Chapter 5
V-Man Generation for 3-D Real Time Animation
***

Jean-Christophe Nebel, Alexander Sibiryakov, and Xiangyang Ju

University of Glasgow, Computing Science Department
17 Lilybank Gardens, G12 8QQ Glasgow United Kingdom
{ jc, sibiryaa, xju}@dcs.gla.ac.uk

Abstract

*The V-Man project has developed an intuitive authoring and intelligent system to create, animate, control and interact in **real-time** with a new generation of 3D virtual characters: **The V-Men.** It combines several innovative algorithms coming from Virtual Reality, Physical Simulation, Computer Vision, Robotics and Artificial Intelligence. Given a high-level task like "walk to that spot" or "get that object", a V-Man generates the complete animation required to accomplish the task. V-Men synthesise motion at runtime according to their environment, their task and their physical parameters, drawing upon its unique set of skills manufactured during the character creation. The key to the system is the automated creation of realistic V-Men, not requiring the expertise of an animator. It is based on real human data captured by 3D static and dynamic body scanners, which is then processed to generate firstly animatable body meshes, secondly 3D garments and finally skinned body meshes.*

1. Introduction

Over the last decade, real-time computer graphics has been subject of vigorous research activity leading to amazing progress and innovation that performance of general-purpose computing platforms has passed the threshold to make it possible to simulate realistic environments and let users interact with these virtual environments with (almost) all senses in real-time. However, creating, animating, and controlling 3D individualized characters is still a long and manual task requiring the skills of experienced modellers and the talent of trained animators on specific software packages- using Maya™, Poser4™ or Reflex-Drama™ - or by scaling generic 3D human models to fit the shape of particular individuals [1] and [2]. Tasks aim at predefining shapes of individuals; deformation and animation are properly incompatible with a really interactive application.

Here we made use of 3D human body scanners to predefine human body shapes. 3D scanning system allowed us to capture whole human body automatically with accurate body shape of individuals and realistic photographic appearances [3, 4, 5, 6]. Directly animating the scanned human body data requires the data to be articulated because there is no semantic information embedded in the data. The articulation can be done manually or semi-automatically [7, 8]. In the conformation approach we proposed, an animatable generic model conformed to the scanned

data that not only articulated the scanned data but also maintained the topology of the generic mesh.

A realistic animation of 3D models requires the knowledge of physical properties regarding the skin and the soft tissues of the real human [9, 10, 11, 12, 13, 14]. In order to collect that information we captured 3D data from people in several key positions. Since we used a scanner based on photogrammetry [15, 16] which has a capture time of few milliseconds, we had the unique opportunity to analyse the skin deformation during the motion between 2 positions. That allowed us to simulate the elasticity of the skin and soft tissues for any vertex of the surface. The association of a skeleton and vertex properties, such as weights, allowed the character skin to be deformed realistically, in real-time, based on correspondence between vertices of the shape and specific bones.

Simulation of cloth in real time is another difficult but possible task if there are static constraints and the detail required is low. Dressing a character is not so simple. This is due to the complicated geometry of a character's body surface and the highly detailed crease patterns that form where clothing is constrained. Simulating the latter patterns in a computationally efficient manner is an open problem but is a necessity to create the detail that essentially characterizes a garment to the human eye. Three main strategies have been used for dressing 3D characters. Garments can be modelled and textured by an animator: this time consuming option is widely spread in particular in the game industry where there are only few characters with strong and distinctive features. Another simple alternative is to map textures on body shape of naked characters; however that technique is only acceptable when characters are supposed to wear tightly fitted garments. In the other approach garments can be assembled: patterns are pulled together and seamed around the body. Then physically based algorithms calculate how garments drape when resting. This accurate and time-consuming technique (requiring seconds [17] or even minutes [18] depending on the required level of accuracy) is often part of a whole clothing package specialised in clothes simulation. Among these strategies, only the third one is appealing since it provides an automated way of generating convincing 3D garments.

Besides conformation, skinning and dressing, a set of skills were built into the characters: there are motion skills, physical characteristics and collision avoidance skills. A V-Man character created is able to adapt his movements and actions to his environment and situation, to walk on any kind of terrain; to go upstairs, downstairs; to calculate paths [19] in order to avoid obstacles. All these interactions will take into account the physical parameters of the V-Man which are added by skill building during character creation. The user can interact with the V-Men that a V-Man understands high-level multimodal commands.

The V-Man system comprises of V-Man character creation and Character Interaction with environment (Table 1) with increasing level of intelligence from top down. All these enabled us to develop our intuitive authoring system allowing any kind of user, without any particular animation skills, to create, animate, control and interact with 3D virtual characters: the V-Men. The system is a stand-alone VR application capable of exporting animation sequences in different standardise formats; it also allows the users to populate their visual simulations or video games with realistic autonomous virtual characters. The V-men characters can be imported into major computer graphics applications (3D Studio, Maya, etc.) and virtual worlds.

