TELKOMNIKA, Vol. 11, No. 10, October 2013, pp. 5553 ~ 5562
ISSN: 2302-4046

Dual Quaternion Blending Algorithm and Its Application in Character Animation

Xiang Feng*[1,2], Wanggen Wan[1,2]
[1]School of Communication and Information Engineering, Shanghai University, 200072 Shanghai, China
[2]Institute of Smart City, Shanghai University, 200444, Shanghai, China
*Corresponding author, e-mail: fengxiang0727@shu.edu.cn

Abstract

In this paper we generalize established techniques and blending algorithm for quaternions to dual quaternions to represent rigid transformations compactly. With the visualization of OpenGL, we employ dual quaternions to achieve character animation in real time. Classical quaternions are only able to characterize rotations although combination of matrix calculation and quaternions operator has been a popular tool in character animation since 1990s. In character animation and some other applications of 3D computer graphics, we are actually faced with rigid transformation which just includes translation and rotation. Similar to the way quaternions represent rotations, dual quaternions represent rigid transformations. Algorithms based on dual quaternions own better properties than those based on quaternions in practical applications, which include reduced overhead, increased computational efficiency and coordinate invariance. Finally we demonstrate the effectiveness of dual quaternions blending algorithm by cartoon male and female mesh models with the animation of walking and waving..

Keywords: *Dual quaternion blending algorithm, Character animation, Rigid transformation*

Copyright © 2013 Universitas Ahmad Dahlan. All rights reserved.

1. Introduction

As is shown in many literatures, quaternions [1] are an advantageous representation of 3D rotations than rotation matrices in many aspects better. But rigid objects not only rotate, but also translate. Since a rotation composed with translation is called a rigid transformation, any displacement of a rigid object in 3D space can be described by a rigid transformation. In this paper, we advocate that dual quaternions are a better representation of rigid transformations than those treating rotation and translation components independently. We combine dual quaternions to implement real time skeletal animation.

In 3D computer graphics, character animation can be done in several ways. When the animations become more complex it poses problems in matter of memory usage we usually employ skeletal animation system [2]. We first build a skeleton and joints inside the meshes we wish to animate, and then animate the skeleton within instead of animating the mesh itself.

In the skeleton, every vertex corresponds to one or more joints through an coefficient. Not only the vertex index and the joint indexes are stored in the coefficient, but also the influence weights between the joints and the vertex are stored [3]. For any vertex, there is one and only one influence weights associated with a joint and the influence weights should add up to one numerically. As the mesh seems like a skin in terms of the skeleton, we typically call this process as skeleton skinning or skining.

2. Related Work

Though the dual quaternion has been around since 1882, it has gained less attention compared to quaternions alone [4]. Comparable to quaternions the dual quaternion has had a taboo associated with them, whereby students avoid quaternion and hence dual-quaternions. While the research community in robotics has started to adopt dual-quaternions in recent years, the research community in computer graphics such as character animation has not embraced them as whole-heartedly.

Schilling used dual quaternion with a mean of multiple computational model to model bodies [5]. Pham used Jacobian matrix in the dual quaternion space to solve linked chain

Received March 27, 2013; Revised June 21, 2013; Accepted July 10, 2013

inverse kinematic problems [6]. Yang used dual quaternions to calculate the relative orientation [7].

Kuang presented a strategy for creating real time animation of clothed body movement [8]. Vasilakis discussed skeleton based rigid skinning for character animation [9]. Selig examined the problem of solving the equations of motion in real-time, put forward how dual quaternions gave a very neat and succinct way to represent rigid-body transformations, and addressed the key problem in computer games [10].

