

HUMAN BODY DEFORMATIONS USING JOINT-DEPENDENT LOCAL OPERATORS AND FINITE ELEMENT THEORY

Nadia Magnenat-Thalmann
Centre Universitaire d'Informatique
University of Geneva, Switzerland

Daniel Thalmann
Computer Graphics Lab
Swiss Federal Institute of Technology, Switzerland

Abstract

This paper discusses the problem of improving the realism of motion not from the joint point-of-view as for robots, but in relation to the deformations of human bodies during animation. Two methods for improving these deformations are described. The Joint-dependent Local Deformation (JLD) approach is convenient when there is no contact between the human being and the environment. A finite element method is used to model the deformations of human flesh due to flexion of members and/or contact with objects.

1. Introduction

One of the main challenges for the next few years is the creation and realistic animation of three-dimensional scenes involving human beings conscious of their environment. This problem should be solved using an interdisciplinary approach and an integration of methods from animation, mechanics, robotics, physiology, psychology, and artificial intelligence. The following objectives should be achieved:

- . automatic production of computer-generated human beings with natural behavior
- . reduction of the complexity of motion description
- . improvement of the complexity and the realism of motion; realism of motion needs to be improved not only from the joint point-of-view as for robots, but also in relation to the deformations of bodies during animation.

In this paper, we emphasize the third objective and propose two types of solutions to this problem:

- 1) A geometric approach: the mapping of surface using Joint-dependent Local Deformation (JLD) operators. This approach is convenient when there is no contact between the human being and the environment.
- 2) A physics-based approach using a finite element method which is used to model the deformations of human flesh due to flexion of members and/or contact with objects.

Modeling for synthetic actors frequently uses skeletons made up of segments linked at joints. This is suitable for parametric key-frame animation, kinematic algorithmic animation or dynamic based techniques (Magnenat-Thalmann and Thalmann 1988). The skeleton is generally surrounded by surfaces or elementary volumes (Badler and Smoliar 1979; Magnenat-Thalmann and Thalmann 1985) whose sole purpose is to give a realistic appearance to body. The model developed by Komatsu (1988) uses biquartic Bezier surfaces, and control points are assigned to the links. Catmull (1972) used polygons, and Badler and Morris (1982) used the combination of elementary spheres and B-splines to model the human fingers. Chadwick et al. (1988, 1989) propose an

approach which combines recent research advances in robotics, physically-based modeling and geometric modeling. The control points of geometric modeling deformations are constrained by an underlying articulated robotics skeleton. These deformations are tailored by the animator and act as a muscle layer to provide automatic squash and stretch behavior of the surface geometry.

2. Environment-independent deformations and JLD operators

2.1 The concept of JLD operator

For the deformation of the bodies, the mapping of surfaces onto the skeleton may be based on the concept of Joint-dependent Local Deformation (JLD) operators (Magnenat-Thalmann and Thalmann 1987), which are specific local deformation operators depending on the nature of the joints. These JLD operators control the evolution of surfaces and may be considered as operators on these surfaces. Each JLD operator will be applicable to some uniquely defined part of the surface which may be called the domain of the operator. The value of the operator itself will be determined as a function of the angular values of the specific set of joints defining the operator. Fig.1 and 2 show deformations using JLD operators.

When the animator specifies the animation sequence, he/she defines the motion using a skeleton, which is a wire-frame character only composed of articulated line segments. In order to animate full 3D characters, the animator has also to position a skeleton according to the body of the synthetic actor to be animated. This operation must be very accurate and it takes time. However it is a useful process, because animation is completely computed from the skeleton.

Fig.1. The effect of JLD operators

Fig.2. The effect of JLD operators

When the skeleton has been correctly positioned, the software will transform the character according to the angles required by the animation without any animator intervention.

Unfortunately, the procedure of positioning the skeleton is probably the longest stage apart from the digitizing of the shapes. This procedure is very important, because all the mapping of the surface shapes are based on the skeleton position relatively to the surface shapes. If a skeleton point is bad positioned, the joint will probably cause abnormal surface deformations in the animation.

We shall use for illustrating this procedure an actor skeleton of 87 points. Each of these points belonging to the skeleton should be positioned relatively to the flesh (considered as the actor surface) by the animator. But the problem arises: where to position this points?

The answer to this question is rather simple for points used as joints: points should be positioned at the center of the joint. For all points representing the extremity of a limb, points should be at the center of the extremity of the limb. Unfortunately, this two rules are insufficient to

position all skeleton points. There are extra-rules to follow. Fig. 3 shows where to position these points.