Table 1. V-Man System		
V-Man Character	Interaction with Environment	Level of intelligence
• Conformation to articulate the scanned data and to create animatable meshes • Skinning to mimic the deformation due to muscle movements • Animation skills and Physical characteristics • Collision avoidance skills	• Physical animation • Path planning • Get target objects • Voice and keyboard control	Low ↓ High

2. Creation of V-Man Character

V-Man character creation started from an animatable generic model. The generic model conformed to the scanned shapes of individuals to define the shapes of characters the same time maintained the animatable body mesh structure. 3D garments were modelled through the conformation procedure also. In skinning we took advantages of our dynamics scanners to associate vertices with weighted bones so that realistic surface deformation could be achieved. Essential skills were built in the characters for interactive and intelligent animation, such as motion skill, collision avoidance, body size and weight and joint limits.

2.1 Conformation

The conformation algorithm, which conforms a generic model to scanned data, comprises a two-step process: global mapping and local deformation. The global mapping registers and deforms the generic model to the scanned data based on global correspondences, in this case manually defined landmarks. The local deformation reshapes the globally deformed generic model to fit the scanned data by identifying corresponding closest surface points between them and then warps the generic model surface in the direction of the closest surface.

2.1.1. Global Mapping

Global registration and deformation are achieved by means of a 3D mapping based on corresponding points on the generic model and the scanned data. The 3D mapping transforms the manually defined landmark points on the generic model to the exact locations of their counterparts (also defined manually) on the scanned data. All other points on the generic model are interpolated by the 3D mapping; this mapping is established using corresponding feature points through radial basis functions [20].

The global mapping results in the mesh of each body component of the generic model being subject to a rigid body transformation and then a global deformation to become approximately aligned with the scanned data.

2.1.2. Local Deformation

Following global mapping, the generic model is further deformed locally, based on the closest points between the surfaces of the generic model and the scanned data. The polygons of the generic model are displaced towards their closest positions on the surface of the scanned data. Since the body models have been segmented, it is possible to avoid making erroneous closest point associations between points within the surfaces of the limbs and the torso.

An elastic model was introduced to the second step of the conformation. The global deformed mesh is regarded as a set of point masses connected by springs. During the second stage of the conformation process, the mesh anchored to landmarks is relaxes to minimize its internal elastic energy. The displacements of the vertices deformed to their closest scanned surface are constrained to minimize the elastic energy of the mass-spring system. The governing equation to vertex ***i*** is

$$m_i \ddot{X}_i + d_i \dot{X}_i + \sum f_{ij} = f_i^{ext} \qquad (1)$$

where m_i is the mass of the vertex ***i***, d_i its damp factor, f_{ij} is the internal elastic force of the spring connect the vertex ***j*** to ***i*** and $f^{ext}{}_i$ the sum of the external forces.

The relaxation takes the following steps, more details can be found in [0] and [0]:

a. Every vertex deformed to its closest scanned surface except the vertices of the facets intersected the landmarks.

b. The sum of the internal forces on each vertex is calculated, no external force in this model.

c. The acceleration, velocity and position of each vertex are updated.

d. Repeat step a to c until the mesh is settled or the iteration exceeds a fixed number.

Having established global correspondences between the generic model and the 3D captured data, through the landmark-guided articulation procedure, the generic model has been deformed globally to align with the 3D captured data. Exact correspondence between landmarks has been maintained while the positions of all other points have been interpolated. Comparing the 3D imaged model data (Figure 1b) to that of the generic model (Figure 1a), the polygons of the generic mesh were deformed towards their closest positions on the surface of the captured data. Figure 1c shows this final conformation result, rendered using smooth shading. Figure 2 shows the magnified wire-frame body surface representations to illustrate the differences in mesh topology between the captured human body data mesh, Figure 2b, and the conformed generic model mesh, Figure 2c. The conformed generic model has the same mesh topology as the generic model, Figure 2a, but has the individualized shape, i.e. topography, of the 3D imaged real-world data.

Figure 1. (a) Generic model; (b) 3D imaged body data; and, (c) final conformed result.

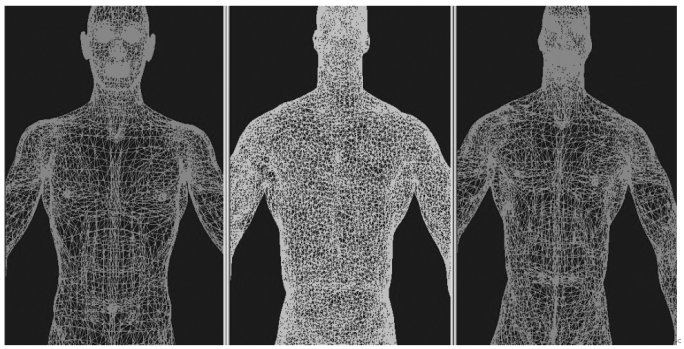


Figure 2. (a) Generic mesh; (b) 3D imaged body mesh; and, (c) final conformed mesh.

2.1.3 Garment

To dressing up the V-Man characters, we offered an innovative technical solution for the generation of 3D garments by capturing a same individual in a specific position with and without clothing. Generic garment meshes are conformed to scans of characters wearing garment (Figure 3) to produce 3D clothes. Then body and garment meshes can be superposed to generate the 3D clothes models.

To dressing up the V-Man characters, we offered an innovative technical solution for the generation of 3D garments by capturing a same individual in a specific position with and without clothing. Generic garment meshes are conformed to scans of characters wearing garment (Figure 3) to produce 3D clothes. Then body and garment meshes can be superposed to generate the 3D clothes models.

