Physical Simulation of the Visible Human Joints

Chandrajit Bajaj Fausto Bernardini Kwun-Nan Lin Elisha Sacks Dan Schikore

Shastra Lab & Center for Image Analysis and Data Visualization
Computer Science Department
Purdue University*

1 Introduction

We are developing modeling and physical simulation tools for human joints reconstructed from the CT, MRI, and photograph imaging data sets of the visible human. The objectives are non-invasive diagnostic tools, systematic preoperative planning, custom prosthesis design, and quantitative models of strain injuries. The research challenges are to compute the physics directly from the data with minimal user intervention and to compute at interactive speed despite the great size of the data sets. We address these challenges by hierarchical modeling, distributed computing, and specialized physical simulations.

The modeling tools of our system construct geometric spline and finite element models of bones from the CT data and tendon/ligaments from the MRI and photographic data. The geometric models are extended to capture material properties, such as mass distribution, elasticity, and friction coefficients. The elasticity and friction coefficients are approximated using qualititative mathematical functions of the CT and MRI intensities [22, 21]. The simulation module computes the joint physics (motions, forces, deformations) from the bone models, material properties, and muscle forces [5, 13]. The visualization module displays the raw data and the solid models, animates the dynamics, displays the stresses, and supports user manipulation and quantitative querying.

The computational requirements of the simulation module govern the design of the other modules. We organize the requirements around the tasks of reconstruction, kinematics, rigid body dynamics, stress analysis, and flexible body dynamics.

*West Lafayette, Indiana
47907-1398 USA

{bajaj,fxb,eps,drs,klin}@cs.purdue.edu
http://ww.cs.purdue.edu/research/shastra

2 Model Reconstruction

We have developed novel techniques to build models of skeletal joints that are suitable for simulation. We can reconstruct polyhedral models from slices extracted directly from the CT data, or we can reconstruct (selectively) smooth models, represented as collections of polynomial surface patches, from unorganized large sets of points sampled on the model's surface.

In this application, we obtain the points by first extracting an isosurface corresponding to the interface between bone and surrounding soft tissue. We then segment the isosurface into its connected components, corresponding to the various skeletal parts. The segmentation process currently requires human intervention to solve ambiguous cases. We are developing new tools to deal with this problem in a more automated way. Finally, we use α-solids [2] to "clean up" the data set from unwanted artifacts. The result is a triangulated two-manifols that can be used as a starting point for a smooth model reconstruction, as explained below.

In the rest of this section, we provide an overview of the algorithms used for the reconstruction tak.

2.1 Reconstruction of Polyhedral Models from Slices

In computed tomography, magnetic resonance imaging and ultrasound imaging, reconstruction of the 3D object from the 2D scalar-valued slices obtained by the imaging system is difficult because of the large spacings between the 2D slices. The aliasing that results from this undersampling in the direction orthogonal to the slices leads to two problems known as the correspondence problem and the tiling problem. A third problem, known as the branching problem, arises because of the structure of the objects being imaged in these applications. Existing reconstruction algorithms typically address only one or two of these problems.

With the technique described in [4], we approach all three of these problems simultaneously. This is accomplished by imposing a set of three constraints on the reconstructed surface and then deriving precise correspondence and tiling rules from these constraints. The constraints ensure that the regions tiled by these rules obey physical constructs and have a natural appearance. Regions which cannot be tiled by these rules without breaking one or more constraints are tiled with their medial axis (edge Voronoi diagram).

Our implementation of the above approach generates triangles of 3D isosurfaces from input which is either a set of contour data or a volume of image slices. Results obtained with synthetic and actual medical data are presented. There are still specific cases in which our new approach can generate distorted results, but these cases are much less likely to occur than those which cause distortions in other tiling approaches.

Figure 3(a) shows an example of reconstruction using this algorithm.

