

# Statistical Analysis of Parameters for Torque-Integrated Skinning

YoungBeom Kim, Byoung-Ha Park, and Kwang-Mo Jung  
Korea Electronics Technology Institute, Seoul, Korea  
Email: {ybkim, bhpark, jungkm}@keti.re.kr

**Abstract**—We propose the statistical analysis of parameters for example-based torque-integrated skinning methods. Recently, the physically-based linear blend skinning is proposed which is in the category of the example-based torque-integrated system. The physically-based linear blend skinning has linear scale/shear blending function and linear rotation blending function. In the current study, the degree of the polynomial blending functions are extended to three, i.e., it is either linear, quadratic, or cubic. We use the arm and leg model as experimental models. The evaluation results show that scale/shear blending function needs to be quadratic in the case of the leg.

**Index Terms**—real-time animation, character animation, skinning, torque

## I. INTRODUCTION

Human movement is produced by muscles that generate torques on skeletal joints. In elbow flexion, for example, the individual fibers of the biceps muscle contract, generating torque on the elbow. Suppose that a dumbbell is to be lifted for weight training. Then, large torque needs to be exerted on the elbow. In general, the muscle bulges more heavily as the required torque becomes larger. See Fig. 1.



Figure 1. Stills of video capturing muscle bulges.

Geometric skinning works with a mesh, a skeleton, and skinning weights for every vertex of the mesh. The traditional skinning methods such as the linear blend skinning method and the dual quaternion skinning method is prevailing geometric skinning technique for the real-time applications. However, the methods have artifacts and do not take the joint torques into account. Recently, the artifacts are refined by many researches such as refinement based methods [1], [2]. The physically-based linear blend skinning [3] is proposed to integrate joint torque which requires blend functions for blending of scale/shear and rotation parameters.

In the current study, we extend the degree of the polynomial blending functions of the physically-based linear blend skinning. The extension include quadratic and cubic. With this extension, total nine conditions are evaluated and it is evaluated with arm and leg models.

## II. RELATED WORK

Physically-based and anatomy-based approaches represent the logical approaches to character animation. They simulate the internal structure of a body composed of bones, muscles, and fat tissues [4]. Although successful results are produced, they are not relevant to real-time applications because of the computational overhead. Commercial packages, such as [5], utilizing the anatomy-based approach can craft realistic character meshes in modeling stage as well.

Linearly blending rotations leads to well-known artifacts such as joint collapse and candy wrapper effect [6], [7]. There have been many techniques that address these problems. For example, variations of linear blend skinning have been proposed that train more blend weights per vertex per joint [8], [9]. While these models are more expressive, they also require more training data to prevent over-fitting.

Example-based skinning also resolves the artifacts of linear blend skinning. It addresses the problems by using multiple input meshes. In addition, it often adds effects such as muscle bulging. Example-based methods use either direct interpolation between example meshes [6], [10], approximation by principal components of example deformations [11], or fitting of the linear blending parameters to match the provided examples [7]. Recently, example-based skinning methods have been improved by using rotational regression [12].

We extend the degree of the polynomial blending functions of the physically-based linear blend skinning [3].

## III. THE BLENDING FUNCTIONS

The blending functions of the physically-based linear blend skinning method are determined at the preprocessing stage using a set of skeleton-mesh-torque examples  $\{(\mathbf{q}_i, \mathbf{y}_i, \tau_i)\}$ . See Fig. 2.

The rotation, scaling, and shearing of a triangle in the animated pose are often described as *deformation gradient* with respect to the triangle of the rest pose, and can be encoded in a 3x3 matrix [13]. In the system

proposed by Wang *et al.* [12], the deformation gradients, which they denote by  $D_s$ , are predicted given the skeleton  $\mathbf{q}$  of an animated pose, and then the deformed mesh  $\mathbf{y}(\mathbf{q})$  is reconstructed using  $D_s$ . For this purpose, deformation gradient predictors, which they denote by  $D(\mathbf{q})$ , are trained using a set of skeleton and deformation gradient pairs  $\{(\mathbf{q}_i; D_i)\}$ .