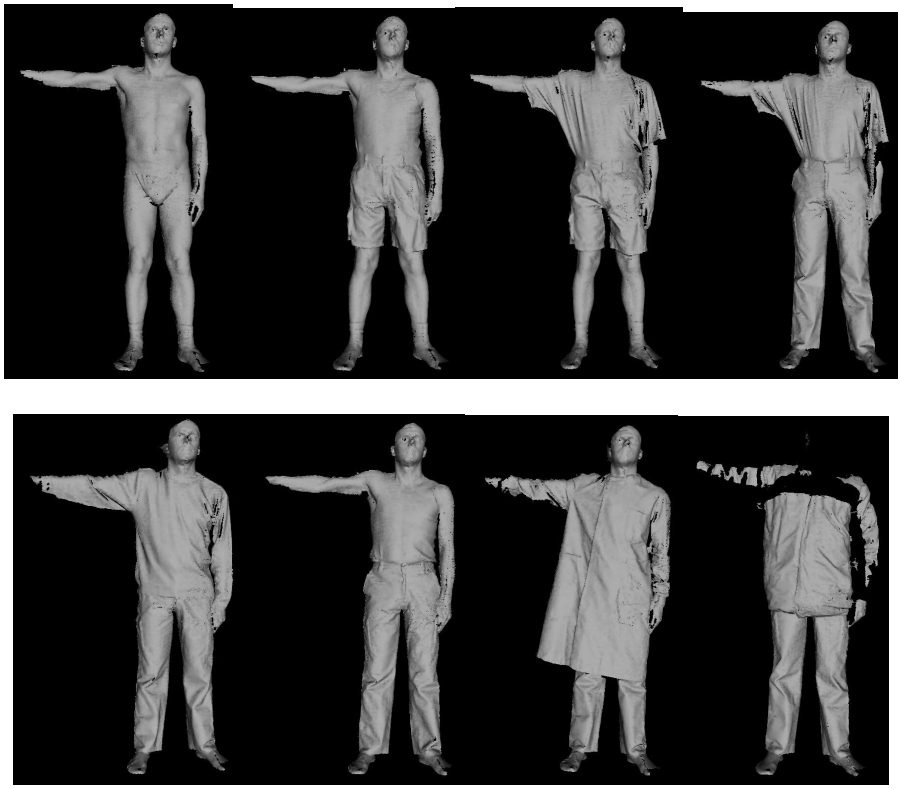


Figure 3. Same male model scanned in different outfits.

Scanning data are conformed using different generic meshes. There is one generic mesh by type of outfit, which provides an automatic garment extraction (Figure 4).

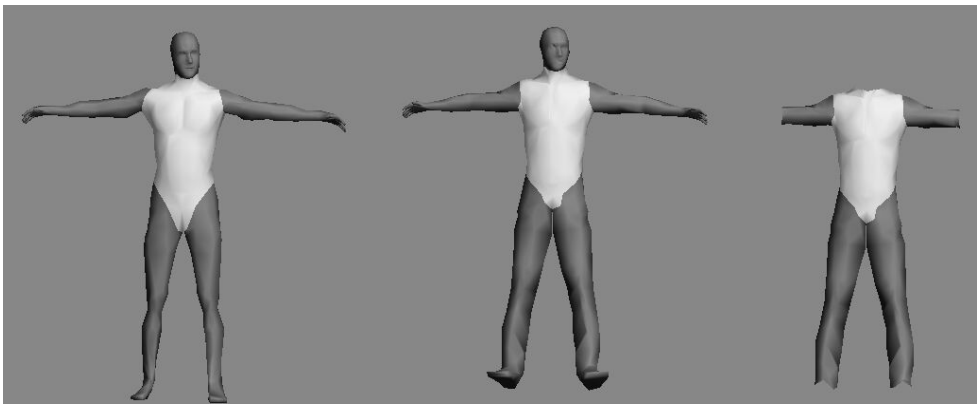


Figure 4. Conformed scanned model without (a) and with (b) garment and garment extraction (c).

The garment mesh is fitted on the naked mesh (Figure 5). Since we work with conformed mesh of generic topology, that fitting process is fully automatic. However, because of the accuracy limitation of the mesh, at that stage it cannot be ensured that the garment mesh will always be above the naked mesh, in particular in areas where clothes are tightly fitted—such as on

shoulders. Therefore a final process detects all triangles from the naked mesh that intersect with the garment mesh and move them backward.

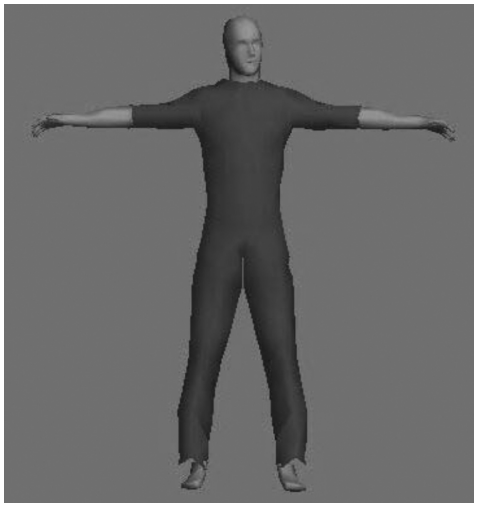


Figure 5. Scan superposition.