3. Dual Quaternion

3.1. Quaternion

Quaternion has been a popular tool in 3D computer graphics for more than 20 years [11], they are four terms real numbers $(q_r\ q_x\ q_y\ q_z)$ which include a three-term vector with components $q_x\ q_y$ and q_z. Quaternion is usually represented in the form

$$Q = q_r + q_x\vec{i} + q_y\vec{j} + q_z\vec{k} = q_r + \vec{q} \tag{1}$$

Where q_r and $\vec{q}$ are the real and vector parts, respectively, and $\vec{i}$, $\vec{j}$ and $\vec{k}$ are the unit vectors associated with the axes of a Cartesian coordinate system. A dual quaternion can be used to define a rigid body rotation of an angle θ about an axis $\vec{u}$ through the origin

$$Q = \cos\frac{\theta}{2} + \vec{u}\sin\frac{\theta}{2} \tag{2}$$

3.2. Dual Quaternion

Classical quaternions are restricted to the representation of rotations, whereas in graphical applications we typically work with rotation composed with translation, i.e. rigid transformations. Dual quaternions are mathematical entities [12] whose four components are dual numbers, and they can be expressed as follows:

$$\hat{Q} = Q + \varepsilon Q_0 \tag{3}$$

Where Q and Q_0 are both quaternions, and ε is dual unit. Dual quaternion can formulate a problem more concisely, solve it more rapidly and in fewer steps, present the result more plainly to others, be put into practice with fewer lines of code and debugged effortlessly. Dual-quaternion has a unified representation of translation and rotation as follows:

$$Q = r\ ;\ Q_0 = \frac{1}{2}t \cdot r \tag{4}$$

Where r is a unit quaternion representing the rotation and t is the quaternion describing the translation represented by the vector t , as we can see from Figure 1.

Dual quaternions [13] represent rigid transformations in the same way as classical quaternions represent rotations. The dual quaternion blending algorithms [14] could be applied in motion blending, motion analysis, spatial key framing, computer vision, graphics hardware.

3.3. Dual Quaternion Linear Blending

After skinning transformations are converted to unit dual quaternions $\hat{q}_1, \dots, \hat{q}_n$, what we need to do next is use the given weights to compute a blended unit dual quaternion $\hat{q}$

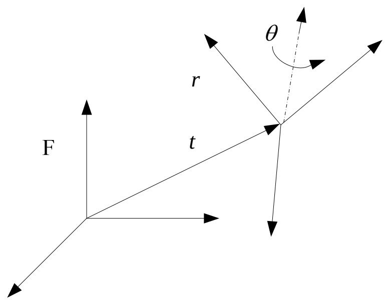


Figure 1. Rigid transformation with dual quaternions

$$w = (w_1, \cdots, w_n) \tag{5}$$

We achieve this by first taking their linear combination and then normalizing the result to get a unit dual quaternion, which is called Dual quaternion Linear Blending (DLB) [15]:

$$DLB(w; \hat{q}_1, \ldots, \hat{q}_n) = \frac{w_1\hat{q}_1 + \ldots + w_1\hat{q}_n}{\| w_1\hat{q}_1 + \ldots + w_1\hat{q}_n \|} \tag{6}$$

DLB has many excellent properties required in skinning. A detailed description appears in literatures [16] and [17]. DLB produces a unit dual quaternion and we can convert it to a rigid transformation matrix, that is, DLB returns a rigid transformation. The following formula proves to be true for any unit dual quaternion $\hat{q}$ to demonstrate the fact that DLB is coordinate-invariant.

$$DLB(w; \hat{r}\hat{q}_1\hat{r}^*, \ldots, \hat{r}\hat{q}_n\hat{r}^*) = \hat{r}DLB(w; \hat{q}_1, \ldots, \hat{q}_n)\hat{r}^* \tag{7}$$

In fact, it can be decomposed to two properties: left invariance and right invariance. The following formulas represent left invariance and right invariance respectively:

$$DLB(w; \hat{r}\hat{q}_1, \ldots, \hat{r}\hat{q}_n) = \hat{r}DLB(w; \hat{q}_1, \ldots, \hat{q}_n) \tag{8}$$

$$DLB(w; \hat{q}_1\hat{r}, \ldots, \hat{q}_n\hat{r}) = DLB(w; \hat{q}_1, \ldots, \hat{q}_n)\hat{r} \tag{9}$$

