

RESEARCH

Open Access



# A conformal geometric algebra method for virtual hand modeling and interaction

Peng Chen\*, Yong Hu and Fengwei Yang

## Abstract

Virtual hand usually simulates human hands by mapping the actual shapes of hands to virtual environment and modeling the hand-object manipulations in human machine interactions. For describing virtual hand and its manipulation in a unified framework, a method based on conformal geometric algebra (CGA) is proposed to solve the problems of virtual hand modeling and interaction in this paper. With the vertex blending based on CGA, the artifacts on the finger joints are improved, which enhances the realism of the virtual hand model. With the same tool, the collision detection between the virtual hand and the manipulated objects is implemented. The gesture of grasp, pinch, and hold are recognized, and the corresponding manipulation rules are established by CGA calculation. To test these three typical manipulations, the manipulated objects are imported and the manipulation effectiveness is evaluated.

**Keywords:** Virtual hand, Vertex blending, Collision detection, Conformal geometric algebra

## 1 Introduction

In virtual reality system, virtual hands are introduced for modeling human hands and mapping the actual shapes of hands to the virtual environment, which manipulates the virtual objects in the human machine interaction. Appropriate virtual hand deformation and collision detection could enhance the realism and interaction effectiveness of the system. The mathematical tool for virtual hand modeling and interaction is significant, and the corresponding manipulation rules are also required.

It is usual to construct a static virtual hand with the dimension parameters of the human hand, but the deformation of the hand skeleton and skin, especially that on the finger joint during motion is difficult to model. McDonald utilized the measurement data and physiological structure for virtual hand modeling [1], while Moccozet provided a multi-layered virtual hand model with deformation of both the skeleton and skin for the application of hand motion simulation [2]. Kurihara constructed the hand shapes with a large amount of medical images, which could control the skin deformation by setting motion weights in different parts interactively [3]. Generally, vertex blending simply added the weighted motions by a linear method for the

transformations of vertices on the skin [4]. For the virtual hand, linear vertex blending might cause constriction on the finger joints. It is because that the rotation transformation space is not linear, and the directly added rotations introduce extra motions. To solve the problem, Kavan employed sphere and dual quaternion vertex blending methods to achieve more realistic skinning [5].

To provide the sense like the real world, most virtual reality systems have the function of collision detection. The layered bonding boxes with the sphere or oriental shapes were used widely in collision detection [6, 7]. To reduce the tests between elementary test pairs, the continuous collision detection method integrated interval arithmetic and hierarchies of oriented bounding boxes [8]. For virtual hand interaction, the intersection of ray and distance calculation became the means of detecting the collision [9]. Wan implemented the collision detection of the virtual hand by dividing it into simple geometric objects by the package RAPID [10]. Besides these geometric methods, algebraic intersections using matrix methods were also introduced to solve the intersection problems of parametric and algebraic curves [11]. Compared with rigid body collisions, Voronoi-based culling algorithm was presented to perform collision and distance queries among multiple deformable models [12]. To improve the real-time performance of both the rigid and deformable models in

\* Correspondence: [chenpeng\\_913@sohu.com](mailto:chenpeng_913@sohu.com)

School of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, China

collision detection, GPU computation techniques for construction of a signed distance field were employed [13].

For further interaction, certain manipulation rules are necessary, which could simulate the process of manipulation based on the physical rules [14]. However, physical-based methods have to rely on the haptic or force feedback equipment. Holz proposed a manipulation method for single hand grasp by the fingers with collision, and there must be a finger applied to the manipulated object in each rendering frame [15]. For the application in industry, Moehring proposed a semi-physical heuristic simulation method for the manipulation with sufficient realism and practicability, which defined the collision points pair and the normal vector. To validate the manipulation, the friction cone was also established for describing the force applied to the manipulated object by the angle of friction [16].

According to the problems in virtual hand manipulation, a method based on the unified mathematical tool, conformal geometric algebra (CGA) is proposed to solve the problems of virtual hand modeling and interaction in this paper. The conversions between linear algebra and CGA objects are presented firstly, and then the CGA-based vertex blending method is introduced to enhance the realism of the finger joints. For the interaction aim, the collision detection is conducted by CGA calculation. With the results of collision detection and hand gesture recognition, the manipulation rules are provided, and the effectiveness of grasp, pinch, and hold is evaluated in the virtual environment finally.

## 2 Method for virtual hand modeling

For describing virtual hand modeling and interaction, the linear algebra methods have to introduce a large amount of parameters because of matrix production. Geometric algebra can operate the geometric objects via addition, subtraction, multiplication, and division [17], and conformal geometric algebra (CGA) is a uniformed geometric language independent of the coordinate frames with both covariates for geometric modeling and invariants for geometric calculation. With these features, CGA becomes a powerful tool in computer graphics and computer vision fields [18]. For mesh deformation, it could also interpolate the positions and orientations smoothly [19], and it reached interactive rates on surface models by GPU computation [20], which was implemented in virtual character simulations [21]. In human model animation, the grid distortion caused by the mesh deformation around the joint points was dealt by CGA for the knee joint motion [22]. Additionally, CGA was presented to determine intersection relationship between two geometric objects of different types in a unified manner [23]. With these bases above, CAG is introduced in this paper for virtual hand modeling and interaction.

### 2.1 Outline of conformal geometric algebra

Conformal geometric algebra provides the intuitive representation of various geometric objects and geometric transforms. In Euclidean space, the basis is  $1, e_1, e_2, e_3, e_{12}, e_{23}, e_{13}, e_{123}$ , and the highest grade is defined as the pseudoscalar  $I = e_1 \wedge e_2 \wedge \dots \wedge e_n$ , which is a multi-vector in the same grades. The geometric objects such as points, lines, planes, circles and spheres have similar expressions in multi-vector form. In this paper, all the descriptions of the geometric objects and transforms follow the symbol system from [24].

