Royal Institute of Technology
Stockholm, Sweden

University of Stockholm
Sweden

DEPARTMENT OF NUMERICAL ANALYSIS
AND COMPUTING SCIENCE

GEOMETRIC MODELLING
- A PROJECTIVE APPROACH -

by
Ambjörn Naeve

TRITA-NA-P8918

CVAP 63

Report from Computer Vision and Associative Pattern Processing Laboratory

NADA (Numerisk analys och datalogi)
KTH
100 44 Stockholm

Department of Numerical Analysis
and Computing Science
Royal Institute of Technology
S-100 44 Stockholm, Sweden

Geometric Modelling
– a Projective Approach –

by

Ambjörn Naeve

TRITA-NA-P8918

CVAP 63

1. INTRODUCTION

The desire to study geometrical relationships between objects arises in many fields of human activity, ranging from architecture to particle physics. In this age of abundant computational power the possibilities of modelling complex geometrical phenomena and simulating their behaviour under "realistic" circumstances are better than ever before. At the same time, however, the knowledge of geometry has declined dramatically over the last 50 years, especially among engineers.

When the "supernova of abstraction" exploded in the beginning of this century, mathematics rapidly took off into the new and exiting dimensions of functional analysis, point set topology and algebraic geometry - to name a few - leaving behind the complex but conceptually familiar research questions of "concrete" geometry. And today, the subtle and ingeneous coordinatization techniques of the old masters - such as Klein, Kummer, Lie, Salmon, Cayley, Clifford and many others - are buried deep in the cellars of the university libraries. Instead of making good use of the representational power of these classical ideas, a world of engineers are busy programming their computers in the same familiar "euclidean and cartesian style", which is the only way they have ever been taught to represent the geometrical world. Euclidean geometry has many excellent qualities, but unfortunately it also has some serious shortcomings that introduce unnecessary analytical and combinatorial complexity in many situations. This complexity can sometimes be overcome by the use of "raw" computational power but it often wreaks havoc with the computational process - leading to unsurmountable problems and computational "dead-ends". The benefits of a unified representation of a geometrical situation - in which the euclidean viewpoint can be embedded as a special case - would therefore be substantial. It is a pleasant mathematical fact that classical geometry provides such a representation - in terms of so called "projective geometry". The present paper is devoted to presenting this projective representation and illustrating how it can be applied to the field of geometric modelling in general. Starting from a (brief) discussion of some of the basic notions of projective geometry (to be read together with [4]), we will describe an interactive environment (called Drawboard) for studying the dynamic properties of geometric constructions in general. The usefulness of this tool will be demonstrated by showing it "at work" on a few explicit constructional examples. Also a few "deductive" examples of our projective representational technique will be presented - in the form of outlines for algorithms to compute geometric entities like the inflection tangent of a curve or the 3D-orientation of a plane. Finally we will discuss the general geometric philosophy that penetrates these ideas as well as the plans for future development.

2. GEOMETRIES AND THEIR GROUPS OF TRANSFORMATIONS

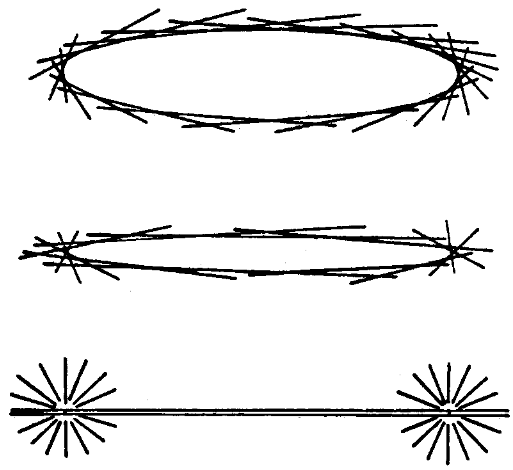
In his famous Erlanger program of 1872, Felix Klein proposed to define a geometry as a collection of statements concerning the relationship between "objects" that remain invariant under a group of transformations. This marks the beginning of the "modern" viewpoint - where each geometry is regarded as a sort of "language", with its own collection of transformations ("verbs") and invariants ("nouns"). "Square" is a noun in euclidean geometry because every instance of this type "survives" the effect of all euclidean verbs i.e. it remains a square when subjected to an arbitrary euclidean transformation (a ridgid motion, a reflection or a similarity transformation). However, "square" is not a noun in affine geometry, because an affine transformation can destroy it and deform it into a parallelogram - which is the affinely invariant aspect of a square. A parallelogram in turn can be destroyed by a general projective transformation - since the latter does not have to preserve the parallelity of two lines - and the only aspect that survives is the quadrangle (or quadrilateral), which is the appropriate projective name for a euclidean "square" or an affine "parallelogram" .

This discussion illustrates the fact that the group of euclidean transformations is a subgroup of the group of affine transformations which is in turn a subgroup of the group of projective transformations. It turns out (see e.g. [1]) that the latter group is the largest of all possible geometric transformationgroups, which is reflected in the fact that it can be represented algebraically as the full matrix group of the corresponding dimension. Every other geometry is characterized by a subgroup of the projective group. This is the content of the famous statement by the great geometer Arthur Cayley: "Projective geometry is all geometry!"

When looked upon from the projective standpoint each metric (distance measure) is derived from a quadratic form Ω in the surrounding projective space. The corresponding geometry (Ω-geometry) is the geometry whose group of transformatons (isometries) preserve the distance measure induced by Ω, and the isometries of Ω-geometry constitute the subgroup of the full projective group that takes each point of Ω into another point of Ω, hence leaving Ω invariant as a whole.

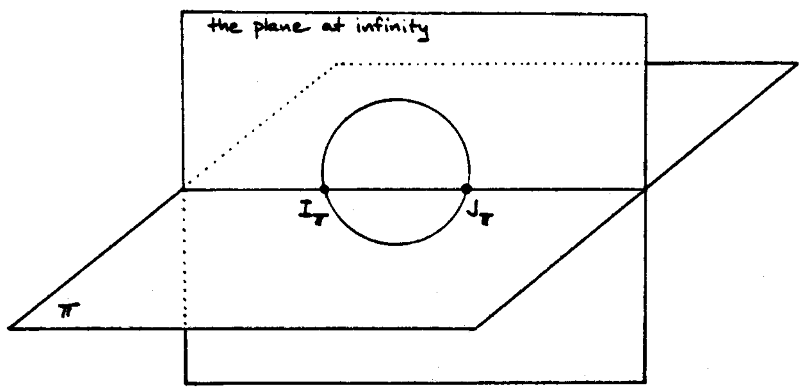
When Ω is a non-degenerate quadratic form, the corresponding Ω-geometry is called "hyperbolic" or "elliptic" depending on whether Ω contains real points or not. Both of these types of geometry are traditionally called "non-euclidean". The "ordinary" euclidean geometry is the result of letting Ω degenerate in a special way.

In two dimensions the degeneration consists of gradually "compressing" a complex, non-degenerate conic Ω and moving it "towards infinity" until it finally "collapses" into two conjugate complex points on a double (real) line at infinity (figure 1). These points are called "the circular points" (or I and J) and it is easy to show that all circles in the plane must pass through them. They are the "rulers" of the metric properties of the euclidean plane, and Cayley was so impressed by their many remarkable properties that he named them "the absolute"!



< figure 1 >

In three dimensions the euclidean form Ω consists of the plane at infinity together with a complex circle on it (figure 2). This circle is called "the sphere circle", since all spheres must contain it. Each euclidean plane π cuts the plane at infinity in a line and this line cuts the sphere circle in two (conjugate complex) points. Hence the intersection of π with the 3D euclidean metric form Ω produces a line at infinity and two conjugate complex points on it (I_π and J_π). This configuration is the 2D euclidean metric form of π and it governs the "internal" euclidean metric of π as embedded in the surrounding 3D-space.



< figure 2 >

Hence we can describe the 2D-metric geometry of π as projectively related to the points I_π and J_π (an example of how this can be done is given in chapter 5) and then relate this "intrinsic" geometry of π to the geometry of the "extrinsic" surrounding 3D-space. For an excellent treatment of these ideas the reader may consult [2], which is a classical "masterpiece" by Felix Klein.

3. THE BENEFITS OF A PROJECTIVE REPRESENTATION

The kind of "intrinsic-extrinsic linkage" described in the last paragraph is highly desirable in many modelling situations where geometrical analysis and deduction (i.e. "geometric reasoning") from a variety of 2D and 3D "input information" is to be performed. As an example, in the field of computer vision the informational contents in a given geometric scene is often a mixture of two different kinds - projective (e.g. incidence relationships as observed by the viewer - or given in some other way) and metric (e.g. knowledge of the size, shape, or relative position of different objects). The informational content within each of these types may itself consist of a mixture of 2D and 3D data. For the viewer to be able to utilize all this information optimally in his attempt to interpret the scene, it is important to have available a spatial representation that allows him to "integrate all the pieces of geometric information into the same puzzle" without making any ad hoc assumptions. The projective representation provides this "common grounds" - making it possible for the totality of given scene-data to restrict the possible image interpretations in a consistent and unified way - hence giving the viewer the power to keep track of "what he doesn´t know" as well as "what he knows". More details on how projective representation techniques can be utilized in computer vision and robotics can be found e.g. in [8].

4. SOME BASIC NOTIONS OF PROJECTIVE GEOMETRY

4.1. PROJECTIVE GEOMETRY IN THE PLANE

The construction of the projective plane $\mathbf{P}^2$ from the ordinary affine plane and the introduction of projective coordinates are described in detail in [4]. We will adopt the same notation here and referring the reader to [4] p.1-12 we will write:

<u>Symbol</u>	<u>Meaning</u>
CTU	the Coordinate Triangle Unitpoint configuration (or base) for the points of $\mathbf{P}^2$
ctu	the dually unified base for the lines of $\mathbf{P}^2$
$[\ \mathbf{x}\ ]_{\mathbf{CTU}}$	the projective coordinates of the point x in the corresponding base

$(x_1:x_2:x_3)$	the projective coordinates of the point **x** written explicitely
$[\ v\ ]_{ctu}$	the projective coordinates of the line v in the corresponding base
$(v_1:v_2:v_3)$	the projective coordinates of the line v written explicitely

4.2. PROJECTIVE GEOMETRY ON THE LINE

In a completely analogous way we can construct the projective line $\mathbf{P}^1$ and introduce projective coordinates for the points on it. Referring again to [4] (p.5) for details we can continue our summary of notation:

Symbol	Meaning
CIU	the Coordinate Interval Unitpoint configuration (or **base**) for the points of $\mathbf{P}^1$
$[\ x\ ]_{CIU}$	the projective coordinates of the point **x** in the corresponding base
$(x_1:x_2)$	the projective coordinates of the point **x** written explicitely

4.3. PROJECTIVE GEOMETRY ON THE POINT

The projective point is dual to the projective line. Hence it is isomorphic to $\mathbf{P}^1$ and is described by the same algebra (see [4] p.8). The lines on a point are thus coordinatized in the same way as the points on a line. For the sake of completeness we list the corresponding notation:

Symbol	Meaning
ciu	the coordinate interval unitline configuration (or **base**) for the lines of P^1
$[\ v\ ]_{ciu}$	the projective coordinates of the line v in the corresponding base
$(v_1:v_2)$	the projective coordinates of the line v written explicitely

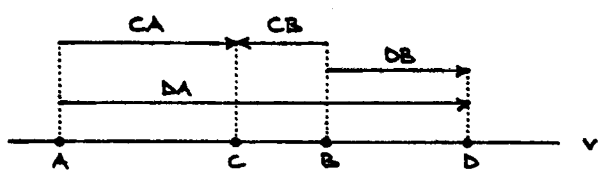
4.4. CROSS-RATIO

There is another way to look at the coordinates of P^1 based on the so called cross-ratio of four points on a line (or four lines on a point). The cross-ratio can be introduced "projectively" - without any reference to metrical ideas (see [2]) - but it is convenient to make use of an "auxiliary metric" in its definition and then prove that it is a well defined projective entity i.e. invariant under a sequence of perspective projections onto other lines or points. This approach also has the advantage of conveying an intuitive feeling for the size of the cross-ratio as a function of the relative position of the four points (lines).

