グラフィクスとCAD 69－8
（1994．7．8）

Implicit Function手法を用いた人物像のための筋肉と皮膚表面のリアルなモデリングとアニメーション

カランシャー シン　　大谷 淳　　岸野 文郎

（株）ATR通信システム研究所
〒619-02 京都府相楽郡精華町光台2 - 2

本稿では、Implicit FunctionとBrep (Boundary Representation)に基づいた種々のアニメーション手法の適用が可能な、人物のような関節構造体を対象とする統合的な筋肉と皮膚表面階層モデルを示す。人物像は階層構造的に表現できる。人物の骨格構造の上に存在する筋肉と皮膚表面層のリアルなモデリングとアニメーションには、人物像の人体パーツ間の滑らかな接続を確保すること、皮膚表面におけるしわの形成のモデリングを行うこと、筋肉の盛り上がりや接触による変形を表現すること、といった課題がある。人物像のアニメーションを行う際に生じるこのような問題を解決するために、Implicit Functionを使う手法を提案する。さらに、前述のモデルと提案手法を、臨場感通信会議のリアリズム増加に応用する方法を検討する。

Realistic Modeling and Animation of a Muscle and Skin Layer for Human Figures using Implicit Function Techniques

Karansher SINGH,　Jun OHYA,　Fumio KISHINO

ATR Communication Systems Research Laboratories
2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-02, Japan

This paper presents a unified muscle and skin layer model for articulated figures, allowing the application of Implicit Function and Brep (Boundary Representation) based animation techniques. Human figures are represented using a layered approach. The realistic modeling and animation of a muscle and skin layer over a skeletal structure involves issues such as ensuring smooth connectivity between the animated segments of the virtual human, modeling the formation of creases, bulges and contact deformation on collisions. Solutions using implicit functions to solve problems resulting due to the animation of the human figure are presented. The application of the above models and ideas to enhancing the realism in a virtual space teleconferencing system is then discussed.

1 Introduction

Character animation is a widely researched area with a large number of applications such as the evolving area of virtual space teleconferencing [8]. Current photo-realistic rendering capabilities make the main problem one of character modeling and animation.

A layered approach [5] to the modeling and animation of articulated figures is currently a widely adopted methodology. With respect to human figures the layers may be broadly classified into:
1. Skeletal
2. Muscle, Skin and underlying tissue
3. Hair, Nails, Blemishes and other such features
4. Clothes and Accessories

The above layers are not necessarily mutually exclusive and are often omitted, collapsed together or further subdivided depending on the sophistication of the model. Building various levels of abstraction and automation of the animation process over this representation is presented in [10]. We would like the lower level representations to be modeled with enough generality, so that a number of high level techniques may be applicable (for example a Brep precludes implicit function deformable models from being applied). There is an outward dependency across the layers. The skeleton affects the shape of the muscle and skin, which in turn shape the apparel. Thus for good cloth model results on a human figure, the models of the underlying layers should be good. Modeling and animation of hair and clothing assuming underlying layers has received a fair amount of attention. An issue that is taken into consideration in this paper is the efficiency of collision detection with the underlying muscle and skin model, which is important for most hair and cloth models.

The skeletal layer is probably the best modeled as it may be approximated as an articulated rigid body. Despite the problems arising from the approximations [10], various robotics techniques adapted to the human figure have provided realistic results.

Modeling and animation of the **muscle, skin layer** has almost entirely dealt with Breps [5],[11],[16]. Polygon based structures are popular due to their simplicity, generality and hardware support. The bulk of modeling and animation techniques currently in use are Brep and often polygon based. Prototype(s) Brep model(s) (typically of the real figure in a relaxed pose is used). This represents the geometric skin. Muscle models then specify deformations on animation.

The deformable nature of human muscle, fatty and skin tissue is described in [16]. Physical tissue characteristics are modeled as spring and damper meshes attaching skin to the underlying skeleton. Forces are applied iteratively and the stabilized network shapes the skin. The paper deals specifically with facial animation. Free form deformations are used to empirically deform the skin layer in [5]. There is no explicit underlying muscle model. An empirical correspondence between joint angles and the deformation is made. A human skin model based on bezier surfaces in [9], controls deformations like [1], by manipulating the surface control points. Position of the skin Brep around joints in terms of a function that is specific and local to the joint skeletal area is presented in [11]. Further the use of FEM in animating a human hand in a grasping situation is shown. The muscle and skin layer is very important for visual realism and its effect on subsequent layers. This paper focuses on this layer and will propose an implicit function based muscle and skin model which addresses the problems of this layer well and further, may be unified with the above Brep techniques.

