Centrum voor Wiskunde en Informatica

Centre for Mathematics and Computer Science

I. Herman

On the projective invariant representation of conics in computer graphics

Computer Science/Department of Interactive Systems Report CS-R8938 October

1989

Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

I. Herman

On the projective invariant representation of conics in computer graphics

Computer Science/Department of Interactive Systems Report CS-R8938 October

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O.).

On The Projective Invariant Representation of Conics in Computer Graphics

Ivan Herman

Centre for Mathematics and Computer Science (CWI), Dept. of Interactive Systems
Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

ABSTRACT

A general formulation for conics and conic arcs for the purpose of computer graphics is given, based on principles and theorems of projective geometry. This approach allows the approximation of these curves by line segments to be postponed in the graphics output pipeline; it results in a more compact storage, faster approximation algorithms and smoother outlook of the curves.

1983 CR Categories: G.0, I.3.2, I.3.5
Keywords & Phrases: projective geometry in computer graphics, conics in design and computer graphics, projective theory of conics.
Note: the present text is to be published in: Computer Graphics Forum 8(1989), No. 4.

1. Introduction

Besides line segments and polygons, conics are also frequently used objects in computer graphics. They are used in geometric design, they are frequently implemented as GDP-s in various graphics standard implementations and some are included in the basic set of output primitives of ISO documents (e.g. CGI[10]). Among the three main classes of conics, namely ellipses, parabolae and hyperbolae, the use of ellipses (first of all circles) is the most widespread. Circles and circular arcs are used in business graphics for charts, in mechanical engineering for rounded corners, for holes etc. Circular arcs may also be used to interpolate curves (see for example Sabin[14]).

Although the role of parabolae and hyperbolae is not so important, we cannot ignore them either. On the one hand they do appear in practical applications (for example there are proposals to use parabolic arcs for curve approximation like the so-called double-quadratic curves in Várady[15]) but, first of all, these curves appear naturally when distorting an ellipse with a projective mapping.

What is the real problem in handling these curves? Mathematically, the (planar) conics are described by a second order polynomial of the form:

$$a_{1,1}x_1^2 + a_{2,2}x_2^2 + 2a_{1,2}x_1x_2 + 2a_{1,3}x_1 + 2a_{2,3}x_2 + 2a_{3,3} = 0$$

While this formula is appropriate to perform all calculations which are necessary in for example a modelling system (see Fraux and Pratt[5]) it is inadequate to *draw* the corresponding conic. Indeed, practically all graphics devices available today are prepared to render (on the hardware/firmware level) line segments; in other words, the "ideal" mathematical curve must be approximated by an appropriate polyline or polygon. To achieve a reasonable outlook, this approximation must be quite dense; for example the number of approximation points to render a circle properly must be at least 100, but an approximation with 360 points (that is one point for each degree) is sometimes also necessary.

It is not an easy task to generate these points properly. Appropriate approximation formulae or equations are necessary; we will have some examples later. Some of these formulae (mostly the one describing ellipses) are already known to the graphics community, whereas some others are relatively unknown. On

Report CS-R8938
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

the other hand, having these formulae in hand does not solve all our problems. Indeed, we have to find an answer to the following question: at which point of the graphics output pipeline do we effectively perform this approximation?

The approach chosen usually is to approximate the curve with a polygon/polyline before performing a transformation in the pipeline. Despite the fact that conics form a class of *projective invariant* curves (that is the image of a conic under the effect of a projective transformation will remain a conic), most of the formulae we know about are not projective invariant, that is the data generating these formulae change their geometrical nature when applying a projective transformation. There has been no real investigation we know about which would have made an attempt to overcome this difficulty. Instead, the curves are approximated beforehand, the resulting polygons/polylines are transformed and rendered following already well established methods.

There are, however, two major problems with this approach. First of all, there is a loss in speed, storage etc. As we have seen, the number of generated points tend to be relatively large; all these points have to be transformed, that is a matrix-vector multiplication has to be applied and, in case of a projective and non-affine transformation, and additional division by the last coordinate value is also to be performed (the so-called "projective division"). By applying some alternative methods which we have developed and are presented below, a speed improvement of at least 20% can be achieved.

Speed is not the only issue (and having all these super-fast computers invading the marketplace, this argument might be less and less important). There is also a problem with the quality of the approximation. When approximating for example a circle with 360 points, we get a fairly regular geometrical ordering of the points which, if displayed directly, will produce an acceptably smooth outlook. However, if a transformation is applied against this set of points, this "regularity" will be lost. Some of the line segments will become much longer than others; in these areas the resulting polyline will have a "jagged" effect whereas on some other parts of the curve the density of the points will be unnecessarily high. This negative effect is even more disturbing if the approximation is connected to some kind of shading of a three dimensional surface. It is very difficult to keep track of these distortions which may be, in case of a more complicated projective transformation, very significant indeed. The only way of reducing this effect is to postpone the approximation step as "far" as possible and to produce the resulting polyline after the transformations instead of prior to it.

The real difficulty with this approach is the fact that a non-affine projective transformation will, as we have already mentioned, "destroy" a number of geometrical characteristics of the points. As an example remember that the centre of an ellipse might not be longer the centre any more; furthermore, the image of an ellipse is not even an ellipse in some cases; it may become a hyperbola or a parabola. Consequently, if we want to describe a means to handle conics by approximating them after the transformation in the output pipeline, we have to achieve two goals: a) represent the conics in a *projective invariant* way and b) have a concise view on all three major classes of conics. This is what we will attempt to do below. For this purpose, we have to make heavy use of the mathematical field which provides us with a unified description of conics, namely projective geometry. Therefore we have to begin by giving an overview of the necessary tools. The reader may also refer to other related works like the book of Penna et al.[13], the tutorial notes of Herman from the Eurographics '88 conference[8], or some other more classical textbooks on projective geometry[2-4, 6, 11].

In the first part of the paper we will deal exclusively with two dimensional conics and two dimensional projective transformations, although the three dimensional case might be more important for practical purposes. This is done for didactical purposes; indeed, the notions and formulae are more easily understandable within the framework of a projective plane than a projective space. Once the underlying principles are understood for a planar environment, their generalisation to three dimensions may be done without too many problems; this will be done in the third part of this paper. The main results may be summerised as follows.

- To each class of conics a small set of points will be assigned; these are the *characteristic points* related to the curve (they are not all points of the curve). These characteristic points have the following major properties:
  - They are geometrically well describable points, that is it is easy to generate them from

different applications.

- It is very simple to generate the matrix of the conic using these points.
- Appropriate approximation formulae may be derived using these points to approximate the curve with polygons or polylines.

