

Applications of Clifford Algebra in Mixed Reality environment

E. Y. T. Ho

Imaging Science & Biomedical Engineering
Medical School, University of Manchester
Stopford Building, Oxford Road
Manchester, M13 9PL
United Kingdom
edward.ho@isbe.man.ac.uk

Abstract

The 'beauty' of Clifford's Geometric Algebras is its ability to incorporate other algebras and it is the 'mother' algebra for all algebras. This paper introduces the advantage of using this algebra by combining and augmenting certain group of algebras, such as linear algebra, quaternion algebra, the Grassmann algebra and projective algebra to simplify mathematical manipulations in 3-dimensional rotations and projective geometry, especially in the context of mixed reality environment. Those 'augmented' representations are shown with applications in the mixed reality environment, especially for registration and computer vision based object recognition issues. Some simple scenarios with place-holder objects are described at the end for a full understanding of the mixed reality applications before other most recent engineering and computer science areas using this algebra for their applications are briefly discussed.

Keywords: Clifford algebra, rotors, projective split, mixed reality, registration, computer vision based object recognition, place holder objects

1. Introduction

A central problem in the 1st-half of the 19th century was how best to represent 3-D rotations. When Hamilton introduced Quaternion Algebra, H in 1843, he was able to manipulate compositions of triplets in R^3 . This algebra contains 4 elements $\{1, i, j, k\}$ and $ij = k = -ji$, $i^2 = j^2 = k^2 = ijk = -1$. This non-commutative algebra is isomorphic to even sub-algebras of Clifford Algebra. In a separate development, Grassmann pioneered the Exterior Algebra in 1844. This defined what we now call a bivector $a \wedge b$ of vectors a and b , and it is anti-commute, $a \wedge b = -b \wedge a$.

Geometric Algebras (GA) were created by William K. Clifford in 1878 when he combined inner products with Grassmann's exterior algebra [1]. One of the main features of GA is the ability to 'improve' Euler angles representations in quaternion terminology, via new defined Rotors in exponential form [2] [3]. These rotors are particular useful for 3-D rotation, involving kinematics and dynamics manipulations [8] [9] [10]. They also 'speed up' concatenation of

matrices [6] [7]. In Computer Vision applications, Geometric Products can further represent homogeneous coordinates and 'map' perspective projections of 4-D onto its 3-D equivalent, via the Projective Split [11] [12].

This special property of GA is important, especially to the video displays which need the highest accuracy possible. Mixed Reality (MR) and Augmented Reality (AR) displays are rather different from the ordinary Virtual Reality (VR) displays we usually see. During training of a new pattern, projective geometry is extensively used in the object recognition algorithms. MR/AR involves extra procedures, such as system calibration and registration of virtual objects overlaid onto the real scene. Best methodologies have to be implemented according to hardware used to assure that an AR system works at its best ability.

2. Using Clifford Algebra

Using orthonormal bases $\{e_1, e_2, \dots, e_n\}$ for R^n , the Exterior Algebra, $\wedge R^n$ has bases

$$\{1, e_1, e_2, \dots, e_n, e_1 \wedge e_2, e_1 \wedge e_3, \dots, e_{n-1} \wedge e_n, \dots, e_1 \wedge e_2 \wedge \dots \wedge e_n\}$$

The special properties, i.e. associativity and anticommutativity of this exterior product make bivectors so useful for practical manipulations.

In conventional vector algebra, there are two standard products : Dot product, $a \cdot b$ (scalar) with magnitude $|a||b|\cos\theta$; Cross product, $a \times b$ (vector) with magnitude $|a||b|\sin\theta$, with $|a|$ and $|b|$ are the length of vectors a and b , θ is the angle between them. The cross product is a fundamental notation used in engineering and sciences, but it is not accurate enough! Please refer to [2] [3] [4] [5] for some fundamental arguments, especially in some physical mathematics definitions. Therefore, a new product called : Outer product, $a \wedge b$ (bivector) with magnitude $|a||b|\sin\theta$ was introduced.

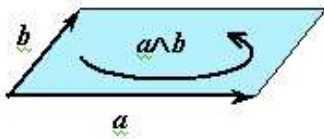


Figure 1: A bivector

It is not a scalar or a vector, but a directed area in the plane segment containing vectors a and b . We interpret $(a \wedge b) \wedge c$ as the oriented 3-D volume obtained by sweeping the bivector $a \wedge b$ along vector c . It is called trivector. It is anticommutative, $a \wedge b = -b \wedge a$; distributive, $a \wedge (b + c) = a \wedge b + a \wedge c$; and associative, $a \wedge (b \wedge c) = (a \wedge b) \wedge c$

So far we have a symmetric inner product and anti-symmetric outer product. Clifford’s great idea was to introduce a new product which combine the two. This is the Geometric Product, written simply as ab , $ab = a \cdot b + a \wedge b$, [1]. The right-hand side is a sum of two distinct objects, a scalar and a bivector. From the symmetry and anti-symmetry properties of the right-hand side, $ba = a \cdot b - a \wedge b$. It follows that, $a \cdot b = \frac{1}{2}(ab + ba)$, $a \wedge b = \frac{1}{2}(ab - ba)$