Physically-based linear blend skinning method extends the framework of Wang *et al.* [12] and uses two deformation gradient predictors. One is trained using the examples generated with the minimum torque. It is denoted by  $D_{min}(\mathbf{q})$ . The other is trained using those with the maximum torque and is denoted by  $D_{max}(\mathbf{q})$ .

First of All  $D_{min}$  and  $D_{max}$  are computed by feeding  $\mathbf{q}_i$  to  $D_{min}(\mathbf{q})$  and  $D_{max}(\mathbf{q})$ , respectively. The  $D_{min}$  is the deformation gradients of the minimum torque example. Similarly,  $D_{max}$  is the deformation gradients of the maximum torque example.  $D_{min}$  is decomposed into a rotation component and a scale/shear component. So is  $D_{max}$ . The method represents the rotation components in quaternions. The quaternions are denoted by  $\rho_{min}$  and  $\rho_{max}$ .

On the other hand, the deformation gradient  $D_i$  is computed from the example mesh  $\mathbf{y}_i$ , as illustrated at the right-hand side of Fig. 2. A rotation component is

extracted from  $D_i$  and is converted into a quaternion  $\rho_i$ . We should be able to obtain  $\rho_i$  by blending  $\rho_{min}$  and  $\rho_{max}$ . The blending weight, which is denoted by  $w_r$ , is calculated by solve a minimization problem.

Let  $S_{min}$ ,  $S_{max}$ , and  $S_i$  denote the scale/shear matrices extracted from the deformation gradients  $D_{min}$ ,  $D_{max}$ , and  $D_i$ , respectively. The blending weight for scaling/shearing, which is denoted by  $w_s$ , is given by solve a minimization problem.

The physically-based linear blend skinning method maintains independent blending functions for rotation and scaling/shearing. They are denoted by  $f_r$  and  $f_s$ . In the torque-integrated skinning method, they are functions of torque  $\tau$ . Assuming that the functions are polynomial,  $f_r(\tau)$  is made identical to  $w_r$ , and the least-square method computes the coefficients of  $f_r$ . Similarly, the coefficients of  $f_s(\tau)$  are computed.

Fig. 3 details the internal mechanism of the run-time system. For each frame, the animated skeleton  $\mathbf{q}$  and the joint torque  $\tau$  are provided as input. Then,  $D_{min}(\mathbf{q})$  produces  $D_{min}$ , which consists of  $\rho_{min}$  (the rotation component represented in a quaternion) and  $S_{min}$  (the scale/shear component represented in a vector). Similarly,  $D_{max}(\mathbf{q})$  produces  $D_{max}$  that consists of  $\rho_{max}$  and  $S_{max}$ .

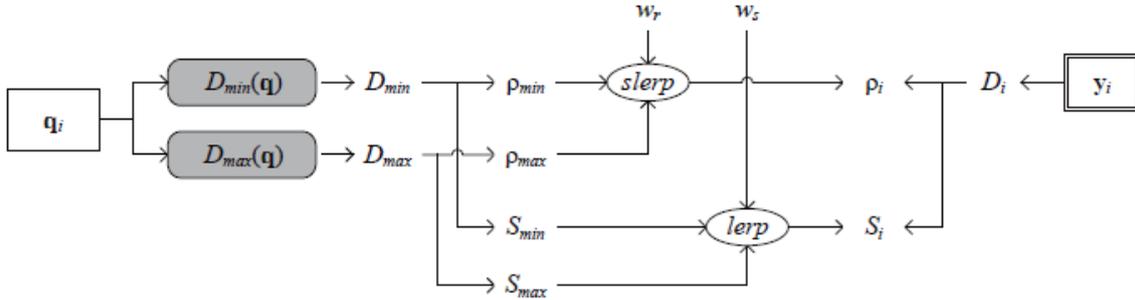


Figure 2. Determining the blending functions. The blending weights,  $w_r$  and  $w_s$ , are computed and then used to determine the coefficients of the polynomial blending functions,  $f_r$  and  $f_s$ , respectively.