Coordinate-invariance includes both left invariance and right invariance. The multiplicative property of the norm and the distributive property of dual quaternions are applied to prove left invariance based on the assumption:

$$\| \hat{r} \| = 1 \tag{10}$$

$$DLB(w; \hat{r}\hat{q}_1, \ldots, \hat{r}\hat{q}_n) = \frac{w_1\hat{q}_1 + \ldots + w_n\hat{q}_n}{\| w_1\hat{q}_1 + \ldots + w_1\hat{q}_n \|} =$$
$$\hat{r}\frac{w_1\hat{q}_1 + \ldots + w_n\hat{q}_n}{\| \hat{r} \| \; \| w_1\hat{q}_1 + \ldots + w_n\hat{q}_n \|} = \hat{r}DLB(w; \hat{q}_1, \ldots, \hat{q}_n) \tag{11}$$

Similarly, the property of right invariance can be proved. The combination of left invariance and right invariance makes DLB coordinate-invariant. For any two unit dual quaternions $\breve{p}$ and $\hat{q}$, the blending is given as $DLB(t;\breve{p},\hat{q})$ and the path of DLB interpolation is shortest.

$$DLB(t;\breve{p},\hat{q}) = \breve{p}\breve{p}^* DLB(t;\breve{p},\hat{q}) = \breve{p}\ DLB(t;1,\breve{p}^*\hat{q}) \tag{12}$$

It is therefore sufficient to show that the path between 1 and $\breve{p}^*\hat{q}$ will be the shortest one which is given by the screw corresponding to $\breve{p}^*\hat{q}$. It's obvious that $\breve{p}^*\hat{q}$ is a unit dual quaternion. We can always find a dual scalar $\breve{\alpha}$ and a unit dual quaternion $\hat{n}$ that satisfys the formula

$$\breve{p}^*\hat{q} = \cos\frac{\breve{\alpha}}{2} + \hat{n}\sin\frac{\breve{\alpha}}{2} \tag{13}$$

Therefore, $DLB(t;\breve{p},\hat{q})$ can be given as:

$$DLB(t;1,\breve{p}^*\hat{q}) = \frac{1-t+t\breve{p}^*\hat{q}}{\|1-t+t\breve{p}^*\hat{q}\|} = \frac{1-t+\cos(\frac{\breve{\alpha}}{2})+\hat{n}t\sin(\frac{\breve{\alpha}}{2})}{\|1-t+t\breve{p}^*\hat{q}\|} \tag{14}$$

It can be concluded the amount of translation and the angle of rotation vary with $t \in [0,1]$ while the screw axis of $DLB(t;1,\breve{p}^*\hat{q})$ keeps the same. $DLB(t;1,\breve{p}^*\hat{q})$ makes the screw motion go along a shortest path

$$\frac{t\sin(\frac{\breve{\alpha}}{2})}{\|1-t+t\breve{p}^*\hat{q}\|} \tag{15}$$

$$\frac{1-t+t\cos(\frac{\breve{\alpha}}{2})}{\|1-t+t\breve{p}^*\hat{q}\|} \tag{16}$$

Similar to linear interpolation of quaternions, the velocity of the screw motion is not constant. But in practical skeleton animation, the skinning will not be influenced seriously by the time-dependent speed. The procedure of Simple Dual quaternion Linear Blending is shown in Table 1.

Table 1. Simple Dual quaternion Linear Blending

Algorithm 1 sDLB: Simple Dual quaternion Linear Blending
procedure $SDLB(\hat{q}_1,\hat{q}_2,t)$ $SDLB(t;\hat{q}_1,\hat{q}_2) = \frac{(1-t)\hat{q}_1 + t\hat{q}_2}{\|(1-t)\hat{q}_1 + t\hat{q}_2\|}$ **end procedure**

3.4. Screw Linear Interpolation

This is a generalization of the well-known Spherical Linear Interpolation (SLERP) [18] scheme. Let denote by $\hat{q}_1$ and $\hat{q}_1$ two dual quaternions expressing the initial and final pose of a rigid body, respectively. The ScLERP function (Screw Linear Interpolation) [19] is defined as follows

$$ScLERRP(t;\hat{q}_1,\hat{q}_2) = \hat{q}_1 * (\hat{q}_1 * \hat{q}_2)^t \ , t \in [0,1] \tag{17}$$

Since $\hat{q}_1^{-1} * \hat{q}_2$ represents the finite screw motion between the initial and final pose of the rigid body, the product

$$(\hat{q}_1^{-1} * \hat{q}_2)^t = \cos(t\frac{\hat{\theta}}{2}) + \sin(t\frac{\hat{\theta}}{2})\hat{u} \tag{18}$$

defines a screw motion of a dual angle along the screw axis.

