Adaptive Visuo-Haptic Rendering for Hybrid Modeling of Macromolecular Assemblies

Stefan Birmanns[1], Maik Boltes[2], Herwig Zilken[2], Willy Wriggers[1]

[1]University of Texas Health Science Center at Houston, School of Health Information Sciences

[2]Research Center Jülich, Central Institute for Applied Mathematics, Germany

Abstract— **We describe an immersive visualization system for structural biology using real-time load balancing of virtual reality and haptic rendering. In structural biology a variety of image reconstruction techniques are employed to determine geometric aspects of large macromolecular assemblies at various levels of resolution. Hybrid modeling techniques are the most promising approach to bridge the resolution gap between data from different biophysical origins. Here we present an interactive solution for the multi-resolution modeling problem that employs a novel load-balancing scheme for the visual and haptic rendering. During critical load conditions the error metric of a dynamic mesh simplification technique is coupled to the force-update rate of the kinesthetic feedback, thereby providing an instantaneous adaptation of the rendering to the modeling efficiency. This enables the construction of nano-scale bio-molecular architectures while avoiding unwanted haptic blackouts during peak modeling demand. Combined with various molecular rendering techniques, adaptive visuo-haptic rendering is embedded in our graphics system "SenSitus".**

I. Introduction

The structure and function of macromolecular cellular machines [1] are routinely investigated in concurrent biology and medicine. Most fundamental cellular processes involve the actions of nano-scale bio-molecular assemblies in a precise and organized way. The mechanism of such a machine can be defined by the dynamic states of all its components in their correct time sequence. The complexity and size of molecular assemblies and the large number of conformations make it unlikely that all the states can be observed directly at atomic resolution. Electron microscopy (short EM) is a powerful technique to gain information about the entire macromolecule in a variety of conformations, but the volumetric data sets generated at the end of the image reconstruction process typically feature only spatial resolutions of around 20Å. Identifying structures at atomic resolution within the low-resolution EM map would enable a detailed understanding of macromolecular interactions.

A. Algorithmic Solutions

Within the past decade, multi-resolution molecular modeling has emerged as a powerful strategy to bridge the resolution gap between crystallographic high-resolution structures and EM volumetric maps. The atomic structure of the entire assembly is constructed by docking probe molecules into the low-resolution volumetric map, following a successive “building-block” strategy. An example can be seen in Fig. 1, where six high-resolution atomic structures (protein data bank [2] entry 2REC) are fitted into a helicase volumetric map [19].

sbirman@biomachina.org
m.boltes@fz-juelich.de
h.zilken@fz-juelich.de
wriggers@biomachina.org

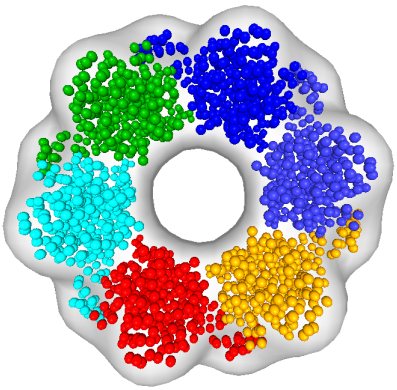

Fig. 1. Multi-resolution molecular modeling: Six high-resolution monomeric structures docked into a volumetric map. The monomers are encoded by different colors and represented by the C_α atoms of the poly-peptide chain.

Computational biology research groups have proposed several algorithmic solutions to the 3D multi-resolution modeling problem [17]. The majority of the matching algorithms adopt voxel-oriented techniques based on the correlation coefficient [16], [11], [13] as a fitting criterion. Typically an exhaustive search is carried out that maximizes the cross-correlation

$$C(\mathbf{R},\mathbf{T}) = \int \rho_{em}(\mathbf{r})\,\rho_{calc}(\mathbf{r},\mathbf{R},\mathbf{T})\,d^3\mathbf{r} \qquad (1)$$

between the low-resolution EM map ρ_{em} (target) and the low-pass filtered map ρ_{calc} derived from the atomic structure (probe). Although successful, this approach of fitting the probe to the target suffers from the ambiguity of the correlation coefficient and is thereby not applicable to all problem settings [17].