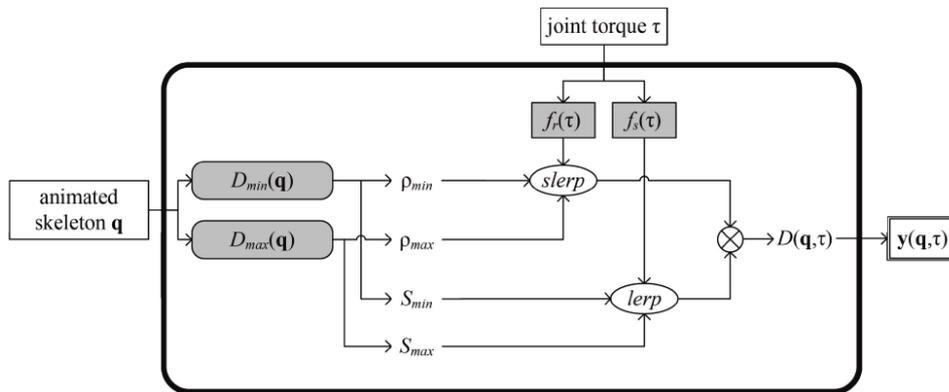


Figure 3. Run-time flow of the physically based skinning with the blending functions.

The input torque  $\tau$  is normalized using the range of minimum and maximum torques, and then  $f_r(\tau)$  is computed. The result is clamped to the range of  $[0, 1]$  and is used for blending  $\rho_{min}$  and  $\rho_{max}$ :

$$R(\mathbf{q}, \tau) = \text{slerp}(\rho_{min}, \rho_{max}, f_r(\tau))$$

where  $R(\mathbf{q}, \tau)$  represents the  $3 \times 3$  matrix derived from the interpolated quaternion. Similarly,  $f_s(\tau)$  is computed and is used for blending  $S_{min}$  and  $S_{max}$ :

$$S(\mathbf{q}, \tau) = (1 - f_s(\tau))S_{min} + f_s(\tau)S_{max}$$

$R(\mathbf{q}, \tau)$  and  $S(\mathbf{q}, \tau)$  are combined to define the deformation gradient:

$$D(\mathbf{q}, \tau) = R(\mathbf{q}, \tau) * S(\mathbf{q}, \tau)$$

Finally, the deformed mesh  $\mathbf{y}(\mathbf{q}, \tau)$  is computed from  $D(\mathbf{q}, \tau)$ .

#### IV. BLENDING FUNCTION EVALUATION

In the current study, the degree of the polynomial blending function is limited up to three, i.e., it is either linear, quadratic, or cubic. Because  $f_r(\tau)$  and  $f_s(\tau)$  are independent, three instances of  $f_s(\tau)$  are also computed, and then nine combinations of  $f_r(\tau)$  and  $f_s(\tau)$  are evaluated.

Given a set of skeleton-mesh-torque triples  $\{(\mathbf{q}_i, \mathbf{y}_i, \tau_i)\}$ ,  $\mathbf{q}_i$  and  $\tau_i$  are provided as input to the run-time system. The system uses a combination of  $f_r(\tau)$  and  $f_s(\tau)$  and outputs  $\mathbf{y}(\mathbf{q}_i, \tau_i)$ . It is compared with  $\mathbf{y}_i$ , and the error is measured per vertex; the squared distance between the corresponding vertices of  $\mathbf{y}(\mathbf{q}_i, \tau_i)$  and  $\mathbf{y}_i$  is computed.

#### V. RESULT

The examples can be generated using a modeling and animation package possibly with the aid of a muscle simulator. In the current implementation, the examples are created by hand using 3ds Max. Fig. 4 shows 3ds Max screen shots at the time of authoring the examples. The lengths of arm and leg in the rest pose are 0.794 and 1.015, respectively.