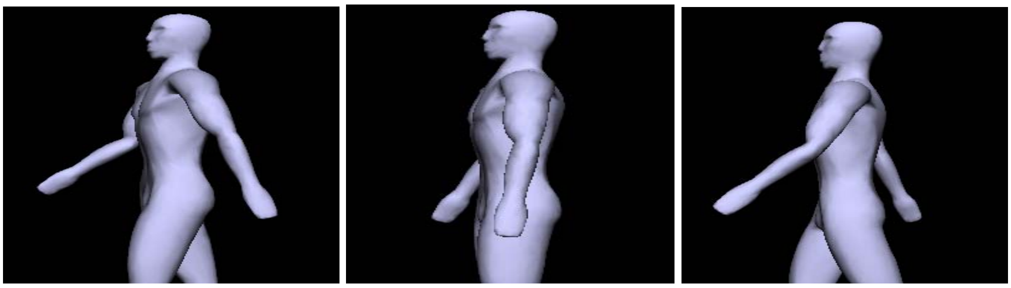


Figure 2. Character animation with dual quaternions, frame by frame, the subfigures show three different frames of walking male with DLB.

3.5. Analytical Comparison between DLB and ScLERP

As is mentioned above, it's plausible to interpolate rigid transformations with Dual quaternion Linear Blending. But it is not a group intrinsic method, thus it's not a perfect mothod, it involves a trick named normal-interpolation. In this paper we discuss whether this will introduce artifacts when employing DLB in the blending procedure. We take two transformations for example. In order to respect the geometry of the underlying group, we establish a perfectly correct blending method.

Theoretically, Spherical Linear Interpolation (SLERP) is a perfect solution for blending [16] [17]. For any two unit quaternions q_1, q_2, let's assume that $\langle q_1, q_2 \rangle \geq 0$, then the formula is $SLERP(t;q_1,q_2) = (q_2 q_1^*)^t q_1$ with parameter $t \in [0,1]$. We can easily generalize SLERP to dual quaternions. In case of dual quaternions, the method is called Screw Linear Interpolation (ScLERP). For q_1 and q_2, the interpolation can be written as $ScLERP(t;\hat{q}_1,\hat{q}_2) = (\hat{q}_2 \hat{q}_1^*)^t q_1$

It is clear that $\hat{q}_2 \hat{q}_1^*$ represents the relative motion between $\hat{q}_1$ and $\hat{q}_2$, and it's also a unit dual quaternion. For a dual angle $\hat{\alpha}$ and dual vector $\hat{n}$, the power of $\hat{q}_2 \hat{q}_1^*$ can be given as follows.

$$(\hat{q}_2 \hat{q}_1^*)^t = \cos(t\frac{\hat{\alpha}}{2}) + \hat{\mathrm{n}}\sin(t\frac{\hat{\alpha}}{2}) \tag{19}$$

In the expression above, the dual vector $\hat{n}$ represents the axis of the screw motion. The translation quantity $t\alpha_\varepsilon$ and the rotation quantity $t\alpha_o$ are included in the dual angle $t\frac{\hat{\alpha}}{2} = t\frac{\alpha_o}{2} + \varepsilon t\frac{\alpha_\varepsilon}{2}$. Two important attributes can be found out obviously: both the amount of translation $t\alpha_\varepsilon$ and the angle of rotation $t\alpha_o$ follow a linear variation with the interpolation quantity t; the axis $\hat{n}$ of the screw motion is constant, that is to say, $\hat{n}$ is independent of t. As a generalization of SLERP, ScLERP is expected to be a shortest path interpolation solution with a constant speed. As another property, ScLERP is coordinate invariant, which can be easily proved.

