The HUMANOID Environment for Interactive Animation of Multiple Deformable Human Characters

R. Boulic[1], T. Capin[1], Z. Huang[1], P. Kalra[2],
B. Lintermann[3], N. Magnenat-Thalmann[2], L. Moccozet[2],
T. Molet[1], I. Pandzic[2], K. Saar[3], A. Schmitt[3], J. Shen[1] and D. Thalmann[1]

1) Computer Graphics Lab, Swiss Federal Institute of Technology
EPFL, DI-LIG, CH-1015 Lausanne, Switzerland

2) MIRALab-CUI, University of Geneva, 24 rue du Général Dufour
CH 1211 Geneva, Switzerland

3) Institut für Betriebs- und Dialogsysteme, University of Karlsruhe
Kaiserstraße 12, D-76128 Karlsruhe, Germany

Abstract
We describe the HUMANOID environment dedicated to human modeling and animation for general multimedia, VR, and CAD applications integrating virtual humans. We present the design of the system and the integration of the various features: generic modeling of a large class of entities with the BODY data structure, realistic skin deformation for body and hands, facial animation, collision detection, integrated motion control and parallelization of computation intensive tasks.

keywords: Articulated Figure Modeling, Animated Deformation, Collision Detection, Parallelization

1. Introduction
Simulating human motion and behavior by computer is an active and challenging area. Numerous papers have been proposed (see [1] and [2] for exhaustive lists). However, there is a lack for descriptions of integrated real-time human animation systems. The HUMANOID environment described in this paper is an environment dedicated to the development of multimedia, VR and CAD applications involving virtual humans. This environment is unique as it integrates highly heterogeneous components such as the environment model, the humanoid model and various motion generators. The HUMANOID environment supports the following facilities:

- real-time display for interactive manipulation on a standard graphic workstation
- a way of accelerating the calculations using a dedicated parallel machine or by distributing the calculation load on several workstations
- a flexible design and management of multiple humanoid entities
- skin deformation of a human body, including the hands and the face
- a multi-layer facial animation module
- collision detection and correction between multiple humanoid entities
- several motion generators and their blending: keyframing, inverse kinematics, dynamics, walking and grasping

We first present the general architecture of the HUMANOID environment to highlight the various results of integration. Then, in the next sections we examine the key features of the system, beginning with generic synthetic human modeling, the associated skin deformation, the collision detection, the motion control and ending with the parallelization issues.

2. Architecture

The general architecture of our environment is organized into four integration levels :

- The SCENE level : This lower level provides a flexible means for the modeling of an heterogeneous environment. A set of typed entities can be associated with the node-3D, i.e. the structuring entity for the construction of n-ary tree hierarchies. The hierarchy is dedicated to motion control [3][4] with possible redefinition of the traversal which makes more efficient comparatively to PHIGS or OPEN-INVENTOR. Moreover, an external developer is able to insert additional modules to cover specific modeling of the environment (e.g. the storage of local data for collision detection as developed in section 5).
- The BODY level : At this second level, we manage a specialized hierarchy with a fixed topology. This hierarchy is parameterized with the SKELETON entity which allows to model various populations of synthetic actors and also a large set of vertebrate animals. At the same level, various deformation generators are integrated for body, the hands and the face.
- The MOTION level : At this third level, we provide motion generators for 3D hierarchies as keyframing, inverse kinematics, dynamics, and for BODY entities as the walking motor, the grasping function and the facial animation control.
- The application level : At the higher level, we find the application layer with an integrated motion control application TRACK, a metaball-based sculptor for designing human bodies, a FFD-based sculpting program for creating human faces, and a high-level facial control system SMILE. In this paper, we emphasize the integration of all the previously mentioned levels. As the TRACK application can manage the motion control of multiple humanoid entities with deforming skin and collision detection, the parallelization of various tasks has been also investigated.

3. Human Modeling with the BODY entity

To achieve modularity and generality we have defined a fixed topology and two entities to encapsulate the geometrical, volumetric and inertial characteristics. The SKELETON entity contains the position and orientation of the joints relative to their parent joint in the articulated structure. The VOLUME entity approximates the body volume by a set of volumic primitives holding a fraction of the body mass.

Designers can create BODY characters at two different levels. First, the low level allows modification of every single parameter of the SKELETON (e.g. setting the length of the right upper arm). The complete BODY design is a very tedious work, but it allows the specification of a wide range of vertebrate animals. These low level operations generates a *template* BODY structure. The second approach for the generation of various individuals is to use five geometric scaling operators and a mass scaling function. A scaled BODY (therefore called an *instance*) is always derived by absolute scaling of a reference template BODY thus preventing the accumulation of rounding errors. (See figure 1 for an overview of these operators). Moreover, a template can be shared by several instances allowing the creation of vertebrate family and multi-actor scene while saving memory space.