Since the number of outfits with different geometries is quite low, the variety of clothes and their important features will come from texture maps. Therefore each outfit is connected to a set of texture maps providing the style of the garment. Moreover users are provided with user-friendly software that allows them to create new texture maps in a few minutes.

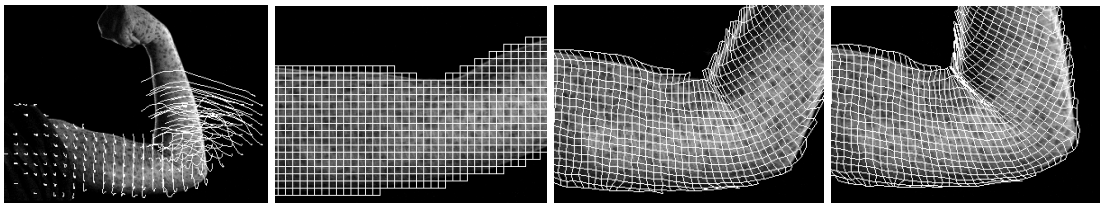
The process of texture generation is the following: using the given flatten mesh of a generic mesh and a set of photos which will be used as texture maps, users set landmarks on the photos corresponding to predefined landmarks on the flatten mesh. Then a warping procedure is operated so that the warped images can be mapped automatically on the 3D mesh (see Figure 5). Moreover areas where angles should be preserved during the warping phase—i.e. seams defining a pocket—can be specified.

2.2 Skin and Soft Tissues

3D character animation in most animation packages is based on the animation of articulated rigid bodies defined by a skeleton. Skeletons are supporting structures for polygonal meshes that represent the outer layer or skin of characters. In order to ensure smooth deformations of the skin around articulations, displacements of vertices must depend on the motion of the different bones of the neighbourhood. The process of associating vertices with weighted bones is called skinning and is an essential step of character animation. Tools are provided by these animation packages usually generate geometric deformation only and it is usually an iterative process to mimic realistic anatomical deformation.

Instead of iteratively trying to converge towards the best skinning compromise by hand, we offer to skin automatically a model from a set of 3D poses which are anatomically meaningful. The technology we use to generate range data is based on stereo-pair images collected by the camera pairs, which are then processed using photogrammetric techniques [0]. By tracking features over

time we could generate optical flows representing the deformation of the human skin. By combining series of range maps with their corresponding optical flows, we can then generate the range flows we need for the analysis of soft tissue deformation. Using the set of 3D points, we can trace each point from the reference posture and obtain the 3D deformation of each point (range flow).

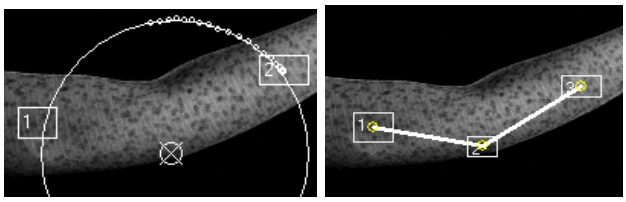


(a) (b) (c) (d)

Figure 6. Using range flows for skinning: (a) point tracing; (b) the reference image with its reference grid; (c) and (d) two images from the sequence with their deformed reference grids.

First we select one reference image and its corresponding 3D model from the sequence. Using the range flow we can trace each point from the reference image and obtain a deformation of any reference grid as shown on Figure 6 for the 2D case. Our method consists in obtaining the joints of the skeleton bones and computing the weights of the points of the 3D model with respect to the bones. The manual step of the method is in the approximate selection of two regions belonging respectively to the parent bone and the child bone. Due to the direct relation between the image pixels and the vertices of the 3D model, this operation can be performed on the reference image. The rest of the process is fully automatic. Using the range flow we obtain the positions of the centre of each region in all the 3D models of the sequence. The centre and the orientation of a global coordinate system are set in the parent bone region and the positions of the centre of the child bone region are then registered in that system. We assume that the bone motion is nearly planar (so we do not consider bone bending). Therefore we can fit a plane passing through the origin of the coordinate system and all the registered positions of the child bone centres. Then we analyse the 2D-motion in that plane. First we project the positions of the child bone centres in that plane, and then since the motion is circular, we can fit a circle on these points. The centre of the circle represents the 2D position of the joint and its 3D position is calculated. In Figure 7a the manually selected regions are shown as rectangles (the first rectangle represents the parent bone region and the second one is the child bone region). The small circles show the positions of the centre of the child bone region in the parent coordinate system. Figure 7a also shows the fitted circle with its centre defining the position of the joint.

Now two 3D vectors connecting the joint and the user defined regions can be fully determined for the whole sequence (Figure 7b). We consider them as virtual bones because they rotate around the joint and coincide more or less with the real bones. With a wider field of view, we could have calculated by the same method the positions of the 3 joints that would have defined more precisely the positions of these virtual bones.



(a) (b)

Figure 7. (a) Joint determination and (b) virtual bones.

The next step of the method is to assign to each vertex of the 3D model a set of weights associated to each bone. We use the following model of vertex motion:

$$\mathbf{x'} = \sum_{i=1}^{n} w_i \mathbf{R}_i \mathbf{x}_i \tag{2}$$

where **x'** is the deformed position of the vertex, n is the number of bones, w_i is the scalar weight associated to the i-th bone, $\mathbf{x}_i$ is the original position of the vertex in the i-th bone coordinate system and $\mathbf{R}_i$ is the transformation matrix of the i-th bone.