It can be seen that the interpolation of ScLERP hold the same behavior as SLERP. ScLERP can be used as a standard reference to carry on an analysis and comparison of DLB. It should be noted that the error is invariant with the coordinate systems, which is an important property. The common property of DLB and ScLERP can be used to simplify the comparison process. In order to make a correct comparison between $ScLERP(t;\hat{q}_1,\hat{q}_2)$ and $DLB(t;\hat{q}_1,\hat{q}_2)$, we express them as $ScLERP(t;1,\hat{q}_2\hat{q}_1^*)\hat{q}_1$ and $DLB(t;1,\hat{q}_2\hat{q}_1^*)\hat{q}_1$ reseparately, which can be easily proved with the property of right-invariance. It is enough to compare $DLB(t;1,\hat{q}_2\hat{q}_1^*)$ with $ScLERP(t;1,\hat{q}_2\hat{q}_1^*)$ as $\hat{q}_1$ can be omitted in both expressions. $\hat{q}_2\hat{q}_1^*$ can be given as $\hat{q}_2\hat{q}_1^* = \cos\frac{\hat{\alpha}}{2} + \hat{n}\sin\frac{\hat{\alpha}}{2}$ as it's a unit dual quaternion.

$$DLB(t;1,\hat{q}_2\hat{q}_1^*) = \frac{1-t+t\hat{q}_2\hat{q}_1^*}{\|1-t+t\hat{q}_2\hat{q}_1^*\|}$$
$$= \frac{1-t+t\cos(\frac{\hat{\alpha}}{2}) + \hat{n}\,t\sin(\frac{\hat{\alpha}}{2})}{\|1-t+t\hat{q}_2\hat{q}_1^*\|} \tag{20}$$

$$ScLERP(t;1,\hat{q}_2\hat{q}_1^*) = (\hat{q}_2\hat{q}_1^*)^t$$
$$= \cos(t\frac{\hat{\alpha}}{2}) + \hat{n}\sin(t\frac{\hat{\alpha}}{2}) \tag{21}$$

It is easy to find that DLB and ScLERP share the same screw axis $\hat{n}$. The only difference between DLB and ScLERP lies in amount of translation and the angle of rotation along the same screw axis. $DLB(t;1,\hat{q}_2\hat{q}_1^*)$ can be given in the following form as it's a unit dual quaternion.

$$DLB(t;1,\hat{q}_2\hat{q}_1^*) = \cos(\frac{\hat{\beta}_t}{2}) + \hat{n}\,t\sin(\frac{\hat{\beta}_t}{2}) \tag{22}$$

From the formula above, we can derive the followig formula by only considering the scalar part.

$$\cos\frac{\hat{\beta}_t}{2} = \frac{1-t+t\cos(\frac{\hat{\alpha}}{2})}{\|1-t+t\,\hat{q}_2\hat{q}_1^*\|} \tag{23}$$

In order to compute the difference between DLB and ScLERP, we compare the dual angle $\hat{\beta}_t$ with the dual angle $\hat{\alpha}t$. It requires a complicated process of mathematical analysis, and it is easy to carry out the computations by means of mathematical tools. The result is consistent with the results reported in early literatures for the case of quaternions. In practice, the differencs are always smaller than theoretical maximum values. In particular, for skeletal animation, the difference between DLB and ScLERP can hardly be observed with naked eyes.

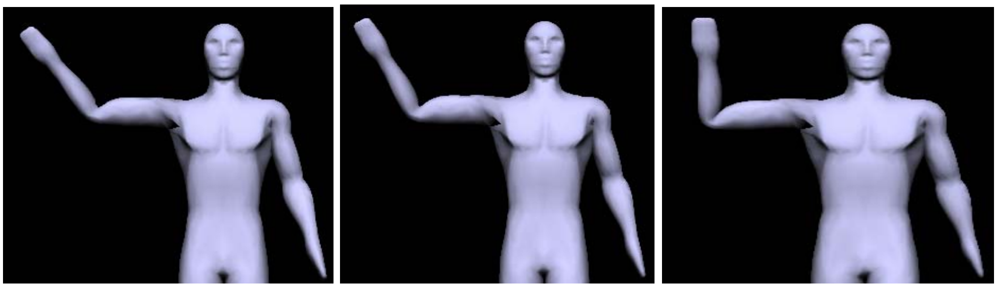


Figure 3. Character animation with dual quaternions, frame by frame, the subfigures show four different frames of waving-arm male with ScLERP.