2.2 Reconstruction of Smooth Models form Points

We reconstruct smooth boundary models of bones from CT data. Although previous research demonstrates the feasibility of polyhedral shape reconstruction, these models are inappropriate for joint modeling. They require far more elements than do smooth models to achieve a given accuracy. Modern graphics hardware can display the smooth models faster and better than the polygonal models. Kinematic analyses of polygonal models are not feasible because the computation time is proportional to the product of the large number of polygons in the bone models (the patella, femur, and tibia in the knee). Incremental modification is difficult because changes in one vertex can effect the entire model. The derivative discontinuities increase model error, degrade dynamical simulation, stress analysis, and shading. Smooth models solve these problems, but complicate reconstruction. The best current algorithms can fail on seemingly straightforward problems.

We have developed two different techniques [2, 3] to reconstruct the shape of 3D objects from a dense, unorganized cloud of points. We build a boundary representation of the object, based on implicit polynomial surface patches of low degree, that has the desired geometric continuity and approximates the data within a user-specified parameter ε. Our first algorithm can reconstruct surfaces that are C^1-smooth everywhere. We have then enhanced the algorithm to make it capable of automatically detecting sharp features and recostructing them accurately.

2.2.1 Reconstruction Algorithm

Our first algorithm is based on the concept of signed distance. For a point q in $\mathbf{R}^3$, let $|\delta_D(q)|$ be the Euclidean distance between q and the closest point on D. An orientable, closed manifold in $\mathbf{R}^3$ partitions the space in two regions, an *interior* and an *exterior*. We define the signed distance function $\delta_D(q)$ as the mapping $\mathbf{R}^3 \rightarrow \mathbf{R}$ that associates to the point q the distance to the closest point on D, with a positive sign if q lies in the exterior of D, and a negative sign otherwise.

We build an initial piecewise-linear approximation of D, so that we can compute $\delta(q)$ for any given q. Obviously for $q \in P$ we know that $\delta(q) = 0$. We then proceed to compute, in an adaptive fashion, a piecewise polynomial approximation of δ. The zero-contour of this function will implicitly define the shape of the reconstructed manifold.

Both steps of the algorithm make use of an incremental 3D Delaunay triangulation construction. In the first phase, we compute the Delaunay triangulation of P. We then extract a triangulated two-manifold T (the boundary of what we call an α-solid, see [2]) from it using a technique inspired by α-shapes [9] and sculpturing [6].

For a point $q \in \mathbf{R}^3$ we can now compute its distance from T and its sign, so that an approximate function δ is defined. The data structures associated with the triangulation and its dual allow us to compute the value of $\delta(q)$ efficiently.

We then start the incremental construction of another 3D Delaunay triangulation, which we will use as the support mesh of our piecewise polynomial reconstruction. We begin with a single tetrahedron containing the region of interest, and compute the 20 weights of the associated polynomial patch based on least-squares approximation of samples of δ in its interior. We then introduce a new site, update the triangulation, and compute the weights of the new tetrahedra, continuing in this way until the approximation error becomes smaller than the given ε. At each step, the new site introduced is the circumcenter of the tetrahedron with the largest associated approximation error. We also make sure that the single-sheetedness constraints are satisfied for each patch.

We build in this way a hierarchy of volume-splines, whose zero-contour converges to the shape of D. Figure 1 shows an example of reconstruction using this algorithm.

2.2.2 Enhancements for Objects with Sharp Features

Some objects have sharp geometric features such as corners and creases. Cubic A-patches can be joined to achieve surface C^1 continuity, but the continuity constraints can be relaxed along selected edges or faces to permit the representation of sharp corners, rectilinear or curvilinear edges, and flat faces [2]. These features must however be

