Robotica: page 1 of 19. © Cambridge University Press 2016
doi:10.1017/S026357471600014X

Forward projection model of non-central catadioptric cameras with spherical mirrors

Nuno Goncalves*, Ana Catarina Nogueira and Andre Lages Miguel

Institute of Systems and Robotics, University of Coimbra
Polo 2, Pinhal de Marrocos, 3030-290 Coimbra, Portugal. E-mails: anacatnog@isr.uc.pt, andrelages@isr.uc.pt

(Accepted February 28, 2016)

SUMMARY
Non-central catadioptric vision is widely used in robotics and vision but suffers from the lack of an explicit closed-form forward projection model (FPM) that relates a 3D point with its 2D image. The search for the reflection point where the scene ray is projected is extremely slow and unpractical for real-time applications. Almost all methods thus rely on the assumption of a central projection model, even at the cost of an exact projection.

Two recent methods are able to solve this FPM, presenting a quasi-closed form FPM. However, in the special case of spherical mirrors, further enhancements can be made. We compare these two methods for the computation of the FPM and discuss both approaches in terms of practicality and performance. We also derive new expressions for the FPM on spherical mirrors (extremely useful to robotics and graphics) which speed up its computation.

KEYWORDS: non central catadioptric cameras, forward projection model, quadric mirrors, reflection point

1. Introduction

In computer vision, in the graphics industry or even in optics, the problem of computing reflections through mirrors can be regarded as a direct or inverse problem. The direct problem is the computation of the image pixel where a given 3D point is projected (also called the FPM) and the inverse problem starts with the image pixel and ends at the 3D point or direction in the space of the incident light ray. While inverse computation of reflections is straightforward, tracking the light path from the pixel to the 3D world scene, the counterpart is not as easy as the former case.

To better understand the image formation process when using reflectors to deviate the light ray direction, let us consider a vision system composed of a camera and a curved reflector. The geometric properties of the reflector, the camera and their relative position and orientation determines if the projection is called central or non-central. Central projection systems are those in which all incident light rays intersect each other in a single viewpoint; they are also difficult to construct since slight changes in the position of the camera centre of projection (COP) can affect the necessary conditions for centrality. Despite this fact, they are widely used in robotics, vision and graphics due to their modelling simplicity. On the other hand, non-central projection systems, in which there isn't a single viewpoint, are much more versatile, allowing far more general configurations and being more adequate to interactive situations (e.g. zooming, changing camera or mirror position and orientation, etc.).

Hence, the geometry of the direct computation of the reflection point from the 3D world to the image is non-linear and does not have any explicit closed-form expressions for the reflection point, in the general case of mirrors.[2,17]

Quadric mirrors are a special type of mirrors since they are analytically modelled by a non-ruled quadric which is a quadratic equation in the space coordinates—x, y and z. This type of mirrors

* Corresponding author. E-mail: nunogon@isr.uc.pt

comprises spherical, elliptical, hyperbolic (of two sheets) and parabolic mirrors. Although there are many types of reflection surfaces in the real world, quadric mirrors are often used in robotics, vision and graphics. Additionally, these types of mirrors are commonly used to locally approximate arbitrary complex surfaces. We emphasize that, when using quadric mirrors such as parabolic or hyperbolic mirrors, there is a special configuration of the camera COP to which the projection is central. To achieve this case, the camera must be located exactly on one of the foci of the quadric. However, any slight deviation from this particular configuration causes the projection to be non-central, which is the general case.

We are particularly interested in the direct computation of reflections on quadric surface mirrors with high accuracy and the highest performance. Two methods have been recently presented for this FPM, having been proposed by Goncalves[17] and Agrawal *et al*.[2] In the former case, the method was also compared with the direct computation of the projection, using the classical Reflection Law and the Fermat's Principle.[18] In this paper, the two methods for this computation are revised and compared in terms of their performance. Their execution cycle is analysed and their optimization is also addressed. The special case of spherical mirrors is analysed and new simplified expressions regarding projection are derived.

As pointed out by Agrawal *et al.*,[2] this is a classical problem known as Alhazen's problem, based on the works of Ptolemy from around 150 A.D. and Alhazen from around 1,000 A.D.[3,16]

The solution to this problem has important implications for computer vision and computer graphics. In the field of vision, almost all applications and methods that use direct projection or reprojection error, for instance, benefit from the existence of a fast and accurate direct projection model for non-central catadioptric vision systems, allowing some slow and impractical computations to become easy and quick. Applications that benefit from this include calibration (using for instance bundle adjustment techniques and others), pose estimation, 3D reconstruction and augmented reality. A framework for robot navigation has been recently presented by Dias *et al.*,[10] using the FPM to accelerate pose estimation and localization.

As for the field of computer graphics, direct projection can enhance the rendering of reflections. Reflections in graphics are still an important research and industry topic since their presence in images gives the observer a profoundly realistic sensation. Well-rendered reflections through mirrors or through general specular or semi-diffuse surfaces are, however, still difficult due to the intensive computation that is required to compute them, especially if there are moving objects in an interactive scene. As we show in this paper, rendering interactive reflections through quadric-shaped mirrors can benefit highly from a faster and accurate computation of the projection point.

Our contributions are thus twofold:

- The case of spherical mirrors is addressed and new expressions for the forward projection are derived for the method proposed by Goncalves.[17] We emphasize that this kind of mirrors is frequently used in robotics.
- The two methods for this FPM computation for generic quadric mirrors, proposed by Goncalves[17] and Goncalves and Nogueira,[18] and by Agrawal, Taguchi and Ramalingam[2] are revised and compared in terms of their performance. Although one would expect that a direct expression for determining the reflection point where the light ray is reflected from the source to the camera (polynomial approach) would perform faster, evidence in experiments shows that these expressions are outperformed in terms of computation time by a method based on implicit equations (the root-finding approach).

This paper is organized as follows: in the next section, we present a literature review of the problem; in Section 3, a detailed revision of the two methods under comparison in this study is presented; and in Section 4, the problem is specifically analysed with regard to spheres. In the following section, a study on the performance of both methods is analysed, and in Section 6, we present a discussion on the experiments. Final conclusions and future work are then presented.

2. Literature Review

Reflections through mirrors or any other specular surface are a physical phenomenon that is explained by the well-known Reflection Law, stating that reflection is a planar process. This law also states that incident and reflected angles with a normal direction are equal in magnitude.[7] The reflection

phenomenon can also be explained and physically described by a more elegant law, called the Fermat's Principle, stating that light travels along the quickest path which, in a macroscopic real world, is also the shortest path.[19] When the direction of incidence or reflection is known, the reflection point is computed by an inverse problem, simply solving its intersection with the mirror surface and computing the other direction using the Reflection Law.