2.1. Applications of Clifford Algebra in 2-Dimensional Space, Cl_2

Consider a 2-D space, i.e. a plane spanned by two orthonormal vectors e_1, e_2 . These basis vectors satisfy $e_1^2 = e_2^2 = -1$, $e_1 \cdot e_2 = 0$. The final entity present in the 2-D algebra is the bivector $e_1 \wedge e_2$. This is the highest grade element in the algebra, which is often called pseudoscalar. The pseudoscalar

is defined to be right-handed, so that e_1 sweeps e_2 in a right-handed sense. The full algebra is spanned by

1	$\{e_1, e_2\}$	$e_1 \wedge e_2$
1 scalar	2 vectors	1 bivector

We denote this algebra by Cl_2 . To study the properties of the bivector $e_1 \wedge e_2$, we first note that $e_1 e_2 = e_1 \cdot e_2 + e_1 \wedge e_2 = e_1 \wedge e_2$, $e_1 \wedge e_2 = -e_2 \wedge e_1$. Also note that, $(e_1 \wedge e_2)e_1 = (-e_2 e_1)e_1 = -e_2 e_1 e_1 = -e_2$, $(e_1 \wedge e_2)e_2 = (e_1 e_2)e_2 = e_1 e_2 e_2 = e_1$. Similarly, acting from the right, we have $e_1(e_1 e_2) = e_2$, $e_2(e_1 e_2) = -e_1$. Normally a pseudoscalar is denoted I . In this case $I = e_1 \wedge e_2$, $I^2 = (e_1 \wedge e_2)^2 = e_1 e_2 e_1 e_2 = -e_1 e_1 e_2 e_2 = -1$

For pure geometric considerations, we have discovered a quantity, whose squares equal to -1 . This fits with the fact that 2 successive left (or right) multiplications of a vector by $e_1 e_2$ rotates the vector through 180° , equivalent to multiplying by -1 . Suppose that we have two completely arbitrary elements of the Cl_2 algebra, A and B , which can be decomposed in terms of $\{e_1, e_2\}$ frame : $A = a_0 + a_1 e_1 + a_2 e_2 + a_3 e_1 \wedge e_2$, $B = b_0 + b_1 e_1 + b_2 e_2 + b_3 e_1 \wedge e_2$

The product of these two elements can be written as $AB = p_0 + p_1 e_1 + p_2 e_2 + p_3 e_1 \wedge e_2$ where

$$p_0 = a_0 b_0 + a_1 b_1 + a_2 b_2 - a_3 b_3, p_1 = a_0 b_1 + a_1 b_0 + a_3 b_2 - a_2 b_3, p_2 = a_0 b_2 + a_2 b_0 + a_1 b_3 - a_3 b_1, p_3 = a_0 b_3 + a_3 b_0 + a_1 b_2 - a_2 b_1$$

This multiplication law is easy to represent as part of a computer language. In general, however, $AB \neq BA$

One of the main applications of Cl_2 is the ability to simplify vector algebra in R^4 . It is isomorphic, as an associative algebra, to the matrix algebra representations of the real $R^2 \times R^2$ matrices. Writing Cl_2 bases in matrix form, we have

$$1 \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, e_1 \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, e_2 \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, e_1 \wedge e_2 \equiv \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Please refer to [6] [7] for further details on the applications of Cl_2 in tri-diagonalising system matrices of large mass matrix, M , damping matrix, C and stiffness matrix, K in vibration terminology. A simple illustration is shown in the figure as follows.

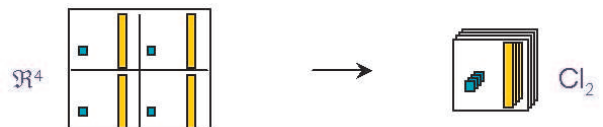


Figure 2: Mapping of $R^2 \times R^2$ of a matrix into Cl_2

2.2. Applications of Clifford Algebra in 3-Dimensional Space, Cl_3

In Cl_3 , the 3-D algebra is spanned by

1	$\{e_i\}$	$\{e_i \wedge e_j\}$	$e_1 \wedge e_2 \wedge e_3$
1 scalar	3 vectors	3 bivectors	1 trivector

These define a linear space of dimension $8=2^3$. Notice that the dimensions of each subspace are given by the binomial coefficients. Recall that, $(e_1e_2)^2 = (e_2e_3)^2 = (e_3e_1)^2 = -1$

