
Representing geometrical knowledge

James A. D. W. Anderson

Phil. Trans. R. Soc. Lond. B 1997 **352**, 1129-1139
doi: 10.1098/rstb.1997.0096

Email alerting service

Receive free email alerts when new articles cite this article - sign up in the box at the top right-hand corner of the article or click [here](#)

To subscribe to *Phil. Trans. R. Soc. Lond. B* go to: <http://rstb.royalsocietypublishing.org/subscriptions>

Representing geometrical knowledge

JAMES A. D. W. ANDERSON

Computational Vision Group, Department of Computer Science, University of Reading, Reading RG6 6AY, UK
(james.anderson@reading.ac.uk)

SUMMARY

This paper introduces perspex algebra which is being developed as a common representation of geometrical knowledge. A perspex can currently be interpreted in one of four ways. First, the algebraic perspex is a generalization of matrices, it provides the most general representation for all of the interpretations of a perspex. The algebraic perspex can be used to describe arbitrary sets of coordinates. The remaining three interpretations of the perspex are all related to square matrices and operate in a Euclidean model of projective space-time, called perspex space. Perspex space differs from the usual Euclidean model of projective space in that it contains the point at nullity. It is argued that the point at nullity is necessary for a consistent account of perspective in top-down vision. Second, the geometric perspex is a simplex in perspex space. It can be used as a primitive building block for shapes, or as a way of recording landmarks on shapes. Third, the transformational perspex describes linear transformations in perspex space that provide the affine and perspective transformations in space-time. It can be used to match a prototype shape to an image, even in so called 'accidental' views where the depth of an object disappears from view, or an object stays in the same place across time. Fourth, the parametric perspex describes the geometric and transformational perspexes in terms of parameters that are related to everyday English descriptions. The parametric perspex can be used to obtain both continuous and categorical perception of objects. The paper ends with a discussion of issues related to using a perspex to describe logic.

1. INTRODUCTION

So far as is possible, this paper sets out to use conventional projective geometry to model the appearance of objects in images, so that computer vision programs can see objects. This is entirely possible in bottom-up vision, where a program explains only the parts of an object that are physically present in an image, but it is not possible in top-down vision where a program brings to the task knowledge of the geometrical structure of objects that might appear in an image. In this case, the program might predict the occurrence of object points, lines, or surfaces that fall exactly in the same part of the image as some other predicted point, line, or surface, respectively. This seemingly innocuous circumstance—the occurrence of non-distinct points, lines or surfaces—seriously undermines the application of projective geometry to computer vision. The solution offered here is to add the 'point at nullity' to projective geometry to deal with such cases. Thus projective geometry, which was obtained from Euclidean geometry by adding the 'line/plane/hyperplane at infinity' is now made complete by adding the 'point at nullity'. This approach is controversial.

In conventional projective geometry, a projective space of some dimension is modelled by a Euclidean space of one more dimension, which can be described in an augmented system of coordinates called 'homogeneous coordinates'. However, it is an axiom of this

mathematical model that the zero vector—the point at nullity—is not included. The point at nullity is said to be punctured from the space. In addition to dealing with space, this paper also deals with time by the parsimonious solution of adding a Euclidean time axis to the spatial axes, giving rise to a Euclidean space-time. The projective space, which has one more dimension, and is made complete by including the point at nullity, is called perspex space because it describes the whole space of coordinates that a perspex can lie in.

An algebraic perspex is introduced as a generalization of arbitrary matrices that can be used to describe all perspexes. The other three perspexes introduced here are the geometric, transformational, and parametric perspexes—all of which are related to square matrices. A perspex derives its name from the loose phrase 'perspective simplex'. The geometric perspex is a simplex in the augmented Euclidean space that contains the point at nullity, now called 'perspex space'. In our terms, a geometric perspex is the simplest straight-edged shape that can contain a volume of space-time. In three and two Euclidean dimensions, a perspex is a tetrahedron and a triangle, respectively. Perspex algebra operates in all whole-numbered dimensions and has a very convenient way of switching dimensions on and off by using j -numbers. This means that the transformational perspex can describe transformations in space-time, in just space, or in any discrete dimensions whatever. The

transformational perspex provides the linear transformations of perspex space which can be manifest as the general linear (affine) and perspective transformations of space-time.

The j -numbers, which hide and reveal dimensions, are intimately related to the j -inverse which annihilates a perspex by hiding dimensions, until what remains can be inverted in the sense of matrix algebra. The j -inverse has a property which can easily be misunderstood. It provides the unique inverse of a matrix that has been annihilated according to the assumptions built into the j -inverse. Thus the j -inverse picks out one, canonical inverse as a representative of all possible generalized inverses. If the j -inverse is applied to a non-singular matrix, then it returns the matrix inverse which is the only possible, hence unique, inverse.

The parametric perspex describes any square perspex in terms of easily understood parameters. The space-time parameters are magnitude, handedness, shear, rotation, and translation. Perspex space has additional variability which can be described by scale factors appropriate to the geometric perspex, or by focal lengths appropriate to the transformational perspex. These two natural kinds of parameter are different, though both kinds of perspex can be understood in terms of either kind of parameter. It is hoped that in future a better resolution of these two parameterizations will be found, perhaps by adopting just one focal length and scale factors in the remaining degrees of freedom.

Perspexes have the property that any matrix can be represented as a perspex. This means, in particular, that a perspex can be interpreted as if it were a geometric, transformational, or parametric perspex. It also means that any operation, whatever, may be applied to a perspex and the result will be meaningful in all three interpretations. This would make it feasible for a program to search randomly for a network of interrelated perspexes that identify objects in images, store geometrical and visual knowledge, and solve visual and geometrical problems. Of course, the search for perspex structure need not be random, there are certain operations and combinations of operations which always retain a simple meaning. This opens up the possibility of providing neural nets of perspexes with sufficient internal structure to get them started on the long road of visual learning. Alternatively, it offers the possibility of designing and implementing very sophisticated symbolic programs.

Throughout this paper mathematical formalism is kept to a minimum and the general reader is advised on which sections may be omitted without risk of losing sight of the overall discussion. However, nothing in this paper can be understood without an appreciation of the algebraic perspex and the j -inverse.

2. ALGEBRAIC PERSPEX

A perspex is a triple $\langle \mathcal{J}_l, A_{mn}, \mathcal{J}_r \rangle$. The middle term, A_{mn} , is a matrix with m rows and n columns. The matrix may contain real or complex numbers, but only real matrices are used in this paper. In a departure

from matrix algebra the matrix part of a perspex may contain no rows and no columns. This matrix is defined to contain one element, zero, and may be written as A_{00} or, simply, 0.