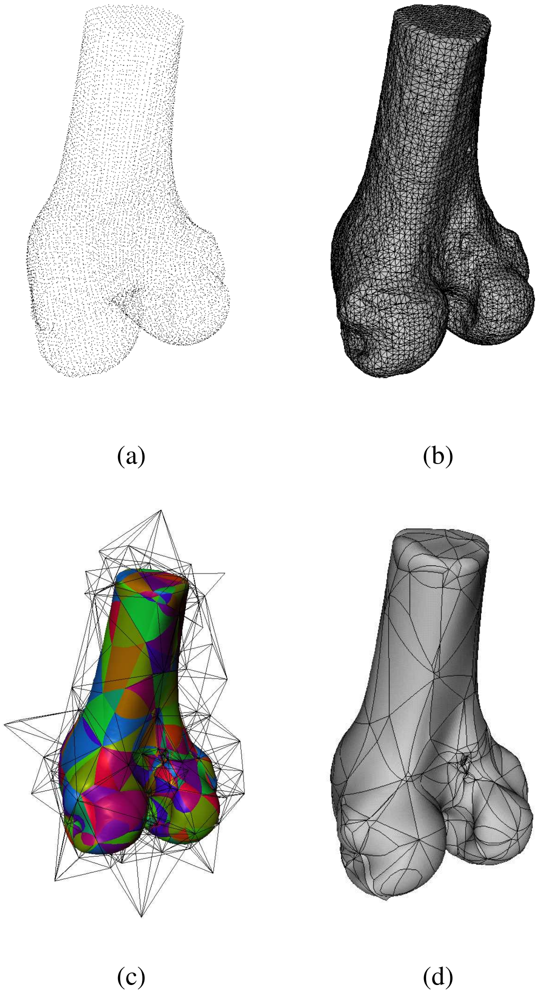


FIGURE 1: Reconstruction algorithm. (a) Points extracted from CT volume via isosurfacing. (b) α-solid of points in (a). (c) Supporting tetrahedral mesh and surface A-patches approximating the surface of the α-solid. (d) Reconstructed surface model of the distal end of the femur. Curves highlight patch boundaries.

completely contained in, or coincident with, a collection of edges or faces of the supporting mesh.

We start by extracting, as in our previous algorithm, a two-manifold from the Delaunay triangulation of P. This fine mesh can be used to estimate dihedral angles between adjacent triangles and detect and tag sharp features. A mesh decimation algorithm is then applied to reduce the number of triangles, especially in areas of small curvature. The reduced mesh is used as the base for a *simplicial-hull* construction. The result is a mesh of tetrahedra surrounding the triangulated two-manifold, that will act as support for the surface patches.

At this points, weights for all patches are computed from estimated surface normals and least-squares approximation of data points, starting with faces containing sharp features. We obtain in this way a mesh of surface patches that are C^1-continuous, except along selected features, and approximate the points. Finally, we run an energy minimization algorithm to optimize distribution of curvature and improve data fitting.

Figures 2 and 3(b) show examples of reconstruction using this algorithm.

3 Kinematics

Kinematic analysis computes the motion constraints imposed by bone contacts. We are developing a fast algo-