A vector can be split to its parallel and perpendicular components, $a = a_{\parallel} + a_{\perp}$, $aB = (a_{\parallel} + a_{\perp})B$, but $B = a_{\parallel} \wedge b$. We see that $a_{\parallel}B = a_{\parallel}(a_{\parallel} \wedge b) = a_{\parallel}(a_{\parallel}b) = a_{\parallel}^2b$ which is a vector, whereas $a_{\perp}B = a_{\perp}(a_{\parallel} \wedge b) = a_{\perp} \wedge a_{\parallel} \wedge b$, which is a trivector. We therefore write $aB = a \cdot B + a \wedge B$. Note that $a \cdot B = a_{\parallel}^2b = -(a_{\parallel}b)a_{\parallel} = -B \cdot a$, $a \wedge B = a_{\perp} \wedge a_{\parallel} \wedge b = a_{\parallel} \wedge b \wedge a_{\perp} = B \wedge a$, $a \cdot B = \frac{1}{2}(aB - Ba)$, $a \wedge B = \frac{1}{2}(aB + Ba)$

Since $(e_1e_2)^2 = (e_2e_3)^2 = (e_3e_1)^2 = -1$, write $B_1 = e_2e_3$, $B_2 = e_3e_1$, $B_3 = e_1e_2$, we have $B_1^2 = B_2^2 = B_3^2 = -1$, $B_1B_2 = -B_2B_1$. This recover the Hamilton's Quaternion Algebra : $i^2 = j^2 = k^2 = ijk = -1$, $ij = -ji$. In fact, Quaternions were bivectors all along! The pseudoscalar, $I = e_1e_2e_3$. Each of the basis bivectors can be expressed as the product of the pseudoscalar and a dual vector, $e_1e_2 = Ie_3$, $e_2e_3 = Ie_1$, $e_3e_1 = Ie_2$. The square of the pseudoscalar, $I^2 = e_1e_2e_3e_1e_2e_3 = e_1e_2e_1e_2 = -1$. Finally, we consider $I(e_1 \wedge e_2) = Ie_1e_2e_3e_3 = Ie_3 = -e_3$. This affords a definition of the vector cross product in 3-D as $a \times b = -I(a \wedge b) = -Ia \wedge b$

Suppose that we reflect the vector a in the hyperplane orthogonal to some unit vector m , $a_{\parallel} = a \cdot mm$, $a_{\perp} = a \wedge mm$. The result of reflection is $a' = a_{\perp} - a_{\parallel} = -a \cdot mm + a \wedge mm = -(m \cdot a + m \wedge a)m = -mam$. This remarkably compact formula only arises in GA. We can also prove that inner products are unchanged by reflections, $a' \cdot b' = (-mam) \cdot (-mbm) = \langle mammbm \rangle = \langle mabm \rangle = \langle mmab \rangle = a \cdot b$. However, the outer product, in this case the bivector under reflections is $a' \wedge b' = (-mam) \wedge (-mbm) = \langle mammbm \rangle^2 = \langle mammbm \rangle^2 = ma \wedge bm$

Bivectors do not quite transform as vectors under reflections. This is the reason for the confusing distinction between polar and axial vectors in 3-D. Axial vectors are really bivectors and should be treated as such.

A rotation in the plane generated by two unit vectors m and n is achieved by successive reflections in the (hyper) planes perpendicular to m and n . $a' = -mam$, $a'' = -na'n = -n(-mam)n = nmamn$. We define rotor, $R = nm$. We can now write rotation as $a \rightarrow RaR^*$. This formula works for any grade of multivectors, in any dimension! Note that R is the geometric product of two unit vectors n and m , $R = nm = n \cdot m + n \wedge m = \cos(\theta) + n \wedge m$. So, what is the magnitude of the bivector $n \wedge m$?

$$(n \wedge m) \times (n \wedge m) = \langle n \wedge mn \wedge m \rangle = \langle nmn \wedge m \rangle = n \cdot [m \cdot (n \wedge m)] = n \cdot (m \cdot \cos(\theta) - n) = \cos^2(\theta) - 1 = -\sin^2(\theta)$$

We therefore define a unit bivector in the $m \wedge n$ plane by $\hat{u} = \frac{m \wedge n}{\sin(\theta)}$, $\hat{u}^2 = -1$. Now, we have $R = \cos(\theta) - \hat{u}\sin(\theta)$. This is nothing else than the polar decomposition of a complex number, with the unit imaginary replaced by the unit bivector, \hat{u} . We can therefore write $R = \exp\{-\hat{u}\theta\}$. Now recall the formula for a rotation $x' = e^{-I\phi/2}xe^{I\phi/2}$, we can write $R = e^{-\hat{u}\theta/2}$, which gives a rotation, $a \rightarrow e^{-\hat{u}\theta/2}ae^{\hat{u}\theta/2}$. Since rotor, R is a geometric product of two unit vectors, we see immediately that $RR^* = nm(nm)^* = nmmn = 1 = R^*R$

Lets see the relation between quaternions and rotors. If $Q = \{q_0, q_1, q_2, q_3\}$ represent a quaternion, then the rotor which perform the same rotation is simply

$$R = \underbrace{q_0}_{\text{scalar}} + \underbrace{q_1(Ie_1) - q_2(Ie_2) + q_3(Ie_3)}_{\text{bivectors}}$$

The quaternion algebra is therefore a similar representation of Cl_3 .

