Example-based Pose Space Parameterization of Skin Deformation

Jong-Hyuk Kim* Ajou University
Jung-Ju Choi† Ajou University
Christoph M. Hoffmann‡ Purdue University

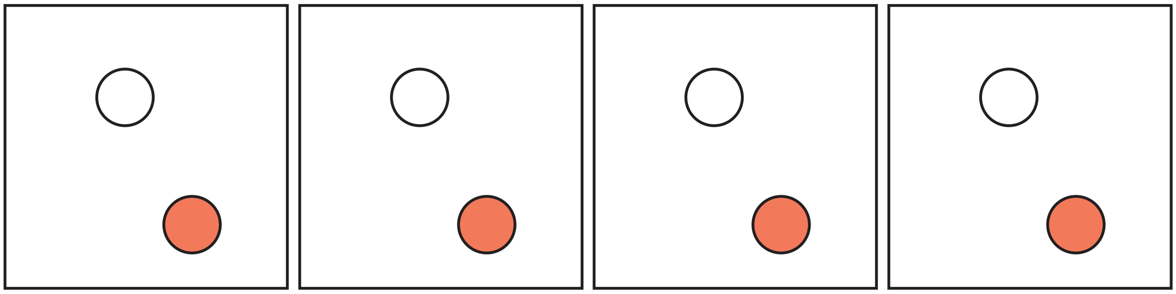


Figure 1: *Skin deformation and retargeting*

Abstract

In this paper, we present a new method to parameterize skin deformation by skeletal posture. Given an example animation set of skin deformation, we extract signals over time by ICA(Independent Component Analysis) and approximate it using RBF(Radial Basis Function) parameterized by joint angles(skeletal posture). During the extraction step, the size of animation data are greatly reduced into a set of signals which plays a role in virtual muscles. Furthermore, during the parameterization step using RBF, we can get a simple and easy control of skin deformation. For given a user input of skeletal posture, we can find its corresponding set of signals by RBF and then its amount of skin deformation by a simple matrix multiplication quickly in a reduced dimension. Since skin deformation is represented by multiplications of matrices, we can transfer the deformation into another skeletal and skin structure simply without considering the input dimension of the deformation data. In these days, there are many virtual characters in interactive environments, games for example. The method in this paper can give an easy interactive control not only for a realistic looking character but also for its variants. According to our experiments, control and retarget take place in real time. We show qualitative and quantitative performance results for the control and retargeting.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: skin deformation, parameterization of motion, independent component analysis

1 Introduction

Representing realistic skin deformation enhances the realism of virtual characters which are shown frequently in interactive environments, games for example. Skin deformation, in general, has high degree of freedoms since it is influenced complicatedly by the motion of underlying structures such as skeletons and muscles. The motion of a point on skin can be represented nonlinearly by several muscles and skeletons and considered as a multiple coupled system of them. Those are the reasons that we have encountered the difficulties to achieve realistic skin deformation.

Traditionally, well-trained artists spend a large amount of time and effort to set up a realistic-looking character. Motion capture approaches[Park and Hodgins 2006; Park and Hodgins 2008] with a large number of markers on skin have greatly reduced such an amount of effort and physical simulations for realistic muscles[Cordier and Magnenat-Thalmann 2000] have generated a realistic motion of skin deformation. Those methods require a large amount of storage space or execution times and lack of controls for skin deformation. Therefore, they have limitations to be applied in an interactive environment that requires particularly realistic looking characters such as games. In such an environment, applications generally require fast execution with small amount of data and easy control method while generating a reasonable amount of realism. It is clear that these three kinds of requirements cannot be achieved easily and have to be trade-off. In this paper, for given an animation of skin deformation, we present a new method to parameterize the deformation by skeletal posture in order to represent the deformation with small amount data and give an easy and fast control.