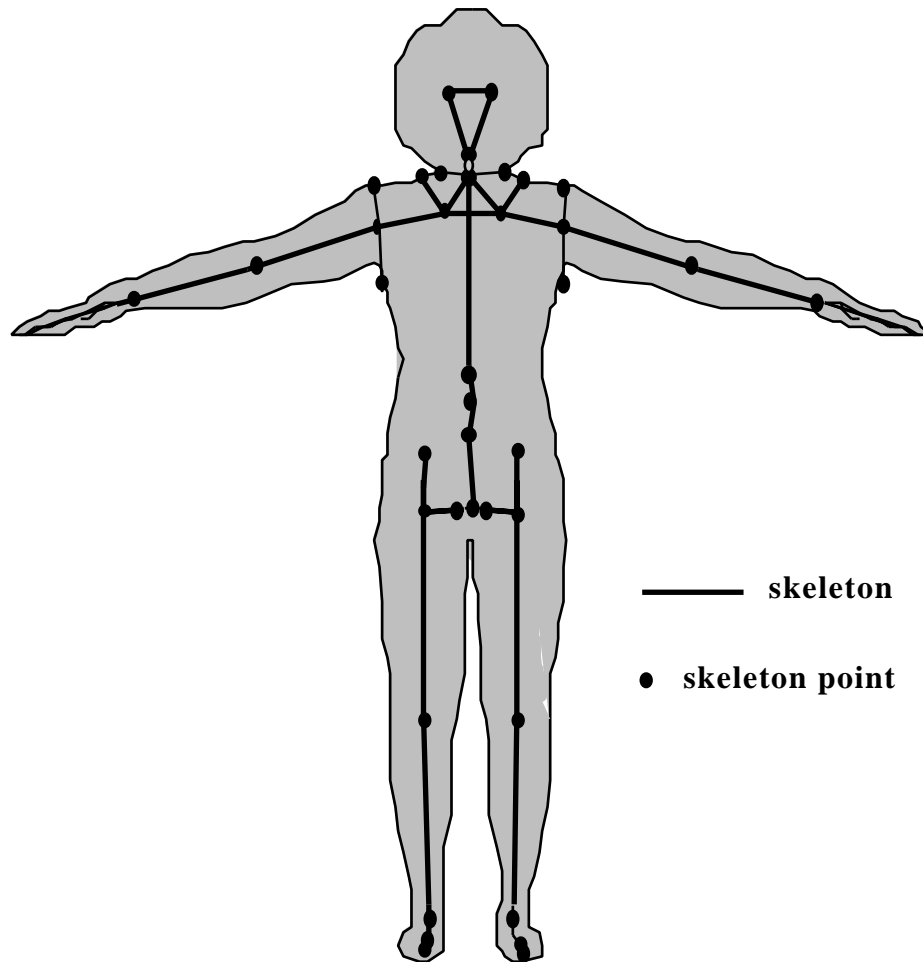


Fig.3. Points for body positioning

2.2 JLD operators for hand covering

The case of the hand is especially complex (Magnenat-Thalmann et al. 1988), as deformations are very important when the fingers are bent, and the shape of the palm is very flexible. Links of fingers are independent and the JLD operators are calculated using a unique link-dependent reference system. For the palm, JLD operators use reference systems of several links to calculate surface mapping. In order to make the fingers realistic, two effects are simulated: joint roundings and muscle inflation. The hand mapping calculations are based on normals to each proximal joint. Several parameters are used in these mapping calculations and may be modified by the animator in order to improve the realism of muscles and joints. These parameters include:

- the flexion axis at each joint
- a parameter to control the inflation amplitude of a joint during the flexion
- a parameter to define the portion of link to round during a joint flexion
- a parameter to control the inflation amplitude of muscles inside the hand during a flexion
- a parameter to define the location of the point where the inflation of the internal muscles is maximum during a flexion

Mapping algorithm

The initial and final normals are first determined for both joints of the link. Then, a modified normal is calculated as the average of the initial and final joint normal. This modified normal is then used as y-axis of the coordinate basis and it will allow the simulation of the external rounding of a joint during a flexion. For palm links, normal and modified normal calculations are also required for the neighbor links. Then, a loop is performed on all vertices associated with the link to be covered, and for each vertex, the process is:

Look for the 3D coordinates of the vertex in the digitized character

Calculate the following information to localize the vertex relative to the link:

Determine a projection of the vertex on the link

Calculate the ratio $R = \frac{\text{distance between the projection and the proximal joint}}{\text{distance between the projection and the distal joint}}$

{The proximal joint is the nearest joint to the wrist, the distal joint is the other joint}

Calculate the "vertex thickness", which is the distance between the vertex and its projection on the link.

A projection is also determined on the link in its final position by calculating $R(PF.D - PF.P) + PF.P$ where R is the above ratio, PF.P is the final position of the proximal joint and PF.D the final position of the distal joint. If the link to be covered is a link of the palm, same projection calculations are performed, but relatively to the neighbor link, in order to obtain an initial projection and a final projection on this link. These projections are used to simulate a virtual link linking the vertex projection on the link to be covered to the vertex projection on the neighbor link. The projection of the final position is also used as reference point or position relative to the link, in order to allow the transformation from the initial position to the final position. This virtual link allows the calculation of a scale factor:

$$F = \frac{\text{length of the virtual link at the initial position}}{\text{length of the virtual link at the final position}}$$

If the neighbor link in the final position is further from the link to be covered than in the initial position, the scale factor F will be greater than 1 otherwise it is in the range [0,1]. In this latter case, the distance between both links has decreased relatively to the initial position and an inflation should be generated.

Case of fingers

Links of fingers are independent and the JLD operators are calculated using a unique link-dependent reference system. In the case of external vertices (upper side of the hand), the link is divided into three areas, using the parameters given by the animator. A different coordinate basis is calculated for each area, because the simulation of joint rounding may be very different for each joint, due to the type of flexion. The normals at each joint are used as Y-axes of the coordinate bases (the positive direction is towards the upper part of the hand) ; the middle area is a buffer and uses an interpolated normal between normals of the other areas. In the case of internal vertices (lower side of the hand), the muscle is inflated along the whole link length.