The 3D-rotation matrices **R** can be found from the previously calculated motions of the virtual bones. Let $\mathbf{r}_0$ and **r** be the vectors defining a virtual bone in the reference 3D model and in any other 3D model. A bone rotation can be described by an axis **p** and an angle α.

$$\mathbf{p} = \frac{\mathbf{r}_0 \times \mathbf{r}}{|\mathbf{r}_0 \times \mathbf{r}|}, \quad c = \cos\alpha = \frac{\mathbf{r}_0{}^T \mathbf{r}}{|\mathbf{r}_0||\mathbf{r}|}, \quad s = \sin\alpha = \frac{|\mathbf{r}_0 \times \mathbf{r}|}{|\mathbf{r}_0||\mathbf{r}|} \tag{3}$$

and using the Rodrigues formula:

$$\mathbf{R} = \begin{bmatrix} c+(1-c)p_x^2 & (1-c)p_x p_y - sp_z & (1-c)p_x p_z + sp_y \\ (1-c)p_x p_y + sp_z & c+(1-c)p_y^2 & (1-c)p_x p_y - sp_z \\ (1-c)p_z p_x - sp_y & (1-c)p_z p_y + sp_x & c+(1-c)p_z^2 \end{bmatrix} \tag{4}$$

For simplicity we present the 2D case of the motion of a two-bone system around a joint. For this we project the 3D-vertices to the plane previously described. In this case the $\mathbf{R}_i$ are 2D-rotation matrices and equation (1) becomes:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = w_1 \begin{bmatrix} a_1 & -b_1 \\ b_1 & a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + w_2 \begin{bmatrix} a_2 & -b_2 \\ b_2 & a_2 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \tag{5}$$

where w1 and w2 can be easily found. Real motions of skin points are more complex then those expressed by the model (1). They are determined not only by the skeleton but also by muscles and soft tissue properties. Therefore the weight values obtained from (3) do not necessary satisfy to the following conditions: $\Sigma\, w_i = 1$, $0 \le w_i \le 1$. To obtain consistent values we normalise and threshold the weights:

$$
\begin{aligned}
& w_1 = w_1 / (w_1 + w_2) \\
& \text{if } w_1 < 0 \text{ then } w_1 = 0 \\
& \text{if } w_1 > 1 \text{ then } w_1 = 1 \\
& w_2 = 1 - w_1
\end{aligned}
\tag{6}
$$

The weights are computed for each vertex of each 3D model generated from the sequence. To obtain a smooth distribution of weights the temporal averaging of the weights of each vertex is used (Figure 8).

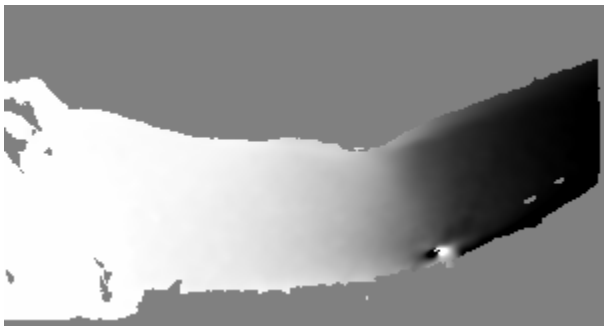


Figure 8. Weight distributed (a white value means $w = 1$ and a black one means $w = 0$).

Using the range flow we obtain the positions of the centre of each limb in all the 3D models of the sequence. The centre and the orientation of a global coordinate system are set in the parent bone region and the positions of the centre of the child bone region are then registered in that system. Then we analyze the motion of each point in its own local coordinate system and assign to each point of the 3D model a set of weights associated to each bone. Finally, once the skinning process is completed, the original 3D mesh can be animated in a realistic way.

3. Skill Building

3.1 Animation Skills and Physical Characteristics

A set of skills were built in character to synthesise motion at runtime depending on their environment, their task and their physical parameters, including motion skills, body sizes and weights and joint limits. Characters adapted skills at runtime to fit specific situations.

Each motion skill is a short sequence of motion capture data defining a single action: real actor motions are transferred and adapted to synthetic characters that might have different limb lengths or body masses. It is also possible to import and use new motion sequences created either with traditional animation software (3DS Max, Softimage, etc.) or with the tools provided with our system (real-time inverse kinematics and footstep control). Therefore authors have the opportunity to extend their library of movements. Transition motion capture data are defined to refine blending between main animations. For example, one could specify that characters should go from a running state to a loitering state through a walking state.

The real time simulation of character motion using physically based algorithm requires the complexity of skeletons to be reduced: for example all the fingers and the wrist are physically animated according to forearm motion. Simplifications can be adjusted to get the appropriate level of detail. Moreover limbs should be assigned bounding volumes (auto-generated from 3D mesh with manual adjustment if required), centres of mass and weights. Finally each joint has a type and angular limits: each join has up to three DOFs limited to believable postures.