However, for the direct problem, when only the source and capture (eye or camera) points are known, computing the reflection point where the light from the source is reflected to pass through the capture can only be solved, in the general case, by inverting the Reflection Law or the Fermat's Principle (see, for instance, Fig. 1). If the former method is used, a set of non-linear equations in at least two parameters must be solved, using an iterative search algorithm that often uses the difference between incident and reflected angles as the cost function of the minimization process. Alternatively, if the latter method, the Fermat's Principle, is used, the total sum of the path travelled by light is minimized in order to find the reflection point on the surface, also parameterized as a function of at least two parameters.[9]

The calibration of catadioptric cameras is a difficult topic since equations are non-linear and error prone. A widely known toolbox for central omnidirectional cameras is presented by Scaramuzza *et al*.[31] Lately, some authors have been adopting a two-step projection model by introducing a stereoscopic projection into a unitary sphere.[4,32,36] These projection models are suited for central catadioptric cameras with a single viewpoint. Additionally, some calibration methods rely on planar restrictions to improve the calibration results.[15,24]

2.1. The forward projection model—FPM

Goncalves[17] has presented an analytical method for reducing the search space of the parameterization of the reflection point to a single parameter. This analytical method is based on the intersection of two quadric surfaces in space, one of them being the mirror surface, expressed by a quadric mirror, and the other one being an analytical surface, derived from the absolute dual quadric. The reflection point is proved to belong to the intersection of these two quadrics and, since their intersection is a quartic in space, its parameterization is reduced to a single parameter curve. Goncalves and Nogueira[18] have then compared this method with the classical Reflection Law and the Fermat's Principle, showing that their method presents one order of magnitude speed-up in relation to these two-spaced parameterizations. This work has the advantage of parameterizing the search space into a single variable, for quadratic reflectors, and it is one of the methods we use to make a comparison in terms of performance on the computation of reflection points.

As for the special case of spherical mirrors, the problem has been extensively studied and some analytical solutions have been proposed. A model of general linear cameras was used by Ding *et al.*[11] to approximate the solution. Agrawal, Taguchi and Ramalingam[1] have presented forward projection equations for axial configurations which also comprise spherical mirrors.

Recently, the same authors have generalized for the same parameterization for a general quadric reflector and have presented the solution to the computation of the reflection point as a problem of solving an eight-degree polynomial whose coefficient expressions are computed in quasi-closed form equations.[2] The solution is non-iterative, given the roots of an eight-degree polynomial, and its accuracy depends on the accuracy achieved by this polynomial step (which is intrinsically iterative). They compare their solution with the methods presented by Lhuillier[22] and Micusik and Pajdla,[25] who use the Law of Reflection as a criterion for minimization. Results show that the polynomial method presented by Agrawal *et al.*[2] performs better in one or two orders of magnitude. We also use this method to make a comparison in terms of performance.

2.2. Applications of the FPM

The solution to the FPM through quadric-shaped mirrors has several applications besides computer vision.[10] As recognized by Agrawal *et al.*,[2] they are mainly useful to rendering for computer graphics,[9] image-base relighting,[34] environment matting and shape from specular flow.[30]

In the field of computer graphics, reflections have been studied in the context of rendering images of objects that reflect the environment around them. As previously explained, ray-tracing is the rendering technique that provides the most perfect images of reflections since it computes the light path backwards, that is, from the pixel to a non-specular or diffuse reflecting surface.[35] Though truly realistic, images rendered by ray-tracing are extremely computationally intensive, seldom making

ray-tracing appropriate for interactive reflections in scenes where objects and/or camera move freely. These types of images are required in many types of applications such as games, movies, 3D reconstruction and real-time pose estimation. Notice that, in interactive scenes, when any of the objects or the observer move, the whole reflected scene must be recomputed since the underlying geometry of reflections changes completely. This characteristic prevents the use of ray-tracing in interactive reflections, even with highly accelerated hardware.

An alternative to ray-tracing, vertex-based techniques are commonly used for rendering reflections, providing less attractive but real-time images. In the context of vertex-based techniques, environment mappings have been extensively used, as proposed by Blinn and Newell,[6] considering that all objects are at an infinite distance from the reflector. Environment mappings suffer from severe parallax problems which are particularly visible in objects that are near the reflectors. The artefacts appearing in images perturb the realistic sensation of the observer. Several improvements to this technique have been proposed since then, the most important ones being those proposed by Martin and Popescu,[23] Yu, Yang and McMillan,[37] and Bjorke.[5]

Other techniques used in computing reflections include the work of Mitchell and Hanrahan,[26] using caustics computed from the surface equations, and the work of Ofek and Rappoport,[27] computing another data structure which they have called explosion map for accelerating reflections. In this technique, the primary image, not considering any reflections, is blended with the reflection image that contains the reflected vertices. The reflections are calculated by tessellating the reflector surface and searching all the triangles for the appropriate reflection point. Chen and Arvo[8,9] have used ray-tracing to solve reflections for a small part of the vertices, having then applied a perturbation to compute the reflection of the neighbouring ones.

Alternatively, the work of Estalella *et al.*[14] comprises general curved mirrors. In a first version, the reflection point is computed iteratively using the Reflection Law, using two-spaced parameters on the surface of the curved reflector. This method yields good and accurate results for reflections, but it is unfortunately slow. A second version of this technique is implemented in the GPU which accelerates the computation of the reflection points.[13] Concurrently, Roger and Holzschuch[29] have used the Fermat's Principle to minimize the optical path length in order to solve the reflection point. The search space is parameterized through triangles expressed as a function of two angular parameters. This process is interrupted as soon as the desired accuracy is achieved. This method is also implemented in the GPU and provides accurate reflections.

3. Review of Two Methods For Accurately Computing the Reflection Point

In this section, we briefly review both methods for computing the reflection point on a quadric surface reflector when reflecting a light travelling from a known source to a known destination, presented in refs. [17,18 and 2]. We call the former method Quadric Intersection Method (the QI Method) and the latter Mitsubishi Method, since it has been developed in the Mitsubishi Labs. Before presenting both methods, we formally describe the problem to be solved.

3.1. Problem statement

Suppose a quadric surface reflector defined by the following quadratic equation:

$$x^2 + y^2 + Az^2 + Bz - C = 0, \tag{1}$$

where the coefficients A, B and C are arbitrary scalars. This parameterization of the quadric mirror comprises rotationally symmetric mirrors such as spherical ($A = 1$ and $C + B^2/4 > 0$), parabolic ($A = 0$ and $C = 0$), hyperbolic ($A < 0$ and $C < 0$) and elliptic ($A > 0$ and $C > 0$).[33] The quadric mirror can also be expressed by a quadric matrix Q, in homogeneous coordinates, given by

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & A & B/2 \\ 0 & 0 & B/2 & -C \end{bmatrix}, \tag{2}$$

where the point $\mathbf{x} = [x \ y \ z \ 1]^T$ belongs to quadric Q if and only if it respects the equation $\mathbf{x}^T Q \mathbf{x} = \mathbf{0}$.

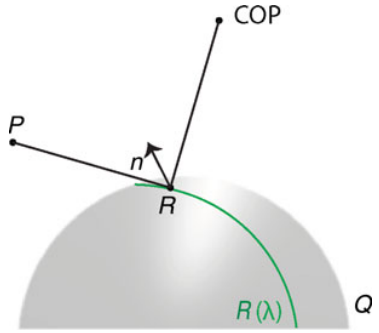

Fig. 1. Reflection through a quadric reflector where the reflection point is searched in a parameterized quartic curve $R(\lambda)$.

The camera COP is considered to be placed at the point $\mathbf{COP} = [c_x\ c_y\ c_z\ 1]^T$ and the 3D point to be projected is defined as $\mathbf{P} = [X\ Y\ Z\ 1]^T$.

As illustrated in Fig. 1, the incident ray intersects the reflector surface at the point **R**, where the light ray is projected to the camera along the reflected direction.

The problem addressed in this paper concerns the performance evaluation of different methods for determining the reflection point **R**.