Case of palm

For the palm, JLD operators use reference systems of several links to calculate surface mapping. In the case of external vertices, three areas are specified (see Fig.4). Areas along each of both links (areas 1, 4, 7 and areas 6, 3, 9) are determined as in the case of single links using animator-defined parameters and they are processed as above. However, these parameters are also used to determine areas between the links, as shown in Fig.3 (areas 2, 5, 8). For these inbetween areas, the calculation of the bases is different, because there are no roundings to be calculated. In the case of internal vertices, we have only three areas: one for the link to be covered, one for the neighbor link and one inbetween area. No internal inflation has to be calculated.

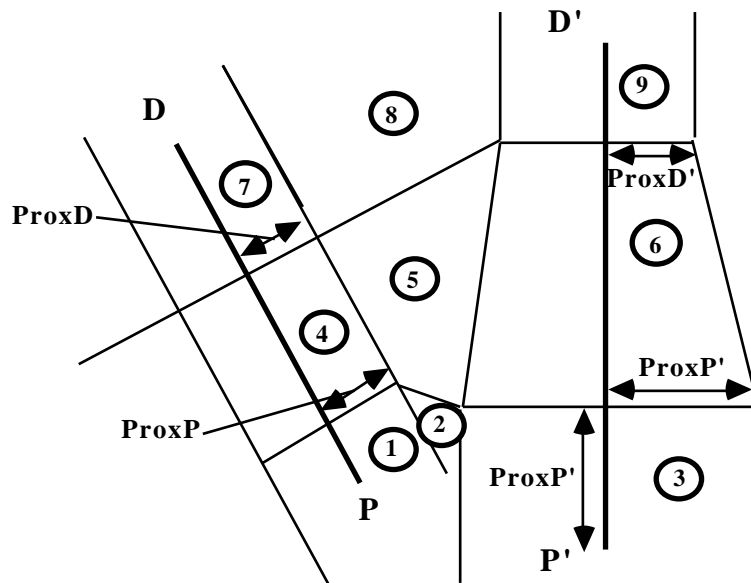


Fig.4. Areas for a palm mapping. P: proximal joint of the link to cover; D: distal joint of the link to cover; P': proximal joint of the neighbor link; D': distal joint of the neighbor link; ProxP: proximity parameter of P; ProxD: proximity parameter of D; ProxP': proximity parameter of P'; ProxD': proximity parameter of D'

Fig.5 shows a sequence of hand animation.

Fig.5. Object grasping

Coordinates may be then modified in order to simulate muscle inflations and the roundings of joints for improving the realism.

3. Environment-dependent deformations and finite element theory

3.1 A same method for modeling both environment objects and human bodies

Our purpose was to improve the control of synthetic human behavior in a task level animation system by providing information about the environment of a synthetic human comparatively to the sense of touch. We also state that the use of the same method for modeling both objects and human bodies improves the modeling of the contacts between them. We developed a finite element method (Gourret et al. 1989) to model the deformations of human flesh due to flexion of members and/or contact with objects. The method is able to deal with penetrating impacts and true contacts. For this reason, we prefer to consider true contact forces with possibilities of sliding and sticking rather than only repulsive forces.

The environment of characters is made up of rigid objects, key-frame deformable objects, mathematically deformable objects (Barr 1984), soft objects represented by scalar combinations of fields around key points (Blinn 1982; Wyvill et al. 1986), or physically deformable objects based on elasticity theory (Terzopoulos et al. 1987). With physical

models, the objects act as if they had a mind. They react to applied forces such as gravity, pressure and contact. Platt and Barr (1988) used finite element software and discuss constraint methods in terms of animator tools for physical model control. Moore and Wilhems (1988) treat collision response, developing two methods based on springs and analytical solutions. They state that spring solutions are applicable to the surface shapes of flexible bodies but do not explain how the shapes are obtained before initiation of contact calculations nor how the shapes are modified as a result of a contact.

Our main objective was to model the world in a grasping task context by using a finite element theory. The method allows simulation of both motion and shape of objects in accordance with physical laws, as well as the deformations of human flesh due to contact forces between flesh and objects. The following two arguments support use of the same method for modeling deformation of objects and human flesh.

First, we want to develop a method which will deal with penetrating impacts and true contacts. For this reason, we prefer to consider true contact forces with possibilities of sliding and sticking rather than only repulsive forces. Our approach based on volume properties of bodies permits calculation of the shape of world constituents before contact, and to treat their shape during contact. When a contact is initiated we use a global resolution procedure which considers bodies in contact as an unique body. Simulation of impact with penetration can be used to model the grasping of ductile objects or to model ballistic problems. It requires decomposition of objects into small geometrically simple objects.