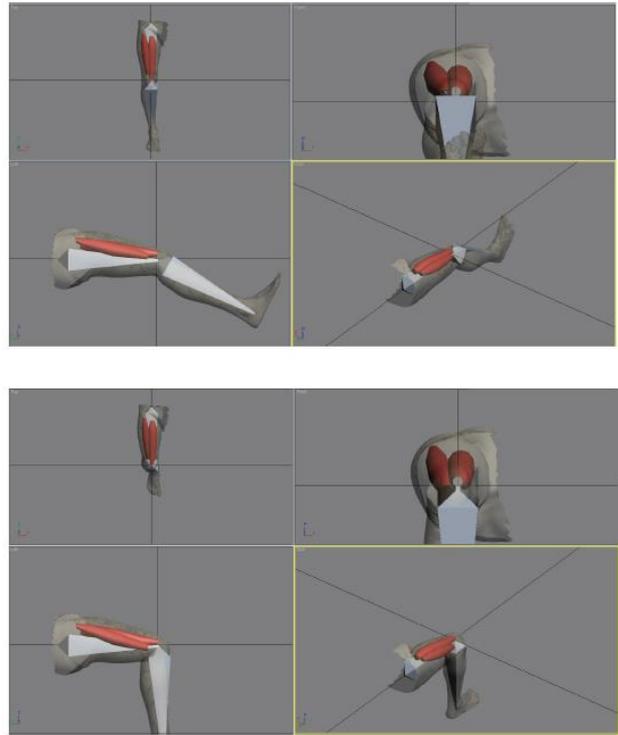


Figure 4. Authoring leg examples.

		$f_r$				M(sd)
		linear (n=1873)	quadratic (n=1873)	cubic (n=1873)	total (n=5619)	
$f_s$	linear (n=1873)	0.00291259 (0.00298195)	0.00293518 (0.0022740830)	0.00290336 (0.002221413)	0.00291704 (0.002240993)	
	quadratic (n=1873)	0.0028879 (0.002259423)	0.00291126 (0.002310925)	0.00287895 (0.002252499)	0.0028927 (0.002274068)	
	cubic (n=1873)	0.00292662 (0.002232847)	0.00295322 (0.002276725)	0.00291479 (0.002226049)	0.00293154 (0.002244977)	
	total (n=5619)	0.00290904 (0.002239856)	0.00293322 (0.002286963)	0.00289903 (0.002233015)	0.00291376 (0.002253318)	

Main effect for  $f_r$ :  $F(2,16848)=.342, p=.711, ns$   
 Main effect for  $f_s$ :  $F(2,16848)=.426, p=.653, ns$   
 Interaction( $f_r * f_s$ ) Effect:  $F(4,16848)=.001, p=1.000, ns$

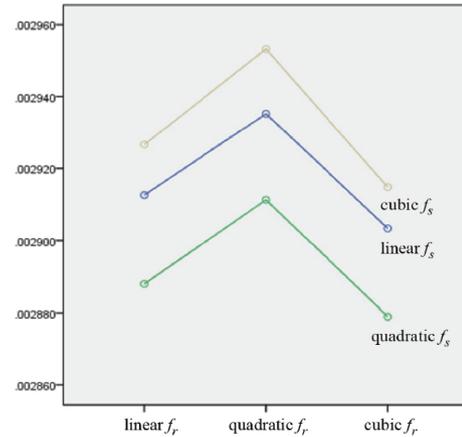


Figure 5. Statistical analysis about the arm model. For the sake of simplicity, we use the symbol < to denote the statistical difference such that the ‘better’ function has a ‘smaller’ value.

		$f_r$				M(sd)
		linear (n=3369)	quadratic (n=3369)	cubic (n=3369)	total (n=10107)	
$f_s$	linear (n=3369)	0.00483203 (0.003402375)	0.00478021 (0.003405308)	0.00481102 (0.003389172)	0.00480775 (0.003398689)	
	quadratic (n=3369)	0.00453176 (0.003535153)	0.0044848 (0.00353487)	0.00450976 (0.003524109)	0.00450877 (0.003531084)	
	cubic (n=3369)	0.00490894 (0.003367239)	0.00485697 (0.003370255)	0.00488836 (0.003354456)	0.00488476 (0.003363725)	
	total (n=10107)	0.00475758 (0.003439196)	0.00470733 (0.003440943)	0.00473638 (0.003426917)	0.00473376 (0.003435639)	

