Research Article

Mathematical Methods in the Applied Sciences

Received XXXX

(www.interscience.wiley.com) DOI: 10.1002/sim.0000
MOS subject classification: XXX; XXX

Real-time rendering under distant illumination with Conformal Geometric Algebra

Margarita Papaefthymiou[1], George Papagiannakis[1]

Precomputed Radiance Transfer(PRT) methods have been established for handling global illumination of objects from area lights in real-time and many techniques have been proposed for rotating the light using linear algebra rotation matrices.Rotating the light efficiently is a crucial part for computer graphics applications since it is necessary to run at interactive frame-rates. Linear algebra matrices, commonly used for handling rotations are not very efficient and require high memory consumption, as a result is established the need for proposing new more efficient rotation algorithms. In this work, we aim to employ the Conformal Geometric Algebra (CGA) as the mathematical background for shading on real-time under distant illumination diffuse surfaces with self-shadowing and rotate efficiently the environment light using CGA entities. Our work is based on Spherical Harmonics(SH) which are used for approximating illumination of irradiance maps.Our main novelty, is that we extend the PRT algorithm by representing SH with CGA. SH basis functions of band index 1 are represented using CGA entities and SH with band index larger than 1 are represented in terms of CGA-SH of band 1. Specifically, we propose a new method for representing SH with CGA entities and rotating SH by rotating CGA entities. In this way,we can visualize the SH rotations,rotate them faster than rotation matrices and provide a unique visual representation and intuition regarding their rotation, in stark contrast to usual rotation matrices.Via our CGA expressed SH we provide a significant boost on the PRT algorithm since we can represent SH rotations by CGA rotors(4 numbers) as opposed to a 9x9 sparse matrices that are usually employed. With our algorithm we pave the way for including scaling(dilation) and translation of light coefficients of the PRT algorithm using the CGA motors.

Keywords: Illumination, Global illumination for real-time, Real-time rendering, Spherical Harmonics, Spherical Harmonics Rotation

1. Introduction

In recent years, many Global illumination for real-time solutions using real environment light have been proposed. These techniques are applied in movies, computer games, virtual reality and augmented reality applications for rendering more realistic and attractive user experience. For attaining high level of believability, illumination is an important aspect that the developers and researchers focus, which is also a challenging research topic due to the hardware performance and memory limitations.

In this work, we propose a new efficient algorithm for illuminating diffuse surfaces under distant illumination represented by environment maps in real-time, using the Conformal model of Geometric Algebra [10, 11]. We approximate distant illumination with a geometrical way and as a result, we are able to rotate efficiently the distant lighting by rotating Conformal Geometric Algebra entities. Specifically, we achieve this by representing Spherical Harmonics functions with Conformal Geometric Algebra entities. SH basis functions of band index 1 are represented using CGA entities and SH with band index larger than 1 are represented in terms of CGA-SH of band 1. Our main novelty, is that we extend Precomputed Radiance Transfer (PRT) algorithm by representing SH with Conformal Geometric Algebra. Spherical Harmonics are orthogonal functions that are defined on the sphere's surface and have been extensively used in Computer Graphics for approximating the incident light for rendering diffuse surfaces. In this way, we achieve faster SH rotation performance, we provide unique visual representation and intuition regarding their rotation and we pave the way for extending our method for including scale and translation of light coefficients

1.Foundation for Research and Technology - Hellas, 100 N. Plastira Str. 70013, Heraklion, Greece, University of Crete, Computer Science Department, Voutes Campus, 70013, Heraklion, Greece
∗Correspondence to: Margarita Papaefthymiou

using Conformal Geometric Algebra motors that combine rotation, translation and scale. Via our CGA expressed SH we provide a significant boost on the PRT algorithm since we can represent SH rotations by CGA rotors (4 numbers) as opposed to a 9x9 sparse matrices that are usually employed. The more SH bands are used the more high dimensional is the rotation matrix. On the other hand CGA rotors consist of 4 numbers regardless the number of coefficients are used. As a result CGA rotors are more memory efficient.

This paper is organized as follows:

- Section 2 provides related work
- Section 3 describes our CGA PRT algorithm and the rotation of CGA-SH using CGA geometry
- Section 4 presents our results
- Section 5 provides our conclusions and future work

2. Related Work

Geometric Algebra (GA) is a mathematical tool that can be used to solve many Computer Graphics problems as demonstrated by Dorst et al. [2]. Dorst et al. [2] presented some examples written in C++ with Gaigen library and code generator. Some of these examples are recursive ray-tracing, interpolating rotations, creating Binary Space Partition (BSP) trees and detecting intersections for collision detection and shadows. Also, Papaefthymiou et al. [4] enhanced CGA as the mathematical background for character animation control and particularly for animation blending and GPU-based geometric skinning (character deformation). In this work, we aim to handle illumination with CGA, as a result we can have a single algebraic framework that handles both illumination and character animation.

2.1. Global illumination for real-time algorithms in Computer Graphics

In recent years, many Global illumination for real-time algorithms using real environment light have been proposed. These techniques are applied in movies, computer games, virtual reality and augmented reality applications for rendering more realistic and attractive simulations. For image-based lighting is necessary to use HDR images that capture a wide range of luminance values especially in environments that include daylight. For attaining high level of believability in Virtual environments, illumination is an important aspect that the developers and researchers have to focus, which is also a challenging research topic due to the performance and memory limitations.

Many mathematical representations such as SH (Spherical Harmonics) [1], SGs (Spherical Gaussians) [5] and ASGs (Anisotropic Spherical Gaussians) [6] have been used to represent spherical functions for converting incident lighting to transfer radiance. Iwasaki et al. [5] proposed a method for rendering dynamic scenes under all-frequency lighting using Integral of Spherical Gaussians with diffuse and high specular BRDFs. The environment light and the BRDF is represented with the linear combination of SGs which replaces the integral of the triple product (visibility, BRDF, light) with the sum of integrals of SGs. To calculate the diffuse component they approximate incident light and the cosine term with the sum of SGs. By changing the sharpness of SGs they can deal with dynamic highly specular BRDFs. Also, Xu et al. [6] proposed a method for handling complex BRDFs and anisotropic materials using Anisotropic Spherical Gaussians (ASGs). They render illumination of surfaces with materials that have dynamic BRDFs, under distant illumination and they derived approximate closed-form solutions for the ASGs integral, product and convolution operators. Their evaluation results show ASGs are able to represent anisotropic spherical function more effectively and efficiently compared with SGs.

In the next section, we focus on giving an overview of SH basis functions and related work done for illumination with SH.

2.2. Spherical Harmonics

SH are a orthonormal basis functions analog to Fourier basis that defined on 3D space:

$$s = (x,y,z) = (sin\theta cos\phi, sin\theta sin\phi, cos\theta) \tag{1}$$

The SH basis functions are defined like below:

$$Y_l^m(\theta,\phi) = K_l^m e^{im\phi} P_l{}^m(cos\theta) \tag{2}$$

where l is the band index ($l >= 0$) and m is the number of the SH spherical function which is defined between $-l$ and l ($-l <= m <= l$), hence there are $2l+1$ functions in every band. θ and ϕ are the spherical angular coordinates, P_l^m are the associated Legendre polynomials and K_l^m is the normalization factor:

$$K_l^m = \sqrt{\frac{(2l+1)}{4\pi}\frac{(l-m)!}{(l+m)!}} \tag{3}$$

The Legendre polynomial of degree l is defined as follows:

$$P_l(x) = \frac{1}{2^l l!}\frac{d^l}{dx^l}[(x^2-1)^l] \tag{4}$$

and the associated Legendre polynomials:

$$P_l{}^m(x) = (-1)^m \sqrt{(1-x^2)^m} \frac{d^m}{dx^m} P_l(x) \tag{5}$$

The previous definitions constitute complex SH. Below are given the real SH which are used for approximating illumination:

$$Y_l^m = \begin{cases} \sqrt{2} K_l^m sin(|m|\phi) P_l^{|m|}(cos\theta), & m < 0 \\ \sqrt{2} K_l^m cos(m\phi) P_l^m(cos\theta), & m > 0 \\ K_l^0 P_l^0(cos\theta), & m = 0 \end{cases} \tag{6}$$

A scalar function f that is defined over the sphere S can be projected to SH coefficients with the integral:

$$f_l{}^m = \int f(s) Y_l{}^m(s) ds \tag{7}$$

Using the coefficients $f_l{}^m$ it can be reconstructed an approximation $\widetilde{f}$ of the function f:

$$\widetilde{f}(s) = \sum_{l=1}^{n} \sum_{m=-l}^{l} f_l{}^m Y_l{}^m(s) \tag{8}$$

or alterantively:

$$\widetilde{f}(s) = \sum_{i=0}^{n^2} f_i Y_i(s) \tag{9}$$

where n^2 is the total number of coefficients and i is the number of the SH basis function ($i = l(l+1)+m$). SH can represent any frequency function if enough coefficients/bands are used. The more coefficients/bands are used for a function approximation the more accurate approximation is produced.