Consider the four points **A, B, C, D** on the line **v** (figure 3). The cross-ratio of the pair **A, B** relative to the pair **C, D** is denoted by **(AB|CD)** and it can be defined as

(1) $$(AB|CD) = (CA/CB)/(DA/DB)$$

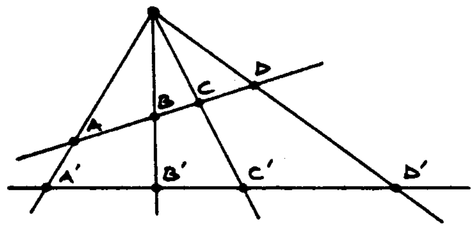
where **CA** is the euclidean (signed) distance between the points **C** and **A** using an underlying orientation chosen on the line **v**.



< figure 3 >

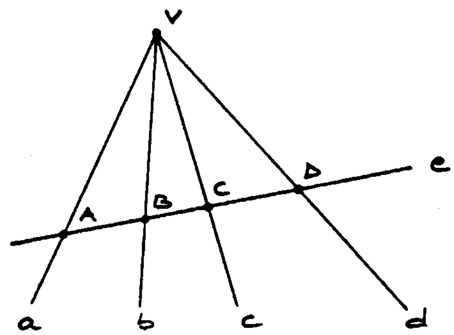
The cross-ratio **(AB|CD)** can be thought of as the ratio between how **C** divides **AB** and how **D** divides **AB**. It is a simple trigonometric exercise to show that this quantity is invariant under perspective projection i.e. that

(2) $$(A'B'|C'D') = (AB|CD)$$



< figure 4 >

This means that the dual situation (two pairs of lines **a,b** and **c,d** on a point **V**) can be assigned a well defined cross-ratio, namely by cutting the four given lines by any line **e** not on **V** (figure 5). The cross-ratio **(ab|cd)** is then simply defined to be the value of the cross-ratio of the four corresponding "cutting points" **(AB|CD)**. By (2) this value is independent of the choice of **e** and hence it is a property of the two linepairs **a,b** and **c,d**.



< figure 5 >

It is easy to see that the position of a variable point **X** on a given line **v** is uniquely determined by the cross-ratio (μ) of **X** and three fixed points **A**, **B**, **C** arbitrarily chosen on **v**. If we let

(3) $\mu = (AB|CX) = (CA/CB)/(XA/XB)$

we can observe that

(4)
$$\begin{aligned} X = A &\iff \mu = \infty \\ X = B &\iff \mu = 0 \\ X = C &\iff \mu = 1 \end{aligned}$$

Hence it is natural to call **A**, **B** and **C** the ∞-point, the 0-point and the 1-point respectively. Furthermore, if we write μ as a ratio:

(5) $\mu = \mu_1/\mu_2 = \mu_1 : \mu_2$

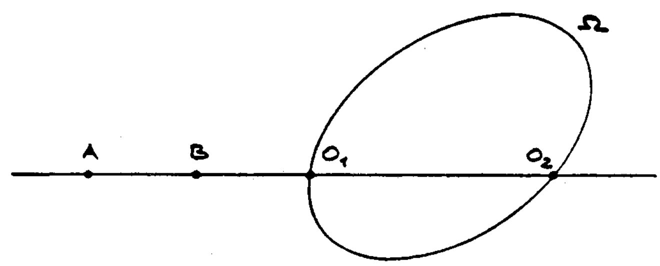
we see that **A**, **B** and **C** correspond to the **(1:0)**-point, the **(0:1)**-point and the **(1:1)**-point. One can easily verify that $(\mu_1 : \mu_2)$ are the **C.I.U.**-coordinates of the point **X** corresponding to the choice of **A** and **B** as coordinate-interval points and **C** as unit point.

5. DESCRIBING METRIC INFORMATION PROJECTIVELY

Having defined the cross-ratio we can now fulfill our promise from chapter 2 to explain how metric properties can be described in a projective way. As we mentioned in that chapter, the embedding of metric geometry against the "universal" projective background is done by expressing the metric information - such as distance and angle - as projective relations with respect to the "metric form" (i.e. the quadratic form Ω that induces the metric as described above). We stress again that this makes it possible to utilize metric information in a projective setting, since the corresponding relations are projectively invariant.

So let us consider the concept of "distance" - which is the fundamental quantity upon which metric geometry is resting. The Ω-distance between two points **A** and **B** (i.e. their distance measure in Ω-geometry) is defined to be (a scaling constant times) the logarithm of the cross-ratio of **A** and **B** with respect to the two points $\mathbf{O_1}$ and $\mathbf{O_2}$ that are common to the line **AB** and the conic Ω:

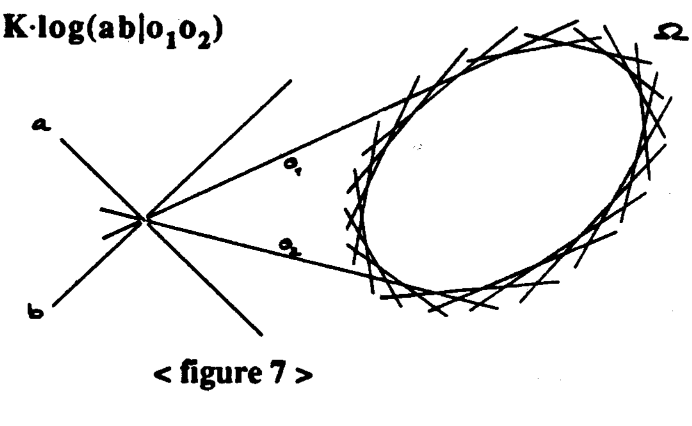
(6) $$\text{distance}(A,B) = k \cdot \log(AB|O_1O_2)$$



< figure 6 >

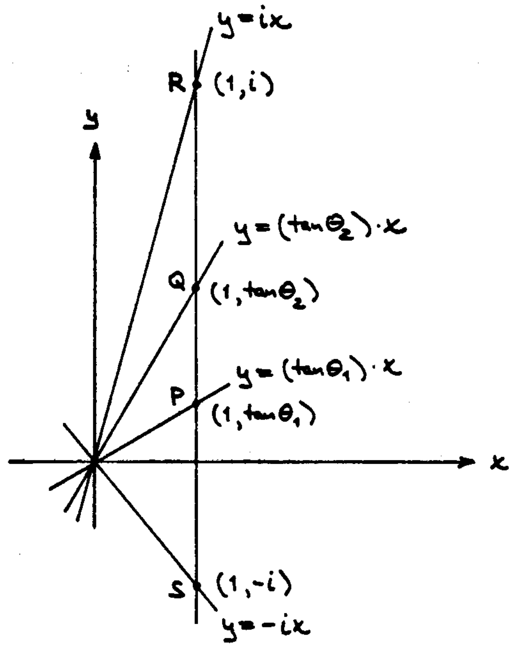
Since points and lines are dual objects in the projective plane, the "distance" (i.e. the angle) between two lines can be obtained by dualisation:
The Ω-distance between two lines **a** and **b** (i.e. their angle) is (a constant times) the logarithm of the cross-ratio of **a** and **b** with respect to the two lines $\mathbf{o_1}$ and $\mathbf{o_2}$ that are common to the point **ab** and the conic Ω:

(7) $$\text{angle}(a,b) = K \cdot \log(ab|o_1o_2)$$



< figure 7 >

It is instructive to calculate the euclidean angle between two lines using this formula. Referring to figure 8 we get



< figure 8 >

$$(PQ|RS) = \frac{RP/RQ}{SP/SQ} = \frac{\dfrac{(i - \tan\theta_1)}{(i - \tan\theta_2)}}{\dfrac{(-i - \tan\theta_1)}{(-i - \tan\theta_2)}} =$$

$$= \frac{(i\cos\theta_1 - \sin\theta_1)(-i\cos\theta_2 - \sin\theta_2)}{(-i\cos\theta_1 - \sin\theta_1)(i\cos\theta_2 - \sin\theta_2)}$$

$$= \frac{(\cos\theta_1 + i\sin\theta_1)(\cos\theta_2 - i\sin\theta_2)}{(\cos\theta_1 - i\sin\theta_1)(\cos\theta_2 + i\sin\theta_2)} =$$

$$= \frac{e^{i\theta_1} e^{-i\theta_2}}{e^{-i\theta_1} e^{i\theta_2}} = \frac{e^{i(\theta_1 - \theta_2)}}{e^{-i(\theta_1 - \theta_2)}} = e^{2i(\theta_1 - \theta_2)}$$

and we see that if we choose the constant $K = 1/(2i)$, the formula (7) is indeed expressing the familiar euclidean angle. This calculation was discovered by Laguerre (1853) and it is referred to as *Laguerre's angle formula.*

6. DRAWBOARD

6.1. GENERAL DESCRIPTION OF THE SYSTEM

Let us consider the problem of representing the world of geometrical constructions in a way that allows positional modifications of the participating parts. Each construction has its own "history" that can be regarded as a mixture of "random choice" (e.g. choose two points **P** and **Q**) and "canonical induction" (draw the line **PQ**). When an object **A** takes part in the construction of another object **B**, we can think of **B** as a child of **A** and **A** as a parent of **B**. In the example above the line **PQ** is a child of the point **P** and the point **Q**, and both of these points are parents of **PQ**. Each "birthprocess" thus contains an element of "choice" and an element of "necessity". Sometimes one of these complementary elements may be missing altogether. In our example the constructions of the points **P** and **Q** contain only choice but no necessity, while the construction of the line **PQ** contains only necessity and no choice. If we carry our example construction one step further and choose e.g. a point **R** on the line **PQ**, we see that **R** contains both elements - the choice being provided by "the random God" and the necessity by the parent **PQ**. In the same way each child must have an element of "parental necessity" in its construction as opposed to the "ancestors" (parentless objects like **P** and **Q**) that are characterized by "pure choice".

By a positional modification (PM) in the construction of a geometric object we mean a "reconstruction" of the object that is changing the element of choice (by making a different choice) but maintaining the same element of necessity (i.e. the same parental relationships). Hence a positional modification of the point **R** means choosing it in a different position but still on the line **PQ**.

