From Visualization to Manipulation of RNA Secondary and Tertiary Structures

Gilles Bailly, David Auber, Laurence Nigay
CLIPS-IMAG lab, University of Grenoble 1, LaBRI, University of Bordeaux
385 rue de la Bibliotheque, 38041 Grenoble Cedex 9, 351, cours de la Liberation, 33405 Talence
FRANCE
gilles.bailly@imag.fr, david.auber@labri.fr, laurence.nigay@imag.fr

Abstract

Ribonucleic Acid (RNA) is an important molecule which performs a wide range of functions in biological systems. We present a method for visualizing, exploring and editing RNA molecules. The method relies on the visualization of the 3D tertiary structure of RNA as well as on the automatic extraction and visualization of the 2D secondary structure. The 2D representation is used to navigate in the tertiary structure. The method has been implemented in a multi-surface interactive system, NAVRNA, that allows collaborative analysis of RNA.

Keywords— **bioinformtics, graph drawing, multi-surface interaction, multiple cursors, two-handed interaction.**

1 Introduction

Ribonucleic Acid (RNA) is an important molecule which performs a wide range of functions in biological systems. RNA is found in the nucleus (where it is synthesized) and in the cytoplasm, as messenger RNA -mRNA- (which carries the genetic information out of the nucleus), transfer RNA -tRNA- (which decodes the information), ribosomal RNA -rRNA- (which is found in the ribosome of cells). These forms of RNA are involved in the protein synthesis. RNA recently became the centre of much attention because of its catalytic properties, leading to an increased interest in obtaining structural information. For example, RNA contains genetic information of viruses such as HIV and therefore regulates the functions of such viruses.

A RNA molecule is a linear polymer in which the monomers -(ribo)nucleotides- are linked together by means of phosphodiester bridges or bonds. Each nucleotide contains a base. The four different bases are Adenine (A), Cytosine (C), Guanine (G) and Uracil (U). Although each RNA molecule has only a single polynucleotide chain, it is not a simple linear structure. Indeed it contains extensive regions of base pairs. Thus, a RNA molecule is characterized by its base sequence and higher order structural constraints. It is usually considered at three levels of detail by biologists:

- the **primary structure**: the linear polymer,
- the **secondary structure**: the linear polymer and a sub-set of bridges of the whole structure,
- the **tertiary structure**: the structure in its entirety.

Crystallography makes it possible to determine the nature and the exact three-dimensional position of each nucleotide that composes a RNA molecule, in other words the tertiary structure. Nevertheless the size of manipulated data space and the occlusion due to the folding up of the molecule in space make the study directly on the tertiary structure very difficult. Our system NAVRNA consequently defines a 2D view of RNA (i.e. the secondary structure) that can be manipulated and edited by the users. The two views (the secondary and tertiary structures) are therefore complementary and visual consistency is maintained at all time. The users can therefore manipulate the secondary view in order to explore the entire structure. Moreover to take into account the biologists habits, getting used to work on specific manual 2D drawing, we propose an efficient tool that enables interactive modification of the 2D representation for obtaining a drawing close to the usual manual drawing. In this paper, we first describe the interactive setup of our NAVRNA system. We then focus on the visualization processes of the tertiary and secondary structures of RNA. We finally describe the interactive techniques for manipulating the views and especially the bending technique of the 2D representation.

2 NAVRNA: Interactive setup

NAVRNA is an interactive system for manipulating RNA structures. As opposed to immersive systems [1, 13, 10], NAVRNA keeps the users in the real world and interaction is based on augmented surfaces. The key characteristics of our setup (fully described in [4]) include multi-surface visualization, collaboration among several users and two-handed interaction.