The multi-resolution docking can also be done visually [8] by exploiting the shape similarity of the two objects. Although advanced graphics workstations are able to render the data sets interactively today, the shape comparison is still a challenging task for the human visual system because of the complex three-dimensional structure of the objects. Therefore the docking process is complicated and time-consuming for large macromolecular systems.

B. Interactive Interior Docking

We proposed in [3] to utilize the human tactile sense to communicate information about the quality of matching the data to the user. In our solution the gradient of the cross-correlation coefficient C (Eq. 1) is used as potential function for the force

calculation of the kinesthetic feedback. The resulting force and torque guides the user towards a better fitting location and orientation in the sense of a higher correlation of probe and target. The user simultaneously maximizes the shape similarity criterion by visual docking, which in turn alleviates the ambiguity problem of the correlation coefficient.

In practice, the kHz force-update rate required for a high-quality haptic rendering [4] prohibits a straightforward implementation of this idea. The calculation of the cross-correlation coefficient is too expensive to be employed in a real-time setting with the required rate. To reduce the computational complexity we utilize a topology representing network (TRN) [10], [3]. The TRN performs a clustering of 3D data termed *vector quantization*, where the molecule is represented by so-called *codebook vectors* $\mathbf{w}_i$ (see Fig. 2).

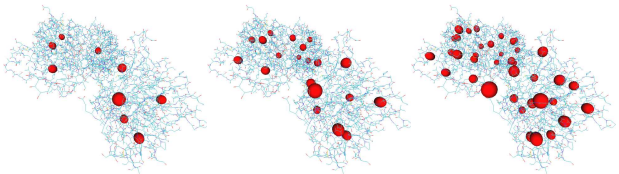

Fig. 2. Atomic structure represented by sets of 10, 20 and 40 codebook vectors (perspective view)

Given N codebook vectors $\mathbf{w}_i$, we approximate ρ_{calc} (Eq.1) by a sum of Dirac delta functions:

$$\rho_{calc}(\mathbf{r},\mathbf{R},\mathbf{T}) \approx \sum_{i=1}^{N} \delta(\mathbf{r}-\mathbf{w}_i(\mathbf{R},\mathbf{T}))$$

which leads to the following simplified form of the correlation coefficient:

$$C(\mathbf{R},\mathbf{T}) = \sum_{\mathbf{i=1}}^{\mathbf{N}} \rho_{\mathbf{em}}(\mathbf{w}_i(\mathbf{R},\mathbf{T})) \qquad (2)$$

The reduced criterion (Eq. 2) performs well for numbers N that are 2-3 orders of magnitudes lower than the number of atoms of the probe molecule [3]. The simplified coefficient is then used as a potential function to calculate a force and torque acting on the probe molecule. It can be shown that the force acting on the center of mass (COM) of the rigid-body is equal to the gradient of the EM density sampled at the codebook vector positions

$$\mathbf{F} = \sum_{i=1}^{N} \nabla\rho_{em}(\mathbf{w}_i(R,T)),$$

and the torque acting on the COM can be computed in a similar manner

$$\mathbf{Q} = \sum_{i=1}^{N} \mathbf{w}_i(R,T) \times \nabla\rho_{em}(\mathbf{w}_i(R,T)),$$

where "$\times$" denotes the vector cross-product. The rotation $\mathbf{R}$ and the translation $\mathbf{T}$ are parameters measured by the sensors of the haptic device, as the molecule is moved about interactively by the user.

With the simplified coefficient we are able to achieve force-update rates of > 2 kHz for test data sets quantized with $N = 200$ codebook vectors, using a standard dual CPU personal computer.

C. Mesh Simplification

The combination of the before mentioned haptic rendering with an immersive visualization of the data sets introduces a competition for the available processor time. Haptic and visual rendering are real-time processes and the quality of both suffers significantly if the refresh rates drop below a critical level. As mentioned before, a force updates per second (short FUS) rate of around 1 kHz is necessary to generate a high-quality haptic rendering [4]. Similarly, the visual rendering in an immersive virtual reality environment is more demanding than in a standard workstation setting, and refresh rates of around 30 frames-per second (short FPS) are required for a realistic visual perception.