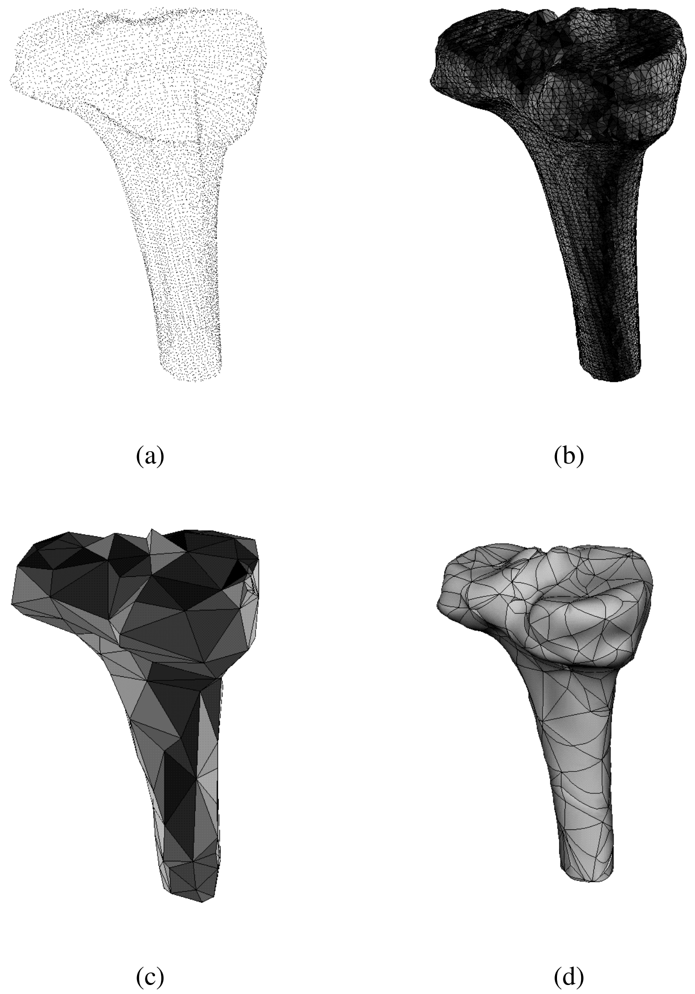
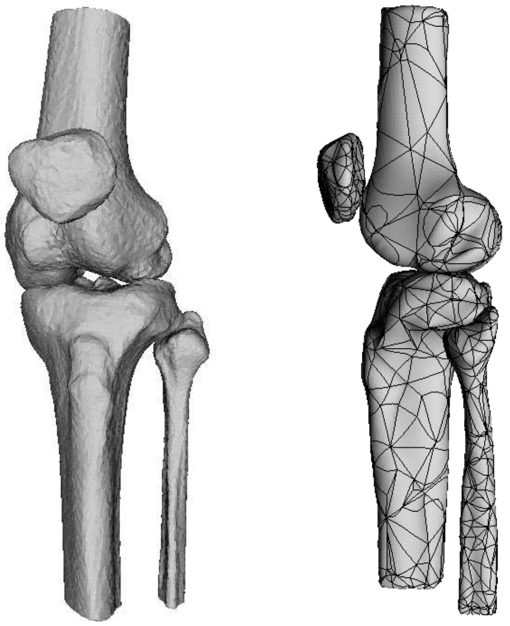


FIGURE 2: Enhanced reconstruction algorithm. (a) Points extracted from CT volume via isosurfacing. (b) α-solid of points in (a). (c) Reduced mesh. (d) Reconstructed model of the tibia. Curves highlight patch boundaries.

rithm for joints with approximately planar motion. Efficient algorithms are available for ball/socket, hinge, and other standard (lower-pair) joints. These cases cover most human and prosthetic joints [10, 8]. The modeling requirements are to reconstruct the bone contours from the clinical data, to separate the bones in a joint, and to construct planar slices parallel to the motion plane. We discuss only the knee, since modeling the wrist and other joints poses analogous problems.

Figure 3 shows a solid model of a knee reconstructed from CT images of the visible male. Kinematic analysis of the knee is difficult. The contacts between the tibia, femur, and patella impose complicated nonlinear constraints on their positions and orientations. The constraints must be computed numerically because analytical solution methods are unavailable or impractical. The constraints induced by the three pairs of bones interact. Computing the interaction requires the simultaneous solution of the contact constraints. The constraints change during knee motion as different portions of the bones come in contact and as the axes of motion shift. In normal walking, the femur rotates in the sagittal plane and the posterior portion of the femoral condyle touches the tibial plateau. In rapid walking or running, the femur rotates around an orthogonal axis at the beginning and end of each step (automatic rotation), which shifts the contact to a portion of the condyle with a different shape. In hyperextension, the proximal portion of the condyle comes in contact with the tibia. Many other interactions are possible.