4. Character Animation

In the procedure of character animation, we use degrees of freedom (DOF) to represent possible range of movement for a joint of the character. For the whole skeleton of a complicated character, the numben of DOFs may be hundreds of thousands. It's the core of skeleton animation to assign specific motion parameters for each joint. The movement over time for the skeleton can be achieved by updating these parameters.

A simple character skeleton and the hierarchical topological graph of skeleton joints are shown in Figure 4. An insight into skeletons is given in reference [20].

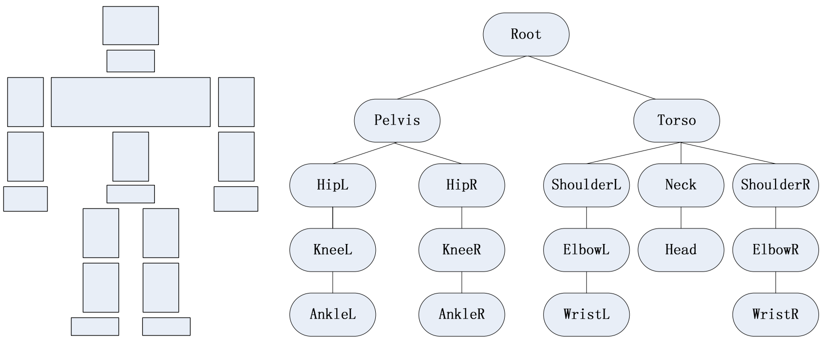


Figure 4. Left is character skeleton, right is hierarchical topological graph of skeleton joints

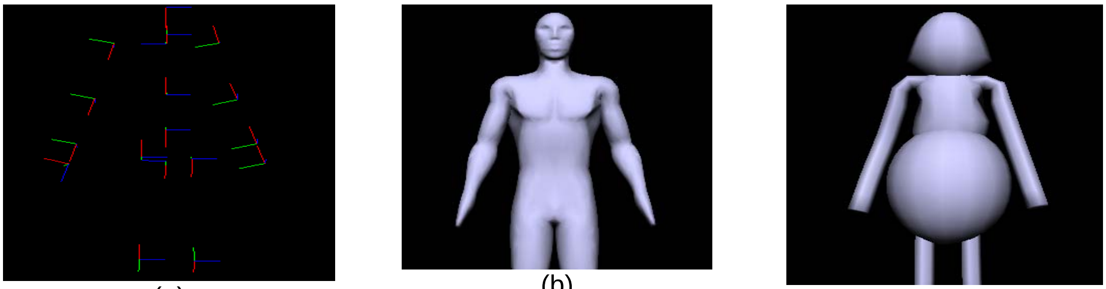
For each joint, we can construct a local matrix with the given DOF values, and define relative orientation and position with respect to its parent joint in the hierarchical topological graph. With the local matrices obtained, the world space matrices for all joints can be computed using forward kinematics. Once world space matrices are obtained, the character can ultimately be rendered frame by frame, such as skeleton animation and collision detection. When transformed by a matrix, a vector undergoes the following conversion:

$$V' = V \cdot M \tag{24}$$

V is a original vector to be transformed, V' is the resulting transformed vector, and $M$is the transformation matrix. In particular, we can use this conversion to place a vertex in local coordinate space to world coordinate space. The inverse of the conversion will lead the following conversion:

$$V = V' \cdot M^{-1} \tag{25}$$

M^{-1} is the inverse matrix of M. The inversion conversion formula tells us we we can also transform a object from world coordinate space to local coordinate space.



(a) (b) (c)

Figure 5. Left represents Hierarchical skeleton, middle and right represent male and female models warping the inner skeleton respectively

In the implementation process with computer graphics and OGENGL, we first build a hierarchical skeleton which consists of several nodes, and then we wrap the hierarchical skeleton with a specified mesh model. A similar implementation that adopts quaternions can be found in [21]. In this paper, we take a male model and a female model to demonstrate the course of character animation and show the effectiveness of the proposed algorithms. In Figure 5, (a) represents the skeleton of a cartoon character, (b) and (c) represents the male and female model warping the same inner skeleton respectively.

5. Implementation

The dual-quaternion unifies the translation and rotation into a single state variable. This single state variable offers a robust, unambiguous, computationally efficient way of representing rigid transform. We combine skeletal animation and dual quaternions to implement character animation with OpenGL.