3.2 Collision Avoidance Skills

When characters interact in a 3D environment, collisions are highly likely and must often be avoided. We developed a technique which automatically produces realistic collision-free animation of the upper arms. Our method is based on the latest models of collision avoidance provided by neuroscience [0] that allows realistic interpolation of keyframes in a few seconds. Our scheme was validated by comparing computer generated motions with motions performed by a sample of ten humans. These motions were defined by start and final postures and by an obstacle which had to be passed.

The automatic generation of collision-free paths is an active field of investigation which has been addressed in many different ways. For physically based animations collisions are detected and reactions are computed [0, 0]. In robotics many exact solutions exist to produce collision-free motions. However, since the complexity grows exponentially with the number of Degrees of Freedom (DOFs) their use is practically impossible on a simulated human [0]. Hence schemes that are not complete [0] (may fail to find a path when one exists) or probabilistically complete [0] have been developed which makes the task feasible though still time consuming. Other papers deal with achieving collision avoidance when articulated figures animated using inverse kinematics are reaching a goal [0, 0]. Collisions are detected using sensors and response vectors integrated into the inverse kinematics equation system. This process operates incrementally but does not ensure a coherent motion.

Since our application requires the generation of collision-free motions at interactive rates, the respect of the postures defined by the keyframes and the motion to be realistic we decided to explore another path, through the field of neuroscience.

Research by behavioural neuroscientists into the processes by which the central nervous system co-ordinates the large number of degrees of freedom of movement of multi-joint limbs started during the late 60s [0]. However the first papers dealing with obstacle avoidance in the neuroscience literature date from the early 80s, when it was established [0, 0] that path planning involves an intermediate point near the obstacle.

Sabes et al. [0] addressed the issue of how the intermediate point would be chosen. They suggested that properties of the limbs should be taken into account and they looked for a way of expressing the constraint of obstacle avoidance. Hence, they studied the sensitivity of the arm at the closest point of the trajectory; the sensitivity should be minimum with regard to uncertainty or perturbations in the direction of the obstacle. Their sensitivity model is based on the inertial

properties of the arm. The definition of sensitivity they used was proposed by Hogan [0], who expressed the mobility matrix of the arm in end-point coordinates, W( ) as:

$$W(\Theta) = J(\Theta)Y(\Theta)J^{t}(\Theta)$$
$$I^{-1}(\Theta) = Y(\Theta) \qquad (7)$$

where *Y( )* is the mobility matrix of the arm in actuator coordinates. *I( )* is the inertia matrix of the arm in actuator coordinates and *J( )* is the Jacobian.

Since the mobility matrix is symmetric it may be diagonalized by rotating the coordinate axes to coincide with its eigenvectors. It may be represented graphically by an ellipsoid as in Figure 9. The eigenvectors of W have a simple interpretation: the major (minor) eigenvector is the direction along which force perturbations have the largest (smallest) effect. Thus, the sensitivity model predicts that the near points should cluster toward a preferred axis which is the mobility minor axis.

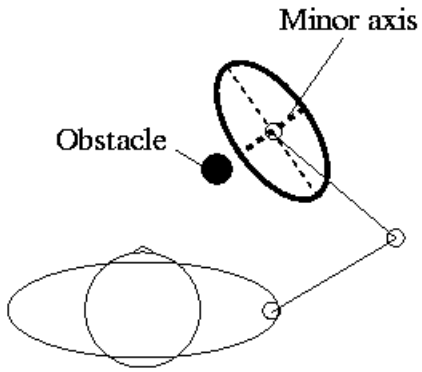


Figure 9. Mobility ellipse in the plane.

Based on Sabes' sensitivity model, our method allows realistic interpolation of keyframes in a few seconds. Keyframes can be created by an animator or selected from previous motion. Once keyframes have been specified, interpolations are achieved to produce the animation. The task we deal with is to offer an interpolation algorithm which generates collision-free motions.

The principle of our scheme is to first detect the objects that should be avoided. An interpolation between keyframes is performed using any classical inbetweening method such as cubic splines. Collisions are sorted and the dominant one is selected for a specific time step. This collision is corrected first. At this time step the frame is modified automatically to generate the intermediate keyframes using geometrical and mobility properties. Finally, this new keyframe is used for a new classical interpolation. This process continues until a collision-free motion is obtained.

This algorithm does not make any hypothesis about the kind of objects which compose the virtual character, only a rigid skeleton is needed. So regardless of the way the articulated figure is defined, our scheme can be used to generate animations without collisions for any collision detection algorithm.

Experimental results comparing a human model with real motions validated our algorithm showing that the generated motions are realistic, simulating the “cautious” (choice of around 30% of participants) way of avoiding obstacles. A more complete description of the process and its validation can be found in the following published paper [0].

4. Interaction Between a V-Man and Its Environment

We developed an innovative technology that enables V-Men to synthesise motion at runtime depending on their environment, their task and their physical parameters. Given a high-level task like “walk to that spot” or “get that object”, a V-Man generates the complete animation required to accomplish the task. A V-Man is able to walk on any kind of terrain, to go upstairs, downstairs, to calculate paths in order to avoid obstacles, and to adapt his movements and actions to his environment. In doing so, the character draws upon its unique set of skills, which are created during the character creation. A V-Man throwing a ball in a pile of cans will generate a realistic animation of the cans. All these interactions take into account the physical parameters of the V-Man. The characters adapts skills at runtime to fit specific situations—for example, the same “grasp” skill can be used to pick up a wide variety of objects, in a multitude of different locations.