3.2. Review of the QI method

The QI Method was first presented by Goncalves[17] and Goncalves and Nogueira,[18] and its name derives from the fact that an additional constraint on the reflection point was imposed, allowing a much faster way to search for the actual reflection point where light is projected from the source to the destination (camera). This constraint imposes that the reflection point belongs not only to the reflector surface, but also to an analytical quadric whose expression depends exclusively on the geometry of the projection (centre of projection and 3D point to be projected). Since the searched reflection point belongs to these two quadrics, it shall be searched in their intersection (see Fig. 1).

As proven in ref.[17], the reflection point **R** belongs to the quadric reflector Q and also to the analytical quadric S whose expression is given by

$$S = M^T Q_\infty^* Q + Q^T Q_\infty^* M, \tag{3}$$

where the matrix Q_∞^* is the absolute dual quadric given by

$$Q_\infty^* = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

and M is a skew-symmetric matrix that depends on the centre of projection of the camera and the 3D point to be projected, expressed by

$$M = \begin{bmatrix} 0 & c_z - Z & -c_y + Y & c_y Z - c_z Y \\ -c_z + Z & 0 & c_x - X & -c_x Z + c_z X \\ c_y - Y & -c_x + X & 0 & c_x Y - c_y X \\ -c_y Z + c_z Y & c_x Z - c_z X & -c_x Y + c_y X & 0 \end{bmatrix}. \tag{5}$$

Given these two restrictions imposed on the reflection point, the intersection of two quadrics can be computed using the Levin's Method[20,21] or its most recent adaptation, presented by Dupont *et al.*,[12] that is optimal in terms of the number of irrationals used in the obtained parameterization (notice that irrational numbers are computationally intense).

The parametric curve given by the intersection algorithm is a function of only one parameter, say λ. Let us represent the parameterized curve by the 4×1 vector $\mathbf{R}(\lambda)$. Though non-linear, the curve

can be searched for the point where the total distance travelled by light is minimum, as stated by the Fermat's Principle. Alternatively, the Law of Reflection can be used to search for the point where incident and reflected angles are equal. However, ref.[18] shows that the best criterion in terms of performance is the Fermat's Principle. It is also proved that the solution is unique for a non-ruled quadric reflector (comprising spheres, hyperboloids, paraboloids and ellipsoids).

For the general case, the obtained parameterization involves the solution of a polynomial up to an eighth degree, in which the solution is found in the intervals where the polynomial is positive. Although the formulation here presented considers the coordinate system as being centred on the reflector, this method can be computed using an arbitrary reference system, regarding that all geometric entities are expressed using the same coordinate system.

The main advantage of this method is that, although the solution is searched iteratively, it is searched in a one-parameter 3D curve in space (called quartic), inversely to other previously presented methods that search the solution in a two-parameter space or higher, making the solution for the QI Method much easier and faster to obtain. Concerning the accuracy of the solution, since the cost function used for the non-linear iterative minimization is metric and the search space is convex due to the geometric construction of the reflector, the accuracy is as high as necessary, making it possible to obtain the solution up to a double precision or higher. In Section 5, a detailed algorithmic approach of the method is studied and presented.

3.3. Review of the Mitsubishi method

The Mitsubishi Method presented by Agrawal, Taguchi and Ramalingam[2] derives explicit expressions for the coefficients of an eighth-degree polynomial whose solutions contain the reflection point **R**. It is based on two intermediate equations (IE) whose intersection gives a parameterization of the solution in a polynomial.

Consider the quadric reflector equation given by expression 1 in Euclidian coordinates. Without loss of generality, a pre-rotation around the z-axis is needed in order to align the centre of projection with the y-axis, assuming that all geometric entities are expressed in the reflector coordinate system and that the reflector is rotationally symmetric along the z-axis.

After this pre-rotation, the reflector equation remains unchanged; the centre of projection is now represented in the new coordinate system by $\mathbf{COP_{rot}} = [0\ d_y\ d_z]^T$ and the 3D point to be projected is now represented using the rotated coordinates as $\mathbf{P_{rot}} = [u\ v\ w]^T$. This step is needed to reduce the complexity of the subsequent projection equations and it is undone at the end of the computation of the reflection point. The reflection point on the reflector surface is expressed in Euclidean coordinates by $\mathbf{R_{rot}} = [x\ y\ z]^T$.

The method then evolves by introducing two IE. The first equation is obtained by defining the reflection plane as the plane that passes through the centre of projection $\mathbf{COP_{rot}}$, the 3D point $\mathbf{P_{rot}}$ and the point **K**, which is the intersection point of the reflector axis and the normal vector to the reflector surface at the reflection point $\mathbf{R_{rot}}$ (see Fig. 2 to visualize the method). The plane equation is derived and it is noticed that it can be represented as a linear function of x and y, allowing to derive x as a function of y and z.

This equation for x can then be substituted in the mirror Eq. (1), yielding the first Intermediate Equation—$\mathbf{IE_1}$:

$$\mathbf{IE_1} : (c_1^2(z) + c_2^2(z))y^2 + 2c_2(z)c_3(z)y + c_3^2(z) + c_1^2(z)(Az^2 + Bz - C) = 0, \tag{6}$$

where

$$\begin{cases} c_1(z) = (B + 2Az)(d_y - v) + 2d_y(w - z) + 2v(z - d_z) \\ c_2(z) = u(B + 2d_z - 2z + 2Az) \\ c_3(z) = ud_y(B + 2Az) \end{cases}. \tag{7}$$

The intermediate equation $\mathbf{IE_1}$ is quadratic in y with the coefficients as functions of z. The geometrical interpretation of this equation is that it represents the intersection of the reflection plane with the reflector surface whose curve can be searched for the reflection point, specifically where incident and reflection angles are equal.

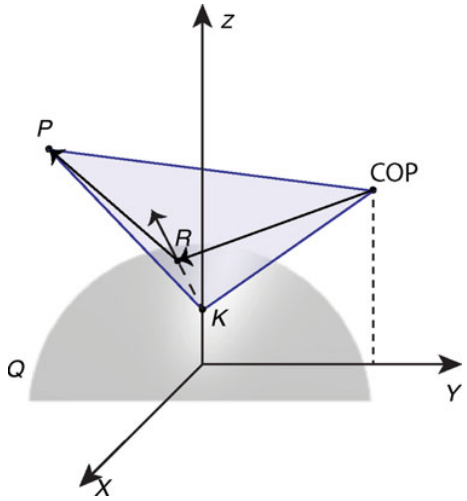

Fig. 2. Reflection through a quadric reflector and geometric construction for the Mitsubishi Method.

Notice here that both methods under revision in this section, the QI Method and the Mitsubishi Method, use intersections between the reflector and a quadric and a plane, respectively, to represent a curve in the space where the reflection point is to be found.

The second intermediate equation ($\mathbf{IE_2}$) is obtained by forcing the reflected vector to respect the Reflection Law such as

$$\mathbf{v_r} = \mathbf{v_i} - 2\mathbf{n}(\mathbf{v_i}^T\mathbf{n})/(\mathbf{n}^T\mathbf{n}). \tag{8}$$

Equation $\mathbf{IE_2}$ is then given by

$$k_{31}(z)y^2 + k_{32}(z)y + k_{33}(z) = 0, \tag{9}$$

where the expressions for $k_{ij}(z)$ are very long. Please see ref.[2] for details.