In this work we utilize a mesh simplification algorithm for a load-balancing scheme of the haptic and visual rendering. We will develop an adaptive version of the algorithm that allows us to adjust the complexity of the triangular mesh according to the current force-refresh rate of the kinesthetic feedback. Thereby we can avoid a dominant usage of computer resources by the visual rendering and ensure the generation of high-quality force-feedback.

II. Adaptive Mesh Simplification

A variety of simplification schemes have been proposed for triangular meshes in the past. One can broadly group these algorithms into three different categories: vertex clustering [12], vertex decimation [14] and edge contraction [7]. Vertex clustering can be very fast and offers a global error bond based on the grid the vertices are divided into, but the quality of the meshes can be quite low. Vertex decimation can produce a very accurate approximation, but the process is often slow and only applicable to manifold surfaces.

In this paper we employ a pair contraction algorithm, based on Garland and Heckbert's work [6]. Their algorithm seems to be ideal for our application as it offers generality, high quality approximation and a fast simplification process. Following the algorithm, we contract two vertices and replace them by a single new vertex, which will result in a simplification of the mesh. To maintain a high-quality approximation a cost function is utilized to select the next contraction pair. To define this cost function we employ a quadric error metric to describe the error at a certain vertex, based on the distance of the vertex to the surface of the mesh. The error metric serves as an energy function with the optimal new vertices - resulting from the vertex contraction - located at the valleys of the energy landscape.

To find possible candidates for the contraction step, we first compile a list of valid vertex pairs. In addition to vertices that

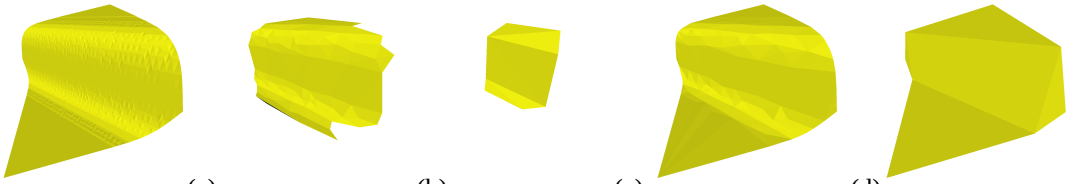

Fig. 3. Model with open boundaries - (a) original model (1250 vertices), (b) and (c) w/o boundary protection (100 and 8 vertices), (d) and (e) with protection (100 and 8 vertices)

share an edge we include also vertices

$$\|\mathbf{v}_1 - \mathbf{v}_2\| < t$$

that enable support of non-manifold surfaces (the above criterion can also be interpreted as adding an "imaginary" edge). The parameter t should be selected carefully, as a high value would lead to $O(n^2)$ pairs.

In a next step we develop a measure based on the vertex distance to the original surface, in order to select a candidate pair for the contraction. Therefore we define the Quadric Q of a vector $\mathbf{v} = (\mathbf{x}, \mathbf{z}, \mathbf{y}, \mathbf{1})^\mathbf{t}$ as the sum of the quadric distance between the vertex and a set of planes, defined by $\mathbf{n}_i = (\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d})^\mathbf{t}$ with $\mathbf{n}_i^\mathbf{t}\mathbf{x} = \mathbf{0}$:

$$Q(\mathbf{v}) = \sum_{\mathbf{i=1}}^{\mathbf{k}} (\mathbf{n}_i^\mathbf{t}\mathbf{v})^2 = \mathbf{v}^\mathbf{t}\mathbf{Q}\mathbf{v}$$

which leads to a cost function for each vertex. The set of planes is defined by the triangles that meet in $\mathbf{v}$. To describe the cost of a certain vertex pair, we choose an additive rule to model the case of a contraction $(\mathbf{v}_1, \mathbf{v}_2) \rightarrow \bar{\mathbf{v}}$:

$$\bar{\mathbf{Q}} = \mathbf{Q}_1 + \mathbf{Q}_2$$

This simple rule follows the idea that the cost function of the new vertex $\bar{\mathbf{v}}$ depends on the planes defined by the two vertices $\mathbf{v}_1$ and $\mathbf{v}_2$ - the larger the distance of $\bar{\mathbf{v}}$ to the original surface, the more expensive the contraction $(\mathbf{v}_1, \mathbf{v}_2) \rightarrow \bar{\mathbf{v}}$ would be. The final cost value $Q(\bar{\mathbf{v}})$ used for a ranking of the vertex pair depends on the concrete vertex $\bar{\mathbf{v}}$, which is so far still undefined. The optimal vertex $\bar{\mathbf{v}}$ is the vertex for which $Q(\bar{\mathbf{v}})$ gets minimal, leading to the following linear problem:

$$\begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \bar{\mathbf{v}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