One important characteristic of SH is the way of handling rotations and specifically, the relation of rotations with projection. $g(s)$ is a function that represents the rotated function $f(s)$ by rotation matrix M. Rotate $f(s)$ with the rotation matrix M and then project it to SH gives the identical result when projecting $f(s)$ and then rotate it. Ivanic and Ruedenberg [8] proposed an efficient method for real SH rotations with the use of rotation matrices. They proposed a recurrence method for constructing the rotation matrices between real SH directly in terms of the elements of the original 3x3 rotation matrix without the intermediary of any parameters. Via a recursive approach the rotation matrix $R^l_{mm\prime}$ of real SH is derived using the orthogonal transformation R and the constructed SH rotation matrices of the previous bands. The rotated SH coefficients are transformed like below:

$$\widetilde{Y}^l_{m\prime} = \sum_{m=-l}^{l} Y^l_m R^l_{mm\prime} \tag{10}$$

Ramamoorthi and Hanrahan [1] were first introduced Spherical Harmonics (SH) for rendering diffuse surfaces shaded under distant illumination, represented by environment maps. Ramamoorthi and Hanrahan [1] have demonstrated that is sufficient to use only 9 light SH coefficients to have a good approximation of the distant illumination. These light coefficients are computed by taking the incident light on different directions over the hemisphere, and scaled them by the SH coefficients. Sloan et al. [3] extended Ramamoorthi and Hanrahan [1] method for rendering diffuse and glossy surfaces with soft shadows, interreflections and caustics using only 9 SH coefficients as proposed by Ramamoorthi and Hanrahan[1]. As a preprocess step, are created transfer functions at each point over the object's surface that represent the irradiance which include shadows and 2-bounce interreflections from the object onto itself. Transfer functions are precomputed for different number of reflection bounces. For view-independent diffuse reflections, incident lighting is transformed by a transfer vector into irradiance from each point. For view-dependent glossy reflections, a transfer matrix is computed to convert source illumination into a distribution of exit radiance that can be evaluated base on the view direction. At run-time, for diffuse surfaces these transfer vectors are applied to incident light and for glossy surfaces these transfer matrices are applied to incident light and convolved with the BRDF.

3. Our CGA PRT algorithm

In our illumination algorithm we project the environment light to CGA-SH basis functions to approximate the illumination using only 9 SH coefficients, as proposed by Ramamoorthi and Hanrahan [1]. Using only 9 SH coefficients is satisfactory to provide a good approximation of the environment light.

For the representation of SH we have used CGA. CGA extends the 3D euclidean GA with 3 basis vectors e_1, e_2 and e_3 from 3D space to 5D space by adding two additional basis vectors e_+ and e_-. With the combination of theses two basis vectors can be formed another two basis vectors e_0 that represents the 3D origin and e_∞ that represents infinity:

$$e_0 = \frac{1}{2}(e_- - e_+) \qquad e_\infty = e_- + e_+ \tag{11}$$

With these two additional basis vectors CGA can handle Conformal transformations and give the ability to represent basic geometric entities like spheres, points and lines.

3.1. Spherical Harmonics representation with CGA

For the computation of Spherical Harmonics basis functions we use spheres generated with Conformal Geometric Algebra. Each spherical function of the first band can be visualized and computed using two spheres. The first sphere corresponds to positive values and the second sphere corresponds to negative values. For computing the value of the spherical function on a specific direction we construct a line using two points: the origin and the direction in Cartesian coordinates. The SH value equals to the distance between the origin and the intersection point of the line and the sphere.The SH of the third band can be expressed in terms of SH of the second band as we will describe in more detail later.

Next, we describe in more detail how we represent SH with geometry entities by providing the needed Conformal Geometric Algebra operations.

1. We convert spherical angular coordinates (theta,phi) to spherical coordinates (x,y,z) like below:
   x = sin (theta)*cos(phi);
   y = sin (theta)*sin(phi);
   z = cos (theta);

2. We construct a line L between the point at the origin $e_0 = (0,0,0)$ and the spherical coordinates $P_1 = (x,y,z)$ computed on the previous step by the outer product $\wedge$ of the two points and the point at infinity (e_∞) using the OPNS (outer product null space) representation [7]

$$L* = P_1 \wedge e_0 \wedge e_\infty \tag{12}$$

3. We construct the two spheres that represent the Spherical Harmonics. For the SH function $Y_{1,-1}$ we construct two spheres placed on y axis, for $Y_{1,0}$ on z axis and for $Y_{1,1}$ on x axis. The radius r of each sphere is 0.5 because circular functions of SH return values in range [-1,1]. P is the center point of the first sphere S which is placed on distance 0.5 along the origin on the appropriate axis depending on the number of the SH function (m):

$$S = P - \frac{1}{2}r^2 e_\infty \tag{13}$$

   The second sphere is symmetrical to the first one, thus the radius of the second sphere is the same with the radius of the first one, and the center of the second sphere is $-P$:

$$S = -P - \frac{1}{2}r^2 e_\infty \tag{14}$$

4. We compute the point pair Pp that is formed by the line (L) and the sphere (S) by constructing the outer product $\wedge$ of L and S. Base on the spherical coordinates we are able to extract with which sphere the line intersects.The first sphere corresponds to the positive values and the second sphere to the negative values.

$$Pp = L \wedge S \tag{15}$$

5. We extract the point P of the point pair Pp that intersects with the sphere S as proposed by Hildenbrand et al. [7]. The second point P of the point pair Pp is the intersection point of the line L and sphere S:

$$P = \frac{-\sqrt{Pp^* \cdot Pp^*} + Pp^*}{e_\infty \cdot Pp^*} \tag{16}$$

6. The Euclidean distance d between the origin e_0 and the intersection point P, multiplied with the scalar part of SH equals to the CGA-SH value. If the line L intersects with the sphere that corresponds to positive values the CGA-SH is d else is $-d$.CGA-SH value
7. The SH basis functions of the third band can be expressed using the SH of the second band. Table 1 indicates the SH values of the first three bands. From this table derive the following equations for the SH of the third band:

$$
\begin{aligned}
Y_{2,-2}(\theta,\phi) &= \sqrt{\frac{15}{4\pi}} SH_{1,1}(\theta,\phi) SH_{1,-1}(\theta,\phi) \\
Y_{2,-1}(\theta,\phi) &= \sqrt{\frac{15}{4\pi}} SH_{1,-1}(\theta,\phi) SH_{1,0}(\theta,\phi) \\
Y_{2,0}(\theta,\phi) &= \sqrt{\frac{5}{16\pi}} (3SH^2_{1,0}(\theta,\phi) - 1) \\
Y_{2,1}(\theta,\phi) &= \sqrt{\frac{15}{4\pi}} SH_{1,1}(\theta,\phi) SH_{1,0}(\theta,\phi) \\
Y_{2,2}(\theta,\phi) &= \sqrt{\frac{15}{16\pi}} (SH^2_{1,1}(\theta,\phi) - SH^2_{1,-1}(\theta,\phi))
\end{aligned}
\tag{17}
$$

Table 1 indicates how we represent CGA-SH with CGA operations in terms Cartesian coordinates and CGA entities. The general representation to compute the CGA-SH for band 1 is given by:

$$SH_l^m =^{+}_{-} Euclidean(S \wedge L, e_0) \tag{18}$$

where L is is the line used to compute the CGA-SH value:

$$L = (vec3(x,y,z) \wedge e_0 \wedge e_\infty) \tag{19}$$

$Euclidean(a,b)$ is the Euclidean distance between point a and b and S represents the sphere which the line L intersects. When we construct the spheres for each SH function of band 1, we define which sphere returns positive or negative SH value. For example, for $l = 1$ and $m = -1$ the sphere S_{y+} is placed on the y positive axis and returns positive values and S_{y-} is placed symmetrically with S_{y+} and returns negative values. if L intersects with S_{y+} then:

$$SH_1^{-1} = Euclidean(S_{y+} \wedge L, e_0) \tag{20}$$

otherwise:

$$SH_1^{-1} = -Euclidean(S_{y-} \wedge L, e_0) \tag{21}$$

In this work, we only present how we compute SH of the first three bands, because are enough to approximate well the illumination for diffuse surfaces. When approximating a function by projecting it into the SH, the more SH bands are used for the approximation, the more accurate approximation is produced. Depending on the application requirements CGA-SH for bands with index $l > 3$ can be represented in terms of CGA-SH of band $l = 1$.

