

SVis: A Computational Steering Visualization Environment for Surface Structure Determination

J. Hernando, J. Martínez, V. Martín
Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo, Madrid 28660, Spain
Email: vicente@fi.upm.es

M.F. López, J.A. Martín-Gago
Instituto de Ciencia de Materiales de Madrid
Consejo Superior de Investigaciones Científicas
Campus de Cantoblanco, Madrid 28040, Spain

Abstract—The arrangement of atoms at the surface of a solid accounts for many of its properties: Hardness, chemical activity, corrosion, etc. are dictated by the precise surface structure. Hence, finding it, has a broad range of technical and industrial applications. The ability to solve this problem opens the possibility of designing by computer materials with properties tailored to specific applications. Since the search space grows exponentially with the number of atoms, its solution cannot be achieved for arbitrarily large structures. Presently, a trial and error procedure is used: an expert proposes an structure as a candidate solution and tries a local optimization procedure on it. The solution relaxes to the local minimum in the attractor basin corresponding to the initial point, that might be the one corresponding to the global minimum or not. This procedure is very time consuming and, for reasonably sized surfaces, can take many iterations and much effort from the expert. Here we report on a visualization environment designed to steer this process in an attempt to solve bigger structures and reduce the time needed. The idea is to use an immersive environment to interact with the computation. It has immediate feedback to assess the quality of the proposed structure in order to let the expert explore the space of candidate solutions. The visualization environment is also able to communicate with the de facto local solver used for this problem. The user is then able to send trial structures to the local minimizer and track its progress as they approach the minimum. This allows for simultaneous testing of candidate structures. The system has also proved very useful as an educational tool for the field.

Keywords-Computational Steering, Visualization systems and Software, Graphical User Interfaces.

I. INTRODUCTION: THE PROBLEM OF FINDING THE SURFACE STRUCTURE

Crystalline solids are made up of a basic cell that is repeated throughout space. The atoms at the surface, freed from the interaction of part of its neighbors, rearrange themselves to form a new structure. This structure, typically formed by 3 to 5 atomic layers, is responsible for many of the properties that make a material interesting for industrial purposes: hardness, chemical reactivity, catalytic activity, etc. The procedure to determine it is a two steps one: first an experimental “signature” is extracted and then a

computational procedure tries to find the atomic arrangement that best reproduces the signature. The computational process is one of optimizing the atomic positions, occupancies and several other, model dependent parameters, subject to some constraints and according to some reasonable distance definition. The set of constraints is dependent also on the experimental technique being used. Here we will concentrate on the computational process associated with Surface X-Ray Diffraction (SXRD), a very precise technique but one that needs to use synchrotron radiation. This means that the signature¹ must be obtained in a very expensive shared facility —only about thirty of them are available in the world— where the time allocated for an experiment is at a premium. This makes even more compelling the need of an environment like the one presented here: Since the size of the search space grows exponentially with the number of atoms, there is no guarantee that the surface is going to be fully solved and it is unclear how big or what characteristics must have the experimental signature to allow for a uniquely determined solution. The present environment, that we call SVis, helps in both cases. On one hand, by interactively guiding the search of a solution, the success rate is increased. On the other, by allowing to test for those variables that are most meaningful in a given case, the design of the experiment can be tuned to maximize the usefulness of the experimental signature and reduce the chance of spending expensive beam time to obtain data that, in the end, would be redundant or not meaningful.