For the point, the base vector  $e_0$  is its origin and  $e_\infty$  is the infinity. Its conformal geometric algebra form is as (1).

$$X = x + \frac{1}{2}x^2e_\infty + e_0 \quad (1)$$

If the center is set as  $c$  and the radius is set as  $\rho$ , the sphere in conformal space is generated as (2).

$$S = c - \frac{1}{2}\rho^2e_\infty \quad (2)$$

The dual form of the subspace  $A$  can be described as (3)

$$A^* = AI^{-1} \quad (3)$$

The outer product of points  $A, B, C$ , and  $D$  the infinity generates a line as (4)

$$S^* = A \wedge B \wedge C \wedge D \quad (4)$$

The outer product of points  $A, B$ , and the infinity generates a line as (5).

$$L = A \wedge B \wedge e_\infty \quad (5)$$

The outer product of points  $A, B, C$ , and the infinity generates a plane as (6).

$$P = A \wedge B \wedge C \wedge e_\infty \quad (6)$$

The outer product of three points  $A, B$ , and  $C$  generates a cycle as (7)

$$C = A \wedge B \wedge C \quad (7)$$

Set  $l$  as the unit direction vector and  $\theta$  as the rotation angle. The Rotor  $R$  is obtained in Euler form as (8).

$$R = \exp\left(-\frac{\theta}{2}l\right) \quad (8)$$

Set  $R\tilde{R} = 1$ , so  $\tilde{R}$  is the inverse form of  $R$ . They are expressed as (9) and (10) in 3D Euclidean space.

$$R = u_0 + u_1e_{23} + u_2e_{13} + u_3e_{12} \quad (9)$$

$$\tilde{R} = u_0 - u_1e_{23} - u_2e_{13} - u_3e_{12} \quad (10)$$

The rotation of point  $X$  can be described as (12).

$$X' = RX\tilde{R} \quad (11)$$

In conformal geometric algebra, the geometric objects, including line, plane, circle, and sphere, all could be represented by the outer product of points  $X$  as (12) to facilitate the translation.

$$R(X_1 \wedge X_1 \wedge \dots \wedge X_n) \tilde{R} = (RX_1 \tilde{R}) \wedge (RX_1 \tilde{R}) \wedge \dots \wedge (RX_n \tilde{R}) \quad (12)$$

The translator is introduced similar to that of rotor as (13) in Euler form.

$$T = \exp\left(-\frac{e_\infty}{2}t\right) \quad (13)$$

Thus,  $X$  is transformed by  $T$  like (14).

$$X' = TX\tilde{T} \quad (14)$$

Motor  $M$  is a type of transform in which transformation  $T$  is applied to the rotor  $R$ . It defines the general motion as (15).

$$M = TR\tilde{T} = \exp\left(-\frac{\theta}{2}(l + e_\infty(t \cdot l))\right) \quad (15)$$

In addition, there are some other types of geometric relation, such as intersection between the geometric objects. With the extension of Morgan Rule, it can be deduced the dual form of intersection between  $A$  and  $B$  as (16).

$$(A \vee B)^* = A^* \wedge B^* \quad (16)$$

Compared with linear algebra, the coordinate frame only plays its role on the input of geometric objects while the transformation process is completely geometrical. In other words, the transformation is applied on the geometric objects directly rather than via analytical equations with coordinates.

## 2.2 Conversion between virtual reality objects and CGA objects

In virtual reality system, geometric objects are generally constructed by triangle patches that include color and illumination information for rendering these models. In this paper, CGA will be employed in the virtual hand modeling and manipulation interaction, but the rendering of current virtual reality system still depends on the graphics API with linear algebra style calculation. The conversion between virtual reality objects and CGA objects is required.

In CGA, the motion could be expressed as a motor which is decomposed as translator and rotor. The translator corresponds to the vector in 3D Euclidean space, but the rotor should be converted to a matrix. In CGA, the rotor  $R$  is expressed as (17).

$$R = u + ae_{23} + be_{31} + ce_{12} \quad (17)$$

The row vector in  $3 \times 3$  rotation matrix is obtained by applying the rotor  $R$  to  $e_1$ ,  $e_2$ , and  $e_3$ . When  $R$  is applied to  $e_1$ , it generates (18).

$$\begin{aligned} Re_1 \tilde{R} &= (u + ae_{23} + be_{31} + ce_{12})e_1(u + ae_{23} + be_{31} + ce_{12}) \\ &= (u^2 + a^2 - b^2 - c^2)e_1 + 2(ab - uc)e_2 + 2(ub + ac)e_3 \\ &= (1 - 2(b^2 + c^2))e_1 + 2(ab - uc)e_2 + 2(ub + ac)e_3 \end{aligned} \quad (18)$$

When  $R$  is applied to  $e_2$ , it generates (19).

$$\begin{aligned} Re_2 \tilde{R} &= (u + ae_{23} + be_{31} + ce_{12})e_2(u + ae_{23} + be_{31} + ce_{12}) \\ &= 2(ba + uc)e_1 + (1 - 2(a^2 + c^2))e_2 + 2(bc - ua)e_3 \end{aligned} \quad (19)$$

When  $R$  is applied to  $e_3$ , it generates (20).

$$\begin{aligned} Re_3 \tilde{R} &= (u + ae_{23} + be_{31} + ce_{12})e_3(u + ae_{23} + be_{31} + ce_{12}) \\ &= 2(zx - ub)e_1 + 2(bc + ua)e_2 + (1 - 2(a^2 + b^2))e_3 \end{aligned} \quad (20)$$

With the formulas (18) to (21), the rotation matrix for rotor  $R$  is obtained as below.

$$R_M = \begin{bmatrix} 1 - 2b^2 - 2c^2 & 2ab - 2uc & 2ac + 2ub \\ 2ba + 2uc & 1 - 2a^2 - 2c^2 & 2bc - 2ua \\ 2ca - 2ub & 2bc + 2ua & 1 - 2a^2 - 2b^2 \end{bmatrix} \quad (21)$$