Second, all the advantages of the physical modeling of objects can be transferred to human flesh. For example, we expect the hand grasping an object to lead to realistic flesh deformation as well as an exchange of information between the object and the hand which will not only be geometrical. When a deformable object is grasped, the contact forces on it and on the fingertips will lead to deformation of both the object and of the fingertips, giving rise to reactive forces which provide significant information about the object and more generally about the environment of the synthetic human body. It is important to note that even if the deformations of the object and of the fingers are not visible, or if the object is rigid, the exchange of information will exist because the fingers are always deformable. This exchange of information using active and reactive forces is significant for a good and realistic grip and can influence the behavior of the hand and of the arm skeleton. For grip, interacting information is as important as that provided by tactile sensors in a robot manipulator. This is a well known problem of robotics called "compliant motion control". It consists of taking into account external forces and commanding the joints and links of the fingers using inverse kinematic or dynamic controls. In the past, authors dealing with kinematic and dynamic animation models oriented towards automatic animation control (Armstrong and Green 1985; Badler 1986; Calvert et al. 1982; Wilhelms 1987), have often referred to works of roboticists (Hollerbach 1980; Lee et al. 1983; Paul 1981). In the same way, we believe that methods intensively used in CAD systems may improve the control of synthetic human animation.

3.2 Finite element approach in computer animation

Solid three-dimensional objects and human flesh are discretized using simple or complex volume elements, depending on the choice of the interpolation function. The finite element approach (Bathe 1982; Zienkiewicz 1977) is compatible with requirements of visual realism because a body surface corresponds to an element face that lies on the body boundary.

Once the various kinds of elements are defined, the modeled shape is obtained by composition. Each element is linked to other elements at nodal points.

In continuum mechanics (see Fig.6), the equilibrium of a body presenting a shape can be expressed by using the stationary principle of the total potential or the principle of virtual displacements:

$$wR = wB + wS + wF \quad (1)$$

where wB represents the virtual work due to the body forces such as gravity, centrifugal loading, inertia and damping, wS represents virtual work of distributed surface forces such as pressure, wF represents the virtual work of concentrated forces and wR represents the internal virtual work due to internal stresses.

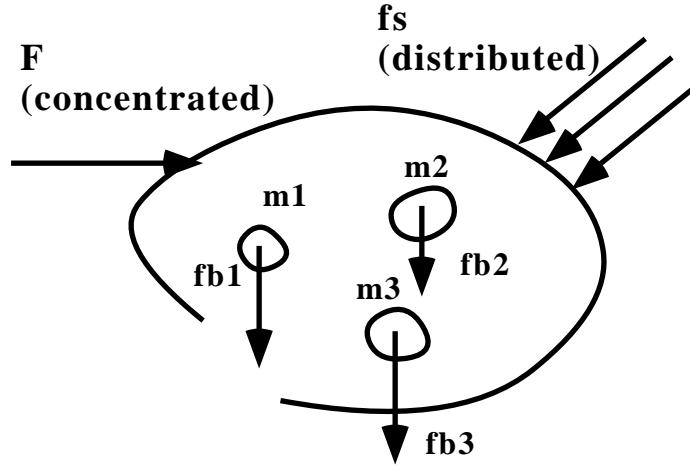


Fig.6. Continuum mechanics

In the finite element method, the equilibrium relation (1) is applied to each element e

$$wRe = wBe + wSe + wFi \quad (2)$$

and the whole body is obtained by composing all elements

$$\sum_{\text{element}=1}^{\text{NBEL}} wRe = \sum_{\text{element}=1}^{\text{NBEL}} wBe + \sum_{\text{element}=1}^{\text{NBEL}} wSe + \sum_{i=1}^{\text{NBR}} wFi \quad (3)$$

Our three-dimensional model uses elements with eight nodes and $\text{NBDOF} = 3$ degrees of freedom per node. These elements are easily modified to prismatic or tetrahedral elements to approximate most existing 3D shapes. The composition of NBEL elements with 8 points give NBP points and $\text{NB} = \text{NBP} * \text{NBDOF}$ equations. From the relation (3) we can write the following matrix equation between vectors of size $[\text{NB} * 1]$ as follows:

$$R + RI = RB + RS + RF \quad (4)$$

where RB is the composition of body forces, RS is the composition of surface forces and RF represents the concentrated forces acting on the nodes. $R + RI$ is the composition of internal forces due to internal stresses. These stresses are initial stresses which give the RI term, and reactions to deformations created by RB , RS and RF which give the R term.

We use the equilibrium relation (4) under the form (5)

$$K.U = R \quad (5)$$

where K is the $[\text{NB} * \text{NB}]$ stiffness matrix, a function of material and flesh constitution, R is the $[\text{NB} * 1]$ load vector including the effects of the body forces, surface tractions and initial stresses, and U is the $[\text{NB} * 1]$ displacement vector from the unloaded configuration.

Relation (5) is valid in static equilibrium and also in pseudo-static equilibrium at instant t_j . Instants t_j are considered as variables which represent different load intensities. In this paper, we do not deal with dynamics when loads are applied rapidly. In this case, true time inertia and damping, displacement velocity and acceleration must be added to (5).

Under this form, the body can be viewed as a huge three-dimensional spring of stiffness K and return force R . The equilibrium relation (3) is a function of volume properties because each component is obtained by the summation of integrations over the volume and the area of each element (see (Zienkiewicz 1977) for more details).

Since the process used consists of the composition of elements to create a global deformable object, we believe that this property and its inverse, i.e. the decomposition of a global deformable object into two or multiple sub-objects, should be used in computer animation.

