International Tables for Crystallography (2006). Vol. B, Section 3.3.1.3, pp. 367–373.

3.3.1.2.3. *Orthogonalization of impure rotations*

There are several ways of deriving a strictly orthogonal matrix from a given approximately orthogonal matrix, among them the following.

(i) The Gram–Schmidt process. This is probably the simplest and the easiest to compute. If the given matrix consists of three column vectors $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3$ (later referred to as primers) which are to be replaced by three column vectors $\mathbf{u}_1, \mathbf{u}_2$ and $\mathbf{u}_3$ then the process is

$$\begin{aligned}
\mathbf{u}_1 &= \mathbf{v}_1/|\mathbf{v}_1| \\
\mathbf{u}_2 &= \mathbf{v}_2 - (\mathbf{u}_1 \cdot \mathbf{v}_2)\mathbf{u}_1 \\
\mathbf{u}_2 &= \mathbf{u}_2/|\mathbf{u}_2| \\
\mathbf{u}_3 &= \mathbf{v}_3 - (\mathbf{u}_1 \cdot \mathbf{v}_3)\mathbf{u}_1 - (\mathbf{u}_2 \cdot \mathbf{v}_3)\mathbf{u}_2 \\
\mathbf{u}_3 &= \mathbf{u}_3/|\mathbf{u}_3|.
\end{aligned}$$

As successive vectors are established, each vector $\mathbf{v}$ has subtracted from it its components in the directions of established vectors, and the remainder is normalized. The method will fail at the normalization step if the vectors $\mathbf{v}$ are not linearly independent. Otherwise, the process may be extended to any number of dimensions.

The weakness of the method is that, though $\mathbf{u}_1$ differs from $\mathbf{v}_1$ only in scale, $\mathbf{u}_N$ may differ grossly from $\mathbf{v}_N$ as the various columns are not treated equivalently.

(ii) A preferable method which treats all vectors equivalently is to iteratively replace the matrix $\boldsymbol{M}$ by $\frac{1}{2}(\boldsymbol{M} + \boldsymbol{M}^{T-1})$.

Defining the residual matrix $\boldsymbol{E}$ as

$$\boldsymbol{E} = \boldsymbol{M}\boldsymbol{M}^T - \boldsymbol{I},$$

then on each iteration $\boldsymbol{E}$ is replaced by

$$\boldsymbol{E}^2(\boldsymbol{M}\boldsymbol{M}^T)^{-1}/4$$

and convergence necessarily ensues.

(iii) A third method resolves $\boldsymbol{M}$ into its symmetric and antisymmetric parts

$$\boldsymbol{S} = \tfrac{1}{2}(\boldsymbol{M} + \boldsymbol{M}^T), \quad \boldsymbol{A} = \tfrac{1}{2}(\boldsymbol{M} - \boldsymbol{M}^T), \quad \boldsymbol{M} = \boldsymbol{S} + \boldsymbol{A}$$

and constructs an orthogonal matrix for which only $\boldsymbol{S}$ is altered. $\boldsymbol{A}$ determines l, m, n and θ as shown in Section 3.3.1.2.1, and from these a new $\boldsymbol{S}$ may be constructed.

(iv) A fourth method is to treat the general matrix $\boldsymbol{M}$ as a combination of pure strain and pure rotation. Setting

$$\boldsymbol{M} = \boldsymbol{R}\boldsymbol{T}$$

with $\boldsymbol{R}$ orthogonal and $\boldsymbol{T}$ symmetrical gives

$$\boldsymbol{T} = (\boldsymbol{M}^T\boldsymbol{M})^{1/2}, \quad \boldsymbol{R} = \boldsymbol{M}(\boldsymbol{M}^T\boldsymbol{M})^{-1/2}.$$

The rotation so found is the one which exactly superposes those three mutually perpendicular directions which remain mutually perpendicular under the transformation $\boldsymbol{M}$.

$\boldsymbol{T} - \boldsymbol{I}$ is then the strain tensor of an unrotated body.

Writing $\boldsymbol{M} = \boldsymbol{T}\boldsymbol{R}$, $\boldsymbol{T} = (\boldsymbol{M}\boldsymbol{M}^T)^{1/2}$, $\boldsymbol{R} = (\boldsymbol{M}\boldsymbol{M}^T)^{-1/2}\boldsymbol{M}$ may also be useful, in which $\boldsymbol{T} - \boldsymbol{I}$ is the strain tensor of a rotated body. See also Section 3.3.1.2.2 (iv).