According to formula (21), the motion could be converted to linear algebra calculation from CAG style. For the objects in virtual reality system, the expressions with coordinates would also be converted to CGA styles. The unit triangle is constructed by basic geometric elements such as point, line, and plane in that there is no direct description for triangles in CGA. The vertices of the triangle in 3D Euclidean space are regarded as the points of CGA directly. In formula (22), the point  $X(x, y, z)$  is expressed as the CGA objects utilizing its 3D coordinates.

$$X = e_0 + xe_1 + ye_2 + ze_3 + \frac{1}{2}(x^2 + y^2 + z^2)e_\infty \quad (22)$$

The edge AB of triangle ABC is described by the line on which it lies.

$$L_{AB} = A \wedge B \wedge e_\infty \quad (23)$$

The edge BC of triangle ABC is (24):

$$L_{BC} = B \wedge C \wedge e_\infty \quad (24)$$

The edge AC of triangle ABC is (25):

$$L_{AC} = A \wedge C \wedge e_\infty \quad (25)$$

The plane on which the triangles lay is also defined in CGA.

$$P_{ABC} = A \wedge B \wedge C \wedge e_{\infty} \quad (26)$$

In virtual reality system, the triangles could be constructed as by grouping its vertices, edges, and plane features with 3D coordinates of vertices.

```
Class Triangle{
  CGAPoint Vertex1., Vertex2., Vertex3.;
  CGALine EdgeLine1.,EdgeLine2.,EdgeLine3.;
  CGAPlane TrianglePlane;
}
```

With the definitions above, the CGA and linear algebra objects and motions could be converted to each other in modeling and interaction of the virtual hand.

### 2.3 Construction of virtual hand

Virtual hand simulates both the appearance and motion of the human hand, which requires the continuity of the hand model during the bending of finger segments. For rigid virtual hand model, the skin artifacts cannot be neglected on the finger joints. According to the physiological structure, the skin deformation on joints relates to the motions of both adjacent finger segments. Thus, the skin deformation is handled by adding the influences of different finger segments with the weighted motion, which is named skeleton subspace deformation method. It is a type of vertex blending of the skeletons. For the vertices  $v_1$  and  $v_2$ , the vertex blending is achieved by the joint transformation matrices  $M_1$  and  $M_2$  with the weights  $w_1$  and  $w_2$  shown in Fig. 1.

In Fig. 1,  $B_i$  transforms the coordinate of finger segment frame to the skin, and the transformation from the finger segment to the global frame is  $W_i$ . Thus, the transformation to the vertices of finger segment  $i$  is  $M_i = B_i^{-1} W_i$ . Generally, the formula of vertex blending is (27).

$$v = \left( \sum_{i=1}^n (w_i M_i) \right) v_i \quad (27)$$

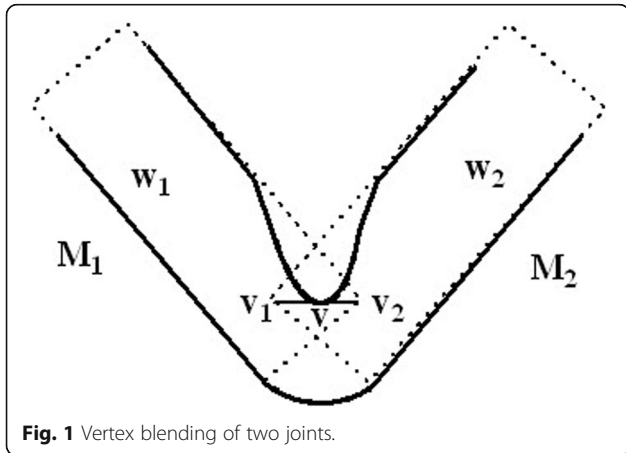


Fig. 1 Vertex blending of two joints.

Attached to  $n$  finger segments, the total weight of vertices is as (28).

$$\sum_{i=1}^n w_i = w_1 + w_1 + \dots w_n = 1 \quad (28)$$

The linear method for vertex blending is simple and less time consuming, but for some finger joints with large rotation angles, the artifacts are unavoidable no matter how the weights are adjusted. It is because that the linear combination of matrices introduces extra shearing and scaling terms, and the vertex blending is not really a linear one that reduces the realism of the deformation. This deflection is caused by the rotation terms of the matrix in non-linear space without smoothing transformation. If the rotation parameters can be operated directly rather than via the matrix, the artifacts of linear vertex blending will be improved. For the intuition of geometric transformation in CGA, the vertex blending based on CGA is presented in this paper.

The characteristics of CGA should be analyzed firstly for vertex blending. The neighbor segments of the vertices in vertex blending rotate in the local frame of joint referencing an axle along some angle. If the axle goes through the origin, the transformation of the local frame of joint is just the translation, thus  $M_i$  can be regarded as the compound motion by translating the rotation to the axle. The transformation in CGA can not only apply to geometric objects but also to other transformations. It means that the compound motion is a type of transformation to other translators. If the translation of the rotation to the line  $L$  in  $\theta$  is expressed as  $T(R_{L\theta})$ , and it assumes that  $R_{L\theta} = \exp(-\frac{\theta}{2}L)$ , the (29) could be deduced by (15).

$$\begin{aligned} R_{L\theta} &= \left(1 + \frac{e_{\infty} t}{2}\right) \exp\left(-\frac{\theta}{2}L\right) \left(1 - \frac{e_{\infty} t}{2}\right) \\ &= \exp\left(-\frac{\theta}{2}(L + e_{\infty}(t \cdot L))\right) \end{aligned} \quad (29)$$

