

SO SÁNH GEOMETRIC ALGEBRA VÀ MA TRẬN TRONG THUẬT TOÁN QUAY VẬT THỂ 3D

COMPARISONS BETWEEN GEOMETRIC ALGEBRA AND MATRIX IN 3D OBJECT ROTATION ALGORITHM

Phạm Minh Tuấn

Trường Đại học Bách khoa, Đại học Đà Nẵng

Email: pmtuan@dut.udn.vn

TÓM TẮT

Quay vật thể trong không gian 3 chiều (3D) là một trong những kỹ thuật quan trọng trong lĩnh vực đồ họa máy tính (computer graphics). Kỹ thuật quay 3D được ứng dụng rộng rãi hiện nay như trong xử lý ảnh, thiết kế vật thể 3D, hay xây dựng phim 3D... Những nghiên cứu về cách quay vật thể trước đây thường sử dụng việc nhân ma trận. Muốn quay một vật theo một trục bất kỳ trong không gian 3 chiều chúng ta cần 3 ma trận 3×3 để xử lý. Điều này đồng nghĩa với việc chúng ta cần tới 27 tham số trong phép quay. Dẫn tới việc dễ làm sai số trong các xử lý đòi hỏi độ chính xác cao. Bài báo này giới thiệu phương pháp sử dụng Geometric Algebra nhằm thực hiện việc xử lý quay vật thể trong không gian 3 chiều. Với phương pháp sử dụng Geometric Algebra này, chúng ta chỉ cần 4 tham số. Nhờ đó giúp cho việc xử lý quay vật thể được chính xác và nhanh hơn. Thực nghiệm cũng cho thấy kết quả của phương pháp sử dụng Geometric Algebra tốt hơn so với các phương pháp trước đó.

Từ khóa: đồ họa máy tính; không gian 3 chiều; quay; geometric algebra; quaternion; số phức

ABSTRACT

Object rotation in 3-dimensional space (3D) is one of the most useful techniques in the field of computer graphics. 3D object rotation is now widely used in image processing such as in designing 3D objects or 3D construction... The conventional rotation method is using the matrix multiplication. To rotate an object around an axis in any 3-dimensional space we need three 3×3 matrices. This means that we need 27 parameters. In this way, it is easy to get wrong in the process of requiring high accuracy. This paper uses the Geometric Algebra to rotate 3D objects. By using the Geometric Algebra, we only need 4 parameters. This method helps to rotate objects exactly and quickly. Experimental results also show that the method using Geometric Algebra is better than the conventional method.

Key words: computer graphics; 3D; rotation; geometric algebra; quaternion; complex number

1. Đặt vấn đề

Hiện nay, kỹ thuật đồ họa máy tính [1,2] được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau như xử lý ảnh, thiết kế vật thể trong game 3D, hay xây dựng phim 3D. Việc quay vật thể trong không gian là một trong những kỹ thuật cơ bản và thiết yếu nhất của lĩnh vực đồ họa máy tính. Nó giúp cho con người có thể biết được hình dáng của vật thể trong không gian 3D một cách chính xác thông qua màn hình máy tính. Trong các nghiên cứu trước đây, thông thường khi quay một vật quanh một trục bất kỳ trong không gian 3D, người ta sử dụng ít nhất 3 ma trận 3×3 để thực hiện phép quay. Điều đó đồng nghĩa cần phải có 27 tham số cho một phép quay. Đối với các xử lý đòi hỏi độ chính xác cao, việc có quá nhiều tham số hay số lượng tính toán

nhiều sẽ gây nên sai số không chấp nhận được. Nghiên cứu trong bài báo này áp dụng Geometric algebra để thực hiện phép quay giúp cho việc xử lý quay trong không gian 3D có độ chính xác cao hơn.

