BRIEFINGS IN BIOINFORMATICS. VOL 12. NO 6. 689–701
Advance Access published on 9 February 2011
doi:10.1093/bib/bbq089

GPU-powered tools boost molecular visualization

Matthieu Chavent, Bruno Lévy, Michael Krone, Katrin Bidmon, Jean-Philippe Nominé, Thomas Ertl and Marc Baaden

Submitted: 18th October 2010; Received (in revised form): 22nd December 2010

Abstract

Recent advances in experimental structure determination provide a wealth of structural data on huge macromolecular assemblies such as the ribosome or viral capsids, available in public databases. Further structural models arise from reconstructions using symmetry orders or fitting crystal structures into low-resolution maps obtained by electron-microscopy or small angle X-ray scattering experiments. Visual inspection of these huge structures remains an important way of unravelling some of their secrets. However, such visualization cannot conveniently be carried out using conventional rendering approaches, either due to performance limitations or due to lack of realism. Recent developments, in particular drawing benefit from the capabilities of Graphics Processing Units (GPUs), herald the next generation of molecular visualization solutions addressing these issues. In this article, we present advances in computer science and visualization that help biologists visualize, understand and manipulate large and complex molecular systems, introducing concepts that remain little-known in the bioinformatics field. Furthermore, we compile currently available software and methods enhancing the shape perception of such macromolecular assemblies, for example based on surface simplification or lighting ameliorations.

Keywords: *GPU visualization; molecular representations; huge systems visualization; molecular shape enhancement; binding site highlighting*

INTRODUCTION

Visualization of macromolecular structures is described in a recent review focusing on traditional approaches to visualize 3D structural data, providing an excellent overview of this field and currently available tools [1]. Here, we take a look at the latest contributions from the computer science field, with the potential to change the future of

Corresponding author. Marc Baaden, Institut de Biologie Physico-Chimique, Laboratoire de Biochimie Théorique, CNRS UPR 9080, 13, rue Pierre et Marie Curie, F-75005 Paris, France. Tel: +33 1 58 41 51 76; Fax: +33 1 58 41 50 26; E-mail: baaden@smplinux.de

Matthieu Chavent is a postdoctoral fellow in molecular visualization at the CEA (Commissariat à l'énergie atomique et aux énergies alternatives, which is a French government-funded technological research organization). He works on GPU computing for modelling and visualization of huge molecular systems.

Bruno Lévy is a senior researcher in INRIA-Nancy Grand-Est (INRIA: French national institute for research in computational science and control), and a member of the LORIA laboratory. He is the scientific head of the ALICE project-team. His main research topic is numerical geometry, creating new algorithms for acquiring, transforming and optimizing the representation of 3D shapes.

Michael Krone is currently working on his PhD thesis in Computer Science at the Visualization Research Centre at the University of Stuttgart, Germany. His main research interest lies in the visualization of biomolecular systems.

Katrin Bidmon finished her doctorate in Computer Science at the Visualization Research Centre at the University of Stuttgart, Germany. Her main research interests lie in geometric aspects of visualization applications.

Jean-Philippe Nominé was previously in charge of HPC application software groups, including data management, post-processing and visualization tools, at the CEA supercomputing complex at Bruyères-le-Châtel, until 2008. He is now coordinating the European PRACE project for CEA. He graduated from Ecole Polytechnique, then received a PhD from Pierre-et-Marie-Curie (Paris 6) University in Robotics in 1991.

Thomas Ertl is a full professor of computer science at the University of Stuttgart, Germany and the head of the Visualization and Interactive Systems Institute (VIS) and the Visualization Research Centre (VISUS). He received a MS in computer science from the University of Colorado at Boulder and a PhD in theoretical astrophysics from the University of Tübingen. His research interests include visualization, computer graphics, and human computer interaction.

Marc Baaden is a computational chemist at CNRS in Paris, focusing on interactive approaches to molecular modelling of complex biological systems. Molecular visualization is an essential part of this work combining high-performance computing and virtual reality, with particularly exciting perspectives offered by the latest graphics processors.

molecular visualization. Yet, many of these new approaches remain largely unknown, which could be explained by two main reasons:

(1) Common interest focuses on available and readily usable tools. Those are only rarely associated with the work of computer scientists, mostly geared towards new state-of-the-art techniques rather than providing end-user software. Most of the results presented in this review relate to ongoing developments (i.e. the corresponding tools are not generally available), but some notable exceptions exist and are already offered to the scientific community. In Table 1, we provided a compilation of these tools; and
(2) A gap subsists between the computer science and bioinformatics fields. Publications in the former often contain very detailed technical explanations, rendering them difficult to read for non-specialists in the field. We tried to reduce the number of such expressions in the present manuscript and the remaining technical terms are described in a glossary at the end of the main text. Our approach is to stress the enormous potential of new visualization methods for structural biology and bioinformatics rather than describe the intrinsics of the techniques applied to obtain such visual effects.

More generally, very few recent reviews address new computer science developments for molecular graphics. The article by Goddard and Ferrin is one of such rare reports [2]. Furthermore, the computer visualization field evolves very quickly due to continuously renewed graphics hardware capabilities. Recently, the performance of graphics cards has drastically increased by a factor of approximately 2.6 over the past 4 years. In order to illustrate this evolution, we will discuss some features of the graphics cards from Nvidia, one of the main fabricants. A big leap forward occurred between the past two generations (GTX 2xx and GTX 4xx series) compared to the previous ones (Figure 1). The main processing power of these graphics processing units (*GPUs*)[1] benefits from the constantly rising *clock speeds* and is largely driven by increasing parallelism. While modern *CPUs* only have up to six *cores*, GPUs can have up to 480 smaller cores (see ref. [3] for more details on GPU architecture). In addition to graphics output and geometry generation, modern GPUs can be used for general purpose calculations [3–5]. This tremendous potential encouraged computer scientists to design new algorithms for massively parallel execution on the GPU. However, using such hardware implies to comply with several constraints. GPU cores are dedicated to a limited set of specific, massively parallel operations. Another bottleneck is the communication between CPU and GPU, imposing limits on the transfer of large amounts of data. Fortunately, the memory bandwidth has increased consequently these past years (Figure 1). To help developers create GPU-optimized algorithms, specific formalisms such as the GLSL [6] (graphic card independent), Cg [7] (Nvidia card dependent), CUDA [8] (Nvidia card dependent) or OpenCL [9] (graphic card independent) languages were created. GLSL and Cg are dedicated for rendering, whereas CUDA and OpenCL are intended for calculations.