Implicit surfaces are a popular approach to object modeling and animation, especially applicable to physically deformable objects [1],[17], [6],[13],[12],[7]. An implicit surface is defined as the set of points P satisfying an equation $F(P) = 0$. Implicit primitive shapes are intuitive for building more complex shapes [1], which can be deformed easily by manipulating the individual primitives [17]. Continuity properties for the resulting surface can be ensured, and topological changes on animation are automatic. The static modeling of a realistic looking human hand using offset surfaces convolved over triangular skeletal facets is shown in [4]. Fitting of real data as in [12], typically creates of the order of hundreds of primitives, making their individual manipulation for desired effects difficult. A single superquadric primitive with modal deformations and a displacement map, however, can fit real data [13]. Implicit functions are used to model exact collision contact surfaces in [7]. Further implicit functions may replace discrete spring models with stiffness fields that are continuous in space. Efficient implicit function evaluation also greatly improves the efficiency of collision detection between objects [13]. Thus implicit surfaces show promise in handling many issues involved in articulated figure animation and would make a useful addition to existing Brep based techniques.

This paper provides a new unified framework for the application of Brep and implicit function techniques for human figure modeling and animation. To achieve this unification, Section 2 describes the use of (Brep) polyhedral structures as implicit surface primitives. A hierarchical model of the human muscle and skin layer, comprising of implicit primitives is then presented in Section 3, and techniques for handling collisions, deformations and their manifestations on animation are discussed. These implicit function techniques may be integrated with existing Brep techniques. Section 4 addresses the display aspects of this implicit model. The effectiveness of the developed muscle and skin model is illustrated by application to a virtual space teleconferencing system in Section 5. The unified nature of the model is also illustrated in Section 5 where various animation techniques are implemented on the same human model. Section 6 describes the implementation and Section 7 presents the conclusion.

2 Polyhedral Implicit Surface Primitives

A useful set of implicit surfaces can be generated as an algebraic combination of polynomial functions each of which is defined over a finite volume. For a summation (smooth blend) $F(P) = \sum f_i(P) - T$, where i runs over the primitive polynomial functions f_i and T is a threshold value $\in [0,1]$. A subset of these surfaces are **distance surfaces** and **convolution surfaces** [3]. Each primitive is defined by a finite volume V (typically spherical), a skeleton within the volume S (typically the center of the sphere) and a function f (typically a polynomial as in Figure 1). The primitive only contributes to the surface within V. Examples of $f : [0,1] \rightarrow [0,1]$, also called a density function, with the desired properties [17] are shown in Figure 1. For a point P within V the function value is determined by first computing a value $\in [0,1]$ called a distance ratio. *f(distanceratio)* is then the function value. The distance ratio is computed by taking the ratio of the shortest euclidean distance from P to a point Q on S, and a value determined by the shape of V (usually a constant radius value R). Such a primitive is an **offset surface** [3]. Distance surfaces are generated when radius R need not be constant.

The shape of an implicit primitive should thus provide a bounded volume [17], the interior of which is the realm of influence of the associated function, as well as a mapping from a point within the realm of influence to a value within the domain of the density function, which is used to calculate the function at that point.

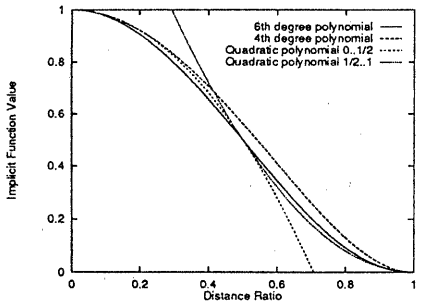

Figure 1: Polynomial Density Functions

2.1 Polyblobs

Define a polyhedral implicit primitive shape (referred to as a polyblob) as $< V, S, f >$, where V is a non empty bounding volume, S is a finite surface (open or closed) contained within V and f the density function (Figure 1). The realm of influence of f is the interior of V. Given a point P within V, let Q be the point on S with the shortest euclidean distance to P. Let the ray from Q through P intersect V at point P'. Let us restrict V, S to be such that the intersection P' is unique. The polygon of V containing P' is called the **defining polygon** of P [14]. Define $g : \mathcal{R}^3 \rightarrow [0,1]$. For a point P, $g(P) = min(1, \frac{|PQ|}{|PP'|})$. Clearly $g(P)$ is a value that ranges from 0 for points on or inside the surface S to 1 for point on or outside the volume V. Now the function value at P is given by $f(g(P))$.