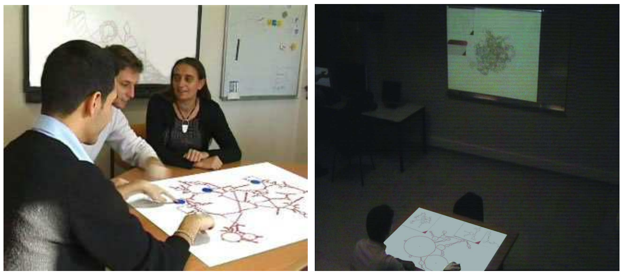

Figure 1: Multi-surface, multi-user setup

Visualization. The size of manipulated data is significant (several thousands of nodes for each of the two views). We thus chose to use visualization surfaces much larger than the traditional 19 inch screens. We define a distributed display environment made of (1) a table on which the secondary structure is projected by a video projector fixed on the top of the table and (2) a whiteboard on the wall on which the tertiary structure is projected by a second video projector as shown Figure 1.

Collaboration. This environment enables co-located synchronous collaborative manipulation of the RNA structures by several biologists. Interaction on the table is based on the magic table [5] and consists of the manipulation of tokens that are tracked by a vision-based mechanism. A tracking video camera is therefore fixed on top of the table, whose field of view is set to encompass the entire projected image on the table.

Two-handed interaction. A salient property of the magic table is to allow each user to interact with two hands as shown in Figure 5. In this version of the magic table, tokens are all the same and their blue color facilitates the tracking mechanism. It returns a different identifier to the application for each token posed on the table. We plan in the near future to experiment with a new version of the magic table that is based on finger tracking instead of using tokens [24].

Apparatus and toolkits. Our interactive system NAVRNA uses two video-projectors, a camera, two laptops with a 2Ghz Pentium 4 processor and a NVIDIA GeForce FX Go 5200 graphic card with 64 MB of memory. NAVRNA is developed in C and C++. For drawing the secondary as well as the tertiary structure we use the Tulip toolkit dedicated to large graph visualization [2] and its extensions for PDB files [37]. It is built on GLUT, the OpenGL Utility toolkit and runs on Linux platform.

3 Visualization of RNA representations

In this section, we describe how we define the different representations of RNA. The key point is how we obtain the secondary structure representation that is close to the usual manual drawing by the biologists.

3.1 Visual mappings

The three dimensional coordinates of nucleotides are given by crystallography (extracted directly from the PDB file). However, the obtained 3D structure is completely folded up on itself. For the exploration of such a structure we propose to unfold it by using a substructure that can be drawn in 2D. We compute a 2D layout of a substructure of the whole RNA. By considering nucleotides as nodes and links between nucleotides as edges, we return to the problem of graph drawing. To do so, in NAVRNA, we use the algorithm first proposed by Auber [3]. This algorithm works in three steps:

- Step1: The first step consists of building a tree that represents the original graph.
- Step 2: The second step draws the tree.
- Step3: The third step computes the final layout using the coordinates of nodes obtained at the second step.

The main feature of the algorithm is to produce a layout of a secondary structure that is close to the one drawn manually by the biologists. Figure 2a shows a manual drawing highlighting aesthetic criteria used by the biologists. The loops are drawn as circles, the stems are drawn as straight lines, the edge length is constant and the drawing is planar. Figure 2b shows the automatic drawing of the RNA presented in Figure 2a. The algorithm is quadratic in the number of nucleotides, enabling us to draw in few seconds the largest RNA we found in the existing database (about five thousands nucleotides).