The decomposition is very easy to implement because the constitutive properties of each element as well as inter-element forces are memorized and are taken into account during numerical calculations. It is possible for example to create a global object made of different sub-objects; each sub-object would have its own constitutive properties and be composed of one or more elements.

There are several ways to exploit the intrinsic properties of the finite element method:

- The decomposition approach can be exploited to model penetrating shocks between two or more deformable objects. Each object is subdivided into many deformable sub-objects which are able themselves to interact with each other because each inherits its own properties. The decomposition approach may also be used in contact problems when contact is released.
- The composition approach can be used for modeling contacts without penetration between two or more objects. In this case, objects can be considered as sub-objects evolve independently until contact is detected and a global object is composed following contact. In practice this means that relations $K_n \cdot U_n = R_n$ are resolved independently before contact and a unique relation $K \cdot U = R$ is resolved after contact of n bodies. This process works if we take into account the contact forces that prevent overlapping in equation (5). We use the composition approach for the grasping and pressing of a ball described in the following section. A survey of contact problems is given by Bohm (1987). Example of 3D treatments can be found in (Chaudary and Bathe 1986).

3.3 A case study: ball grasping and pressing

To show how the physical modeling of deformable objects can contribute to human animation, we present an example of a contact problem dealing with the grasping and pressing of a ball. Details may be found in (Gourret et al. 1989). Starting with the facet-based envelopes of ball and hand obtained from the image synthesis system SABRINA (Magnat-Thalmann and Thalmann 1987b), we mesh the volume of the objects to create full 3D bodies or shell bodies depending on the application. After calculations of the deformations using our method based on finite element theory, the facet-based deformed envelopes are extracted from the data base used in our calculations and restored to SABRINA for visualization. In this way, visual realism is always ensured by the image synthesis system.

Bones are connected to the segment and joint skeleton animated by the HUMAN FACTORY system. The hand envelope and segment-joint skeleton are sufficient for realistic hand animation without contact but are not able to reproduce skin deformations due to a contact. A mere bone segment is not sufficient to give realistic large deformation of skin

under contact forces because, as in human fingers, skin deformations are restricted because of bones. For this reason, we use realistic bones. This has an impact on visual realism and behavior of the hand during grasping, because bone parts are flush against the skin in some regions and are more distant in others. Moreover, in the future, more complex modeling will probably take into account nerves and muscles tied to bones (Thomson et al. 1988).

We use a composition approach based on the resolution of relation (5) including contact forces between ball and finger. This relation works perfectly in a grasping problem because loads are applied slowly.

However, contact modeling is not easy because the equilibrium equation (5) is obtained on the assumption that the boundary conditions remain unchanged during each time t_i . Two kinds of boundary conditions exist: geometric boundary conditions corresponding to prescribed displacements, and force boundary conditions corresponding to prescribed boundary tractions. We cannot control a single degree of freedom in both position and force; consequently, the unknown displacement will correspond to known prescribed force and conversely known prescribed displacement will correspond to unknown force.

In matrix notation, the problem can be stated in the following way: U_k are known prescribed displacements, U_u are unknown displacements, R_k are known prescribed forces and R_u are unknown forces. In this way, relationship (5) can be written

$$\begin{matrix} K_{11} & K_{12} & U_u & R_k \\ K_{21} & K_{22} & U_k & R_u \end{matrix} = \begin{matrix} R_k \\ R_u \end{matrix} \quad (6)$$

If NP degrees of freedom are displacement prescribed, $NBEQ=NB-NP$ equations are necessary to find the U_u unknown displacements. Matrix dimensions are $[NBEQ*NBEQ]$ for K_{11} , $[NBEQ*NP]$ for K_{12} , $[NP*NBEQ]$ for K_{21} and $[NP*NP]$ for K_{22} .

Equations for solving U_u are

$$K_{11}.U_u = R_k - K_{12}.U_k \quad (7)$$

Hence, in this solution for U_u , only the $[NBEQ*NBEQ]$ stiffness matrix K_{11} corresponding to the unknown degrees of freedom U_u need to be assembled. Once U_u is evaluated from (7) the nodal point forces corresponding to U_k can be obtained from (8)

$$R_u = K_{21}.U_u + K_{22}.U_k \quad (8)$$

Boundary conditions can change during grasping and pressing when prescribed forces or displacements are sufficient to strongly deform the ball and hand skin. This situation creates other contact points between ball and table, and between ball and fingers. Consequently the calculations are more complicated because the number of unknown displacements U_u , and reactive forces R_u , will vary depending upon the number of contact points which prescribe R_k and/or U_k .

Fig. 7. Grasping of a ball submitted to internal pressure

In a first step, sight allows us to evaluate certain dimensions, mass, roughness, elasticity etc. i.e. to imagine our position in relation to the ball. In the domain of animation, this information is contained in the ball data base (e.g. volume point coordinates, and physical characteristics such as constitutive law, mass density, and texture which can be related to roughness). The hand grasps or presses the ball applying a prescribed force, whose intensity is dictated by the knowledge acquired by the sense of sight. The prescribed contact force is created by muscular forces acting on bones and using flesh as an intermediary. Generally, the grip is as gentle as possible without letting go of the ball. This can be viewed as a "minimization of the power due to the muscles", as pointed out by Witkin and Kass (1988). A gentle grip not only prevents damage to a fragile object, but also results in a grip that is more stable (Cutkosky 1985; Slotine and Asada 1986). In a second step, the sense of touch allows an exchange of information between the ball and the fingers, implying contact forces, sliding contacts, deformations, and internal stresses in the fingers. In computer animation the first step is difficult to implement because the animator does not dispose of force transducers for forces applied directly to the bones. Consequently the first step based on given prescribed forces R_k on bones is not presently possible. For this reason our solution for the grasping problem is different from the robot or human solution. It is displacement-driven rather than force-driven.