Also we do not allow positional modifications that destroy any parental relationships of future generations. If we want to modify the position of the point **Q**, we are free to choose a new location for it anywhere - except at the point **P** - since this would effectively annihilate the existence of the line **PQ**, hence destroying the parental relationships between **P**, **Q** and **PQ**.

In order to implement these restrictions on the allowable positional modifications we can regard each object as possessing a certain "type" - containing information about its own kind of parental relationship. The points **P** and **Q** of our example are of type "free-point" (since they have no parents), the line **PQ** is of type "line-on-2-points" and the point **R** is of type "point-on-1-line". Hence we can state the following general rule concerning the "allowable changes" in our system:

A positional modification of an object in a given construction is allowable if and only if it maintains the type of every object that is part of the construction.

Let us introduce the name APM for an Allowable Positional Modification of an object in a geometric construction. When we carry out an APM at a certain stage of the construction, its effect will in general influence the "future part" of the construction - leading to other APMs of children, grandchildren etc. etc. Note that we cannot be certain that a PM is really an APM until we have observed its effects on all the future generations of the construction - convincing ourselves that all the generated PMs are in fact APMs. This means that we must maintain some sort of "singularity check" that signals the original PM if - at some later stage - it leads to a construction that is "close enough" to collapsing - by being too ill conditioned to be allowable. Only if our PM passes this test can it be accepted as an APM - and be allowed to propagate its effects through the entire construction. If the test fails the entire construction must be "rolled back" and the system restored to its previous state.

6.2. INDUCING INTRINSIC COORDINATES BY PROJECTION

The projective representation discussed earlier gives us the ability to administer the entire hierarchy of APMs that is involved in updating a geometrical construction consistently. This is done by equipping each object "at birth" with its own intrinsic coordinate system (**ics**) that is used to keep track of the positions of all future children of this object. When a child is subjected to an APM, it is the coordinates of the child in the **ics** of its parents that are allowed to change - provided there is room for such a change (i.e. provided there was an element of choice present in its creation). The **ics** of each created object is constructed by projection from the projective coordinate system that is used to describe the entire surrounding space. This global, unchanging coordinate system - that is chosen once and for all - will be denoted by the term "**superbase**".

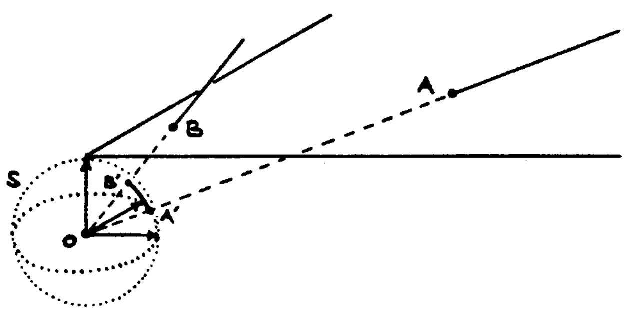
6.3. THE TWO-DIMENSIONAL IMPLEMENTATION

In what follows we will restrict our discussion to two dimensions and describe an implementation in $\mathbf{P}^2$ of our dynamic constructional hierachy called "**Drawboard**", which is the only functioning system that has been designed so far. Drawboard presently works with points lines and conics, but it is easily expandable to include other types of objects as well.

The system is implemented in an object oriented style - using the Common Lisp flavors package that supports multiple inheritance of instance-variables and methods for different object types. This makes it possible to greatly increase the modularity and decrease the length of the code - by doing things "at the right level of abstraction". As an example, the types (flavors) "point" and "line" are abstract flavors in the system. We never create any instances of these types since this would prevent us from an optimal exploitation of

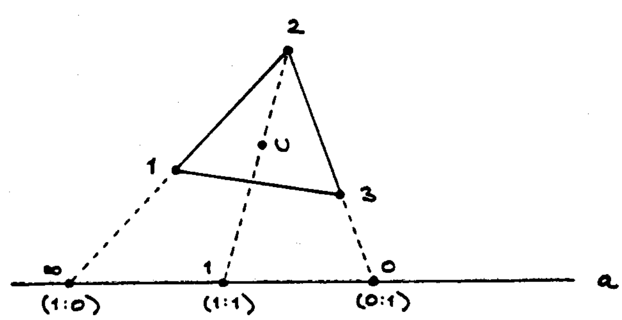
duality. What we actually create in the system are instances of type "free-point", "point-on-one-line" and "point-on-two-lines" (and the analogous flavors for lines). The (concrete) flavor "free-point" is a mixture of the abstract flavors "point" and "free-dual", the flavor "point-on-one-line" is a mixture of "point" and "dual-on-one-dual" and the flavor "point-on-two-lines" is a mixture of "point" and "dual-on-two-duals". The three different "dual-flavors" are in turn mixed from the abstract flavor "dual". The flavor "dual-on-one-dual" - for instance - is a mixture of the flavors "dual" and "cross-ratio" - the latter flavor representing the "internal coordinate" of the object with respect to the ics of its "parent-dual". This has the advantage of treating dual situations in the same setting - using the identical algebra - which is one of the major benefits of projective geometry in the first place. Hence we can defer until the last possible moment (i.e. the moment of creation of a graphic representation) the decision in our system of which of the objects that are points and which that are lines - something that makes the dualization of a given construction virtually automatic.
The hierarcical parents-children relationships are handled by a basic (abstract) flavor "relative" which is an ingredient flavor of every object in the system. This flavor is responsible for checking the consistency of a PM in the construction, and allowing its effects to propagate if these are allowable on each level - or restoring the entire construction to its previous state if "something goes wrong somewhere".
Let us illustrate the induction of an ics for a line or a point in Drawboard. (This is done in the same way for both kinds - since the corresponding method works on the type "dual".) We will always assume (unless we explicitely state otherwise) that all projective coordinate triples (for points as well as lines) have been normalized so that the sum of the squares of the three components is equal to one. This amounts to introducing a unit sphere **S** centered on the origin **O** of the euclidean "superspace" of [4] figure 4. Using this sphere we can induce a "local metric" on $\mathbf{P}^2$ by measuring the "local distance" of two points **A** and **B** by (the minimum of) the corresponding arc-length on **S** of the two "equatorial arcs" **A´B´** subtended by the projections of **A** and **B** onto **S** from **O** (figure 9). It is easy to see that this construction fails to produce a (global) metric on $\mathbf{P}^2$ but it works locally and this "local metricity" gives us a way to determine two important things:

(i) if two points are close enough to be considered identical
(ii) if a point and a line are close enough to be considered incident.



< figure 9 >

Referring to figure 10, imagine that we have selected a **CTU** (superbase) for the points and the corresponding (dually unified) **ctu** (superbase) for the lines of $\mathbf{P^2}$. Imagine further that at some point of our constructional process we are creating the line-object **a**. To create an **ics** on the line **a** we will project the **CTU** onto **a** from one of its vertices **1**, **2** or **3**. Using our local metric we can now determine which one of these points that is "least close" to the line **a**. (Note that **a** cannot be close to all three vertices - unless our **CTU** is "pathological").Using this point (the vertex **2** in figure 10) as a centre, we ensure ourselves of the numerically best conditioned "**CTU**-projection" onto the line **a**. The rest of the vertices (**1** and **3**) are projected onto the Coordinate Interval points **(1:0)** and **(0:1)** - they can be taken in any order - and the unit point **U** is projected onto the Unit point **(1:1)**. Hence we get a **CIU** coordinate system for the line **a** which serves as its **ics** to keep track of any "intrinsic" points that we might want to create on **a** in the constructional future.

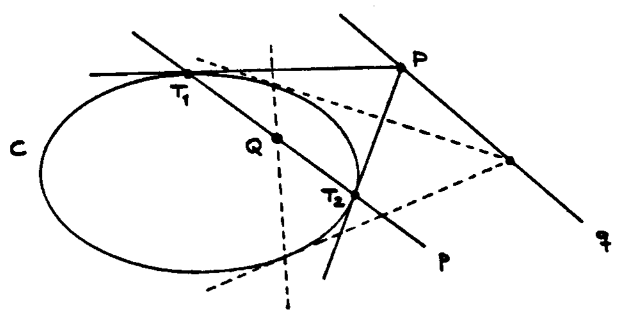


< figure 10 >

7. EXPERIMENTS WITH DRAWBOARD

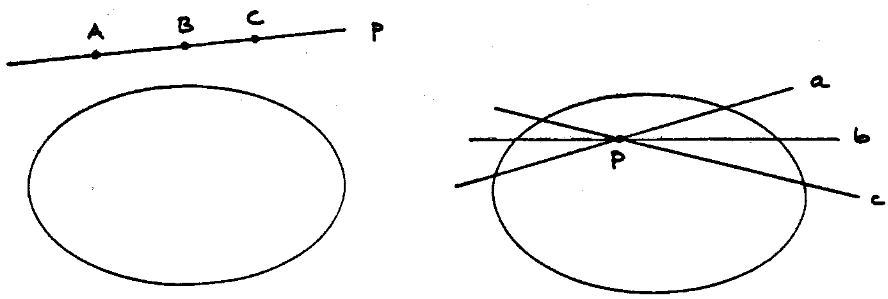
7.1. STUDYING THE EFFECTS OF DUALIZATION

The process of dualizing a geometrical situation is often an important means to promote the understanding of its structure and to increase the computational accuracy of some of its participating components. The so called "Hough transform" in the field of image analysis is a well known example: We can find the "best fitting" line to a given collection of points by dualizing and looking for the best fitting point (dual-line) to the corresponding collection of lines (dual-points). In the digital "pixel-world" this turns out to generate a much better computational process - since we can easily count the number of times that any point (pixel) is on a line of our collection. Doing so for each point in the dualized image and choosing the "max-count-point" gives us a good candidate for the best fitting dual-line and hence (by dualizing back again) for the best fitting line.
There are of course many ways to achieve a dualization of a given geometrical construction in $\mathbf{P}^2$. The only thing that is required is to carry out a bijective mapping from $\mathbf{P}^2$ to itself - taking points to lines and lines to points in such a way that the points on a line are mapped onto the lines on a point and vice versa. In Drawboard, one way to accomplish this is by making use of a classical construction known as "polarization in a conic".