If the matrix is not invertible, a reasonable approximation for the optimal vertex can be found on the edge between $\mathbf{v}_1$ and $\mathbf{v}_2$. The procedure is used to acquire $\bar{\mathbf{v}}$ and the corresponding quadric $Q(\bar{\mathbf{v}})$ for all the vertex pairs. The simplification process selects the best vertex pair for the contraction, relying on $Q(\bar{\mathbf{v}})$ as criterion.

A. Boundary Protection

Here the described algorithm is applied to general triangular meshes, that in structural biology often result from an isosurface generation. It is not unusual that these meshes exhibit open boundaries, but in this case the standard form of the algorithm will fail to preserve the shape of the object. As the edges at the boundary are only part of a single triangle, they are very likely early candidates for the contraction, resulting in a zig-zag appearance of the borders of the model.

To protect the boundaries we introduce an imaginary plane at the vertices $\mathbf{v}_i$ and $\mathbf{v}_j$, if the edge $(\mathbf{v}_i, \mathbf{v}_j)$ is only part of a single triangle. The imaginary plane is perpendicular to the existing triangle, such that the quadric error metric will tend to move the new vertices $\bar{\mathbf{v}}$ towards the boundary and thereby preserve the overall shape of the object. If the triangle at $\mathbf{v}_i$, $\mathbf{v}_j$ features the normal vector $\mathbf{n}$, the new triangle is defined by:

$$\mathbf{n}^* = \frac{\mathbf{n} \times (\mathbf{v}_j - \mathbf{v}_i)}{\|\mathbf{n} \times (\mathbf{v}_j - \mathbf{v}_i)\|}$$

Initial experiments were promising, but sharp boundaries were still vanishing at a relative early stage. Therefore we further pronounced the boundary protection by applying a scale factor f to the quadric. Thereby $\bar{\mathbf{v}}$ is even more attracted by the boundary, which leads for $f = 1000$ to the results shown in Fig. 3. The boundaries of the object are preserved even at a very high level of compression, e.g. when the original mesh of 1250 vertices is reduced to only 8 vertices. In comparison, the original algorithm cannot preserve the shape of the object, already at a simplification level of 100 vertices the approximation is very coarse.

The "invisible" surface is not rendered, but otherwise not different from the other surfaces. Thereby this boundary protection method can be integrated into the algorithm without major modifications.

We applied the final algorithm to biological data sets. Three flat shaded isosurfaces of the CCT chaperonin [9], [18] are shown in Fig. 4. The first picture shows the initial isosurface generated by the marching cube algorithm (12,835 vertices). Then a simplified isosurface of only 2000 vertices - a reduction of about 86 percent regarding the original surface - is drawn, that clearly preserves most of the detail information. Although the overall appearance of the third picture is quite coarse (98% compression), one still can detect the most important features of

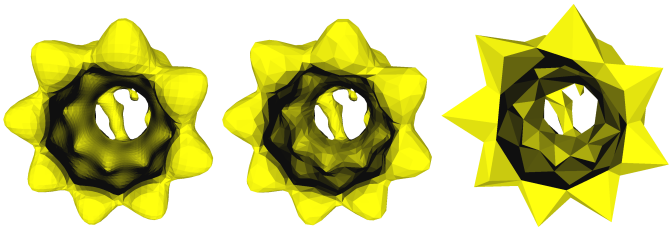

Fig. 4. Isocontour of the CCT chaperonin - 12,835 vertices, 2,000 vertices, 200 vertices

the surface (for example the eight peaks at the front and back seam of the cylindrical structure).