Matrix algebra has an identity matrix I that does not alter any matrix which is multiplied by I . The matrix I is a diagonal matrix that has ones everywhere on the major diagonal and zeros everywhere else. Thus the two-dimensional matrix I is

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

By contrast the matrix \mathcal{J} is defined to be a diagonal matrix which may contain zeros or ones on the major diagonal. Thus

$$\mathcal{J}_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathcal{J}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \mathcal{J}_1 = [1] \quad \mathcal{J}_0 = [0].$$

The pattern of zeros and ones on the major diagonal, reading from the least significant bit at top-left to the most significant bit at bottom-right, is defined to be the binary encoding of a j -number that is usually written in decimal notation. The j -number is shown as a subscript to the letter \mathcal{J} . Thus \mathcal{J}_3 has 11 on the major diagonal and $11 \rightarrow 1 \times 2^0 + 1 \times 2^1 = 3$. Similarly, \mathcal{J}_2 has 01 on the major diagonal and $01 \rightarrow 0 \times 2^0 + 1 \times 2^1 = 2$. Every non-negative binary number describes a j -matrix.

In a perspex $\langle \mathcal{J}_l, A_{mn}, \mathcal{J}_r \rangle$, the term \mathcal{J}_l is the left j -number, and \mathcal{J}_r is the right j -number. The j -numbers are used to zero whole rows and columns of a matrix. Where a zero occurs in the binary encoding of the left j -number the corresponding row is zeroed; where a zero occurs in the binary encoding of the right j -number the corresponding column is zeroed. Thus

$$\langle 5, A, 10 \rangle = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \\ \times \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & a_{12} & 0 & a_{14} \\ 0 & 0 & 0 & 0 \\ 0 & a_{32} & 0 & a_{34} \end{bmatrix}.$$

Notice that there is no need to store the rows and columns that the j -numbers zero, so a concise encoding is

$$\langle 5, A, 10 \rangle = \left\langle 5, \begin{bmatrix} a_{12} & a_{14} \\ a_{32} & a_{34} \end{bmatrix}, 10 \right\rangle.$$

However, it is often useful not to discard the data in the matrix part of a perspex, so that operations on perspexes can be undone, or so that rows and columns of the matrix can be hidden during an operation and revealed later. Therefore, it is defined that no operation in perspex algebra discards data zeroed by the j -numbers, except for the explicit operation 'contract (P) $\rightarrow P'$, where P is a perspex.

The reader who prefers to skip the following technical example and subsequent comments need only

note that all perspexes can be multiplied together regardless of their dimensions.

A technical example of hiding and revealing parts of a matrix is to consider the calculation of a perspective view in space-time. The conventional way of doing this using homogeneous coordinates (Foley *et al.* 1987; Riesenfeld 1981), would apply perspective to the time axis, resulting in an image inverted in time. This presents no mathematical problems, but it would be more natural to require that time is projected without modification (orthogonally) regardless of the perspective applied to space.

Consider a point $[xyztw]^T$, with coordinates (x, y, z) on the spatial axes, coordinate t on the time axis, and conventional coordinate $w = 1$ on the augmenting axis. Then a thin-lens camera, with optical axis conventionally along the z -axis, would apply a perspective transformation with focal ratio $-1/f$, for a non-zero focal length f . But in constructing the transformation the time axis is hidden by setting the fourth bit of the left j -number zero. Thus, with j -numbers shown in base ten:

$$\left\langle 23, \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1/f & 0 & 1 \end{bmatrix} \right\rangle, 31 \left\langle \begin{bmatrix} x \\ y \\ z \\ t \\ 1 \end{bmatrix} \right\rangle$$

$$\rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/f & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ 1 - z/f \end{bmatrix}.$$

Dividing throughout by the w -coordinate, if non-zero, gives a geometrically correct perspective view in space with a redundant, conventional coordinate $w = 1$:

$$\begin{bmatrix} x \\ y \\ z \\ 1 - z/f \end{bmatrix} \rightarrow \begin{bmatrix} x/(1 - z/f) \\ y/(1 - z/f) \\ z/(1 - z/f) \\ (1 - z/f)/(1 - z/f) \end{bmatrix} = \begin{bmatrix} fx/(f - z) \\ fy/(f - z) \\ fz/(f - z) \\ 1 \end{bmatrix}.$$

Setting the fourth bit of the left j -number to unity then restores the unmodified time coordinate as required:

$$\left\langle 23, \begin{bmatrix} fx/(f - z) \\ fy/(f - z) \\ fz/(f - z) \\ 1 \end{bmatrix} \right\rangle \rightarrow \left\langle 31, \begin{bmatrix} fx/(f - z) \\ fy/(f - z) \\ fz/(f - z) \\ t \\ 1 \end{bmatrix} \right\rangle.$$

One very important consequence of j -numbers is that all perspexes can be multiplied together, because matrices can be augmented with zeroed rows or columns to make them the right size to be conformable for multiplication. The addition of zeroed rows and columns has negligible effect on the speed of a computer algorithm to perform the multiplication, because these nominal rows and columns are skipped. This does not involve a deletion of information in the perspex arguments, so in the following example it is acceptable not to record the entirely redundant third row of zeros in the solution. Thus

$$\langle 2, A, 2 \rangle \langle 3, B, 3 \rangle = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} =$$

$$\left\langle 2, \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} & a_{11}b_{13} + a_{12}b_{23} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} & a_{21}b_{13} + a_{22}b_{23} \end{bmatrix}, 3 \right\rangle.$$

This conformability of perspexes, extended to all matrix operations, has the extremely important consequence that a computer program that uses perspexes instead of matrices need not branch to special cases to deal with data structures of mixed dimensionality—as occurs very often in vision where, for example, three-dimensional space is projected over time onto a two-dimensional retina in time.

In the section on the j -inverse this advantage is carried further. A program using perspexes need not branch in the face of singularities which reduce the effective dimensionality (rank) of the data. This remains true even in the extreme case of zero rank where no data is visible in the image, the correspondence is then given by the null perspex $\langle 0, 0, 0 \rangle$, which describes the point at nullity.

3. J -INVERSE

The presentation of the j -inverse in this section is technical, but the numerical example might be more generally accessible. The j -inverse (Anderson 1996) is a generalized inverse (Ben-Israel & Greville 1974; Nashed 1976) that annihilates rows and columns of a matrix until a matrix inverse can be found.

Given a perspex $\langle \mathcal{J}_l, A_{mn}, \mathcal{J}_r \rangle$, the j -inverse, denoted by the superscript ' $-j$ ', resets the j -numbers in the perspex it is applied to so that the first d linearly independent rows and columns are preserved, the remainder being zeroed, where d is the rank of the matrix. The perspex inverse is computed by interchanging the left and right j -numbers and setting the matrix part of the perspex inverse to the matrix inverse if $d \neq 0$, or setting the matrix part to 0 if $d = 0$.