The scaling method operates by multiplying the translation vector between successive joints by a scaling factor. For the general scaling the scaling factor is the ratio of the desired instance height over the template height. Other scaling operators act on a subset of the SKELETON and VOLUME parameters, and only in some specific direction. These scaling operators permit easy generation of characters with proportions matching general gender differences. Considering that some of the segment lengths show low correlation coefficients with stature [5], the uniform scaling of the skeleton lengths by the ratio of body heights may produce an instance out of a given population sample of interest. However this feature is only required for some ergonomic design decision program and can be achieved by using a proper template model or an anthropometric database [6].

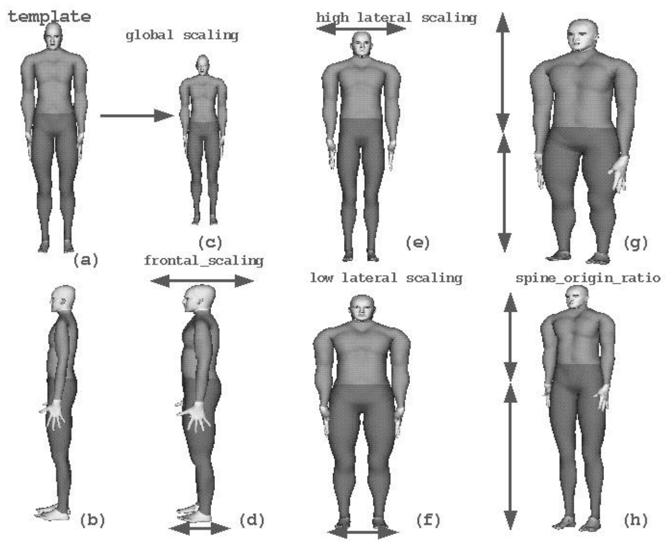
Special care is taken to protect the data integrity while performing the scaling transformations. For example, the length parameter of the volumic primitives are systematically derived from the SKELETON structure. All these operations modify also the mass distribution as now described. The volumic primitives may be scaled in the three directions of their local coordinate system or only along some directions. Assuming a constant density, the mass associated with a primitive is scaled by the ratio of the new volume over the old volume. Finally, the global mass scaling function acts on all the volumic primitives. The associated mass is updated according to the fraction of the total mass it holds.

4. Realistic Body Surface Deformation

Skin deformation problems have been handled as three independent tasks each dedicated to a specific type of deformation: body skin, hands and face. The integration of the resulting deformed surfaces is achieved at the lower level of a common polygonal surface representation thus allowing a coherent management of the display and the collision detection. Each deformation module, respectively called SKIN, HAND and FACE, query the BODY module for the current state of the SKELETON. Conversely, the BODY module centralizes the activation of the deformation in one generic function. The SKIN module is responsible of the connection operation of the hands and the face surfaces (figure 5a) .

4.1 Body Skin Deformation

Realistic modeling and deformation of human body shapes is an important but difficult problem. [7]. We propose here an effective multi-layered approach for deforming human bodies.



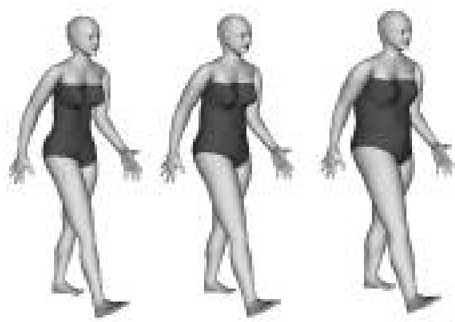


Figure 1. Automatic skin scaling.

Figure 2. Body morphing

Ellipsoidal metaballs are used to simulate the gross behavior of bone, muscle, and fat tissue; they are attached to the skeleton and arranged in an anatomically-based approximation. The skin construction is made in a three step process. First, the implicit surface resulting from the combination of the metaballs influence is automatically sampled along cross-sections with a ray casting method [8]. Second, the sampled points constitute control points of a B-spline patch for each body part (limbs, trunk, pelvis, neck). Third, a polygonal surface representation is constructed by tessellating those B-spline patches for seamless joining different skin pieces together and final rendering.