II. COMPUTATIONAL STEERING WITH SVIS

Without a global optimizer for this problem, local methods are the only way. From an steering interactive environment perspective [1] such as the present one, the ideal situation would be to have a reliable and fast local minimizer that would allow for a real time feedback. However, this is not the case. The de facto standard method for the SXRD technique [2] takes from hours to days of CPU time to come

¹In this case, the signature is a set of “structure factors”. Since their precise definition is not relevant to the environment, we refer the interested reader to the specialized literature. See [2]

Partially supported by project CAM-UPM 178/Q06 1005-127

out with a possible solution, depending on its complexity. Hence we have splitted the problem by using a method that only quantifies the goodness of a candidate solution and another one that refines it. The former is very fast in order to work interactively. The latter works by sending the candidate solution to the local minimizer and keeping track of its outcome. In this we have used as much as possible of the standard optimization method. This is because it has been extensively tested and all of the users in the field are familiar with its workings. By using it, is much more easy to build trust in the user. Since the local minimizer works essentially as a standalone application, it is easy to use networked resources. In its present form, SVis is able to send candidate solutions to different client computers for local optimization. Communication overhead is low, hence it is well adapted to high latency environments such as a grid.

SVis presents a virtual environment to the user with an extensively configurable layout. This allows also for a good way to organize the big amount of information that is presented in panels lining up the walls of an imaginary room. This environment is to be ideally navigated in an immersive Virtual Reality infrastructure like a CAVE. In order to support different display configurations and devices it has been developed under VRJuggler [3]. Since its main purpose is to visualize and interact with structures in 3D, it is just natural to use this model when proper navigation and manipulation metaphors are provided [5]. At the same time, this decision brings some advantage for the long term: this is the first result of an ongoing project that intends to tackle problems that are becoming increasingly important in surface science and nanotechnology. For instance, solving problems like the positioning of complex molecules on a surface are technologically very important and, as new analysis techniques become of widespread use, so the availability of data for these cases. Working with them, require a sophisticated 3D environment to interact with the complex 3D structures. Further, we plan the inclusion of haptic feedback and extensions to other optimizers. However, immersive environments also pose some problems, not the least the need of sophisticated and expensive hardware. This is alleviated because SVis can take advantage of specialized desktop hardware, such as stereoscopic glasses, and present a 3D environment without the need of a CAVE.

Working in a virtual environment using traditional human-computer interfaces can be troublesome, for this reason, the metaphors used to navigate and interact with the data are different in the immersive and desktop configurations. For example, rotating an object uses the real wand orientation in the immersive environment, but uses screen aligned axes and the mouse movement in the desktop. Also, the environment is highly customizable and extra windows can be opened with fixed views to increase its usability when executed using only an standard computer monitor. The working space

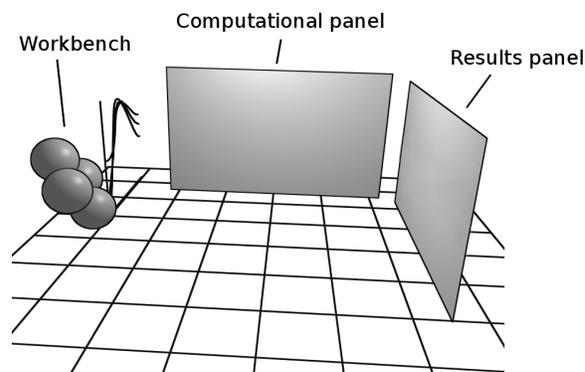


Figure 1. Scheme of the SVis environment. The Workbench is where the editing of the surface takes place. In that wall, also the experimental signatures are presented together with the calculated ones. The Computational panel holds the edited surfaces. A surface can be recalled from it for further modification whenever needed and also it can be sent to the local minimizer. Results are received in the Results panel. A surface can also be picked for editing from the Results panel.

is arranged in a workbench and two floating sets of panels. An scheme is given in Fig. 1. An extra accessory window is included as a default option to ease the work in a standard monitor. The configuration of environment layout is done through an XML file which is read at start-up.