- It will be shown that the best way to transform a conic is to transform three of its points and its matrix; indeed this step can be done in a projective invariant way. Additionally, both the points and the matrix may be derived easily using the characteristic points.
- Finally, it will be shown that the characteristic points may be generated out of three points of the curve and its matrix. This can be done for all classes of conics and, furthermore, the exact classification of the conic can also be done without problems.

Using these facts we can deduce the following approach for conic generation and/or approximation:

i) The conic is defined either directly by its matrix and three of its points or by its characteristic points;

ii) in case the characteristic points are given in *i)*, the matrix and three points of the curve are generated;

iii) the points and the matrix of the curve are transformed;

iv) the characteristic points of the (transformed) curve are generated; at the same time it is clarified whether the image is an ellipse, a hyperbola or a parabola;

v) using the characteristic points the approximation formula is generated and the curve may be rendered.

2. The two dimensional case

2.1. Mathematical preliminaries

Points in a projective plane are described by homogeneous coordinates (for the sake of simplicity, we will denote the set of homogeneous coordinates by $\mathbf{PR}^3$; its elements are column vectors). These coordinates denote *affine* points if the last coordinate value is non-zero and *ideal* ones, if it is zero. Affine points can be identified with usual Eucledian points by performing the projective division, that is by using

$$(x_1,x_2,x_3)^T \rightarrow (x_1/x_3,x_2/x_3)^T \tag{2.1.1}$$

Conversely, the relation

$$(x_1,x_2)^T \rightarrow (x_1,x_2,1)^T \tag{2.1.2}$$

shows how an affine projective point can be created out of an Eucledian one.

Ideal points are sometimes called "points at the infinity"; although this is not a precise definition from a mathematical point of view, it is intuitively helpful (for a more precise definition of ideal points see again the tutorial on the subject[8]). We can imagine the relationship between ideal points and Eucledian notions as follows. If $x = (x_1,x_2)^T$ denotes a usual Eucledian *vector*, than the ideal point $x = (x_1,x_2,0)^T$ will denote the point in the infinity which is in the *direction* defined by x. In other words, ideal points may also be thought of as representing the direction of a line.

In (planar) projective geometry the set of all ideal points is also considered to be a line, the *ideal line*. The traditional Eucledian lines are also lines in the projective sense; the only difference is that the projective counterpart of the Eucledian lines contain one additional ideal point, namely the one describing the direction of the given line. Taking all these notions into account we might now say that any two lines in a projective plane have an intersection point. Indeed, the directions of two parallel lines are the same and, consequently, the intersection of the two lines is exactly the ideal point which represents this common direction.

Just as in Eucledian geometry, we may also speak of different equations of a line. The traditional equation based on normal vectors can be turned into homogeneous form by the following relationship:

If l is a line in the projective plane, then there exists an $n \in \mathbf{PR}^3$ so that

$$x \in l \leftrightarrow n^T x = 0$$

In other words, lines are also represented by homogeneous coordinates; conversely, each element of $\mathbf{PR}^3$ may represent either a point or a line in the projective plane (in fact, n in the equation above represents the normal vector of the line in the classical Eucledian geometry). This fact leads to a remarkable unity in the formulae when describing points or lines; this unity is also referred to as the principle of *duality* in projective geometry. This principle can also be used to simplify a number of formulae (see for example Arokiasamy[1]).

Two distinct points of a projective plane determine a line. To describe this line, a parametric equation is more suitable than the previous one. Indeed, if we denote the two points by $a \in \mathbf{PR}^3$ and $b \in \mathbf{PR}^3$, and the generated line is denoted by ab then:

$$ab = \{ \lambda a + \mu b \} \qquad (2.1.3)$$

$$\lambda \neq 0 \; or \; \mu \neq 0$$

We may also remark that the ideal line of the projective plane might be represented by the homogeneous vector $(0,0,1)^T \in \mathbf{PR}^3$.

2.2. Elements of the projective theory of conics

In Eucledian environment, a conic is described by the equation

$$a_{1,1}x_1^2 + a_{2,2}x_2^2 + 2a_{1,2}x_1x_2 + 2a_{1,3}x_1 + 2a_{2,3}x_2 + 2a_{3,3} = 0$$

By defining the matrix $A = (a_{i,j})_{i,j=1}^3$ so that $a_{i,j} = a_{j,i}$, and by using homogeneous coordinates instead of Eucledian ones, the equation has its counterpart for projective environments as well, namely:

$$\sum_{i,j=1}^{3} a_{i,j} x_i x_j = 0$$

Finally, this formula can be abbreviated by the so called bilinear form, that is:

$$xAx = 0 \qquad (2.2.1)$$

which is a shorthand notation of $x^T(Ax)$. Formula (2.2.1) will be used throughout the whole article. Here and in the whole section we will consider A to be a non-singular matrix, that is $det(A) \neq 0$.

In fact, (2.2.1) can be used in a somewhat more general way to define the notion of *conjugate points*. This definition is as follows:

> The points x and y on the projective plane are said to be conjugate points with respect to the conic represented by the symmetric matrix A if and only if the following equation holds:
>
> $$xAy = 0$$

We could also say that the points of the curve may be characterised by the fact that they are auto-conjugate.

The notion of conjugate points have a number of remarkable properties. Indeed, the following facts are true (their proofs may be deduced from the definitions or they may be found in the textbooks cited above): If $x \in \mathbf{PR}^3$ is a fixed point, the set of all points $y \in \mathbf{PR}^3$ which are conjugate to x form a line of the projective plane. This line is called the *polar* of x; it may be represented by the homogeneous vector Ax. If l is a line in the projective plane, than there is one and only one point whose polar is l; this point is called the *pole* of l. The pole may be also characterised as follows: it is the (unique) intersection point of all the polars generated by the points on l (see figure 1). If $x \in \mathbf{PR}^3$ is a point on the projective plane, x belongs to its own polar if and only if x is a point of the conic itself. In this case, the polar of x will be tangential to the conic at the point x and the homogeneous representation of this tangential is Ax. The pole of the ideal line is called the *centre* of the curve; in case of ellipses and hyperbolae, this coincides with the traditional, Eucledian definition of the centre of these curves.

Two lines l_1 and l_2 are said to form a *conjugate pair of lines* if the pole of l_1 belongs to l_2 and, conversely, the pole of l_2 belongs to l_1. We may speak of a pair of *conjugate chords* as well as of a pair of *conjugate diameters*, denoting a pair of conjugate lines which are chords (resp. diameters) of the conic (diameter is a

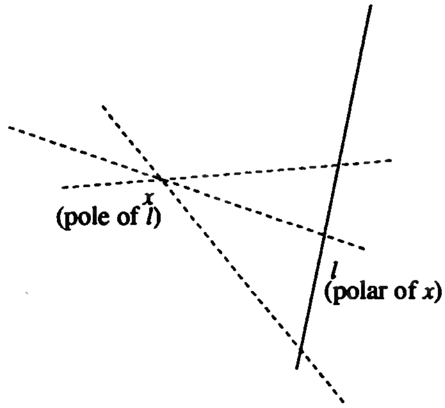

Figure 1

chord containing the centre).