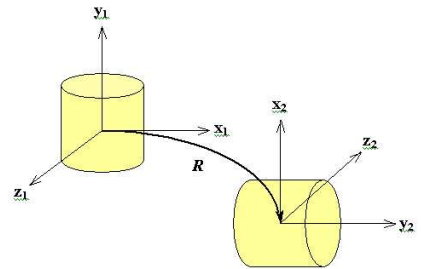


Figure 3: Rotation of a Rigid Body

2.3. Applications of Clifford Algebra in 4-Dimensional Space, Cl_4

In Cl_4 , the 4-D algebra is spanned by

1	$\{e_i\}$	$\{e_i \wedge e_j\}$	$I \wedge e_i$	I
1 scalar	4 vectors	6 bivectors	4 trivectors	pseudoscalar

These define a linear space of dimension $16=2^4$. The Grassmann-Cayley's double algebra expresses the ideas of projective geometry, such as the meet and join very elegantly, however, it lacks some key concepts, which the geometric products can complement to reduce the computational cost in calculations.

This section introduce projective geometry and computer vision in general, then how Cl_4 can be used to represent 3-D transformations and generalised projection in 4-D, via the concepts of projective split and projective transformations.

The next section explains briefly projective geometry and its best representation.

A 3D-to-3D (P^3 to P^3) projective transformation has 15-dof,

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ t_{41} & t_{42} & t_{43} & t_{44} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}$$

The ability of homogeneous coordinates to represent a general displacement as a single 4x4 matrix and to linearize nonlinear transformations has made perspective projection to be investigated easily on its properties, such as intersection of lines, its 'principle of duality', differential and integral invariants. This is crucial especially when there are a series of perspective transformations to be concatenated into a single homogeneous matrix. Please refer to [14] for further details on these.

A general 4x4 perspective projective transformation matrix can make use of the GA framework to analyse one of the best properties of homogeneous coordinates, i.e. to linearize nonlinear transformation, as illustrated below.

$$x = \frac{x_1}{x_4} = \frac{t_{11}X_1 + t_{12}X_2 + t_{13}X_3 + t_{14}X_4}{t_{41}X_1 + t_{42}X_2 + t_{43}X_3 + t_{44}X_4}$$

$$y = \frac{x_2}{x_4} = \frac{t_{21}X_1 + t_{22}X_2 + t_{23}X_3 + t_{24}X_4}{t_{41}X_1 + t_{42}X_2 + t_{43}X_3 + t_{44}X_4}$$

$$z = \frac{x_3}{x_4} = \frac{t_{31}X_1 + t_{32}X_2 + t_{33}X_3 + t_{34}X_4}{t_{41}X_1 + t_{42}X_2 + t_{43}X_3 + t_{44}X_4}$$

In order to convert these nonlinear equations of 3-D Euclidean space onto the 4-D vector linear space, we introduce the following mapping functions :

$$f_{e_1} = t_{11}e_1 + t_{12}e_2 + t_{13}e_3 + t_{14}e_4$$

$$f_{e_2} = t_{21}e_1 + t_{22}e_2 + t_{23}e_3 + t_{24}e_4$$

$$f_{e_3} = t_{31}e_1 + t_{32}e_2 + t_{33}e_3 + t_{34}e_4$$

$$f_{e_4} = t_{41}e_1 + t_{42}e_2 + t_{43}e_3 + t_{44}e_4$$

If e_4 is a selected direction, we define the mapping of the associated bivectors, $e_i e_4$, $i = 1, 2, 3$ in perspective space with vectors, σ_i , $i = 1, 2, 3$ in 3-D Euclidean space,

$$\sigma_i \equiv e_i e_4, \quad i = 1, 2, 3 \quad \text{where} \quad \sigma_i^2 = +1$$

Let us now see how we can associate points on the perspective space onto 3-D Euclidean space, via projective split. Suppose a vector, $\chi = X_1 e_1 + X_2 e_2 + X_3 e_3 + X_4 e_4$ in projective space is obtained, the projective split in the direction of e_4 is merely the geometric product of χ and e_4 ,

$$\begin{aligned} \chi e_4 &= \chi \cdot e_4 + \chi \wedge e_4 \\ &= X_4 \left(1 + \frac{\chi \wedge e_4}{\chi} \right) \\ &\equiv X_4 (1 + \gamma) \end{aligned}$$

where $\gamma = x\sigma_1 + y\sigma_2 + z\sigma_3$ in the 3-D Euclidean space.

3. Clifford Algebra for the Mixed Reality Applications

Augmented Reality (AR) is a technology in which a computer-generated image is superimposed onto the user's vision of the real world and Mixed Reality (MR) is another version in the Reality-Virtually continuum which may make use of Place Holder Objects (PHOs) to manipulate virtual objects in a mixed environment [13]. AR or MR gives the user extra information generated from the computer model and it is not similar to Virtual Reality (VR) environments, in which an immersed scene in a virtual environment is displayed onto the user's view. By using an AR/MR system, the user's view is enhanced in the form of labels, 3-D rendered models or shaded modifications.