We demonstrate the real time result in Figure 2, Figure 3, Figure 6 and Figure 7. In Figure 2, the real time animation process of a walking male is shown in sequencing frames, which employs Dual quaternion Linear Blending algorithm. In Figure 3, the real time animation process of a waving male is shown in sequencing frames, which employs Screw Linear Interpolation algorithm. In Figure 6, the real time animation process of a walking cartoon girl is shown in sequencing frames, which employs Dual quaternion Linear Blending algorithm. In Figure 7, the real time animation process of a waving cartoon girl is shown in sequencing frames, which employs Screw Linear Interpolation algorithm.

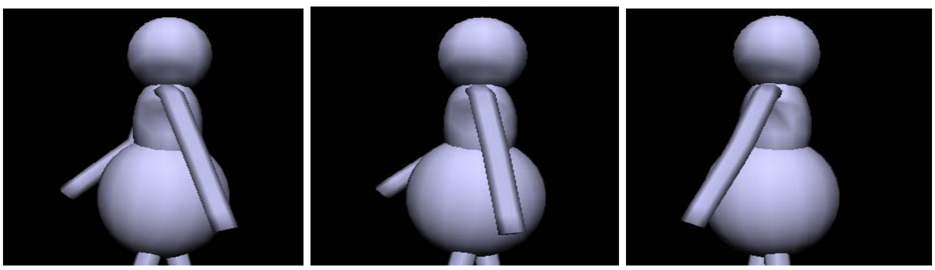


Figure 6. Animation with dual quaternions, frame by frame, the subfigures show three different frames of walking cartoon girl with DLB.

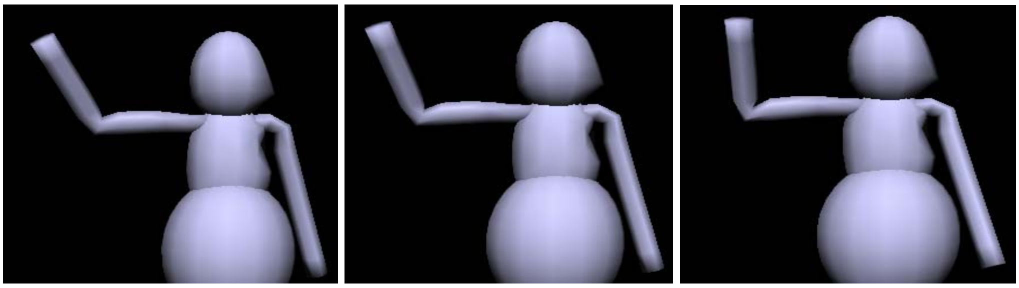


Figure 7. Animation with dual quaternions, frame by frame, the subfigures show three different frames of waving cartoon girl with ScLERP.

Dual quaternion model is an accurate, computationally efficient, robust, and flexible method of representing rigid transforms, which should not be overlooked. It enables the creation of more elegant and clearer computer programs that are easier to work with and control when we implement pre-programmed dual quaternion modules including multiplication and normalization.The computational cost of combining matrices and dual-quaternions:

Matrix4x4 : 64mult + 48adds
Matrix4x3 : 48mult + 32adds
DualQuaternion : 42mult + 38adds

6. Conclusion

In character animation, skeletons consist of several articulated joints that connect the rigid bones in accordance with the hierarchy topology. Rigid transformation blending of bones based on dual quaternions exhibit advantageous properties, and fast execution time. In this paper, we introduce dual quaternion and take the advantage of it. We make use of dual quaternion to represent the translation and rotation, and have implemented real-time character animation with OpenGL.

We have generalized established techniques and blending algorithm for quaternions to dual quaternions to represent rigid transformations compactly. With the experimental results shown in figures above, we have implemented real time character animation with dual quaternion blending algorithm under the platform of OpenGL. We have demonstrated the effectiveness of dual quaternions in character animation by cartoon male and female mesh models with the animation of walking and waving.

In addition to digital character, skeletons are also applicable to real world animals as long as the animals own true bones, such as mammals and humans. Moreover, we can also use skeletons in combination with dual quaternion blending algorithm to simulate the movement of soft tissues.