All these definitions are, unfortunately, rather abstract and a certain time is needed to get used to them and to get an intuitive feeling as far as their geometrical meaning is concerned. Figure 2 shows an example which might help in using these definitions. The line l has two intersection points with the conic, p and q. The polars of these points are the two tangentials l_1 and l_2 respectively. In view of what has been said before, the intersection point of these lines, that is r, is the pole of the line l. This procedure is the usual way of generating the pole of a line, provided the line has two intersection points with the curve (which is not always the case). We have also to remark that if the point r is connected to the centre of the curve (denoted by c on figure 2), the resulting line will intersect the line segment pq in its middlepoint. The proof of this fact is, however, much beyond the scope of this paper.

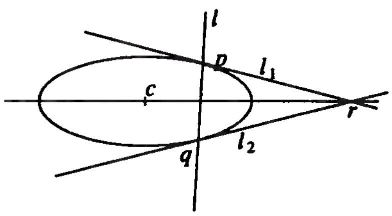

Figure 2

The importance of these definitions becomes clearer when the behaviour of conics in relationship to projective transformations is examined. As we know, a general projective transformation of the plane can be described by a 3×3 matrix. If we denote this matrix by M, and if a conic is represented by the symmetric matrix A then for all $x,y \in \mathbf{PR}^3$:

$$xAy = x^T(Ay) =$$
$$= (M^{-1}Mx)^T(AM^{-1}My)$$

that is, if we denote

$$M(A) = (M^{-1})^T A (M^{-1}) \tag{2.2.2}$$

then we can say that $x,y \in \mathbf{PR}^3$ are conjugate with respect to A if and only if $Mx,My \in \mathbf{PR}^3$ are conjugate with respect to $M(A)$. In other words, the image of a conic under the effect of a projective transformation remains a conic and, furthermore, formula (2.2.2) gives an easy way to calculate the matrix of the image. Also, the property of conjugation is projective invariant. The pole-polar relationship also remains valid across the transformation. Beware! The image of a centre is *not* necessarily a centre: although it is true that the image of the centre will still be the pole of the image of the ideal line, it is by no means sure that the image of the ideal line will still be the ideal line.

This last remark leads us to the notion of *affine transformation*. These transformations are those projective transformations which map the ideal line onto the ideal line. It can be proved that a projective transformation is affine if and only if its representation is of the form:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & \lambda \end{bmatrix}$$

where λ is a non-zero value (the matrices of projective transformations may also differ by a non-zero multiplicative factor just like coordinates; they are ''homogeneous'' matrices). Those familiar with ISO graphics standards like GKS[9] or CGI[10], may recognise segment transformations in this format.

As projective transformations, affine transformations will keep the relationship of conjugate points; additionally, the image of a centre will still be a centre; in other words this property is *affine invariant.*

The last general question we have to deal with is the mutual relationship of a conic and a line. In this respect, the following fact is true:

the number of intersections of a line and a conic may be 0, 1 or 2.

Although we have not proved the statements listed in the paragraph, we make an exception for this latter one. The reason is that the proof gives an effective way of calculating the (possible) intersection points and this is a feature we will need in the future.

The line to intersect with can be described, as we have seen in (2.1.3), by:

$$ab = \{\lambda a + \mu b : \lambda \neq 0 \text{ or } \mu \neq 0\}$$

where a and b are two points on the line. We are looking for two values λ and μ for which

$$(\lambda a + \mu b)A(\lambda a + \mu b) = 0 \tag{2.2.3}$$

holds. In fact, because of the homogeneous nature of the formulae, we are not interested in the exact values of λ and μ; only their relative ratio λ/μ is of interest for us.

Equation (2.2.3) can be also rewritten by:

$$aAa\lambda^2 + 2aAb\lambda\mu + bAb\mu^2 = 0$$

If $aAa \neq 0$, we may consider $\mu \neq 0$; indeed, if this were not the case, then $\lambda = 0$ would also hold, which is impossible. Similarly, if $bAb \neq 0$ then $\lambda \neq 0$. We may consider the first case; that is we can divide the equation by μ^2 to get

$$aAa(\lambda/\mu)^2 + 2aAb(\lambda/\mu) + bAb = 0$$

clearly, this equation may have 0, 1 or 2 solutions; by solving it we also get an explicit value for the (possible) intersection point(s).

The relationship between lines and conics has a very important consequence. Indeed, we can apply the theorem to a special case to get a simple means of classification for conics. Namely:

> The number of ideal points belonging to a conic may be 0, 1 or 2. (The set of ideal points being the ideal line, this is just the special case of the previous statement). In case this number is 0, the curve is an ellipse (or a circle); in case this number is 1, the curve is a parabola with the axis determining the ideal point; in case this number is 2, the curve is a hyperbola, with the two asymptotes determining the two ideal points.

2.3. Some General Formulae

In the present section a list of formulae or short calculation methods will be presented. These formulae should be considered as forming a set of *elementary steps* like a set of elementary routines in a more complicated programming environment. In fact all our further calculations will be performed by a repetitive use of some of these methods. This also means that an efficient implementation of these steps may be crucial to the overall performance of the algorithms to be presented in the later paragraphs.

The formulae themselves are by no means new mathematically. Some of them may be found in textbook like Penna et al.[13], while others may be deducible easily. What might be considered new is the fact that the emphasis is different from the one in classical projective geometry textbooks; we are interested in constructive ways of our formulae, to provide an implementability on a computing environment.

The reason of this somewhat formal approach is as follows. If all subsequent algorithms in the sequel are based on these steps then a way to generalise our approach from 2D to 3D might be to try to generalise these elementary steps only; by this approach, all subsequent description remain valid. This is, in fact, what we will try to do in section 3. This is why we have to list here statements which are very simple in 2D (see e.g. [1] and [2] in the list); while these are trivialities in case of 2D, subsequent calculations might be necessary for the very same steps in 3D.

To begin with, three formulae must be presented which are not directly used in the sections to come, only by the general steps listed in this section (they are "local subroutines" to them) and they will have to be modified in case of 3D.

If $a \in \mathbf{PR}^3$ and $b \in \mathbf{PR}^3$ are two points, compute ab (the line determined by the two points).

