization is not exploitable on a desktop screen and so genomic analysis is unavailable to biologists. Two limitations can be listed: firstly, a classical desktop screen has a very limited visualization space, and secondly, even if the objects are modeled three dimensionally, only 2D projections are perceived on these screens. Real 3D perception could help us to overcome such limitations.

6. IMMERSIVE EXPLORATION

Of course, we have some 3D perception when objects move (manipulations). Nevertheless, the main perceptual tool that provides 3D visualization is a stereoscopic one. Human beings obviously perceive real 3D objects, thanks to binocular vision. So, when the user observes 3D objects on 2D screen, depth perception is appreciably affected. Besides, if the user interacts with objects using classical devices (e.g. keyboard and mouse), he cannot manipulate these 3D objects easily. Classical human-machine interaction paradigms are well adapted for desktop environments. However, they are useless when 3D objects are manipulated in 3D space. Stereoscopic mechanisms and large visualization areas make it possible to significantly decrease the gap between the virtual world and the real one. For these reasons, the *ADN-Viewer* was integrated within the LIMSI Virtual Reality platform (LIMSI-CNRS, 2002). This immersive platform is made up of two components: the hardware and the software architecture. For the hardware, we use two rear-projected orthogonal screens (2mx2m in size) and a SGI Onyx2® as graphical computer. This hardware device is managed by middleware called *EVI3D* developed at LIMSI-CNRS (Touraine, Bourdot, Bellik & Bolot, 2002). This middleware includes two different parts. The first part handles the geometric kernel that manages the model's views and projections on screens. The second part (*VEServer*) manages various 3D and immersive devices such as the data glove for gesture recognition, a speech recognition system, a 3D position and orientation tracking system, an immersive mouse, and so on. DNA chromosome exploration and perception are significantly increased (because of large screens, stereoscopy, 3D human-machine interfaces, multimodal interaction and so on) allowing the user to completely focus on his main task. Another potentially powerful interaction is *via* the interfacing of the vehicle paradigm *HCNav* (Bourdot & Touraine, 2002) with *ADN-Viewer*. It consists of navigating in the virtual world hands free. Thanks to tracking head position and orientation (captured by Flock of Birds® fixed on stereoscopic glasses), the user can drive the vehicle in the virtual world with his head and body movements. This tracking system is also used to dynamically adapt the stereoscopic view according to the user's position and orientation. Such navigation offers us two advantages: the user does not need to use the mouse to observe a chromosome any longer, and he can use his hands for other tasks (Figure 8) such as gene designation or selection.

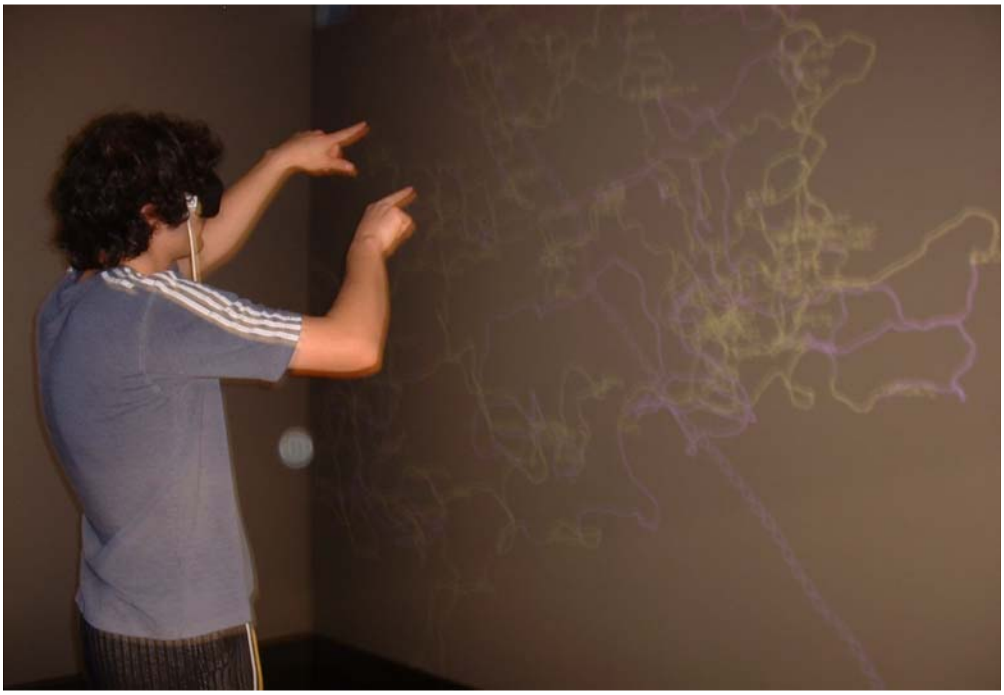

Figure 8. *HCNav* system allows free-hand interaction.