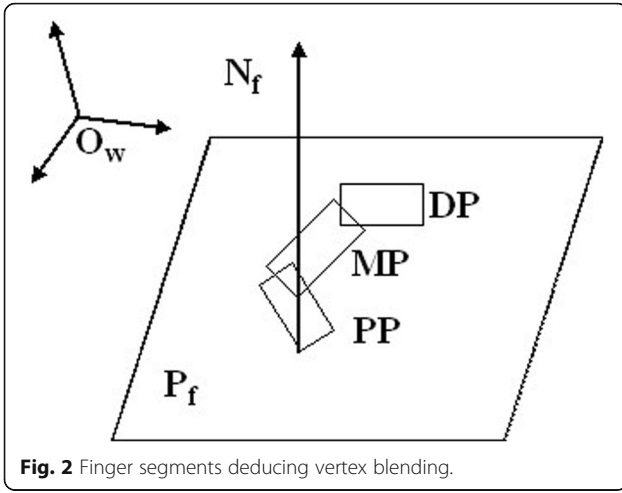
Set  $\hat{L} = L + e_{\infty}(t \cdot L)$ , the transformation of joint  $i$  is as formula (30).

$$M_i = \exp\left(-\frac{\theta_i}{2}\hat{L}_i\right) \quad (30)$$

The linear combination of transformations is that of the motors. With exponential expression, the rotation angle  $\theta$  is on the exponential part. Thus, the vertex blending based on CAG is as formula (31).

$$M_v = \sum_{i=1}^n w_i M_i = \exp\left(\sum_{i=1}^n \left(-w_i \frac{\theta_i}{2}\right) \hat{L}_i\right) \quad (31)$$

$w_i \theta_i$  intuitively represents the linear transformation style for rotation and eliminates the extra scaling and shearing



**Fig. 2** Finger segments deducing vertex blending.

effects caused by a matrix operation. In the virtual hand model, the three finger segments deducing vertex blending are on the same plane, and it can be shown in Fig. 2.

The segment of the finger could be named as DP, MP, and PP according to the position of them from the palm to the fingertip, and the corresponding joints are DIP, MIP, and PIP respectively [25]. The skin on PIP and DIP appears artifacts because of the large bending angles of these joints. Set the plane on which all segments from the same finger locates as  $P_f$ , and the normal of the plane as  $N_f$ . DP, MP, and PP all rotate on the plane  $P_f$  to the axle  $N_f$  with the vertex blending on DIP  $v_{DIP}$  by  $M_{DP}$  and  $M_{MP}$  and that on PIP  $v_{PIP}$  by  $M_{MP}$  and  $M_{PP}$ . Deduced by formula (31), the two vertex blending expressions are as formulas (32) and (33).

$$\begin{aligned} M_{DIP} &= w_{DP}M_{DP} + w_{MP}M_{MP} \\ &= \exp\left(-w_{DP}\frac{\theta_{DP}}{2}N_f\right) \\ &\quad + \exp\left(-w_{MP}\frac{\theta_{MP}}{2}N_f\right) \end{aligned} \quad (32)$$

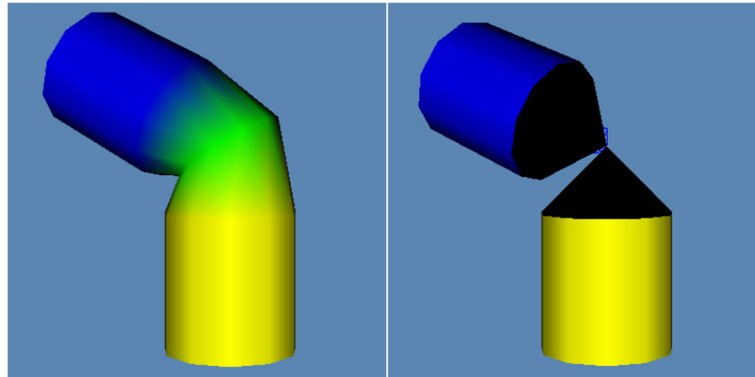
$$\begin{aligned} M_{PIP} &= w_{PP}M_{PP} + w_{MP}M_{MP} \\ &= \exp\left(-w_{PP}\frac{\theta_{PP}}{2}N_f\right) \\ &\quad + \exp\left(-w_{MP}\frac{\theta_{MP}}{2}N_f\right) \end{aligned} \quad (33)$$

Because the three finger segments are on the same plane, the vertex blending is simplified by the same referenced axle. For testing, the vertex blending for two joints is implemented. For cylinder shape segments, the rendering result of vertex blending is shown in Fig. 3 on the left side, compared with the one on the right side without vertex blending.

For the whole hand, the triangle patches and the transformations are converted to CGA style so that the vertex blending could be applied by CGA calculation. The comparison of the virtual hand without vertex blending, linear vertex blending, and vertex blending based on CGA are demonstrated in Figs. 4 and 5. Among them, the first group of models cracks on joints obviously, the second ones are separated by the unexpected shearing and scaling, and these artifacts are avoided by CAG-based vertex blending like the third ones.

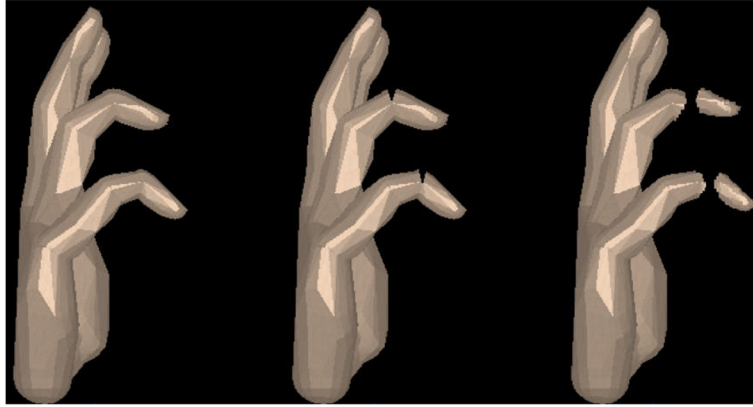
### 3 Method for proposed manipulation rules of virtual hand

As the response of the interaction between virtual hand and the objects in the virtual environment, collision detection is described as the intersection of geometric elements in CGA, so the virtual hand and the manipulated objects can be decomposed as geometric elements on a different level for such intersection tests. Besides, the shapes of virtual hand also correspond to various gestures. With the defined gestures and the results of collision detection, the rules of virtual hand manipulation are established.