B. Progressive Encoding

The algorithm can produce meshes with an arbitrary number of vertices by iteratively applying the vertex contraction step. In a real-time environment the simplification procedure has to be applied as pre-processing step in order to allow an instantaneous adaption of the mesh complexity to the modeling situation. Since storing n meshes would lead to an excessive memory consumption, we combine the algorithm with the progressive mesh data structure proposed by [7]. In his algorithm not only the simplified version of a mesh is stored, but also the vertex contraction steps that led to the coarser model. The benefit is, that one cannot only reconstruct the original mesh, but also all intermediate stages, without storing the meshes explicitly.

An initial mesh $\hat{M} = M_n$ can be simplified into a coarser mesh M_0 by performing a sequence of vertex contraction operations:

$$M_n \underset{\psi_n^{-1}}{\overset{\psi_n}{\rightleftharpoons}} M_{n-1} \underset{\psi_{n-1}^{-1}}{\overset{\psi_{n-1}}{\rightleftharpoons}} \ldots \underset{\psi_1^{-1}}{\overset{\psi_1}{\rightleftharpoons}} M_1 \underset{\psi_0^{-1}}{\overset{\psi_0}{\rightleftharpoons}} M_0$$

with $\psi_k(M_k) = M_{k-1}$ (vertex contraction) and $\psi_k^{-1}(M_{k-1}) = M_k$ (vertex split), $k \in \{1,\ldots,n\}$. By performing these steps we can simplify or refine any mesh M_k and reach thereby any other version of the mesh M_l.

To perform the operation ψ_k and it's inverse ψ_k^{-1} one has to build a data structure that can be traversed easily and at the same time is as compact as possible. For the vertex contraction ψ_k we store only the indexes of the vertex pair $(\mathbf{v}_i, \mathbf{v}_j)$ and the coordinates of the new vertex $\bar{\mathbf{v}}$. The inverse operation ψ_k^{-1} is more complex, as two new vertices are generated whose relationship to the existing and new triangles has to be specified.

To keep the progressive mesh data structure compact we encode the data in a concise way. Storing a difference vector between two vertices has the benefit that only relative small numbers have to be encoded, which makes more efficient use of the limited range of standard precision floating point variables. For the contraction ψ_k we assume that $\bar{\mathbf{v}} = \mathbf{v}_i + \mathbf{t}$, so that we only have to encode the difference vector $\mathbf{t}$. As $\bar{\mathbf{v}}$ represents $\mathbf{v}_i$ in the new mesh, we also avoid storing another vertex index.

The encoding of the splitting step ψ_k^{-1} is more memory intensive. Again we encode one vertex of the splitting pair $\mathbf{v}_i$ as a difference vector $\mathbf{v}_i = \bar{\mathbf{v}} + \mathbf{t}$. But in addition one also has to maintain two triangle lists. The first list stores the indexes of the triangles, affected by the splitting of $\bar{\mathbf{v}} \rightarrow (\mathbf{v}_i, \mathbf{v}_j)$. As $\mathbf{v}_i$ features the vertex index of $\bar{\mathbf{v}}$, nothing changes for the triangles that have $\mathbf{v}_i$ as a vertex. One only has to include triangles indexes in the list that are defined by $\mathbf{v}_j$. On average a vertex is part of six adjacent triangles, but as two of them are new triangles and are included in the next list, we have to store in the first list on average four indexes. The second list administers the indexes of the new triangles, which are two on average.

On average one has to store nine floating point and eight integer variables for each combined split and contraction step. Since the described data structure is so compact, todays graphics workstations can hold very large meshes with more then 100,000 vertices in main memory for an interactive adaptive simplification.

C. Adaptation

Preprocessing the mesh in the above described way enables us to select the level of detail according to an external measure. In our application we adjust the number of vertices v_{cnt} dynamically according to a target measure r_g and actual measure r_c:

$$v_{cnt} := v_{cnt} - s \cdot v_{cnt} \cdot \frac{r_g - r_c}{r_g} \text{ if } r_c < (1-c) \cdot r_g$$

$$v_{cnt} := v_{cnt} + s \cdot v_{cnt} \cdot \frac{r_g - r_c}{r_g} \text{ if } r_c \geq (1-c) \cdot r_g$$

In a pure visual rendering scenario a natural measure would be the current FPS rate. By selecting a target FPS rate, the user can balance the interactivity and the detail level of the rendering. As described above, the temporal bandwidth of the human tactile sensoric system is much higher than of the visual system. Therefore the level of detail of the visual rendering is adjusted such that the consumed processor performance still allows high-quality haptic rendering. In our application r_g and r_c are the target and the actual force-update rate.