For polyblobs, scaled volumes result as the isolated primitive shape at various thresholds when S is a single skeletal point contained within V (See Figure 2). In general, isolated primitive shapes for various thresholds are an intermediate shape between V and S. Throughout this paper, S for a polyblob is assumed to be a single skeletal point contained within V.

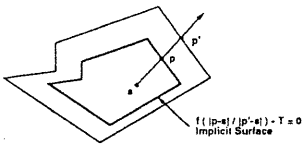

Figure 2: 2 Dimensional Polyblob

The computation of g associates a unique point P' on V, with every point P within V. P' can be used in determining the normal, texture and other attributes (that are translation, uniform scaling invariant) for the primitive at point P. The attributes for all contributing primitives are weighted appropriately to obtain the attribute at P for the implicit surface.

There may be cases such as Figure 7, where, clearly a single point is not sufficient to allow for functional interaction at both fingers of the hand. Further with many different regions of interaction in a primitive a measure of local control over the interaction in different regions of the primitive is useful. A single point within a primitive lacks this property.

2.2 Modified Polyblobs

Modified polyblobs may be motivated from the example of articulated figure modeling and animation, where each limb is a polyblob and the figure a blended result. It may be impossible to satisfy the restriction on V by the placement of a single skeletal point within the polyblob, for arbitrarily shaped limbs. We would also like to localize the blending centered around joints (for example the upper arm may blend differently at the shoulder and elbow). This could be provided by the general definition of the polyblob where the inner skeleton S could be the set of joint centers. The problem, however, is that the final shape of the isolated polyblob is unpredicatable for an arbitrary S. This is unacceptable as we

wish to preserve the shape of the realistically obtained outer volume V.

Modified polyblobs rely on the assumption that for many applications such as articulated figure animation, the threshold value of an implicit object remains constant. A modified polyblob is defined as $< W, S, F, T >$. W is the shape of the isolated primitive desired at a surface threshold T. $S = s_1..s_n$ is a discrete set of points within W. Typically S would represent the various blend centers. $F = f_1..f_n$ is a set of blending functions corresponding to each point in S, each of which possess the properties of Figure 1. $T \in [0, 1]$ is the fixed threshold value for the object. Construct volumes $V_1..V_n$, where V_i is $Trans(s_i)Scale(1/f_i^{-1}(T))Trans(-s_i)W$. Put simply, V_i is the appropriately scaled up volume, which treated as an isolated polyblob $< V_i, s_i, f_i >$, would give the shape W for a surface threshold value of T (See Figure 3).

The function value at a point P for modified polyblob $< W, S, F, T >$ is computed as follows:

1. Let $s_i = min_{j \in 1..n}(|s_j - P|)$.
2. The function value at point P is the function value returned by polyblob $< V_i, s_i, f_i >$ at point P.

Hence the isolated shape of the modified polyblob for the given threshold T, is W. Further blending in a region near a blend center is locally controlled by its corresponding function due to spatial proximity.

This provides a satisfactory solution as long as the modified polyblob has no functional interaction around a voronoi boundary of the set S. For points on the boundary the functions are only guaranteed to return the same value for points on W or where all functions evaluate to T. Thus if there is no interaction with other primitives on a boundary, the surface at T around the boundary will be continuous and precisely W.

A robust solution in [14] maintains continuity across the voronoi boundaries at all thresholds and preserves the previously achieved properties. Here the function values across voronoi boundaries are blended together with first order continuity, the extent of the blend user controlled by a parameter R (See Figure 3). This is illustrated in Figure 7 where the hand is a modified polyblob that blends seamlessly and differently at two centers in the thumb and forefinger respectively.

3 Muscle and Skin Model

The muscle and skin model aims at automating the dynamic interaction of muscles with the environment. This in turn shapes the skin which has its own physical characteristics. The models proposed combine ideas dealing with Brep and implicit function representations, allowing their application through a unified framework. The models attempt to address both geometric and physical issues involved at various levels of sophistication.