Geometric algebra (GA) hay còn gọi là Clifford algebra, là một mô hình toán học phát triển từ sự kết hợp giữa đại số và hình học không gian [3,4,5]. GA có thể biểu diễn các vector hay các mối liên kết của chúng trong không gian 3D một cách đơn giản và chính xác. Vì vậy GA bắt đầu được các nhà nghiên cứu trong lĩnh vực công nghệ thông tin quan tâm tới. Có rất nhiều ví dụ điển hình cho việc áp dụng thành công geometric algebra như là: mô hình xử lý tín hiệu, xử lý ảnh sử dụng không gian GA như số phức [6,7,8] hay quaternions [9,10,11]. Ta cũng có thể

thấy với sự kết hợp GA đã tạo ra được những phương pháp học máy hữu hiệu như mô hình GA-valued neural network [12] hay trích chọn đặc tính từ GA [13,14].

Bài báo này sử dụng phép quay trong GA để áp vào việc quay vật thể 3D trong lĩnh vực đồ họa máy tính. Khi cho 2 vectors đồng gốc thể hiện tham số của một phép quay bất kỳ trong không gian 3D, bài báo chuyển đổi 2 vectors đó thành 2 vectors trong không gian GA và tìm tích của 2 vector đó trong không gian GA. Kết quả thu được sẽ là 1 quaternion thể hiện một tham số trong phép quay của sử dụng không gian GA. Vì mỗi quaternion chỉ có 4 tham số nên so với phép quay sử dụng ma trận quay với $3 \times 3 \times 3 = 27$ tham số thì phép quay sử dụng GA sẽ chính xác hơn.

Bài báo này gồm có 5 phần. Phần 1 trình bày vấn đề của bài toán trong phép quay. Phần 2 của bài báo giới thiệu những nghiên cứu trước đây liên quan phép quay trong không gian. Phần 3 trình bày phương pháp quay sử dụng GA. Phần 4 trình bày đóng góp chính của bài báo đó là thực nghiệm để chứng minh sự ưu việt của phương pháp sử dụng GA so với phương pháp quay sử dụng ma trận. Phần 5 kết luận nội dung và kết quả đạt được.

2. Phương pháp quay trong không gian

Phương pháp quay trong không gian là một trong những phép biến hình quan trọng trong đồ họa máy tính. Có rất nhiều phép biến hình như là: tịnh tiến, phóng to, thu nhỏ, quay, phản xạ, làm nghiêng (skew)... Tuy nhiên phần này trong bài báo chỉ chú trọng phân tích phép quay trong không gian 2 chiều (2D) và 3 chiều (3D).

2.1. Quay trong không gian 2D

Khi quay điểm $X = (x, y)$ ngược chiều kim đồng hồ một góc θ quanh điểm gốc $O = (0, 0)$ ta được kết quả là $Y = (x', y')$ thỏa mãn công thức sau:

$$\begin{cases} x' = x \cos \theta - y \sin \theta, \\ y' = x \sin \theta + y \cos \theta. \end{cases} \quad (1)$$

Ở đây, giả sử ta biểu diễn điểm X dưới dạng vector $X = \begin{bmatrix} x \\ y \end{bmatrix}$, thì phép quay có thể được

biểu diễn dưới dạng tích của một ma trận và một vector như sau:

$$Y = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \quad (2)$$

Để dàng thấy được trong không gian 2D thì ta cần $2 \times 2 = 4$ tham số (phần tử) của ma trận và 4 phép nhân để thực hiện phép quay.

2.2. Quay trong không gian 3D

Phép quay bất kỳ trong không gian 3D chính là sự kết hợp 3 cách quay theo 3 trục Ox , Oy , Oz trong không gian $Oxyz$. Cũng như việc quay trong không gian 2D, các phép quay trên các trục trong không gian 3D được biểu diễn dưới dạng tích của một ma trận và một vector như sau:

Quay quanh trục Ox một góc α :

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_x(\alpha)X. \quad (3)$$

Quay quanh trục Oy một góc β :

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_y(\beta)X. \quad (4)$$

Quay quanh trục Oz một góc γ :

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_z(\gamma)X. \quad (5)$$

Như vậy, để quay một góc bất kỳ trong không gian 3D, ta có thể biểu diễn dưới dạng tích của 3 ma trận và 1 vector như sau:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_x(\alpha)R_y(\beta)R_z(\gamma) \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (6)$$

Ta có thể thấy được rằng trong không gian 3D, ta cần $3 \times 3 \times 3 = 27$ tham số (phần tử) của ma trận để thực hiện phép quay.