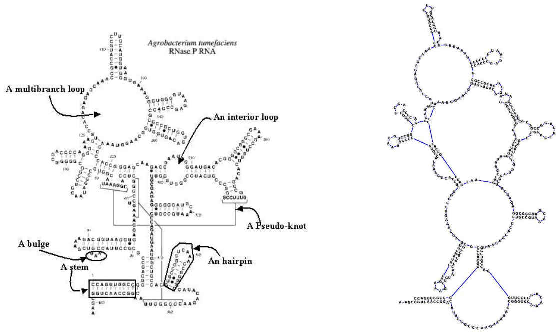

Figure 2: (a) Manual drawing, (b) Automatic drawing

Several algorithms enable us to draw the secondary structure of RNA [25, 3, 9]. In this work we have chosen Auber's one because it produces planar drawings in quadratic time that are very close to the one expected by the biologists. Thus, it can be used interactively to visualize several secondary structures extracted from a PDB file. Even if the drawing is close to the one expected by the biologists, editing operations are necessary to obtain a perfect layout. Based on meetings with experts , we concluded that they become used to working on a very specific representation. Directly obtaining such a representation is still an open problem. Moreover some RNA representations have been drawn by hand and intensively used for several years. Many RNA specialists use such drawings. Thus it is part of a visualization culture. So changing the way these RNA are represented will deeply affect the performance of the experts and finally the tool may not be used. To take into account these requirements, we have developed an efficient way to edit the drawing obtained by Auber's algorithm. The users are therefore able to rearrange the representation to obtain a layout close to their work habits. The deforming technique, described in section 4.2, enables the user to interact on the representation resulting from Step 2 of Auber's algorithm. After editing the tree layout, we then run Step 3 of Auber's algorithm in order to obtain a final layout. Due to the construction of the tree, the manipulation is very natural for the user and we do not need to display the underlying tree during interaction.

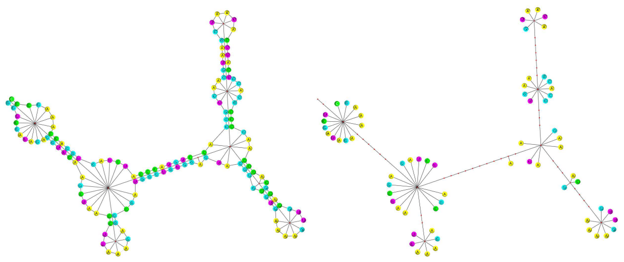

Figure 3: (a) RNA + Tree, (b) Tree

In this work, we apply our tree deformation method on RNA. However, such a method can be applied on other data. Indeed our method is based on the fact that we can extract a tree from the original graph that is then used to draw the final graph. Figure 3 shows a tree structure that we extract from a graph. The construction method consists of creating a new node for each face of the original graph and of linking two nodes if the corresponding faces, that they represent, share an edge in the original graph. Then if a node of the original graph belongs to only one face (we do not consider the outer face), we link it to the node that represents that face. While being different, our construction is close to that of a dual graph. It can be proven [2] that the construction produces a tree for any sub-cubic outer-planar bi-connected graph, thus our method can be applied on such graphs.

3.2 View transformations

We therefore have two visual structures, the secondary and tertiary structures. We enable the user to visualize them at the same time within the multi-surface environment described in section 2. When the secondary structure is displayed, an overview is also displayed in order to always keep the context. Moreover, it is possible to visualize several focus views to be kept apart in a separate window. As biologists would always keep the global view of the tertiary structure, no extra 3D views are useful.

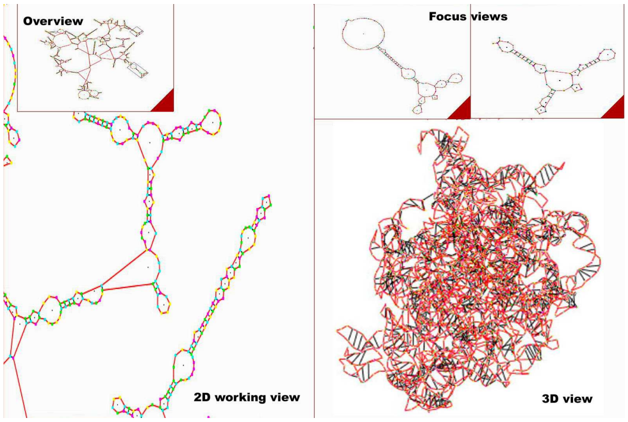

Figure 4: Different views of RNA. On the left, the 2D working view and the overview displayed on the table. On the right, the focus views and the 3D view displayed on the wall.)

4 User interaction

In the previous section, we explained the visualization process and we presented the five different views obtained from a PDB file (2D working view, overview, focus views and 3D view). We now detail how the users interact with them. The three key features are: (1) two-handed interaction for an efficient exploration of RNA, (2) the manipulation of the secondary structure for exploring the tertiary structure, and (3) an efficient collaborative technique for deforming the RNA. In the following sections, we distinguish user's actions that imply view transformations, from the ones that involve view mappings.