Main effect for  $f_r$ :  $F(2,30312)=.546, p=.579, ns$   
 Main effect for  $f_s$ :  $F(2,30312)=33.845, p=.000, ***$   
 [Scheffe post hoc test: quadratic  $f_s <$  linear  $f_s, cubic f_s$ ]  
 Interaction( $f_r * f_s$ ) Effect:  $F(4,30312)=.001, p=1.000, ns$

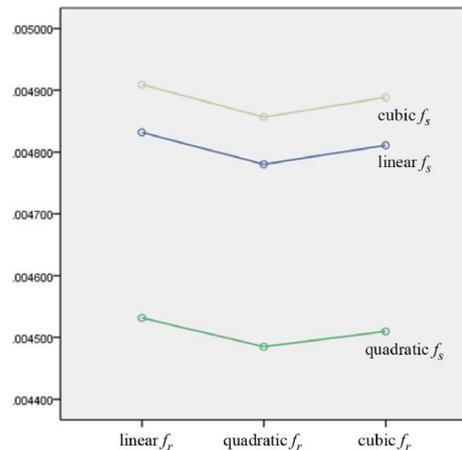


Figure 6. Statistical analysis about the leg model. For the sake of simplicity, we use the symbol < to denote the statistical difference such that the ‘better’ function has a ‘smaller’ value.

Fig. 5, Fig. 6 show statistical analysis of the errors measured for 30 instances of skeleton-mesh-torque triples. In the case of arm, any of linear, quadratic, and cubic polynomials can be chosen for both  $f_r$  and  $f_s$ , and no statistically significant difference would be made. In contrast,  $f_s$  needs to be quadratic in the case of leg.

To maximize the performance, per-triangle deformation gradients were computed using CUDA. Mesh generation from the deformation gradients required solving a Poisson equation. Matrix-vector multiplications used for solving the equation were implemented using cuBLAS API. The vertex normal of the deformed mesh were computed using the one-ring neighbors within the GPU. Using interoperability of the GPU, the deformed mesh was immediately rendered without the CPU-GPU read-back. Fig. 7 and Fig. 8 show the render results.

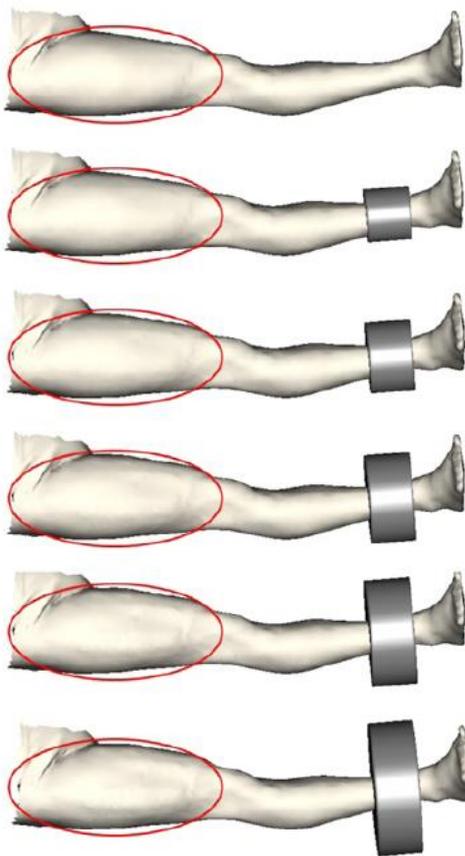


Figure 7. Leg surfaces deformed with varying torques.

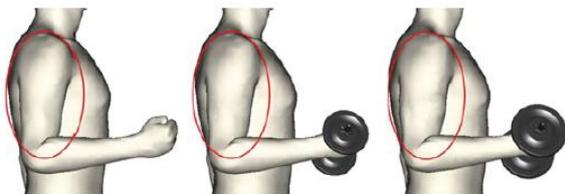


Figure 8. Arm surfaces deformed with varying torque.