Some information must be provided to start SVis. Again, this is done using the same structures used for the standard program [2] in order to make it easier to the existing user base. This information includes a guess solution, which serves as a starting point for work, and the set of structure factors that conforms the experimental signature. At program start up, the main panel appears in a front view, as presented in Fig. 2. It contains the candidate solution in the workbench, ready to be modified, a general view presenting a customizable number of structure factors graphs and an enlarged view with the full set of graphs arranged as a stack. Only the first one is presented, but the user can flip through the stack by using the arrows at the corners. Each graph contains two curves, the experimental signature and the one theoretically derived from the surface being modified in the workbench. The calculated one is updated in real time whenever an atom is moved and a χ^2 value, to measure the discrepancy among the candidate solution and the experimental value, is presented. This fast update allows to explore the search space and pinpoint those atoms and movements that affect more strongly to the signature. In order to move the atoms with the required precision, whenever an atom in the workbench is selected a 3D set of axes are drawn, with the axis origin centered at the atom's last position. The movement is constrained to the axes and scaled down, so that it can be accurately positioned either using a pointing device in a CAVE or with the mouse when using the basic 2D interface.

A secondary window that is always presenting a front

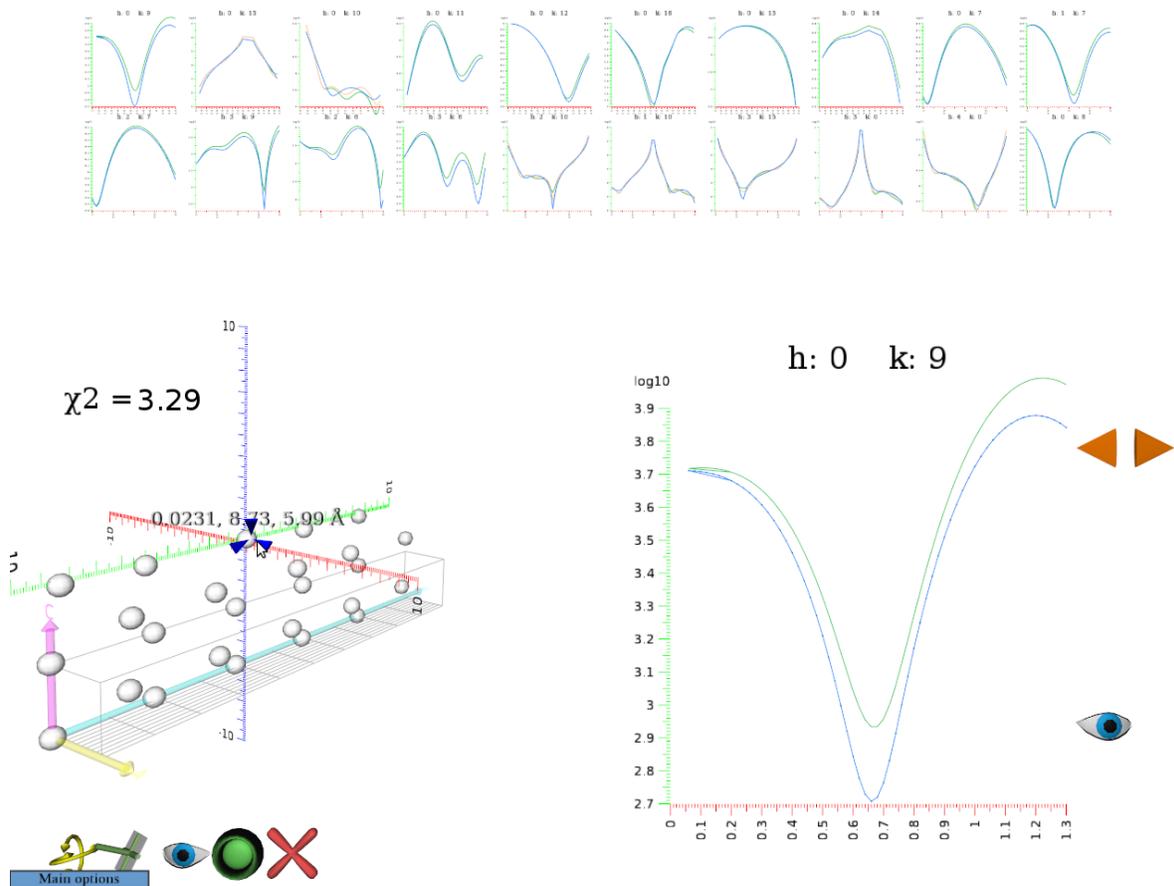


Figure 2. Main panel view at start up: the workbench. The upper part presents the set of structure factors that conform the experimental signature, the lower right shows the detailed view of one of these. In the lower right corner the working surface is presented, with one of the atoms selected for its modification.