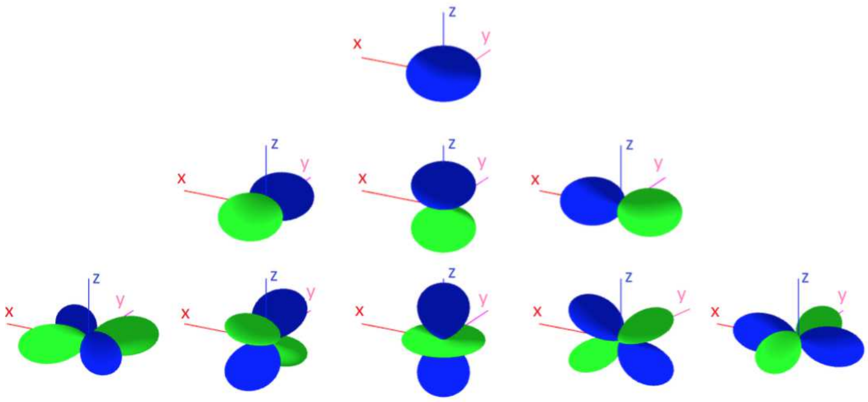


Figure 1. SH basis functions of the first 3 bands, generated with CGA. Green color represents negative values and blue positive values.

3.2. Precomputed Radiance Transfer (PRT) rendering with CGA-SH

To illuminate our models we have used the method proposed by Sloan et al. [3], and we have employed the CGA to represent the SH basis functions as described in previous section. We uniformly generate samples on the surface of a sphere using Monte Carlo Integration and then we project the incident light $L(\theta,\phi)$ that comes from these directions to CGA-SH basis functions using the Equation 7 in order to approximate the illumination with 9 SH light coefficients L_l^m:

$$L_l^m = \int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} L(\theta,\phi) Y_l{}^m(\theta,\phi) sin\theta d\theta d\phi \tag{22}$$

As a preprocessing step, we compute the transfer radiance E_v on each vertex v of the mesh. Because we focus on rendering only diffuse surfaces the light is reflected equally in all directions and depends only on the angle between the vertex normal and the light direction:

$$E_v = \frac{\rho_v}{\pi} \int_S L_i(v, \omega_i) Y_l{}^m(\theta, \phi) max(N_v \cdot \omega_i, 0) d\omega_i \tag{23}$$

where ρ_v is the albedo of the vertex v, $L_i(v, \omega_i)$ is the incident light on vertex v from direction ω_i and N_v is the normal of vertex v. For integrating shadows to the vertex coefficients we construct a Binary space partitioning(BSP) tree and specifically a KD-Tree. A BSP tree subdivides hierarchically a space into irregularly sized sub-spaces. The casting ray from a point traverses the tree from the root node to the leafs to detect if it intersects with any triangle in the scene. To add shadows to vertex coefficients $E_{v,l,m}$ we add the visibility term $V(\omega_i)$ to the integral:

$$E_{v,l,m} = \frac{\rho_v}{\pi} \int_S L_i(v, \omega_i) Y_l{}^m(\theta, \phi) max(N_v \cdot \omega_i, 0) V(\omega_i) d\omega_i \tag{24}$$

To compute the final color E_v on each vertex we sum up the product of light coefficients L_l^m with vertex coefficients $E_{v,l,m}$:

$$E_v = \sum_l^n \sum_{-l}^l E_{v,l,m} L_l^m \tag{25}$$

where n is the number of bands.

3.3. Rotation of CGA-SH

We rotate the SH light coefficients by rotating entities generated with CGA operations. Specifically, we use the CGA-SH described in section 3.1 and a line that is predefined for each SH basis function.

We achieve rotations by multiplying each SH light coefficient with a scalar co-sinusoidal value computed with the help of the CGA-SH normalized entities. In the following steps we describe in more detail how we achieved CGA-SH rotations:

1. Firstly, we place a line $Line_{l,m}$ on every CGA-SH entity. A line in CGA is generated using two points that lie on it and the point at the infinity (e_∞):
$$L^* = P_1 \wedge P_2 \wedge e_\infty \tag{26}$$
The first point P_1 of the line is the point with the maximum distance from the origin, that gives a positive value as shown on Figure 2 and the second point P_2 is the origin e_0. On Figure 2 we demonstrate visually the initial orientation of the line on each SH basis function that is denoted with black color.
2. To compute the CGA-SH rotation we construct a rotor R that corresponds to the desired rotation, which transforms each line $Line_{l,m}$ of each SH basis function. To rotate the line we sandwich it between the rotor R and its inverse rotor R^{-1} as shown below:
$$R Line_{l,m} R^{-1} \tag{27}$$
3. The rotation value equals to the distance $d_{l,m}$ of the intersection point of the rotated line $Line_{l,m}$ and the CGA-SH entity to the origin e_0. We extract the intersection point and the distance as describe in section 3.1. The product of the light coefficient L_l^m and the distance d_l^m equals to the rotated light coefficient $L_l'^m$:
$$L_l'^m = d_l^m L_l^m \tag{28}$$

When placing lines as shown on Figure 2 we can notice that SH basis function remain rotation invariant under specific rotations as they suppose to be. For instance, for CGA-SH basis functions with index 2 ($l = 1$ and $m = 0$) and 6 ($l = 2$ and $m = 0$) we notice that they remain rotation invariant under rotations on z axis because the intersection point does not change during rotation.

4. Results

In this section we present our results with different environment lights on static surfaces. Table 2 indicates the 9 light coefficients obtained from the projection of light:

$$light(\theta, \phi) = max(0, 5cos(\theta) - 4) + max(0, -4sin(\theta, -\pi) * cos(\theta - 2.5) - 3) \tag{29}$$

as illustrated on Figure 3 to CGA-SH basis functions and the rotated light coefficients using CGA rotation method. The static object is placed such that the one part of the light in the upper hemisphere illuminates the object from above and the other part of the light illuminates the object from the back side. On Table 2 we notice that by applying rotation to light 190^o around y axis the object is illuminated from the front side.

Our CGA rotation method is slightly faster than the efficient rotation matrices proposed by Ivanic [8]. Table 3 shows average time in msecs for applying rotation to light coefficients with rotation matrices [8] and our CGA-SH rotation method.

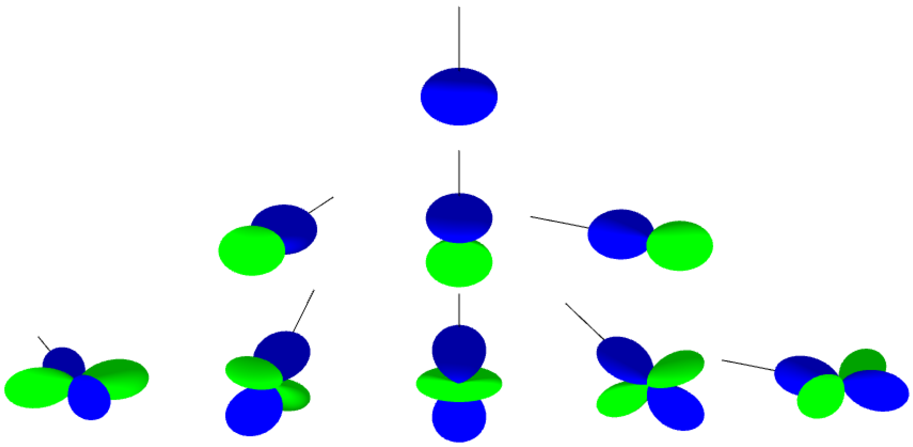


Figure 2. The lines defined for each SH basis function to rotate the SH light coefficients