D. Results

We performed several test docking sessions to investigate the practical properties of the described approach. A monomeric structure (protein data bank entry 2REC) sampled by 200 codebook vectors was docked into a volumetric map of RecA helicase [19]. In Fig. 5 the plot of the recorded FUS rate is pictured.

Curve C represents the first test session, where the standard rendering without adaptive mesh simplification was used. The FUS rate oscillates around 550, which is a relative low value where only a limited quality of the haptic rendering can be achieved. Curve B represents a test session where the adaptive simplification with a goal FUS rate of $r_g = 800$ was used. Clearly the actual FUS rate r_c deviates only insignificant from r_g. Almost the same behavior can be observed at curve A where r_g with a value of 1200 was used. In this case the computer performance was exhausted, as it was impossible to achieve higher force recalculation rates - even in absence of visual rendering.

With the proposed method we were able to double the FUS rate and thereby a fully satisfactory haptic rendering was possible. To increase the FUS rate the adaption had to reduce the complexity of the triangular mesh significantly by almost 72%, but the visual representation was still sufficient as can be seen in Fig. 6.

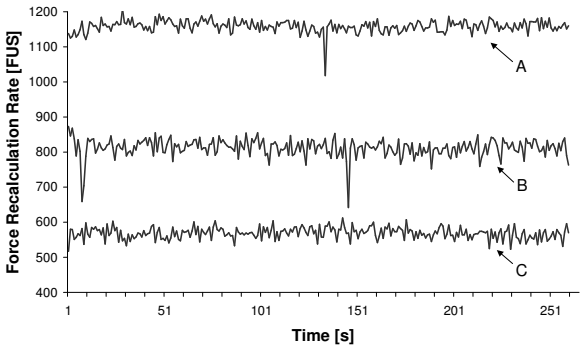

Fig. 5. Force recalculation rate during interactive docking session - curve A w/o adaption, curve B with $r_g = 800$, curve C with $r_g = 1200$

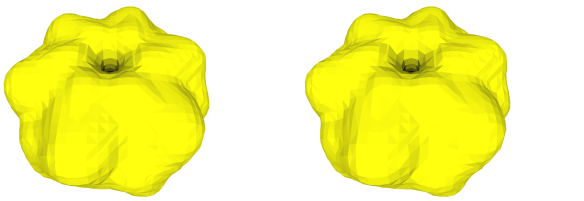

Fig. 6. RecA helicase isosurface rendered with (a) 10140 and (b) 2776 triangles

After these positive results we tested how the adaption would react on the influence of an external CPU load. Therefore another program was executed during an interactive docking session. The program performs random-number based calculations in order to consume CPU performance. Again we tested the behavior of the standard approach without adaptation. In Fig. 7 one can observe that in this case the force recalculation rate drops in presence of the background process down to about 250 FUS. Although the force update rate oscillates more noticeable when the additional process is present, the adaptive visuo-haptic rendering method - shown in Fig. 8 - is successful in maintaining a high force update rate near r_g.