In general,

$$\langle \mathcal{J}_l, A_{mn}, \mathcal{J}_r \rangle^{-j} = \langle \mathcal{J}_a, A_{dd}, \mathcal{J}_b \rangle^{-j} = \langle \mathcal{J}_b, A_{dd}^{-1}, \mathcal{J}_a \rangle,$$

giving right and left inverses:

$$\langle \mathcal{J}_a, A_{dd}, \mathcal{J}_b \rangle \langle \mathcal{J}_b, A_{dd}^{-1}, \mathcal{J}_a \rangle = \langle \mathcal{J}_a, I_d, \mathcal{J}_a \rangle,$$

$$\langle \mathcal{J}_b, A_{dd}^{-1}, \mathcal{J}_a \rangle \langle \mathcal{J}_a, A_{dd}, \mathcal{J}_b \rangle = \langle \mathcal{J}_b, I_d, \mathcal{J}_b \rangle.$$

Notice that when $\mathcal{J}_a = \mathcal{J}_b$ the left and right inverses are identical. This is the case, for example, if the original matrix is square and of full rank, that is, non-singular. Hence, formulas in matrix algebra may be rewritten in perspex algebra with the same result if the original matrix is non-singular, but a useful result if singular. This is vitally important. It means that a program using perspex algebra need not branch to different cases even in the face of singularities.

A numerical example might help to clarify things. Given:

$$\begin{aligned} \left\langle 7, \begin{bmatrix} 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 4 \end{bmatrix}, 15 \right\rangle^{-j} &\rightarrow \left\langle 5, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 10 \right\rangle^{-j} \\ &= \left\langle 10, \begin{bmatrix} -2 & 1 \\ 3/2 & -1/2 \end{bmatrix}, 5 \right\rangle \end{aligned}$$

the right j -inverse is

$$\begin{aligned} \left\langle 5, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 10 \right\rangle \left\langle 10, \begin{bmatrix} -2 & 1 \\ 3/2 & -1/2 \end{bmatrix}, 5 \right\rangle &\rightarrow \\ \begin{bmatrix} 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 4 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ -2 & 0 & 1 \\ 0 & 0 & 0 \\ 3/2 & 0 & -1/2 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathcal{J}_j \end{aligned}$$

and the left j -inverse is

$$\begin{aligned} \left\langle 10, \begin{bmatrix} -2 & 1 \\ 3/2 & -1/2 \end{bmatrix}, 5 \right\rangle \left\langle 5, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 10 \right\rangle &\rightarrow \\ \begin{bmatrix} 0 & 0 & 0 \\ -2 & 0 & 1 \\ 0 & 0 & 0 \\ 3/2 & 0 & -1/2 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 4 \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \mathcal{J}_{10}. \end{aligned}$$

Note that the matrix \mathcal{J}_a is described by the perspex $\langle \mathcal{J}_a, I, \mathcal{J}_a \rangle$ so $\mathcal{J}_a \rightarrow (0, 0, 0)$ describes the point at nullity.

4. POINT AT NULLITY

Perspex algebra is intended to be used by an unsupervised computer program, especially one controlling a robot, which seeks to understand the shape and disposition of objects in space and how they change in time. This raises a problem which was not envisaged when projective geometry was first formalized during the 15th century in Renaissance Italy (Boyer & Merzbach 1989). At that time the problem was to describe how an image should be drawn in perspective on a canvas. It was of no importance that two objects might fall at the same place in the image. In such circumstances an artist can see just the physically nearer object and draws this. The assumption that projective geometry deals with separate, so called 'distinct' points, lines, and surfaces was formalized in the projective geometries of the 17th, 18th, and 19th centuries (Boyer & Merzbach 1989) and survives to this day. Incidentally, Leonardo da Vinci's and Johannes Kepler's theories of projection through an aperture are described in (Lindberg & Cantor 1985).

A robot which has a bottom-up visual sense is in exactly the same position as a Renaissance artist: non-distinct lines are not physically possible in an image, so classical projective geometry is entirely adequate. However, when a robot seeks to match a geometrical

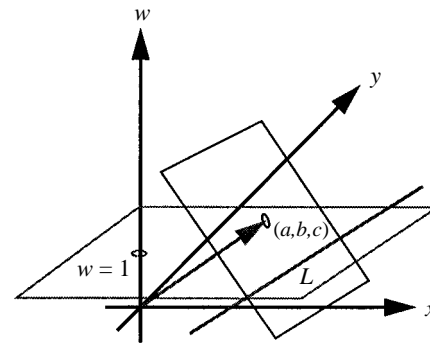


Figure 1. Line-point duality.

model of an object in its database to an image of an object in the world it might very well happen that, say, two lines on the model line up in the image, becoming non-distinct. This seemingly innocuous circumstance seriously undermines the application of projective geometry to computer vision.

The problem of finding the intersection of two lines occurs very commonly in computer vision. For example, the simple task of finding the part of a polyhedral model that is visible in an image involves cutting the lines of the model at the edges of the image. It is of course desirable that the calculation of the intersection of two lines should be consistent with projective geometry, but this is not the case for non-distinct lines.

The remainder of this section is highly technical. Figure 1 shows the plane model of homogeneous coordinates (Greenberg 1994; Riesenfeld 1981; Stoffli 1991). In the vocabulary of this paper, this is a model of perspex space with the plane at $w = 1$ corresponding to space-time. The position vector (a, b, c) touches and is perpendicular to a plane that intersects the $w = 1$ plane in the line L . Thus, a line L in space-time can be described by a point in perspex space, the position vector (a, b, c) . The converse is also true. The point (a, b, c) in perspex space, can be described by the line L in space-time, which occurs at the intersection of the $w = 1$ plane with the plane that is normal to and touches the position vector (a, b, c) . This is an example of line-point duality. Suppose, as is commonly the case in computer vision, that it is desired to find the intersection between two lines L_1 and L_2 , described by their dual points (a_1, b_1, c_1) and (a_2, b_2, c_2) . By construction L_1 lies in a plane perpendicular to (a_1, b_1, c_1) and L_2 lies in a plane perpendicular to (a_2, b_2, c_2) . So the point which is the intersection of L_1 and L_2 lies in a plane which is perpendicular to both (a_1, b_1, c_1) and (a_2, b_2, c_2) . In other words, the intersection lies in a plane whose defining vector is the cross product of (a_1, b_1, c_1) and (a_2, b_2, c_2) . The cross product produces a result which is satisfactory for classical projective geometry except in one case—where $L_1 = L_2$, that is, where $(a_1, b_1, c_1) = (a_2, b_2, c_2)$. In this case the cross product is $(a, b, c) \times (a, b, c) = (bc - bc, ac - ac, ab - ab) = (0, 0, 0)$. This is the zero vector which lies at the point at nullity and is excluded from the augmented Euclidean model of classical projective geometry by axiom. In other words, classical projective geometry cannot find the intersection of two non-distinct lines. This result

generalizes, by line-point duality, to say that classical projective geometry cannot find the intersection of two non-distinct points, and then generalizes to planes by non-classical dualities or by geometrical construction. Thereafter, it generalizes to every conceivable geometrical figure.

So if the reader wishes to use projective geometry, but also wishes to allow the possibility that points, lines, or planes might overlap in an image, becoming non-distinct, then the point at nullity enters geometrical constructions. But then the reader is caught in a contradictory position, because the point at nullity is not part of classical projective space. The reader must either give up classical projective geometry, or give up non-distinctness, or adopt a resolution of the problem such as that offered next.