**Fig. 3** Vertex blending of two joints.





**Fig. 4** Comparison of virtual hand models on DP joints.

### 3.1 Collision detection based on conformal geometric algebra

The virtual hand and objects in the virtual environment are generally triangulated patches, so the collision of them could be regarded as the intersection of triangles. However, the computing is very intensive if all the intersections among these triangles are tested. Therefore, the layered bonding box is also introduced here. In CGA, the intersection of the two geometric elements is represented as formula (34).

$$B = X \vee Y = (IX) \cdot Y \quad (34)$$

For the pseudoscalar  $I$ , it has the representation as formula (35).

$$I = e_1 \wedge e_2 \wedge e_3 \wedge e_0 \wedge e_\infty = e_1 e_2 e_3 e_0 e_\infty \quad (35)$$

It is inferred that  $B$  is the linear combination of different basis vectors as formula (36).

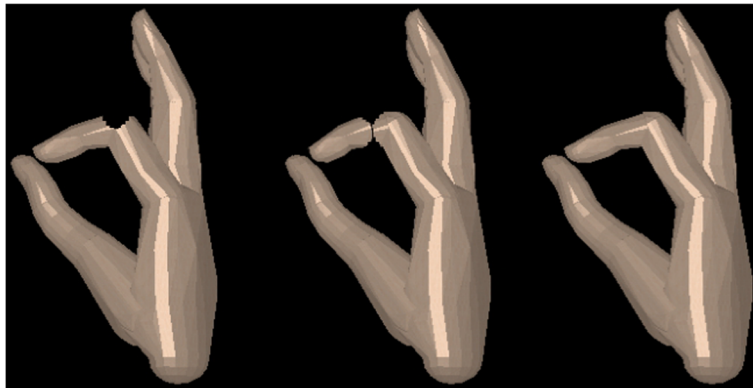
$$B = \beta_0 + \beta_1 e_1 + \beta_{12} e_1 e_2 + \cdots + \beta_{12345} e_1 e_2 e_3 e_0 e_\infty \quad (36)$$

$B^2$  is the criteria of the intersection of the two geometric objects [26]. When  $B^2 > 0$ ,  $X$  and  $Y$  have at least two intersecting points; when  $B^2 = 0$ ,  $X$  and  $Y$  have one intersecting point; when  $B^2 < 0$ ,  $X$  and  $Y$  have no intersecting point. The intersections of lines, planes, and spheres are discussed in this section.

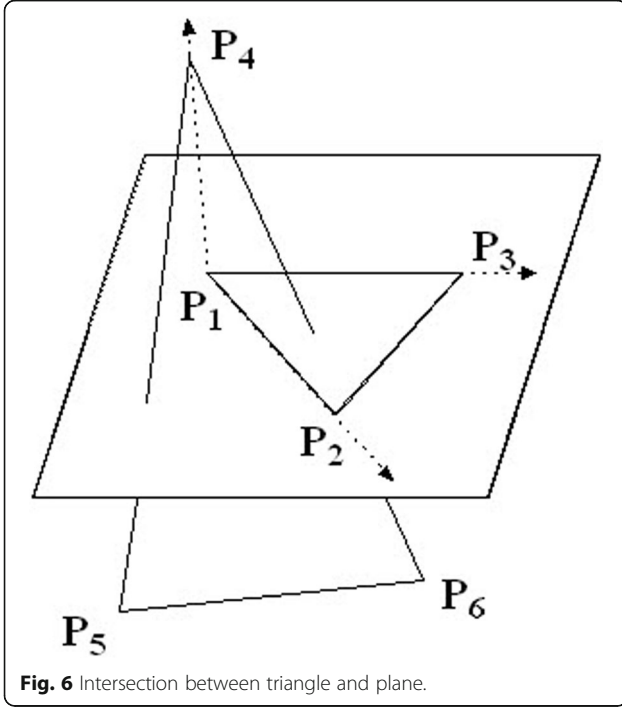
Firstly, the intersections of lines are presented. Set the lines as  $L_1 = P_1 \wedge P_2 \wedge e_\infty$  and  $L_2 = P_3 \wedge P_4 \wedge e_\infty$ , and the end points as  $P_1 = a_1 e_1 + a_2 e_2 + a_3 e_3$ ,  $P_2 = b_1 e_1 + b_2 e_2 + b_3 e_3$ ,  $P_3 = c_1 e_1 + c_2 e_2 + c_3 e_3$ , and  $P_4 = d_1 e_1 + d_2 e_2 + d_3 e_3$ . The intersection of  $L_1$  and  $L_2$  is expressed as formula (37).

$$B = L_1 \vee L_2 = (IL_1) \cdot L_2 \quad (37)$$

According to  $P_1$  and  $P_4$ ,  $B$  has the expansion style as formula (38).



**Fig. 5** Comparison of virtual hand models on MP joints.



**Fig. 6** Intersection between triangle and plane.

$$\begin{aligned} \alpha_1 &= \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}, \alpha_2 = \begin{vmatrix} a_2 & b_2 \\ a_3 & b_3 \end{vmatrix}, \alpha_3 = \begin{vmatrix} a_3 & b_3 \\ a_1 & b_1 \end{vmatrix} \\ \beta_1 &= \begin{vmatrix} c_1 & d_1 \\ c_2 & d_2 \end{vmatrix}, \beta_2 = \begin{vmatrix} c_2 & d_2 \\ c_3 & d_3 \end{vmatrix}, \beta_3 = \begin{vmatrix} c_3 & d_3 \\ c_1 & d_1 \end{vmatrix} \\ \alpha_4 &= a_1 - b_1, \alpha_5 = a_2 - b_2, \alpha_6 = a_3 - b_3 \\ \beta_4 &= c_1 - d_1, \beta_5 = c_2 - d_2, \beta_6 = c_3 - d_3 \end{aligned} \quad (38)$$

Combining the variables above,  $B^2$  is expressed as formula (39).