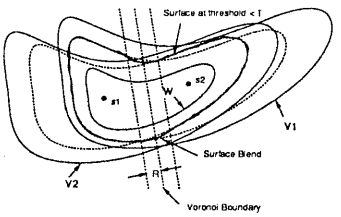

Figure 3: 2 Dimensional Modified Polyblob

The following two implicit function concepts are of importance to our proposed model.

Interacting Implicit Objects: Implicit Objects have their own implicit functions that determine their surface. Different objects may interact by imparting an additional implicit function to one another [7]. The fields imparted model interactions such as adhesion, collision deformations, fusion and fission. Collision contact [7] (See eyeballs in Figure 7) is of great importance in this paper and is used to handle collisions involving the human figure such as the precise crease formation at the elbow joint (See Figure 9,11).

Directional Blends : The notion of directional blending deals with a variation in the behavior of the density function or the functional combination within the same primitive. The problem lies in intuitively specifying that a particular region of the primitive blends in a different way from another. Modified polyblobs provide an intuitive solution to directional blend specification (See different blends at forefinger and thumb in Figure 7). Further functional attributes maybe assigned to the polyhedral vertices of a polyblob. Functional calculation at a point is then a smooth functional interpolation of the attributes at the vertices of the defining polygon.

3.1 Geometric Model

The skin comprises a fixed number of general implicit surface primitives combined in a prespecified manner. Prototypes for these may be obtained from real data as Brep (polyblobs) or superquadrics. (See Figure 4). Additional analytic primitives may be used for control in special conditions like animating bumps, scars, blemishes, or for exaggerated muscle motion (See Figure 11).

The primitives are selectively blended based on the hierarchical structure in Figure 4. Primitives that do not blend mutually, are treated as different implicit objects. All implicit objects in the environment interact with each other for collision detection and deformation. Thus auto collisions between different parts of the body are homogeneously handled. Joint regions such as the elbow where both blending and collision deformations occur, may be handled using directional blending or by

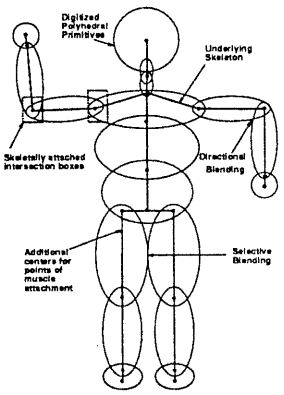

Figure 4: Hierarchical Implicit Primitive Human Model

introduction of a dummy primitive that blends the two arm primitives together (See Figure 11,9).

The model may be animated simply by manipulation of the individual polyblobs based on the underlying skeletal model. The implicit function model automatically maintains a smoothly blended body as well as collision deformations.

3.2 Physical Model

Physical interpretations of the above model, which in conjunction with the skeleton control the animation of the geometric model are proposed. Issues that need to be addressed are the deformable nature of the tissue, modeling of forces generated by the figure due to muscle actuations and reactions (both auto and with external objects/forces) The results should be manifested in terms of collision contact, creases and bulges.

The general deformable model techniques proposed in [15], may be applied directly in the context of polyblobs, by working on the discrete Brep primitives and then converting to the implicit function model. [7] uses the density function gradient to model radial spring stiffness. Objects are considered as rigid with a deformable layer superposed. Such a model is well suited to articulated figures that are supported by more or less rigid skeletal components. Variations in tissue characteristics [16] are modeled by piecewise smooth polynomials whose gradient reflects the change in stiffness.

The modified polyblobs in addition to joint blending centers also have skeletal points that serve as points of attachment of muscles (Figure 4). Muscle actuations are then modeled by changes in the shape of the field function. This causes an internal force due to the change in stiffness which will affect the shape of the implicit surface as the threshold value remains the same.

The computational loop [7] then applies the given forces, torques in the system to the primitives and other rigid components of the system. Implicit surface representations using the deformable model are generated and implicit functions updated. New forces and torques in the system are then generated from function values as well as forces such as reaction and friction calculated from collision contact surfaces.

The implicit model can be combined with existing Brep based techniques for polyhedral primitives (Figure 10). The rigid primitives are deformed using techniques presented in [5],[11], before application of the implicit model on the deformed primitives.