References

[1] J Mc Donald. Teaching Quaternions is not Complex. *Computer Graphics Forum*. 2010; 29(8): 2447-2455.
[2] X Feng, W Wan. Real Time Skeletal Animation with Dual Quaternion. *Journal of Theoretical and Applied Information Technology*. 2013; 49(1).
[3] TC Xu, Mo Chen, Ming Xie and Enhua Wu. A Skinning Method in Real-time Skeletal Character Animation. *The International Journal of Virtual Reality*. 2011; 10(3):25-31.
[4] Ben Kenwright. *A Beginners Guide to Dual-Quaternions: What They Are, How They Work, and How to Use Them for 3D Character Hierarchies*. The 20th International Conference on Computer Graphics, Visualization and Computer Vision, WSCG 2012 Communication Proceedings. 26-28 June 2012:1-13.
[5] M Schilling. Universally manipulable body models--dual quaternion representations in layered and dynamic MMCs. *Autonomous Robots*. 2011; 30(4): 399-425.
[6] HL Pham, V Perdereau, BV Adorno, and P Fraisse. *Position and orientation control of robot manipulators using dual quaternion feedback*. 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, China. 18-22 October 2010: 658–663.

[7] Y Lin, H. Wang, and Y. Chiang. *Estimation of relative orientation using dual quaternion*. 2010 International Conference on System Science and Engineering (ICSSE2010). 2010: 413-416.
[8] Y Kuang, A Mao, G Li, and Y Xiong. *A strategy of real-time animation of clothed body movement*. The 2nd International Conference on Multimedia Technology (ICMT2011), Hangzhou, China. 2011: 4793-4797.
[9] A Vasilakis and I Fudos. Skeleton-based rigid skinning for character animation. *Computer Graphics Theory and Applications – GRAPP.* 2009: 302–308.
[10] J Selig. Rational interpolation of rigid-body motions. Advances in the Theory of Control, Signals and Systems with Physical Modeling. Lecture Notes in Control and Information Sciences. 2011; 407: 213-224.
[11] Yongmin Zhong, Shesheng Gao and Wei Li. A Quaternion-Based Method for SINS/SAR Integrated Navigation System. *IEEE Transactions on Aerospace and Electronic Systems*. 2012; 48(1): 514-524.
[12] Jiang Feng, Wang Hui-nan, Huang Chao Song. Algorithm for Relative Position and Attitude of Formation Flying Satellites Based on Dual Quaternion. *Chinese Space Science and Technology*. 2012; 32(3): 20-26.
[13] FZ Ivo and H Ivo. *Spherical skinning with dual quaternions and QTangents*. SIGGRAPH 2011, Vancouver, British Columbia, Canada. 2011:11.
[14] Ilie MD, Negrescu C, Stanomir D. *An efficient parametric model for real-time 3D tongue skeletal animation*.2012 9th International Conference on Communications (COMM). 2012: 129-132.
[15] A Perez Gracia. Synthesis of spatial RPRP closed linkages for a given screw system. *Journal of mechanisms and robotics*. 2011; 3(2): 11-19.
[16] Ladislav K, Steven C, Jiri Z, Carol O'Sullivan. Geometric skinning with approximate dual quaternion blending. *ACM Transactions on Graphics*. 2008; 27 (4): 105.
[17] Ladislav Kavan, Steven Collins, Jiri Zara, Carol O'Sullivan. *Skinning with Dual Quaternions*. In Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. ACM Press. 2007: 39–46.
[18] CH Chen, IC Lin, MH Tsai, PH Lu. *Lattice-Based Skinning and Deformation for Real-Time Skeleton-Driven Animation*. Proc. Intl. Conf. on Computer-Aided Design and Computer Graphics (CAD/Graphics'11), IEEE Computer Society. Jinan, China. 2011: 306-312.
[19] Ettore Pennestrì. Pier Paolo Valentini. Dual quaternions as a tool for rigid body motion analysis: a tutorial with an application to biomechanics. *Archive of Mechanical Engineering*. 2010; LVII(2): 187–205.
[20] Information on http://graphics.ucsd.edu/courses/cse169_w05/2-Skeleton.htm
[21] Information on http://www.it.hiof.no/~borres/gb/exp-skeletal/p-skeletal.html