Since both $\mathbf{IE_1}$ and $\mathbf{IE_2}$ are quadratic in y, we can eliminate the dependence on y^2 and substitute y back into either $\mathbf{IE_1}$ or $\mathbf{IE_2}$, obtaining

$$\begin{aligned} k_{41}(z)(k_{43}(z)k_{32}^2(z) &- k_{42}(z)k_{32}(z)k_{33}(z) + k_{41}(z)k_{33}^2(z)) \\ &- k_{31}(z)(-k_{33}(z)k_{42}^2(z) + k_{43}(z)k_{32}(z)k_{42}(z) \\ &+ 2k_{41}(z)k_{43}(z)k_{33}(z)) + k_{43}^2(z)k_{31}^2(z) = 0. \end{aligned} \tag{10}$$

Equation (11) can be simplified so as to become an eighth-degree polynomial whose coefficients depend on the known mirror parameters (A, B and C), the known location of the centre of projection (d_y and d_z) and the known 3D point (u, v and w).

The solution to the z coordinate of the reflection point $\mathbf{R_{rot}}$ is then one of the eight solutions to the polynomial (actually, only real solutions are considered), from which all candidate solutions (up to 16, since each polynomial solution produces two candidate reflection points) are computed for the reflection point. It is important to point out that an inverse rotation is needed in order to reposition the solution at the original coordinate system. All candidate solutions are then tested using the Law of Reflection and the solution that best fits the Law (incident and reflection angles are equal) is considered the actual solution.

This method has the advantage of providing explicit quasi-closed form expressions for the coefficients of an eighth-degree polynomial that, when solved, produces a small number of candidates for the reflection point. This small set of candidates is easily searched for to obtain the final solution. This method is hence non-iterative if one considers the polynomial solved. However, strictly speaking, the solution of an eighth-degree polynomial is an iterative process.

In Section 5, we compare both the QI and the Mitsubishi Methods in terms of performance to obtain an accurate reflection through quadric mirrors.

4. Spherical Mirrors

Spheres are the most frequently used geometric shapes for mirrors, after the planar ones. They are physically easy to build, beautiful and computationally practical for robotics and rendering. In this section, we analyse the evolution of the QI Method for these particular specular surfaces, presenting new expressions for the computation of the reflection point.

Let us consider, without loss of generality, that the world reference system is centred on the spherical mirror. Notice that a simple translation can be applied to all geometric entities in order to guarantee this condition, since orientation is not needed for a sphere. The sphere is then given by Eq. (1), where $A = 1$, $B = 0$ and C is the radius to the power of two.

Since the normal vector to any surface point in a sphere is radial and, as stated by the Reflection Law, reflection is a planar phenomenon, it can be concluded that the reflection plane must contain the mirror origin, as well as the 3D point to be projected and the camera COP. It turns out, then, that for the QI Method, the quadric S degenerates to this plane, whose equation can be directly derived from Eq. (3), yielding the plane equation as

$$m_{14}x + m_{24}y + m_{34}z = 0, \tag{11}$$

where m_{ij} is the corresponding element of matrix M, of Eq. (5).

Since the intersection of a sphere with a plane passing through its origin is a circumference in space, its equation can be written in the matrix form as

$$s_1x^2 + s_2y^2 + 2s_3xy - C = 0 \Leftrightarrow \mathbf{x}^T \begin{bmatrix} s_1 & s_3 \\ s_3 & s_2 \end{bmatrix} \mathbf{x} = C, \tag{12}$$

where $\mathbf{x} = [x \ y]^T$ and

$$\begin{cases} s_1 = \dfrac{m_{14}^2}{m_{34}^2} + 1 \\ s_2 = \dfrac{m_{24}^2}{m_{34}^2} + 1 \\ s_3 = \dfrac{m_{14}m_{24}}{m_{34}^2} \end{cases} \tag{13}$$

which parameterize the curve where the QI Method searches for the FPM solution.

In order to speed up the search process and transform it into a single-parameter problem, after diagonalizing the matrix, one obtains the final equation of the intersection of the quadric mirror Q and the quadric S, given as

$$\lambda_1 x'^2 + \lambda_2 y'^2 = C \tag{14}$$

which is an ellipse centred on the origin, where

$$\begin{cases} \lambda_1 = \dfrac{s_1 + s_2 - \sqrt{\Delta}}{2} \\ \lambda_2 = \dfrac{s_1 + s_2 + \sqrt{\Delta}}{2} \\ \Delta = s_1^2 + s_2^2 - 2s_1s_2 + 4s_3^2 \end{cases} \tag{15}$$

or, by substituting it for Eq. (13), it yields

$$\begin{cases} \lambda_1 = 1 \\ \lambda_2 = 1 + \dfrac{m_{14}^2 + m_{24}^2}{m_{34}^2} \end{cases} \tag{16}$$

and the transformation equations that relate original 3D coordinates (x and y) and 2D search space coordinates (x' and y') are

$$\begin{cases} x = -\dfrac{m_{24}}{m_{14}} x' + y' \\ y = \dfrac{m_{14}}{m_{24}} x' + y' \end{cases} . \tag{17}$$

Finally, regarding the search for the reflection point, as stated by the QI Method, it is still necessary to parameterize the ellipse in a single variable. The following parameterization is the standard one:

$$\begin{cases} x' = \sqrt{\frac{C}{\lambda_1}} cos\theta \\ y' = \sqrt{\frac{C}{\lambda_2}} sin\theta \end{cases} \tag{18}$$

with $\theta \in]-\pi, \pi]$.

Hence, when applied to spheres, the QI Method parameterizes the solution in much simpler expressions that degenerate to an optimization process in an ellipse curve. For the Mitsubishi Method, as presented by Agrawal *et al.* in ref.[1], the solution degenerates in a fourth degree polynomial, for which there is a closed-form solution. However, as we shall see, its computation cycle is much longer due to extremely difficult and slow algebraic equations.

5. Performance Evaluation and Comparison

The main objective of this paper is to study and compare the two methods that were presented in the previous section in terms of their performance in the computation of the reflection point.

We designed a comprehensive set of experiments to fully test both methods in several conditions and measured the execution time, excluding overhead phases so that the evaluated performance refers to the exact computation of the reflection point. For all tests, three configurations with different mirrors were tested, comprising a spherical, a hyperbolic and a parabolic mirror. The position of the camera and the 3D point are randomly generated at the beginning of each iteration, in which we compute the execution time for both methods (hence with the same point). The implementation code was entirely written in C++ language and the tests were performed in a desktop computer equipped with a 2.4 GHz Intel Core i7, running Linux. We also used the implementations of Numerical Recipes in C++.[28]

For the implementation of the Mitsubishi Method,[2] we adapted the code provided by the authors (originally programmed in Matlab) to a faster C++ implementation. As with the QI Method, we used Numerical Recipes in C++[28] for computing the roots of an eighth-degree polynomial.

The experimental results showed that both methods achieve a solution that is correct up to the floating point precision (single or double), so both methods are equal in terms of accuracy, which depends mainly on the tolerance of the minimization algorithm used to stop the search for the solution. Since the nature of the parameters to be searched is different in both methods, we opted to use standard values.

Before detailing the tests that were designed and performed, we analysed the execution cycle of both methods in order to seek for possible optimizations and improvements.