Structural biology is another field where tremendous progress was achieved over the past decade. Experimentalists have developed new techniques to routinely crystallize proteins [10–12], increasing the number of available structures in databases. Using experimental techniques (such as electron-microscopy or small angle X-ray scattering) or computational methods (such as protein docking), it is now possible to study huge macromolecular structures such as chaperone proteins [13], the ribosome [14] or viral capsids [15] in atomic detail. Displaying such complex macromolecules requires efficient tools, an area where new scientific visualization techniques could provide a promising answer. These techniques furthermore provide an improved visual perception, which could drastically impact the way to visualize—and consequently to think about—molecular structures. Backed by these observations, we endeavour to highlight the latest frontier research in the field and complement previous reports by presenting new developments originating in computer science and visualization.

[1] The first time a technical term appears, it is emphasized in italics. These terms are then briefly explained at the end of the document, in a glossary part.

Table 1: Visualization software and methods with related URL links

Method/Software	Availability	Cost	OS	Molecular Metaphors	Visual effects	URL
Amira	Available	$	Win, Mac, Linux	Ball and Stick; VdW; Licorice, secondary structures; molecular surface	blur effects	http://amira.zib.de/mol
BALLView [41, 42]	Available	Free	Win, Mac, Linux	Ball and Stick; VdW; Licorice; secondary structures; molecular surface...	silhouette contouring, cel-shading for 2ndary structure, reflections and shadows	http://www.ball-project.org/Ballview/
Chimera [40]	Available	Free	Win, Mac, Linux	Ball and Stick; VdW; Licorice; secondary structures; molecular surface...	silhouette contouring	http://www.cgl.ucsf.edu/chimera/
PMV [39]	Available	Free	Win, Mac, Linux	Ball and Stick; VdW; Licorice; secondary structures; molecular surface...	Cel-shading, silhouette contouring	http://mgltools.scripps.edu/packages/pmv
QuteMol [22]	Available	Free	Win, Mac, Linux	Ball and Stick, VdW, Licorice using GPU	ambient occlusion, silhouette contouring, halo	http://qutemol.sourceforge.net
TexMol [21]	Available	Free	Win, Mac, Linux	VdW, secondary structures using GPU	–	http://www.cs.utexas.edu/~bajaj/cvc/software/texmol.shtml
VMD [20]	Available	Free	Win, Mac, Linux	VdW using GPU; Ball and Stick; Licorice; secondary structures...	–	http://www.ks.uiuc.edu/Research/vmd/
Yasara	Available	Free	Win, Mac, Linux	Ball and Stick, VdW, Licorice using GPU; secondary structures; molecular surface...	ambient occlusion, global shadowing	http://www.yasara.org
Method1: Quadratic Surface primitives [18]	Not available	–	–	Ball and Stick, VDW, Licorice using GPU	global shadowing, silhouette contouring	**Video:** http://cg.www.techfak.uni-bielefeld.de/publications/pbg06.avi
Method2: Two-Level approach [23]	Not available	–	–	Ball and Stick, VdW on GPU	silhouette contouring, depth cueing	**Video:** http://dlampe.com/ove/protein.dynamics/protein.dynamics.ieee.htm
Method3: MegaMol: Molecular dynamics visualizations [28]	On request *(work in progress)*	–	Win, Linux	Molecular Surface on GPU, Secondary structures on CPU-GPU, Ball and Stick using GPU...	transparency, depth cueing, silhouette contouring	**Video:** http://www.vis.uni-stuttgart.de/~kroneml/
Method4: MetaMol: Molecular Skin Surface [30]	Not available	–	–	Molecular Skin Surface on GPU	crystal effect, HDR rendering	**Video:** http://www.loria.fr/~chavent/metamol.htm
Method5: Metaballs [33]	Not available	–	–	Metaballs visualization using GPU	silhouette contouring	**Video:** http://www.cs.tsukuba.ac.jp/~kanamori/projects/metaball
Method6: Molecular Surface abstraction [35, 36]	Webserver	Free	–	Simplification of Molecular Surface to keep only representative features	silhouette contouring, ambient occlusion, texture mapping	**Video:** http://pages.cs.wisc.edu/~gregc/Research/molsurfabstract.html **Webserver Link:** http://mitchell-lab.org/bacter/GRAPE/
Method7: Text scaffold [49]	Not available	–	–	Well suited annotations on molecular surface	–	**Video:** http://pages.cs.wisc.edu/~gregc/Research/textscaffolds.html

The $ symbol indicates that it is necessary to pay in order to use the program.