view of the stack of structure factors is intended as an extra aid for the 2D interface. A consequence of the design decision of using an immersive environment is that the user can actually navigate through the whole environment. Although the fact that the surface can be viewed from any angle, even from the inside, can be a welcome addition when modifying a surface, it can also be weird to see a panel from the side: an essentially 2D object is represented in the environment as a 3D object with just an small depth. In order to minimize this inconveniences when using the 2D mode, a set of buttons at the bottom of each panel automatically perform operations like restoring the point of view or reset the viewing angle.

The green, round, button in the workbench serves to send the edited surface to the computational panel, the one that is on the right hand of the user at start up. The computational panel serves the purpose of keeping the user guesses. From it, a surface can be recalled for further editing or sent directly to the local optimizer. This is done providing a full shell for the execution of the optimizer. In this way is easy to use a

network of computers to do the calculations, thus increasing the throughput in a cost efficient manner. The results of the calculations are received in the corresponding positions in the output panel, behind the user at start up, from where they can be picked up again for editing in the workbench if needed to continue the steering process.

III. CONCLUSION

We have described an environment to steer the computational process of finding the surface structure from SXRD data. Its aim is to substitute the costly trial and error procedure followed to date. Some of the design decisions include to use a 3D immersive environment as a base. This is a natural election to guide a process that is essentially performed in physical space and that has many more data associated than can comfortably be presented using a desktop environment. However, the fact that not many researchers have access to such environments, lead us to consider a basic usage by using a 2D environment. Some extra aids are included for this case, and experience

has demonstrated that the immersive paradigm works also quite well, in this case, in desktop usage. The environment is currently under testing and plans are being laid to leave it publicly available at one of the major synchrotron radiation facilities [4]. Next logical step is adding haptics feedback to link the improving or worsening of the discrepancy measure with the resistance that poses the atom to the movement. It would also be desirable to speed up the local optimization procedure, either by judicious sequential tuning or resorting to the parallelization of the optimizer, as opposed to the parallel processing of independent candidate structures as is done in the present environment. In doing this, a balance must be strike by considering the speed at which one solution is optimized against the total throughput, specially since throughput is much easier to obtain with the present scheme that can tolerate loosely connected environments such as a grid or a LAN. The ability of testing and tracking several solutions at the same time could be more rewarding than the ability to test just one although faster. A further modification would be to adapt the environment to steer new optimization algorithms, in particular, population based algorithms, having very different characteristics than the one used now.

REFERENCES

- [1] M. Riedel *et al.* "Computational Steering and On line Visualization of Scientific Applications on Large-Scale HPC Systems within e-Science Infrastructures," Proceedings of the Third IEEE International Conference on e-Science and Grid Computing, p. 483, 2007.
- [2] E. Vlieg, "ROD: A program for surface X-ray crystallography." *J. of Appl. Cryst.* vol. 33, p. 401, 2000.
- [3] A. Bierbaum, C. Just, P. Hartling, K. Meinert, A. Baker, C. Cruz-Neira. "VR Juggler: a virtual platform for virtual reality application development." *Proc. Virtual Reality 2001 Conference (VR'01)*, IEEE Computer Society, p. 89, 2001.
- [4] The European Synchrotron Radiation Facility. <http://www.esrf.eu/>
- [5] A. van Dam, A.S. Forsberg, D.H. Laidlaw, J.J. LaViola R.M. Simpson. "Immersive VR for Scientific Visualization: A Progress Report" *IEEE Comp. Graph. and App.* vol. 20, p. 26, 2000.