For the QI Method,[17,18] we present the algorithm described in Table I.

The Mitsubishi Method[2] was then subdivided in the algorithm described in Table II.

Table I. Algorithm for the QI Method.

Algorithm for the QI Method		
Time flag	Phase #	Description
1		Begin with quadric Q, 3D point P and camera centre of projection COP.
	1	Parameterization of intersection curve and calculation of solution intervals.
2		Parameterization and intervals are obtained.
	2	Bracketing the solution.
3		Solution is bracketed.
	3	Minimization of cost function using the Golden Section Method.
4		Minimization performed, producing a set of candidates.
	4	Solution selection using the Fermat's Principle.
5		Solution found.

Table II. Algorithm for the Mitsubishi Method.

Algorithm for the Mitsubishi Method		
Time flag	Phase #	Description
1		Begin with quadric Q, 3D point P and camera centre of projection COP.
	1	Pre-rotation.
2		Quadric, 3D point and Camera in rotated coordinates.
	2	Computation of the polynomial coefficients (nine coefficients).
3		Polynomial defined.
	3	Computation of candidate solutions (up to 16).
4		All candidate solutions computed.
	4	Solution selection using the Law of Reflection.
5		Solution found.

As our objective is to assess both methods in terms of performance, and since we also want to characterize them in terms of stability, temporal and spatial locality phenomena, and convergence rate, we have designed the following four tests.

Test ONE
This test is to compare both methods in terms of their total execution time. We tuned both methods for their best performance and ran them for 100,000 points. The 3D points to be projected and the camera COP were generated randomly in a wide area around the mirror surface. The computed statistics for the test (and each point) are mean, standard deviation and total execution time. Test ONE details are summarized in Table III.

Figure 3 shows the results, expressed in microseconds, for all three types of mirrors. The results are also summarized in Table IV. We can observe that the QI Method is faster than the Mitsubishi one and is able to compute the forward projection in around half its execution time for all mirrors. We can also observe from Table IV that the execution time for the QI Method is much more stable than the Mitsubishi Method, for parabolic mirrors, since the standard deviation of the execution time is much lower in the former case. This characteristic is very important when considering all implementation details and deciding whether to use one method or the other.

Table III. Test ONE details for the performance evaluation of the FPM computation.

Test ONE	
Parameter	Value
Number of points	100,000
Points distribution	Random
	Wide area
Minimization threshold	1e-3
Mirror type	Sphere
	Parabolic
	Hyperbolic
Inputs	3D points
	COP
Output	Execution time
	Number of wrong points

Table IV. Test ONE—Mean value and standard deviation for the total execution time for the computation of the FPM. Comparison between the QI and the Mitsubishi Methods. All times are expressed in microseconds.

	Test ONE results			
	QI		Mitsubishi	
	Mean	Std.	Mean	Std.
Parabolic	167.9	10.83	333.4	70.33
Hyperbolic	167.2	10.59	291.6	8.67
Spherical	57.1	7.43	125.9	6.30

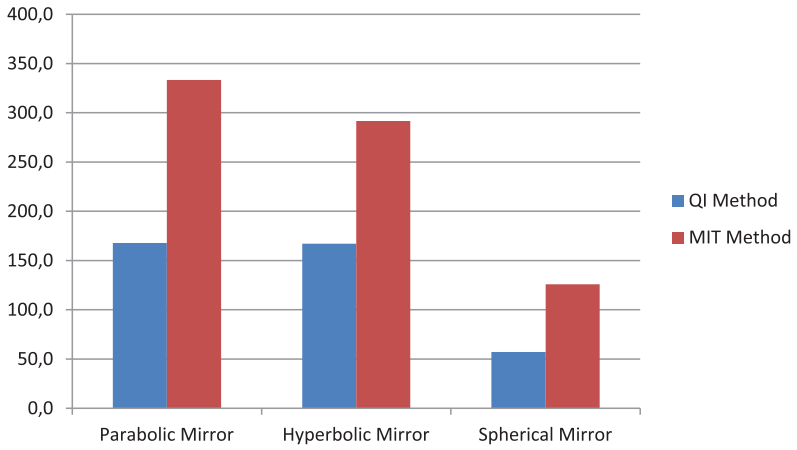

Fig. 3. Test ONE - Mean execution time for the computation of the reflection point using the QI and the Mitsubishi Methods. All times are expressed in microseconds.

Test TWO

As for Test TWO, our aim is to analyse the execution cycle of each method. We compute the execution time for each phase of each method in the same conditions as in the previous test. Both the QI and the Mitsubishi Methods are compared in terms of their cycles. All other conditions and parameters remain unchanged from test one. Table V summarizes the test details.

The results are shown in Table VI. One can observe that, for hyperbolic and parabolic mirrors, the phases that consume more elapsed time in the QI Method are phases #1 and #3, for parameterization and minimization, respectively. As for the spherical mirror, there is no parameterization phase due to

Table V. Test TWO—Test TWO details for the performance evaluation of the FPM computation, analysing in detail the execution cycle of both methods.

Test TWO	
Parameter	Value
Number of points	100,000
Points distribution	Random Wide area
Minimization threshold	1e-3
Mirror type	Sphere Parabolic Hyperbolic
Inputs	3D points COP
Output	Execution time for: - All phases of the QI Method - All phases of the MIT Method

Table VI. Test TWO—Execution time for all phases of the QI and the MIT Methods. All times are expressed in microseconds and represent the average execution time for a single point.

		Test TWO results			
		Phase 1	Phase 2	Phase 3	Phase 4
QI	Par.	138.63	1.08	29.06	2.75
	Hyp.	134.34	1.08	31.62	3.10
	Sph.	0	11.20	38.23	4.37
MIT	Par.	1.46	247.66	89.65	4.68
	Hyp.	1.46	246.66	46.85	5.00
	Sph.	0	1.27	116.32	13.34

the explicit new expressions for the two quadrics intersection derived in this paper; hence, the phase which consumes more elapsed time is minimization. On the other hand, for the Mitsubishi Method, phase #2, where the polynomial coefficients are computed, represents around two-thirds of the total execution time, followed by the computation of the 16 candidate solutions.

Test THREE

The third set of performance tests aims to analyse the effects of spatial locality phenomena. The main idea of these tests is to test if calculating the reflection point for two 3D points that are very close to each other can, by any means, benefit from the fact that the geometric situation is similar. We notice here that, for the Mitsubishi Method, we do not expect such phenomena since all polynomial coefficients must be recomputed at every point. On the other hand, as in the case of the QI Method, we expect to measure significant differences at the minimization step of the algorithm, since the starting point (initial parameter) shall have an impact on the number of performed iterations. We then performed these tests only for the QI Method, repeating them for three different situations:

- **Wide area random point** - The 3D points to be reflected through the mirror are generated in a very wide area, so that the probability of having two consecutive points, very close to each other, is very small.
- **Low area random point** - The 3D points to be reflected through the mirror are generated in a small area, so that the probability of having two consecutive points, very close to each other, is high.

Table VII. Test THREE—Test THREE details for the performance evaluation of the FPM computation, analysing spatial locality phenomena.

Test THREE	
Parameter	Value
Number of points	100,000
Points distribution	Wide area random
	Low area random
	Linear
Min. threshold	1e-3
Mirror type	Sphere
	Parabolic
	Hyperbolic
Min. initial guess	Centre of the interval (standard)
	With the previous best value
Inputs	3D points
	COP
Output	Execution time for:
	the QI Method

- **Linear random point** - The 3D points to be reflected through the mirror are sampled in a line in space, so that the probability of having two consecutive points, very close to each other, is high and in regular steps.