We assume that the animation data of skin deformation are represented by a sequence of motions of points on skin over time, which can be obtained either by motion capture devices or by physical simulations for muscles with skinning. First, we extract independent signals from the input data, which play role in virtual muscles using ICA(Independent Component Analysis). The number of data can be greatly reduced since the number of signals are much smaller than the number of time frames of animation data. Each signal at a certain time instance is represented by a point on a curve with respect to a time as well as by a point on a surface with respect to its corresponding joint angles. Once we transform the parameter space from a time into joint angles and approximate the surface using RBF(Radial Basis Function) with respect to joint angles, we can parameterize the signals by their corresponding skeletal posture. This method gives a fast control of skin deformation by simply specifying joint angles to generate skin deformation using RBF evaluation and matrix multiplications.

*e-mail:naidem@ajou.ac.kr
†e-mail: jungju@ajou.ac.kr
‡e-mail:cmh@purdue.edu

2 Related Work

- Skeletal subspace deformation [Magnenat-Thalmann et al. 1988]
- Pose space deformation [Lewis et al. 2000]
- Motion capture [Park and Hodgins 2006; Park and Hodgins 2008]
- Physical simulation [Cordier and Magnenat-Thalmann 2000].

For mathematical tools,

- ICA [Hyvärinen and Oja 2000]
- RBF [Buhmann 2003]

3 Parameterization of Skin Deformation

We assume that the example animation data of skin deformation are represented by a sequence of motions of points on skin over time, $\mathcal{P}(t)$ and their corresponding joint angles $\Theta(t)$.

3.1 Eliminating Rigid Motion of Skin Deformation

The skin deformation data $\mathcal{P}(t)$ are influenced by their underlying skeletal and muscular structures and can be partitioned into a rigid body motion $\mathcal{R}(t)$ and a non-rigid body motion $\mathcal{X}(t)$.

$$\mathcal{P}(t) = \mathcal{R}(t) + \mathcal{X}(t)$$

In this paper, we handle the nonlinear motion of the deformation $\mathcal{X}(t)$, but not the linear rigid motion $\mathcal{R}(t)$. In order to remove the rigid motion, we represent the deformation in a local coordinate system, $P(t)$, defined by its corresponding skeletal subspace such that,

$$P(t) = W^{-1}\mathcal{P}(t) = W^{-1}\mathcal{R}(t) + W^{-1}\mathcal{X}(t) = R(t) + X(t),$$

where W is a transform matrix from a local to a world coordinates and $R(t)$ and $X(t)$ are the rigid and non-rigid motions in a local coordinate system, respectively. Since $R(t)$ are rotation and translation invariant over time, $R(t) = R(0) = P(0)$ if we assume that there is no non-rigid motion at the beginning of motions without loss of generality. The non-rigid body motion can be defined by $X(t) = P(t) - P(0)$. Remaining of this paper, we consider only the non-rigid deformation $X(t)$ which can be described complicatedly as a combination of skeletal and muscular motions.

During the elimination of rigid motion, we are implicitly required to partition a set of points on skin into sets of points, each of which is defined by its corresponding skeleton. For each point, we find a nearest bone then simply transform its coordinate into the bone's local coordinate (See Figure 2).

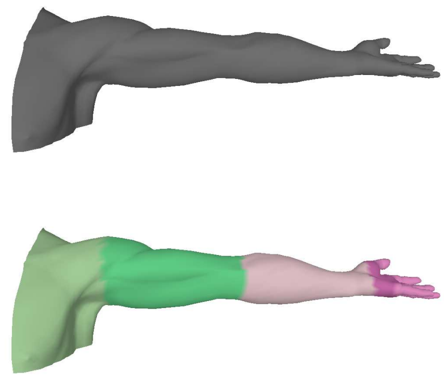


Figure 2: *Partition of points on skin by transforming their local coordinate systems in order to eliminate rigid motion of skin deformation.*

3.2 Extracting Virtual Muscle Signals

Without considering the non-rigid motion $X(t)$ of skin deformation, it is not probable to represent the delicate, complex motion of skin deformation realistically. We consider that the non-rigid motion of skin deformation is influenced by the motion of a certain underlying structure which is triggered by skeletal motion. Anatomically the main underlying structure between skeletal bones and skin is muscles. In this paper, we take an ICA to extract the virtual muscular components and to analyze the input skin deformation. ICA is a mathematical method to separate multivariate signals into additive components assuming that they are *statistically independent* [Hyvärinen and Oja 2000]. In this paper, we consider the motion of skin deformation(signals) are multiply coupled to motions of muscles, each of which is independent to each other muscles.