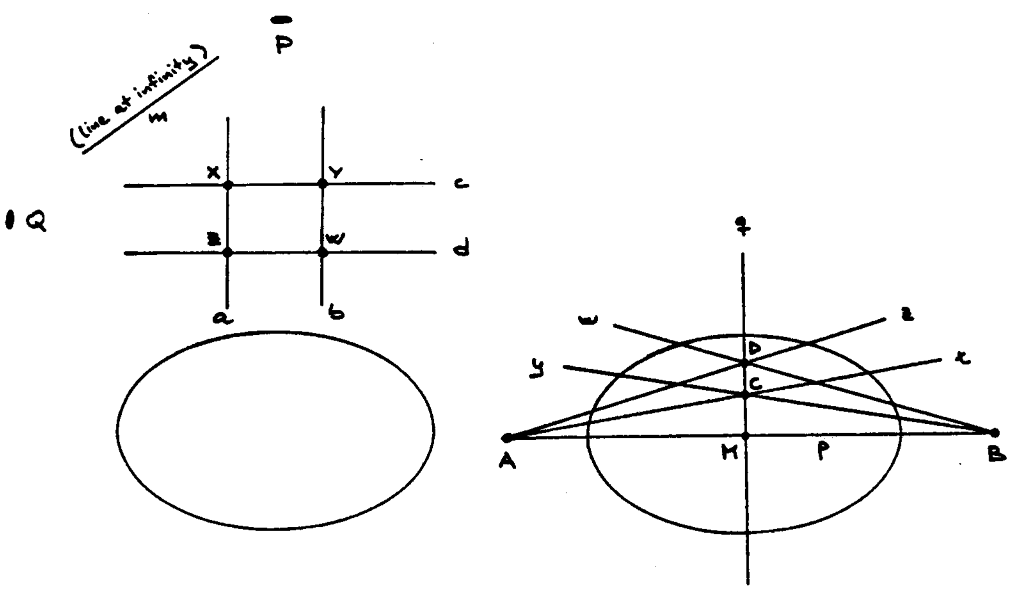


< figure 11 >

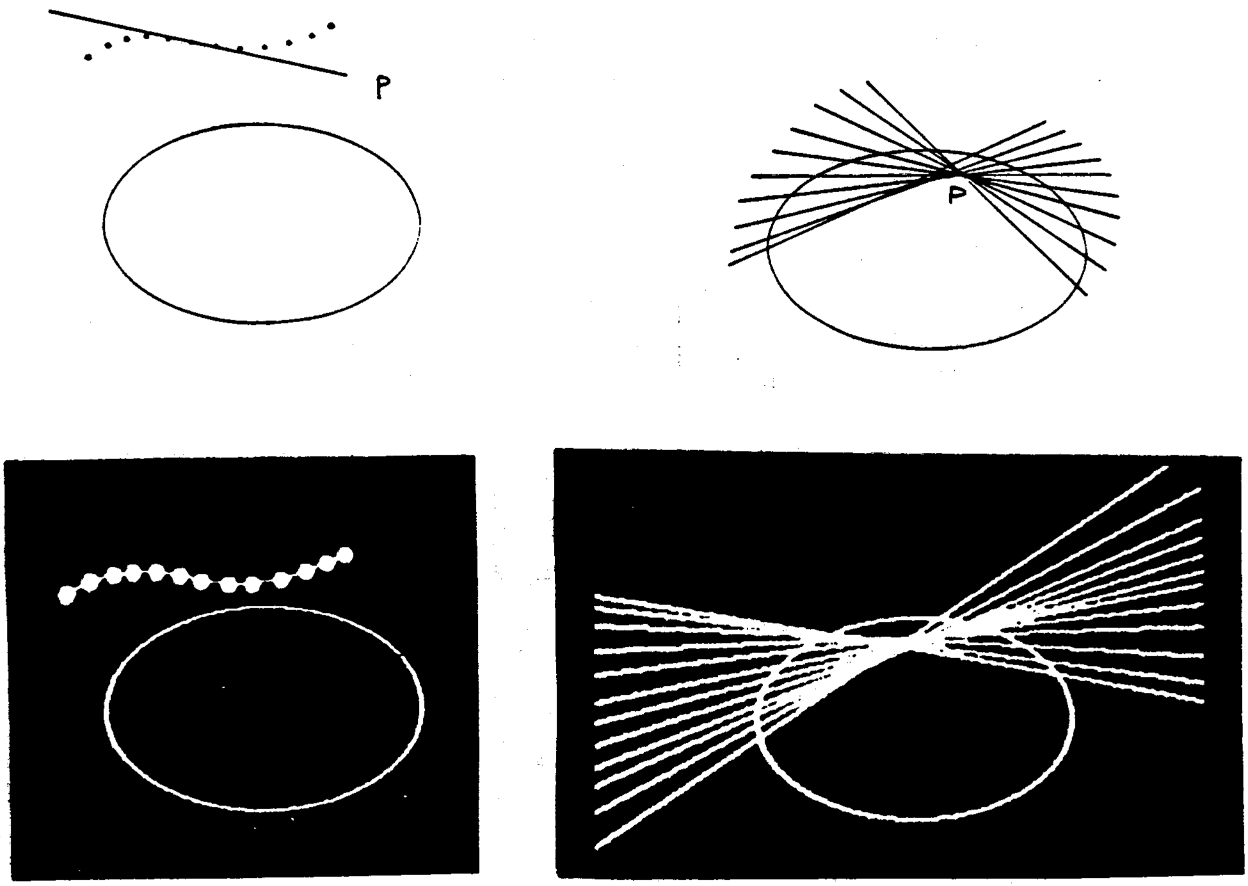
Imagine a fixed conic **C** as in figure 11 and consider a point **P**. The "polarized image" of **P** (the polar line of **P** with respect to the conic **C**) is the line **p** joining the points of tangency (T_1 and T_2) with **C** of the two tangent lines to **C** that pass through **P**. The "dualization property" of this construction is a consequence of the following "theorem of reciprocity": If the point **P** is moved along a line **q**, the polar line **p** will be moving "along a point" **Q** - i.e. it will be turning around **Q**.The point **Q** is called the polar point (or the pole) of **q** with respect to **C**. Note that if **P** is "outside" of **C**, then **p** has two real points of intersection with **C** and if **P** is "inside" of **C** - like **Q** - then **p** has no real points of intersection with **C**. Finally, if **P** is located on **C**, then **p** coincides with the tangent to **C** at **P**. We therefore have created a correspondence (**P <--> p , q <--> Q**) of the required type, working fo all points and lines - and hence we have "implemented" a dualization of $\mathbf{P}^2$.



< **figure 12** >



< **figure 13** >



< figure 14 >

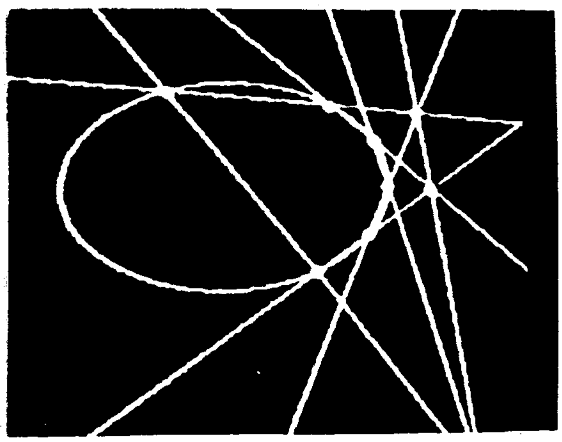
The figures 12 to 14 show the effect of polarizing a few simple geometrical constructions. We can observe from figure 14 (and it can be proved analytically) that the polarized image (polar reciprocal) of a point-curve with inflection-tangent **p** is a line-curve with a cusp-point **P**. This duality can be used to develop an algorithm to compute inflection-tangents by computing their dual cusps instead - using the same counting method as in the Hough transform described above. This is a good example of how a deductive idea can emerge from an interactive experiment. However, we will leave this idea for now and return to it later (chapter 8.1) when discussing some deductive aspects of the projective representation.

7.2. INTERACTIVE EXPLORATION OF GEOMETRICAL THEOREMS

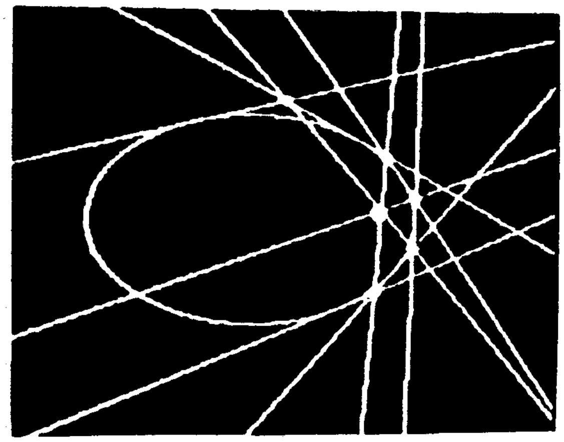
An obvious way to use Drawboard is to study geometric constructions interactively. This can greatly increase the intuitive feeling for the contents of a geometric theorem - since such a theorem almost invariably deals with the effect of performing some geometrical construction - stating different kinds of relations between the participating parts. These

relations can often be directly observed on the screen and their invariance tested "on line" by performing various APMs of the "constructional indata" of the theorem.
As an example let us consider the so called "Pascal theorem" of classical projective geometry (figure 15) - discovered by the 16 year old Blaise Pascal in 1642:

<u>Theorem:</u> Let the points $A_1 \ldots A_6$ be the six vertices of a hexagon inscribed in a given conic C. Consider the three pairs of opposite sides A_1A_2 - A_4A_5 , A_2A_3 - A_5A_6 and A_3A_4 - A_6A_1 and their corresponding points of intersection P, Q and R. The three latter points are always on one line (x).



< figure 15 >



< figure 16 >

This theorem made such a deep impression on the scolars of his time that Pascal´s construction was generally referred to as "hexagramum mysticum"!
By constructing this "mystical" configuration in Drawboard we can move the points $A_1 \ldots A_6$ around on the conic C and observe the corresponding positions of the "Pascal line" x. This gives an excellent appreciation of the "dynamics" of the configuration which is impossible to convey on a "static" piece of paper like this one.
By polarizing the Pascal configuration in the conic C (figure 16) we get an illustration of the corresponding dual theorem - discovered by Brianchon around 1810:

<u>Theorem:</u> Let the lines $a_1 \ldots a_6$ be the six sides of a hexagon circumscribed on a given conic c. Consider the three pairs of opposite vertices a_1a_2 - a_4a_5 , a_2a_3 - a_5a_6 and a_3a_4 - a_6a_1 and their corresponding lines of intersection p, q and r. The three latter lines are always on one point (X).

The Brianchon point X and the Pascal line x are of course pole and polar of each other with respect to the conic C. It is a remarkable illustration of the subtlety of the concept of duality that it took the geometers more than 150 years to discover the dual of the Pascal configuration - and yet another 30 to realize that the two configurations were in fact dually related.