We also performed these tests using two different strategies for the initial parameter of the minimization step. Notice that, in the QI Method, the search is performed in the intervals where the characteristic polynomial is positive, which means that one has to search up to five intervals (although, in most cases, the solution is found in the first or second interval). Minimization is then set using the two following strategies:

- **Centre of the interval** - The parameter λ to initialize the minimization step is set to the middle of the interval.
- **Best previous value** - The parameter λ to initialize the minimization step is set to the best value from the previous reflection point. If the previous best value does not fall into any valid interval, initialization is made using the standard strategy (using the middle of the interval).

Table VII summarizes the test details.

The results obtained from test THREE are presented in Table VIII. One can observe that, for all mirrors, there is a significant locality phenomenon; however, the achieved improvement is very small. Except for the case of spherical mirrors, the mean improvement is around 2% of the execution time. In the case of spherical mirrors, however, the improvement is near 30%.

Test FOUR

As for test FOUR, our aim is to analyse the halting condition of the QI Method. Since its main step is a minimization process, the tolerance used in the Golden Section Method of the Numerical Recipes[28] is very important since, without an appropriate threshold, the method could be iterating in useless steps. Since all points are tested to guarantee that they respect the Law of Reflection, we can evaluate the threshold goodness by counting the number of wrong points. This test is also meaningless for the Mitsubishi Method; hence, we only performed it for the QI one. We tested several threshold values in a logarithmic scale. Table IX summarizes the details of these tests.

The results are shown in Fig. 4. One can observe that the average execution time for the computation of the FPM decreases linearly with the threshold attributed to the minimization method. We also observe that, for spherical mirrors, the execution time stabilizes for tolerances higher than 1e-5. This influence is also observed in the number of iterations, which was expected since the execution time is highly dependent on the number of iterations. Additionally, we noticed that, for all the tests that were run, the parabolic mirror was the only one where we observed wrong computations of the FPM

Table VIII. Test THREE—Total Execution time for test THREE. All times are expressed in microseconds and represent the average execution time for a single point.

	Test THREE results		
	Wide area	Low area	Linear
Parabolic			
Centre of interval	175.8	181.9	179.1
Previous best value	170.4	174.9	186.5
Hyperbolic			
Centre of interval	175.4	177.4	173.1
Previous best value	171.7	174.1	171.1
Spherical			
Centre of interval	50.8	54.8	54.5
Previous best value	40.7	41.8	31.1

Table IX. Test FOUR details for the performance evaluation of the FPM computation, analysing the influence of the halting condition (threshold) on the performance and accuracy of the QI Method.

Test FOUR	
Parameter	Value
Number of points	100,000
Points distribution	Random
	Wide area
Minimization threshold	1e-1
	1e-2
	1e-3
	1e-4
	1e-5
	1e-6
	1e-7
	1e-8
	1e-9
	1e-10
Mirror type	Sphere
	Parabolic
	Hyperbolic
Inputs	3D points
	COP
Output	Execution time
	Number of wrong points
	Average iterations per point

for thresholds above 1e-3. Table X summarizes the number of wrong reflection points computed for all cases.

6. Discussion

After presenting the results on the performance of the QI and the Mitsubishi Methods, we now discuss and interpret the previously presented results. The main conclusions we draw are:

- As can be easily observed, the QI Method always performs better than the Mitsubishi Method, being generally two times faster. This conclusion is verified for all types of mirrors, including the

Table X. Test FOUR—Number of wrong computations of the reflection point as a function of the threshold (maximum tolerance) of the minimization method. Total computed points are 100,000.

	Number of wrong points		
	Parabolic	Hyperbolic	Spherical
1e-10	0	0	0
1e-9	0	0	0
1e-8	0	0	0
1e-7	0	0	0
1e-6	0	0	0
1e-5	0	0	0
1e-4	0	0	0
1e-3	0	0	0
1e-2	10	0	0
1e-1	1287	0	0

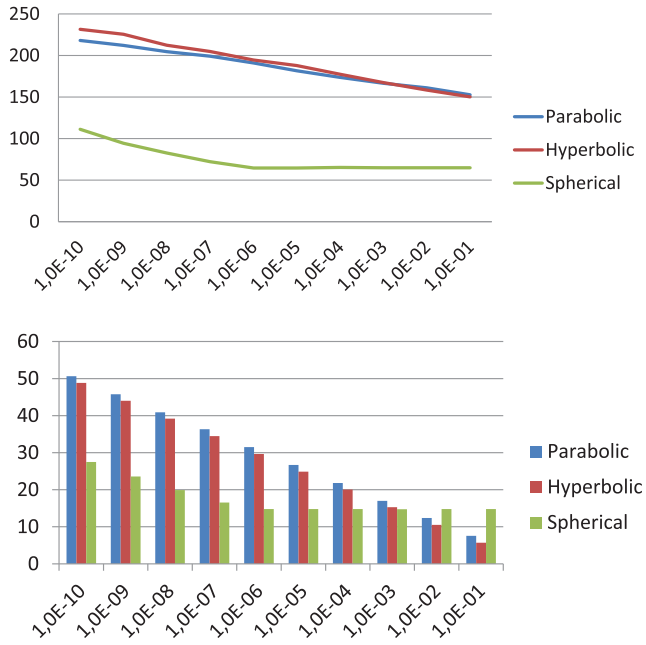

Fig. 4. Execution time (in microseconds) and number of iterations performed as a function of the maximum tolerance (threshold) attributed to the minimization method in the FPM computation. (a) Execution time. (b) Number of iterations.

spherical ones where the new expressions, derived in Section 4 for the QI Method projection, are used.

- The Mitsubishi Method is an algebraic method that provides explicit closed-form expressions for the coefficients of an eighth-degree polynomial whose roots are found (iteratively, since there is no closed-form solution to this problem), providing a means to compute 16 candidate solutions. For the special case of spherical mirrors, the polynomial is of degree 4. As can be observed from the results, although the solutions are easily found, this method is slower since the computation of the polynomial coefficients is extremely computationally intense. Notice that the expressions provided by the authors are very long and comprise the computation of around 50,000 power and product operations.
- Spatial locality phenomena are not generally observed for the Mitsubishi Method, since all eighth-degree polynomial coefficients must be recomputed for every 3D point to be reflected.

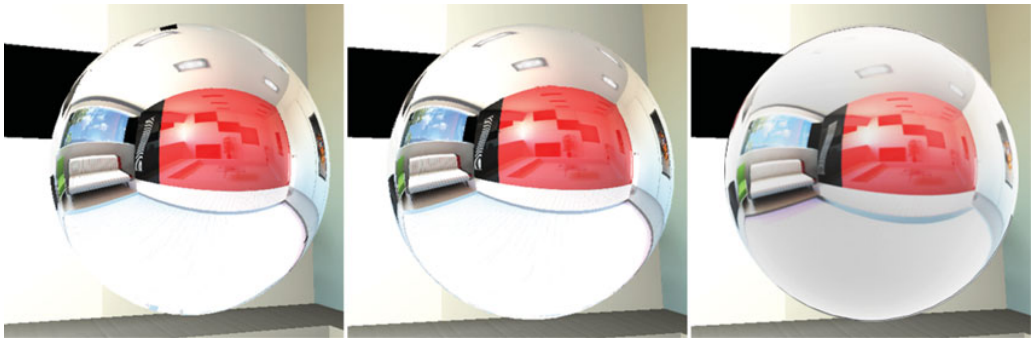

Fig. 5. Rendering of an image using the Mitsubishi Method (left), the QI Method (centre) and ray-tracing (right).