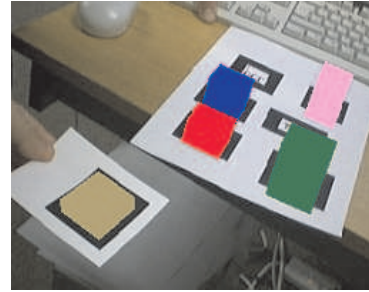


Figure 4: Virtual cubes superimposed onto some pre-trained patterns, [18]

Since the concepts behind the MR technologies are the same to those which are already properly investigated and defined in the Machine Vision contexts, therefore I am borrowing those definitions used, again please refer [14]. Before an AR/MR system can work properly, it needs to be registered and calibrated. Those registration and calibration procedures are based on the 'amazing' properties of geometry.

Augmented Round Table for Architectural and Urban Planning (ARTHUR) [16], [17] is an example of using the AR technologies for architectural and urban planning applications. The project exploits the use of relevant technologies and hardware, such as computer vision based object recognition, real-time network synchronisation and high-ended optical see-through glasses stereoscopic visualisation for multi-user collaborative discussions on a round table.

To make an AR system functions properly, many pieces of hardware and toolkit are needed. Therefore, Clifford Algebra could be used to improve its efficiency for implementation work. A generic mixed reality system involves precise registration and calibration of the coordinate reference frame of object-to-world, world-to-camera, and camera-to-display. In ARTHUR, system registration procedures were developed, i.e. the registration of the optical see-through displays, trackers and cameras on the headset, and the user's eyes position, based on a single user registration. The registration diagram can be shown as follow :



Figure 5: ARTHUR : An application of Augmented Reality exploitation for multi-user collaborative work

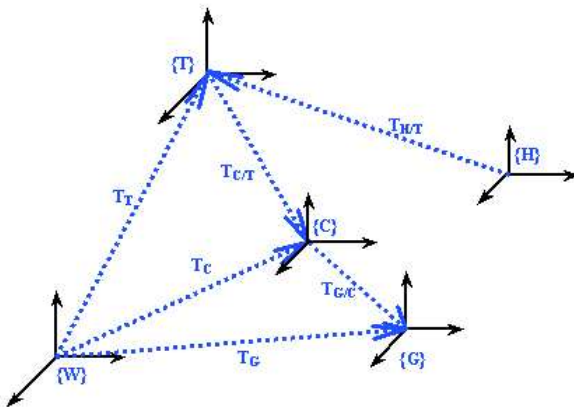


Figure 6: ARTHUR registration set-up

c.f. = coordinate reference frame; t.m. = Euclidean transformation matrix

$\{W\}$: c.f. of pattern 'A' to be trained as the world reference c.f.; $\{T\}$: c.f. of another tracking device; $\{H\}$: c.f. of tracker on the headset; $\{C\}$: c.f. camera on the headset; $\{G\}$: c.f. of the glasses and display on the headset (assuming that the Euclidean transformation between the display and the optical see-through glasses has already been taken care of)

T_T : t.m. from $\{W\}$ to $\{T\}$; $T_{H/T}$: t.m. from $\{T\}$ to $\{H\}$; T_C : t.m. from $\{W\}$ to $\{C\}$; $T_{C/T}$: t.m. from $\{T\}$ to $\{C\}$; T_G : t.m. from $\{W\}$ to $\{G\}$; $T_{G/C}$: t.m. from $\{C\}$ to $\{G\}$

Unknowns, $T_G = T_{G/C} \times T_C$, $T_T = -T_{C/T} \times T_C$

$T_T, T_{H/T}, T_C, T_{C/T}, T_G, T_{G/C}$ are all extrinsic parameters of homogeneous transformation matrix and in the form of $\begin{bmatrix} [Rot] & [Trans] \\ \vec{0} & 1 \end{bmatrix}$, where $[Rot]$ and $[Trans]$ are the 3-D rigid body rotation and translation respectively in Euclidean space, $SE(3) = SO(3) \cdot R^3$. I have shown in Section 2.2 that a rotor in Euler representation is $R = e^{-\hat{u}\theta/2} = \cos(\theta/2) - \hat{u}\sin(\theta/2)$ where $\hat{u} = u_x e_2 e_3 + u_y e_3 e_1 + u_z e_1 e_2$

is the 3-D rotational axes spanned by the bivector bases. 3-D rotation and translation from one position to another through 3-D rigid body motion is simply $a \rightarrow R(a+t)R^* = R[(a_x + t_x)e_1 + (a_y + t_y)e_2 + (a_z + t_z)e_3]R^*$ Concatenation of a group of rotors, R_1, R_2, \dots, R_n is simply $R = R_n \dots R_2 R_1$

Another software developed in ARTHUR, which allow a user to display the exact location of pre-trained place holder objects (PHOs) with respect to the camera view point by just clicking at an edge of the PHOs. The diagram below shows a brief interface of the software when the camera is looking at a pre-trained pattern 'A' with the 3D-coordinate values as displayed.