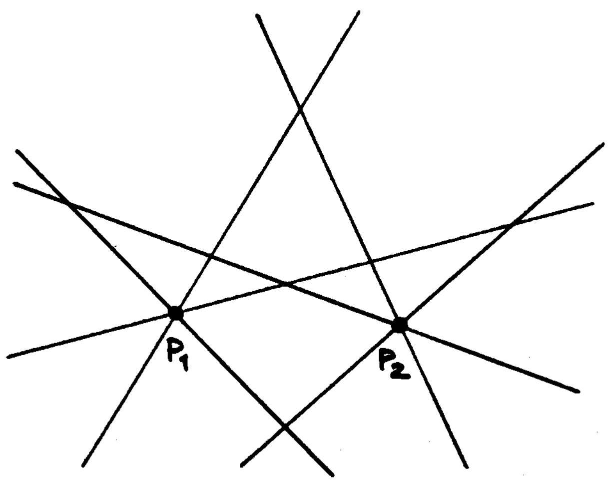
As another example, let us consider the so called "Steiner conic" - a construction discovered by the German geometer Jacob Steiner.
We have pointed out earlier (figure 5) that four lines on a point **P** have a well defined cross-ratio. Hence each line on **P** can be characterized by its cross-ratio relative to three fixed lines on **P** (see also the discussion in chapter 6.3).
Now, consider two points P_1 and P_2 and choose three lines a_1 , b_1 , c_1 on P_1 and three lines a_2 , b_2 , c_2 on P_2 (figure 17). Hence, to each line l_1 on P_1 there corresponds a unique line l_2 on P_2 such that the two four-tuples of lines have the same cross-ratio i.e.

$$(a_1 b_1 \mid c_1 l_1) = (a_2 b_2 \mid c_2 l_2)$$

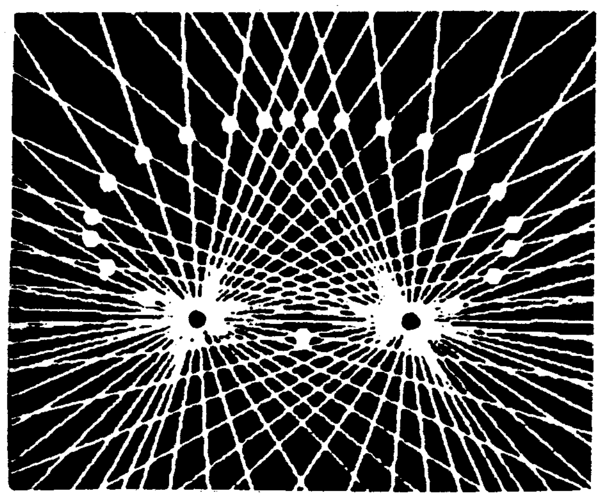
Steiner's theorem can now be expressed the following way:

Theorem: The locus of the point of intersection of the lines l_1 and l_2 is a conic that passes through the points P_1 and P_2.

The Steiner conic that corresponds to the line triples of figure 17 is depicted in figure 18.



< figure 17 >

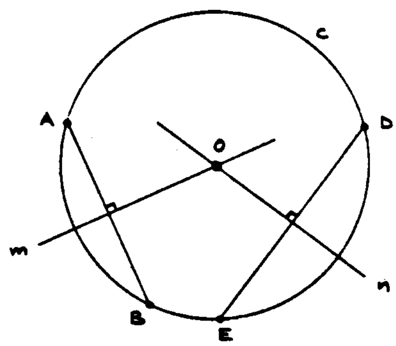


< figure 18 >

7.3. AUTOMATED DISCOVERY OF GEOMETRICAL RELATIONSHIPS

A conceptually more interesting way to work with Drawboard is to use it as a generator of "geometrical concepts" - by finding constructions whose resulting objects are independent of any choices made in each intermediate constructional step. Such objects can therefore be regarded as functions of the constructional indata alone - thus representing a geometrical concept involving only these. As an example, suppose that we start with the constructional indata "circle" (i.e. an instance **c** of this type) and perform the following construction on it (figure 19):

1. Choose two points **A** and **B** on the circumference of the given circle **c**.
2. Draw the line segment **AB**
3. Choose two points **D** and **E** on the circumference of **c**.
4. Draw the line segment **DE**.
5. Draw the midpoint normal **m** of **AB**.
6. Draw the midpoint normal **n** of **DE**.
7. Select the common point **O** of **m** and **n**.



< figure 19 >

Suppose further that we introduce the following "fix-check" procedure after each step:

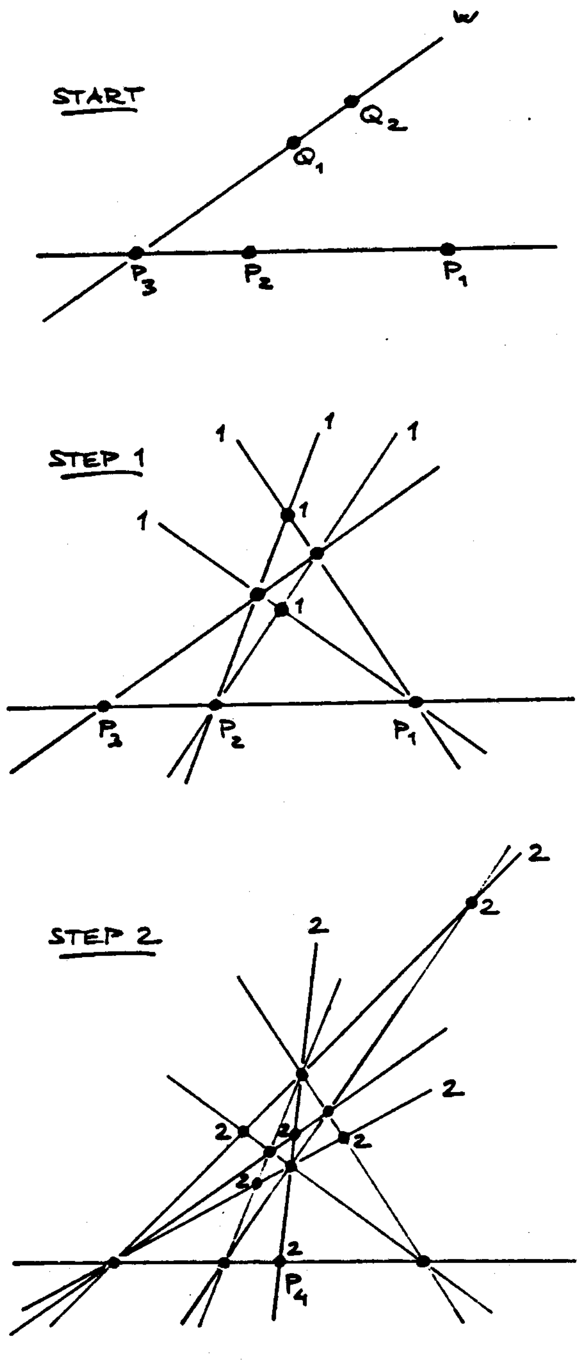
1. Variation of choice:
   Perform an APM on each intermediate object created up til "now"
   that has a "non-zero" element of choice, and compare the position of each "zero-choice" object before and after each modification.
2. Invariance under variation of choice:
   If the change of position of a constructed object is "small enough" during all these APMs, redo the construction of this object with a new set of intermediate constructional choices.
3. Conclusion:
   If the positional change of the object is still "small enough" return the object and consider it a function of the given constructional indata.

It is easy to see that running this fix-check procedure after each step of our construction on the circle **c** above will return the point **O** after the fix-check of construction step 7, because **O** is the first object - the position of which is invariant under variation of each intermediate choice that was made in its construction. Hence the system will recognize **O** as a function of the constructional input data (**c**) alone, and therefore conclude that **O** is a point that can be naturally associated with the circle **c**. The point **O** is of course the centre of **c**, and thus the system has taken the input object "circle" and discovered the concept of centre point for such an object.

In this way we can write programs in Drawboard that "go looking for geometrical relations" (i.e. theorems) of various kinds. Of course the underlying algorithms will be highly exponential and suceptible to "combinatorial explosion" if we just go on constructing new objects by "combining everything with everything" in an indiscriminate way. To utilize the system effectively we should have "a hunch of where to look" and even better - an idea of "how to modify our search strategy if we don´t find what we´re looking for" within reasonable computational time and space. This of course suggests an interactive programming environment (of type LISP) where it is easy to write and test out different search-strategies and strategy-modifying functions. But in spite of its lack of "geometric intelligence" the brute force method can still produce interesting results - as is shown by the following example:

Let us start with the indata "two points (**P_1** and **P_2**) and a third point (**P_3**) collinear with **P_1** and **P_2**", and see if we can program our system to find any geometrical objects associated with this configuration. Referring to figure 20, we will start our constructional process by choosing a line (**w**) on the point **P_3** and two points (**Q_1** and **Q_2**) on the line **w**. Hence our "input configuration" consists of the point pair **P_1** , **P_2** and the point **P_3** - all three on the line v, and our "choice configuration" is made up of the line **w** and the two points **Q_1** and **Q_2** on **w**. Our search strategy will be extremely simple - we iterate the system by letting it construct at each step all the "new" (i.e. non-existing up til now) lines that can be constructed by the pairwise combination of existing points and then (dually) all the new points that can be constructed from the totality of constructed lines. When we have completed this (constructional) part of each iterative step, we perform a random update of our choice configuration - i.e. we generate random positional modifications of the line **w** and the points **Q_1** and **Q_2** until we succeed in making the system accept them as an APM. Knowing now that "positions really have been modified" we can perform our "fix-check" - which must be carried out on all "zero-choice-objects" that have been constructed during the iteration. If we find any fixed objects, we break the iteration and return with these - otherwise we let the construction go on into its next iterative step.

The result of performing this "structure-search" is shown in figure 20. In iteration step number 1 the process creates 4 new lines and 2 new points (marked by 1 in step 1) - none of which are fixed. Hence the process continues with iteration step number 2, creating 3 new lines and 6 new points (marked by 2 in step 2) - of which the point **P_4** turnes out to be fixed. This point is in fact the well known "harmonic conjugate" of **P_3** with respect to **P_1** and **P_2** - and it is characterized by the fact that **$(P_1P_2|P_3P_4) = -1$**. Hence the system has in effect "discovered" the concept of harmonic conjugacy of a point with respect to two other (collinear) points.


START
w
Q2
Q1
v
P3
P2
P1
STEP 1
1
1
1
1
1
1
P3
P2
P1
STEP 2
2
2
2
2
2
2
2
2
2
P4

< figure 20 >

7.4. THE CONIC ON FIVE POINTS

Through the choice of a coordinate system (CTU) in $\mathbf{P}^2$, a conic will be represented by (the zeroes of) a homogeneous quadratic polynomial S(x,y,z):

$$(8) \qquad ax^2+by^2+cz^2+dxy+exz+fyz = 0$$

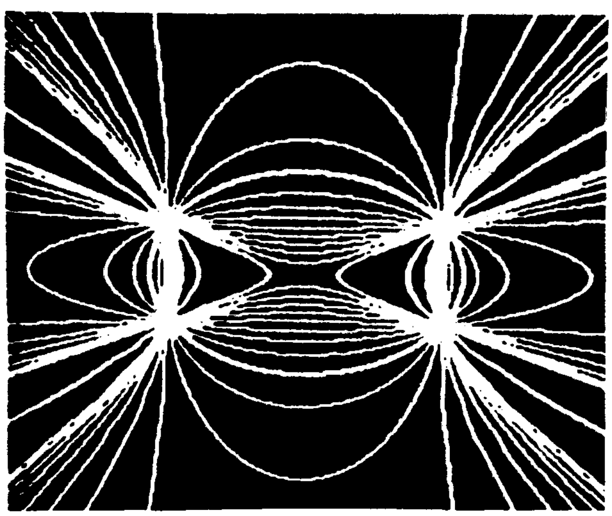
From this it is clear that we can require of a conic (C) that it should pass through five given points (P_1, ..., P_5) - and that this request will determine C uniquely, provided that the five points are chosen so that the corresponding linear system of (five) equations in the six unknown coefficients **a**, ..., **f** (the "coordinates" of C) has rank five. An experimental question that is natural to ask in this respect is the following: How does the geometric appearance of the conic C change with the variation of position of its parents - i.e. the participating points? This kind of information is potentially useful e.g. for the study of curvature phenomena in the field of shape analysis.