- For the QI Method, there is a slight locality phenomenon for parabolic and hyperbolic mirrors. However, when using spherical mirrors, this spatial effect is strong and able to speed-up the FPM computation by a factor of 30%. This spatial locality phenomenon can be explained by the fact that, when the best reflection point is searched in the proximity of the best solution to the previous point, the probability of finding the solution very closely is high.
- As for the optimizations of the Mitsubishi Method, as previously explained, this method is already highly optimized. Actually, its phase 2, which is relative to the computation of the polynomial coefficients, is responsible for around two thirds or more of the total execution time. Furthermore, as previously mentioned, since their calculation is unique for each 3D point and involves the computation of around 50,000 powers, multiplications and summations, the method is slow and non-optimizable.
- A parameter that has a great influence on the performance of the QI Method is the minimization tolerance (a parameter of the Golden Section Method, used for parabolic and hyperbolic mirrors, and of the Brent Method, used for spherical mirrors), which controls the amount of iterations performed for each reflection point. It was observed that reducing the tolerance until 1e-3 reduces the execution time linearly while preserving the accuracy of the computation (up to single or double floating point precision). The number of iterations has very little variability, which is a good characteristic for parallel computing techniques to be applied in the future. Additionally, for spherical mirrors, we observed that the execution time and number of iterations stabilize for tolerances higher than 1e-5.

7. Application to Computer Graphics

In what concerns real-time applications of the fastest method, the QI Method computes a reflection point in an average of 170 microseconds for parabolic and hyperbolic mirrors, and 50 microseconds for spherical mirrors, making it perfectly suitable for real-time applications, especially if one considers the fact that further accelerations are being studied and a GPU implementation is also intended for it.

To show the accuracy and performance of the reflection point computation, Fig. 5 (centre image) was rendered using C++, where we used some primitives of Numerical Recipes in C++ [28] for the exact reflection points. For the sake of comparison, we also rendered the same image using the Mitsubishi Method (leftmost image) and ray-tracing (rightmost image), which is the reference in terms of accuracy (all implemented in C++).

One can observe that the mirror (a sphere, in this example) correctly projects the points of the environment with no artefacts. As the purpose of our work is not the rendering of images itself, but rather the computation of the reflection point on the mirror surface, there are several improvements that can be made in order to get even better images, including illumination effects. With this example, we aim to show that the QI Method can be included in the pipeline for rendering realistic images. Furthermore, although the GPU is not used in the calculations of the reflection point, it must be pointed out that further accelerations may possibly be studied, if these computations were to be optimized for the GPU.

Table XI. Rendering time (in seconds) for the two methods under comparison and the ray-tracing reference, in C++ at the CPU.

Rendering time for the image		
Mitsubishi	QI (ours)	Ray-tracing
2.76	0.7	3.5

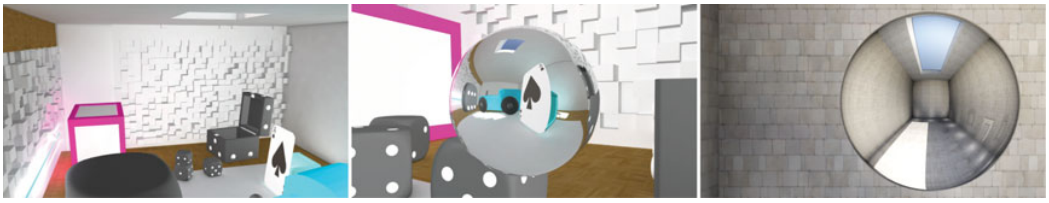

Fig. 6. The image on the left shows one of the created environments (a room with toys) and, in the centre, its reflection on the surface of the mirror. The image on the right represents an ancient environment. Both images were computed in C++ using the QI Method.

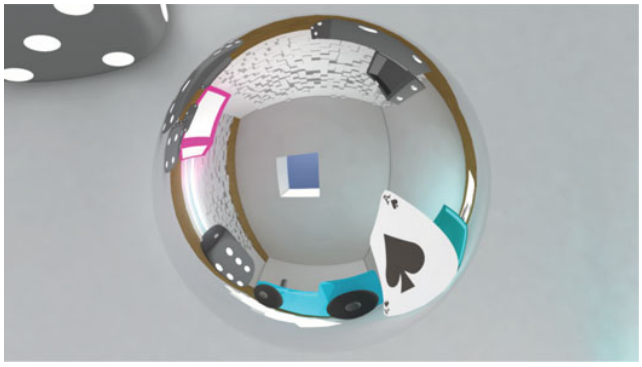

Fig. 7. Room filled with toys, rendered using the QI projection model, projecting 10,000 vertices.

As for performance, Table XI summarizes the computation time for the three methods, showing that the QI Method is the fastest one.

Additionally, we present the results of using the QI Method to render two different scenes: one representing a room filled with toys, the other representing an ancient environment (Fig. 6). As can be observed, the accuracy of the method for rendering scenes in graphics is very high, and we expect to achieve real time soon. A detailed render of the first scene is shown in Fig. 7.

In what concerns the C++ code of the implementations, it can be downloaded from the publicly available link: `http://scrat.isr.uc.pt/uniprojection/`.

8. Conclusion and Future Work

This paper addresses the problem of computing the reflection point on a surface reflector where a light ray is redirected from a 3D point to a camera centre of projection, called the FPM.

Recently, two different methods that address this problem have appeared. Goncalves and Nogueira have presented the QI Method that intersects the quadric mirror with an analytical quadric depending on the geometry, defining a curve where the reflection point is searched for and parameterized by a single parameter. The search in this curve, though iterative, presents higher performance. On the other hand, Agrawal, Taguchi and Ramalingam have presented a method that determines algebraically the solution as one of the roots of an eighth-degree polynomial whose coefficients are determined in closed-form expressions (notice that there is, however, no closed-form solution for polynomials of degree 8). This method, called the Mitsubishi Method, is simpler to solve and presents higher

accuracy in relation to the standard two-parameter space approach. In this paper, we also focus on the specific case of spherical mirrors for the QI Method, deriving simplified expressions for their projection model.

The experiments that were presented analyse the execution cycle of both the QI and the Mitsubishi Methods, and compare their performance. We concluded that the QI Method, though iterative, presents better performance and is able to find the reflection point with the same accuracy as the Mitsubishi Method. In turn, the Mitsubishi Method is slower since its polynomial coefficients must be recomputed for each 3D point and this step represents around two thirds of the execution time—with around 50,000 powers and multiplication operations being involved. Finally, the QI Method was found to be, in average, two times faster than the Mitsubishi Method. The QI Method was then used to render two images using C++, integrating it in the rendering pipeline for computing the reflection point on the mirror surface (in real time).