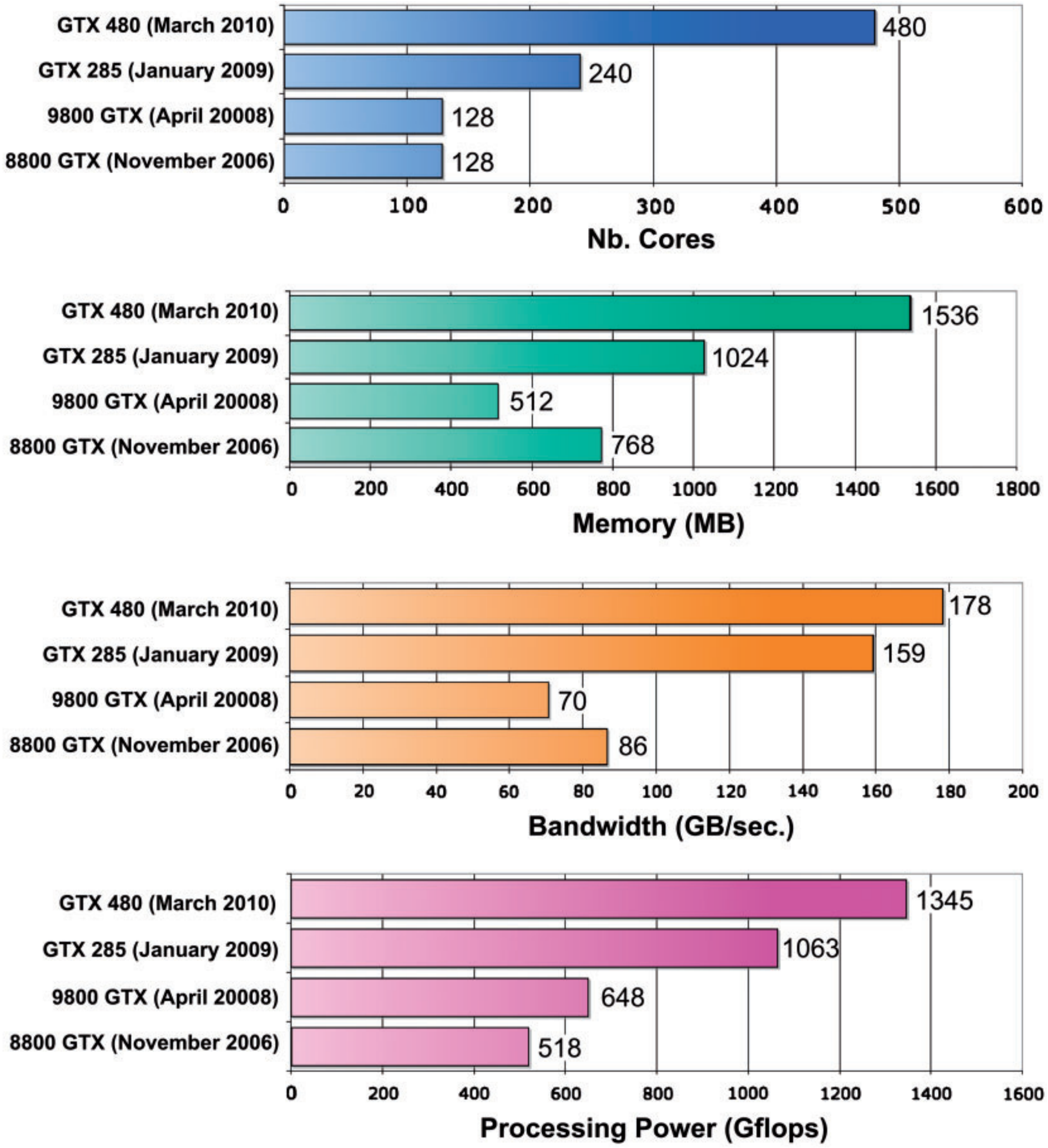

Figure 1: Four generations of Nvidia graphics cards. Comparison of critical parameters for four graphics card generations (based on information available on the Nvidia website: http://www.nvidia.com). The number of cores provides information on potential parallelization. Memory and bandwidth are important as they govern the amount of data that can efficiently be passed between CPU and GPU. Processing power provides an indication of hardware performance measured in Giga flops (Gflops). The term flops is the abbreviation of FLoating point OPerations per Second which is related to the number of instructions per second.

FROM SMALL MOLECULES TO MACROMOLECULAR ASSEMBLIES: WELL-KNOWN MOLECULAR METAPHORS ALIVE AND KICKING

Recently, D. S. Goodsell described several techniques to fill the gap between visualizing atomic and cellular scales [16]. A major goal of computer science approaches to molecular visualization is to overcome current performance limits by improving existing rendering techniques in order to efficiently display such large structures. Conversely, results obtained for huge assemblies are beneficial for small molecule visualization. In this part, we will focus on studies that redesign traditional algorithms to exploit new graphics card capabilities, hence radically improving display performance. We will also describe methods creating new renderings of well-known molecular metaphors.

Conventional molecular representations

Simplified molecular representations such as ball and stick, Van der Waals (also referred to as space-filling or CPK) or licorice are among the oldest but also the most used structural representations [17]. They were first obtained using simple lines to link atoms (simplified licorice model) or subsequently based on more sophisticated models of cylinders and spheres (ball and stick, space-filling and licorice). Until recently, the latter representations were generally implemented by *triangulating* spherical and cylindrical surfaces. The main limitation of this approach is that rendering huge assemblies creates a significant amount of triangles, especially when high quality smooth rendering is desired. Such a huge number of triangles is computationally very demanding, even using the latest graphics hardware. With the development of programmable graphics cards, a new technique was introduced to tackle this problem. This method, referred to as *ray casting*—or sometimes *ray tracing*—on GPUs, first creates a roughly triangulated envelope, then uses the programmable *graphics pipeline* to analytically define a *pixel accurate* surface (for more details see reference articles [18, 19]). With this approach, it is possible to represent, in real time, a massive number of spheres and cylinders with a pixel accurate precision for any zoom level. One of the first implementations of GPU ray casting for molecular representations goes back to Toledo and Lévy's work on space-filling models in 2004 [19]—this method was subsequently implemented in the VMD molecular viewer [20] (Table 1). At the same time, Bajaj *et al.* [21] presented their TexMol program (Table 1) able to display space-filling models as well as other representations such as secondary structures with the equivalent GPU technique. A few years later, in 2006, this method was extended by Sigg *et al.* [18] (Method1 in Table 1) and Tarini *et al.* [22] to represent huge assemblies with particular lighting effects thus enhancing shape perceptions (Figure 2A and C). We note that Tarini's QuteMol program is freely available (Table 1). In 2007, Lampe *et al.*, presented a two-level approach to visualize dynamic changes in macromolecular assemblies [23] (see Method2 in Table 1) using the GPU ray casting technique for space-filling or ball and stick models (see Figure 2B for a GPU ball and stick example). These examples represent significant improvements for accurately and efficiently visualizing standard molecular models as compared to CPU post-processing of molecular scenes using ray tracing to achieve similar rendering quality (as with the PoV-Ray software: http://www.povray.org). The GPU representations are obtained in real time and can be handled interactively, which opens up new possibilities. For example, pixel-accurate rendering and fast *display rates* can be achieved. Hence, all new representations are well suited for immersive environments-some of them were indeed created for this purpose [23]. Several other GPU implementations exist, but are not described in the literature: we can cite the Yasara visualization tool (Table 1) that uses GLSL code for hardware *tessellation*, where the GPU increases the geometric details of the polygon *mesh* or Krone *et al.*'s implementation (http://www.vis.uni-stuttgart.de/∼kroneml) which allows interactive GPU ray casting of large dynamic data sets.

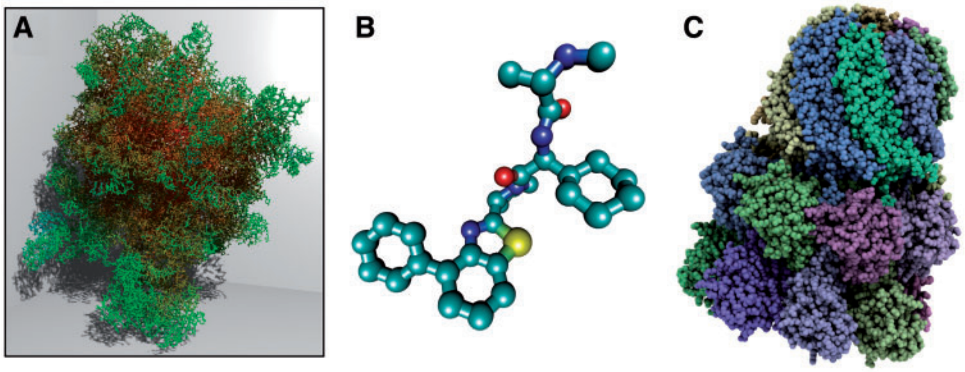

Figure 2: Known molecular metaphors taking advantage of graphic hardware capabilities. (**A**) Global shadowing of the ribosome structure represented in Ball and Stick [18]; (**B**) Ray-casting ball and stick visualization of a small molecule; (**C**) Van der Waals representation of the GroEL protein using the ambient occlusion technique [22]. Picture A was reproduced with the permission of the respective copyright holders.