Assuming that a given observed signal at time t be represented by a column vector $\mathbf{x}(t) = (x_1(t), \ldots, x_n(t))^T$ and its source signal by $\mathbf{s}(t) = (s_1(t), \ldots, s_k(t))^T$, ICA can represent $\mathbf{x}(t)$ as

$$x_i(t) = \sum_{j=1}^{k} a_{i,j} s_j(t) \tag{1}$$

For a set of n points on skin, in this paper, an input deformation data over m frames can be represented by a matrix equation using ICA such that

$$\begin{aligned} \mathbf{x}(t) &= \mathbf{A}\mathbf{s}(t) \\ \mathbf{X} &= \mathbf{A}\mathbf{S}, \end{aligned} \tag{2}$$

where $\mathbf{X} = [x_{i,j}]_{n\times m}$, $\mathbf{A} = [a_{i,j}]_{n\times k}$, and $\mathbf{S} = [s_{i,j}]_{k\times m}$. The $[\cdot]_{n\times m}$ represents a matrix whose dimension is $n \times m$. In general, the skin deformation data are represented by a motion of several thousands of points on skin and several hundreds of frames while the number of source signals is within several tens according to our experimental result; that is, $k < m < n$. During extraction of virtual muscle signals, we can greatly reduced the number of data from nm to $nk + km$.

3.3 Parameterization of Virtual Muscle

Input data of skin deformation described with respect to time are represented by a simple multiplication of matrices using ICA. In this section, we present a method to parameterize the skin deformation with respect to skeletal motion which has been a simplest but most powerful control for character animation.

For a set of points on skin, we specify a set of bones that are influencing the deformation. It is well known that there are at most four bones influencing the deformation of a point on skin [Lander 2001]. This parameterization problem can be solved by finding a function between the source signal S in Equation (2) and its corresponding joint angle $\Theta = \left(\vec{\theta}_1(t), \ldots, \vec{\theta}_l(t)\right)^T = [\theta_{i,j}]_{l\times m}$, where l stands for the number of joints. An ordered pair of a source signal and a joint angle $(s(t), \vec{\theta}(t))$ is represented by a point in $(m + l)$-dimensional space. Once those points for all t are projected onto a l-dimensional space(skeletal space), they form points on an m-dimensional graph.

We assume that the graph are smooth and continuous over the l-dimensional skeletal space. In this paper, we use an RBF to approximate and finally to parameterize the m-dimensional graph with respect l-dimensional parameters such that

$$\mathbf{s}_j(\vec{\theta}) = \sum_{i=1}^{m} w_{i,j}\phi\left(\|\vec{\theta} - \vec{\theta}_i\|\right). \tag{3}$$

We use a quaternion to represent the rotation angle of a joint θ and a thin plate function $\phi(r) = r^2\log(r)$ for the kernel function of RBF. The weights $w_{i,j}$ can be represented by the matrix $W = [w_{i,j}]_{m\times k}$, which can be computed by a singular value decomposition method for a standard linear least square problem. Once we obtain $\mathbf{s}_j$ parameterized by $\vec{\theta}$ in Equation (3), we can represent motions of vertices on skin with respect to joint angles by a simple substitution such that

$$\mathbf{x}(\vec{\theta}) = \mathbf{A}\mathbf{s}(\vec{\theta}). \tag{4}$$

There has been done several research to parameterize motions of vertices on skin by approximating $\mathbf{X}$ with respect to Θ, which has to be taken place directly in higher $n \times m$-dimensional parameter space while ours indirectly in lower $m \times k$-dimensional space. Even though the approximation problems are recognized as underdetermined linear system, the approximation in low dimensional space can avoid over fitting problem and hence it generates a smooth surface with respect to parameters.