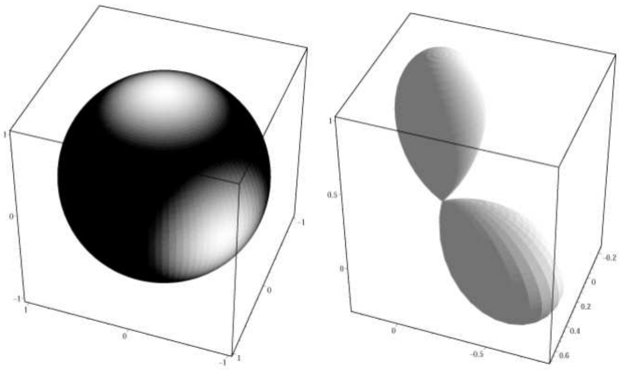


Figure 3. An example Light provided by Green [12] which illuminates the object of Figure 2

On Tables 4, 5 and 6 we show our illumination and rotation results produced using irradiance maps. We demonstrate our results for bands 3-6 however, the results produced do not have any significant visual difference. On each case we show the rotation matrix computed with Ivanic method [8] and the CGA rotor used to compute the rotation of light coefficients. On these tables we also present the geometric intuition of CGA-SH rotations using CGA in contrast with rotation matrices that do not allow the visualization of rotations. Using the rotor we rotate the lines (denoted with black color) defined for each CGA basis function. To compute the rotated light coefficients we multiply the light coefficients with the distance of e_0 and the intersection point of the line and the CGA-SH entity. Table 7 shows the Mean Square Error (%) produced on some example rotations when rotating with CGA rotors compared to Rotation matrices. Also, Tables 9, 8, and 10 show the Mean Square Error (%) produced when rotating on x, y, z axis respectively with CGA rotors compared to Rotation matrices using 3 different environment maps. We give the range of error produced when rotating on specific range of rotations on each axis. On Table 11 we present the qualitative comparison of rotations using CGA rotors and Rotation Matrices compared to Monte-Carlo global illumination produced by Maya software. You can find a video with our results in the following link: `https://dl.dropboxusercontent.com/u/62639752/ENGAGE2017.mp4`.

5. Conclusions and Future work

In this work, we have presented a method for handling illumination of diffuse surfaces of static objects under distant illumination in real-time, using CGA. We achieve rotation of light by rotating CGA entities thus, we are able to present the rotation of the light in a visual way and make them more understandable, in contrast with the transformation matrices that cannot present rotations in a visual way. Our algorithm runs at interactive frame rates with faster SH rotation performance compared to Ivanic rotation matrices. The more light coefficients are used the more computationally fast is our CGA-SH rotation algorithm compared to Ivanic rotation matrices.

In the future, we aim to use CGA to translate and scale SH light coefficients as CGA allows handling translation and scale. In this way, we will create a single algebraic framework that handles both distant illumination in real-time and animation interpolation and skinning [4], and avoid the generation and usage of large transformation matrices, which hopefully will increase

Table 1. Spherical Harmonics values of the first three bands. 1st column: gives the band index, 2nd column: illustrates the SH value in terms of spherical angles, 3rd column: gives the SH value in terms of Cartesian coordinates, and 4th column: represents the SH values in terms of CGA operations. Spheres S_x, S_y and S_z correspond to sphere that the line $(*(vec3(x,y,z) \wedge e_2 \wedge e_\infty)$ intersects. Depending on which of the two spheres the line intersects, the Euclidean distance takes positive or negative sign($Euclidean(a,b)$ gives the distance between points a and b).

Band Index	Spherical	Cartesian	CGA-SH
l=0	$Y_{0,0}(\theta,\phi) = \sqrt{\frac{1}{4\pi}}$	$\sqrt{\frac{1}{4\pi}}$	$\sqrt{\frac{1}{4\pi}}$
	$Y_{1,-1}(\theta,\phi) = \sqrt{\frac{3}{4\pi}} sin\phi sin\theta$	$\sqrt{\frac{3}{4\pi}}\frac{y}{r}$	$\sqrt{\frac{3}{4\pi}} SH_{1,-1}(\theta,\phi) = \sqrt{\frac{3}{4\pi}} Euclidean(S_y \wedge (*(vec3(x,y,z) \wedge e_0 \wedge e_\infty)), e_0)$
l=1	$Y_{1,0}(\theta,\phi) = \sqrt{\frac{3}{4\pi}} cos\theta$	$\sqrt{\frac{3}{4\pi}}\frac{z}{r}$	$\sqrt{\frac{3}{4\pi}} SH_{1,0}(\theta,\phi) = \sqrt{\frac{3}{4\pi}} Euclidean(S_z \wedge (*(vec3(x,y,z) \wedge e_0 \wedge e_\infty)), e_0)$
	$Y_{1,1}(\theta,\phi) = \sqrt{\frac{3}{4\pi}} cos\phi sin\theta$	$\sqrt{\frac{3}{4\pi}}\frac{x}{r}$	$\sqrt{\frac{3}{4\pi}} SH_{1,1}(\theta,\phi) = \sqrt{\frac{3}{4\pi}} Euclidean(S_x \wedge (*(vec3(x,y,z) \wedge e_0 \wedge e_\infty)), e_0)$
	$Y_{2,-2}(\theta,\phi) = \sqrt{\frac{15}{16\pi}} sin(2\phi) sin^2\theta$	$\sqrt{\frac{15}{4\pi}}\frac{xy}{r^2}$	$\sqrt{\frac{15}{16\pi}} SH_{1,-1}(\theta,\phi) SH_{1,1}(\theta,\phi)$
	$Y_{2,-1}(\theta,\phi) = \sqrt{\frac{15}{4\pi}} sin\phi sin\theta cos\theta$	$\sqrt{\frac{15}{4\pi}}\frac{yz}{r^2}$	$\sqrt{\frac{15}{4\pi}} SH_{1,-1}(\theta,\phi) SH_{1,0}(\theta,\phi)$
l=2	$Y_{2,0}(\theta,\phi) = \sqrt{\frac{5}{16\pi}} (3cos^2\theta - 1)$	$\sqrt{\frac{5}{16\pi}}(\frac{3z^2}{r^2} - 1)$	$\sqrt{\frac{5}{16\pi}}(3SH^2_{1,0}(\theta,\phi)) - 1)$
	$Y_{2,1}(\theta,\phi) = \sqrt{\frac{15}{4\pi}} cos\phi sin\theta cos\theta$	$\sqrt{\frac{15}{4\pi}}\frac{xz}{r^2}$	$\sqrt{\frac{15}{4\pi}} SH_{1,1}(\theta,\phi) SH_{1,0}(\theta,\phi)$
	$Y_{2,2}(\theta,\phi) = \sqrt{\frac{15}{16\pi}} cos(2\phi) sin^2\theta$	$\sqrt{\frac{15}{16\pi}}\frac{x^2-y^2}{r^2}$	$\sqrt{\frac{15}{16\pi}}(SH^2_{1,1}(\theta,\phi) - SH^2_{1,-1}(\theta,\phi))$
	$Y_{3,-3}(\theta,\phi) =$	$\frac{1}{4}\sqrt{\frac{35}{2\pi}}\frac{(3x^2-y^2)y}{r^3}$	$\frac{1}{4}\sqrt{\frac{35}{2\pi}}(3SH^2_{1,1}(\theta,\phi) - SH^2_{1,-1}(\theta,\phi))SH_{1,-1}(\theta,\phi)$
	$Y_{3,-2}(\theta,\phi) =$	$\frac{1}{2}\sqrt{\frac{105}{\pi}}\frac{xyz}{r^3}$	$\frac{1}{2}\sqrt{\frac{105}{\pi}} SH_{1,-1}(\theta,\phi) SH_{1,0}(\theta,\phi) SH_{1,1}(\theta,\phi)$
	$Y_{3,-1}(\theta,\phi) =$	$\frac{1}{4}\sqrt{\frac{21}{2\pi}}\frac{y(4z^2-x^2-y^2)}{r^3}$	$\frac{1}{4}\sqrt{\frac{21}{2\pi}} SH_{1,-1}(\theta,\phi)(4SH^2_{1,0}(\theta,\phi) - SH^2_{1,-1}(\theta,\phi) - SH^2_{1,-1}(\theta,\phi))$
l=3	$Y_{3,0}(\theta,\phi) =$	$\frac{1}{4}\sqrt{\frac{7}{\pi}}\frac{z(2z^2-3x^2-3y^2)}{r^3})$	$\frac{1}{4}\sqrt{\frac{7}{\pi}} SH_{1,0}(\theta,\phi)(2SH^2_{1,0}(\theta,\phi) - 3SH^2_{1,-1}(\theta,\phi) - 3SH^2_{1,-1}(\theta,\phi)))$
	$Y_{3,1}(\theta,\phi) =$	$\frac{1}{4}\sqrt{\frac{21}{2\pi}}\frac{x(4z^2-x^2-y^2)}{r^3}$	$\frac{1}{4}\sqrt{\frac{21}{2\pi}} SH_{1,-1}(\theta,\phi)(4SH^2_{1,0}(\theta,\phi) - SH^2_{1,-1}(\theta,\phi) - SH^2_{1,-1}(\theta,\phi))$
	$Y_{3,2}(\theta,\phi) =$	$\frac{1}{4}\sqrt{\frac{105}{\pi}}\frac{(x^2-y^2)z}{r^3}$	$\frac{1}{4}\sqrt{\frac{105}{\pi}}(SH^2_{1,-1}(\theta,\phi) - SH^2_{1,-1}(\theta,\phi))SH_{1,0}(\theta,\phi)$
	$Y_{3,3}(\theta,\phi) =$	$\frac{1}{4}\sqrt{\frac{35}{2\pi}}\frac{(x^2-3y^2)x}{r^3}$	$\frac{1}{4}\sqrt{\frac{35}{2\pi}}(SH^2_{1,-1}(\theta,\phi) - 3SH^2_{1,-1}(\theta,\phi))SH_{1,-1}(\theta,\phi)$