Supporting the five normalized scaling is straight forward in our model, because we simply need to scale the axis lengths of 3 principal directions of each metaball accordingly. Figure 1 illustrates the effect of the scaling operators on the skin envelope. Moreover, the interpolation of the metaballs parameters between two key designs can generate interesting 3D morphing of human shape. In figure 2, the left image shows a thin person. By adjusting several metaballs, we model the thick person of the right image. An intermediate envelope is shown with the middle image.

4.2 Hand Deformations

The motivation for a different deformation technique for the hands comes from its very specific behavior. Deformations of hands can not be simulated using the skin deformation module because the interior side of the hand is crisscrossed by lines and wrinkles creating discontinuities on the surface when deforming. Metaballs and splines are not accurate enough to reproduce these features.

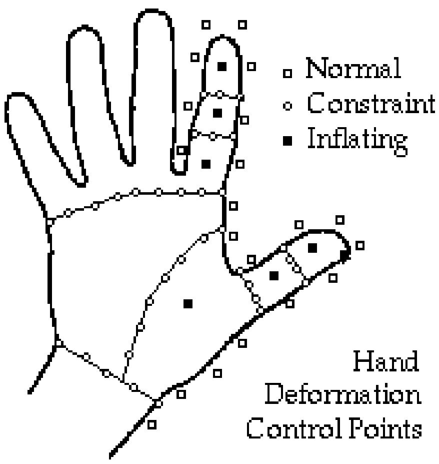


Figure 3. The 3 types of control points

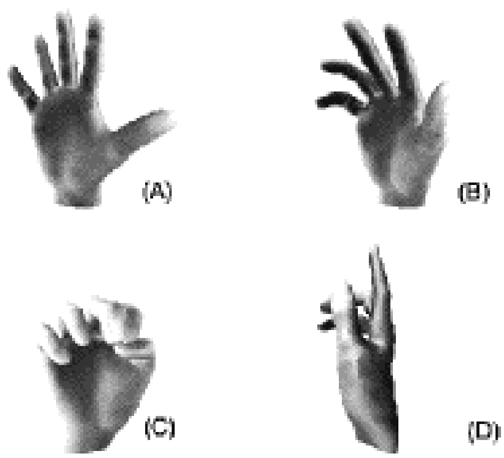


Figure 4. Resulting hand deformation

An alternate approach consists in enhancing the free-form deformations (FFD's) techniques [9] by using results of data interpolation [10] to remove existing limitations of the current FFD's models [11] (especially for the animation of articulated characters). In [10], Farin extents the natural neighbors interpolant based on the natural neighbors coordinates, by using the natural neighbors as the support for a multivariate Bézier simplex, in which any point can be expressed with a similar relation as in FFD's. Farin defines a new type of surfaces defined with this extended interpolant called Dirichlet surfaces. Combining FFD's and Dirichlet surfaces leads to an enhanced model of FFD's : Dirichlet FFD's or

DFFD's. One major advantage of this technique is that it removes any constraint on the position and topology of control points. Three types of control points are defined (figure 3) :

- *normal* : they are defined around or inside the surface at a variable distance from it.
- *constraint* : they lie on the surface and allow its direct manipulation. They are generally assigned to a point of the surface, so that any displacement of the control point is completely passed to the surface's point.
- *inflating* : a special type of normal control points but with varying weight, so that they can attract the surface in their neighborhood and simulate inflation of the surface.

From that general free-form deformation model, we derive a specialized data structure for multilayered deformations of articulated object, where the set of control points is used to simulate the muscle layer [12,13]. According to hand topography, the main lines and wrinkles are associated with joints of the skeleton. The idea consists in defining a data structure called wrinkle on the hand surface and associate it with each of the skeleton joints. A wrinkle is defined by the following: (1) a set of constraint points approximating the real wrinkle. (2) A set of normal points surrounding the surface of the hand which is influenced when the joint moves and (3) an inflating point simulating the flesh inflation according to the joint angle value. Joint angle variations are used to rotate the control points of the wrinkle around the joint's rotation axis. The weight of the inflating points is a linear function of the joint angle value. DFFD's are then applied to the hand's surface according to the current deformed skeleton configuration. Figure 3 shows how wrinkle's are designed over the hand's skin

The use of constraint points allows flexible control of the shape around the joint and constrains it to maintain a constant shape during the animation, so there is no need to scale empirically control points [12] at the joints to avoid unwanted deformations like pinching. Combining control points with constraint points not only allows the control of the deformation extension on the surface, but also allows mixing smooth deformations with discontinuities on the surface. Finally, modeling skin deformation around joints with multi-segments is almost impossible with former FFD's techniques, because it would require a combination of many control boxes in such a way that it satisfies the continuity constraints between the boxes. With DFFD's, a unique set of control points can surround the joints and their segments, and the continuity of the deformation is naturally kept inside it (figure 4).