4 Skin Deformation Retargeting

One of the contributions of the method in this paper is to provide a simple way of mathematical analysis for skin deformation. Given two or more sets of skin deformation data, we can compare one to another and transfer the difference between two into another by simple matrix computations. In this section, we show a simple example to transfer a difference between two skin deformation data into another deformation (See Figure 3).

$$\begin{tikzcd}
X_1 = A_1S_1 \arrow[r, "T=A_2S_2S_1^+A_1^+"] & X_2 = A_2S_2 \arrow[d, "U^+"] \\
Y_1 = B_1R_1 \arrow[u, "U=A_1S_1R_1^+B_1^+"] \arrow[r, dashed, "T'"'] & Y_2 = B_2R_2
\end{tikzcd}$$

Figure 3: *Overview of skin deformation retargeting: Finding the skin deformation Y_2 from Y_1 which reflects the difference between X_1 and X_2.*

Assume that two sets of skin deformation X_1 and X_2 with the same skeletal structures but different muscular structures, skinny and muscular characters' deformation data with the same bone lengths for example, and a set of the deformation Y_1 with a different structure are given. The skin deformation retargeting problem can be described as finding the deformation Y_2 from Y_1 which reflects the difference between X_1 and X_2.

Let the matrix $T : X_1 \to X_2$ be the transformation from X_1 to X_2, then

$$\begin{aligned} X_2 &= TX_1 \\ A_2S_2 &= TA_1S_1 \\ T &= A_2S_2S_1^+A_1^+, \end{aligned} \tag{5}$$

where M^+ stands for the Moore-Penrose pseudo inverse matrix of M. Let n_1 and n_2 be the numbers of vertices, k_1 and k_2 be the numbers of independent signals, and m_1 and m_2 be the numbers of frames in the deformation data X_1 and X_2, respectively. The transformation matrix T can be obtained only if $k_1 = k_2$ and $m_1 = m_2$ without concerning the number of vertices n_1 and n_2. It is easy to make $k_1 = k_2$ by specifying the same number of independent signals as an input of ICA for X_1 and X_2. Once X_1 and X_2 have the same numbers of signals, matrices S_1 and S_2 have the same numbers of rows. Furthermore, S_1 and S_2 are parameterized by joint angles θ as in Equation (3), we can always generate the same number of columns m in S_1 and $S2$ by specifying m sequences of joint angles such that $m = m_1 = m_2$.

Let the matrix $U : Y_1 \to X_1$ be the transformation from Y_1 to X_1, then

$$\begin{aligned} X_1 &= UY_1 \\ A_1S_1 &= UB_1R_1 \\ U &= A_1S_1R_1^+B_1^+. \end{aligned} \tag{6}$$

Let n_1', k_1', and m_1' be the numbers of vertices, independent signals, and frames in Y_1, respectively. Without concerning n_1 and n_1', the Equation (6) can be also obtained only if $k_1 = k_1'$ and $m_1 = m_1'$, which can be achieved as easily as describe in the Equation (5).

Finally, we can obtain the desired deformation Y_2 such that

$$\begin{aligned} Y_2 &= T'Y_1 \\ &= U^+TUY_1 \\ &= B_1R_1S_1^+A_1^+A_2S_2S_1^+A_1^+A_1S_1R_1^+B_1^+B_1R_1 \\ &= U^+A_2S_2. \end{aligned} \tag{7}$$

The only required condition to obtain the deformation data Y_2 using the Equation (7) is $n_1 = n_2$. Without loss of generality, we assume that $d_n = n_1 - n_2 > 0$. Putting d_n zero row vectors into the bottom of A_2 satisfies the required condition $n_1 = n_2$ while it does not destroy any necessary conditions for computing ICA, pseudo inverse, and so on.