We have already a formula for that purpose, which is:

$$ab = \{ \lambda a + \mu b \} \qquad (2.3.1)$$

$$\lambda \neq 0 \; or \; \mu \neq 0$$

If $u \in \mathbf{PR}^3$ and $v \in \mathbf{PR}^3$ represent two lines, let us denote by uv the intersection of these lines. Determine the homogeneous coordinates of this point.

The three coordinates are given by:

$$uv = \det \begin{bmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} \qquad (2.3.2)$$

where $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ denote the vectors (1,0,0), (0,1,0) and (0,0,1) respectively.

If $a \in \mathbf{PR}^3$ and $b \in \mathbf{PR}^3$ are two points, compute the homogeneous representation of ab.

The three coordinates are given by:

$$ab = \det \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} \qquad (2.3.3)$$

Let us see now the general computation steps we will need in what comes.

[1] We know the coordinates of at least two ideal points.
Indeed, $(1,0,0)^T \in \mathbf{PR}^3$ and $(0,1,0)^T \in \mathbf{PR}^3$ are two distinct ideal points.

[2] We know the homogeneous representation of the ideal line.
Indeed, $(0,0,1)^T \in \mathbf{PR}^3$ represents the ideal line.

[3] If two lines are known by having two points on each of them, compute their intersection point.
This is just the application of the formulae (2.3.2) and (2.3.3).

[4] If $a \in \mathbf{PR}^3$ and $b \in \mathbf{PR}^3$ are two points and A is a symmetric matrix representing a conic, compute the number of intersection points and the eventual intersection points themselves.

See previous section.

[5] If $a \in \mathbf{PR}^3$ and $b \in \mathbf{PR}^3$ are two points and A is a symmetric matrix representing a conic, compute the pole of ab.

Based on what was said in the previous section, Aa and Ab are the polars of the points a and b respectively. Following the previous section again (and figure 1), the pole is $(Aa)(Ab)$. This latter can be calculated by using (2.3.2).

[6] If $a \in \mathbf{PR}^3$ and $b \in \mathbf{PR}^3$ are two points, A is a symmetric matrix representing a conic and, furthermore, $c \in \mathbf{PR}^3$ is a point on the conic, compute the intersection of the tangential at c and ab.

This could be done reusing the previous steps but an alternative method is as follows. We want to find appropriate λ and μ numbers so that:

$$(\lambda a + \mu b)Ac = 0$$

taking into account that Ac gives the homogeneous representation of the tangential line at c.

That is:

$$\lambda aAc + \mu bAc = 0$$

Again, if $aAc \neq 0$ then $\mu \neq 0$, that is we can divide; the result is:

$$\lambda / \mu = -bAc / aAc$$

2.4. General Overview of the Projective Representation

As it might be clear from the previous sections, if the matrix of a conic is known, it is fairly straightforward to classify the conic. Indeed, the number of intersection points with the ideal line determines the nature of the conic itself. The ideal line being just a normal line in a projective environment, using step [1], formula (2.3.1) and finally step [4] of the previous section these intersection points (if any) may be easily calculated. We also know that it is easy to keep track of the effect of a projective transformation on a conic by transforming its matrix (see (2.2.2)). It is therefore straightforward to concentrate on the matrix of a conic to find a compact form of storage and projective invariant representation of a conic. This is indeed what we will do.

The following steps will be elaborated in more detail below.

- A parametric equation for each type of conic will be described. These equations will have in common the fact that they will define the points of the conic in a function of a real parameter t running in a finite interval (mainly $[0,2\pi]$). This description gives an easy way of generating the points of the curve; additionally, by defining a finite subset of the base interval, an approximation of the required density may also be defined.
- It will also be shown that each type of parametric equation depends on a small number of geometrically significant *characteristic points* related to the curve. In other words, knowing the exact coordinates of these points the parametric equation is automatically generated.
- Finally, it will also be shown that knowing the matrix of the curve plus maximally three points of the curve itself, all characteristic points may be calculated using the steps [1]-[6] of the previous paragraph.

Let us consider for now that these facts are known. As a consequence, a means of storing and representing a conic might be simply to store three of its points plus its matrix. We have seen that these data are easily transformable and by calculating the (eventual) ideal points of the curve the fact whether the curve is transformed into an ellipse, a hyperbola or a parabola may be easily decided. As a next step, the set of characteristic points of the curve should be determined out of the transformed data and, finally, the parametric equation of the curve may be generated. It is therefore true that by elaborating on the previous points we will have found a solution to the problem presented in the introduction. This is what we will do in what follows.

2.5. Parametric Equations

The parametric equations for the three major kinds of conics will be given below. In all three cases the general curve will be described as the affine image of a simple one. This has two major advantages: it helps to find the parametric equation but it also gives a way (using (2.2.2)) to find the matrix of the curve. We have to remark that in case of an actual implementation, the transformation given below should be combined with the actual projective transformation of the pipeline first to use formula (2.2.2) only once to avoid unnecessary computations. However, these are already technical details.[‡]

We will make use of the following unicity theorem: if four non-collinear points of a conic plus the tangential in one of these points are given, the conic is uniquely defined. This unicity theorem assures us that the transformations we will use to define the parametric equations may not generate another conic than the one we want. The reader who has no practice in projective geometry should be aware that in some cases one of the four points is an ideal one or, as in the case of the parabola, the tangential may also be the ideal line.

2.5.1. Ellipse

The simplest ellipse is a unit circle. There is also a well known parametric equation to describe the points of the circle, namely:

$$(cos\,(t), sin\,(t))^T \quad (0 \le t \le 2\pi) \tag{2.5.1}$$

The matrix of the circle is also very simple, namely

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{2.5.2}$$

For our purposes, we have to describe some geometric features of the circle from a projective geometric point of view. If we look at figure 3 and compare it with figure 2, we can deduce that the pole of the X axis is in the ideal point of the Y axis, that is $(0,1,0)^T$. Conversely, the pole of the Y axis is $(1,0,0)^T$, that is the ideal point of the X axis. In other words, the two coordinate axis form a conjugate pair of lines, more exactly a conjugate diameter pair (we could also say that the radii cq and cr form a pair of conjugate radii).