The system allows smooth blending between animation sequences and the simulation. The characters can therefore be programmed to take advantage of animation sequences for complex movements while at the same time being sensitive to the physical surroundings in a truly open, interactive way. The overall effect blends the rich, artist-created character of traditional animation with the realism and emergent behaviour of a simulation. The transition between movements required the integration of motion blending algorithms, animation sampling methods and real-time physical simulation of the body.

Path planning, which is of paramount importance for character autonomy [0], was implemented from Luga’s algorithm [0, 0]. This path planner allows a virtual character to autonomously compute, in real time, collision free paths respecting its movement constraints as well as its areas of interest. This algorithm, based on genetic algorithms, is extremely innovative and has been acknowledged as a major step forward by the scientific community.

A V-Man understands high-level commands thanks to a declarative control system. Declarative control is a recent Artificial Intelligence technique that allows the interpretation of high-level multi-modal commands such as “put that there” or “grab this”, where the commands are expressed through out a voice control system while the deictics (“that”, “there”, “this”) are defined by mouse clicks. This technique combines natural language interpretation with a constraint solver [0]. The simplest level of interaction will enable the user to simply define the path for the character to walk along. Beyond that, the characters are provided, through V-Man declarative multi-modal dialogue engine, with declarative commands such as “watch that character”, “follow that character”, or “go there”.

5. Results

Here we present three animations created by our V-Man system to demonstrate the key aspects on interactive animation authoring, physical animation and path planning.

5.1 Demo of interactive Animation Authoring

A female character in the demo picked the white ball on the floor and threw it. A male character walked to a chair to sit down and then walked near the shelf to dance (Figure 10). High level commands such as “pick white ball”, “sit on blue chair”, and “dance” were issued through a keyboard or a microphone. The characters completed the tasks automatically.

Figure 10. Demo of interactive authoring.

5.2 Demo of Physical Animation

In the demo (Figure 11), a man attempted to jump over a wood bar. He collided with the bar and fell. The jump was controlled by motion capture data. When the man collided with the bar, the physical animation started functioning to animate action after the collision. Animations between jump and falling were blended.

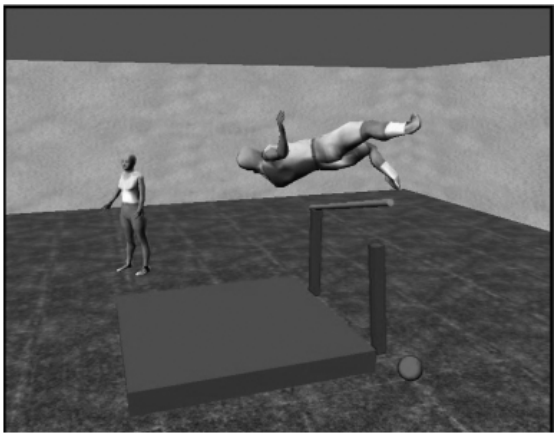


Figure 11. Physical animation and animation blending.

5.3 Demo of Fire Situation

Initially in the demo, people were walking around the building or standing at some positions. After fire alarm (external trigged), each person walked toward his closest exit. Path of individual was planned to avoid obstacles and other people (Figure 12).

Figure 12. Fire in a building.

6. Conclusions

In the paper, we presented our V-man system aimed to create, animate, control and interact in real-time with as minimum user intervention as possible. The key to achieve minimum intervention is the automated creation of V-Man characters. Taking advantages of static and dynamic 3D human body scanner, conformation converted scanned data to an animatable model with an accurate human body and garment shapes; range flows were analyzed for skinning that made surface deformation realistic. In the end, we gave demos to illustration key aspects of our system.

Acknowledgements

We gratefully acknowledge the European Union's Framework 5 IST programme in funding this work. We also want to thank our EU partners who contributed to this work: CS-SI, CSTB, HD Thames, MathEngine, and Sail Labs