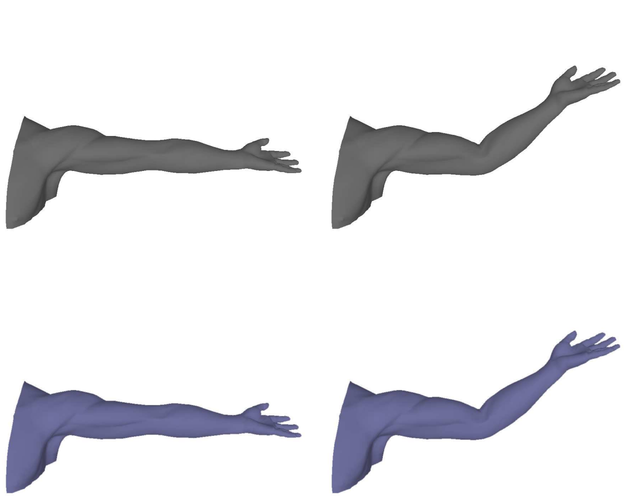
JH and Prof. Hoffmann, I'm not quite sure whether the above statement is true. We have to evaluate it by experiments.

The primary advantage of the method proposed in this section is that we do not have to care of the numbers of vertices, frames, and independent signals. They are easily achievable as described so far.

5 Experimental Result

In this section, we show several quantitive and qualitative experimental result for skin deformation. For a given skin model shown in Figure 2, the deformation data are generated by bending elbow and twisting wrist. The numbers of vertices on skin, frames, and computed source signals by ICA are 3,234, 100, and 7, respectively. ICA can generate the signals within *xxx* sec and the RMSE(Root Mean Squared Error) is within 0.1 mm compared with the original data (See Figure 4). The number of storage required to represent the skin deformation is reduced from $3 \times 3,234 \times 100$ to $3 \times 3,234 \times 7 + 7 \times 100$ which is approximately 7.1% with respect to the size of original data.

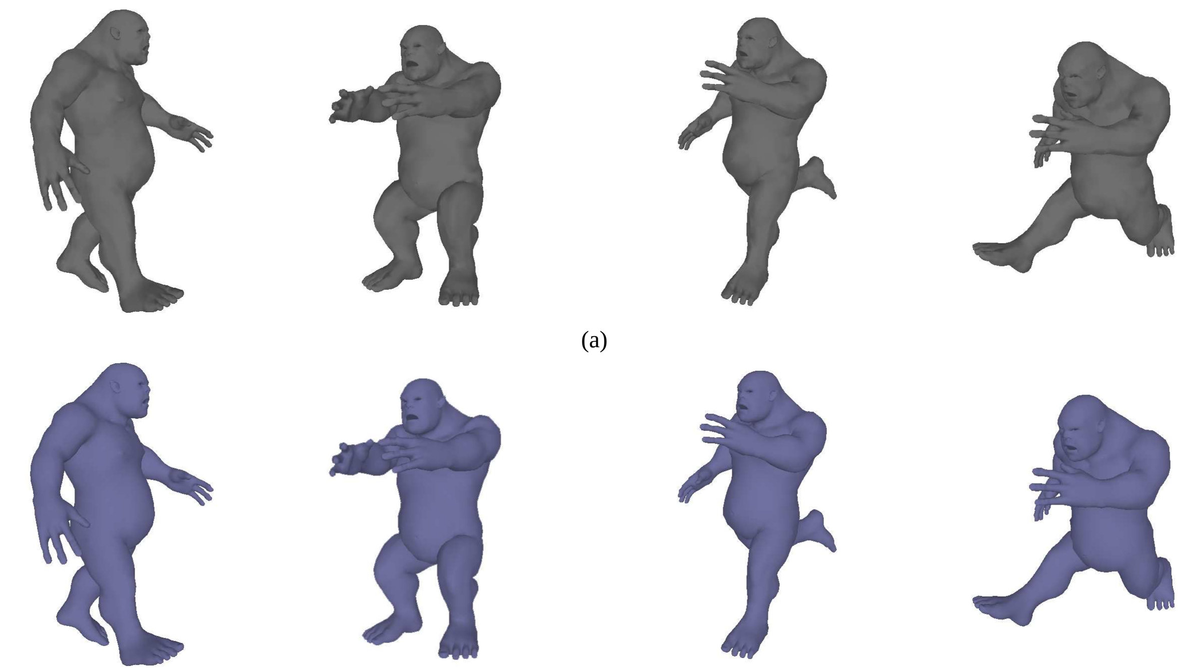
For a virtual character shown in Figure 5 with *xxx* vertices and *yyy* frames, ICA can generate *zzz* independent signals within *www*



(a)

(b)

Figure 4: *(a) Original input deformation data X and (b) their corresponding reconstructed data using ICA.* Should be replaced with high quality data. Prepare the files, JH.