4.1 View transformations

Pan, zoom and rotate actions are performed on the views by manipulating tokens on the table as shown in Figure 5. With one hand, a panning action can be performed by sliding a token. With two hands, the users can zoom and rotate the view displayed on the table (distance between tokens defining the zoom factor). If the user is dragging the two tokens in the same direction at the same speed, a natural

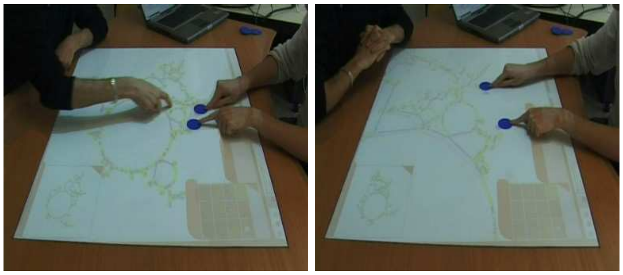

Figure 5: Two-handed interaction based the manipulation of tokens: rotate and zoom.

compound panning action is performed. By continuously manipulating two tokens, the user can therefore perform pan-zoom-rotate actions without tool switching: such two-handed interaction therefore facilitates a rapid exploration of the visualized secondary and tertiary structures.

All the displayed structures are strongly linked together. Indeed a rotation of the secondary structure implies the rotation of the overview, the focus views, and the tertiary structure. But when users zoom the secondary structure displayed on the table, it has no effect on others views and in particular on the tertiary structure.

Moreover the users can manipulate a transparent window with two hands for specifying a zone of the 2D structure displayed on the table: the corresponding zone of the 3D structure displayed on the wall will then be highlighted as shown in Figure 6. For example: " On the one hand, it could be very useful to locate in the 3D structure a particular pattern recognized in the secondary structure. " On the other hand, it could be useful to select two branches in a zone of the tertiary structure to show to a student for instance that the two branches are actually very far from each other in the secondary structure.

In conclusion, although the user can directly interact on the tertiary structure, we mainly designed the interactive system NAVRNA so that the user manipulates the secondary structure displayed on the table in order to explore the tertiary structure displayed on the wall. Facing the difficulties of directly exploring the 3D structure, the user manipulates a 2D structure that is not simply a projection of the 3D structure but a complex unfolding of the RNA molecule.

4.2 View mappings

One of the main advantages of the algorithm [3] presented in section 3.1 is that it draws the secondary structure automatically according to some of the biologists' aesthetic criteria. In order to fill the gap between this automatic drawing and ten years of manual drawing, we define an interactive technique for deforming the secondary structure and thus allowing the biologists to rearrange the drawing and obtain a perfect drawing close to their work habits. In addition deforming or bending RNA branches may help to understand the overall structure by putting next to each other RNA chains that are close in the tertiary structure.

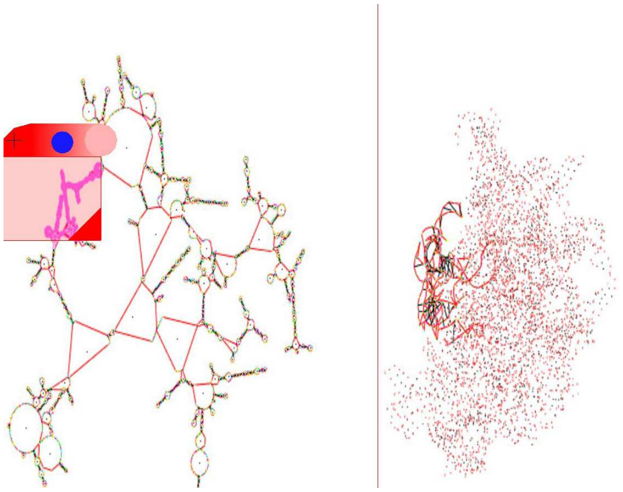

Figure 6: With a transparent window, the user can select a zone on the secondary structure. The corresponding selected zone of the tertiary structure is then highlighted.

For deforming the secondary structure, the users select one or several nodes using tokens, and by dragging them deform the entire structure while the edge size is kept constant.