4.3 Facial Deformations

Facial movements can be understood as externalization or manifestation of verbal or non-verbal communication agents on a face. These agents activate certain channels of the face associatively. Each activated channel in turn triggers the relevant muscles. Activation of muscles eventually deforms the face. Our module for facial animation resolves the difficulty of manually manipulating the facial model by offering a multi-level structure to the system, where each level is independently controllable [14]. The different levels encompass information from the various levels of abstraction from sentences and emotions to the facial geometry and image visualization. For simulation of muscle actions we use Rational Free Form Deformations (RFFD) as they are: simple and easy to perform, intuitive and rapid to use [15]. RFFD includes an extra term for weights for the control lattice of FFD. This provides additional control for deformations. The muscle design uses region based approach which means that the region of interest is defined where a muscle comprising RFFDs can be associated. The deformations which are obtained by actuating muscles to stretch, squash, expand and compress the inside facial geometry, are simulated by displacing the control points of the lattice and/or by changing the weights of the control points.

Human face may render complex movements of its different parts during a facial expression. There are more than thirty muscles responsible for the facial movements. These muscles are associated with the source of movements such as speech or emotion. Using muscles for specifying an animation is difficult for the user. The multi level structure enables users to specify animation in terms of high level global actions like emotion and sentences for speech. These actions are decomposed into discrete actions like expressions and phonemes for words. The discrete actions contain the basic motion parameter referred to as Minimum Perceptible Action (MPA) [16]. These MPAs are in fact aggregation of muscle actions. In order to control the duration and instance of each global action we provide a synchronization mechanism. The high level specification of animation is independent of low level facial model.

5. Collision Handling

If the simulation of the real world is to appear realistic, our system has to enforce the property of solid objects not to penetrate one another [17]. The visual inspection and interpretation of the 3D scene on the 2D screen is a tedious task. To release the user from the burden of visual verification and to reduce the time necessary to complete the design of animation sequences the HUMANOID environment offers the following facilities:

- *collision determination:* indication if there are contact or penetration situations
- *collision analysis:* report of details as the worst colliding vertex and the direction for correction
- *collision response:* visual and acoustical feedback to the user, correction of body postures

5.1 Specific Requirements for Human Animation

According to the SCENE level philosophy presented in section 2 we encapsulate the collision handling functionality into an entity called CNODE. Instances of this structure are attached to all objects in the scene and especially to the various parts of the human body. They are derived from BODY instances and provide transparent and fast access to collision analysis results.

The static collision detector operates frame by frame in a quantum model of the world: at specific key times all object have discrete positions and orientations coming from the motion generators, and shapes coming from the deformation module. The lowest level tests for mutual penetrations operates on the polygonal surface representation. A sufficient condition for a surface intersection is the intersection of any edges of one object with faces of the other and vice-versa [18]. For the detection of inclusions we employ additional vertex-polyhedra-tests. Naive approaches to this problem can be slow: Given n objects in the scene with an average of m faces each (typical values range from hundreds to several thousand triangles), a brute-force algorithm requires $O(n^2m^2)$ intersection tests. To alleviate the quadratic nature of the problem and reduce the number of low-level intersection tests the HUMANOID environment provides several optimization techniques [19-23]. The basic ideas are the usage of a cascade of tests, each computing faster than the next, the exploitation of spatial and temporal coherence, and the possibility for the user to advise the system where to expend its resources: on accuracy or on efficiency.

5.2 Optimizing the Collision detection

Fast checks for intersections of bounding volumes as necessary conditions for object intersections are used to eliminate most of the object-pairs from detailed considerations. Since complex objects like the limbs of the human body are generally better approximated by complex volumes than by simple ones, we attach polymorphous boundaries to each object: a sphere, a non-isothetic bounding parallelepiped and a bounding cylinder, all represented analytically with a small amount of scalar parameters which are determined by the system. Instead of imposing limits on the deformability of the skin and enlarging the volumes to contain all possible shapes, each CNODEs determines if the underlying surface has been deformed during the last time step and updates the size of the volume according to the surface geometry.