3. Phương pháp quay sử dụng Geometric algebra

3.1. Geometric algebra

Geometric algebra (GA) hay còn gọi là Clifford algebra, là một mô hình toán học phát

triển từ sự kết hợp giữa đại số và hình học không gian [3,4,5]. GA có thể biểu diễn các vector hay các mối liên kết của chúng trong không gian 3D một cách đơn giản và chính xác. GA định nghĩa không gian bằng cách định nghĩa tích của $p + q$ cơ sở trực giao $\mathcal{O} = \{\mathbf{e}_1, \dots, \mathbf{e}_p, \mathbf{e}_{p+1}, \dots, \mathbf{e}_{p+q}\}$ trong không gian như sau:

$$\mathbf{e}_i \mathbf{e}_j = \begin{cases} 1 & i = j \in \{1, \dots, p\} \\ -1 & i = j \in \{p+1, \dots, p+q\} \\ -\mathbf{e}_j \mathbf{e}_i & i \neq j \end{cases} \quad (7)$$

Bài báo này biểu diễn không gian được định nghĩa bởi $\mathcal{O}$ là $\mathcal{G}_{p,q}$. Như vậy không gian thực m chiều $\mathcal{R}^m$ có thể được biểu diễn bởi $\mathcal{G}_{m,0}$ trong mô hình GA.

Từ định nghĩa của GA, ta có tích hình học (geometric product) của 2 vectors $\{\mathbf{a}_l = \sum_{i=1}^m a_{li} \mathbf{e}_i; l = 1, 2\}$ trong $\mathcal{R}^m$ là:

$$\begin{aligned} \mathbf{a}_1 \mathbf{a}_2 &= \sum_{i=1}^m a_{1i} a_{2i} \\ &+ \sum_{i=1}^{m-1} \sum_{j=i+1}^m (a_{1i} a_{2j} - a_{2i} a_{1j}) \mathbf{e}_i \mathbf{e}_j \\ &= \mathbf{a}_1 \cdot \mathbf{a}_2 + \mathbf{a}_1 \wedge \mathbf{a}_2 \end{aligned} \quad (8)$$

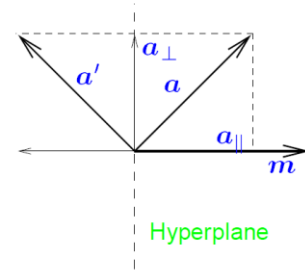
Tại đây, $\mathbf{a}_1 \cdot \mathbf{a}_2 = \sum_{i=1}^m a_{1i} a_{2i}$ là nội tích (inner product) hay còn gọi là tích vô hướng. $\mathbf{a}_1 \wedge \mathbf{a}_2 = -\mathbf{a}_2 \wedge \mathbf{a}_1$ là ngoại tích (outer product) của 2 vector $\mathbf{a}_1$ và $\mathbf{a}_2$. Chú ý, ngoại tích ở đây khác với tích có hướng (cross product) trong mô hình đại số tuyến tính. Từ công thức trên, ta thấy tích hình học của 2 vector trong không gian chính là tổng của nội tích và ngoại tích.

3.2. Phép phản xạ ảnh trong geometric algebra

Xét bài toán tìm ảnh phản xạ của một vector $\mathbf{a}$ qua một (siêu) mặt phẳng ((hyper) plane) chứa tọa độ gốc và có pháp tuyến là một vector $\mathbf{m} (|\mathbf{m}|^2 = 1)$. Hình 1 thể hiện bài toán đã đặt ra. Gọi $\mathbf{a}_\perp$, $\mathbf{a}_\parallel$ và $\mathbf{a}'$ lần lượt là hình chiếu của $\mathbf{a}$ lên mặt phẳng, hình chiếu của của $\mathbf{a}$ lên vector $\mathbf{m}$ và ảnh phản xạ của $\mathbf{a}$ qua mặt phẳng, ta có:

$$\mathbf{a} = \mathbf{a}_\perp + \mathbf{a}_\parallel \quad (9)$$

$$\mathbf{a}' = \mathbf{a}_\perp - \mathbf{a}_\parallel \quad (10)$$



Hình 1. Phản xạ một vector qua một mặt phẳng

Mặt khác, ta có $\mathbf{a}_\parallel$ là hình chiếu của của $\mathbf{a}$ lên vector $\mathbf{m}$ nên:

$$\mathbf{a}_\parallel = (\mathbf{a} \cdot \mathbf{m}) \mathbf{m} \quad (11)$$

Suy ra,

$$\begin{aligned} \mathbf{a}_\perp &= \mathbf{a} - (\mathbf{a} \cdot \mathbf{m}) \mathbf{m} \\ &= (\mathbf{a} \mathbf{m} - \mathbf{a} \cdot \mathbf{m}) \mathbf{m} = (\mathbf{a} \wedge \mathbf{m}) \mathbf{m} \end{aligned} \quad (12)$$

Từ đó ta có kết quả của phép phản xạ ảnh như sau:

$$\begin{aligned} \mathbf{a}' &= \mathbf{a}_\perp - \mathbf{a}_\parallel \\ &= (\mathbf{a} \wedge \mathbf{m}) \mathbf{m} - (\mathbf{a} \cdot \mathbf{m}) \mathbf{m} \\ &= -(\mathbf{m} \wedge \mathbf{a} + \mathbf{a} \cdot \mathbf{m}) \mathbf{m} \\ &= -\mathbf{m} \mathbf{a} \mathbf{m} \end{aligned} \quad (13)$$

3.3. Phép quay trong geometric algebra

Quay một vector theo một trục quay bất kỳ trong không gian chính là việc phản xạ vector đó qua 2 mặt phẳng trong không gian. Hình 2 trình bày việc quay vector $\mathbf{a}$ qua 2 mặt phẳng có vector pháp tuyến lần lượt là $\mathbf{m}$ và $\mathbf{n}$. Trong đó $\mathbf{a}'$ là phản xạ của $\mathbf{a}$ qua mặt phẳng có vector pháp tuyến $\mathbf{m}$, $\mathbf{a}''$ là kết quả của phép quay và cũng là phản xạ của $\mathbf{a}'$ qua mặt phẳng có vector pháp tuyến $\mathbf{n}$. Dựa vào công thức tính phản xạ trong mô hình toán học GA, ta có:

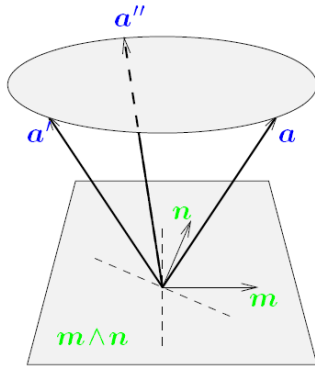
$$\mathbf{a}' = -\mathbf{m} \mathbf{a} \mathbf{m},$$

$$\mathbf{a}'' = -\mathbf{n} \mathbf{a}' \mathbf{n} = -\mathbf{n} (-\mathbf{m} \mathbf{a} \mathbf{m}) \mathbf{n} = \mathbf{n} \mathbf{m} \mathbf{a} \mathbf{m} \mathbf{n}$$

Định nghĩa $\mathbf{R} = \mathbf{n} \mathbf{m}$ là rotor trong phép quay, ta có:

$$\mathbf{a}'' = \mathbf{R} \mathbf{a} \tilde{\mathbf{R}}$$

Trong đó $\tilde{\mathbf{R}}$ là siêu số phức liên hợp của $\mathbf{R}$.



Hình 2. Phép quay trong không gian

Nhận thấy rằng R gồm có 1 tham số thực (nội tích của $\mathbf{n}$ và $\mathbf{m}$) và 3 tham số ảo (ngoại tích của $\mathbf{n}$ và $\mathbf{m}$). Như vậy, đối với phép quay trong không gian 3D, ta chỉ cần 4 tham số là có thể thực hiện được. So với việc sử dụng ma trận quay (27 tham số) thì tiết kiệm bộ nhớ hơn nhiều và phép tính cũng đơn giản hơn. Đối với trường hợp đặc biệt trong không gian 3D ta có, R chính là một quaternion đã được hàm số hóa trong các công cụ xử lý 3D như DirectX hay OpenGL. Tuy nhiên, phép quay trình bày trong bài báo này không chỉ giới hạn trong không gian 3D mà có thể thực hiện trong không gian với số chiều bất kỳ.