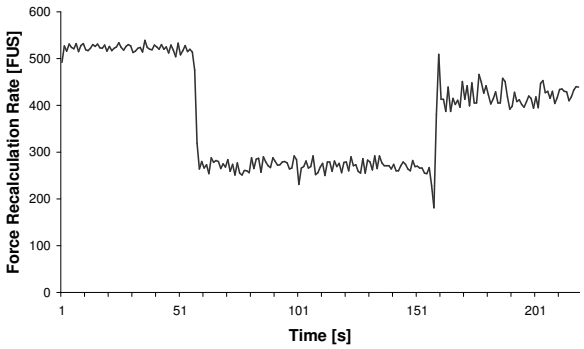

Fig. 7. Standard algorithm under influence of an external CPU load

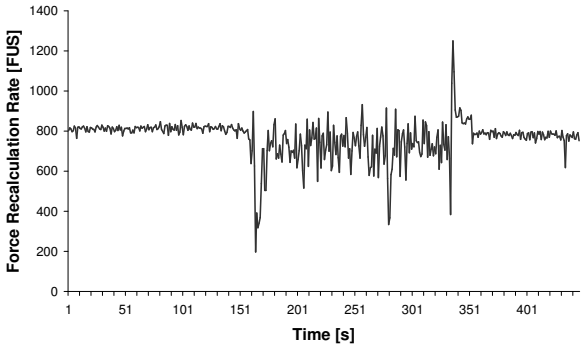

Fig. 8. Adaptive visuo-haptic rendering under influence of an external CPU load

III. Virtual Reality Setup

The adaptive visuo-haptic docking molecular approach was integrated in a Virtual Reality environment. Our VR setup is a self-built, low-cost back-projection system (see Fig. 9), employing polarization filters and inexpensive polarized glasses for stereoscopic viewing. Two DLP projectors ensure an exceptional brightness of the image, which is projected to a polarization preserving screen. A system of two mirrors fold the light-path, reducing the distance required between the projectors and the screen when using back-projection. An electro-magnetic tracking system measures the position and orientation of up to four sensors, whereby one sensor is attached to a pair of glasses to determine the viewers position and viewing direction. On runtime the computer is able to adjust the 3D projection according to the users standpoint and thereby gives the impression of a stationary scene.

Our setup serves as a prototype VR system for structural biology laboratories. As these research groups have to maintain complex and expensive experimental machinery, there is typically no tendency to install a large immersive virtual environment like a CAVE [5]. The system presented here is inexpensive and also compact - it has the footprint of a standard cubicle. Still it offers full interactive, head-tracked, stereoscopic imaging like a conventional VR environment and a haptic device which delivers kinesthetic feedback for multi-modal user interaction. For the haptic rendering we use a kinesthetic device, a SensAble Phantom 6DOF [15], which offers full force- and torque feedback and sensors.

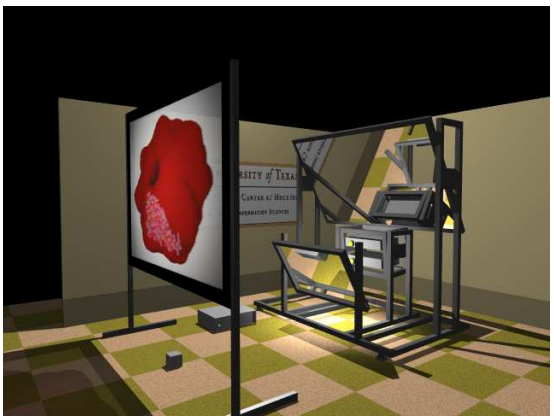

Fig. 9. Virtual Reality System

IV. Summary and Conclusion

We have described an interactive system for the multi-resolution docking problem in structural biology. The initial idea of using haptic rendering for the exploration of the cross-correlation coefficient "landscape" was implemented and combined with an immersive visualization system. The challenge of such a combination lies in the real-time nature of both haptic and visual rendering. Since the haptic rendering has higher performance requirements than the immersive visualization, we decided to reduce the complexity of the visual rendering during a critical load situation. This is done by a mesh simplification algorithm that was modified to deliver better support for our triangular models. In addition the final algorithm was embedded in a progressive mesh data structure to provide adaptive rendering capabilities. The quadric error metric was then coupled to the recalculation rate of the kinesthetic feedback in order to prioritize the haptic rendering quality.

Our approach cannot guarantee under all circumstances to stabilize the FUS rate precisely at r_g, as there is in principle no direct relationship between the visual and haptic rendering. However we can measure the indirect performance loss of the kinesthetic feedback in form of the FUS rate, caused in part by the visualization. In practice this load balancing scheme proved to be quite stable and was also able to cope with external CPU load. Maintaining a high force-update rate enables us to target macromolecular assemblies, whereby in the previous approach the visualization of these large complexes would have dominated the usage of the computing resources.