We know that an affine transformation keeps the conjugate diameters (it keeps conjugation because it is a projective transformation and it keeps the centre because it is affine). If we define therefore an affine transformation which transforms the circle on figure 3 into the ellipse of figure 4 by $c \to c'$, $q \to q'$ and $r \to r'$, the result will be an ellipse (the ideal line does not change) and the lines $c'q'$ and $c'r'$ will be a pair of conjugate radii. The matrix of the transformation is also straightforward. By denoting $u' = q' - c'$ and $v' = r' - c'$ we get:

$$\begin{bmatrix} u_1' & v_1' & c_1' \\ u_2' & v_2' & c_2' \\ 0 & 0 & 1 \end{bmatrix} \tag{2.5.3}$$

This fact is also true conversely: if we know a pair of conjugate diameters of an ellipse, this conjugate diameter pair will define an affine transformation of the form (2.5.3) which will transform the unit circle into the given ellipse. The transformation in (2.5.3) can be combined with the parametric equation in (2.5.1) to produce the parametric equation of the ellipse, that is (by using a vector equation to simplify the formulae):

[‡] In all three cases the simple conic serving as a starting point for our calculations could be described by simpler formulae as well (e.g. $(x,y) = (t,t^2)$ in case of parabolae) but we should never forget that we are looking for formulae which are independent of the special position of the curve with respect to the coordinate system.

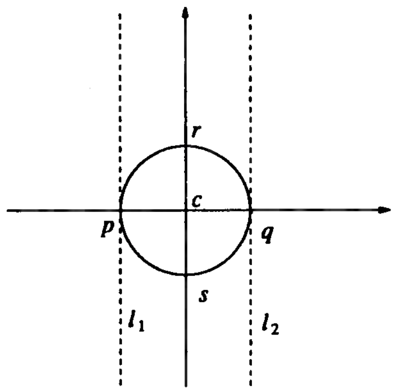

Figure 3

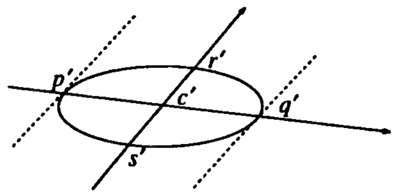

Figure 4

$$\cos(t)u' + \sin(t)v' + c' \qquad 0 \le t \le 2\pi \tag{2.5.4}$$

This formula is the one which has been adopted by the CGI Standard Proposal[10] to describe an ellipse. Clearly, formula (2.5.4) is the kind of parametric equation we were looking for; the characteristic set of points consists of the centre and the endpoints of two conjugate radii.

2.5.2. Hyperbola

The case of the hyperbola is quite similar to an ellipse; only the resulting formulae will be a little bit more complicated. The starting point is again a simple hyperbola, which is the one described by the equation $x^2 - y^2 = 1$ (figure 5). A parametric equation may also be given for that curve (see Penna et al[13].) which is as follows:

$$(\sec(t), \tan(t))^T \quad (0 \le t \le 2\pi) \tag{2.5.5}$$

(the singularities corresponds to the ideal points of the curve). The matrix of the curve is again very simple, namely:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{2.5.6}$$

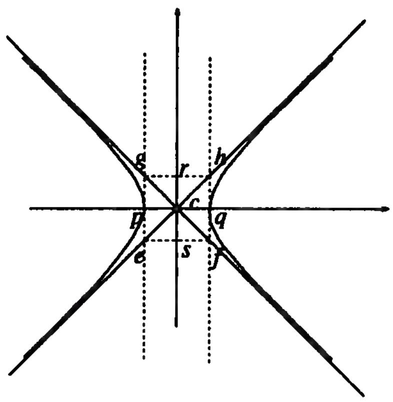

Figure 5

Here again, the two main axis form a conjugate diameter pair, just as in case of a circle. The significant difference is the exact geometrical description of the points r and s of figure 5. As we can see from the figure, the points e,f,g,h are the intersection points of the two asymptotes and the tangentials of the curve in p and q respectively. The asymptotes themselves are also tangential lines; indeed, they are the two tangentials in the two ideal points of the conic. Finally, the points r and s may be generated as the intersection points of the Y axis and the lines gh and ef respectively.

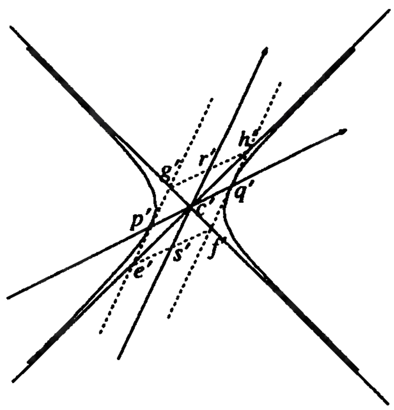

Figure 6

An affine transformation (in fact, all projective transformations) transforms tangentials into tangentials. In other words, an arbitrary affine transformation will transform the configuration of figure 5 into a configuration like figure 6; c will be transformed into c', p into p' etc. The important fact is that the construction we have described for r and s uses affine invariant properties only, that is the image of s will be s' (respectively, r will be transformed into r'). If we define therefore an affine transformation which transforms figure 5 into figure 6 by $c \rightarrow c'$, $q \rightarrow q'$ and $r \rightarrow r'$ (whose matrix will be (2.5.3) again!), we get a parametric equation of the hyperbola, which is:

$$sec(t)u' + tan(t)v' + c' \qquad (2.5.7)$$

$$0 \leq t \leq 2\pi$$

(where u' and v' have the same meaning as in case of an ellipse).

Here again, if we know the centre and the points marked by p,q,r,s on the figures, we can reconstruct the parametric equation; in other words, this set of points might be considered as being the characteristic set of points for a hyperbola.†

Care should be taken when using (2.5.7) to effectively render the curve; the points tend to the infinity, that is both singularities and overflow may occur. In most cases, however, some kind of run-time clip is necessary for a graphics environment anyway; this clip combined with a check on overflow will cut the dangerous line segments automatically.

2.5.3. Parabola

The parabola we start from is the one described by the equation $x^2 = y$ (figure 7). A parametric equation may also be given for that curve (see Penna et al[13].) which is as follows:

$$\left[\frac{cos(t)}{1 - sin(t)}, \frac{1 + sin(t)}{1 - sin(t)} \right]^T \tag{2.5.8}$$

$(0 \le t \le 2\pi)$

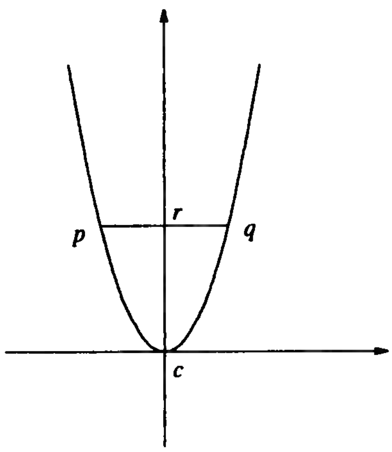

Figure 7

The singularity corresponds to the ideal points of the curve. The interval for the parameter t might be changed; in (2.5.8) the approximation will begin at $(-1, 1)$, will "go around" through (0,0) and $(1, 1)$. Choosing e.g. the interval $-\pi/2 \le t \le 3\pi/2$ would give a more symmetric arrangement. The matrix of the curve is again very simple, namely:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{2} \\ 0 & -\frac{1}{2} & 0 \end{bmatrix} \tag{2.5.9}$$

The Y axis of the parabola is a diameter; indeed, the centre of the parabola is the (only) ideal point of the curve. The line pq is of course *not* a diameter in this case (in contrast to the previous ones); it is, however, conjugate to the Y axis (following figure 1, the pole of the Y axis is the intersection point of the ideal line

† An alternative and somewhat better known parametric equation for the hyperbola would be

$$ch(t)u' \pm sh(t)v' + c'$$

Using the so called hyperbolic cosine and hyperbolic sine functions. However, in this case the parameter t would be defined on the infinite interval $0 \le t \le +\infty$, which would be computationally unstable.

and the X axis, which is the direction of this latter one; this coincides with the direction of pq). It is also known that the point r will be the middlepoint of the line segment pq.