3.3 Special Features

Veins may be modeled effectively using a technique similar to [4]. Analytic offset surfaces around free form curves may be used to model a network of veins blended with the geometric skin model (See Figure 9).

Bones often come close to the surface and actually define the shape of the skin on animation [10]. The implicit model handles this elegantly. Primitives for the skeletal structure are specified and blended with the geometric skin primitives. On animation the skeletal primitives contribute to the shape of the skin automatically, only when the primitive is close to the geometric skin primitive (See the left elbow in Figure 10).

Wrinkles are handled well by directional blending. They may be generated by directionally changing the density function gradient, which physically corresponds to muscle actuation. Bulges are handled similarly.

4 Rendering the Implicit Model

A major drawback of implicit surfaces is the time taken to sample space in order to determine the implicitly defined surface [1],[14].

Implicit surface rendering may be done by directly determining the ray surface intersections by solving the implicit equation along the ray using numerical or analytic techniques [1],[6],[14]. A detailed description of various changes to existing optimizations catered to polyblobs and some specific ones maybe found in [14].

The interval along a ray over which the polyblob is active is partitioned into a sequence of intervals, subtended by adjacent defining polygons. The distance ratio over the interval of a defining polygon is a linear function of t, the parametric distance along the ray [14]. Thus the implicit function along the ray is a number of quadratic functions in t (See Figure 1). These can then be solved in order, for the first intersection point. The technique is robust and efficient for rays with a small number of defining polygons. The complexity grows linearly with the number of defining polygons. Its locality of application and robust nature make it couple well with the optimizations and heuristics described in [1],[14]. Incremental span computations also make polyblobs suitable for efficient scanline rendering.

Alternatively a **surface reconstruction** algorithm may be applied to construct a Brep of the surface [2],[18], which is subsequently rendered. Polygonization of general implicit surfaces is described in [2] using convergence and surface tracking techniques with an octree structure.

A motivating factor of polyhedral primitives was to reduce the number of modeling primitives used. In such cases there are likely to be regions where a single primitive defines the surface. Here, the primitive itself, appropriately scaled, defines a polygonal representation of the surface. This provides both a partial polygonization of the resultant object and information of the behavior of the surface at the point where the primitive starts interacting with other primitives. The following algorithm for polygonization of polyblobs is proposed.

1. Compute the intersection of bounding boxes of all interacting primitives.

2. Clip the polyhedral models of each isolated primitive to the the intersection boxes. Retain the outside as part of the final polyhedral structure.

3. The clipped edges of the primitives determine the octree structure on the boundary of each intersection box and also provide a number of seed cubes.

4. The seed cubes are used by an adaptive surface tracker [2] to polygonize each intersection box.

An approach to exploiting the parallelism potential of polygonization is described in [18]. This approach is further catered to parallelism as a result of the polygonization of various disjoint intersection boxes.

We need to ensure a seamless connection without cracks of the retained polyhedral structure and the polygonized surface within an intersection box. For each clipped edge the following action as shown in Figure 5 is taken, the results of which on a blended elbow are shown in Figure 8.

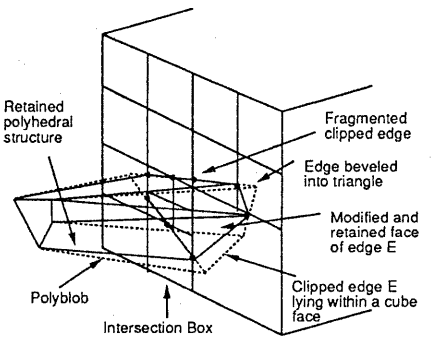

Figure 5: Clipping Isolated Polyblobs

- The edge is fragmented using a DDA type line algorithm to the edges of the octree cubes in the plane of the intersection box.

- The edges of retained polygons adjacent to the clipped edge are bevelled to triangles.

- The retained polygon of a clipped edge lying entirely within a single cube is connected to the cube-surface intersections. A sequence of clipped edges all within the same cube cause partitioning until at most a single clipped edge lies within a cube.

5 Virtual Space Teleconferencing

At ATR CSRL, the authors are trying to build the VIrtual Space TELeconferencing (VISTEL) [8] system, which aims at an environment where teleconference participants at different sites can feel as if they are all at one site, allowing them to hold meetings and work cooperatively. In VISTEL, the 3D models of the participants at different sites are combined into an artificially created 3D image of a virtual space, and by displaying the 3D image of the virtual space on the 3D screen at each site, the participants will get the feeling of meeting each other in a common space. The VISTEL framework is shown in Figure 6 and Figure 12.