$$B^2 = \alpha_1\beta_6 + \alpha_2\beta_4 + \alpha_3\beta_5 + \alpha_5\beta_3 + \alpha_6\beta_1 \quad (39)$$

It could be expressed the intersection results as algebra style of the points parameters on the line. By the same mean, the intersections between line and plane can also be described. Set the line as  $L = P_1 \wedge P_2 \wedge e_\infty$  and the plane as  $Pi = P_3 \wedge P_4 \wedge P_5 \wedge e_\infty$ . The intersection of them could be as formula (40).

$$B = \Pi \vee L = (\Pi \Pi) \cdot L \quad (40)$$

Similarly, construct the points  $P_6$ ,  $P_7$ , and  $P_8$  as the spheres  $S_1 = P_1 \wedge P_2 \wedge P_3 \wedge P_4$  and  $S_2 = P_5 \wedge P_6 \wedge P_7 \wedge P_8$ . The intersection of  $S_1$  and  $S_2$  is as formula (41).

$$B = S_1 \vee S_2 = (IS_1) \cdot S_2 \quad (41)$$

The intersections between line and plane or spheres could also be represented as algebra style. Additionally, the intersection of spheres is as formula (42).

$$B = \Pi \vee S = (\Pi \Pi) \cdot S \quad (42)$$

Based on the expressions above, the detection between triangle surfaces, sphere bonding boxes, and that between triangles surfaces and sphere bonding boxes are implemented. For the intersections between triangle patches, it could be simplified as the intersection between the three edges and the other triangle patch. If any edge intersects with the patch, the two triangles patches collide. It is shown in Fig. 6 the intersection of the triangles  $P_4P_5P_6$  and  $P_1P_2P_3$ .

The edges of  $P_4P_5P_6$  are line sections rather than lines. If  $P_4$  and  $P_5$  locate on the same side of the plane, the intersection of  $P_4P_5$  and the plane is invalid. In geometric algebra, the tri-vector represents the volume. Set the vectors for  $P_1P_2$ ,  $P_1P_3$ , and  $P_1P_4$  as  $a = a_1e_1 + a_2e_2 + a_3e_3$ ,  $b = b_1e_1 + b_2e_2 + b_3e_3$ , and  $c = c_1e_1 + c_2e_2 + c_3e_3$  respectively. The tri-vector is calculated by formula (43).

$$a \wedge b \wedge c = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} e_{123} \quad (43)$$

If the end of the line section  $P_4$  is on the surface, the coefficient of  $e_{123}$  in formula (43) is 0. It could be inferred whether  $P_5$  is on the plane in the same method. The relationship between  $P_4P_5$  and other edges or the plane are also inferred.

### 3.2 Establishment of the virtual hand manipulation rules

The manipulation rules are required for the interaction of virtual hand. Driven by the motion data input to the virtual hand, the virtual hand has different shapes that can be projected to two orthogonal planes, and the gestures



**Fig. 7** Definition of manipulation gestures.

**Table 1** Visible states of gestures on two orthogonal planes

Finger segment	Grasp	Pinch	Hold
Thumb PP	Y/N	Y/N	Y/N
Thumb MP	Y/Y	Y/N	Y/Y
Thumb DP	Y/N	Y/N	Y/N
Index PP	Y/Y	Y/Y	Y/Y
Index MP	Y/Y	Y/Y	Y/Y
Index DP	Y/Y	Y/N	Y/N
Middle PP	Y/Y	Y/Y	N/Y
Middle MP	Y/Y	Y/Y	Y/Y
Middle DP	Y/Y	N/N	Y/N

are obtained. With the meaningful gestures and the collision detection results, the manipulation is validated and then the virtual hand manipulation rules are established.

For certain gesture of virtual hand manipulation, the angles between finger segments and that between fingers keep steady. Therefore, the shapes of virtual hand could be divided into free and manipulation states. For the angle between thumb and other fingers, if the variation is less than the threshold  $\delta_{\text{var}}$  in a  $2n$  frame, the state of virtual hand is determined by formula (44).

$$\frac{\phi_{ij}^{t+n} - \phi_{ij}^t}{\phi_{ij}^t - \phi_{ij}^{t-n}} < \delta_{\text{var}} \quad (44)$$

When the virtual hand state is determined, the gesture is judged by other hand motion parameters. The physiological constraints are considered here because the motions of different finger segments relate to each other. It assumes that the manipulation state starts when the angles MIP reaches a quarter of its maximum range. If these angles have the tendency of becoming smaller, it will be the release state. Although there is still a large number of gestures, the most frequently used three gestures are defined in this paper, which are grasp, pinch, and hold as shown in Fig. 7.

In Fig. 7, the gaps between thumb and index appear from wide to narrow, which adjust to the manipulated objects with different dimensions. In the virtual environment, if the finger segments are projected to the orthogonal planes

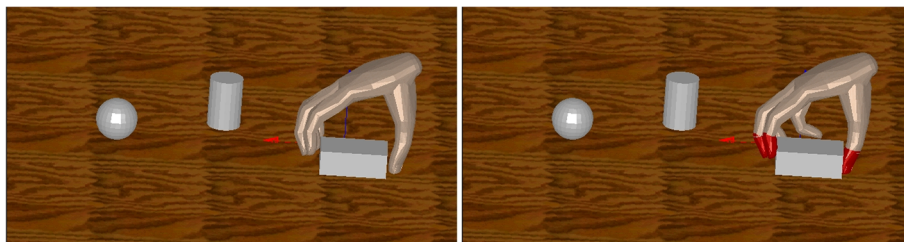
around the manipulation space, the visible states of the finger segments will characterize the gesture of the virtual hand. Because the ring and little finger rarely attend the three manipulations above, the state collection is focused on the thumb, the index, and the middle finger. If  $Y$  and  $N$  mean visible or not, the state combinations for grasp, pinch, and hold are shown in Table 1.