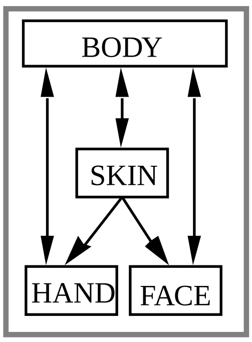
The articulated structure of the human body suggests a hierarchical arrangement of CNODEs: specific instances of nodes group together several chained objects to multi-level hierarchies. These are accompanied by bounding volumes large enough to comprise all possible configurations of the articulated structure. Optimal dynamic hierarchies lead to a logarithmic reduction of inter-object tests. Application dependent parameters (like the distance to the camera) can control the maximum depth of collision checking. An additional level of volumes is introduced by *fractions,* i.e. subdivisions of CNODEs attached to objects composed of a large number of triangles. Fractions contain connected regions of the underlying triangle mesh and use either shared or non-shared memory to access the surface. Intersections on the triangle mesh of two objects are only performed if all higher level checks returned positive results. We exploit our knowledge about the topology and geometry of the surface and employ a sweep-line approach for the triangle-edge intersection.

In certain situations the user may desire to focus the attention to specific locations on the body surface. Arbitrary vertices can be selected by interactively picking them and assigning them the role of sensors. Only these vertices and incident edges are sensitive to penetrations into other objects.

By giving the user the freedom to exploit a priori knowledge and classify object-pairs by setting up *collision qualifying matrices*, we are able to prune many more unnecessary computations completely [24]. The HUMANOID environment allows to suppress tests for collisions between objects with no relative movement, for collisions between objects which are kept apart by known constraints (e.g. intersection of a human's head and feet is a very exceptional event), and for collisions between adjacent limbs of articulated structures.

6. Integrated Motion Control with the TRACK Application

In this section we describe the integration of motion control motors and the tools to meet the objective of interactive performance while deforming and animating multiple human characters.



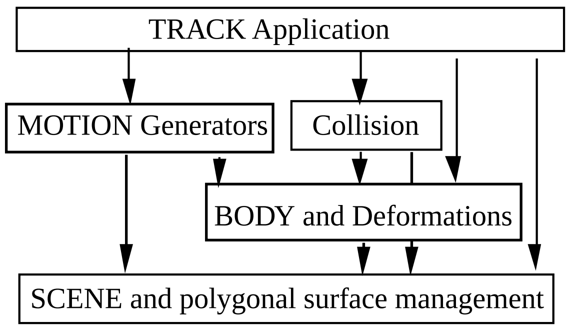


Figure 5a. BODY and deformation modules **Figure 5b.** TRACK and lower level libraries

TRACK is an application providing:

· *integration of five motion generators for multiple actors* : keyframing, inverse kinematics, dynamics, walking (based on [25]) and grasping (see [26]). The low level keyframe representation is used to integrate the various motion generators by allowing the recording of their sampled output. Then, a motion blending function can be used to produce natural transition between any keyframed motion.

· *integration of deformation with animation* : the skin of human model can be deformed in the motion control process. It allows animator to see and control both motion and shape at the same time.

· *integration of collision detection*: self collision detection and multiple actor collision detection during motion control is another specific feature. It is computed on the polygonal surface representation of the body surface which contains thousands of triangles per actor. After a collision is detected, the motion can be corrected by applying inverse kinematics on the worst colliding vertex in order to push it along the direction of correction (resulting from the analyze of the collision detection module).

· *parallelization of skin deformation and dynamics* : we make use of parallel processing to increase the computational performance (see section 7 for more details).

Figure 5a details the relationships between the BODY module and the deformation modules while figure 5b shows the TRACK system organization and Figure 6 outlines TRACK control flow (motion generation, deformation, collision detection and response). Various examples are detailed in the next section. They successively present the collision detection, the collision correction and the grasping.