Human figure data is obtained as a number of digitized parts using a Cyberware Color 3D digitizer. The parts are then registered with color textures and fitted together on a virtual skeleton (See Figure 10,12). The implicit model may be used to connect digitized segments and blend the employed textures (See Figure 8,9). Animation requires ensuring a smooth connectivity between the animated pieces of the virtual human, modeling the formation of creases, wrinkles and bulges, modeling deformable contact surfaces and preventing penetration on collisions. Further, efficiency of the model for a real time implementation is desired.

Figure 10, shows an example of the hybrid model. Free form deformations [5] on the spine animate the torso. The right elbow joint is modeled using a joint local deformation [11] and the left with the implicit model.

6 Implementation

The modified polyblobs were raytraced using a hybrid intersection algorithm [14]. Figure 9 with object complexity of around 1000 polygons took 220 sec. on the average, for a 400X400 adaptively antialiased image. The time spent sampling space in the region of blending due to the implicit model was 30 sec. A hierarchical slabs, BSP of the polyblobs is used. Further the region of the blend was isolated, greatly improving the implicit function evaluation efficiency.

The polygonization algorithm implemented is the high level algorithm described in section 4, with the surface tracking algorithm described in [2] operating within each intersection box. Figure 8 was polygonized in 3 sec. with 0.2 sec. spent on the intersection box computation and clipping operation. The number of original faces retained due to clipping was 774 out of 907. The polygonized model comprised of 1017 faces.

All the above timings were taken on a SUN Sparc2.

Figure 12 shows the VISTEL system. The WS(Iris Crimson, Reality Engine), characterized by its fast texture mapping function, is used for real-time 3D display in a synthesized virtual space, on a 70-inch stereoscopic display.

The polygonization of the implicit model in (Figure 10) employs a precomputed intersection box around the joint (See Figure 4) where the function is computed. Motion at the joint may be considered relative to one of the limbs. Thus in this case the upper arm values are precomputed at regular intervals within the intersection box. This greatly reduces the number of evaluations of upper arm polyblob.

7 Conclusion

The implementation and results show implicit surfaces to be an elegant way of handling many of the issues involved in modeling the muscle and skin of an articulated figure. Polyblobs unify Brep and Implicit Surface techniques. In conjunction with implicit surface polygonization, the object model can easily change representations. Polyblobs also facilitate incremental modeling which keeps the number of polyhedral primitives down and affords local control. Modified polyblobs are a useful extension to polyblobs as they allow a more general class of primitives, facilitate directional blends and local control. A major disadvantage of polyblobs is the loss of implicit functional definition in a closed form. The closed form exists only over a single defining polygon. Thus better algorithms that efficiently determine the defining polygon need to be sought. Polygonization timings show that with optimizations, parallelism and faster hardware real time animation with the implicit model is feasible.

References

[1] J. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3):235–256, 1982.

[2] J. Bloomenthal. Polygonization of implicit surfaces. *Computer Aided Geometric Design*, 5:341–355, 1988.

[3] J. Bloomenthal and K. Shoemake. Convolution surfaces. *Computer Graphics*, 25(4):251–256, 1991.

[4] J. Bloomenthal. Hand Crafted. *SIGGRAPH Implicit Surface Course Notes*, 11.1–11.3, 1993.

[5] J. Chadwick, D. Haumann and R. Parent. Layered construction for deformable animated characters. *Computer Graphics*, 23(3):234–243, 1989.

[6] A. Fujimoto T. Tanaka and K. Iwata. Arts: Accelerated ray tracing system. *Computer Graphics and Applications*, 6(4):16–26, 1986.

[7] M. Gascuel. An implicit formulation for precise contact modeling between flexible solids. *Proc. of SIGGRAPH*, pages 313–320, 1993.

[8] F. Kishino. Communication with realistic sensations through integrated multi-media environment. *Proc. of Language and Vision Workshop(1st US-Japan Workshop on Integrated Systems in Multimedia Environments)*, 49–58, 1991.

[9] K. Komatsu. Human skin model capable of natural shape variation. *Visual Computer*, 3:265–271, 1988.