7.5. LINEAR FAMILIES OF CONICS - DISTRIBUTION OF SINGULAR POINTS

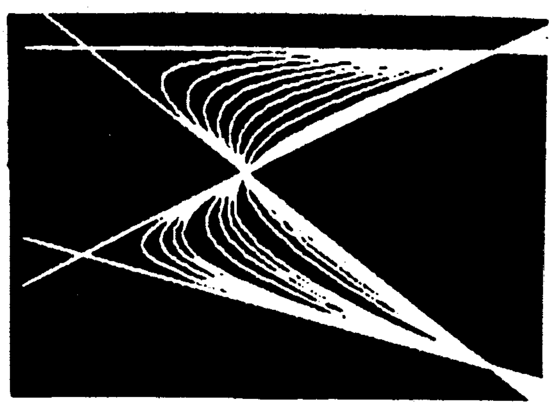
Given two conics S_1 and S_2 we can form the set of all conics the equations of which are linear combinations of the equations of S_1 and S_2:

$$(9) \qquad \mu_1 S_1+\mu_2 S_2 = 0$$

This 1-dimensional linear family of conics is called the **pencil** on S_1 and S_2. Two examples of pencils of conics are shown in figures 21 and 22.



< figure 21 >

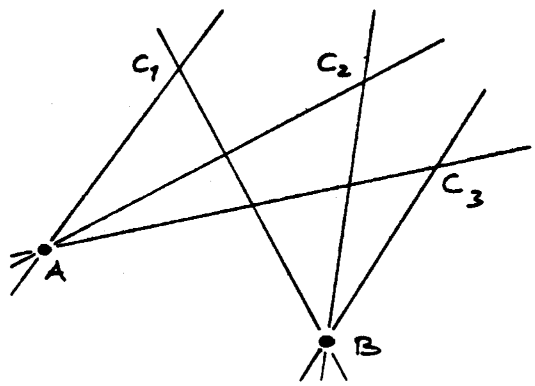


< figure 22 >

The corresponding 2-dimensional linear family on three given conics S_1, S_2 and S_3

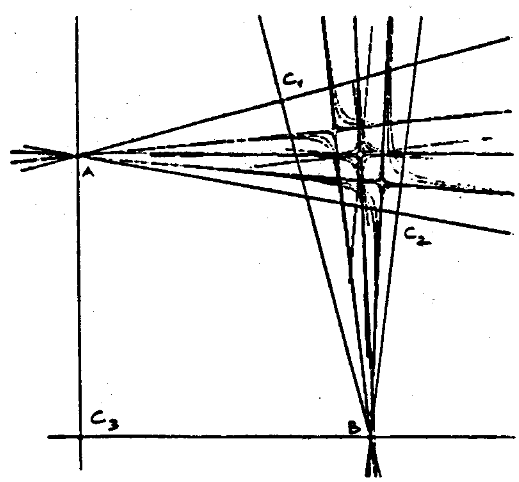
(10) $\mu_1 S_1 + \mu_2 S_2 + \mu_3 S_3 = 0$

is called the **net** of conics on S_1, S_2 and S_3.



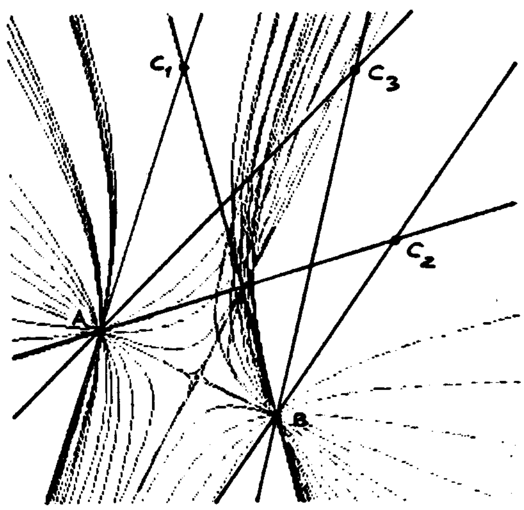
< figure 23 >

A fellow researcher in mathematics here at the RIT in Stockholm a while ago was thinking about the following problem: Consider three pairs of lines in $\mathbf{P}^2$ with one line of each pair passing through a certain point A and the other line of each pair passing through another point **B** (figure 23). The three line-pairs can be regarded as (degenerated) conics C_1, C_2, and C_3. Now consider the net of conics on these three. A generic member of this family will not be degenerated, but for certain values of the "net-coordinates" $(\mu_1{:}\mu_2{:}\mu_3)$ the corresponding conic will degenerate into a pair of lines and hence posess a singular point (the point of intersection of the two lines).
The question was now:
How are these singular points of the net distributed?



< figure 24 >

It didn't take us long to get an idea by performing an experiment in Drawboard. We simply plotted the curves in the net for some suitably chosen values of $(\mu_1:\mu_2:\mu_3)$. The result (figure 24) indicated that the singularities were located on the conic through the five points **A**, **B** (and the singular points of) $\mathbf{C_1}$, $\mathbf{C_2}$ and $\mathbf{C_3}$, and the more we plotted the stronger the indications grew. Of course the line **AB** was a natural candidate to be part of the singular locus, and with a bit of "parameter fiddling" we were able to produce experimental support of this idea (figure 25).

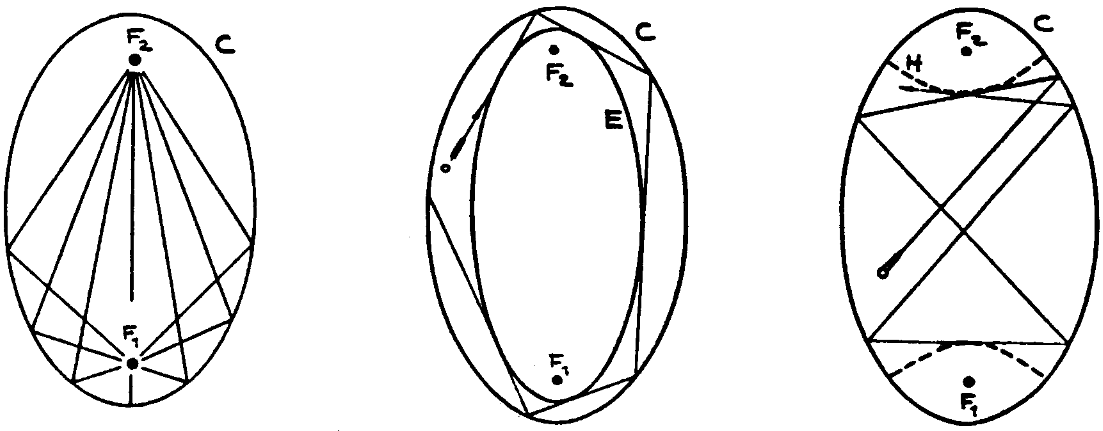


< figure 25 >

By now we had found a conic and a line of singular points - hence the singular locus had to be at least of degree 3 (a cubic curve). We also had developed a strong "gut feeling" that there were no singularities left, i.e. that the singular locus was in fact a cubic (factoring into a conic and a line). Having gotten this far we were encouraged to summon enough mathematical energy to prove our hypothesis geometrically. Later on it turned out that all this is a consequence of a general theorem concerning the singular loci of algebraic curves but that is another story!

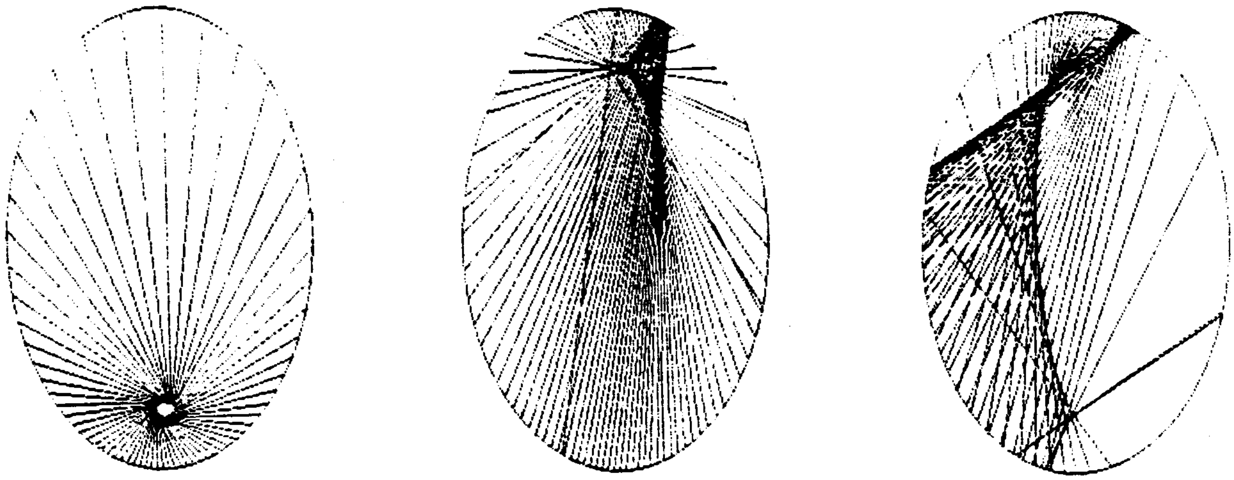
7.6. RAYTRACING THE REFLECTIONS FROM A SLIGHTLY PERTURBED FOCAL-POINT-SOURCE OF LIGHT IN A CONIC MIRROR

It is a well known fact that a mirrorized conic will reflect the light-rays from a point-source situated at one of its two focal points in the direction of the other (figure 26, left).



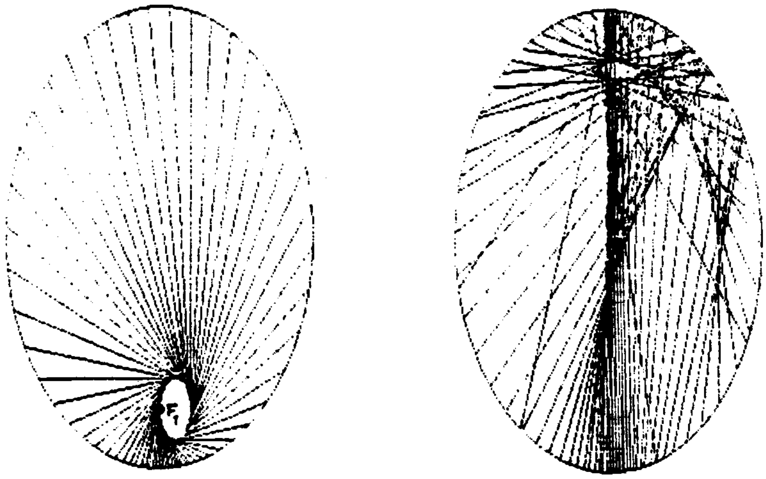
< figure 26 >

Hence a focal-point-source will remain a focal-point-source after any number of reflections in the conic. This is of course a highly unstable condition, and if the initial focal-point-source is ever so slightly perturbed - either by moving the point-source a little "out of focus" or by disrupting the point-source property itself - the successive reflections will scatter the light-rays "all over the place". In the context of studying this perturbed point-focus situation it is natural to look for types of perturbations that remain invariant under reflection in the conic - i.e. that are reflected into perturbations of the same type. To see an example of this kind of behaviour, let us consider a conic mirror in the form of an ellipse **C** with focal-points $\mathbf{F_1}$ and $\mathbf{F_2}$. It is a remarkable geometrical fact that the collection of all rays that are tangent to any ellipse (or any hyperbola) that is confocal to **C** (such as the ellipse **E** or the hyperbola **H** in figure 26, (middle and right) will be reflected by **C** into itself. Of course these "confocal perturbations" are not perturbations at all - in the strict sense of the word - since they can´t be created by an arbitrarily small change of the point-focus configuration. In the search for real point-focus perturbations with some sort of invariance property under reflection, it became interesting to consider perturbations of the following type: Consider a small ellipse **E** with its major axis coinciding with the major axis of **C** and one of its focal-points coinciding with the focal-point $\mathbf{F_1}$ of **C**, and perturb the rays emanating from $\mathbf{F_1}$ in such a way that they are all tangent to **E** (figure 27, left). How will this perturbation behave under reflection? Will the reflected rays be tangent to a similar ellipse centered on the other focal-point $\mathbf{F_2}$? This question may seem a bit strange to ask, but there were other geometrical facts indicating that the answer might in fact be in the affirmative.