Figure 2 shows the spherical model of classical projective space (Greenberg 1994; Riesenfeld 1981; Stolfi 1991) with the point at nullity added at the origin of the sphere, here a circle of radius $\lambda = 1$. In the vocabulary of this paper space-time is shown as the hatched hyperplane tangent to the sphere and perspex space is the whole space of the figure.

It is important to note that classical projective space, and indeed perspex space, is unorientable, so the spatial concepts 'left/right', 'above/below', 'forward/back', 'inside/outside' and the temporal concept 'before/after' cannot be defined, except in the bottom-up case where these are resolved on an image, or in the case of full-rank simplexes (Stolfi 1991) which would require direct perception of perspex space.

Each of these concepts is usually defined with respect to a directed line along which two points are marked for comparison. Thus a scale is obtained on which relationships like 'left-of/right-of' and 'before/after' can be defined. The difficulty is that in classical projective geometry there are two distinct straight lines that connect any two distinct points in space-time and these lines run in opposite directions—so that no scale can be constructed. This difficulty can be finessed in the case of full-rank simplexes (Stolfi 1991), but this would require a robot to have direct perception of perspex space and would deny it the possibility of using lower dimensional objects, say, lines and points in a four-dimensional space-time.

It is worth seeing how these two lines arise, because the construction shows the critical role played by the

point at nullity and hence relates classical projective geometry to perspex geometry.

Cartesian coordinates in space-time are obtained by drawing a straight line through the point at nullity, the origin of the sphere at $\lambda = 1$, and any other point in perspex space. The intersection of this line with the hyperplane at $\lambda = 1$, defines a point in perspex space whose non- λ coordinates are the Cartesian coordinates in space-time. For example, a point A , in the hyperplane corresponding to space-time, is defined by any and all points on the line aa' excluding the point at nullity. If, contrary to hypothesis, both end points were at the point at nullity the line would have no length, would not intersect the hyperplane, and would lie exactly at the point at nullity. This degeneracy is considered monstrous in classical projective geometry and is barred by the topological property that classical projective space does not contain the point at nullity. However, this degeneracy is essential to computer vision as the previous example of the intersection of non-distinct lines showed.

Now consider the problem of drawing a segment of a straight line between two distinct points A and B in space-time. One straight line in space-time is the projection of all of the diameters of the great circle that lie between a and b , these also pass between a' and b' . This line is called the *inner* line segment and corresponds to the everyday choice of the line that connects A and B in space-time. The second straight line is the projection of all of the diameters of the great circle between a and b' , these also pass between a' and b . This line is called the *outer* line segment and corresponds to a rather unintuitive line in space-time that starts at A , moves off in a straight line to a point at minus infinity, instantly jumps to a point at plus infinity, and moves back in a straight line to B . Thus, it completes a straight line from A to B . This property of the double connectivity of points in projective space, along with the unorientability of projective space, is well known in mathematics and computer graphics. The problem is solved in computer graphics by a process known as *clipping*, which systematically clips off the outer-line segments between points in space-time (Foley *et al.* 1987; Riesenfeld 1981; Stolfi 1991), thus restoring orientability. So if clipping is used in classical projective geometry, or in perspex geometry, then the concepts 'left/right', 'above/below', 'forward/back', 'inside/outside' and 'before/after' can all be defined. This is a happy consequence of clipping, but it does not help solve the problem of the intersection of non-distinct lines.

Consider, first, the intersection of two distinct lines in space-time which can be handled by classical projective geometry. These lines back-project onto two distinct great circles of the sphere, which define two sets of diameters of the sphere. The intersection of these two sets is a single diameter whose end points on the sphere are the intersections of the two great circles. Classical projective geometry does not contain the point at nullity, so these points on the sphere are the only points in consideration. These points project onto a single point in space-time which is the intersection of the two distinct lines in space-time.

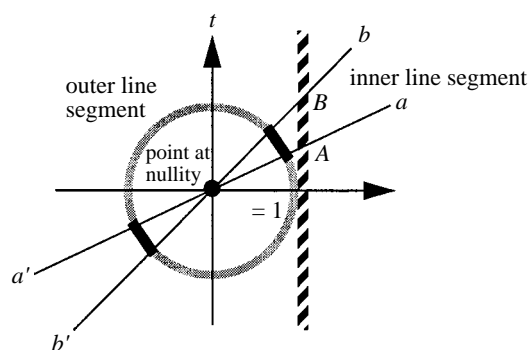


Figure 2. Spherical model of projective space.

Now consider the intersection of two non-distinct lines in space-time which cannot be handled by classical projective geometry without introducing serious difficulties. The two non-distinct lines back-project onto the sphere and define two identical sets of diameters of the sphere. The intersection of these sets is the original set, which defines a great circle of the sphere which projects onto the original line in space-time. So far, so good for classical projective geometry. But, the duality theorems of classical projective geometry (Greenberg 1994; Harris 1992; Stolfi 1991), require that the intersection of two lines is a point, and the result just obtained cannot be represented by anything other than a line in classical projective geometry. This need not be a difficulty. In perspex geometry the classical operations of projective geometry are redefined in terms of operations on diameters of the spherical model of projective geometry including the point at nullity. This always gives rise to a well formed solution, because set theory is consistent. The solution is identical to the classical case when distinct objects in the sphere are considered, but generally has a different dimensionality when non-distinct objects are considered. In the case of two non-distinct lines, just discussed, the intersection defined in perspex geometry is a line not some monstrous point as would be required in classical projective geometry. If this distinction in the dimensionality of objects is to be maintained then perspexes must be given a type defining the dimensionality of the objects they describe.

The point at nullity can arise in two ways. First, by geometrical construction, say as the intersection of two distinct diameters. Second, the point at nullity can arise as a classical error state, as in the cross-product algorithm for computing the intersection of two lines. Here the correct solution in perspex algebra is to return the intersection point, typed as a point, if non-nullity, but to return either of the arguments, typed as a line, if the algorithm computes the intersection point as nullity. Thus all of the results and intermediate results in perspex geometry can be described in the spherical model of projective space including the point at nullity and can therefore be given a geometrical interpretation.

Classical projective geometry deals with distinct geometrical entities on the sphere of the spherical model and is a proper subset of perspex geometry which admits non-distinct geometrical entities in the sphere and at the point at nullity. So, the reader has nothing to lose by adopting perspex geometry.