Secondary structure visualization

The secondary structure (or cartoon) representation is a particularly useful metaphor to depict complex molecular structures and highlight specific points of interest [1, 16, 17]. This abstract representation simplifies intricate macromolecular structures by removing some details and providing a hierarchical visualization. The metaphor focuses on molecular backbone organization known as secondary structure, distinguishing alpha-helices, beta-sheets and disorganized parts (coils) by describing them as ribbon spirals (or cylinders), flat arrows and tubes, respectively. As explained above, visualizing such metaphors with a classical triangulation technique requires a compromise between image quality and rendering efficiency. In 2004, Bajaj *et al.* created simple visual primitives to represent alpha-helices via GPU ray casting [21] (Figure 3A). This visualization was implemented in the TexMol program (Table 1). More recently, Krone *et al.*, proposed a hybrid GPU–CPU technique to visualize secondary structures of large proteins with interactive *frame rates* [24] (Figure 3B), using the programmable *Geometry Shader* of modern GPUs to create the cartoon representation. A similar approach was recently implemented by Wahle and Birmanns [25] (Figure 3C). Although these methods generate triangulations similar to the classical CPU implementations, they achieve higher frame rates by exploiting the parallelism of the GPU, while minimizing the amount of data to be transferred between CPU and GPU. These improvements enable interactive manipulation of protein secondary structures and real time observation of backbone transformations even for very large systems, an important issue for *in silico* protein folding simulations [26].

Molecular surface visualization

The most common definition used to represent molecular surfaces is the one by Connolly [27] also known as Molecular Surface—or MS. Much like the conventional representations discussed above, surfaces are traditionally depicted by large amounts of triangles. Recently, the GPU ray casting technique was successfully applied to implement alternative surface representations [28, 29], hence significantly reducing the number of triangles so that dynamic surface evolutions could be followed in real time (see Method3 in Table 1). The surface evolution can be displayed fluently for systems up to mid-size proteins (a few thousand atoms). Furthermore, for the first time, a pixel-accurate surface visualization could be achieved at any zoom level (Figure 4A). The GPU ray casting technique was further used by Chavent *et al.*, to represent the Molecular Skin Surface—or MSS [30] (Figure 4B). This surface, defined by Edelsbrunner at the end of the nineties [31], is slightly different from the molecular surface defined by Connolly and offers several advantages. First, its mathematical definition confers properties such as the absence of self intersections and improved smoothness, lacking in the MS. Second, even if MS and MSS are both piecewise defined surfaces, the MSS is exclusively composed by second degree equations (defining sphere and hyperboloid surfaces) while the MS requires integration of second degree equations (defining sphere surfaces) and fourth degree equations (defining torus surfaces), slowing down surface calculations. On the downside,

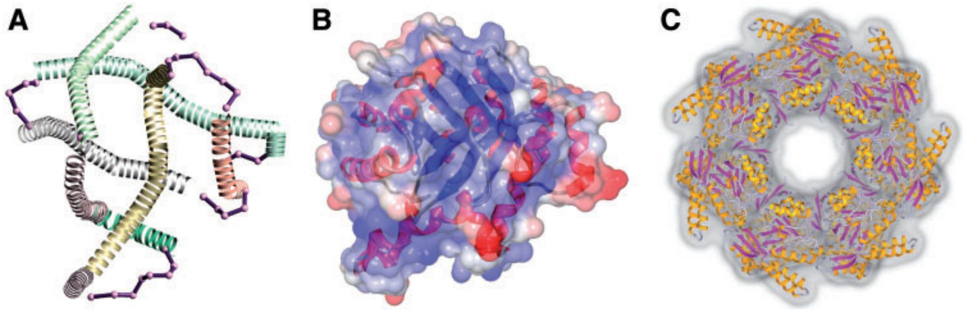

Figure 3: Secondary Structure representations. (**A**) TexMol secondary structure representation using GPU-based rendering [21]; (**B**) GPU secondary structure visualization [24] with a transparent molecular surface; (**C**) Chaperone protein using GPU-accelerated secondary structure representation implemented by Wahle and Birmanns. Picture B was reproduced with the permission of the respective copyright holders.

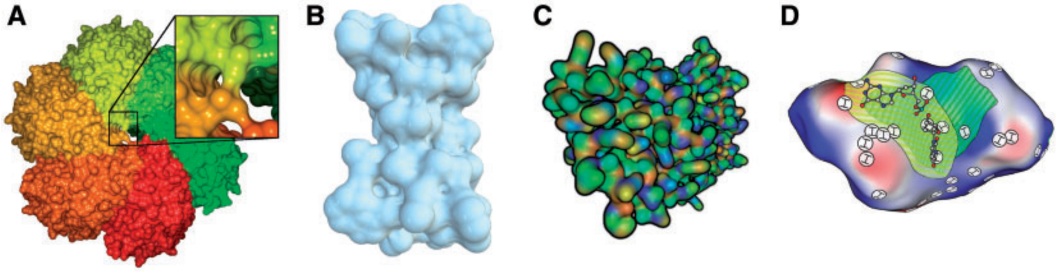

Figure 4: Molecular surface representations. (**A**) molecular surface defined by Connolly displayed using GPU-ray casting [28]; (**B**) GPU-ray casting of the molecular skin surface [30]; (**C**) GPU-ray casting of a metaballs surface [33]; (**D**) abstracted surface [35]. (A, C and D) were reproduced with the permission of the respective copyright holders.