4. Khảo sát và đánh giá kết quả

Phần này so sánh mức độ xử lý chính xác của 2 cách quay, quay bằng ma trận và quay bằng phương pháp sử dụng GA. Giả sử có một vật thể chứa N đỉnh trong không gian 3D. Sau khi thực hiện M lần quay ngẫu nhiên, ta so sánh tất cả chiều dài của các cạnh tương ứng với các đoạn thẳng nối các cặp đỉnh của vật thể sau khi quay và vật thể ban đầu. Ta có công thức để tính độ sai lệch độ dài của các cạnh như sau

$$L = \frac{1}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^N |d_{ij}^{(0)} - d_{ij}^{(M)}| \quad (14)$$

Trong đó, $d_{ij}^{(t)} = \|\mathbf{x}_i^{(t)} - \mathbf{x}_j^{(t)}\|$ là chiều dài của đoạn thẳng nối 2 đỉnh $\mathbf{x}_i^{(t)}$ và $\mathbf{x}_j^{(t)}$ của vật thể sau khi quay ngẫu nhiên t lần. Và đỉnh $\mathbf{x}_i^{(t)}, i = 1, \dots, N$ được tính toán theo công thức sau: $\mathbf{x}_i^{(t)} = \text{Round}(\text{Ro}(\mathbf{x}_i^{(t-1)}), E)$ (15)

Ở đây, $\text{Round}(\cdot, E)$ là hàm số làm tròn tất cả các thành phần của vec tơ lấy kết quả E chữ số sau dấu phẩy thập phân. Ta có ngầm thể hiểu

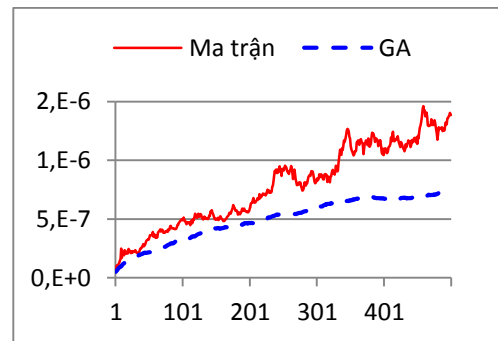
rằng, hàm $\text{Round}(\cdot, E)$ thể hiện mức độ tính toán chính xác của hệ thống khi sử dụng phép quay. $\text{Ro}(\cdot)$ là hàm quay có thể được tính theo 2 cách đã giới thiệu ở bài báo này.

Bảng 1 là kết quả sai lệch độ dài của các cạnh tùy thuộc vào M và E sau khi quay tất cả các đỉnh số đỉnh của vật thể ($N = 100$) bằng ma trận và quay bằng GA. Biết rằng các đỉnh của vật thể được khởi tạo ngẫu nhiên sao cho tất cả các phần tử của một đỉnh bất kỳ nằm trong khoảng $[-1, 1]$.

Bảng 1. So sánh độ sai lệch chiều dài các cạnh của vật thể lúc ban đầu và sau M lần quay theo 2 phương pháp, quay bằng ma trận và quay bằng geometric algebra.

E	M	Ma trận	GA
9	100	$5.3(\pm 1.9) \times 10^{-9}$	$3.3(\pm 0.16) \times 10^{-9}$
9	500	$1.1(\pm 0.4) \times 10^{-8}$	$0.7(\pm 0.03) \times 10^{-8}$
7	100	$5.3(\pm 2.0) \times 10^{-7}$	$3.3(\pm 0.15) \times 10^{-7}$
7	500	$1.1(\pm 0.4) \times 10^{-6}$	$0.7(\pm 0.04) \times 10^{-6}$
5	100	$4.9(\pm 1.4) \times 10^{-5}$	$3.3(\pm 0.15) \times 10^{-5}$
5	500	$1.1(\pm 0.4) \times 10^{-4}$	$0.7(\pm 0.03) \times 10^{-4}$

Từ Bảng 1 ta thấy, khi số lần quay càng lớn và độ tính toán chính xác của hệ thống càng thấp thì mức độ sai lệch độ dài các cạnh của cả hai phương pháp quay đều tăng. Tuy nhiên so với quay bằng phương pháp ma trận thì phương pháp GA có mức độ sai lệch ít hơn. Hình 3 là kết quả của một trường hợp đặc biệt khi $M = 500$ và $E = 7$. Ta cũng thấy được mức độ sai lệch trong phương pháp quay GA cũng ổn định hơn. Điều đó có thể kết luận được rằng sử dụng phương pháp quay bằng GA là tốt hơn phương pháp sử dụng ma trận.