The knee moves primarily in the saggital plane, so we



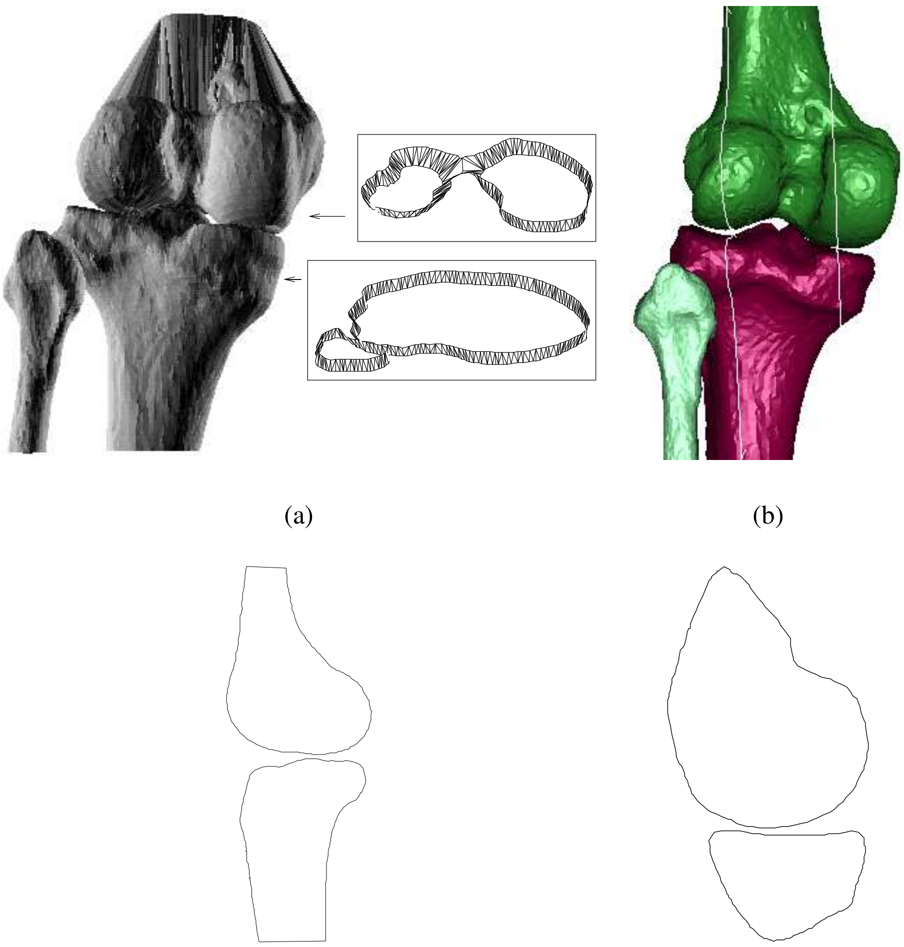
(a) (b)

FIGURE 3: Reconstructed knee model of the Visible Male, showing femur, tibia, fibula and patella. (a) Polyhedral model approximating the extracted surface from the CT volume. (b) Smooth surface patches model of (a), with curves highlighting patch boundaries.

approximate the joint with a slice through the left lobe of the femur, a second slice through the right lobe, and a third slice through the middle. Figure 4 shows two of these slices extracted from the reconstructed polyhedral model of the knee. We approximate the full kinematics with the planar kinematics of several such slices. We derive the kinematics directly from the bone geometry with our configuration space computation algorithm [15, 16, 19, 18]. This approach complements physical measurement of the kinematics, which we can perform with the Purdue knee simulator [24]. We will validate our computations on cadaveric simulations then use the simulator to obtain custom kinematic models for natural or prosthetic knees.

The configuration space of the knee is a geometric space (manifold) whose points represent the configurations (positions and orientations) of the bones. The dimension of the configuration space equals the number of degrees of freedom of the parts. It is at most six times the number of bones, since each part may translate along and rotate around three independent axes, but is generally much lower due to part contacts. For example, the pin joint in a robot knee allows a single, rotational motion of the femur relative to the tibia, whereas human knees allow six independent motions. Configuration space partitions into free space where the bones do not touch and into blocked space where they overlap. The common boundary, called contact space, contains the configurations where the objects touch. The contact space geometry encodes the motion constraints that we need for dynamical simulation, including contact points, contact normals, and contact changes.