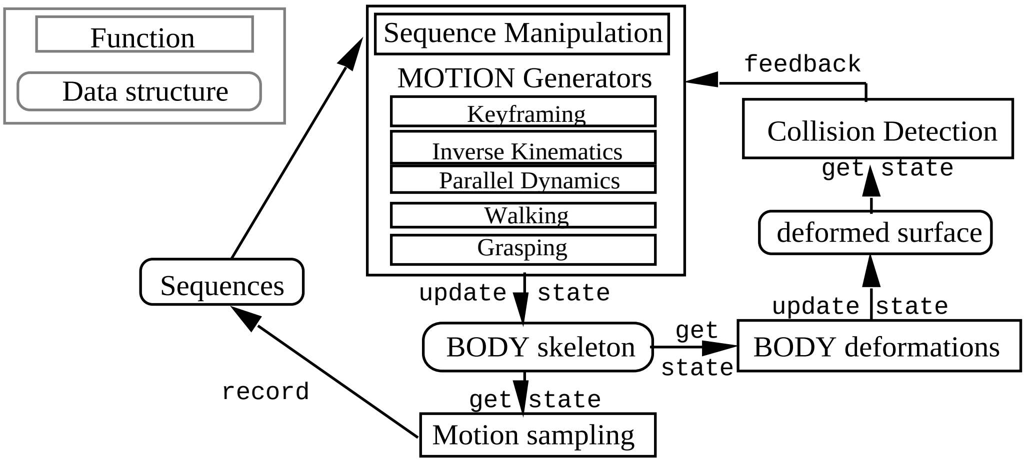
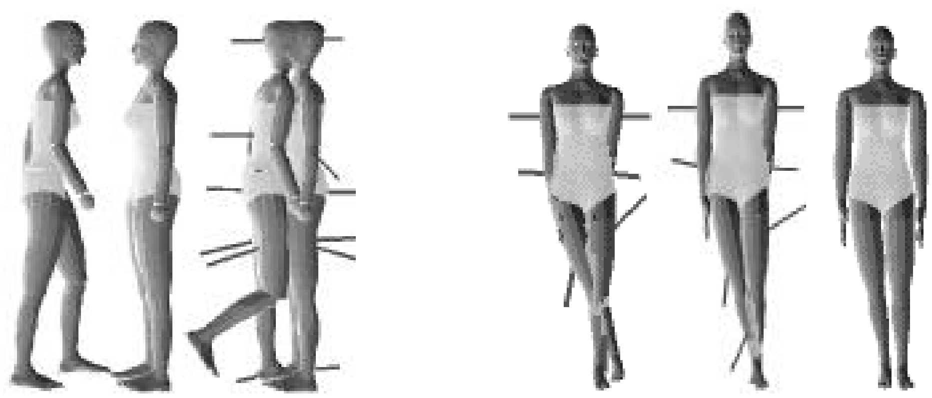


Figure 6. The system control flow of TRACK.



(a) (b)

Figure 7. Collision detection (the lines shown are response vectors). In context (a) a walking actor is colliding with a standing actor. In context (b) self collision are detected and corrected according to the direction of correction (evaluated by the collision analysis).

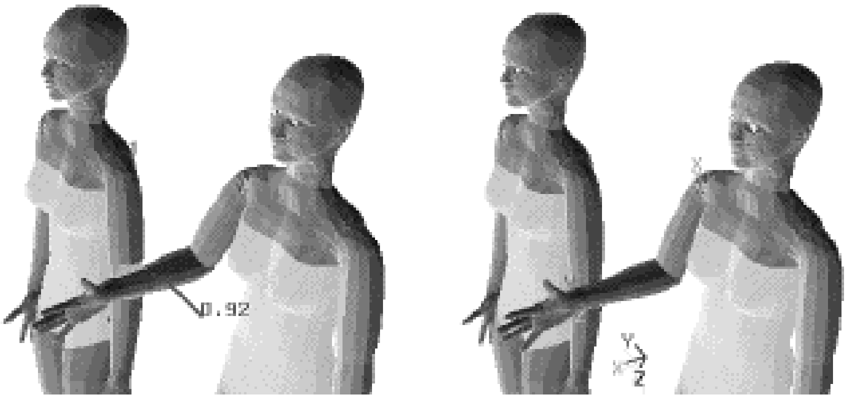


Figure 8. Multi-actor collision detection and correction with inverse kinematics, the guiding frame displayed in the right figure is derived from the response vector.

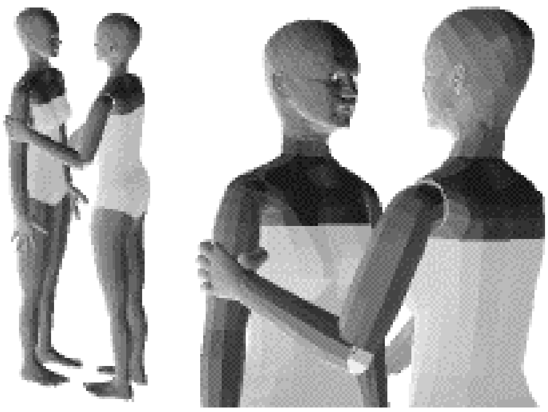


Figure 9. Grasping Interaction between actors

Figure 10. BODY entities displayed at various levels (here deformed skin and solid volume)

7. Parallelization Issues

In our environment, parallelization may be used in order to speed up the animation tasks. This is especially important when the performance of the graphics workstation becomes unsatisfactory as the computational requirements for the system increase, especially with complex models and environments.