According to the visible states above, the gestures of the virtual hand are recognized. In the virtual environment, whether the gesture represents meaningful virtual manipulation should be further examined by the virtual hand shape. The human hand could manipulate objects because of its sense of touch and the envelope of fingers to the objects. Without the tactile feedback, the friction cone is introduced for the manipulation validation. There are two assumptions for the success manipulation. One is that the thumb must attend to the manipulation, and the other is that the angle of friction is larger than the threshold, which is the angle between the connection line  $L_{tf} = P_t \wedge P_f \wedge e_{\infty}$  which goes through the colliding point of thumb  $P_t$  and that of the other finger  $P_f$ . The line  $N_c$  is the normal of the contact point. The angle  $\theta_f$  is calculated by the formula (45).

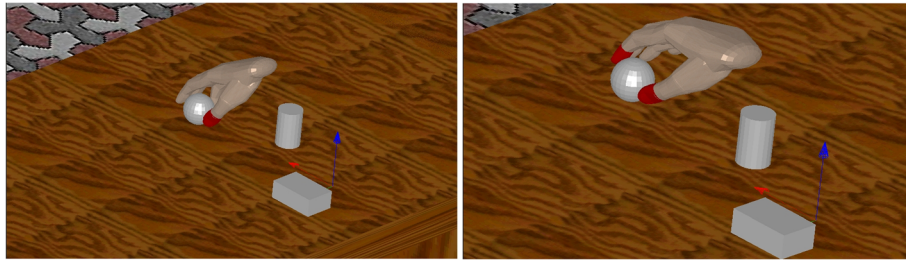
$$\theta_f = \arccos \frac{L_{tf}^* \cdot N_c^*}{\|L_{tf}^*\| \|N_c^*\|} \quad (45)$$

Combining the collision detection results with the friction cone, the state of virtual hand could be obtained by the following five steps:

- (1) Input the virtual hand and the manipulated object.
- (2) If the hand collides with the bounding box of the object, the intersections of the triangle surfaces of them are judged.
- (3) For the triangle patches of the virtual hand model, if the number of intersection points is larger than 2, keep the state of the hand.
- (4) If the triangle belongs to the thumb, calculate the angles of friction and compare them with the threshold.
- (5) Return the state of the virtual hand.

**Fig. 8** Effectiveness evaluation of grasp manipulation.





**Fig. 9** Effectiveness evaluation of pinch manipulation.

The rules of effective virtual hand manipulation are summarized based on the state of the virtual hand.

- (1) The gesture of the virtual hand could be recognized by analyzing the visible states of the finger segments.
- (2) The virtual hand contact with the manipulated object, and the number of contact points is larger than 2.
- (3) The contact must occur on the DIP of the thumb.
- (4) If the 1–3 are satisfied, the angle of friction is larger than the threshold.

#### 4 Experimental results and discussion

With the rules mentioned in the former sections, the manipulations of grasp, pinch, and hold are tested. After the gesture recognition, the collision detection and the effectiveness are evaluated, in which the collision is reminded by red color. In Fig. 8, the collision does not occur in the left scene (a), so the grasp is invalid, while the thumb, index, and middle finger all collide with the cuboid in the right scene (b), and the friction angles are larger than the threshold, so the grasp is successful.

In Fig. 9, only the DP collides with the sphere, and the contact points less than 2, so it is not a valid pinch in the left scene (a), while the thumb, index, and middle finger all collide with the sphere in the right scene (b), and the friction angles are larger than the threshold, the pinch is successful.

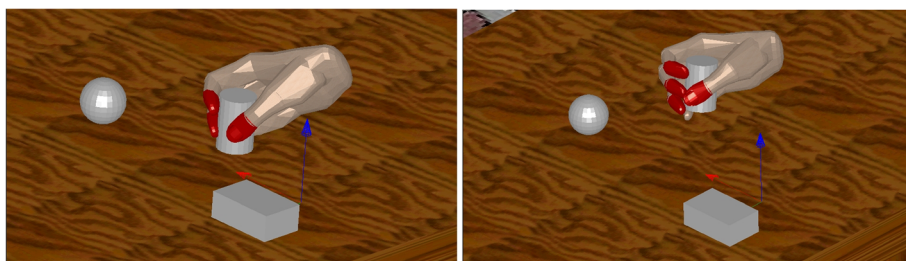
In Fig. 10, both the thumb and the index collide with the cylinder, but the angle of friction is smaller than the threshold, so it is an invalid hold in the left

scene (a), while all the conditions are fulfilled in the right scene (b), so the hold is successful. Thus, the manipulation rules proposed in this paper provides credible results for grasp, pinch, and hold manipulations tests.

Both the CGA geometric objects and CGA transforms are provided to model the virtual hand and its manipulation. On the hand modeling aspect, the hand model based on CGA is not merely constructed for the deformation in animations but for collision detection, which is the basis for interactions in the virtual environment. On the collision detection aspect, the layered bonding boxes are also kept rather than calculating CGA geometric objects intersections directly. On the manipulation aspect, the collision detection and angle of friction are combined by CGA method to establish the manipulation rules for the three typical gestures, and it finally integrates all the elements of virtual hand interaction in a unified framework.

#### 5 Conclusions

According to the requirements of human machine interaction, the virtual hand model is constructed by converting the objects between virtual reality system and the CGA ones. Utilized the vertex blending based on CGA, the artifacts of the finger joints are improved which enhances the realism of the virtual hand. Meanwhile, the collision detection between the virtual hand and the manipulated objects is implemented by CGA method. With the gesture recognition results, the



**Fig. 10** Effectiveness evaluation of hold manipulation.

manipulation rules for grasp, pinch, and hold gestures are established, and the effectiveness evaluations are finally conducted in the virtual environment. To obtain better real-time performance, the parallel computations based on OpenCL will be considered in the future work.

#### Abbreviations

CGA: Conformal geometric algebra; GPU: Graphics processing unit; API: Application programming interface; MIP: Mobile instant pages; DP: Displayport

#### Acknowledgements

The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

#### Introduction of all authors

##### Author1

Chen Peng is currently a lecturer at School of Mechanical Engineering, Southwest Jiaotong University, China. His research interests include virtual design and manufacturing, human machine interaction, and point cloud data processing.