(a)

(b)

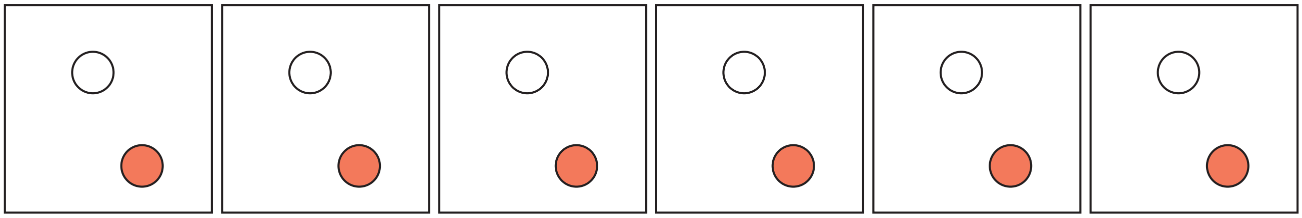
Figure 6: *(a) Original input deformation data X and (b) their corresponding reconstructed data using ICA.* Should be replaced with high quality data. Prepare the files, JH.

RMSE and their reconstructed skin deformations are shown in Figure 6.

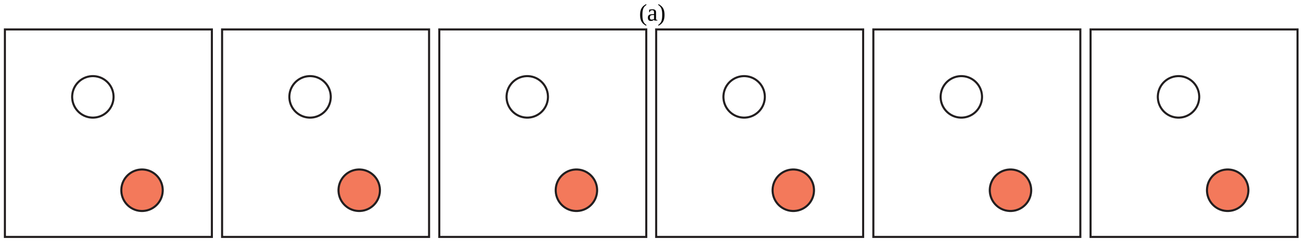
For our skin deformation retargeting, we generate two character models (a man and a woman) and generate two motions (one for skinny and other for muscular). Let the skin deformation data for a skinny man be X_1, the deformation for a muscular man be X_2, and the deformation for a skinny woman be Y_1, we compute the skin deformation for a muscular woman as describe in Section 4 using Equation (7). The result is shown in Figure 7.

6 Discussion and Future Work

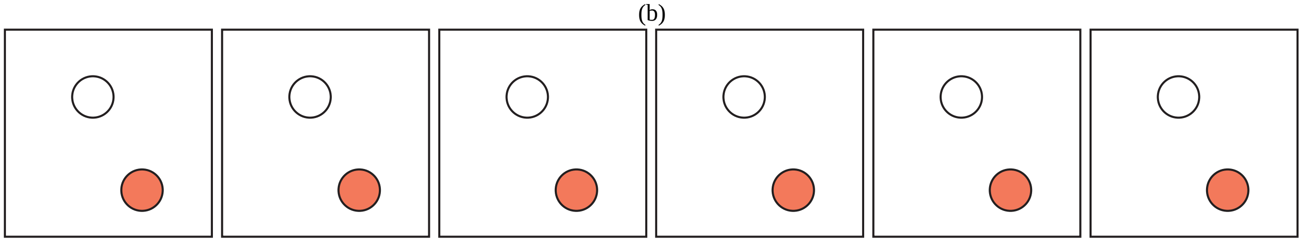
In this paper, we presented a method to parameterize the skin deformation with respect to skeletal motion using ICA and RBF. Example skin deformation data obtained using motion capture and/or physical simulation are represented by a multiplication of matrices using ICA, one of the matrices represents source signals for virtual muscles. During this step, the number of required storage to represent the deformation is greatly reduced with small amount of errors. The signals with respect to time are parameterized by joint angles of skeletal structure using RBF, which gives a simple but