Our corresponding algorithm for deforming RNA works in two steps: (1) The first step consists of deforming the internal tree of the RNA structure calculated at Step 1 of Auber's algorithm (section 3.1). (2) The second step consists of computing the final layout from the deformed tree (i.e. Step 3 of Auber's algorithm). More generally speaking, our algorithm consists of deforming a tree while preserving constant the size of the edges. This is a generalization of the algorithms [32, 22] which use only two nodes and one edge between them. We first describe our deformation algorithm and then its extension to allow several anchors for deformation.

We consider a tree T = (V,E), where V is a set of nodes and E a set of undirected edges $(a, b) = (b, a) \in V \times V$. The number of nodes is denoted $|V|$ and the number of edges $|E|$. The principle of the deformation algorithm (Algorithm 1) is the following one: when the user translates a node n from the position Pn(x,y) to Pn'(x',y') then the node n propagates the deformation at all its children.

The recursive Algorithm 2 propagates the deformation as illustrated in Figure 7. The deformation is based on the formula of Line 1 which loosely corresponds to a physical model.

Algorithm 1 Deformation(n, Pn, Pn')

```
1: set position of n to Pn'
2: for each neighbor ni of n do
3:     propagation(n, ni, Pn, Pn')
4: end for
```

Algorithm 2 propagation(father, child, PfOld, pfNew)

```
1: $PcNew' = PfNew + \frac{PcOld-PfNew}{\|PcOld-PfNew\|} * \|PcOld - PfOld\|$
2: for all neighbours ni of child ($ni \neq father$) do
3:     propagation(child, ni, PcOld, pcNew)
4: end for
```

This algorithm can create overlapping when the position of a node is updated. To solve this problem, we add at Line 1 of Algorithm 2 the following sequence to avoid collision as shown in Figure 8.

```
1: for all nodes nj not neighbour of child do
2:     if distance(child, nj) $<$ epsilon then
3:         updatePositionFromCollision(father, child, nj);
4:     end if
5: end for
```

In order to control the deformation (e.g. deform one branch only, as shown in Figure 9), it is useful to let the user stop the propagation where needed. It is therefore possible to define anchors associated to nodes (Algorithm 3) where the propagation of the deformation is stopped. Anchors are defined by manipulating tokens. Since several anchors are possible, several users can deform the graph collaboratively by manipulating tokens.

Moreover we modify Algorithm 2 by adding the following sequence before Line 2:

```
1: if child is a stop then
2:     return;
3: end if
```

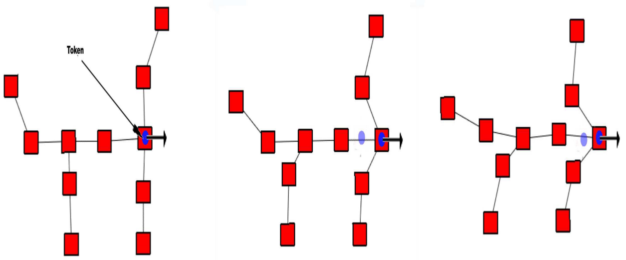

Figure 7: On the left, the user selects a node with a token. S/he then drags the token to the right. The tree is deformed accordingly, following the movement of the token while the size of the edges is maintained constant.

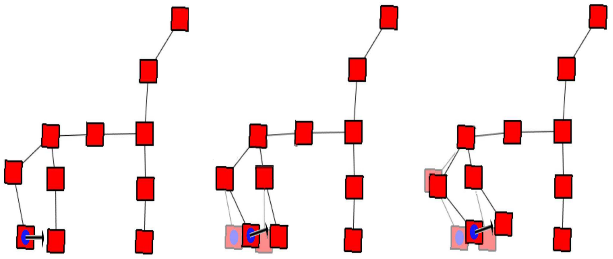

Figure 8: When the grabbed node (node 1) is close to another one (node 2), the node 2 is pushed away (repulsion).