[10] N. Magnetat-Thalmann, D. Thalmann. Complex models for visualising humans, *SIGGRAPH Course Notes C20*, 3–10, 1991.

[11] N. Magnetat-Thalmann, D. Thalmann. Human body deformations using Joint Dependent Local Operators and Finite Element Theory. *Making Them Move*, Morgan Kaufmann, 243–262.

[12] S. Muraki. Volumetric shape description of range data using the blobby model. *Computer Graphics*, 23(3):234–243, 1991.

[13] S. Sclaroff and A. Pentland. Generalized implicit functions for computer graphics. *Computer Graphics*, 25(4):247–250, 1991.

[14] K. Singh and R. Parent. Polyhedral shapes as gneral implicit surface primitives. *Ohio State Univ. Tech. Rep. OSU-CISRC-5/94-TR24*, 1994.

[15] D. Terzopoulos and K. Fleischer. Deformable Models. *Visual Computer*, 4:306–331, 1988.

[16] D. Terzopoulos and K. Waters. Physically based facial modeling, analysis and animation. *Journal of Visualization and Computer Animation*, Vol.1, No.2, 73–79, 1990

[17] G. Wyvill C. McPheeters and B. Wyvill. Data structures for soft objects. *Visual Computer*, 2:227–234, 1986.

[18] B. Wyvill, H. Zhang and A. Davis. Parallel implicit surface polygonization on a simulated parallel architecture. *Proc. of the Western Computer Graphics Symposium*, 35–46, 1992.

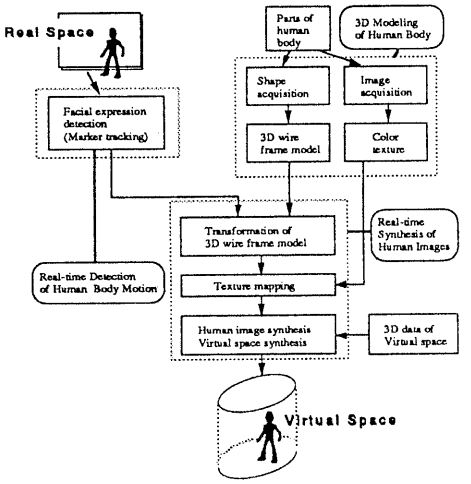

Figure 6: VISTEL Framework

Figure 7: Modified Polyblob

Figure 8: Polygonized Elbow

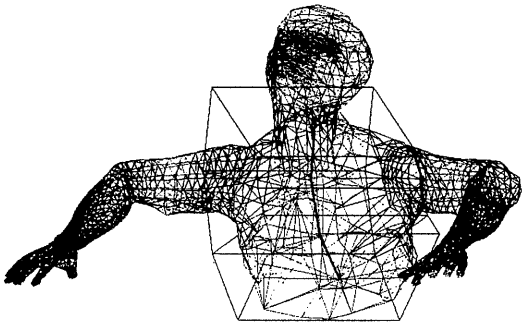

Figure 10: Hybrid Muscle/Skin Model

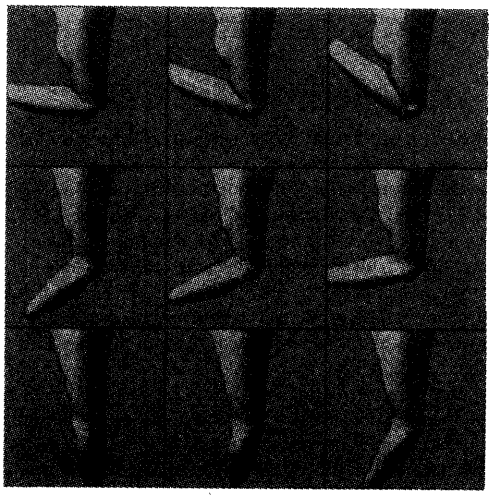

Figure 11: Animated Arm with Directional blending

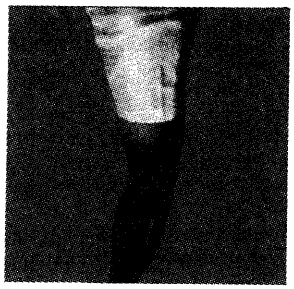

Figure 9: Arm with Selective blending

Figure 12: VISTEL system