In this way, the animator is not concerned with forces but with the hand key position required by the script. For ball grasping, shown in Fig.7, the ball is made up of a rubber envelope and is submitted to internal pressure. The animator imposes prescribed displacements U_k on the hand bones using a "classical" method (parametric, kinematic or dynamic) and places the ball between the fingers. During this process the animator can ignore the material of which the ball is built. It can be a very soft ball or a very stiff bowl. The animator positions the fingers (skin and eventually bones) inside the ball. The purpose of calculations is to decide if the chosen finger position is or is not a realistic one and its consequences on skin and ball shapes. This is the reaction of the ball on the fingers which will decide the validity of grasping. Since finger position is prescribed by the animator, the ball must be repelled to prevent overlaps, ignoring, as a first approximation, whether it is stiff or soft.

Fig.8 shows an iterative procedure for obtaining both the contact forces and the displacements under contact.

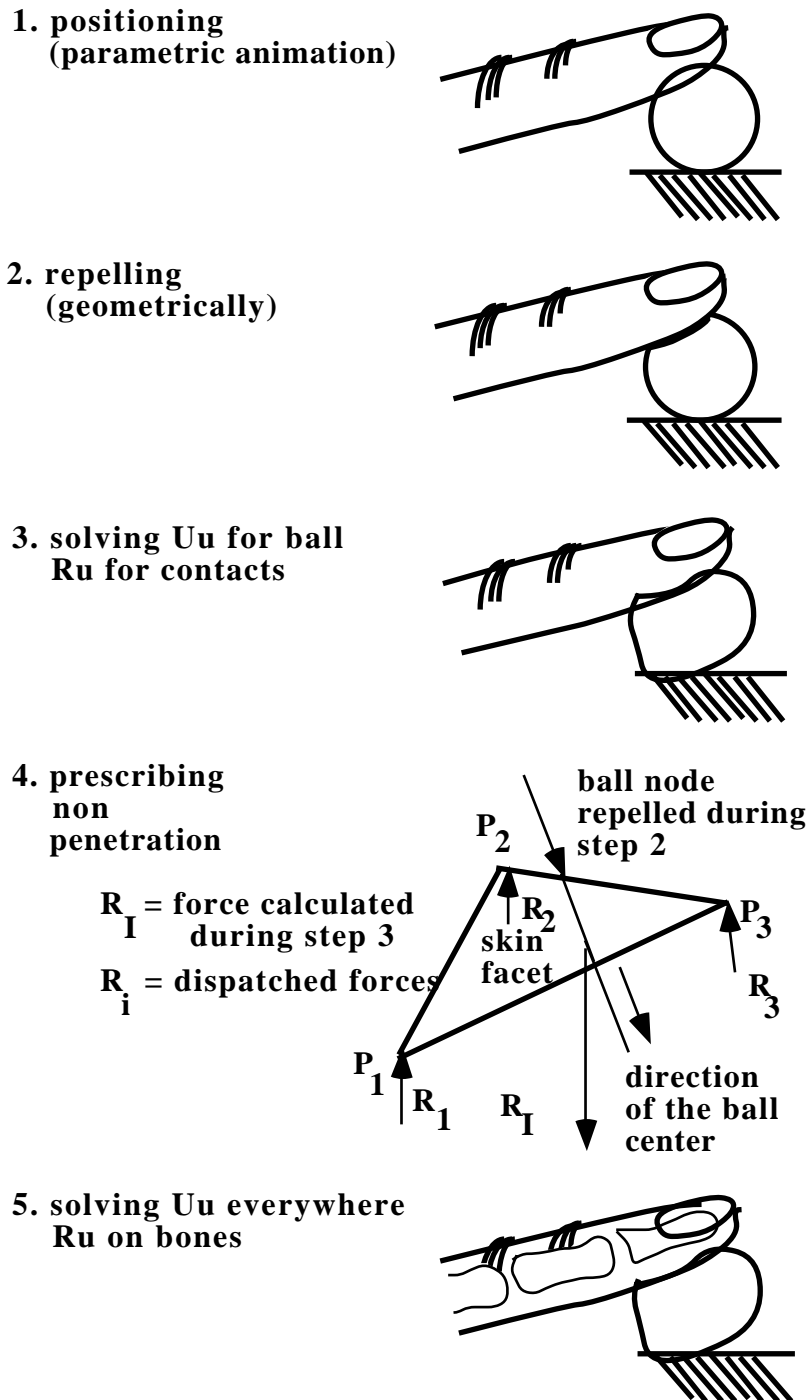


Fig.8. The algorithm

Relation (7) is solved using a direct solution (Gauss method). Convergence occurs when for all points the variation between two successive iterations is less than some fixed threshold. In this procedure, relations (7) and (8) represent the global system made up of ball and hand. The first step requires animator manipulation (e.g. parametric keyframe animation). In the second step all skin displacements, and those ball displacements resulting from overlap suppression are prescribed. Step 3 will give displacements U_u on ball and reacting forces on repelled ball nodes by solving Equations (7) and (8). Step 4 ensures that at equilibrium, contact forces between ball and skin will be equal and opposite to maintain compatible surface displacements. Because ball nodes are not repelled in coincidence with skin nodes, but on skin polygon surfaces, the reacting force calculated in step 4 is distributed among the three nodes constituting the skin facet, with weights depending on the position of