to delimit each part of the MSS surface, it is necessary to compute a particular structure called the *Mixed complex* for each configuration. This calculation has to be performed on the CPU and is computationally demanding. Thus, surface changes in real time are limited to molecules ranging from a few hundred atoms to about a thousand atoms, i.e. small molecules and peptides (see Method4 in Table 1). Recently, the MS and MSS ray casting methods were extended by an implementation that also takes advantage of multi-core CPUs in addition to GPU capabilities [32]. This implementation is soon to be integrated in the Amira Software (Table 1). Alternatively, molecular surfaces can be approximated by *metaballs*, representing an isosurface of a density field defined by a set of points. Kanamori *et al.* have used such metaballs in conjunction with GPU ray casting to represent mid-size proteins [33] (Figure 4C) and observe their surface evolution in real time (see Method5 in Table 1). An alternative, fast approximation of the molecular surface is to extract isosurfaces from a density map, i.e. a volume. Isosurfaces of high quality can be extracted using the well-known *marching cubes algorithm* or rendered in real time via GPU volume ray casting [34], which hints at the suitability of likewise techniques for dynamic datasets. In addition to GPU implementations, some new methods are still based on standard molecular triangulation techniques to represent novel depictions of surfaces. For example, in 2007, Cipriano *et al.*, used a technique to *remesh* a molecular surface triangulation and remove small details to retain only significant features of the shape [35]. A web server, called GRAPE [36] (Graphical Abstracted Protein Explorer), has recently been developed using this particular method (Figure 4D and Method6 in Table 1). This molecular surface abstraction is particularly well suited to display ligand binding sites and gather a quick idea of the molecular shape.

VISUAL EFFECTS TO ENHANCE MOLECULAR STRUCTURE REPRESENTATIONS AND FACILITATE THEIR PERCEPTION

Lighting effects have become an essential tool for displaying molecules. The complexity of macromolecular structures calls for particular effects to correctly appreciate and understand intricate molecular shapes. One of the most basic visual effects is the addition of *specular lighting* to a scene [37], although this is not enough to depict complex shapes. More sophisticated effects such as *depth cueing* or *cel-shading* may be used [1]. Furthermore, it is sometimes desirable to combine several representations, such as ribbons and surface renderings. In this case, the use of transparency becomes important. Such visual effects are already implemented in new molecular visualization methods (see Figures 2 and 3 and visual effects field in Table 1). Additional visual effects exist, can be applied in real time and considerably improve molecular shape perception. Here, we intend to present a few compelling examples.

Lighting effects

Nowadays, we observe a convergence between virtual and real molecular representations. Molecular models can be examined as real 3D manufactured objects (for example in plastic), and can be manipulated in virtual environments using augmented reality approaches [38]. Conversely, new lighting effects are used to create photo-realistic virtual molecular objects. Both approaches help to comprehend the

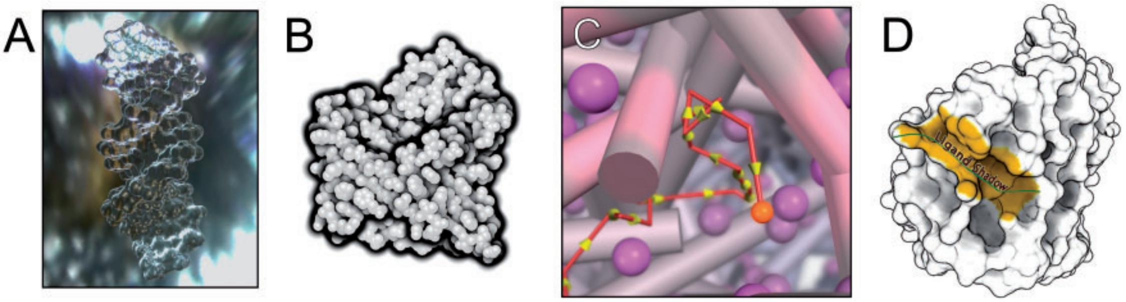

Figure 5: Lighting effects to enhance molecular structure perception. (**A**) high dynamic range rendering with a crystal effect [30]; (**B**) halo effect [22]; (**C**) Depth of field blur and colour desaturation [47]; (**D**) molecular annotation using text scaffolds [49]. (C and D) were reproduced with the permission of the respective copyright holders.

complicated three-dimensional molecular architectures. Using visualization approaches, it is possible to add global shadowing to a scene and position a molecule in a 3D environment to improve its shape perception [18] (Figure 2A). Ambient occlusion lighting is an increasingly popular technique in this respect. This method takes the global darkness due to global/ambient lighting into account and darkens re-entrant parts thus drastically improving depth perception. Tarini *et al.* applied the ambient occlusion technique to molecular representations [22] (Figure 2C) and implemented it in the QuteMol program. Other programs, such as the Yasara molecular viewer, have since implemented similar techniques (Table 1). Shape perception can be enhanced by emphasizing the contours of a structure wherever two overlapping parts of a molecule are separated by a distance larger than a given threshold (Figures 4C and 5D). This method, sometimes referred to as silhouette contouring, was first applied, in real time, to proteins by Tarini *et al.* [22]. It is used in several other methods to enhance shape [23, 28, 33, 35] and is extensively employed by D. S. Goodsell in combination with cel-shading to render the PDB molecules of the month (http://www.rcsb.org/pdb/motm.do). Similar representations are increasingly integrated in well known molecular viewer programs such as PMV [39] or Chimera [40]. In the same spirit, the BALLView [41] program provides a cel-shading effect with silhouette contouring to represent the secondary structure (Table 1). The silhouette contouring method was extended to create a halo effect (Figure 5B) that helps to identify depth discontinuities [22, 23]. Finally, it is possible to add more realistic lighting effects using High Dynamic Range (HDR) rendering (Figure 5A). With this effect, the addition of perceptual cues increases the apparent brightness of some parts of the given shape. HDR rendering affects how light is preserved in optical phenomena such as reflections and refractions. This feature is particularly important for transparent materials such as glass, and HDR rendering can be used to create artistic effects such as crystal-like molecules. Finally, interactive ray-tracing offers shadow and reflection effects, such as those implemented in the BALLView program using the RTfact library [42]. This technique is still less efficient than other techniques cited above, but it represents an important first step towards including interactive high level visual techniques within general purpose molecular frameworks such as BALLView.

Blur effects

Molecules are intrinsically flexible objects. This property is particularly important for protein function or for adaptation of therapeutic molecules to a binding site. However, techniques such as X-ray crystallography, used to determine molecular structures, depict static objects. In contrast, molecular dynamics simulations are widely used to investigate molecular functions and generate dynamic data. O'Donoghue *et al.* [1] describe programs to create and visualize molecular motions and propose to superimpose several dynamic snapshots. This method is limited in the number of snapshots that can be taken into account in order to avoid overloading the scene and still obtain a comprehensible representation. An interesting method to represent the uncertainty in atomic positions without overcrowding the scene is to use blur effects [43, 44]. These methods are particularly well suited to represent metastable conformations of molecules or superimpose docking ligand poses [45]. The method of

Lee and Varshney [43] is based on using multi-layered transparent surfaces to create the blur effect while the method of Schmidt-Ehrenberg *et al.*, is based on volume rendering [44]. Another application of volumetric representations was recently presented by Phillips *et al.*, where the molecule itself is rendered as a blurry density gradient, providing the context for the cavities, which are extracted by segmenting the volume [46]. Blur effects can also emphasize the depth of field. Falk *et al.* have applied depth blur in combination with *colour desaturation* [47], thereby drawing the attention of the user to a region of interest, while simultaneously clarifying the depth complexity of the scene (Figure 5C). These techniques were used on a simplified visualization of signal transduction in a cell (consisting of cylinders representing the cytoskeleton and spheres representing the signal proteins) and can also be applied to more detailed molecular models.