the performance of the algorithm in order to be applicable to deformable characters. Also, we are planning to achieve more interactive frame rates by developing the illumination algorithm on GPU. Due to the reason that we use Gaalop precompiler to generate C++ code that is not depended on any libraries, it will be simple to employ our algorithm on GPU.

Acknowledgement

The research leading to these results has received funding from the European Union People Programme (FP7-PEOPLE-2013-ITN) under grant agreement n^O 608013.

Table 2. The light coefficients arising by projecting the light of Figure 2 to CGA-SH basis functions, and the rotated light coefficients around y axis by 190^o using the CGA rotation method. On the last row we show the result by illuminating an object with the initial light and the rotated light coefficients

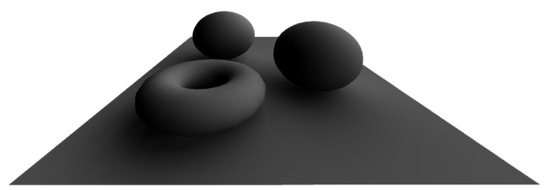
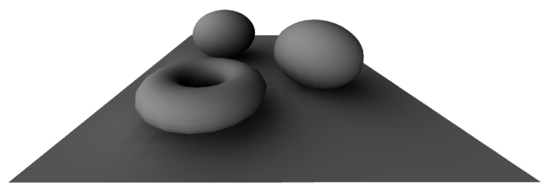
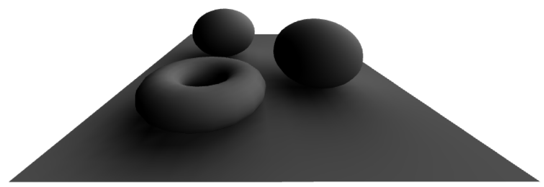
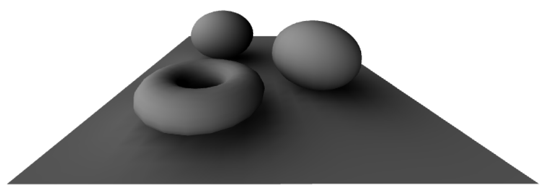
Light Coefficients	0.39809 -0.21072 0.28598 0.28165 -0.31430 -0.00145 0.13110 0.00041 0.09321	0.39809 -0.21072 -0.28164 -0.27737 0.309526 0.001428 0.12517 0.00038 0.09039
Rotation	no rotation	190^o around y axis
CGA-SH rotation algorithm (3 bands)		
Ivanic rotation matrices [8] (3 bands)		

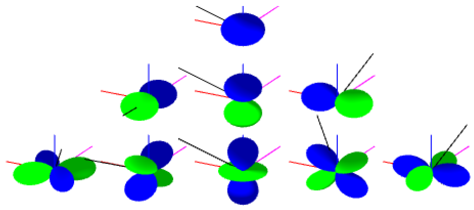
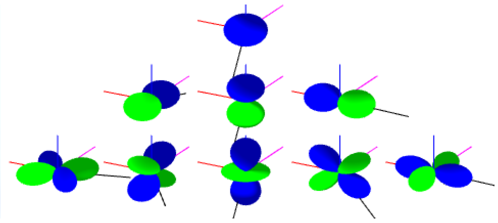
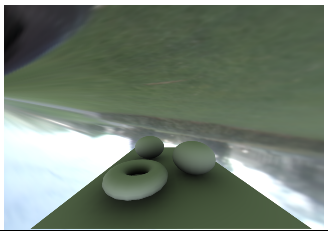
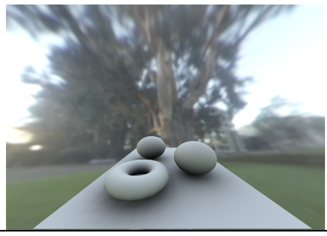
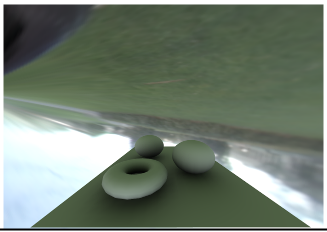
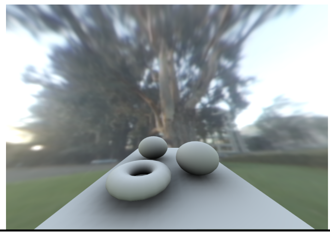
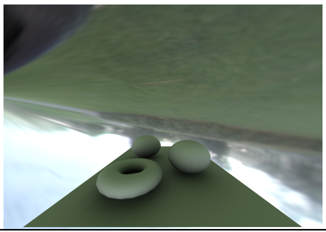
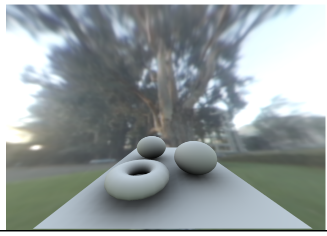
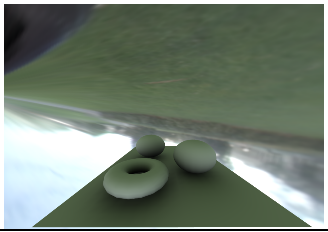
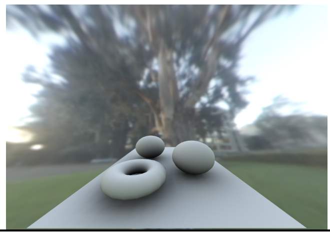
Table 3. Average time in msecs for applying rotation to light coefficients of two different datasets for each band with rotation matrices [8] and CGA rotors.