Future directions to pursue include further optimizing the computation of the FPM solution and comparing it with non-exact methods for computing the reflection point in quadric mirrors. We intend to test these methods in the field of graphics and compare them in the rendering of images with specular objects, represented by arbitrary surfaces that could be approximated by quadrics. Additionally, we intend to develop the QI Method in the GPU (in a vertex shader pipeline) in order to further accelerate it.

References

1. A. Agrawal, Y. Taguchi and S. Ramalingam, "Analytical Forward Projection for Axial Non-Central Dioptric and Catadioptric Cameras," *European Conference on Computer Vision (ECCV)*, Springer Berlin Heidelberg, Heraklion, Greece (Sep. 2010), pp. 129–143.
2. A. Agrawal, Y. Taguchi and S. Ramalingam, "Beyond Alhazen's Problem: Analytical Projection Model for Non-Central Catadioptric Cameras with Quadric Mirrors," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, US (2011), pp. 2993–3000.
3. M. Baker, "Alhazen's problem," *Am. J. Math.* **4**(1), 327–331 (1881).
4. J. Barreto and H. Araujo, "Geometric properties of central catadioptric line images and their application in calibration," *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(8) (2005) 1327–1333.
5. K. Bjorke, "Finite-Radius Sphere Environment Mapping," **In**: *GPU Gems – Programming Techniques, Tips, and Tricks for Real-Time Graphics* (Addison-Wesley, 2004) ISBN: 0-321-22832-4.
6. J. F. Blinn and M. E. Newell, "Texture and reflection in computer generated images," *Commun. ACM* **19**(10), 542–547 (1976).
7. M. Born and E. Wolf, *Principles of Optics* (Pergamon Press, Oxford, UK, 1965).
8. M. Chen and J. Arvo, "Perturbation methods for interactive specular reflections," *IEEE Trans. Vis. Comput. Graph.* **6**, 253–264 (2000).
9. M. Chen and J. Arvo, "Theory and application of specular path perturbation," *ACM Trans. Graph.* **19**(4), 246–278 (2000).
10. T. Dias, P. Miraldo, N. Goncalves and P. Lima, "Augmented Reality on Robot Navigation using Non-Central Catadioptric Cameras," *IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS*, Hamburg, Germany (Sep. 2015), pp. 4999–5004.
11. Y. Ding, J. Yu and P. F. Sturm, "Multiperspective Stereo Matching and Volumetric Reconstruction," **In:** *Proceedings of the International Conference on Computer Vision (ICCV)* Kyoto, Japan (Sep. 2009), pp. 1827–1834.
12. L. Dupont, D. Lazard, S. Lazard and S. Petitjean, "Near-optimal parameterization of the intersection of quadrics: I. The generic algorithm," *J. Symb. Comput.* **43**(3), 168–191 (2008).
13. P. Estalella, I. Martin, G. Drettakis and D. Tost, "A gpu-Driven Algorithm for Accurate Interactive Specular Reflections on Curved Objects," *Proceedings of the 2006 Eurographics Symposium on Rendering* (2006).
14. P. Estalella, I. Martin, G. Drettakis, D. Tost, O. Devillers and F. Cazals, "Accurate Interactive Specular Reflections on Curved Objects," *Vision Modeling and Visualization (VMV)* (2005) pp. 8.
15. S. Gasparini, P. Sturm and J. Barreto, "Plane-Based Calibration of Central Catadiotpric Cameras," *IEEE International Conference on Computer Vision* (2009), pp. 1195–1202.
16. G. Glaeser, "Reflections on spheres and cylinders of revolution," *J. Geom. Graph.* **3**(2), 121–139 (1999).
17. N. Goncalves, "On the reflection point where light reflects to a known destination in quadric surfaces," *Opt. Lett.* **35**(2), 100–102 (Jan. 2010).
18. N. Goncalves and A. C. Nogueira, "Projection through quadric mirrors made faster," *ICCVW: 9th Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras*, Kyoto, Japan (Oct. 2009), pp. 2141–2148.
19. E. Hecht, *Optics* (Addison-Wesley, Massachusetts, USA, 1987).
20. J. Levin, "A parametric algorithm for drawing pictures of solid objects composed of quadric surfaces," *Commun. ACM* **19**(10), 555–563 (1976).

21. J. Levin, "Mathematical models for determining the intersection of quadric surfaces," *Comput. Graph. Image Process.* **11**(1), 73–87 (1979).
22. M. Lhuillier, "Automatic scene structure and camera motion using a catadioptric system," *Comput. Vis. Image Underst.* **109**(2), 186–203 (2008).
23. A. Martin and V. Popescu, "Reflection Morphing," *ACM SIGGRAPH 2004 Sketches*, SIGGRAPH '04, ACM (Los Angeles, California, USA, 2004), pp. 152–156.
24. C. Mei and P. Rives, "Single View-Point Omnidirectional Camera Calibration from Planar Grids," **In:** *IEEE International Conference on Robotics and Automation*, Rome, Italy (Apr. 2007), pp. 3945–3950.
25. B. Micusik and T. Pajdla, "Autocalibration & 3d Reconstruction with Non-Central Catadioptric Cameras," *IEEE Conference on Computer Vision and Pattern Recognition* (Washington, DC, USA, 2004) pp. 58–65.
26. D. Mitchell and P. Hanrahan, "Illumination from curved reflectors," *SIGGRAPH Comput. Graph.* **26**(2) (1992) 283–291.
27. E. Ofek and A. Rappoport, "Interactive Reflections on Curved Objects," *SIGGRAPH '98*, New York, NY, USA, ACM (1998) pp. 333–342.
28. W. Press, S. Teukolsky, W. Vetterling and B. Flannery, *Numerical Recipes - The Art of Scientific Computing*, 3rd ed. (Cambridge University Press, Cambridge, UK, 2007).
29. D. Roger and N. Holzschuch, "Accurate specular reflections in real-time," *Comput. Graph. Forum* **25**(3), 293–302 (2006).
30. S. Roth and M. J. Black, "Specular Flow and the Recovery of Surface Structure," **In:** *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, New York, USA (Jun. 2006), pp. 1869–1876.
31. D. Scaramuzza, A. Martinelli and R. Siegwart, "A Toolbox for Easily Calibrating Omnidirectional Cameras," **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China (Oct. 2006), pp. 5695–5701.
32. P. Sturm and J. Barreto, "General Imaging Geometry for Central Catadioptric Cameras," **In:** *IEEE European Conference on Computer Vision*, Springer Berlin Heidelberg, Marseille, France (Oct. 2008), pp. 609–622.
33. R. Swaminathan, M. Grossberg and S. Nayar, "Non-single viewpoint catadioptric cameras: Geometry and analysis," *Int. J. Comput. Vis.* **66**(3), 211–229 (2006).
34. J. Unger, A. Wenger, T. Hawkins, A. Gardner and P. E. Debevec, "Capturing and Rendering with Incident Light Fields," *Proceedings of the 14th Eurographics Workshop on Rendering Techniques*, Leuven, Belgium (Jun. 25–27, 2003), pp. 141–149.
35. T. Whitted, "An improved illumination model for shaded display," *Commun. ACM* **23**(6), 343–349 (Jun. 1980).
36. X. Ying and Z. Hu, "Catadioptric camera calibration using geometric invariants," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(10), 1260–1271 (2004).
37. J. Yu, J. Yang and L. McMillan, "Real-Time Reflection Mapping with Parallax," *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games*, I3D '05, ACM (Washington, DC, USA, 2005) pp. 133–138.