Algorithm 3 deformationWithAnchors(set<node> benchMark, n, Pn, Pn')

```
1: Deformation(n,Pn, Pn');
2: for all nodes b in benchMarks do
3:     Deformation(b, Pb, pbOld);
4: end for
```

```
4: if child is a benchmark then
5:     for all his neighbors n (less his father) do
6:         mark n as stop.
7:     end for
8: end if
```

The final algorithm takes $O(|V|^2)$ time to complete. Experimental tests show that interaction was too slow when manipulating huge graphs like RNA structures. To speed up our algorithm, we applied the Fruchterman solution [15] which is based on a grid of squares. The function for identifying collisions then takes constant time instead of $O(|V|)$. Our resulting optimized algorithm takes $O(|V|)$ time to complete, guaranteeing swift responses even with huge RNA graphs.

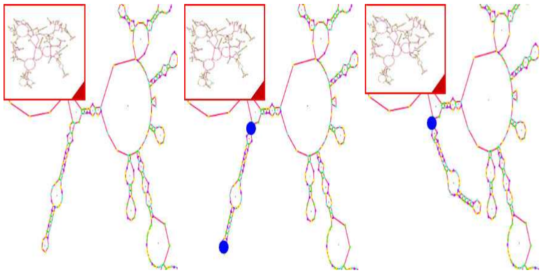

Figure 9: On the left, the initial 2D structure (1JJ2B.pdb). In the middle, the lower token defines the selected node while the upper token defines an anchor node. On the right, the obtained deformed secondary structure after bending the branch.

5 Related work and discussion

Due to their usefulness in understanding RNA, many tools have been developed to visualize the secondary structure of RNA [19, 20, 21, 28, 34, 33]. Few of them provide both an efficient manipulation on huge graphs and a link with the tertiary structure. In this section, we review the capabilities of three interactive recent tools in comparison with NAVRNA.

jViz.RNA [21] makes use of a non-deterministic spring model when drawing the classical structure plot. In the spring model, each nucleotide becomes a node and small springs are used to connect each node sequentially, forming a backbone. Once the backbone is created, the hydrogen bonds are added as edges which pull the paired nodes (nucleotides) together, forming the final structure. Each node has a small repulsive force which helps to keep them distanced from each other. The edges act as springs, pulling the connected nodes together with a user-selected amount of force. The result is an elastic classical structure that the user can stretch and adjust to obtain a desired layout. All the elements of the structure can be moved without any obstruction or disconnection of structural elements. Nevertheless with a huge graph, the interactive technique can be difficult to use (responsiveness and screen stability issues). As opposed to jViz.RNA, NAVRNA is based on a deterministic algorithm facilitating the comparison of two RNA structures. Indeed as the algorithm is deterministic, it insures that two equivalent structures (or substructures) are drawn in a constant and planar way. Moreover jViz.RNA does not allow the visualization of the tertiary structure.

RNAMLView [33] provides fine controls over the secondary structure representation defined by RNAView from PDB files [37]. As opposed to NAVRNA, the obtained drawing of the secondary structure includes many overlapping helices. It then requires significant actions by the user (translation, rotation and flip) to make the representation readable.

S2S [20] is a unified framework in which a user can display, manipulate and interconnect heterogeneous RNA data such as multiple sequence alignments, secondary and tertiary structures. It is composed of PyMol [35] to visualize tertiary structures and RNAMLview [33] to generate, visualize and manipulate secondary structures. S2S offers the advantage of linking these two structures. For example, it is possible to select one helix on the secondary structure in order to highlight it within the tertiary structure. In comparison with NAVRNA, S2S offers more advanced biological functions such as alignment of RNA secondary structures as well as complex algorithms for deriving the tertiary structure from a secondary structure (RNA structure prediction). In NAVRNA we focused on the interactive techniques for exploring and editing the RNA structures. For example in S2S only the last selected helix of the secondary structure is highlighted within the tertiary structure. In NAVRNA we allow the user to select an area that can contain several helices and loops. Moreover the user can also select an area of interest within the tertiary structure, an action that is not possible using S2S. Finally for exploring the structure using S2S, the user must always shift between modes (pan, zoom and rotate) which takes time as opposed to NAVRNA that allows continuous mode-free exploration of the structure with two-handed interaction.