Number of bands	Rotation matrices [8]	CGA rotors (our method)
3	9.746e-06	2.239e-06
3	10.054e-06	2.312e-06
4	16.141e-06	3.211e-06
4	18.357e-06	3.749e-06
5	31.183e-06	4.991e-06
5	33.615e-06	4.948e-06
6	49.796e-06	6.074e-06
6	50.510e-06	6.141e-06

References

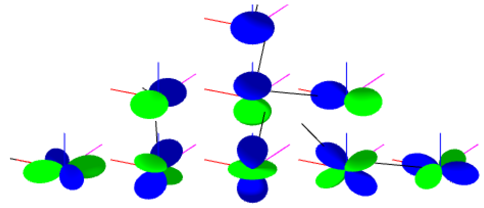
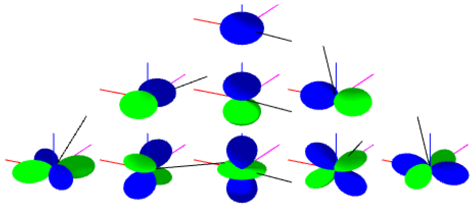
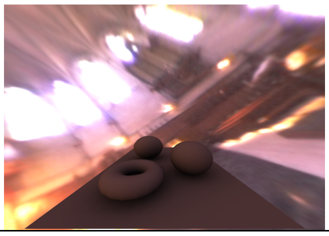
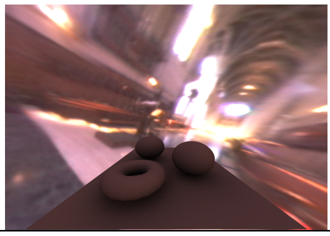
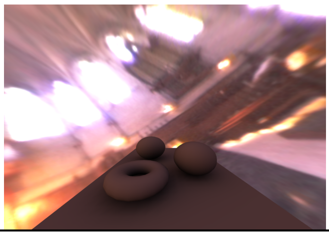
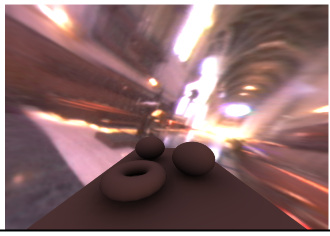
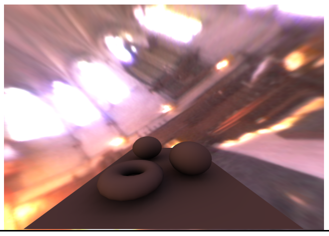
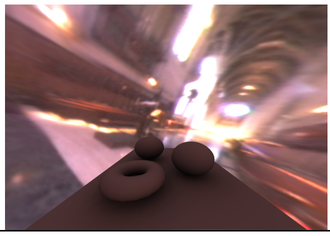
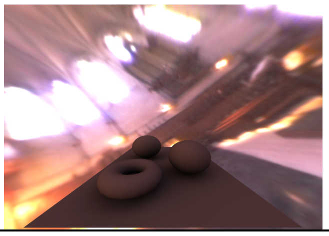
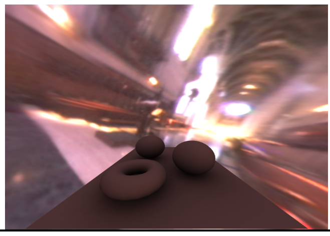
1. Ramamoorthi R. and Hanrahan P. An efficient representation for irradiance environment maps. *In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 497-500, New York, NY, USA, 2001. ACM.
2. Dorst L., Fontijne D., and S. Mann. Geometric Algebra for Computer Science. Morgan Kaufmann, 2007.
3. Sloan P., Kautz J., and Snyder J. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 02, pages 527-536, New York, NY, USA, 2002. ACM.
4. Papaefthymiou M., Hildenbrand D. and Papagiannakis G., An inclusive conformal geometric algebra GPU animation interpolation and deformation algorithm The Visual Computer, pages 19, 2016, also presented in CGI 2016, Heraklion, Greece
5. Iwasaki K., Furuya W., Dobashi Y. and Nishita T. Real- time rendering of dynamic scenes under all-frequency lighting using integral spherical gaussian. Comput. Graph. Forum, 31(2pt4):727734, May 2012.
6. Xu K. , Wei-Lun S., Zhao D., Dan-Yong Z., Run-Dong W., and Shi-Min H.. Anisotropic spherical gaussians. ACM Transactions on Graphics, 32(6):209:1209:11, 2013.
7. Hildenbrand D. Foundations of Geometric Algebra Computing, volume 8. Springer, 2013.

Table 4. Two different examples of illumination rotation using our CGA-SH rotation algorithm. On each case we provide the rotation applied to the light coefficients in rotation matrix representation computed with Ivanic method [8], in CGA rotor representation computed with our CGA-SH rotation algorithm and in euler angles. Also, on each case we present the geometric intuition of CGA-SH rotations using CGA and the result of illuminating a static object with the the rotated light coefficients computed with our CGA-SH rotation algorithm

Rotation Matrix (3 bands)	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.967 & -0.202 & -0.149 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.053 & 0.414 & -0.908 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.245 & -0.887 & -0.390 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.340 & 0.809 & 0.311 & 0.211 & 0.295 \\ 0 & 0 & 0 & 0 & 0.435 & -0.400 & -0.072 & -0.061 & 0.093 \\ 0 & 0 & 0 & 0 & -0.084 & 0.038 & -0.242 & -0.651 & 0.712 \\ 0 & 0 & 0 & 0 & -0.244 & 0.054 & -0.636 & 0.644 & 0.341 \\ 0 & 0 & 0 & 0 & -0.240 & -0.413 & 0.646 & 0.316 & 0.503 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.948 & 0.316 & 0.019 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.317 & -0.947 & -0.043 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.004 & 0.046 & -0.998 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.947 & -0.045 & 0.026 & -0.315 & -0.023 \\ 0 & 0 & 0 & 0 & -0.017 & -0.898 & -0.259 & -0.018 & -0.151 \\ 0 & 0 & 0 & 0 & -0.024 & -0.521 & 0.846 & 0.071 & -0.086 \\ 0 & 0 & 0 & 0 & -0.317 & 0.011 & -0.077 & 0.944 & 0.042 \\ 0 & 0 & 0 & 0 & -0.022 & -0.300 & -0.085 & -0.052 & 0.948 \end{bmatrix}$
Rotor	[0.118 0.539 0.044 -0.832]	[0.023 0.006 0.986 0.160]
Euler angles	x: 7.44, y: 65.13, z: -159.08	x: 161.54, y: 2.48, z: 178.89
CGA intuition (3 bands)		
Illumination with our CGA-SH rotation algorithm (3 bands)		
Illumination with our CGA-SH rotation algorithm (4 bands)		
Illumination with our CGA-SH rotation algorithm (5 bands)		
Illumination with our CGA-SH rotation algorithm (6 bands)		

8. Ivanic J. and Ruedenberg K.: Additions and corrections: rotation matrices for real spherical harmonics. j. phys. chem. 45(102):90999100, 1998.
9. Perwass C. Geometric Algebra with Applications in Engineerings. Springer, 2009.
10. Hitzer E. Introduction to clifford s geometric algebra. SICE Journal of Control, Measurement, and System Integration, 4:001011, 2011.

Table 5. Another two different examples of illumination rotation using our CGA-SH rotation algorithm using a different environment map. On each case we provide the rotation applied to the light coefficients in rotation matrix representation computed with Ivanic method [8], in CGA rotor representation computed with our CGA-SH rotation algorithm and in euler angles. Also, on each case we present the geometric intuition of CGA-SH rotations using CGA and the result of illuminating a static object with the the rotated light coefficients computed with our CGA-SH rotation algorithm

<table>
<tr><td>Rotation Matrix</td><td>$\begin{array}{|c|c|c|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0.719 & 0.301 & 0.625 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & -0.369 & 0.929 & -0.022 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & -0.587 & -0.219 & 0.779 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0.194 & -0.332 & -0.112 & 0.100 & 0.911 \\ \hline 0 & 0 & 0 & 0 & -0.123 & 0.669 & 0.242 & 0.581 & 0.126 \\ \hline 0 & 0 & 0 & 0 & 0.014 & -0.595 & 0.794 & -0.035 & -0.118 \\ \hline 0 & 0 & 0 & 0 & -0.275 & -0.466 & -0.346 & 0.729 & -0.234 \\ \hline 0 & 0 & 0 & 0 & -0.908 & -0.090 & -0.038 & -0.356 & 0.195 \\ \hline \end{array}$</td><td>$\begin{array}{|c|c|c|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0.934 & -0.077 & -0.350 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0.358 & 0.161 & 0.919 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & -0.014 & -0.984 & 0.178 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0.171 & -0.917 & 0.131 & 0.331 & -0.049 \\ \hline 0 & 0 & 0 & 0 & 0.366 & 0.150 & -0.011 & -0.056 & -0.328 \\ \hline 0 & 0 & 0 & 0 & 0.571 & 0.099 & -0.461 & 0.256 & 0.621 \\ \hline 0 & 0 & 0 & 0 & 0.050 & -0.355 & -0.274 & -0.876 & 0.169 \\ \hline 0 & 0 & 0 & 0 & 0.324 & 0.086 & 0.833 & -0.202 & 0.390 \\ \hline \end{array}$</td></tr>
<tr><td>Rotor</td><td>[-0.925 0.181 0.0522 -0.327]</td><td>[-0.753 -0.144 0.631 0.111]</td></tr>
<tr><td>Euler angles</td><td>x: -21.70, y: 1.25, z: 38.69</td><td>x: 65.79, y: -66.68, z: -63.06</td></tr>
<tr><td>CGA intuition</td><td></td><td></td></tr>
<tr><td>Illumination with our CGA-SH rotation algorithm (3 bands)</td><td></td><td></td></tr>
<tr><td>Illumination with our CGA-SH rotation algorithm (4 bands)</td><td></td><td></td></tr>
<tr><td>Illumination with our CGA-SH rotation algorithm (5 bands)</td><td></td><td></td></tr>
<tr><td>Illumination with our CGA-SH rotation algorithm (6 bands)</td><td></td><td></td></tr>
</table>