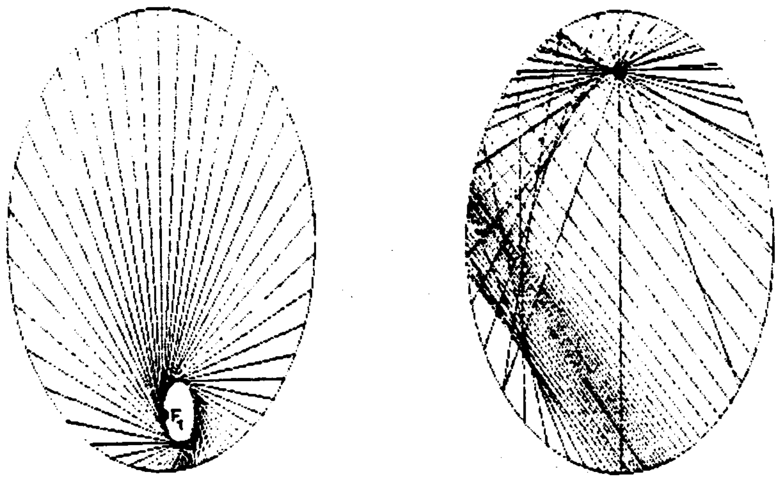


< figure 27 >

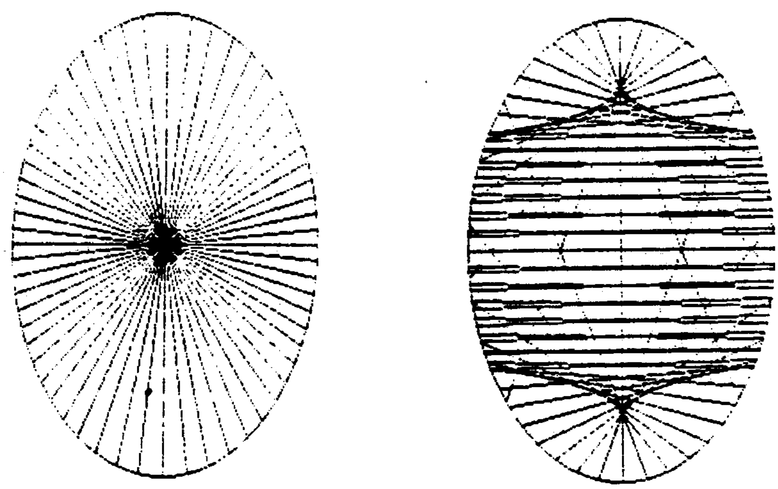
The hypothesis formulated above can easily be tested experimentally in Drawboard. This provides a good example of a "euclidean problem" embedded in the general projective setting - which is how Drawboard represents all geometric information internally. The test can first be performed with the choice of **E** as a circle - since this extra symmetry will give us an intermediate result - telling us if it´s meaningful to continue or not. Figure 27 shows the initial "circular" perturbation of the focal-point-source as well as the result of two successive reflections. Obviously the answer to our question is in the negative and the hypothesis must be rejected. The asymmetry of the first reflection in figure 27 is due to the asymmetry of the start configuration of rays. The rays carry a direction - from the point of tangency with **E** and towards the point of impact with the mirror **C** - and only the part between these two points is drawn in the "start configuration" of figure 27. Figures 28 to 30 show the result of a few other experiments with this configuration. In figure 28 we have translated the "light-source-ellipse" **E** so that $\mathbf{F_1}$ coincides with a point of minimal curvature on **E**, and in figure 29 we have turned the "input-rays" of figure 28 around and made them travel in the opposite direction. It is interesting to observe that this simple change in the input has quite a dramatic (and highly non-symmetrical) effect on the reflected configuration - because of the directional asymmetry discussed above. Finally, in figure 30 we have placed a point source of light in the centre of **C**. The reflected configuration shows two very interesting cusps that seem to be located in the focal points $\mathbf{F_1}$ and $\mathbf{F_2}$ of the mirror **C**. The author hasn´t proved this mathematically, but he would be very surprised if the focal points didn´t turn up in a closer examination of these cusps - since the smell of them is so strong! The point to be made here is not to produce a proof, but to point out how easy it is to generate a variety of "experimental surprises" that help to deepen the intuition and promote the feeling for "what´s going on" in the problem under study.



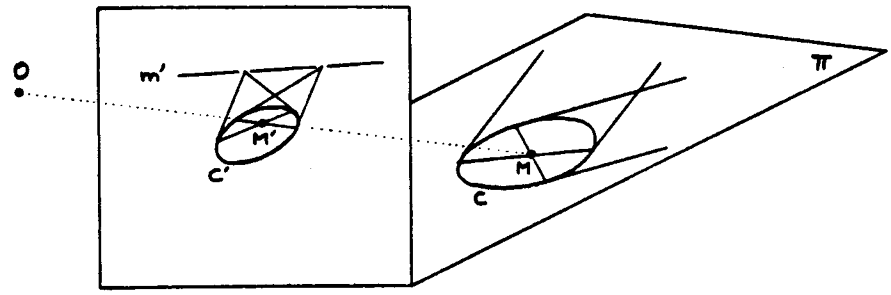
< figure 28 >



< figure 29 >



< figure 30 >



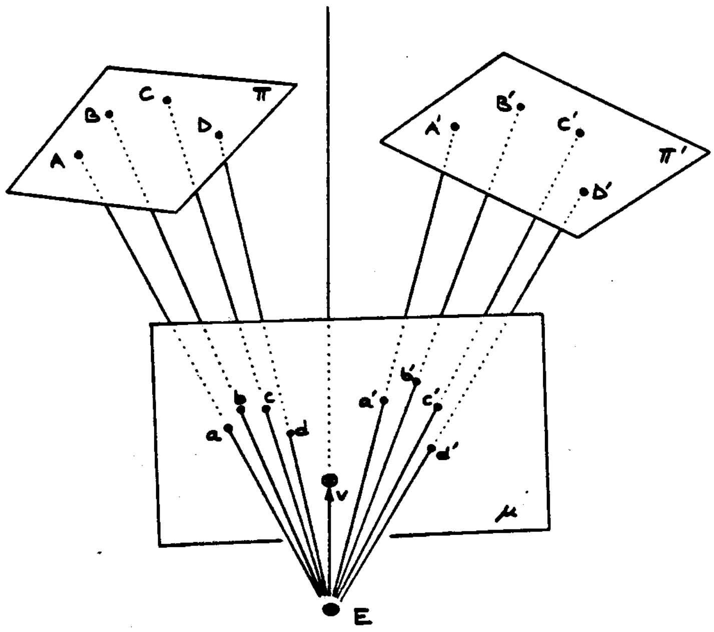
< figure 32 >

8.2. AN ALGORITM FOR COMPUTING THE 3D-POSITION OF A PLANE FROM THE "CENTRE-INFINITY" POLARITY OF A CENTRAL CONIC

It is easy to see (figure 13) that the centre point (**M**) of a central conic (e.g. an ellipse) and the line at infinity (**m**) have a pole-polar relationship - they are the polarized images of one another. This fact can be used to determine the 3D orientation of a plane π in which we are observing a central conic **C** with its centre-point **M**. Referring to figure 32, the polar line **m** of the point **M** with respect to **C** is the line at infinity in π. Since all pole-polar relationships are projectively invariant, if we polarize **M´** (our image of **M**) in the conic **C´** (our image of **C**) the resulting line **m´** must coincide with our image of the line of infinity (**m**) in π. Hence **m´** is identical to the "horizon-line" of our image of π, and if we compute the plane containing **m´** and our point of observation **O**, we have a plane parallell to π - and hence the orientation of π in space. Of course this method of orientation is highly noise-sensitive in the case of a single observation - especially if the deviation of **M´** from the centre-point of **C´** is small, which would be the case if we were observing the plane π almost "head-on" - i.e. in a direction close to the normal of π. But if we were to observe e.g. a plane field of oil-barrels (with their centre-plugs) from a direction that is "tilted enough" relative to the normal of the plane, both the statistics and the larger centre-deviation would contribute toward a more robust computational process.

8.3. AN ALGORITHM FOR RECONSTRUCTING THE 3D-MOTION OF A PLANE FROM TRACKING THE IMAGE OF FOUR OF ITS POINTS

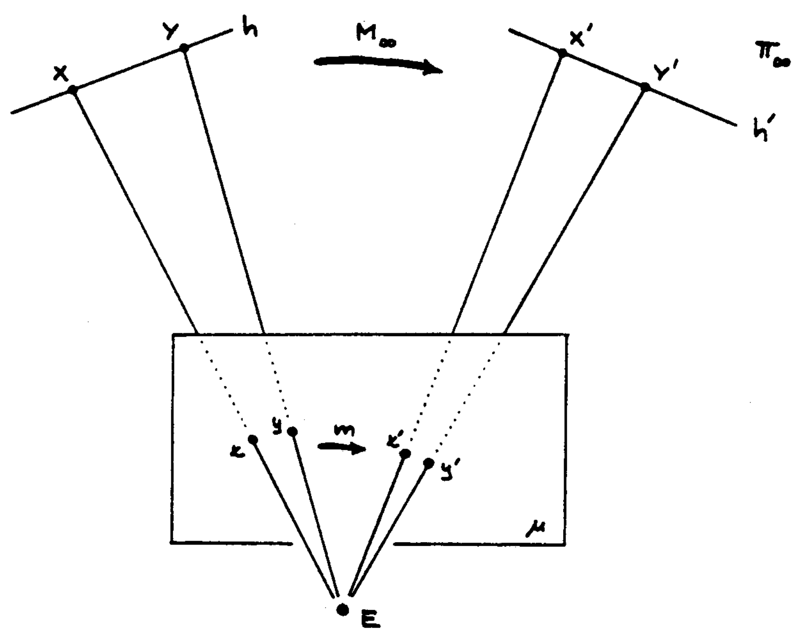
We will finish our list of "deductive examples" using projective techniques by describing a projective algorithm for determining the motion of a rigid planar patch from two matched images of four of its points. This is a well known problem in the field of image analysis, and it has been studied by several people using a variety of different approaches (see e.g. [3], [5] and [8]). The one that is used here is characterized by its explicit use of the sphere circle (see chapter 2) as a means to express the metric information present in the concept of "rigid motion" projectively. Hence it is another example of a "projective embedding" of a euclidean configuration.