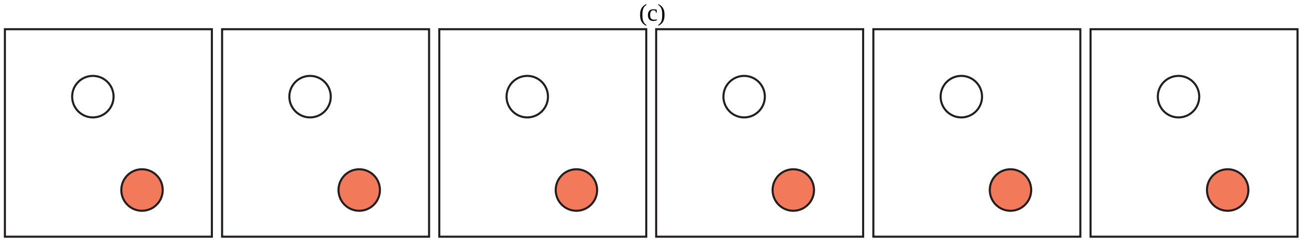
(a)



(b)



(c)



(d)

Figure 7: *Retargeting result of skin deformation. (a) The deformation data for a skinny man, (b) the deformation data for a muscular man, (c) the deformation data for a skinny woman, and (d) the computed deformation data for a muscular woman using skin deformation retargeting.* Should be replaced with high quality data. Prepare the files, JH.

Figure 5: *Segmentation of a virtual character*

powerful control of the skin deformation. The primary contribution of this paper is providing the skin deformation data with a mathematical analysis way. Given multiple sets of skin deformation, we formulate the difference between the deformation and transfer the difference to another deformation data using simple matrix computations.

Since we concerned the deformation as a motion triggered by the motion of underlying structure between skin and skeleton, the signals computed so far are considered as muscles. In conventional animation softwares, the skin deformation can be controlled using simple parameters of muscles, bulging rate for example. However, to alter a single signal is not directly meaningful to change the shape of local skin deformation. Hence, we cannot expect the local change only with a single signal rather with a combination of signals which is turned out to be extremely difficult.

Acknowledgements

References

BUHMANN, M. D. 2003. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press.

CORDIER, F., AND MAGNENAT-THALMANN, N. 2000. Integrated system for skin deformation. In *CA '00: Proceedings of the Computer Animation*, IEEE Computer Society, Washington, DC, USA, 2.

HYVÄRINEN, A., AND OJA, E. 2000. Independent component analysis: Algorithm and applications. *Neural Network 13*, 4-5, 411–430.

LANDER, J. 2001. A heaping pile of pirate booty. *Game Developer 8*, 4 (April), 22–30.

LEWIS, J. P., CORDNER, M., AND FONG, N. 2000. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 165–172.

MAGNENAT-THALMANN, N., LAPERRIERE, R., AND THALMANN, D. 1988. Joint-dependent local deformations for hand animation and object grasping. In *Graphics Interface*, 26–33.

PARK, S. I., AND HODGINS, J. K. 2006. Capturing and animating skin deformation in human motion. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, ACM, New York, NY, USA, 881–889.

PARK, S. I., AND HODGINS, J. K. 2008. Data-driven modeling of skin and muscle deformation. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, ACM, New York, NY, USA, 1–6.