Adding information onto surfaces: texture mapping and annotation

New graphics card capabilities open the door for improving illustrative rendering. Recently, Weber added texture mapping onto ribbon representations thus creating some appealing pictures [48]. Beyond this illustrative and artistic goal, texture mapping can be used to annotate molecular representations. For example, Cipriano *et al.*, used little patches on molecular surfaces to enhance the visibility of ligand binding sites or to highlight a specific area on a surface [35] (see Figure 4D and Method6 in Table 1). Such a mapping procedure preserves other surface annotations that may for example be colour coded. Furthermore, the user can manually define a particular point of interest where a patch will be applied [35]. Recent approaches to annotate protein surfaces (Figure 5D) include text scaffolds [49], i.e. a smoothed invisible surface, used to position the text (see Method7 in Table 1). This technique is particularly well suited to annotate binding sites as the label is not just anchored to a single point, but follows the molecular topology. The obvious benefits are a better visibility and positioning of the text, thus avoiding some problems of traditional annotations, typically located in screen space, where labels may be hidden depending on the orientation of the scene.

TOWARDS A NEW MOLECULAR WORLD

In less than a decade, substantial progress was achieved in molecular visualization. Many new programs draw benefit from the latest capabilities of graphics cards, yet may not work correctly on older hardware and basic laptop computers. To be more specific, any GPU supporting at least the Shader Model 3.0 (indicated in the vendor specifications) meets all the technical requirements. As a guideline for the minimum hardware prerequisites, we have listed the equipment used in recently described methodologies (Table 2). Given the rapid evolution of the graphics hardware, many of these configurations start to be obsolete. As a guideline for choosing which graphics card to use at the time of writing, a basic configuration could consist in a medium grade graphics card such as the Nvidia GTX 460 (approximately $230), and a higher end configuration in an Nvidia GTX 480 or 580 (approximately $600). We note that it is not necessary to use a professional card such as the Nvidia Quadro

Table 2: Visualization methods and corresponding hardware configurations from the literature

Method/Software	Capabilities[a]	CPU tested[b]	GPU tested[b]
Method1: Quadratic Surface primitives [18]	Mainly GPU	Pentium 4 2.4 GHz	Nvidia Geforce 6800 GT
Method2: Two-Level approach [23]	Mainly GPU	–	Nvidia Geforce 8800 GTX
Method3: MegaMol: Molecular dynamics visualizations [28]	Mainly GPU	Intel Core2 Duo 3 GHz	Nvidia Geforce GTX 280
Method4: MetaMol: Molecular Skin Surface [30]	Mainly GPU	Intel core2 Duo 2.4 GHz	Nvidia Geforce 8800 GTX
Method5: Metaballs [33]	Mainly GPU	Intel Core2 quad 2.66 GHz	Nvidia Geforce 8800 Ultra
Method6: Molecular Surface abstraction [35]	CPU	Athlon 4400	Nvidia Geforce 7900 GT

In this table we present hardware configurations used in papers presenting new visualization methods (whenever this information is available). [a]The capabilities column describes whether a given method uses classical CPU calculations and the OpenGL pipeline (CPU), or shader code (such as Cg or GLSL) to benefit from GPU capabilities (GPU). [b]Both CPU and GPU hardware evolves very quickly and the models presented in this table correlate with the year when each paper was published. A general suggestion on hardware configurations for all methods/software is given in the conclusion.

series (clearly more expensive than the previous models) for the purposes described in this article, except for stereo rendering (not discussed in this review, but supported by some programs). Here, we focus on Nvidia cards, as many methods use code dedicated to hardware from this vendor (i.e. based on Cg or CUDA). Concerning the CPU, a basic configuration could be composed by an Intel quad-core i7-870 2.93 GHz (approximately $330) or, for a high performance machine, by an Intel hexa-core i7-7980x 3.33 GHz (approximately $1200). It is difficult to make a detailed prediction about the evolution of such hardware, as it largely depends on global marketing strategies of hardware constructors. Yet, there has been a big increase in graphics card capacities these last generations (Figure 1) and this trend will probably continue. This progress is driven by new GPU uses such as general calculations and will benefit graphics computation as well.

Thus, in the new virtual visualization world, scientists will be able to efficiently interact with molecular structures rather than merely display an elaborate pre-calculated picture—which may induce misconceptions [50]. Enabling such technologies raises issues of sharing, organizing and annotating visualizations of molecular structures and related data for efficient collaboration among scientists. Emerging collaborative virtual environments offer possible solutions [51], but their wide adoption remains a challenge for the near future. A related area concerns managing provenance of molecular visualizations with tools such as VisTrails (http://www.vistrails.org/). Thanks to close collaboration between molecular scientists and visualization experts, prototypes of such virtual worlds already exist in computer scientists' labs and may soon become available to the whole scientific community.

GLOSSARY

Cel-shading: the full term is celluloid shading, sometimes also called toon-shading. It is a lighting technique that is qualified as non-photorealistic. The goal of cel-shading is to obtain a 'cartoon'-like picture. For this purpose, the colour panel is limited and the shadows are not based on a gradient but are changed as a function of cut-off values, hence creating clear shadow frontiers. Furthermore, object contours are outlined to create an effect as if 'drawn by hand'.

Central Processing Unit (CPU): computer component that executes the instructions of an informatics program. CPUs are often designed for general purpose calculations.

Clock speed: speed at which a microprocessor executes instructions.

Colour desaturation: this effect diminishes colour intensity, often in order to highlight specific parts of a scene. If colours are totally desaturated, a grey-scale image is obtained.

Core: The core is the part of a CPU or GPU processor that actually reads and executes instructions. Processors can have single or multiple-core architectures. At time of writing, CPUs exist with 1 core (single core), 2 cores (dual cores), 4 cores (quad cores) and more recently 6 cores (hexa cores) or 8 cores (octo cores). For GPUs, the number of cores is much greater (up to 480 cores), but these cores are clearly dedicated to more specific operations compared to the general purpose CPU cores.