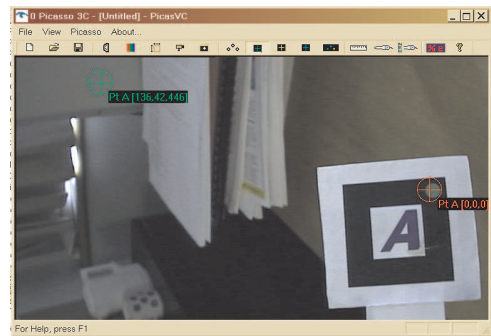


Figure 7: Software for 'instant' registration of user into the AR environment

The purpose of this software is to later on allow the user to register himself (the user's head) to the MR environment instantly. When the user is wearing the headset, according to his own adjustment on the inter-pupillary distance, IPD of the display, he can then register himself onto the MR environment by just clicking at a corner of the PHOs. He is then registered to the MR environment precisely when playing with the PHOs. Since this software uses planar homographic projection terminology, 4 points are enough to solve the correspondences in the computer vision based pattern recognition algorithms. Using GA framework, the equivalent 'linearisation' of the nonlinear transformations in 3D-to-plane (P^3 to P^2) projective transformations are the generalisation of plane-to-plane (P^2 to P^2) or homographic projection. A P^3 to P^2 projective transformation has 11-dof, which is sometimes also called the 'pin-hole' camera model, can be represented as

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \underbrace{\begin{bmatrix} t_{11} & t_{21} & t_{31} & t_{41} \\ t_{12} & t_{22} & t_{32} & t_{42} \\ t_{13} & t_{23} & t_{33} & t_{43} \\ t_{14} & t_{24} & t_{34} & t_{44} \end{bmatrix}}_{\text{projective matrix}} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}$$

A general P^3 to P^2 projective model can be written in its homogeneous matrix form, make up of intrinsic parameters of the camera (focal length, principal axes, scaling, shearing and distortion), 4-D to 3-D perspective projection, and

extrinsic parameters representing the Euclidean transformation (rotation and translation about the optical centre).

$$\begin{bmatrix} \text{intrinsic} \\ \text{parameters} \\ \text{of camera} \end{bmatrix} \begin{bmatrix} 4D \rightarrow 3D \\ \text{perspective} \\ \text{projection} \end{bmatrix} \begin{bmatrix} \text{extrinsic} \\ \text{parameters} \\ \text{(Euclidean} \\ \text{transformation)} \end{bmatrix}$$

$$x = \frac{x_1}{x_3} = \frac{t_{11}X_1 + t_{12}X_2 + t_{13}X_3}{t_{31}X_1 + t_{32}X_2 + t_{33}X_3}$$

$$y = \frac{x_2}{x_3} = \frac{t_{21}X_1 + t_{22}X_2 + t_{23}X_3}{t_{31}X_1 + t_{32}X_2 + t_{33}X_3}$$

And the equivalent mapping functions are

$$f_{e_1} = t_{11}e_1 + t_{12}e_2 + t_{13}e_3 + t_{14}e_4$$

$$f_{e_2} = t_{21}e_1 + t_{22}e_2 + t_{23}e_3 + t_{24}e_4$$

$$f_{e_3} = t_{31}e_1 + t_{32}e_2 + t_{33}e_3 + t_{34}e_4$$

The projective split in this P^2 to P^2 projection is simply

$$\begin{aligned} \chi e_3 &= \chi \cdot e_3 + \chi \wedge e_3 \\ &= X_3 \left(1 + \frac{\chi \wedge e_3}{\chi} \right) \\ &\equiv X_3(1 + \gamma) \end{aligned}$$

where $\chi = X_1e_1 + X_2e_2 + X_3e_3$; $\gamma = x\sigma_1 + y\sigma_2$; $\sigma_i \equiv e_i e_3$, $i = 1, 2$

We have seen above is only an example of the 'pin-hole' camera projective model. However, in the actual world, an AR system normally needs a few concatenation of system matrices, although usually only one of the matrices will involve those camera properties. Besides the intrinsic and extrinsic parameters have to be calibrated and registered accurately from the 3-D world onto a 2-D image plane, AR systems are expected to work in real-time, move about freely within the scene and see a properly rendered augmented image align onto the desired objects.

In ARTHUR, the physical counterpart of the interaction units are those PHOs, and simple hand gestures. Since computer vision based techniques for calculating the position and orientation of 3-D objects from single projective images are well developed, using image features, such as points, edges and junctions, accuracy of the recognition strongly depends on the quality of extracted 2-D features, i.e. pre-trained images, and the ability of corresponding those features on the images to their 3-D real object features. The diagram below shows the flow-chart for a training procedure using computer vision based object recognition method. Once again, GA is essential for writing the code of this homographic object recognition training procedure, especially for the mathematical library implementation. Source codes are much neater and less symbols representation when using GA terminology.