Hình 3. Độ sai lệch chiều dài trong một trường hợp khi $M=500$ và $E=7$.

5. Kết luận

Bài báo này đã trình bày 2 phương pháp quay trong không gian 3D là phép quay sử dụng nhân ma trận và phép quay sử dụng mô hình toán học GA. Bài báo đã cho thấy phép quay 3D trong mô hình GA chính là phép quay sử dụng quaternion đã được hàm số hóa trong các công cụ xử lý 3D như DirectX hay OpenGL. Tuy

nhiên không giới hạn về số chiều như quaternion là chỉ sử dụng được đối với 3 chiều. Qua khảo sát với việc quay ngẫu nhiên một vật thể trong không gian 3D bài báo này đã kết luận được rằng việc sử dụng GA để quay vật thể sẽ có độ chính xác hơn hẳn so với phương pháp quay bằng ma trận.

TÀI LIỆU THAM KHẢO

- [1] James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, Computer graphics: principles and practice (2nd ed.), Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1990.
- [2] Eberly, D. H. *3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics*, Morgan Kaufmann Publishers. 2001.
- [3] C. Doran and A. Lasenby, Geometric algebra for physicists, *Cambridge University Press*, 2003.
- [4] D. Hestenes, New foundations for classical mechanics, Dordrecht, 1986.
- [5] L. Dorst, D. Fontijne, and S. Mann, *Geometric Algebra for Computer Science: An Object-oriented Approach to Geometry* (Morgan Kaufmann Series in Computer Graphics), 2007.
- [6] I. Sekita, T. Kurita, and N. Otsu, Complex Autoregressive Model for Shape Recognition, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 4, 1992.
- [7] A. Hirose, Complex-Valued Neural Networks: Theories and Applications, *Series on Innovative Intelligence*, Vol. 5, 2006.
- [8] T. Nitta, An Extension of the Back-Propagation Algorithm to Complex Numbers, *Neural Networks, Volume 10, Number 8*, pp. 1391–1415(25), November 1997.
- [9] N. Matsui, T. Isokawa, H. Kusamichi, F. Peper, and H. Nishimura, Quaternion neural network with geometrical operators, *Journal of Intelligent and Fuzzy Systems*, Volume 15, Numbers 3–4, pp. 149–164, 2004.
- [10] S. Buchholz and N. Le Bihan, Optimal separation of polarized signals by quaternionic neural networks, *14th European Signal Processing Conference, EUSIPCO 2006*, September 4–8, Florence, Italy, 2006.
- [11] E. Hitzer, Quaternion Fourier Transform on Quaternion Fields and Generalizations, *Advances in Applied Clifford Algebras*, 17(3), pp. 497–517 (2007).
- [12] G. Sommer, *Geometric Computing with Clifford Algebras*, Springer, 2001.
- [13] M. T. Pham, K. Tachibana, E. Hitzer, T. Yoshikawa, T. Furuhashi, Classification and Clustering of Spatial Patterns with Geometric Algebra, in G. Scheuermann, E. Bayro-Corrochano (eds.), *Geometric Algebra Computing*, Springer, New York, pp. 231–248, 2010.
- [14] M.T. Pham, K. Tachibana, E. Hitzer, S. Buchholz, T. Yoshikawa, T. Furuhashi, Feature Extractions with Geometric Algebra for Classification of Objects, *Proceedings of IEEE World Congress on Computational Intelligence - International Joint Conference on Neural Networks (IJCNN 2008)*, Hong Kong, 1-6 June 2008, pp. 4069–4073, 2008.

(BBT nhận bài: 30/07/2013, phản biện xong: 30/07/2013)