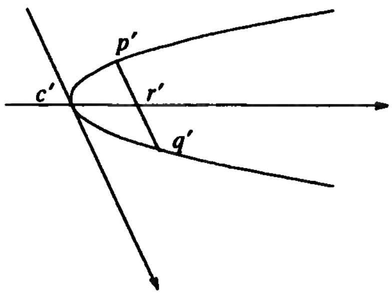

Figure 8

An affine transformation keeps conjugation and keeps also the property of being the middlepoint of a line segment. This means that transforming the parabola on figure 7 to the one of figure 8 would result in the points p', q', r' and c', where the lines $p'q'$ and the axis running through c' (which is the same as the line $c'r'$) will be conjugate to one another. Furthermore, r' will be the middlepoint of the line segment $p'q'$.

We can therefore apply the same methods as before. In this case, however, the meaning of u' and v' is different; indeed, let us define $u' = q' - r'$ and $v' = r' - c'$. Using these definitions the transformation described in (2.5.3) will transform the parabola of figure 7 into the parabola of figure 8. The characteristic points are therefore p', r', q' and c'; the parametrics equation is:

$$\frac{\cos(t)}{1-\sin(t)}u' + \frac{1+\sin(t)}{1-\sin(t)}v' + c' \qquad (2.5.10)$$

$$-\pi/2 \le t \le 3\pi/2$$

2.6. Generation of the Characteristic Points.

The last and most important step in our procedure is to reconstruct the characteristic points of the curve out of its matrix and three of its points. Let us remind what these characteristic points are:

- In case of an ellipse, the centre and two endpoints of two conjugate radii (see figures 3 and 4).
- In case of a hyperbola, the centre, the two intersection points of a chord and the two points (denoted by r and s on figure 5) of the line which is conjugate to the chord and contains the centre.
- In case of a parabola, the intersection point of an axis of the curve (which is, in fact, the other intersection point of a diameter), the two intersection point of the curve and a chord conjugate to the axis chosed before and, finally, the intersection point of the axis and this latter chord.

As we have already seen, it is fairly straightforward to decide whether a conic is an ellipse, a hyperbola or a parabola. Furthermore, as a result of this calculation, the ideal point(s) of the conic (if any) are also known. Additionally, out of the three points which are stored together with the matrix of the conic, at least one will be transformed into an affine (that is non-ideal) point. In fact, in case of an ellipse, all of them will become affine (although we will need only one), in case of a hyperbola at least one will be affine and, finally, two affine points will be at hand for a parabola.

In all three cases, the centre of the curve may be calculated easily as well. In fact, for a parabola this step is not necessary at all: the centre will be the (only) ideal point of the curve. For the other two cases the steps [1] and [6] will lead to the necessary result by calculating the pole of the ideal line.

For an *ellipse* (figures 3 and 4) we may start with one of the known points of the curve; let us take the one denoted by q. Using step [5] and the points q,c, the pole of the line qc may be calculated (in fact, this will be an ideal point). Finally, using this new point and the point *c*, the two intersection points *r* and *s* may also be calculated (although only one of them is needed). The resulting *c*,*q* and *r* points form the requested set of characteristic points.

For a *hyperbola* (figures 5 and 6), the situation is a little bit more complicated. Taking one of the affine points of the hyperbola and denoting it by *p*, the other intersection point of the corresponding diameter may be calculated (step [4]). For each of the two asymptotes two points are already known (the centre and the two ideal points of the hyperbola respectively). With the help of these data the points *e*,*f*,*g* and *h* (intersection points of the asymptotes and the tangentials at the diameter endpoints) may also be computed (using step [6]). Furthermore, the diameter which is conjugate to the diameter *pq* is also determined; indeed, the pole of *pq* determined by [5] gives a second point to it (the first being the centre). Finally, the intersection points *s* and *r* may be calculated using step [3].

In case of a *parabola* (figures 7 and 8) two affine points of the curve are already known (*p* and *q*). The pole of the line *pq* may be connected to the (known) ideal point of the curve; this will be a diameter. Based on the remark we made in connection with figure 2, the intersection of this diameter and the line segment *pq* (which can be calculated using [3]) will be the middlepoint of *pq*. Additionally, the (other) intersection point of the diameter with the curve may be calculated using [4]. This will be *c*, the last missing characteristic point.

2.7. Arcs

Before we move to 3D, we have to address the problem of arcs. Indeed, the use of conic arcs might be even more important (e.g. for surface modelling) than whole curves.

One of the advantages of the parametric formulae given in the previous paragraphs is that they are all invertible. In other words, if a point on the curve is known, the corresponding *t* value might also be calculated. Hence, handling of arcs becomes quite straightforward: the two endpoints of the arc and one intermediate point must also be transformed and these data will determine an appropriate subinterval of the parameter interval. The only small problem which should be handled run-time is to know which "half" of the curve should be taken for the arc (think of an ellipse); the presence of the third point on the arc will however decide that unambiguously.

3. The 3D Case

3.1. Mathematics Again

The projective space is very similar to the projective plane. Ideal points are defined following the same scheme; homogeneous coordinates are used to describe points (of course, they are now members of $\mathbf{PR}^4$ instead of $\mathbf{PR}^3$) and the relationship between affine and ideal points is essentially the same as in paragraph 2.1.

There are of course differences. In case of projective space, the inherent duality described before stands for points and *planes*: an element of $\mathbf{PR}^4$ may describe a plane. The set of all ideal points is now considered to be a plane, the *ideal plane*, representable by $(0,0,0,1)^T \in \mathbf{PR}^4$. Lines may be expressed by formula (2.1.1).

Geometrically, planes of the projective space are projective planes (this seems to be a triviality but, in fact, requires a formal proof in mathematics). The ideal points belonging to a given plane (that is the directions parallel to it) form the ideal line of the plane. We may speak of the intersection of an affine plane and the ideal plane, which is the ideal line of the plane. In other words, every two planes in the projective space have an intersection which is a line (just as every two lines in the projective plane have an intersection which is a point).