For other kinds of interaction (grasping, selection and so on), many other devices could be used, such as speech commands or data gloves for objects manipulation. The *ADN-Viewer* also offers the possibility of using a 3D pointing device called the WANDA$^{\mathbf{TM}}$. It has proved very useful in drawing a virtual ray that has as its origin the WANDA$^{\mathbf{TM}}$ and for direction the orientation of the device, designates or selects any object while navigating.

7. CONCLUSION AND FUTURE WORK

In this paper, we addressed the potential offered by Virtual Reality and scientific simulation for 3D modeling and the immersive visualization of large genomic sequences. Advanced work on 3D data modeling and structuring was proposed. The representation of DNA's 3D structure allows biologists to observe and analyze genomes in an interactive way at different levels: from gene to chromosome. We developed a powerful software platform (*ADN-Viewer)* that is used by biologists for sequences analysis. This software manages huge amounts of data in real-time thanks to new scene management algorithms and uses immersive human-computer interactions to enable *user-friendly* data exploration. The *ADN-Viewer* is currently being validated by CNRS in order to make it available for the bioinformatics community.

In the future, further work will deal with the modeling and visualization of a fully annotated DNA sequence (genes, promotors, enhancers, transposons, introns, exons and so on). The problem that remains to be resolved is how to visualize 3D information (the DNA double-helix) with textual annotations that are not 3D (names, biological origin, function and so on). In an immersive environment, all visualized objects have to appear in 3D to stop the user from alternating between flat 2D text and 3D objects and to reduce his cognitive load.

Future work would also include investigating the use of a PC-based graphical cluster to display very large DNA sequences (several hundred millions of base pairs, for plants, animals and human organisms). To achieve a smart collaboration between the cluster nodes, research in databases and distributed application domains will be required.

8. ACKNOWLEDGMENTS

We would like to thank P. Bourdot, J.M. Vézien, T. Convard and D. Touraine (from VENISE team at LIMSI-CNRS) for several fruitful discussions about Virtual Reality and Bioinformatics and collaborations for interfacing *ADN-Viewer* with *EVI3D*. We also are grateful towards the PPF Bioinformatics project of Paris-Sud University for its grant support.

9. REFERENCES

Bourdot, P., & Touraine, D., (2002) Polyvalent display framework to control virtual navigations by 6DOF tracking. *IEEE Virtual Reality*, Orlando. Florida, USA.

Gherbi, R., & Hérisson, J., (2002) Representation and Processing of Complex DNA Spatial Architecture and its Annotated Content. *International Pacific Symposium on Biocomputing*. Lihue, Hawaii, USA.

Gros, P.E., Férey, N., Hérisson, J, & Gherbi, R., (2004) GenoMEDIA, a Midlleware Platform for Distributed Genomic Information. *IEEE International Conference on Information & Communication Technologies: from Theory to Applications*. Damascus, Syria.

Guex, N., & Peitsch, M. C. (1996) Swiss-PdbViewer: A Fast and Easy-to-use PDB Viewer for Macintosh and PC. *Protein Data Bank Quaterly Newsletter 77*, 7.

Hamilton, S. W. R. (1843) On a new species of Imaginary Quantities connected with the Theory of QUATERNIONS. *Irish Academy Proceedings 2,* 424-434.

LIMSI-CNRS (2002) VENICE Transversal Action V&AR. Retrivied June 10, 2005 from the world wide web: http://www.limsi.fr/Recherche/ActionVenise/.

Neider, J., Davis, T., & Woo, M. (1993) *OpenGL Programming Guide* (3rd Edition, 2001). Massachusetts, USA: Addison Wesley.

RCSB (n.d.) Homepage of Protein Data Bank. Available from http://www.rcsb.org/pdb.

SEQTools (2001) Homepage of SEQTools. Available from http://www.dnatools.org.

Shpigelman, E. S., Trifonov, E. N., & Bolshoy, A. (1993) CURVATURE: software for the analysis of curved DNA. *CABIOS 9*, 435-440.

Touraine, D., Bourdot, P., Bellik, Y., & Bolot, L. (2002). A framework to manage multimodal fusion of events for advanced interactions within virtual environments. *Workshop on Virtual environments* (pp. 159-168). Barcelona, Spain.

Trolltech (n.d.) Homepage of Trolltech. Available from http://www.trolltech.com.

University of Massachusetts (1995) Homepage of Rasmol. Available from http://www.umass.edu/microbiol/rasmol/index2.htm.

Watson, J. D., & Crick, F. H. C. (1953) Molecular Structure of Nucleic Acids. *Nature 171*(4356), 737-738.

Wilhelms, J., & Gelder, A. V. (1990). *Octrees* for Faster Isosurface Generation. *Workshop on Volume visualization* (pp. 57-62). San Diego, California, USA.