The above described systems are based on classical Desktop interaction also called WIMP interfaces (Windows, Icons, Menus, Pointing device, as mouse). For making the interaction more efficient with such huge spaces of data, few existing interactive systems in bioinformatics are based on post-WIMP [12] interfaces [1, 13, 10, 31, 30, 27]: they are mainly based on the immersion of the user within the 3D structure. For example in [1, 13, 10], the visualization techniques are implemented in CAVEs (automated virtual environment) to dynamically simulate the physical and chemical properties of the molecular system. As opposed to these systems based on an immersive environment, interaction with NAVRNA is based on an augmented collaborative environment made of interactive surfaces (a table and a whiteboard).

6 Conclusion

We presented an interactive system, NAVRNA, for the visualization, the exploration and the manipulation of the secondary and tertiary structures of a RNA molecule. Our main contributions are (1) to enable the visualization of the secondary and tertiary structures at the same time while maintaining consistency between the two views, (2) to use the 2D view as a tool to navigate within the 3D view and (3) to provide an efficient algorithm to bend RNA branches with a high level of interactivity. Although NAVRNA is dedicated to the manipulation of RNA molecules, the interactive system can be used for the manipulation of any sub-cubic outer-planar bi-connected graph.

7 Acknowledgements

The work presented in the article is partly funded by the multidisciplinary French project NAVGRAPHE (as part of a national research program on large data sets).

References

[1] Ai, Z., and Frolich T., Molecular Dynamics simulation in Virtual Environments, Computer Graphics Forum, vol. 17, pp. 267-273, 1998.

[2] Auber, D. Tulip: A huge graph visualisation framework. In P. Mutzel and M. Jnger, editors, Graph

Drawing Software, Mathematics and Visualization, pp. 105-26, 2003.

[3] Auber D., Delest, M., Domenge, J. P., and Dulucq S. Efficient drawing and comparaison of RNA secondary structure. Tech report LaBRI, pp. 1325-04, 2004.

[4] Bailly, G., Nigay, L., Auber, D., NAVRNA: Visualization - Exploration - Manipulation of RNA, In proceedings of the Conference on Advanced Visual Interfaces (AVI'06), 2006.

[5] Berard, F. The Magic Table: Computer-Vision Based Augmentation of a Whiteboard for Creative Meetings. In IEEE workshop on Projector-Camera Systems (PROCAM), 2003.

[6] Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N. and Bourne, P.E. The Protein Data Bank. Nucleic Acids Res., 28, pp. 235-242. 2000.

[7] Bertault, F., A force-directed algorithm that preserves edge-crossing properties. Information Processing Letters, Vol. 74(1-2),pp. 7-13, 2000.

[8] Bier, E., Stone, M., Fishkin, K., Buxton, W., Baudel, T. A Taxonomy of See-Through Tools. Readings in Human-computer interaction, Morgan-Kaufman, pp. 358-364, 1994.

[9] Bruccoleri, R., E., An improved algorithm for nucleic acid secondary structure display, Bioinformatics, vol. 4, pp. 167-173, 1988.

[10] Chastine, J., Zhu, Y., Brooks, J., Owen, G.S., Harrison, R. A Collaborative Multi-View Virtual Environment for Molecular Visualization and Modeling. In IEE Conference on Coordinated & Multiple Views in Exploratory Visalization (CMV), 2005.

[11] Chi, E., A Taxonomy of Visualization Techniques using the Data State Reference Model. In Proceedings of Information Visualization (InfoVis '00), pp. 69–75, 2000.

[12] van Dam, A., Post-WIMP User Interfaces. In Communications of the ACM, 40(2), pp. 63-67, 1997.

[13] Disz T., Papka, M., Stevens, R., Pellegrino, M., and Taylor, V., Virtual Reality Visualization of Parallel Molecular Dynamics Simulation, in Proceedings of High-Performance Computing, 1995.

[14] Eades, P., A Heuristic for Graph Drawing, Congressus Numerantium 42, pp. 149-160, 1984.

[15] Fruchterman, T. M. J., Reingold, E. M., Graph drawing by force directed placement. Software-Practice and Experience, 21(11), pp. 1129-1164, 1991.