5. GEOMETRICAL PERSPEX

A few technical remarks need to be made about the geometrical perspex, before practical applications are

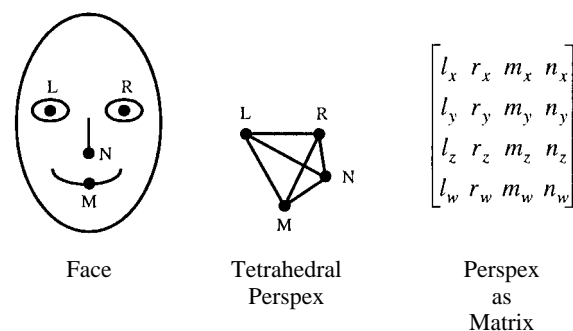


Figure 3. Perspex description of Pentland's face normalization.

discussed. First, the geometrical perspex is defined to be a square matrix of arbitrary homogeneous coordinates. It may, therefore, contain the zero vector—the point at nullity. Second, the geometrical perspex is put into canonical form by dividing vectors throughout by the absolute value of their w -coordinate, if non-zero. This is a departure from the conventional mathematical scheme (Riesenfeld 1981), and from a computer graphics scheme (Vanarsdale 1994), but it has the advantage that it retains all of the information needed for clipping (Foley *et al.* 1987), and does not alter a transformational perspex that describes a linear transformation in space-time. Pentland (1997) describes a human face recognition system, an early stage of which is to find the left and right eye (L, R), nose (N), and centre of the mouth (M). See figure 3. These landmarks are used to define a transformation to bring the picture of the head into a standard view. But the four points in homogeneous coordinates, corresponding to four points in the three-dimensional Euclidean space of a human head, describe a tetrahedron—a perspex. Similarly, any triangle composed of three of these four points defines a tetrahedron with one dimension annihilated, a triangle, which is also a perspex. Pentland's program takes this two-dimensional route, which establishes the use of a perspex in image processing (Garrett & Anderson 1995).

The tetrahedron has the advantage that it describes the well known Y-junction (tetrahedral junction) first used in the Blocksworld programs which are reviewed in (Boden 1977). The use of j -matrices allows a tetrahedral perspex to have parts annihilated so that, for example, it lies flat in the plane, turns into a T-junction, an L-junction, a line, a point, or nothing at all. Thus a single tetrahedral perspex can be matched to any of these configurations (figure 4).

The tetrahedral perspex is a good candidate for three-dimensional vision where the sensor provides depth information directly or computes it from several

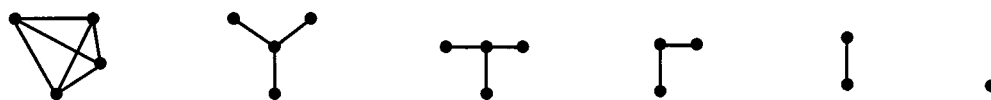


Figure 4. Views of a tetrahedral perspex and some of its annihilations.

views using, for example, Longuet-Higgins four-point algorithm, reviewed in (Marr 1982). But the tetrahedron does not contain enough information to recover an unannihilated transformation from a single view using methods related to Roberts's method (Roberts 1965).

Roberts reported the first computer vision program to use geometrical models of objects described in homogeneous coordinates (Roberts 1965). His method was to find points in an image that correspond to points on the model and to express this relationship as a simultaneous equation. In order to do this Roberts assumed that the focal length of the camera is known and that objects occur at a standard distance. The method is explained further in (Ballard & Brown 1982). A minor improvement is made in (Anderson 1992a) so that the transformational matrix recovered can be corrected by matrix multiplication if, and when, the true size or distance of the object is obtained.

Figure 5 is taken from (Anderson 1996) which contains a worked example of Roberts's method using the j -inverse. Given the assumptions of known focal length, standard distance, and a set of corresponding points on the model and image, the unique transformation of the model from (a) to (b) is found using either the j -inverse, or a standard matrix inverse as in Roberts's work. The standard matrix inverse fails in cases (c) and (d) because the matrices become singular. Roberts does discuss the possibility of using an *ad hoc* method to obtain solutions when it is known which of cases (c) and (d) obtains, but this proposal is unlikely to lead to a general, programmable algorithm. By contrast, the j -inverse directly computes the j -matrices that are needed to annihilate parts of the model in (a) so that the remaining parts can be transformed to match the image. Thus Roberts's *ad hoc* method, requiring human intervention, or detailed programming of every case, or approximation using singular value decomposition (SVD; Golub & van Loan 1983), is replaced by the j -inverse that computes a numerically exact answer for any matrix.

In the above example a cube was modelled by an eight-dimensional perspex with one dimension for each vertex. The perspex was described by an algebraic perspex corresponding to a matrix and the j -inverse deleted the unnecessary parts of the model in order to bring it into correspondence with the image.

A more compact representation of the visible parts of a cube would be two tetrahedral perspexes with one

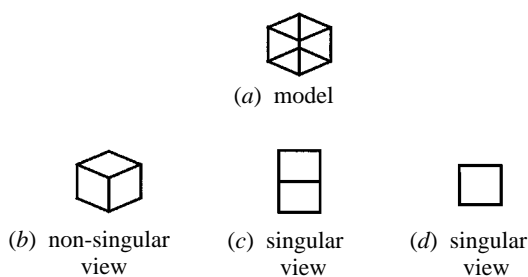


Figure 5. General and accidental views of a cube.

Phil. Trans. R. Soc. Lond. B (1997)

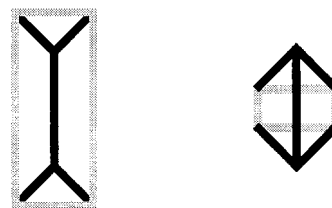


Figure 6. The triangular-prism perspex gives rise to Müller-Lyer figures.

point coincident, so that the perspexes describe a triangular prism. This would then form a primitive shape that could be used in Blockworld vision programs. In extreme annihilations the triangular-prism perspex would describe all of the geometrical forms in figure 4. In perspective views it would also describes the figures that arise in the Müller-Lyer illusion, figure 6. Coincidentally it is the same shape as the winged-edge polyhedron (Baumgart 1975) which is used in all modern CAD systems to describe topologically correct polyhedral models (Mäntylä 1988). Blockworld might well repay renewed examination using perspexes.

6. TRANSFORMATIONAL PERSPEX

A few technical remarks need to be made about the transformational perspex. First, it is defined to be a square matrix of arbitrary homogeneous coordinates. It may, therefore contain the zero vector, the point at nullity. Second, applying the j -inverse twice to the matrix part of the perspex, excluding the last row and column, isolates the first linearly independent rows and columns of the perspex that correspond to linear transformations of space-time. The linearly dependent rows and columns are zeroed as recorded in the j -numbers. This puts the linear part of the transformation into a canonical form. The whole matrix is put into a canonical form by dividing throughout by the parameter describing the scale of the augmenting, w -axis, if this is non-zero. The advantage of using the parameter, rather than just the element of the matrix in the last row and column is that this element can be zero, even when the w -axis has a well defined scale. In any case, this is a superset of the usual definition, so nothing is lost by this device.

7. PARAMETRIC PERSPEX

There are very many ways to parameterize the affine and perspective transformations. Some of these are reviewed in (Anderson 1992a) and a parameterization generalized to all dimensions appears in (Vanarsdale 1994). However, only Anderson (1992a,b, 1997; see matrix below), appears to offer a parameterization which preserves a one-to-one mapping between parameters and non-singular transformations. A parametric perspex is defined to be an arbitrary square matrix, so the parameterization must be extended to deal with linearly dependent rows and

columns and, in particular, it must allow the zero vector—the point at nullity.