Figure 5(b) shows the configuration space of the planar knee slice shown in Figure 5(a). We hold the tibia fixed and allow the femur to move freely with position (x, y) and orientation θ. The contact space is grey. The blocked space is the interior of the contact space and the free space is the exterior. Figure 5(c) shows the configuration space slice $\theta = 0.5$, which shows the contacts when the bones translate in that fixed orientation. The shaded region is the blocked space where the femur and the tibia overlap. The white region is the free space where they do not touch. The filled circle in free space marks the displayed configuration of $(5, -3)$. The solid boundary between free and blocked space is the contact space where the bones touch. It consists of curves that represent contacts between the features of the tibia and the femur.



(a) (b)

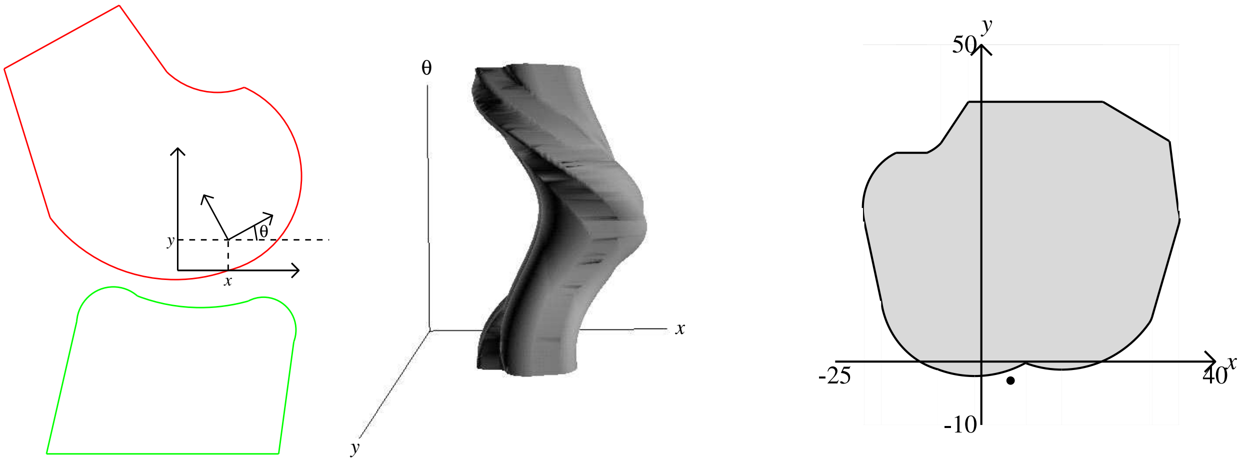
(c) (d)

FIGURE 4: (a) Reconstructed polyhedral model of the knee showing detail of two slices. (b) Sections extracted for the 2D 3-dof kinematics simulation. (c) and (d) show planar views of the two extracted sections from (b) (left and right sections respectively).

4 Dynamics

Rigid body dynamics computes the large-scale motion of joints by integrating Newton's equations for each bone subject to the contact constraints. We have developed a novel simulator that computes the contact constraints directly from the bone models. This computation is the bottleneck in dynamical simulation. At each time step, the simulator computes the touching parts, the contact points, and the contact geometry. It uses this information to detect collisions, to compute the resulting velocity changes, and to compute contact forces. Current contact analysis algorithms have to examine all pairs of part features in the worst case, making the running time per step quadratic in the geometric complexity of the parts. Although clever methods exist for speeding up the computation, they do not apply to joints because of the complex geometry and the closeness of the bones [14, 23].

We have developed a simulation algorithm that replaces collision detection with configuration space computation [20]. The configuration spaces of the pairs of parts in the mechanism are complete, concise contact models. Our algorithm precomputes these contact models from the bone models before the simulation and stores them with the bone models. It handles contacts between parts with curved boundaries without approximation. The worst-case running time of the precomputation is quadratic in the geometric complexity of the parts, as is a single collision detection. At each simulation step, the algorithm obtains the necessary contact data by querying the pair-



(a) (b) (c)

FIGURE 5: (a) Planar knee model. (b) Configuration space for planar knee model. (c) Configuration space slice $\theta = 0.5$.

wise configuration spaces. Each query takes linear time in the number of parts and is independent of their geometric complexity.