4. Application Scenarios

Some architectural application scenarios were developed for the ARTHUR's partners meeting in February 2002. These

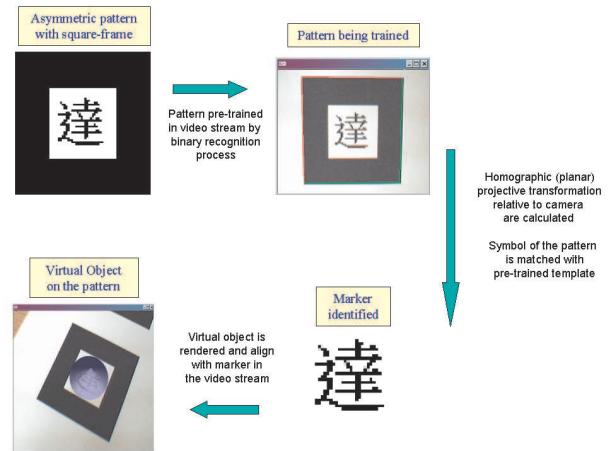


Figure 8: Computer Vision based Object Recognition

scenarios were created using the object recognition tool integrated with the available intelligent architecture software of interactive 3-D design creation toolkit [19], based on scripting language.

For 'real-world' illustration purposes, the model of SwissRe building is first recognised by a pre-trained PHO. It is then placed at model of the London city plan at the desired location as shown in figure below.

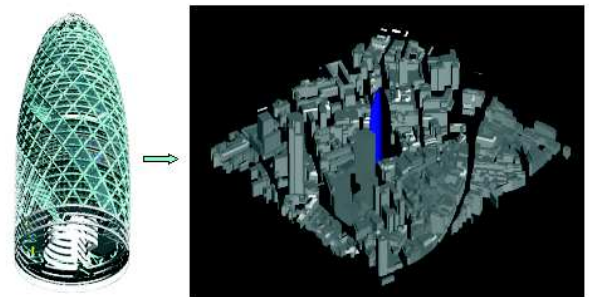


Figure 9: Model of SwissRe building is placed onto the city plan by object recognition technique

4.1. Scenario 1 {Figure 10}

This demonstration shows 3-D models of SwissRe building and the financial city plan in London associated with PHOs to allow them to be picked up and turned around. The aim is to review that the complexity of surface geometry and texture can be handled within the toolkit developed. In the animation, the city plan is recognised by pattern 'B' and the SwissRe model by pattern 'A'. First, the city is placed on the ground in the 'virtual' world, then the SwissRe building is controlled and placed at the empty area within the city plan.



Figure 10: City plan model is detected by pattern 'B' and placed on the ground; SwissRe building model is detected by pattern 'A'; SwissRe building is placed on the desired empty area within the city plan

4.2. Scenario 2 {Figure 11}

This scenario shows form manipulation of a simple multi-storey building can be controlled by the PHOs. Total floor area is fixed with adjustable size and number of floor using scripting in the toolkit. In this simple demo, a semi-transparent yellow building is detected by the multiple PHOs 'A', 'B' and 'C'. Patterns 'A', 'B', and 'C' can control and manipulate the height, width and depth of the building.



Figure 11: Creating a simple building in the city plan; Patterns 'A', 'B' and 'C' are represented by PHOs in the environment; A multi-storey building is built by controlling the PHOs

4.3. Scenario 3 {Figure 12}

In this final scenario, we show that the interface is able to switch on and off various elements, so that different data sets can be viewed. In this demo, we are showing that a blue cube is manipulated by the PHOs 'A', 'B' and 'C'. The scene can also be switched to different view points such as to upper and lower ground of the SwissRe building.

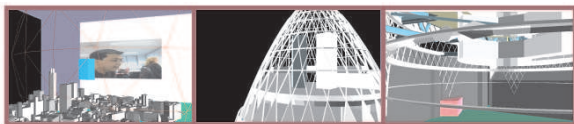


Figure 12: A blue cube is controlled by the PHOs; The view is switched to the top floor of SwissRe model; Then, the ground floor of SwissRe model

5. Discussion & Conclusion

This paper introduced Clifford Algebra as a unified mathematical framework to all technical researchers. Although much research work have already been investigated for physical applications, and recently some in the computer science

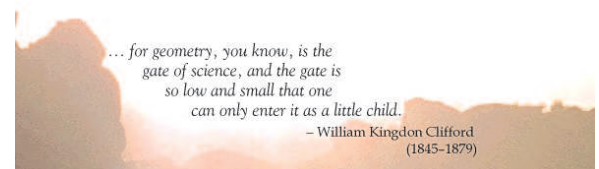
and engineering researches, since Clifford's Geometric Algebras (GA) serve as a generalisation framework for all algebras.