11. Sommer G. Geometric computing with Clifford algebras: theoretical foundations and applications in computer vision and robotics. Springer London, 2001.
12. Green, R., Spherical Harmonic Lighting: The gritty details, GDC2003, http://silviojemma.com/public/papers/lighting/spherical-harmonic-lighting.pdf/, last accessed at 17/02/2017

Table 6. Another two different examples of illumination rotation using our CGA-SH rotation algorithm using a different environment map. On each case we provide the rotation applied to the light coefficients in rotation matrix representation computed with Ivanic method [8], in CGA rotor representation computed with our CGA-SH rotation algorithm and in euler angles. Also, on each case we present the geometric intuition of CGA-SH rotations using CGA and the result of illuminating a static object with the the rotated light coefficients computed with our CGA-SH rotation algorithm

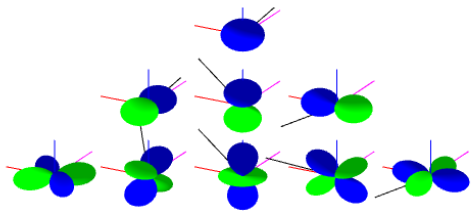
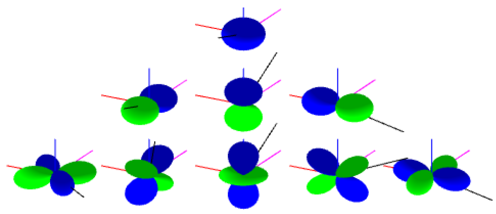
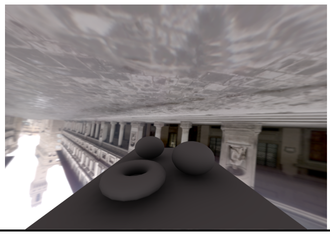
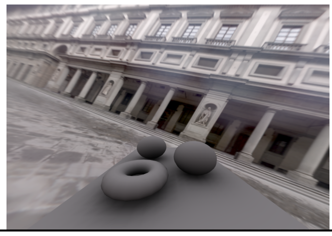
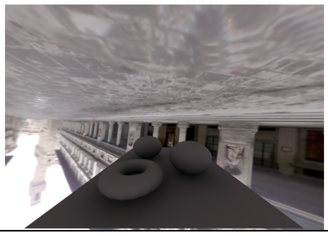
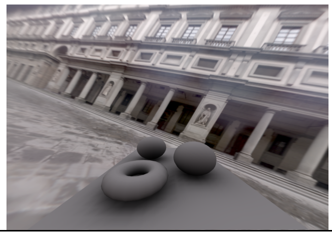
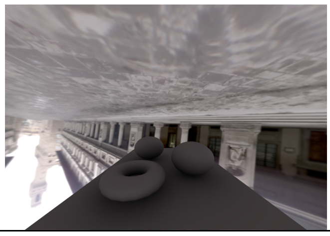
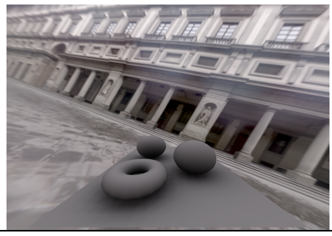
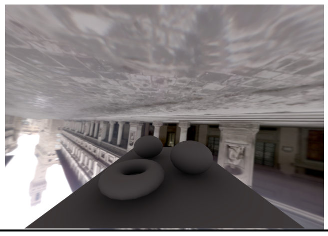
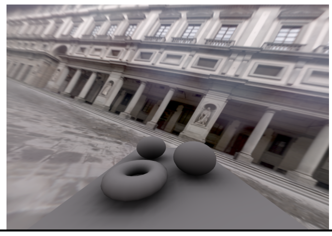
<table>
<tr><td>Rotation Matrix (3 bands)</td><td>
<table>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0.996</td><td>0.0822</td><td>-0.031</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>-0.047</td><td>0.795</td><td>0.604</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0.074</td><td>-0.600</td><td>0.796</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0.791</td><td>-0.592</td><td>-0.085</td><td>0.084</td><td>-0.098</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0.301</td><td>0.792</td><td>0.056</td><td>-0.025</td><td>0.014</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>-0.049</td><td>-0.064</td><td>0.449</td><td>0.832</td><td>0.314</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0.007</td><td>0.087</td><td>-0.827</td><td>0.271</td><td>0.484</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0.089</td><td>-0.126</td><td>0.306</td><td>-0.475</td><td>0.809</td></tr>
</table>
</td><td>
<table>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>-0.911</td><td>-0.413</td><td>0.013</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>-0.391</td><td>0.873</td><td>0.289</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>-0.131</td><td>0.258</td><td>-0.957</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0.870</td><td>-0.181</td><td>-0.184</td><td>0.398</td><td>-0.132</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>-0.134</td><td>-0.795</td><td>-0.312</td><td>0.011</td><td>-0.176</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>-0.196</td><td>-0.592</td><td>0.644</td><td>0.437</td><td>-0.060</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0.337</td><td>-0.215</td><td>0.391</td><td>-0.761</td><td>-0.328</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0.137</td><td>-0.409</td><td>-0.089</td><td>-0.241</td><td>0.864</td></tr>
</table>
</td></tr>
<tr><td>Rotor</td><td>[0.028 0.947 0.034 0.318]</td><td>[0.967 -0.037 0.141 -0.208]</td></tr>
<tr><td>Euler angles</td><td>x: -3.37, y: -37.16, z: -2.22</td><td>x: -24.15, y: -16.80, z: 179.22</td></tr>
<tr><td>CGA intuition (3 bands)</td><td></td><td></td></tr>
<tr><td>Illumination with our CGA-SH rotation algorithm (3 bands)</td><td></td><td></td></tr>
<tr><td>Illumination with our CGA-SH rotation algorithm (4 bands)</td><td></td><td></td></tr>
<tr><td>Illumination with our CGA-SH rotation algorithm (6 bands)</td><td></td><td></td></tr>
<tr><td>Illumination with our CGA-SH rotation algorithm (6 bands)</td><td></td><td></td></tr>
</table>

Table 7. Mean Square Error (%) produced when rotating with CGA rotors compared to Rotation matrices