## VI. LIMITATION AND FUTUREWORK

In the current study, motion such as lifting a dumbbell is assumed to require torque on a single joint. In reality,

however, torques are often exerted on multiple torques due to such motion. Our future work is developing of the new skinning, which cope with multi-joint torques. It can be also extended if the anatomic data of human is considered. The anatomy and physics based simulation methods can be used as skeleton-mesh-torque example generator.

## ACKNOWLEDGMENT

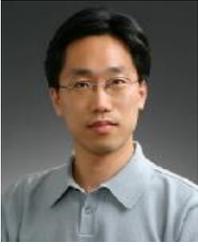
This work was supported by the Ministry of Trade, Industry & Energy grant funded by the Korea government (No. 100S51089).

## REFERENCES

- [1] Y. Kim and J. Han, "Bulging-free dual quaternion skinning," *Computer Animation Virtual Worlds*, vol. 25, no. 3-4, pp. 323-331, May 2014.
- [2] B. H. Le and J. K. Hodgins, "Real-time skeletal skinning with optimized centers of rotation," *ACM Trans. Graph.*, vol. 35, no. 4, article 37, July 2016.
- [3] Y. Kim and J. Han, "Physically-based linear blend skinning," presented at 11th International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing, The Lisbon, Portugal, July 20-22, 2017.
- [4] J. Teran, E. Sifakis, G. Irving, and R. Fedkiw, "Robust quasistatic finite elements and flesh simulation," in *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2005, pp. 181-190.
- [5] CGCHARACTER: Absolute character tools 1.6. (2003). [Online]. Available: <http://www.cgcharacter.com>
- [6] J. P. Lewis, M. Cordner, and N. Fong, "Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation," in *Proc. 27th Annual Conference on Computer Graphics and Interactive Techniques*, 2000, pp. 165-172.
- [7] A. Mohr and M. Gleicher, "Building efficient, accurate character skins from examples," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 562-568, July 2003.
- [8] X. C. Wang and C. Phillips, "Multi-weight enveloping: Least-squares approximation techniques for skin animation," in *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2002, pp. 129-138.
- [9] B. Merry, P. Marais, and J. Gain, "Animation space: A truly linear framework for character animation," *ACM Trans. Graph.*, vol. 25, no. 4, pp. 1400-1423, Oct. 2006.
- [10] P. P. J. Sloan, C. F. Rose III, and M. F. Cohen, "Shape by example," in *Proc. Symposium on Interactive 3D Graphics*, 2001, pp. 135-143.
- [11] P. G. Kry, D. L. James, and D. K. Pai, "Eigenskin: Real time large deformation character skinning in hardware," in *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2002, pp. 153-159.
- [12] R. Y. Wang, K. Pulli, and J. Popovic, "Real-time enveloping with rotational regression," *ACM Trans. Graph.*, vol. 26, no. 3, article 73, July 2007.
- [13] R. W. Sumner and J. Popovic, "Deformation transfer for triangle meshes," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 399-405, Aug. 2004.



**YoungBeom Kim** is an associate research engineer of the Korea Electronics Technology Institute (KETI). He received a BS (2008) degree in Computer Science and Engineering at Korea University. He also received MS (2010) and PhD (2016) degrees in Computer Science and Engineering at same university. His research interests focus on real-time character animation and virtual training.



**Byung-Ha Park** is a senior research engineer of the Korea Electronics Technology Institute (KETI). He received a BS (1999) degree in Computer Science at Sejong University, obtained an MS (2001) degree in Computer Engineering at Sejong University. His research interests focus on smart media contents, realistic media, and virtual training.



**Kwang-Mo Jung** is a principle research engineer of the Korea Electronics Technology Institute (KETI). He received a BS (1990) degree in Electronics and Communication at KwangWoon University. He also received MS (2002) and PhD (2006) degrees in Electronics and Communication at same university. His research interests focus on smart media contents, realistic media, and virtual training.