References

1. Hilton A., Beresford D., Gentils T., Smith R. and Sun W., “Virtual People: Capturing human models to populate virtual worlds”, CA'99 Conference, 26-29 May 1999, Geneva, Switzerland.
2. D'Apuzzo N., Plänkers R., Gruen A., Fua P. and Thalmann D. “Modeling Human Bodies from Video Sequences”, Proc. Electronic Imaging, San Jose, California, January 1999.
3. Sederberg T. W. and Parry S. R.. “Free Form Deformation of Solid Geometric Models”, Proc. SIGGRAPH'86, pp 151-160.
4. Terzopoulos D. and Metaxas D., “Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics”, IEEE Transactions on pattern analysis and machine intelligence, Vol. 13, No. 7, 1991.
5. Thalmann N. M. and Thalmann D. “State-of-the-Art in Computer Animation”, ACMCS96, Switzerland, 1996.
6. Thalmann D., Shen J. and Chauvineau E., “Fast Realistic Human Body Deformations for Animation and VR Applications”, Proc. Computer Graphics International, 1996.
7. Nurre J. H. “Locating landmarks on human body scan data”. International conference of recent advances 3D digital imaging and modelling, pp 289-295, 1997, IEEE NJ, USA
8. Ju, X., Werghi, N. and Siebert, J. P., “Automatic Segmentation of 3D Human Body Scans”, IASTED International Conference on Computer Graphics and Imaging 2000 (CGIM 2000), 19-23 Nov. 2000.
9. Daly C. H. “Biomechanical properties of dermis”, the journal of investigative dermatology, Vol. 79, pp 17-20, 1982.
10. Lanir Y. “Skin mechanics”, Chapter 11, in Handbook of Bioengineering, McGraw-Hill, USA, 1987.
11. Behnke A. R. and Wilmore J. H. “Evaluation and regulation of body build and composition”, Prentice-Hall, USA, 1974.
12. measurement of subcutaneous fatness”, Am J Clinical Nutrition, Vol. 32, pp 1734-1740, 1979.
13. Lean M. EJ, Han T. S. and Deurenberg P. “Predicting body composition by densitometry from simple anthropometric measurements”, Am J Clinical Nutrition, Vol. 63, pp 4-14, 1996
14. Han T. S., McNeill G., Seidell J. C. and Lean M. EJ “Predicting intra-abdominal fatness from anthropometric measures: the influence of stature”, Int. J. of Obesity, Vol. 21, 1997
15. Siebert J. P. and Marshall S. J. “Human body 3D imaging by speckle texture projection photogrammetry”, Sensor Review, 20 (3), pp 218-226, 2000.
16. Vareille G. “Full body 3D digitizer”, International Conference of Numerisation 3D - Scanning 2000, 24-25 May 2000, Paris, France.
17. Vasilev T. “Dressing Virtual People”, SCI'2000 conference, Orlando, July 23-26, 2000.
18. Volino P., Thalmann N. M., “Comparing Efficiency of Integration Methods for Cloth Animation”, Proceedings of CGI’01, Hong-Kong, July 2001
19. Luga H., Panatier C., Torguet P., Duthen Y. and Balet O. “Collective Behaviour and Adaptive Interaction in a Distributed Virtual Reality System”, Proceedings of ROMAN’98, IEEE International Conference on Robot and Human Communication, 1998.
20. Ju X and Siebert J. P. “Individualising Human Animation Models”, Proc. Eurographics 2001, Manchester, UK, 2001.
21. Brown J., Sorkin S., Bruyns C., Latombe J. C., Montgomery K. and Stephanides M. “Real-Time Simulation of Deformable Objects: Tools and Application”. Computer Animation, Seoul, Korea, November 2001.

22. Vassilev T., Spanlang B. and Chrysanthou Y. “Efficient Cloth Model and Collision Detection for Dressing Virtual People”, (in CD proc. GeTech Hong Kong), January 2001.
23. Sabes P. N. and Jordan M. I. and Wolpert D. M. “The role of inertial sensitivity in motor planning”, Journal of neuroscience, 1998, 18 (15), 5948-5957.
24. Volino P. and Thalmann N. M. and Shi J. and Thalmann D. “Collision and Self-Collision Detection: Robust and Efficient Techniques for Highly Deformable Surfaces”, Eurographics Workshop on Animation and Simulation, 1995.
25. Bandi S. and Thalmann D. “An Adaptive Spatial Subdivision of the Object Space for Fast Collision of Animated Rigid Bodies”, Eurographics'95, 259-270, 1995.
26. Canny J.F. The complexity of robot motion planning, MIT Press, 1988.
27. Koga Y., Kongo K., Kuffner J. and Latombe J.-C. “Planning motions with intensions”, SIGGRAPH'94, 395-408, 1994.
28. Barraquand J. and Kavraki L. and Latombe J-C. and Li T-Y. and Motwani R. and Raghavan P. “A random sampling scheme for path planning”, Journal of robotics research, 1997, 16 (6), 759-774
29. Zhao J. and Badler N. I. “Interactive body awareness”, Computer-aided design, 1994, 26 (12), 861-867.
30. Huang Z. “Motion control for human animation”, PhD thesis from EPFL-DI-LIG, 1996
31. Bernstein N. The co-ordination and regulation of movements, Oxford: Pergamon Press, 1967
32. Abend W. and Bizzi E. and Morasso P. “Human arm trajectory formation”, Brain, 1982, 105, 331-348.
33. Flash T. and Hogan N. “The coordination of arm movements: an experimentally confirmed mathematical model” , The journal of neuroscience, 1985, 5 (7), 1688-1703
34. Hogan N. “The mechanics of multi-joint posture and movement control”, Biological cybernetics, 1985, 52, 315-331.
35. Nebel J.-C. “Realistic collision avoidance of upper limbs based on neuroscience models”, Computer Graphics Forum, 2000, volume 19(3).
36. Luga H., Balet O., Duthen Y. and Caubet R. “Interacting With Articulated Figures Within The PROVIS Project”, Proceedings of the 11th ACM International Conference on Artificial Intelligence & Expert Systems (AIE), Published in Springer Lecture Notes in Artificial Intelligence, 1998.
37. Panatier C., Sanza C.and Duthen Y. ”Adaptive Entity thanks to Behavioral Prediction”, From Animal to Animates, MIT Press, 2000.
38. Kwaiter G., Gaildrat V., Caubet R., “Controlling objects natural behaviors with a 3D declarative modeller”, Computer Graphics International, CGI'98 Hanover, Germany, 24-26 Jun 1998.