The notion of conics may be generalised onto projective spaces as well; the only difference is that the symmetric matrix in use should be 4×4 instead of 3×3. These conics are the so-called quadratic surfaces (hyperboloids, paraboloids, hyperbolic paraboloids etc.). Their classification is much more complicated than in the case of planar curves; however, they form again a class of surfaces which is invariant to projective transformations (the way we have deduced formula 2.2.2 was independent of dimensions). By luck, we do not need a complete overview of all these surfaces in computer graphics; in fact, they are rarely in use.

The intersection of a plane and a quadratic surface leads to a planar conic on the plane. Indeed, if our plane happens to be the plane $x^3 = 0$, this can be easily seen by just putting a 0 to all relevant places of the equation of the surface; the result is a second order equation for the remaining coordinate values. If the plane is of a general position, it can always be transformed into the plane $x^3 = 0$ by using an orthogonal transformation.

These intersection curves are what we are really interested in. Also, it is very easy to associate a quadratic surface to a planar conic: one has to construct a generalised *cylinder* (we might also called it a sweep surface). This means that the curve should be moved along a line not contained by the plane of the conic (see figure 9). In the simplest case, when the conic lies in the $X-Y$ plane, it also very simple to give the equation of such a surface. Indeed, if A is a 3×3 matrix then the matrix describing the corresponding surface may be:

$$A_c = \begin{bmatrix} a_{1,1} & a_{1,2} & 0 & a_{1,3} \\ a_{2,1} & a_{2,2} & 0 & a_{2,3} \\ 0 & 0 & 0 & 0 \\ a_{3,1} & a_{3,2} & 0 & a_{3,3} \end{bmatrix} \tag{3.1.1}$$

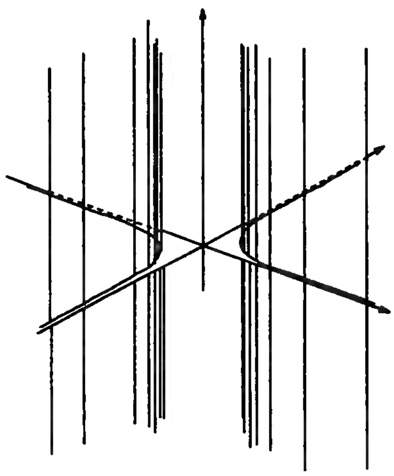

Figure 9

Care should be taken that the matrix A_c is singular. In fact, it can be shown that in case of 3D, if the matrix of a conic is singular, it is either the matrix of a (generalised) cylinder or that of a cone. The proof of this theorem would lead us far beyond the scopes of this article; the interested reader should consult for example Kerekjártó[11] for the details.

3.2. Generalisation of the Projective Representation of Conics for 3D

The way which seems to be promising is as follows. The original curve we want to describe is an ellipse, a hyperbola or a parabola in space. The plane of the curve is determined by the points describing the curve (conjugate radii endpoints etc.). Out of these the normal of the plane can be calculated easily.

Formulae (2.5.2), (2.5.5) and (2.5.9) give the matrices of the simplest ellipse, hyperbola and parabola in a projective plane. Using (3.1.1), the corresponding cylinders may also be described by appropriate matrices. As a next step, we have defined in 2D an affine transformation which transformed the simple conic curves into somewhat more complicated ones (see 2.5.3). In all three cases, we had to define some vectors (u' and v') based on the geometric nature of the curve and the transformation was defined by the relations:

$$(0,0,1)^T \to c'$$
$$(1,0,1)^T \to q'$$
$$(0,1,1)^T \to r'$$

The generation of an appropriate affine transformation may be generalised to 3D as well. Indeed, by defining again $u' = q' - c'$, $v' = r' - c'$ and additionally $w' = u' \times v' - c'$, we may define the following transformation:

$$\begin{bmatrix} u_1' & v_1' & w_1' & c_1' \\ u_2' & v_2' & w_2' & c_2' \\ u_3' & v_3' & w_3' & c_3' \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2.1}$$

This affine transformation has the following properties:

$$(0,0,0,1)^T \to c'$$
$$(1,0,0,1)^T \to q'$$
$$(0,1,0,1)^T \to r'$$
$$(0,0,1,1)^T \to u' \times v'$$

In fact, $u' \times v'$ is the normal vector of the plane containing the curve. As a result, out of the originally planar curve in space we get the matrix of a special quadratic surface. The matrix of this surface may then be transformed using formula (2.2.2) again.

If we choose the same representation of the conic as in 2D, that is its matrix and three points on it, what we have got after the transformation is the matrix of a quadratic surface and three points of the required (transformed) curve again. These three points may be used for an additional purpose in this case: they determine the *plane* of the image of the original (planar) conic (we will see in the next paragraph how the homogeneous representation of that plane can be generated). What we are really interested in, is to reconstruct the set of characteristic points of the resulting curve *on the image plane* (in other words, we are not interested in the overall behaviour of the resulting quadratic surface). As we have already mentioned before, if we succeed in giving a set of alternative methods to cover the steps listed in 2.3, the construction leading to the characteristic points described in 2.6 may be applied without change. Taking into account that all parametric equations listed in 2.5 are vector equations, their use in 3D becomes straightforward to generate the curve (or an arc) in 3D space. This means, that to complete the generalisation of a our approach to 3D, we have only to reconstruct the steps [1]-[6] listed in 2.3. This is what we will do in the next paragraph.

3.3. General Formulae Revisited

The first formulae we have to generalise are the ones corresponding to (2.3.2) and (2.3.4). The new formulae are very much alike indeed; roughly speaking, one dimension should be added where necessary. That is:

If $u \in PR^4$, $v \in PR^4$ and $w \in PR^4$ represent three planes, determine the homogeneous coordinates of the intersection point of the three planes (the intersection point will be denoted by uvw).

The four coordinates are given by:

$$uvw = \det \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \\ v_1 & v_2 & v_3 & v_4 \\ w_1 & w_2 & w_3 & w_4 \\ e_1 & e_2 & e_3 & e_4 \end{bmatrix} \tag{3.3.1}$$

If all determinants in 3.2.1 are zero, the planes meet in a line; this is a singular case.

If $a \in PR^4$, $b \in PR^4$ and $c \in PR^4$ represent three points, determine the homogeneous representation of the planes determined by these three points (the generated plane will be denoted by abw).

The corresponding formulae are the same as (3.3.1); the singularity means that the three points are collinear.