Depth cueing: colour effect to improve depth perception. The overall idea is to alter the actual object colour to increasingly match the background colour as a function of the distance from the camera. The colour of an object far away from the camera will be very similar to the background colour. This effect is sometimes referred to as 'fog effect'.

Display rate: measured in frames per second (fps), is the number of images that can be displayed per second. The higher this number, the better the user's perception of the fluidity of the scene. A good frame rate is around 30 fps, as the human brain is only able to perceive up to about 25 images per second (called the persistence of vision). If such high display rates are achieved, the rendering is qualified as real time. Below 30 and above 10 fps, the rendering is qualified as interactive, because the scene is still more or less fluid, with some apparent latencies. For values less than 10 fps, the scene suffers from very noticeable decelerations.

Frame rate: equivalent to display rate (see above).

Geometry shader: a shader is a set of software instructions used to calculate rendering effects and sent to the graphics pipeline (see definition hereafter). There are three types of shaders: the geometry shader, the vertex shader (used when vertices are computed) and the pixel/fragment shader (used when pixels are computed). The geometry shader is used to generate graphic primitives such as points, lines or triangles.

Graphics pipeline: this term refers to the ensemble of steps of GPU treatments necessary to create a final image. First, the graphics pipeline processes vertices that can be assembled into primitives such as points, lines, triangles or polygons. During the rasterization step, these primitives are transformed in discrete parts called fragments. Finally, these fragments will be converted into pixels used to form the final image.

Graphics Processing Units (GPUs): component of a graphics card that executes instructions of an informatics program. Until recently, GPUs were dedicated to display operations and graphics data manipulations.

Marching cubes algorithm: algorithm that recreates a triangulated object from a set of discrete points. It is an iso-surface approximation.

Mesh: approximation of the three-dimensional shape of an object by polygons.

Metaballs: graphics technique often used to represent "organic" objects or fluids using polynomial functions.

Mixed complex: Mathematical structure mixing Delaunay tetrahedralization and its dual, the 3D Voronoi diagram. The Delaunay tetrahedralization creates a set of tetrahedra from a set of points (using constraints). Each node of a Voronoi diagram is constituted by the circum centre of each tetrahedron.

Phong shading: lighting technique used to colour surfaces in order to model the reflection of light. This technique is based on the calculation of the angle made between surface normal vector and the light vector. It is obtained by combining different "layers" of colours: the mean layers are the diffuse lighting (i.e. the object colour) and the specular lighting (see hereafter).

Pixel accurate: the pixel, for picture element, is the smallest surface element of an image. It is related to the image resolution: a resolution of 640 × 480 for an image means that this image is composed of 307 200 pixels. A 'pixel accurate' surface is no more approximated by polygons (such as triangles) but by the actual pixels used to display it. For a given resolution, it is the best approximation possible. In general, pixel accurate calculations are performed on the GPU so that the surface approximation can be recalculated on the fly when the camera position is changed.

Ray casting/ray tracing: Computer graphics technique where rays initiated from pixels are cast to detect their intersection with the 3D objects in the scene. The pixel value is then updated according to the object properties. This approach facilitates working with higher order representations of surfaces, avoiding fine tessellations. In this article we consider that 'ray casting' and 'ray tracing' are equivalent to a first approximation. To be precise, ray casting can be thought of as a faster version of ray tracing: in the first case, calculations are performed on primary rays (just taking into account the first intersection) whereas in the second case all intersections are considered with all the 3D objects in the scene.

Remesh: recreate a mesh (see mesh definition above). In general, this approach is used to increase or decrease the number of polygons to better approximate a shape or, oppositely, to diminish the resolution and perform less calculations.

Specular lighting: is related to *phong shading* (see definition above). This method models the brightest (in general white) part of the surface by taking into account the angle between surface and light vectors.

Tessellation: the tessellation of a surface consists in covering it with elements (for example polygons) so that there are no overlaps and no gaps between these elements.

Triangulation/Triangulating: is an approximation of a surface by triangles. It is a special case of the more general tessellation (see above).

Key Points

- New hard- and software based on GPU acceleration enables rendering all-atom representations of very large molecular systems such as virus capsids in real time.
- New algorithms using GPU capabilities, such as Ray-Casting, can precisely and interactively display all types of molecular representations. This approach is generally much faster than current molecular viewers based on surface triangulation.
- New types of representations simplify complexity and highlight significant details on structures in order to focus on a desired area of interest.
- New lighting effects can help to improve the perception of complex shapes of macromolecular assemblies.

FUNDING

This work was supported by the French Agency for Research (grant ANR-07-CIS7-003) and the German Research Foundation (as part of the collaborative Research Centre SFB 716).