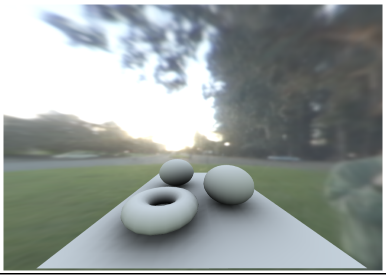
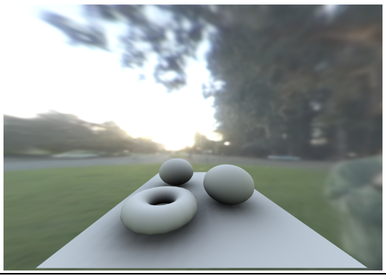
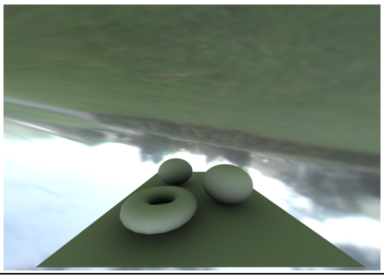
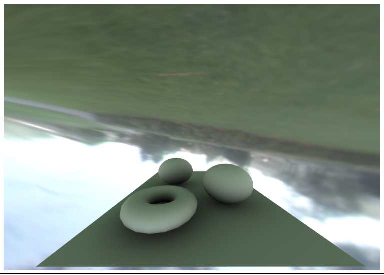
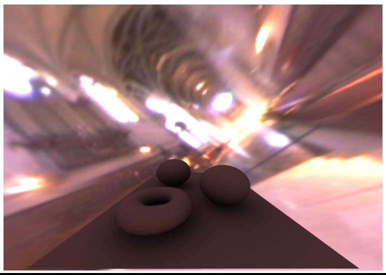
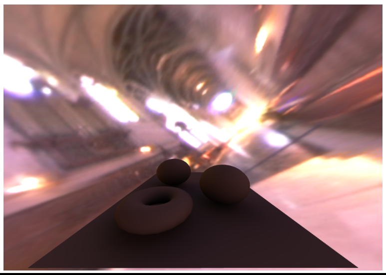
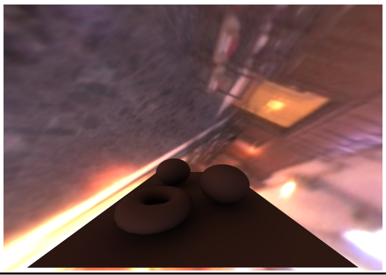
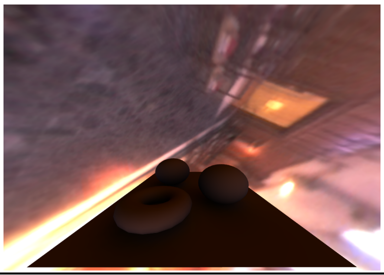
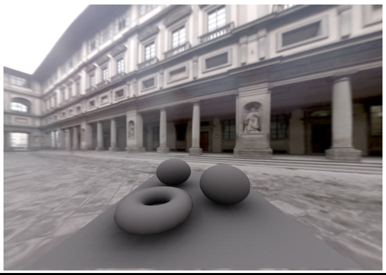
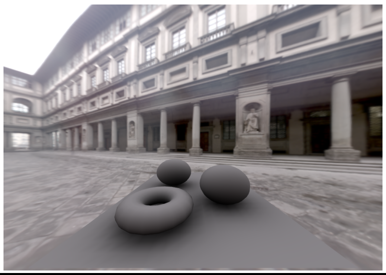
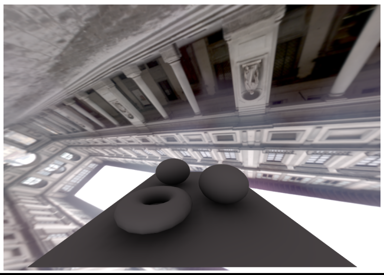
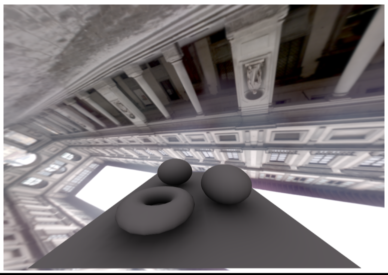
Rotation	CGA rotors	Rotation matrices	MSE (%)
[-0.54, -0, 0.83, -0]			2.97
[0.08, 0.59, 0.11, 0.79]			2.62
[-0.72, 0.28, 0.58, -0.23]			2.71349
[0.19, -0.67, 0.39, -0.59]			3.08619
[-0.23, -0, 0.97, -0]			1.52144
[-0.09, -0.21, 0.41, 0.87]			4.15277

Table 8. Mean Square Error (%) produced when rotating on y axis with CGA rotors compared to Rotation matrices when using 3 different environment maps. We give the range of error produced when rotating on specific range of rotations

Bands	$0^o - 90^o$	$91^o - 180^o$	$181^o - 270^o$	$271 - 360^o$
3	0-2.76	2.76-0	0-2.77	2.77-0
4	0-2.82	2.82-0	0-2.84	2.84-0
5	0-2.75	2.75-0	0-2.78	2.78-0
6	0-2.74	2.74-0	0-2.79	2.79-0
3	0-1.42	1.42-0	0-1.37	1.37-0
4	0-1.44	1.44-0	0-1.39	1.39-0
5	0-1.46	1.46-0	0-1.41	1.41-0
6	0-1.46	1.46-0	0-1.40	1.40-0
3	0-3.48	3.48-0	0-3.48	3.48-0
4	0-3.62	3.62-0	0-3.62	3.62-0
5	0-3.77	3.77-0	0-3.78	3.78-0
6	0-3.77	3.77-0	0-3.77	3.77-0

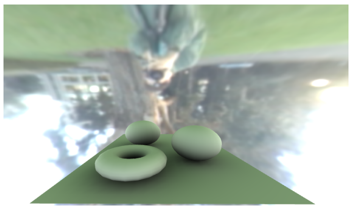
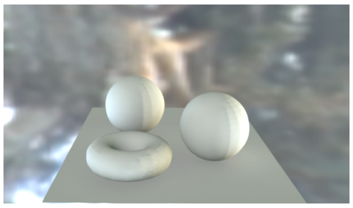
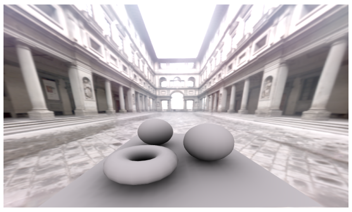
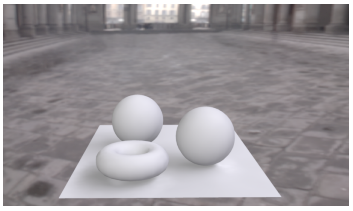
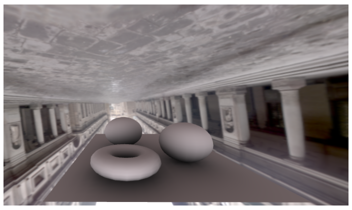
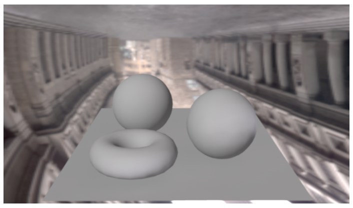
Table 9. Mean Square Error (%) produced when rotating on x axis with CGA rotors compared to Rotation matrices when using 3 different environment maps. We give the range of error produced when rotating on specific range of rotations

Bands	$0^o - 90^o$	$91^o - 180^o$	$181^o - 270^o$	$271 - 360^o$
3	0-9.73	9.73-0	0-11.10	11.10-0
4	0-9.79	9.79-0	0-11.78	11.78-0
5	0-9.83	9.83-0	0-11.90	11.90-0
6	0-9.81	9.81-0	0-11.90	11.90-0
3	0-3.37	3.37-0	0-3.40	3.40-0
4	0-3.39	3.39-0	0-3.43	3.43-0
5	0-3.42	3.42-0	0-3.40	3.40-0
6	0-3.43	3.43-0	0-3.41	3.41-0
3	0-8.37	8.37-0	0-3.25	3.25-0
4	0-8.39	8.39-0	0-3.24	3.24-0
5	0-8.48	8.48-0	0-3.23	3.23-0
6	0-8.48	8.48-0	0-3.23	3.23-0

Table 10. Mean Square Error (%) produced when rotating on z axis with CGA rotors compared to Rotation matrices when using 3 different environment maps. We give the range of error produced when rotating on specific range of rotations

Bands	$0^o - 90^o$	$91^o - 180^o$	$181^o - 270^o$	$271 - 360^o$
3	0-10.68	10.68-0	0-10.68	10.68-0
4	0-10.61	10.61-0	0-10.61	10.61-0
5	0-10.61	10.61-0	0-10.61	10.61-0
6	0-10.60	10.6-0	0-10.60	10.60-0
3	0-3.23	3.23-0	0-3.23	3.23-0
4	0-3.23	3.23-0	0-3.23	3.23-0
5	0-3.23	3.23-0	0-3.23	3.23-0
6	0-3.24	3.24-0	0-3.24	3.24-0
3	0-5.48	5.48-0	0-5.48	5.48-0
4	0-5.48	5.48-0	0-5.48	5.48-0
5	0-5.48	5.48-0	0-5.48	5.48-0
6	0-5.46	5.46-0	0-5.46	5.46-0

Table 11. Qualitative comparison of rotations using CGA rotors and Rotation Matrices compared to Monte-Carlo global illumination produced by Maya software

Rotation	CGA rotors	Rotation Matrices	Monte-Carlo (Maya)
[-0.20,0.97,0,0]			
[0.72,0,-0.692,0]			
[-0.03,0.71,-0.04,0.70]			
[0.06,0,-0.99,0]