The simulator requires the material properties and external loads on the bones. We currently use values from the biomechanical literature [12, 7, 17].

We plan to generate customized dynamical models that incorporate contact forces due to bone interactions, external forces due to body weight, muscle forces, and elastic forces due to ligaments, tendons, and other flexible membranes. The contact forces will come from the kinematics model. The user will provide the body weight or the program will compute it from clinical data. The user will provide the muscle forces or the program will compute them from high-level specifications of patient activity, such as "running" or "jumping rope". We will model flexible membranes as rigid links connected by compression springs, torsion springs, and dampers. The program will estimate the model parameters, such as link lengths and spring coefficients, from the clinical data. The user will simulate standard motion regimens by varying the parameters. For example, tibial rotation is inhibited during flexion by making the torsion spring very stiff.

5 Joint Stress Analysis

Joint stress analysis meshes the bones, calculates the loading from the dynamics [13], and invokes a finite-element package (ABAQUS). The modeling requirement is to reconstruct smooth bone models from rough clinical data. The visualization requirement is to display the computed

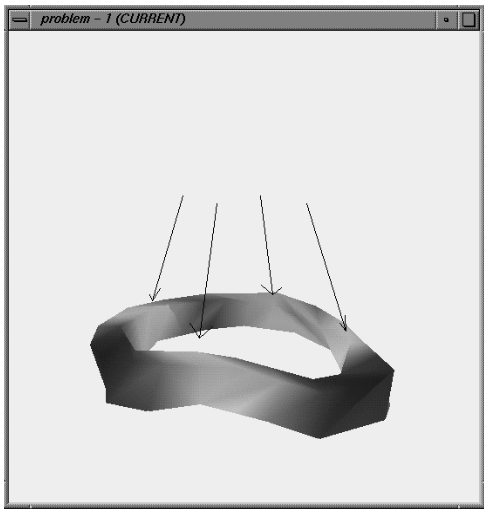


FIGURE 6: Stress loading and visualization for a slice of the femur model.

tensor field of stresses superimposed on the joint model. Flexible body dynamics computes large-scale motions of parameterized deformable objects, such as generalized spring-damper systems, by integrating Lagrangian equations. The modeling requirement is to construct parametric models of muscles, ligaments, and cartilage (the deformable parts of joints) from clinical data. Rigid and flexible dynamics share the modeling requirement of computing the material properties of the bones from the clinical data and share the visualization requirements of displaying the motion of the bones and the contact forces, which specify vector fields in physical space.

In our Shastra environment [1], a set of distributed tools are used in a collaborative manner to complete the various stages of analysis. Modelling from images is performed by a medical imaging toolkit, with results communicated directly to a finite element preprocessor and visualization tool. Here the model is meshed to user specifications. Material properties for the mesh are computed directly from the CT data, and boundary conditions are interactively specified by one or more collaborating users. Analysis is performed by one of several external solvers which are supported in the environment, and results are read back into the system for post-processing and visualization. An example stress loading of a slice of bone from a femur is given in Figure 5.

6 Acknowledgements

This research supported in part by NSF grant CCR 92-22467, AFOSR F49620-94-10080, ONR grant N00014-91-1-0370, NSF GER-9253915-02, CCR-9505745, and by the Purdue Center for Computational Image Analysis and Scientific Visualization.

References

[1] Anupam, V. and Bajaj, C. SHASTRA: Collaborative Multimedia Scientific Design. *IEEE Multimedia* **1** (1994) 39–49.

[2] Bajaj, C., Bernardini, F., Chen, J., et al. Automatic reconstruction of 3D CAD models. in: *Proc. of the International Conference on Theory and Practice of Geometric Modeling*. Springer-Verlag, 1996. Accepted for publication.

[3] Bajaj, C., Bernardini, F., and Xu, G. Automatic reconstruction of surfaces and scalar fields from 3D scans. in: *Computer Graphics Proceedings*, Annual Conference Series. Proceedings of SIGGRAPH 95, pages 109–118. ACM SIGGRAPH, 1995.