GA was introduced in a simple manner, suitable for all scientific researchers, starting from Cl_2 with a simple application illustration. Then, Cl_3 with the usage of rotors. This was followed by using Cl_4 for 3D-to-3D perspective projective transformation. The next section illustrated the applications of Clifford Algebra in the Mixed Reality environment, started with an AR system registration of various coordinate frames of reference in Euclidean space. Then, an instant registration software for a user wearing a head-mounted display to register himself into the MR environment were demonstrated using Clifford Algebra terminologies. Finally, Clifford Algebra was incorporated into computer vision based object recognition technologies. The final section showed some architectural application scenarios for the direct usage of the end-users in 'real-world' applications.

Today, the applications of GA cover Classical Mechanics, Black Holes and Cosmology, Quantum Tunnelling, QFT, Electromagnetism, Lie Groups, Symbolic Algebras, Screw Theory, Numerical Analysis, Structural Dynamics and Buckling, Elasticity and Solid Mechanics, Robotics, Computer Vision, Cybernetics, Signal and Image Processing, Control Theory, Quantum Information, Computer Programming, Biomedical Engineering and many more yet to be discovered!

Acknowledgements

Firstly, thanks to the PPARC for sponsoring me in a short research studies at the Institute of Astronomy, University of Cambridge, U.K. and Dr. C. J. L. Doran at the Cavendish Laboratory for introducing me to Clifford Algebra. Also, freedom given to me when I did a one-year research work under supervision of Prof. S. D. Garvey in the University of Nottingham, U.K. in 2001. I would also like to thank Prof. B. F. Buxton for his guidance in Computer Vision, Dr. A. Steed for Augmented Reality supervision and people in the European Commission funded project, ARTHUR for my one-year work at the University College London in 2002. Finally, thanks to my present supervisor, Prof. C. J. Taylor for letting me to complete and publish this paper.



References

1. Clifford W. K. *Applications of Grassmann's extensive algebra*. Am. J. Math., 1878.

2. Lounesto, P. *Clifford Algebras and Spinors, 2nd Edition*. London Mathematical Society Lecture Note Series 286, Cambridge University Press, 2001.
3. *Physical Applications of Geometric Algebra*. Part III Physics Lecture Notes, University of Cambridge.
4. Doran C. J. L. *Geometric Algebra and its Application to Mathematical Physics*. Ph.D. thesis, University of Cambridge, 1994.
5. Hestenes, D. *New Foundations for Classical Mechanics*, Kluwer, 1986.
6. Garvey, S. D. et. al. *Clifford Algebra, Cl_2 in the analysis of passive periodic structures*. The Fourth International Conference on Modern Practice in Stress and Vibration Analysis, 2000.
7. Garvey, S. D. et. al. *The Role of Clifford Algebra in Structure-Preserving Transformations for Second-Order Systems*. Applications of Geometric Algebra in Computer Science and Engineering, Dorst, L. Editors, Birkhauser, 2002.
8. Selig, J. M. *Robot Kinematics and Flags*. Geometric Algebra with Applications in Science and Engineering, Corrochano, E. B. and Sobczyk, G. Editors, Birkhauser, 2001.
9. Ahlers, S. G. and McCarthy J. M. *Clifford Algebra and the Optimisation of Robot Design*. Geometric Algebra with Applications in Science and Engineering, Corrochano, E. B. and Sobczyk, G. Editors, Birkhauser, 2001.
10. Lasenby, J. and Stevenson, A. *Using Geometric Algebra for Optical Motion Capture*. Geometric Algebra with Applications in Science and Engineering, Corrochano, E. B. and Sobczyk, G. Editors, Birkhauser, 2001.
11. Corrochano, E. B. and Lasenby, J. *The Geometry Algebra of Computer Vision*. Geometric Algebra with Applications in Science and Engineering, Corrochano, E. B. and Sobczyk, G. Editors, Birkhauser, 2001.
12. Koenderink J. J. *A Generic Framework for Image Geometry*. Applications of Geometric Algebra in Computer Science and Engineering, Dorst, L. Editors, Birkhauser, 2002.
13. Ohta, Y. and Tamura, H. *Mixed Reality : Merging real and virtual Worlds*, Springer-verlay, 1999.
14. Zisserman, A. *The British Machine Vision Association and Society for Pattern Recognition (BMVC) Tutorial*. 1992.
15. Kutulakos, K.N., Vallino, J. *Affine Object Representations for Calibration-Free Augmented Reality*. IEEE Virtual Reality Ann. International Symposium, 1996.
16. Information Societies Technology (IST) Programme. *IST-2000-28559 : Augmented Round Table for Architecture and Urban Planning (ARTHUR)*. Annex 1 'Description of Work', 2001.
17. Broll, W. et. al. *The Virtual Round Table - a Collaborative Augmented Multi-User Environment*. ACM Collaborative Virtual Environments, 2000.
18. Billinghurst, M. et. al. *ARToolkit, Version 2.33*. Human Interface Technology Laboratory, University of Washington, 2000.
19. Mottram, C. et. al. *PANGEA Version 2.1b*. University College London, 1996.
20. Ho, E. Y. T. *Ultrasound Calibration for Augmented Reality of Therapy systems*. Nanyang Technological University, 2000.