< figure 33 >

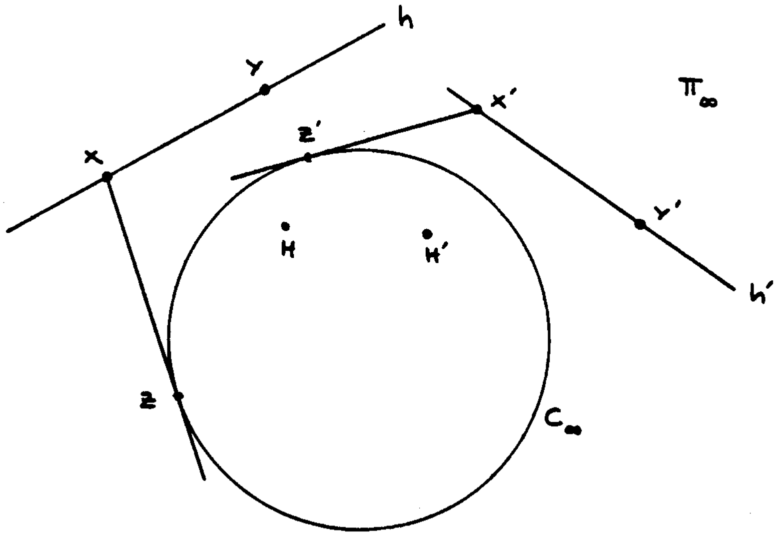
Mathematical formulation :

Consider a plane π in P^3 with four marked points **A**, **B**, **C** and **D**. Imagine that we are observing π through a "pinhole" camera - i.e. projecting the points in π from a fixed point **E** (the pinhole lense or the "eye") onto a fixed plane μ (the image plane or the "retina"). Let the images of **A**, **B**, **C** and **D** be **a**, **b**, **c** and **d** respectively (figure 33). Moving π rigidly (by subjecting it to the unknown motion **M**) to make it coincide with another plane π´ and observing the change of the image in μ of the marked points (now located at **A´**, **B´**, **C´** and **D´** in π´) induces a map **m** : μ --> μ taking **a** --> **a´**, **b** --> **b´** etc. It can be shown (see [5]) that **m** is a projectivity - and hence that its action can be expressed by a (non-singular 3x3) matrix operating on the (projective) coordinates of the points in μ. This matrix (which can be computed from "image data") has at least one real eigenvector **v** and this vector corresponds to a fixed point of the map **m**. The idea of the algorithm that we shall present is to use this fixed point to decompose the desired motion **M** into a rotation **R** around **v** (which also can be determined directly from image data) followed by an (unknown) translation **T** along **v**. This decomposition (**M** = **TR**) is valid because of a well known theorem of classical mechanics, and it leaves us with only one single parameter left to be determined - the "size" (or distance) of the translation along **v**.



< figure 34 >

This can be done by "observing the effects at infinity" of the motion **M**. Let the plane at infinity be called π_∞ and let the lines at infinity in π and π' be called **h** and **h**$'$ respectively.Since a motion is a special type of affine transformation it must leave π_∞ invariant as a whole. Hence **M** induces a projectivity $\mathbf{M}_\infty : \pi_\infty \dashrightarrow \pi_\infty$ taking **h** --> **h**$'$. It is easy to see that the translational distance that we are looking for can be determined from a complete knowledge of the map $\mathbf{M}_\infty$. By the fundamental theorem of projective geometry, $\mathbf{M}_\infty$ is determined by its action on four generic points (i.e. four points - no three of which are collinear). So far we only know the action of $\mathbf{M}_\infty$ on two such points - namely any two points on the line **h**. Choosing two such points **X** and **Y** (figure 34), their images (**X**$'$ and **Y**$'$) under $\mathbf{M}_\infty$ can be obtained by projecting them from **E** onto μ - hence obtaining the image points **x** and **y**, mapping these with **m** to **x**$'$ and **y**$'$ and finally projecting the latter points back again onto π_∞. Note that the points on **h** are the only points in π_∞ that can be mapped by $\mathbf{M}_\infty$ in this way - i.e. by transferring to the "image map" **m**.



< figure 35 >

To find the image under M_∞ of two more "independent" points we can make use of the sphere circle C_∞ in π_∞. Since **M** is a euclidean isometry it must transform the euclidean fundamental form (and hence the sphere circle) into itself. Choosing two points **X** and **Y** on the line **h** (with images **X´** and **Y´** on **h´**), we can therefore obtain the necessary information in the following way: Constructing an arbitrary tangent from **X** to C_∞ gives us a point of tangency **Z** (figure 35). Since **h** is real and C_∞ is purely imaginary, **h** cannot be tangent to C_∞ and therefore **Z** must be non-collinear with **X** and **Y**. Now, tangency is a projectively invariant condition and hence the image **Z´** of **Z** under M_∞ must be one of the two point of contact between C_∞ and its two tangent lines through **X´**. Which one to choose is decided by inspecting the resulting value of the translational distance - which is computed from an overdetermined linear system of equations. It turns out that choosing the wrong point of tangency for **Z´** will produce a value with a large residual while choosing the right point will produce a value with (almost) no residual at all. The final (fourth) point that is needed to gain complete control of the map M_∞ is provided by the polar point **H** of the line **h** with respect to C_∞. Since polarity is another projectively invariant relation, the image **H´** under M_∞ of the point **H** must be the polar point of the line **h´** with respect to C_∞. Referring to figure 35 it is obvious that we can choose our initial points **X** and **Y** so that none of them is collinear with **Z** (this can never happen as - we already know) or **H**. Hence we have determined the image under M_∞ of four generic points (**X**, **Y**, **Z** and **H**) and therefore we have the information we need to compute the required translational distance.

9. CONCLUDING DISCUSSION AND FUTURE WORK

In this paper we have presented a projective framework for representing geometric structure in a unified way. We have demonstrated how this representation can be exploited to embedd all kinds of geometric information against a "universal" background, and we have given a few algorithmic examples of the benefits that can be extracted from this point of view. We have also presented an interactive geometric "tool-box" - called Drawboard - that is built entirely on the projective representation, and we have illustrated how it can be used to promote the understanding of geometric phenomena in general - including the dynamic behaviour and the invariance properties of a geometric construction. Finally , we have demonstrated the "experimental mode" of using Drawboard in order to gain insight into geometric problems from other areas of mathematics. It is our opinion that the collection of these ideas make a strong case for the importance of the projective representation in the entire field of geometric modelling. The underlying reason for this is first of all that "everything can be expressed projectively" and secondly that "the expressions are valid without any exceptions whatsoever". In "ordinary" euclidean geometry there is the old familiar fact that "two lines in a plane have a common point **except** when they are parallel". This innocent looking exception creates "a mountain of combinatorial complexity" when the number of participating lines grows large. In projective geometry "two lines in a plane have a common point with **no exceptions**, and it is precisely the lack of such exceptions that gives projective geometry its great power of conceptual and representational unity.

Regarding future work, it is of course quite obvious that the implementation of Drawboard presented here is still in its infancy. The full 3D-version ("Drawspace") is an endeavour that involves difficulties of a different order of magnitude. Nevertheless, the benefits of having such a system would in our opinion more than compensate for the labours of constructing it. Just to mention one area of application, the CAD/CAM systems that could be built on top of a projective Drawspace system would be able to utilize its representational powers to implement many geometric features unheard of in the leading systems of today. One such potential feature well worth exploring is the construction of a decent (i.e. generic) "intersection-algorithm" for computing and geometrically representing the set-intersection of two general 3D geometrical bodies. This involves representational techniques from other areas of mathematics - including differential geometry and combinatorial topology - to generate a "mathematically decent" description of a general surface in 3D. It is our ambition to integrate such "classical gems" of pure mathematics into the 3D-expansion of the projective "world-view" discussed here - in order to exploit the representational and computational "synergy" of such a mixture in solving a variety of different geometric "engineering" problems.

10. LIST OF REFERENCES

[1] Klein, F.,
Vorlesungen über höhere Geometrie,
Springer, Berlin, Germany, 1926.

[2] Klein, F.,
Vorlesungen über nichteuklidische Geometrie,
Göttingen, Germany, 1926.

[3] Tsai, R. & Huang, T.,
Estimating Three-Dimensional Motion Parameters of a Rigid Planar Patch,
IEEE, Trans. ASSP, 30:4, 1982.

[4] Naeve, A.,
Projective Line Geometry of the Visual Operator,
TRITA-NA-8606, Department of Computer Science,
Royal Institute of Technology, Stockholm, Sweden, 1986.

[5] Svensson, L. & Naeve, A.,
Estimating the N-dimensional Motion of a (N-1)-dimensional Hyperplane from Two Matched Images of (N+1) of its Points,
TRITA-NA-P8708, Department of Computer Science,
Royal Institute of Technology, Stockholm, Sweden, 1987.

[6] Naeve, A. & Eklundh, J.O.,
On Projective Geometry and the Recovery of 3D-structure,
TRITA-NA-8608, Department of Computer Science,
Royal Institute of Technology, Stockholm, Sweden, 1986.

[7] Naeve, A.,
On the use of Exterior Algebra in Image Analysis,
TRITA-NA-P8709, Department of Computer Science,
Royal Institute of Technology, Stockholm, Sweden, 1987.

[8] ARTIFICIAL INTELLIGENCE
Vol 37 , Dec. 1988.

11. ACKNOWLEDGMENTS

It is a great pleasure for me to express my heart-felt thanks to all the people who have contributed to the Drawboard system in various ways. I am especially endebted to the following persons:

Jan-Olof Eklundh - for his never ending support in bringing my mathematical daydreaming to a computational fruition. Since he was the one that brought me into this field it is impossible for me to give him credit for every one of the individual ideas that he has contributed. Let me just say that over the years there has been quite a number!

Lars Svensson - for many stimulating discussions and penetrating insigths into the underlying philosophy of the system. They sure helped to clear up my thinking!

Johan Appelgren - for his generous and never tiring support of my programming efforts. Without his expert help, the system would probably never have developed past the theoretical stage at all!

Örjan Ekeberg - for lending me some of his "programming magic" to produce especially elegant and extendible code!

Mats Danielsson - for performing the "gargantuan task" of porting my computational Drawboard kernel (written in C) to the Macintosh - MPW - environment and making it collaborate with Object-Pascal!

My thanks must also include the entire "MacDrawboard group" of students:

Niklas Andersson	Magnus Berg
Mats Danielsson	Dennis Gyllensporre
Catrin Johnsson	Åke Järvklo
Tommy Levitte	Dorit Paltzer

for persevering to create the user-friendly application *MacDrawboard* on the Mac-II in spite of all the problems. It has turned out to be a powerful tool in the study of geometrical constructions with points and lines in $\mathbf{P}^2$.

Last (but not least!) I would like thank STU and STUF for financial support of this project.