In the future we would like to enhance the human-computer interaction mechanisms. In order to support the construction of large assemblies, a visual browser of the available building-blocks would be beneficial during the docking session. The load-balancing of the visuo-haptic rendering enables the construction of very large (millions of atoms) biomolecular complexes by modeling the "exterior docking" of a probe molecule to alreadyplaced subunits. A successive difference mapping approach can be applied to provide for a soft repulsive force in the presence of steric hindrance among placed structures. However, to implement this in the current conceptual framework, the load-balancing is critical, since the complexity and size of the building blocks is frequently changing when they are selected for the matching by the user. Therefore, the real-time character of the haptic rendering will facilitate the future implementation of novel concepts in interactive 3D modeling.

V. Acknowledgments

We thank Dr. Rüdiger Esser, Dr. Marcia O'Malley and Dr. Lennart Johnsson for discussions. This work was supported by NIH grant 1R01-GM62968, by Human Frontier Science Program grant RGP0026/2003 and by a training fellowship from the W.M. Keck Foundation to the Gulf Coast Consortia through the Keck Center for Computational and Structural Biology.

References

[1] B. Alberts. The cell as a collection of protein machines: Preparing the next generation of molecular biologists. *Cell*, 92:291–294, 1998.

[2] F.C. Bernstein, T.F. Koetzle, G. J. B. Williams, E. F. Jr. Meyer, M.D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi. The protein data bank: A computer-based archival file for macromolecular structures. *J. Mol. Biol.*, 112:535–542, 1977.

[3] Stefan Birmanns and Willy Wriggers. Interactive fitting augmented by force-feedback and virtual reality. *J. of Struct. Biol.*, pages 123–131, 2003.

[4] E. Chen and B. Marcus. Force feedback for surgical simulation. *Proceedings of the IEEE*, 86(3):524–530, 1998.

[5] Carolina Cruz-Neira, Daniel J. Sandin, and Thomas A. DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the cave. In *ACM Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 135–142, 1993.

[6] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216. ACM Press/Addison-Wesley Publishing Co., 1997.

[7] Hugues Hoppe. Progressive meshes. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 99–108. ACM Press, 1996.

[8] T. A. Jones, J.-Y. Zou, and S. W. Cowan. Improved methods for building protein models in electron density maps and the location of errors in these models. *Acta Cryst. A*, 47:110–119, 1991.

[9] Oscar Llorca, Jaime Martín-Benito, Monica Ritco-Vonsovici, Julie Grantham, Gillian M. Hynes, Keith R. Willison, José L. Carrascosa, and José M. Valpuesta. Eukaryotic chaperonin cct stabilizes actin and tubulin folding intermediates in open quasi-native conformations. *EMBO Journal*, pages 5971–5979, 2000.

[10] T. Martinetz and K. Schulten. Topology representing networks. *Neural Networks*, 7:507–522, 1993.

[11] Alan M. Roseman. Docking structures of domains into maps from cryo-electron microscopy using local correlation. *Acta Cryst. D*, 56:1332–1340, 2000.

[12] J. Rossignac and P. Borrel. Multi-resolution 3d approximations for rendering complex scenes. *Modeling in Computer Graphics: Methods and Applications*, pages 455–465, 1993.

[13] Michael G. Rossman. Fitting atomic models into electron-microscopy maps. *Acta Cryst. D*, 56:1341–1349, 2000.

[14] William J. Schroeder, Jonathan A. Zarge, and William E. Lorensen. Decimation of triangle meshes. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 65–70. ACM Press, 1992.

[15] SensAble Corporation. Phantom force-feedback devices. http://www.sensable.com.

[16] Niels Volkmann and Dorit Hanein. Quantitative fitting of atomic models into observed densities by electron microscopy. *J. of Struct. Biol.*, 125:176–184, 1999.

[17] W. Wriggers and P. Chacón. Modeling tricks and fitting techniques for multiresolution structures. *Structure*, 9:779–788, 2001.

[18] Willy Wriggers, Pablo Chacón, Julio Kovacs, Tama Florence, and Stefan Birmanns. Topology representing neural networks reconcile biomolecular shape, structure, and dynamics. *Neurocomputing*, pages 365–379, 2004.

[19] X. Yu and E. H. Egelman. The reca hexamer is a structural homologue of ring helicases. *Nature Struc. Biol.*, pages 101–104, 1997.