[4] Bajaj, C. L., Coyle, E. J., and Lin, K.-N. Arbitrary topology shape reconstruction from planar cross sections. *Graphical Models & Image Processing* (1996). To Appear.

[5] Blacharski, P. A., Somerset, J. H., and Murray, D. G. A three-dimensional study of the kinematics of the human knee. *Journal of Biomechanics* **8** (1975) 375–384.

[6] Boissonnat, J.-D. Geometric structures for three-dimensional shape representation. *ACM Trans. Graph.* **3** (1984) 266–286.

[7] Ciarelli, M. J., Goldstein, S. A., Dickie, D. L., et al. Experimental determination of the orthogonal mechanical properties, density, and distribution of human trabecular bone from the major metapyseal regions utilizing material testing and computed tomography. *Trans. of the Orthop. Res. Soc.* (1986) 42.

[8] Crouch, J. E. *Essential Human Anatomy*. (Lea & Fegiger, Philadelphia, PA, 1982).

[9] Edelsbrunner, H. and Mücke, E. P. Three-dimensional alpha shapes. *ACM Trans. Graph.* **13** (1994) 43–72.

[10] Freudenstein, F. and Woo, L. S. Kinematics of the human knee joint. *Bulletin of Mathematical Biophysics* **31** (1969) 215–232.

[11] Goldberg, K., Halperin, D., Latombe, J., et al. (Eds.). *The Algorithmic Foundations of Robotics*. (A. K. Peters, Boston, MA, 1995).

[12] Goldstein, S. A., Matthews, L. S., Braunstein, E. M., et al. The distribution of tibial trabecular bone properties as determined by computer assisted tomography. *Trans. of the Orthop. Res. Soc.* (1982) 302.

[13] Hayes, W. C. and Swenson, L. W. Finite element stress analysis of the human knee. *Orthopaedic Biomechanics: Procedures and Devices* (1981) 29–78.

[14] Hirokawa, S. Biomechanics of the knee joint. *Critical Reviews in Biomedical Engineering* **21** (1993) 79–135.

[15] Joskowicz, L. and Sacks, E. Computational kinematics. *Artificial Intelligence* **51** (1991) 381–416. reprinted in [11].

[16] Joskowicz, L. and Sacks, E. Configuration space computation for mechanism design. in: *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. IEEE Computer Society Press, 1994.

[17] Lang, S. M., Moyle, D. D., Berg, E. W., et al. Estimation of trabecular bone mechanical properties by cat. *Trans. of the Orthop. Res. Soc.* (1987) 51.

[18] Sacks, E. and Bajaj, C. Sliced configuration spaces for curved planar bodies. Technical Report 95-078, Purdue University, 1995.

[19] Sacks, E. and Joskowicz, L. Computational kinematic analysis of higher pairs with multiple contacts. *Journal of Mechanical Design* **117** (1995) 269–277.

[20] Sacks, E. and Joskowicz, L. Mechanical systems simulation using configuration spaces. Technical Report 96-046, Purdue University, 1996.

[21] Sheh, M., Butler, D. L., and Stouffer, D. C. Mechanical and structural properties of the human cruciate ligaments and patellar tendon. *Trans. of the Orthop. Res. Soc.* (1986) 236.

[22] Struhl, S., Goldstein, S. A., Dickie, D. L., et al. The distibution of mechanical properties of trabecular bone within vertebral bodies and itiac crests: Correlation with computer tomography density. *Trans. of the Orthop. Res. Soc.* (1987) 262.

[23] Whittle, M. W. Dynamic assesment of knee joint function. *Engineering in Medicine* **15** (1986) 71–75.

[24] Zachman, N., Hillberry, B., and Kettelkamp, D. Design of a load simulator for the dynamic evaluation of prosthetic knee joints. in: *ASME paper 78-DET-59, presented at ASME Design Engineering Conference*, Minneapolis, Minnesota, September 25 1978.