Based on these formulae, we can compute the homogeneous representation of the plane containing the image of the conic curve we want to render. This element of $\mathbf{PR}^4$ will be denoted by n. For the sake if simplicity, the plane itself will be denoted by $\mathbf{P}$. We can also suppose that we have in hand at least one point of the space which is not on the plane. For example, one of the ideal points $(1,0,0,0)^T$, $(0,1,0,0)^T$ or $(0,0,1,0)^T$ should be outside the plane; if not, the curve cannot be rendered at all (all its points are ideal). This point will be denoted by p.

Let us now see the steps listed in 2.3 and how can they be generalised.

[1] We know the coordinates of at least two ideal points of $\mathbf{P}$.

Let us suppose for a while that all three transformed points of the curve are affine. By performing the projective division and substracting one of them out of the two others we get two Eucledian vectors which are parallel to the plane. These vectors will also determine two ideal points. (see (2.1.2).

If one (or two) of the transformed points are ideal, then this is just what we were looking for.

[2] We know the homogeneous representation of the ideal plane.

Indeed, $(0,0,0,1)^T \in \mathbf{PR}^3$ represents the ideal plane.

[3] If two lines of P are known by having two points on each of them, compute their intersection point.

For each line a plane may be defined which contains the line and is different from P. Indeed the two points of the line and p will determine such a plane. The homogeneous representation of these planes may be determined; then, these two elements of $\mathbf{PR}^4$ plus n will determine the intersection point.

[4] If $a \in \mathbf{PR}^4$ and $b \in \mathbf{PR}^4$ are two points in $\mathbf{P}$ and A is a symmetric matrix representing a conic, compute the number of intersection point and the eventual intersection points themselves.

The very same method as in two dimensional case can be applied without change.

[5] If $a \in \mathbf{PR}^4$ and $b \in \mathbf{PR}^4$ are two points in $\mathbf{P}$ and A is a symmetric matrix representing a conic, compute the pole of ab.

Similarly to the two dimensional case Aa and Ab represent the polar of a and b respectively. The difference is that the polar is now a plane instead of a line. However, by calculating $(Aa)(Ab)n$, we get the two dimensional pole on $\mathbf{P}$.

[6] If $a \in \mathbf{PR}^4$ and $b \in \mathbf{PR}^4$ are two points in $\mathbf{P}$, A is a symmetric matrix representing a conic and, furthermore, $c \in \mathbf{PR}^4$ $(c^T n = 0)$, is a point on the conic, compute the intersection of the tangential at c and ab.

Essentially the same formula can be used as in 2.3. Indeed, the tangential plane of A is given by Ac and the intersection of this plane with $\mathbf{P}$ will give the tangential in $\mathbf{P}$.

We have shown therefore that all steps listed in 2.3. may be generalised into 3D, which also means that the projective representation proposed for conics can be used in 3D as well.

4. Conclusions

Beyond the specific algorithmic side of the previous results, two more general conclusions might also be drawn from them. These are as follows.

It seems to be feasible that more research activities should be concentrated on the question of *projective invariant representations* of the output primitives in use. Using somewhat more elaborate tools of projective geometry seems to lead to a number of interesting formulations and algorithms which might be faster, shorter, might result in data compression when describing the primitives etc. Examples have been presented in the already cited tutorial of EG'88[8], the paper of Arokiasamy[1], the paper of Krammer on modelling clip[12] or the presentation of the so called 2.5D graphics systems[7]. There is still a lot to do in the field and a number of interesting results to find.

Another conclusion which might be of interest is related to the approach we have chosen to generate the characteristic points. In fact, we have used a relatively small number of basic tools which are reminiscent of the elementary steps allowed in Eucledian geometry for geometrical constructions (possibility to draw a line, to draw the intersection of two lines etc.). Of course, our basic steps are related to computational methods but, significantly enough, the way of reconstructing the characteristic sets was just the repetitive use of these elementary steps in the right order. In this sense, the whole method we have used might remind us of all those very classical geometric exercises where one has to reconstruct sometimes very complex geometric objects out of simple ones, using a ruler and some elementary steps. It might be an interesting intellectual exercise to look after these old, traditional methods (some geometrical constructions are several hundred or even a thousand years old!); it might well be that with a fresh look at them new approaches can be found which might be useful in graphics algorithms, in drafting systems etc.

References

1. A. Arokiasamy, "Homogeneous Coordinates and the Principle of Duality in Two Dimensional Clipping," *Computers and Graphics* 13 (1989).
2. F. Ayres Jr., *Theory and Problems of Projective Geometry,* Schaum's Outline Series, McGraw-Hill, New-York-Toronto-London (1967).
3. H.S.M. Coxeter, *The Real Projective Plane,* McGraw-Hill, New-York-Toronto-London (1949).
4. L. Cremona, *Elements of Projective Geometry,* Dover Publications, Inc., New-York (1960).
5. I.D. Faux and M.J. Pratt, *Computational Geometry for Design and Manifacture,* Ellis Horwood, New York - Chichester - Brisbane - Toronto (1979).
6. G. Fisher, *Analytische Geometrie,* Vieweg Studium, Grundkurs Mathematik, Vieweg and Sohn, Braunschweig (1985).
7. I. Herman, "2.5D Graphics Systems," in *Eurographics'89 Conference Proceedings*, ed. W. Hansmann, F.R.A. Hopgood and W. Strasser, North-Holland, Amsterdam (1989).
8. I. Herman, "Projective Geometry and Computer Graphics," in *Advances in Computer Graphics IV*, ed. M. Grave and M. Roch, EurographicSeminar Series, Springer Verlag, Berlin - Heidelberg - New York - Tokyo (1989).
9. ISO, "Information processing systems - Computer graphics - Graphical Kernel System (GKS) functional description," ISO IS 7942 (1985).
10. ISO, "Information processing systems - Computer graphics - Interfacing techniques for dialogues with graphical devices (CGI) functional description," ISO DP 9636/1-6 (1988).
11. B. Kerékjártó, *Les Fondements de la Géométrie, Vol. II, Géométrie Projective,* Akadémiai Kiadó, Budapest (1966).
12. G. Krammer, "Notes on the Mathematics of the PHIGS Output Pipeline," *Computer Graphics Forum* 8 (1989).
13. M.A. Penna and R.R. Patterson, *Projective Geometry and Its Application to Computer Graphics,* Prentice-Hall, New Jersey (1986).
14. M.A. Sabin, "The Use of Piecewise Forms for the Numerical Representation of Shape," Reports of the Computer and Automation Institute of the Hungarian Academy of Sciences, Budapest (1977).
15. T. Várady, "Basic Equations and Simple Geometric Properties of Double-quadratic Curves and Surfaces," *CAD Group Document 117,* Cambridge University Engineering Department (1984).