the ball node on the facet. In step 5, reacting forces are assumed to be known and all degrees of freedom of ball and skin are released. In other words, we rearrange matrix relation (6) because the number of equations is modified in comparison with step 3. The method can be interpreted as a Lagrangian multiplier method that forces the non-penetration condition between the ball and the hand with additional equations. Steps 2 to 5 are repeated until convergence is reached. Otherwise they are stopped when the evaluation of the reacting force on bones overruns a force threshold allowable by the human musculature.

In a parametric computer animation system, the reacting force on bones can be used to suggest solutions to the animator as in an expert system. In a system with inverse dynamics, the position of bones is modified automatically using calculated reacting force. Calculation of finger deformations are necessary even if fingers and palm deformations are not visible. An exchange of information will take place since the fingers are always deformed. It is finger flexibility and frictional resistance which permit human grasp of rigid objects. This is the reason why actual robot hands are made up of elastic extremities equipped with tactile sensors (Jacobsen et al. 1988; Pugh 1986). Our actual simulation is based on prescribing and releasing the displacement of contact points during each iteration. This allows us to release dynamically the parts of the two bodies. A more sophisticated model, now being developed, must include an evaluation of frictional resistance. For example a Coulomb friction law may be used to simulate the adhesion of papillary ridges. In this law, a coefficient of friction μ relates the normal force F_n to the tangential force F_t at contact points. Force $\mu \cdot F_n$ represents the frictional resistance during contact, and sliding contact is initiated when $F_t > \mu \cdot F_n$.

During the second step of grasping, if the initial prescribed force R_k has been poorly evaluated by the sense of sight, the ball and finger(s) will slide. This information can then be used to increase the prescribed force, or to modify the position of fingers on the ball. The evaluation of sliding and the increase in the prescribed force must be repeated until an equilibrium or an unstable condition is obtained.

Both the tactile sensor model and the command model must be included in a complete automatic motion control, because the stiffness of grip is a function of the stiffness of finger tissue and of the disposition of fingers around the ball. This compliant motion control scheme, which is made to sustain the environmental factors, might be made easier by the fact that kinematic and dynamic models dealing with articulated bodies can be looked upon as a displacement based finite element method applied on trusses and bars.

The global treatment of contact presented here can be applied to inter-deformation of fingers, deformation between fingers and palm, or, more generally, between two synthetic human parts following a compression or a stretching of skin. For this purpose, each part of the body must be considered as an entity able to interact with each other part. An entity cannot interpenetrate itself and some entities cannot reach all others because of joint angle limits imposed on the skeleton (Isaacs and Cohen 1987). For example, during finger flexing, we consider the third phalanx unable to interpenetrate the second and first phalanges. In the same way we also consider that a foot cannot reach the face unless we are simulating a chubby baby.

3.4 Animation control

Animation of a physically deformable object can be simply supported in the HUMAN FACTORY system by defining prescribed displacements called U_k in the preceding sections as new state variables. When the three degrees of freedom of some point are prescribed, state variables of VECTOR type are sufficient for defining the movement of prescribed points. When less than three degrees of freedom are displacement prescribed, a new type of state variable has been defined. With this approach deformable bodies are processed as actors.

Object points are not only submitted to translations and rotations, but automatically follow the prescribed degrees of freedom according to constitutive laws and other potential constraints such as body forces and surface forces. It should be noted that the use of state variables is not always required and can often be omitted. For example the problem of free fall of a deformable object is implicitly defined and does not require an extra physical law and prescribed degrees of freedom.

4. Conclusion

Two methods for improving deformations of human bodies have been described. The JLD approach is convenient when there is no contact between the human being and the environment. A finite element method in three dimensions has been introduced to model both object deformations and synthetic human flesh deformations. By doing this, the method gives information about the synthetic environment of the human. This environment is made up of physical objects, which should act as it would if they had a mind. They should react to applied forces such as gravity, pressure and contact. This approach provides a significant contribution to automatic motion control in 3D character animation. We believe that it can be used to improve the behavior of human grasp and human gait and more generally to improve synthetic human behavior in the synthetic environment.

5. Acknowledgments

The authors are grateful to Dr. J.P. Gourret from Ecole Nationale Supérieure de Physique de Marseille for his work in the finite-element project and R. Laperrière from University of Montreal for his cooperation in the JLD project. The research was supported by le Fonds National Suisse pour la Recherche Scientifique, the Natural Sciences and Engineering Council of Canada, the FCAR foundation and the Institut National de la Recherche en Informatique et Automatique (France).