The parameterization shown below is based on the Givens' orthogonalization (Golub & van Loan 1983), which is a well-known numerical technique for describing a matrix as the product of an orthogonal matrix, describing a rotation and/or reflection, and an upper triangular matrix:

$$\begin{bmatrix} h_1 m_1 & s_1 & s_2 & t_1 \\ r_1^+ & h_2 m_2 & s_3 & t_2 \\ r_2 & r_3 & h_3 m_3 & t_3 \\ k_1 & k_2 & k_3 & k_4 \end{bmatrix}.$$

This leads to a natural set of parameters. The r_i terms are the parameters of plane (two-dimensional) rotations in each combination of two coordinate axes—shown by the row and column in which the parameter appears. These terms describe orientations of objects. Though, in a departure from the Givens algorithm and previously published modification of it, the r_i terms should be taken with alternating sign, so that they preserve the right-hand rule of three-dimensional rotations and its generalization to higher dimensions (Vanarsdale 1994). In addition, no constraint applies between the rotation parameters as was mistakenly reported earlier.

The m_i terms are strictly positive magnitudes that describe lengths of objects along each coordinate axis. The t_i terms describe translations, that is, changes of position along each of the coordinate axes.

The s_i terms describe how an object is sheared in each of the corresponding rotational planes. Shears have been misdescribed in the general computer graphics and computer vision literature, but they are the transformations which transform a rectangle into a parallelogram (Anderson 1992*a,b*). The term h_i , with highest index, here h_3 , is a parameter ± 1 that changes the handedness of an object; it describes reflection. The remaining terms and k_i , are a little more abstract.

All of the terms h_i , except the last one, modify the last rotation term in their column. Here, h_1 modifies r_2 and h_2 modifies r_3 . The modifications arise as follows. In order to preserve a one-to-one mapping between rotation parameters and rotational transformations, only the r_i terms in the last row describe a whole rotation, the others describe a semi-rotation. So the h_i parameters, all of which are ± 1 , are used to carry the semi-rotations in the last row of rotational parameters into transformations describing a whole rotation. In this way, all of the rotation parameters take on the same range, as do all of the handedness parameters. This is a natural interpretation. The rotations themselves are encoded as tangents. Tangents have the advantage that they cover the whole of the real number range.

The k_i terms described here are simply parameters describing the scale of the w -axes in the coordinate interpretation of perspexes. Alternatively the w -partition could be parameterized as focal lengths, which would be a natural way to parameterize the transformational interpretation of perspexes. It remains to be

seen which of these should be regarded as the primary encoding from which the other is derived.

The parameterization just described is extended to cover arbitrary square matrices by introducing a canonical form. Specifically, a square matrix is parameterized by first putting the matrix in the canonical form of a transformational perspex. The parametric perspex is then in one-to-one correspondence with the canonical transformational perspex. All of the parameters in a parametric perspex may take on any real numbered value, except the magnitude components. If a parametric perspex has any zero magnitudes then it is unparameterized to give a singular transformational perspex, which is then put into the canonical form of a transformational perspex and is reparameterized. Thus every square matrix corresponds to some parametric perspex.

The important properties of the parametric description are that it is in 'natural' terms and provides a one-to-one mapping with the matrix part of a non-singular transformational perspex as may be recovered by the j -inverse.

These properties were exploited in (Byne *et al.* 1996) with the results shown in figure 7. A robot was used to obtain a picture of some children's wooden blocks. The robot was calibrated so that the distance from the camera to the work bench and the focal lengths of the camera along each of the horizontal and vertical array of pixels was known. In addition the correspondence was given between points on an eight-dimensional perspex and points found in the image by a model-based vision program. The blocks were processed sequentially. For each block the j -inverse was used to obtain the transformation of a unit cube to the block. The transformation was then parameterized. The parameters were then summed in different ways to produce a measure of the 'error' in the recovered transformation. One measure was the sum of squares of shear parameters, another was the sum of squares of differences from expected length parameters. The position of points in the image was then perturbed and the 'error' terms were minimized by stochastic hill-climbing which performed as well as a genetic algorithm. In every case the parameters converged to values close to their true values with no significant displacement of the image points as shown by the black lines surrounding the blocks in figure 7.

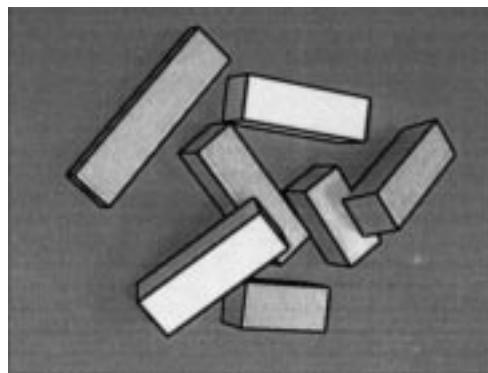


Figure 7. Parametric fitting of a unit-cube.

This example shows two kinds of prior knowledge being deployed, equally effectively, in a robot vision task. First, the minimization of the shear parameters is an example of continuous perception favouring the most rectilinear interpretation of the data. Second, the minimization of differences from known lengths is an example of categorical perception favouring the interpretation of the data which comes closest to some previously known prototypes. Both kinds of perception are available—all the programmer need do is provide a suitable ‘error’ function. Alternatively, a computer program could find appropriate ‘error’ functions using a genetic algorithm or neural net, as described in Cliff (1997) and Hinton (1997).

A very important practical detail of these experiments is that hundreds or thousands of evaluations were used during the hill-climbing to optimize a single parametric description of an object, so, it might be supposed that the combinatorial explosion contingent on searching all of the permutations of points on a model and in the image would be devastating. This is not so. The j -inverse is exact and the parameterization is very accurate, so it is reasonable to expect that all of the symmetries of the correspondences could be found on a single evaluation. Further, it seems that correspondences which are objectively bad, for example because they involve cross-overs, yield parameters that are extremely far from expected values, so these may be discarded very quickly. Thus the parameterization appears to offer a practical method of obtaining both continuous and categorical visual perception, along with detection of all affine and perspective symmetries, for those visual tasks where candidate landmarks can be found to bring into correspondence with a model. It should be remarked that the landmarks need not be points, the duality theorems of projective geometry would allow, lines, planes, and in general, hyperplanes to be substituted for points.

8. GESTALT PUZZLE

Now that perspexes have been introduced, the reader is invited to consider a thought experiment in which a robot seeks to solve a gestalt puzzle. The explanation of the robot’s behaviour is in terms of Palmer’s transformational theory of human visual perception (Palmer 1983), in which an observer searches for regularities in transformations applied to pictorial elements. However, Palmer did not have access to all of the affine and perspective transformations offered by the transformational interpretation of the perspex, nor did he have access to the coordinate and parametric interpretations.

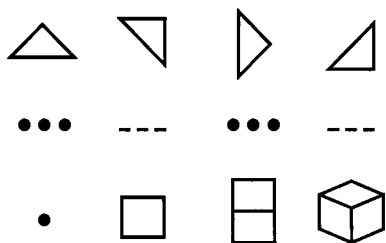


Figure 8. A gestalt puzzle.