The architecture of the system is shown in Figure 11. The parallel machine is seen by the front-end SGI workstation as a black-box accelerator for computation-intensive animation tasks. The following tasks are performed during each frame: 1) Motion generation, 2) Body deformation, 3) Collision detection, 4) Facial animation. The system should be flexible enough to permit the user to run above tasks on a parallel machine optionally (for example, the system should also work efficiently if the deformation module is used, without the dynamics and collision detection modules). Therefore, we have designed the task distribution such that the processors are not dedicated to a special task, but rather each task is computed by all the processors. The second requirement is to handle multiple human models. Thus, for different virtual actors we have different groups of processors defined in the beginning of the animation. For integration of the modules, we try to achieve maximum locality for different tasks for the same body part in order to decrease the volume of communication between the processors for different modules.

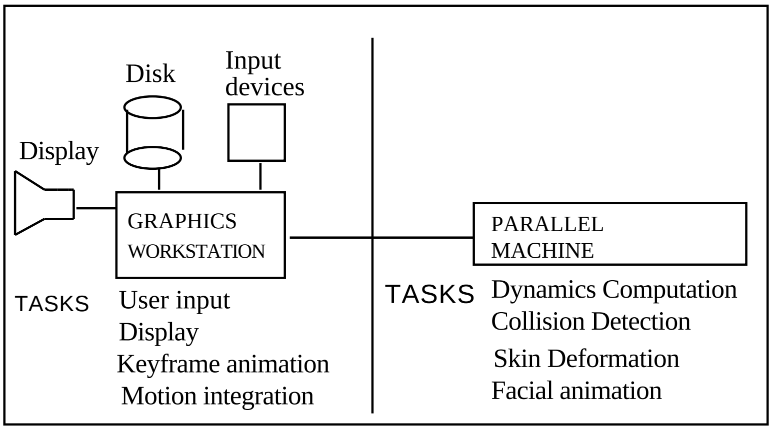


Figure 11. Architecture of the system.

For the forward dynamics module we exploit the Armstrong-Green algorithm [27], because it is an efficient algorithm for articulated bodies with rotational joints. The parallelization of this algorithm is based on the fact that different branches of the tree-shaped limb hierarchy of the figure can be processed in parallel. Additionally, different actors can be processed in parallel by different processor subgroups.

For the deformations module on the parallel computer, we selected as an atomic task, the generation and intersection of the rays with the list of metaballs corresponding to a body part. The computation of deformations in some body parts depends on the deformations in some other parts. For example, for computing deformations for the left shoulder, the deformations for left arm and the upper torso should have been computed. For this reason, the generation of the Bspline-Net for every body part of the virtual human (e.g. left arm) is performed concurrently by the processors, rather than simultaneously deforming several body parts of a virtual human.

In a real-time facial animation application, an average of 50% of the processor time is spent calculating the Free Form Deformations that deform appropriate regions of the face to give the desired expression. The rest of the time is mostly spent on rendering. By delegating the deformation calculation to the parallel computer and performing it concurrently with the rendering, the whole application can be run faster. The total time is determined by the time of the slower process and the communication overhead. The rendering time depends on the performance of the workstation and can't be improved by the programmer. So, to achieve the maximal total speed it is ideal to reduce the time spent in deformation calculations under the rendering time.

Currently, we use SGI Indigo-2 Extreme workstations for display, and the T9000-Transputer based Telmat parallel computer for parallel computation. The parallel processes share the processors of the parallel machine, in order to have maximum performance.

8. Conclusion

We have described the HUMANOID environment dedicated to human modeling and animation for general purpose applications integrating synthetic humans from animation design to ergonomics. Various features have been integrated in this system : generic modeling a large class of entities with the BODY data structure, realistic skin deformation, collision detection, integrated motion control and parallelization of computer demanding tasks.

Current researches include a more general motion correction according to the collision state and an alternate parallelizing approach. The new parallel approach is based on coarse-grain processing of different actors using a workstation network, and helps to use the system efficiently in general computing environments. We are porting the programs to SGI, DEC Alpha and SUN workstations and use the PVM environment for distributing the tasks.

9. Acknowledgments

This research is part of the ESPRIT project HUMANOID (P 6079) involving our industrial partners RTL, TELMAT and Silicon Graphics. The authors are also indebted to Serge Rezzonico for the upgrade of the LIG 5D Toolkit and the Interface Builder. The research was partly supported by the Swiss National Foundation for Scientific Research, Silicon Graphics and l'Office Fédéral de l'Education et de la Science and Silicon Graphics.