5. References

- Armstrong WW, Green MW (1985) Dynamics for Animation of Characters with Deformable Surfaces in: N.Magnenat-Thalmann, D.Thalmann (Eds) Computer-generated Images, Springer, pp.209-229.
- Badler NI, Morris MA (1982) Modeling flexible articulated objects. Proc. Comp. Graphics '82, Online conf., pp 305-314.
- Badler NI, Smoliar SW (1979) Digital representation of human movement. Computing Surveys, Vol 11, No 1, pp 19-38
- Badler NI (1986) Design of a Human Movement Representation Incorporating Dynamics, in: Enderle G, Grave M, Lillehagen F, Advances in Computer Graphics I, Springer, Heidelberg, pp.499-512
- Barr AH (1984) Global and local deformations of Solid Primitives, Proc. SIGGRAPH '84, Computer Graphics, Vol.18, No3, pp.21-30
- Bathe KJ (1982) Finite element procedures in engineering analysis. Prentice Hall
- Blinn JF (1982) A generalization of algebraic surface drawing. ACM Trans. on graphics, Vol 1 No 3, pp 235-256
- Bohm J (1987) A comparison of different contact algorithms with applications. Comp.Struc., Vol 26 N 1-2, pp 207-221
- Calvert TW, Chapman J, and Patla A (1982) Aspects of the kinematic simulation of human movement. IEEE Computer Graphics and applications, november issue, pp 41-52
- Catmull E (1972) A System for Computer-generated Movies. Proc. ACM Annual Conference, Vol. 1, pp.422-431.

- Chadwick J, Haumann DR, Parent RE (1989) Layered Construction for Deformable Animated Characters, Proc. SIGGRAPH '89, Computer Graphics, Vol. 23, No3, pp.234-243
- Chadwick J, Parent R (1988) Critter Construction: Developing Characters for Computer Animation, Proc. Pixim 88, pp.283-305
- Chaudary AB and Bathe KJ (1986) A solution method for static and dynamic analysis of three dimensional contact problems with friction. Comp.Struc. Vol 24 N 6, pp 855-873
- Cutkosky MR (1985) Robotic grasping and fine manipulation. Kluwer Academic Publ.
- Gourret JP, Magnenat-Thalmann N and Thalmann D (1989) Simulation of Object and Human Skin Deformations in a Grasping Task, Proc. SIGGRAPH '89, Vol.23, No3, pp.21-30
- Hollerbach JM (1980) A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation. IEEE Trans. on systems, man and cyber., SMC-10 No 11, 1980, pp 730-736
- Isaacs PM and Cohen MF (1987) Controlling Dynamic simulation with kinematic constraints, behavior functions and inverse dynamics. Proc. SIGGRAPH' 87, pp 215-224
- Jacobsen SC, McCammon ID, Biggers KB, Phillips RP (1988) Design of tactile sensing systems for dextrous manipulators. IEEE Control Systems Magazine, Vol 8, No 1, pp 3-13
- Komatsu K. (1988) Human skin model capable of natural shape variation. The Visual Computer, No 3, pp 265-271
- Lee CSG, Gonzales RC and Fu KS (1983) Tutorial on robotics. IEEE Comp. Soc. Press
- Magnenat-Thalmann N and Thalmann D (1985) Computer animation: Theory and Practice. Springer, Tokyo
- Magnenat-Thalmann N and Thalmann D (1987) The direction of synthetic actors in the film Rendez-vous à Montréal. IEEE Computer Graphics & applications, Vol 7, No 12, pp 7-19
- Magnenat-Thalmann N and Thalmann D (1987b) Image Synthesis: Theory and practice. Springer, Tokyo
- Magnenat-Thalmann N, Laperrière R, Thalmann D (1988) Joint-Dependent Local Deformations for hand animation and object grasping. Proc. Graphics Interface '88, Edmonton
- Magnenat-Thalmann N, Thalmann D, Construction and Animation of a Synthetic Actress, Proc. EUROGRAPHICS '88, Nice, France, 1988, pp. 55-66.
- Moore M, Wilhelms J (1988) Collision detection and response for computer animation. Proc.SIGGRAPH '88, pp 289-298
- Paul RP (1981) Robot manipulators: mathematics, programming and control. The MIT Press, Cambridge, Mass.
- Platt JC, Barr AH (1988) Constraint method for flexible models. Proc. SIGGRAPH '88, pp 279-288
- Pugh A (ed.) (1986) Robot sensors. Vol 2. Tactile and non-vision. IFS publications Ltd (Bedford) and Springer Verlag
- Slotine JJE, Asada H (1986) Robot analysis and control, John Wiley and Sons
- Terzopoulos D, Platt J, Barr A and Fleischer K (1987) Elastically deformable models. Proc.SIGGRAPH '87, pp 205-214
- Thomson DE, Buford WL, Myers LM, Giurintano DJ and Brewer III JA (1988) A hand biomechanics workstation. Proc. SIGGRAPH '88, pp 335-343
- Wilhelms J (1987) Toward automatic motion control. IEEE Computer Graphics and applications, Vol 7, No 4, pp 11-22
- Witkin A, Kass (1988) Spacetime Constraints. Proc. SIGGRAPH '88, pp. 159-168
- Wyvill G, McPheeters C, Wyvill B (1986) Data structure for soft objects. The Visual Computer, No 2, pp 227-234
- Zienkiewicz OC (1977) The finite element method. Third edition, McGraw-Hill, London