##### Author2

Hu Yong is currently studying for his master degree in mechanical engineering in Southwest Jiaotong University. His research interests include virtual design and human machine interaction.

##### Author3

Yang Fengwei is currently studying for his master degree in mechanical engineering in Southwest Jiaotong University. His research interests include virtual design and human machine interaction.

#### Funding

The work is supported by National Natural Science Foundation of China (Grant No. 51305368) and Fundamental Research Funds for the Central Universities (Grant No. 2682017CX036).

#### Availability of data and materials

We can provide the data.

#### Authors' contributions

All authors attend the discussion of the work described in this paper. CP wrote the first version of this paper. HY developed the improved vision of the related programs. YF edited and revised the paper according to the updated programs. All authors read and approved the final manuscript.

#### Ethics approval and consent to participate

Approved.

#### Consent for publication

Approved.

#### Competing interests

The authors declare that they have no competing interests. We confirm that the content of the manuscript has not been published or submitted for publication elsewhere.

#### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 27 June 2018 Accepted: 10 August 2018

Published online: 20 August 2018

#### References

1. J. McDonald, J. Toro, K. Alkoby, A. Berthiaume, R. Carter, Approximating polyhedra with spheres for time-critical collision detection. *Vis. Comput.* **17**(3), 158–166 (2001)
2. L. Moccozet, N. M. Thalmann, Dirichlet free-form deformations and their application to hand simulation. In *Computer Animation 97. USA* (1997)
3. T. Kurihara, N. Miyata, Modeling deformable human hands from medical images. In *ACM SIGGRAPH Symposium on Computer Animation 2004*, New York (2004)
4. J. P. Lewis, M. Cordner, N. Fong, Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, (2000)
5. Kavan L. Real-time Skeletal Animation. PhD thesis, Czech Technical University, Prague, Czech (2007)
6. P. M. Hubbard, Approximating polyhedra with spheres for time-critical collision detection. *ACM Trans. Graphics* **15**(3), 179–210 (1996)
7. J. T. Klosowski, M. Held, J. S. B. Mitchell, H. Sowizral, K. Zikan, Efficient collision detection using bounding volume hierarchies of k-dops. *IEEE Trans. Vis. Comput. Graph.* **4**(1), 21–36 (1998)
8. S. Redon, A. Kheddar, S. Coquillart, Fast continuous collision detection between rigid bodies. *Comput. Graph. Forum.* **21**(3), 279–287 (2002)
9. M. Lee, R. Green, M. Billinghurst, 3rd natural hand interaction for AR applications. In *23rd International Conference on Image and Vision Computing, USA* (2008)
10. H. Wan, F. Chen, X. Han, A 4-layer flexible virtual hand model for haptic interaction (In *International Conference on Virtual Environments, Human-Computer Interfaces and Measurements Systems, USA*, 2009)
11. D. Manocha, J. Demmel, Algorithms for intersecting parametric and algebraic curves. In *Proceedings of the Conference on Graphics Interface 92*, San Francisco, (1992)
12. A. Sud, N. Govindaraju, R. Gayle, I. Kabul, D. Manocha, Fast proximity computation among deformable models using discrete voronoi diagrams: Implementation details. In *ACM SIGGRAPH 2006 Sketches*, New York (2006)
13. F. Liu, Y. J. Kim, Exact and adaptive signed distance fields computation for rigid and deformable models on gpus. *IEEE Trans. Vis. Comput. Graph.* **20**(5), 714C–725 (2014)
14. M. Prachyabrued, C.W. Borst, *Dropping the ball: Releasing a virtual grasp* (In *IEEE Symposium on Digital Object Identifier, USA*, 2011)
15. D. Holz, S. Ullrich, M. Wolter, T. Kuhlen, J. Herder, 2008. Multi-contact grasp interaction for virtual environments. *J. Virt. Real. Broadcast.* **5**(7) (2008)
16. M. Moehring, B. Froehlich, Enabling functional validation of virtual cars through natural interaction metaphors. In *IEEE Virtual Reality 2010, USA* (2010)
17. Hestenes D. The design of linear algebra and geometry. *Acta Appl. Mathematicae.* **23**(1), 65–93 (1991)
18. Wareham R., Cameron J. & Lasenby J. Applications of conformal geometric algebra in computer vision and graphics. *Lecture Notes Comput. Sci.* **3519**(1), 329–349 (2005)
19. M. C. L. Bel'on, Applications of conformal geometric algebra in mesh deformation. In *Proceedings of the, XXVI Conference on Graphics, Patterns and Images, Washington, DC. USA* **2013** (2013)
20. M. C. L. Bel'on, D. Hildenbrand, Practical geometric modeling using geometric algebra motors. *Adv Appl. Clifford Algebras*, 1–15 (2017)
21. M. Papaefthymiou, D. Hildenbrand, G. Papagiannakis, An inclusive conformal geometric algebra GPU animation interpolation and deformation algorithm. *Vis. Comput.* **32**(6), 751–759 (2016)
22. L. Hu, K. Hao, X. Huang, Y. Ding, *Individual Three-Dimensional Human Model Animation Based on Conformal Geometric Algebra* (Springer, Berlin Heidelberg, 2012)
23. F. Zhang, X. Jiang, X. Zhang, Y. Du, Z. Wang, R. Liu, Unified spatial intersection algorithms based on conformal geometric algebra. *Mathematical Problems in Engineering* **2016**, 7412373 (2016)
24. E. Bayro-Corrochano, *Geometric Computing* (Springer Press, New York, 2010)
25. Rhee T. Articulated human body deformation from in-vivo three-dimensional image scans. PhD thesis (University of Southern California, USA 2008)
26. E. Roa, V. Theokisto, M. Fairén, I. Navazo, GPU Collision Detection in Conformal Geometric Space. *Am. Symposium Comput. Graph.* **2011**, 153–156 (2011)