References

1. O'Donoghue SI, Goodsell DS, Frangakis AS, *et al*. Visualization of macromolecular structures. *Nat Methods* 2010;**7**:S42–55.
2. Goddard TD, Ferrin TE. Visualization software for molecular assemblies. *Curr Opin Struct Biol* 2007;**17**:587–95.
3. Dematté L, Prandi D. GPU computing for systems biology. *Brief Bioinform* 2010;**11**:323–33.
4. Stone J, Hardy DJ, Ufimstev IS, *et al*. GPU-accelerated molecular modeling coming of age. *J Mol Graph Model* 2010;**29**: 116–25.
5. Stone JE, Phillips JC, Freddolino PL, *et al*. Accelerating molecular modeling applications with graphics processors. *J Comput Chem* 2007;**28**:2618–40.
6. Rost R. OpenGL(R) Shading Language. 2nd edn. Boston, MA: Addison-Wesley Professional, 2006.
7. Fernando R, Kilgard M. The Cg Tutorial: the definitive Guide to Programmable Real-Time Graphics. Boston, MA: Addison-Wesley Longman Publishing, 2003.
8. Sanders J, Kandrot E. CUDA by Example: an introduction to General-Purpose GPU Programming. Boston, MA: Addison-Wesley, 2010.
9. Tsuchiyama R, Nakamura T, Iizuka T, *et al*. The OpenCL Programming Book. Fixstars Corporation, 2010.
10. Derewenda ZS. Rational protein crystallization by mutational surface engineering. *Structure* 2004;**12**:529–35.
11. Heinemann U, Illing G, Oschkinat H. High-throughput three-dimensional protein structure determination. *Curr Opin Biotechnol* 2001;**12**:348–54.
12. Stevens RC. High-throughput protein crystallization. *Curr Opin Struct Biol* 2000;**10**:558–63.
13. Lasker K, Sali A, Wolfson HJ. Determining macromolecular assembly structures by molecular docking and fitting into an electron density map. *Proteins* 2010;**78**:3205–11.
14. Trabuco LG, Villa E, Mitra K, *et al*. Flexible fitting of atomic structures into electron microscopy maps using molecular dynamics. *Structure* 2008;**16**:673–683.
15. Andre I, Bradley P, Wang C, *et al*. Prediction of the structure of symmetrical protein assemblies. *Proc Natl Acad Sci USA* 2007;**104**:17656–61.
16. Goodsell DS. Making the step from chemistry to biology and back. *Nat Chem Biol* 2007;**3**:681–4.
17. Goodsell DS. Visual methods from atoms to cells. *Structure* 2005;**13**:347–54.
18. Sigg C, Weyrich T, Botsch M, *et al*. GPU-based ray-casting of quadratic surfaces. In: Botsch M, Chen B, Pauly M, Zwicker M, (eds). *Eurographics Symposium on Point-Based Graphics*. Boston, MA: Eurographics Association, 2006.
19. Toledo R, Levy B. Extending the graphic pipeline with new GPU-accelerated primitives. INRIA Tech report. http://alice.loria.fr/index.php/publications.html?redirect=0&Paper=ray_tracing_gpu@2004 2004 (1 February 2011, date last accessed).
20. Humphrey W, Dalke A, Schulten K. VMD: visual molecular dynamics. *J Mol Graph* 1996;**14**:33–38, 27–38.
21. Bajaj C, Djeu P, Siddavanahalli V, *et al*. TexMol: Interactive Visual Exploration of Large Flexible Multi-Component Molecular Complexes. *Proc Conf Vis '04* 2004;**15**:243–50.
22. Tarini M, Cignoni P, Montani C. Ambient occlusion and edge cueing to enhance real time molecular visualization. *IEEE Trans Vis Comput Graph* 2006;**12**:1237–44.
23. Daae Lampe O, Viola I, Reuter N, *et al*. Two-level approach to efficient visualization of protein dynamics. *IEEE Trans Vis Comput Graph* 2007;**13**:1616–23.
24. Krone M, Bidmon K, Ertl T. GPU-based Visualization of protein secondary structure. *EG UK Theory and Practice of Computer Graphics*, London, 2008, pp. 1–8.
25. Wahle M, Birmmans S. GPU-accelerated visualization of protein dynamics in ribbon mode. *SPIE, Vis Data Explor Anal* 2011.
26. Kreylos O, Max NL, Hamann B, *et al*. Interactive Protein Manipulation'. *IEEE Vis* 2003;581–8.
27. Connolly ML. Analytical molecular surface calculation. *J Appl Crystallogr* 1983;**16**:548–58.
28. Krone M, Bidmon K, Ertl T. Interactive visualization of molecular surface dynamics. *IEEE Trans Vis Comput Graph* 2009;**15**:1391–1398.
29. Krone M, Dachbacher C, Ertl T. Parallel computation and interactive visualization of time-varying solvent excluded surfaces. *ACM Int Conf Bioinform Comput Biol* 2010;402–5.
30. Chavent M, Levy B, Maigret B. MetaMol: high-quality visualization of molecular skin surface. *J Mol Graph Model* 2008;**27**:209–16.
31. Edelsbrunner H. Deformable smooth surface design. *Discrete Comput Geom* 1999;**21**:87–115.
32. Lindow N, Baum D, Prohaska S, *et al*. Accelerated visualization of dynamic molecular surfaces. *Comput Graph Forum* 2010;**29**:943–52.
33. Kanamori Y, Szego Z, Nishita T. GPU-based fast ray casting for a large number of metaballs. *Comput Graph Forum* 2008;**27**:351–60.
34. Falk M, Klann M, Reuss M, *et al*. 3D visualization of concentrations from stochastic agent-based signal transduction simulations. *Proc IEEE Int Symp Biomed Imaging* 2010; 1301–4.
35. Cipriano G, Gleicher M. Molecular surface abstraction. *IEEE Trans Vis Comput Graph* 2007;**13**:1608–15.
36. Cipriano G, Wesenberg G, Grim T, *et al*. GRAPE: GRaphical Abstracted Protein Explorer. *Nucleic Acids Res* 2010;**38(Suppl.)**:W595–601.
37. Blinn JF. Models of light reflection for computer synthesized pictures. *Int Conf Comput Graph InterTech*, 1977; **11**:192–8.
38. Gillet A, Sanner M, Stoffler D, *et al*. Tangible interfaces for structural molecular biology. *Structure* 2005;**13**:483–91.
39. Sanner M. Python: a programming language for software integration and development. *J Mol Graph Model* 1999;**17**: 57–61.
40. Pettersen EF, Goddard TD, Huang CC, *et al*. UCSF Chimera - a visualization system for exploratory research and analysis. *J Comput Chem* 2004;**25**:1605–12.
41. Moll A, Hildebrandt A, Lenhof H, *et al*. BALLView: an object-oriented molecular visualization and modeling framework. *J Comp Aid Mol Design* 2005;**19**:791–800.
42. Marsalek L, Dehof AK, Georgiev I, *et al*. Real-time ray tracing of complex molecular scenes. *14th Int Conf Inform Vis* 2010;239–45.

43. Lee HC, Varshney A. Representing thermal vibrations and uncertainty in molecular surfaces. *SPIE Conference on Visualization and Data Analysis* 2002;**4665**:80–90.

44. Schmidt-Ehrenberg J, Baum D, Hege HC. Visualizing dynamic molecular conformations. *IEEE Vis* 2002; 235–42.

45. Wiley EA, Deslongchamps G. PostDock: a novel visualization tool for the analysis of molecular docking. *Comput Visual Sci* 2009;**12**:1–7.

46. Phillips M, Dehof AK, Georgiev I, *et al*. Measuring properties of molecular surfaces using ray casting. *HICOMB* 2010; 1–7.

47. Falk M, Klann M, Reuss M, *et al*. Visualization of signal transduction processes in the crowded environment of the cell. *Proc Pacific Vis* 2009;169–76.

48. Weber JR. ProteinShader: illustrative rendering of macromolecules. *BMC Struct Biol* 2009;**9**:19.

49. Cipriano G, Gleicher M. Text scaffolds for effective surface labeling. *IEEE Trans Vis Comput Graph* 2008;**14**:1675–82.

50. Ottino JM. Is a picture worth 1,000 words? *Nature* 2003; **421**:474–6.

51. Kolatkar A, Kenndey K, Halabuk D, *et al*. C-ME: A 3D community-based, real-time collaboration tool for scientific research and training. *PLoS One* 2008;**3**:e1621.