Phil. Trans. R. Soc. Lond. B (1997)

Suppose that a robot looks at figure 8 and is programmed to search for the most concise description of an image. In Barlow’s terms (Barlow 1997), the robot is programmed to extract as much knowledge as it can from the figure. How might it operate?

The robot might start by fitting a perspex directly to the triangle in the top-left of the figure and then to successive triangles in what we see as the top row. The robot might describe these as a set of four coordinate perspexes, but would fail to find any other triangles. Thus the robot would demonstrate a gestalt of common form.

If the robot were to assume the first triangle fixed and apply the j -inverse to the second it would obtain the transformation which carries the first triangle into the second. Assuming the second triangle fixed and applying the j -inverse to the third it would obtain the same transformation. Proceeding along the row it would find that in every case the same transformation would generate the next figure from its predecessor. The robot might describe this situation by using one geometric perspex as the starting position and one transformational perspex as a generator. It would then need just one number to indicate the number of times the generator is to be applied. The robot would then have deduced a rule for generating an infinitely long sequence of these transformations, thus demonstrating the gestalt of common fate. The robot would also have achieved a reduction from four perspexes to two in order to describe the triangles, which is an increase in knowledge in Barlow’s terms.

If the robot were to examine the next unexplained object in raster order (the arbitrary order top-left to bottom-right) it would find what we see as the first dot of a group of three. The robot might match this to a geometric perspex annihilated to a point and might then find all of the other points in the figure, again demonstrating the gestalt of common form. Proceeding as for the triangles it might then find the transformation that generates a group of three dots from the first one. The robot might use three coordinate perspexes to generate the starting positions of what we see as the two groups of three dots and the isolated dot. It could then use one transformational perspex to generate the two series of three dots. Thus the robot demonstrates the gestalt of common fate when it generates a group of three dots.

The next unexplained object in raster order is a dash. The robot might encode the groups of three dashes in a similar way to the dots. That is, as two starting positions for an annihilated perspex describing a line and two generators. The robot might then detect that the generators are identical, perhaps by examining the perspexes directly. But suppose that the figure is badly drawn, how might the robot detect that nearly the same transformation is used as a generator each time? A simple way would be to take each transformational perspex and draw it into an image as if it were a geometric perspex. Measuring the area of overlap would soon show that the transformations were nearly the same. Thus the robot might detect this fact by applying an image processing operation to its own thoughts—the transformational perspexes in its own ‘mind’s eye’. The robot might perform this operation

very quickly because the required drawing routines are supported in hardware in many contemporary computer workstations. Here the gestalt of common form in the transformational perspexes gives rise to the gestalt of common fate in the coordinate perspexes. In this case the gestalts are in fact identical computations applied to two different kinds of perspex.

By a similar viewing of the transformations in its own 'mind's eye' the robot might see, firstly, that the same transformation carries the first group of three dots onto the second group of three dots as carries the first group of three dashes onto the second. It might go on to see that the same transformation carries each group of three things onto the next. Thus, the robot might use two geometric perspexes to represent the shapes of the dots and the dashes and two transformational perspexes to represent the repetition of three elements in a group, followed by the infinite sequence of alternating groups. The alternation of shape could be represented by a transformational perspex that selects the required shapes from an iconic memory composed of geometric perspexes, for example, by using reflection or rotation to visit different areas of the icon. The lone dot might be described by a single geometric perspex.

The robot might go on to recognize the sequence of dots and dashes as the Morse code for 'SOS', now the international distress signal, 'Mayday' (best pronounced in French). If we were to point at the dots and dashes and ask the robot to explain what is at this location in the image we would expect, amongst other answers, 'the international distress signal'. Conversely, if we were to ask the robot, 'where is the international distress signal?' we would expect that, amongst other answers, it would indicate the dots and dashes in the image. This is an example of visual knowledge (Anderson 1989) being related to linguistic knowledge of facts. The importance of making a link between visual and linguistic representations is that we could question the robot in order to satisfy ourselves as to what extent it understands what it sees.

The robot might describe the remaining three objects as annihilations of a cube and might see the common transformation that carries each figure onto the next, but it might do rather better than this. If it were to convert the geometric perspexes of the three views of the cube into parametric perspexes it would see that all of the cubes have the same parameter for translation and that they all have the same parameter for rotation. It could then go on to generate the infinite sequence of hypercubes that are rotated by 45° in successive planes. Furthermore, the robot might see that the first row of triangles has the same translation and rotation parameters. It might then see that all of the figures have common translation parameters along rows and a different common translation parameter down columns. Thus the parametric interpretation reveals factors of transformations (parameters) that would otherwise be conflated in its perception. The parameters have been carefully chosen so that they are related in a simple way to those that English speakers use, so that we might establish an effective dialogue with a robot programmed to use perspexes.

9. DISCUSSION

This section is entirely independent of the previous sections and contains philosophical speculation which the reader might prefer to skip.

The geometric, transformational, and parametric perspexes are all interchangeable, either at random or in more meaningful ways. This raises the possibility of constructing self-organizing networks of perspexes that solve visual and geometrical problems.

Future work is now aimed at developing an interpretation of a perspex as logic, extending earlier work (Anderson 1995), so that a computer can be programmed with perspexes to carry out the operations of perspex algebra discussed in this paper. This raises an intriguing possibility.

I define that X is visually conscious of Y if X can see Y. Thus programs in these proceedings are variously visually conscious of: wooden bricks; human faces; hand signs in American Sign Language; and parts of the human anatomy, including the human brain, revealed by non-invasive imaging techniques. This definition is saved from emptiness by observing that its force depends on the meaning of 'seeing' (Swartz 1965). In Leonardo da Vinci's use of the word 'seeing' (Lindberg & Cantor 1985), a stone could be said to 'see' the sun. I would allow that the stone is 'visually conscious', but only to the degree that it can 'see'. By comparison with the 'seeing' abilities of computer programs and animals, this is so low as to admit no more consciousness than would ordinarily be allowed to a stone.

The gestalt puzzle raises the possibility that a computer program could be visually conscious of its own attempts to search for regular and semi-regular patterns (transformations) of objects in space-time, or of partial regularities in terms of the parameters of transformations. If a logical perspex can be developed, then a computer program could be programmed to be visually conscious of all of its operations. Such a program might be programmed to experience qualia (sensory qualities). For example: the primary sensible quality that disk access is slower than core access; the secondary sensible quality that perspexes in the part of the core tagged as the 'frame store' can be shared with input on a serial keyboard port describing itself as 'a human observer', whereas other areas of core cannot be immediately shared; the vital perception that error rates in memory vary within well defined bounds, or are at serious risk of causing program malfunction and termination.

The implementation of such a program in a robot is the subject of current research with the objective of allowing empirical testing of the philosophical question, 'to what extent is this robot visually conscious?'