10. References

1. Badler N.I., Philips C.B., Webber B.L. “Simulating Humans”, Oxford University Press, 1993
2. Magnenat-Thalmann N., Thalmann D."Six Hundred Indexed References on Computer Animation", Journal of Visualization and Computer Animation, Vol.3, No.3, 1992, pp.147-174.
3. Boulic R., Renault O., "3D hierarchies for animation", in " New Trends in Animation and Visualization", Wiley Professional Computing, pp 59-77, John Wiley & son, Chichester 1991.
4. Boulic R., Huang Z., Magnenat-Thalmann N., Thalmann D.(1994) "Goal Oriented Design and Correction of Articulated Figure Motion with the TRACK System", Journal *of Computer and Graphics,* Vol 18 n°4, pp. 443-452, Pergamon Press, October 1994
5. Kroemer et Al. "Advances in Human Factors Ergonomics for Engineering Physiology", *Correlations*, pp. 17-18, *Anthropometric Inputs*, pp. 109-115, Editor : Gavriel Salvendy, 1986.
6. Grosso M., Quach R., Badler N. "Anthropometry for Computer Animated Human Figures", Proceedings of Computer Animation'89, pp. 83-96, Springer Verlag, 1989.
7. Magnenat-Thalmann N., Thalmann D. "Human Shape Design and Deformations" SIGGRAPH 91 Course Notes n°20 "Advanced Techniques in Human Modeling, Animation, and Rendering"
8. Shen J., Thalmann D., "Interactive Shape Design Using Metaballs and Splines", Implicit Surface95, Eurographics Workshop on Implicit Surfaces, Grenoble, France, Apirl, 1995.
9. Sederberg T. W., Parry S. R. Free-Form Deformation of Solid Geometric Models. Proceedings of Siggraph'86, Computer Graphics, 20, 4 pp 151-160, August 1986.
10. Farin G., Surfaces Over Dirichlet Tesselations, Computer Aided Design, 7, pp 281-292, 1990.
11. Bechmann D., Space Deformation Models Survey, Computer and Graphics, Vol. 18, No. 4, pp. 571-586.
12. Chadwick J. E., Hauman D. and Parent R. E. Layered Construction for Deformable Animated Characters. Computer Graphics, 23(3), pp 243-252, July 1989.
13. Lamousin H. J., Waggenspack Warren N., Nurbs-based Free-Form Deformations, IEEE Computer Graphics and Applications, 14, 16, pp 59-65, November 1994.
14. Kalra P. (1993) "An Interactive Multimodal Facial Animation System", Ph.D Thesis, Swiss Federal Institute of Technology, Lausanne, 1993.

15. Kalra P., Mangili A., Magnenat-Thalmann N., Thalmann D. (1992) "Simulation of Muscle Actions using Rational Free Form Deformations", Proc Eurographics '92, Computer Graphics Forum, Vol. 2, No. 3, pp. 59-69.
16. Kalra P., Mangili A., Magnenat-Thalmann N., Thalmann D. (1991) "SMILE: A Multilayered Facial Animation System", Proc IFIP WG 5.10, Tokyo, Japan, pp. 189- 198.
17. Baraff D., "Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies", *Computer Graphics*, Vol 23(3), July 1989, pp. 223-232
18. Boyse J., "Interference Detection among Solids and Surfaces", *Communications of the ACM*, Vol. 22(1), 1979, pp. 3-9
19. Canny J., "Collision Detection for Moving Polyhedra", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 8(2), 1986, pp. 200-209
20. Moore M., Wilhelms J., "Collision Detection and Response for Computer Animation", *Computer Graphics*, Vol. 22(4), August 1988, pp. 289-298
21. Gilbert E., Foo C., "Computing the Distance Between General Convex Objects in Three-Dimensional Space", *Proceedings IEEE International Conference on Robotics and Automation*, 1990, pp. 53-61
22. Lin M., Canny J., "Efficient Collision Detection for Animation", *Proceedings of the Third Eurographics Workshop on Animation and Simulation*, 1992
23. Pentland A., "Computational Complexity Versus Simulated Environment", *Computer Graphics*, Vol. 24(2), 1990, pp. 185-192
24. García-Alonso A., Serrano N., Flaquer J., "Solving the Collision Detection Problem", *IEEE Computer Graphics and Applications*, May 1994, pp. 36-43
25. Boulic R., Magnenat-Thalmann N. M.,Thalmann D. "A Global Human Walking Model with Real Time Kinematic Personification", *The Visual Computer*, Vol.6(6),1990
26. Mas-Sanso. R. and Thalmann D. (1994) A Hand Control and Automatic Grasping System for Synthetic Actors, Proceedings of Eurographic'94, pp.167-178.
27. Armstrong W.W., Green M.W. , Lake R., "Near-Real-Time Control of Human Figure Models", IEEE Computer Graphics and Applications, Vol 7, No. 6 1987.