[16] Gainant, G., Auber, D. ARNA, Interactive comparison and Alignement of RNA Structure. In IEEE Symposium on Information Visualization (INFOVIS 04), pp. 1-2, 2004.

[17] Han, K.Byun Y., Pseudoviewer 2: Visualization of RNA psuedo-knots of any type, Nucl. Acids Res., vol. 31, no. 13, 2003.

[18] Hinckley, K., Czerwinski, M., Sinclair, M. Interaction and modeling techniques for desktop two-handed input, In ACM Symposium on User interface software and technology (UIST'98), pp. 49-58, 1998.

[19] Hofacker, I. L., Fontana, W., Stadler, P. F., Bonhoeffer, L. S., Tacker, M., Schuster, P., Fas folding and comparison of RNA secondary structures, Monatsh. Chem., vol. 125, pp. 167-188, 1994.

[20] Jossinet, F., Westhof, E. Sequence to Structure (S2S): display, manipulate and interconnect RNA data from sequence to structure. Bioinformatics. 15. pp. 3320-1, 2005.

[21] Kay, C., Wiese, E. G., Vasudevan, A., jViz.Rna - A Java Tool for RNA Secondary Structure Visualization, IEEE Transactions on NanoBioscience, Vol. 4(3), pp. 212-218, 2005.

[22] Kruger, R., Carpendale, S., Scott, S. D., Tang A. Fluid integration of rotation and translation. In ACM Conference on Human Factors in Computing Systems (CHI'05), pp. 601-10, 2005.

[23] Kurtenbach, G., Fitzmaurice, G., Baudel, T., Buxton, B. The Design of a GUI Paradigm based on Tablets, Two-hands, and Transparency, In ACM Conference on Human Factors in Computing Systems (CHI'97), pp. 35-42, 1997.

[24] Letessier, J., Berard, F. Visual Tracking of Bare Fingers for Interactive Surfaces. In ACM Symposium on User Interface Software and Technology (UIST), pp. 119-122, 2004.

[25] Muller, G., Gaspin, C., Etienne, A., Westhof, E., Automatic Display of RNA Secondary Structures, CABIOS, Vol. 9, Num. 5, pp. 551-561, 1993.

[26] Nigay, L., Vernier F. Design method of interaction techniques for large information spaces. In ACM Conference on Advanced Visual Interfaces (AVI), pp. 37-46, 1998.

[27] Ouh-young, M., Pique, M., Hugues, J., Srinivasan, N., Brooks, F. P., Using a manipulator for force display in molecular docking, in Proceedings IEEE International conference on Robotics and Automation, 1988.

[28] Rijk, P. D., Wuyts J., Wachter D., RnaViz 2:An improved representation of RNA secondary structure, Bioinformatics, vol. 19, no. 2, pp. 299-300, 2003.

[29] Stromer, J. N., Quon, G. T., Gordon, P. M. K., Turinsky, A. L., Sensen, C. W. Jabiru : harnessing Java 3D behaviors for device and display portability, IEEE Computer Graphics and Applications, 25(2), pp. 70-80, 2005.

[30] Schkolne, S., Ishii, H., and Schroder, P., Immersive Design of DNA Molecules with a Tangile Interface, in Proceedings of the IEEE Visualization 2004 (VIS'04), vol. 0, 2004.

[31] Su, S., and Loftin R. B., A Shared virtual Environment for Expoloring and Designing Molecules, Communications of the ACM, Vol. 44, pp. 57-58, 2001.

[32] Venolia, D., Facile 3D direct manipulation. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '93). pp. 31-36, 1993.

[33] Yang, H., Jossinet, F., Leontis, N., Chen, L., Westbrook, J., Berman, H. and Westhof, E. Tools for the automatic identification and classification of RNA base pairs. Nucleic Aids Res., 31, pp. 3450-3460. 2003.

[34] Zucker, M., Mfold Web server for nucleic acid folding and hybridization prediction, Nucl. Acids Res., vol. 31, no. 13, pp. 3406-3415, 2003.

[35] PyMol; http://pymol.sourceforge.net

[36] Tulip; http://www.tulip-software.org

[37] PDB; http://www.rcsb.org/pdb/