10. CONCLUSION

This paper introduces the algebraic perspex which describes all perspexes. Three particular perspexes are introduced. First, the geometric perspex which can be used as a building block for shapes in space-time, or to record landmarks on shapes in space-time. The land-

marks need not be points; the duality theorems allow points to stand for lines, planes, and hyperplanes. Second, the transformational perspex which describes the general linear (affine) and perspective transformations in space-time. The j -inverse can be used to recover a transformation that carries a model onto an image even in singular cases, which arise in the so called 'accidental' views, where the depth of an object disappears from view, or an object stays in the same place over time. Third, the parametric perspex which can be used to obtain continuous and categorical perception of shapes in space-time and can probably be used efficiently to discover affine and perspective symmetries of shapes in space-time.

The geometric, transformational, and parametric perspexes all have the property that every possible perspex is interpretable in each of the three ways. This holds out the exciting possibility for a program to build a random network of interrelated perspexes and to have it trained by feedback to: identify objects in images from their geometric structure; describe the movements and interrelationships between objects in terms of transformations; and, factor out individual parameters of a transformation or geometrical shape. The search for an effective perspex network need not be random, there are certain operations and combinations of operations which always retain a simple meaning. This opens up the possibility of providing neural nets of perspexes with sufficient internal structure to get them started on the long road of visual learning. Alternatively, it offers the possibility of designing and implementing very sophisticated symbolic programs.

Perspex algebra and perspex geometry are supersets of matrix algebra and classical projective geometry, respectively, so the reader has nothing to lose by adopting perspexes.

It was always exciting working in Geoff Sullivan's laboratory. Geoff can never be replaced, but his excitement and spirit of adventure live on in the laboratory.

This paper has benefited immeasurably from discussions with Steve Maybank and my co-organizers Horrace Barlow and Richard Gregory. Magnus Byne's work on our robot vision system was invaluable. Anthony Worrall and I spent many hours, years ago, discussing the parameterization. The perspective simplex was developed while on sabbatical at the kind invitation of Professor Bernd Neumann of the Arbeitsbereich Kognitive Systeme, Fachbereich Informatik, Universität Hamburg, Germany. The sabbatical was generously funded by both the Department of Computer Science and the Research Board at The University of Reading, UK.

Any errors, omissions, or confusions in this paper are entirely the author's responsibility.

REFERENCES

- Anderson, J. A. D. W. 1989 Visual conviction. *Proceedings of the 5th Alvey Vision Conference*, 301303. University of Reading, UK.
- Anderson, J. A. D. W. 1992a Canonical description of the perspective projections. Ph.D. thesis, Department of Computer Science, The University of Reading, UK.
- Anderson, J. A. D. W. 1992b Canonical description of the perspective projections. *Vision Geometry*, SPIE 300–311.
- Anderson, J. A. D. W. 1995 Models of possible visibility. I. Introductory discussion. Technical report RUCS/95/108,

Department of Computer Science, The University of Reading, UK.

- Anderson, J. A. D. W. 1996 Apologia for perspex algebra. Technical report RUCS/96/001, RUCS/97/TR/018/A, RUCS/97/TR/019/A, Department of Computer Science, The University of Reading, UK.
- Anderson, J. A. D. W. 1997 Understanding perspex algebra. Poster presented at The Royal Society Meeting, Knowledge-Based Vision in Man and Machine, February 1997 and published as technical report RUCS/97/EP/006/A, Department of Computer Science, The University of Reading, UK.
- Ballard, D. H. & Brown, C. M. 1982 *Computer vision*. New Jersey, USA: Prentice-Hall.
- Barlow, H. 1997 The knowledge used in vision and where it comes from. *Phil. Trans. R. Soc. Lond. B* **352**, 1143–1149 (This volume.)
- Baumgart, B. G. 1975 A polyhedron representation for computer vision. *Proc. AFIPS Nat. Comput. Conf.* pp. 586–596.
- Ben-Israel, A. & Greville, T. N. E. 1974 *Generalized inverses: theory and applications*. New York: Wiley.
- Boden, M. A. 1977 *Artificial intelligence and natural man*, 2nd edn. London: MIT Press.
- Boyer, C. B. & Merzbach, U. C. 1989 *A history of mathematics*, 2nd edn. New York: Wiley.
- Byne, J. H. M., M-Rouhani, B. & Anderson, J. A. D. W. 1996 CAD-based vision in a robotic workcell environment. *Intelligent Industrial Automation*, pp. 57–63.
- Cliff, D. & Noble, J. 1997 Knowledge-based vision and simple visual machines. *Phil. Trans. R. Soc. Lond. B* **352**, 1167–1177. (This volume.)
- Foley, J. D., van Dam, A., Feiner, S. K. & Hughes, J. F. 1987 *Computer graphics: principles and practice*. Massachusetts, USA: Addison Wesley.
- Garrett, J. & Anderson, J. A. D. W. 1995 Affine field-morphing. Technical report RUCS/95/030, Department of Computer Science, The University of Reading, UK.
- Golub, G. H. & van Loan, C. F. 1983 *Matrix computations*. Baltimore, USA: Johns Hopkins University Press.
- Greenberg, M. J. 1994 *Euclidean and non-Euclidean geometries*. New York: W. H. Freeman.
- Harris, J. 1992 *Algebraic geometry*. New York: Springer.
- Hinton, G. E. & Ghahramani, Z. 1997 Generative models for discovering sparse distributed representations. *Phil. Trans. R. Soc. Lond. B* **352**, 1179–1192. (This volume.)
- Lindberg, D. G. & Cantor, G. 1985 The discourse of light from the Middle Ages to the Enlightenment. University of California, LA: The William Andrew Clark Memorial Library.
- Mäntylä, M. 1988 *An introduction to solid modelling*. Maryland, USA: Computer Science Press.
- Marr, D. 1982 *Vision*. San Francisco: W. H. Freeman.
- Nashed, M. Z. 1976 *Generalized inverses and applications*. New York: Academic Press.
- Palmer, S. E. 1983 The psychology of perceptual organization: a transformational approach. In *Human and machine vision* (ed. J. Beck, B. Hope & A. Rosenfeld), pp. 268–339. Orlando, FL: Academic Press.
- Pentland, A. 1997 Content-based indexing of images and video. *Phil. Trans. R. Soc. Lond. B* **352**, 1285–1292. (This volume.)
- Riesenfeld, R. F. 1981 Homogeneous coordinates and projective planes in computer graphics. *IEEE CG&A* (Jan.) 50–55.
- Roberts, L. G. 1965 Machine perception of three-dimensional solids. In *Optical and electro-optical information processing* (ed. J. T. Tippet), pp. 59–197. Massachusetts, USA: MIT Press.
- Stolfi, J. 1991 *Oriented projective geometry*. London: Academic Press.
- Swartz, R. J. 1965 *Perceiving, sensing and knowing*. New York: Anchor Books.
- Vanarsdale, D. 1994 Homogeneous transformation matrices for computer graphics. *Computing & Graphics* **18**(2), 177–192.

BIOLOGICAL
SCIENCES

B

THE ROYAL
SOCIETY

PHILOSOPHICAL
TRANSACTIONS
OF

BIOLOGICAL
SCIENCES

B

THE ROYAL
SOCIETY

PHILOSOPHICAL
TRANSACTIONS
OF