3.3.1.2.4. *Eigenvalues and eigenvectors of orthogonal matrices*

If $\boldsymbol{R}$ is the orthogonal matrix given in Section 3.3.1.2.1 in terms of the direction cosines l, m and n of the axis of rotation, then it is clear that (l, m, n) is an eigenvector of $\boldsymbol{R}$ with eigenvalue unity because

$$\boldsymbol{R}\begin{pmatrix} l \\ m \\ n \end{pmatrix} = \begin{pmatrix} l \\ m \\ n \end{pmatrix}.$$

Consideration of the determinant $|\boldsymbol{R} - \lambda\boldsymbol{I}| = 0$ shows that the sum of the three eigenvalues is $1 + 2\cos\theta$ and that their product is unity. Hence the three eigenvalues are 1, $e^{i\theta}$ and $e^{-i\theta}$. Since $\boldsymbol{R}$ is real, its product with any real vector is also real, yet its product with an eigenvector must, in general, be complex. Thus the eigenvectors must themselves be complex.

The remaining two eigenvectors $\mathbf{u}$ may be found using the results of Section 3.3.1.2.1 (*q.v.*) according to

$$\boldsymbol{R}\mathbf{u} = \mathbf{u} + \frac{2}{1+t^2}\{(\mathbf{r} \times \mathbf{u}) + [\mathbf{r} \times (\mathbf{r} \times \mathbf{u})]\} = \mathbf{u}e^{\pm i\theta} = \mathbf{u}\frac{1 \pm it}{1 \mp it},$$

which is solved by any vector of the form

$$\mathbf{u} = \mathbf{l} \times \mathbf{v} \mp i\mathbf{l} \times (\mathbf{l} \times \mathbf{v})$$

for any real vector $\mathbf{v}$, where $\mathbf{l}$ is the normalized axis vector, $\mathbf{l}t = \mathbf{r}$, $|\mathbf{l}| = 1$, $t = \tan(\theta/2)$. Eigenvectors for the two eigenvalues may have unrelated $\mathbf{v}$ vectors though the sign choices are coupled. If the vector $\mathbf{v}$ is rotated about $\mathbf{l}$ through an angle φ the corresponding vector $\mathbf{u}$ is multiplied by $e^{-i\varphi}$ and remains an eigenvector. Using superscript signs to denote the sign of θ in the eigenvalue with which each vector is associated, the matrix

$$\boldsymbol{U} = (\mathbf{l},\ \mathbf{u}^+,\ \mathbf{u}^-)$$

has the properties that

$$\boldsymbol{R}\boldsymbol{U} = \boldsymbol{U}\begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{i\theta} & 0 \\ 0 & 0 & e^{-i\theta} \end{pmatrix}$$

and

$$\boldsymbol{U}^{*T}\boldsymbol{U} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2|\mathbf{l} \times \mathbf{v}^+|^2 & 0 \\ 0 & 0 & 2|\mathbf{l} \times \mathbf{v}^-|^2 \end{pmatrix}$$

which places restrictions on $\mathbf{v}$ if this is to be the identity. Note that the 23 element vanishes even in the absence of any relationship between $\mathbf{v}^+$ and $\mathbf{v}^-$.

A convenient form for $\boldsymbol{U}$, symmetrical in the elements of $\mathbf{l}$, is obtained by setting $\mathbf{v}^+ = \mathbf{v}^- = [111]$ and is

$$\boldsymbol{U} = \begin{pmatrix} l & \{(m-n) - i[l(l+m+n)-1]\}/d & \{(m-n) + i[l(l+m+n)-1]\}/d \\ m & \{(n-l) - i[m(l+m+n)-1]\}/d & \{(n-l) + i[m(l+m+n)-1]\}/d \\ n & \{(l-m) - i[n(l+m+n)-1]\}/d & \{(l-m) + i[n(l+m+n)-1]\}/d \end{pmatrix}$$

in which the normalizing denominator is given by

$$d = 2\sqrt{1 - lm - mn - nl}.$$

3.3.1.3. *Projection transformations and spaces*

In the following section we address the question of the relationship between the coordinates of a molecular model and the corresponding coordinates on the screen of the graphics device. A good introduction to this topic is given by Newman & Sproull (1973), and Foley *et al.* (1990) give a comprehensive account of the field, including recent developments, especially those arising from the development of raster-graphics technologies.

3.3.1.3.1. *Definitions*

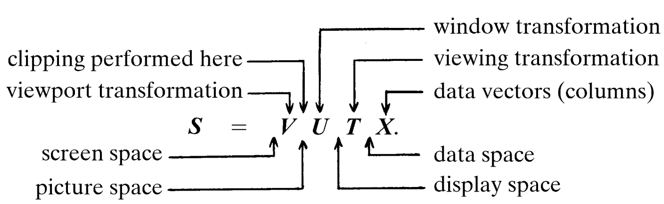
Typically, the coordinates, $\mathbf{X}$, of points in an object to be drawn are held in homogeneous Cartesian form as described in Section 3.3.1.1.2. Such coordinates are said to be in *data space* or world coordinates and this coordinate system is generally a constant aspect of the problem.

In order to view these data in convenient ways such coordinates may be subjected to a 4×4 *viewing transformation* $\boldsymbol{T}$, affecting orientation, scale *etc.*, the resulting coordinates $\boldsymbol{T}\mathbf{X}$ being then in *display space*. Here, and throughout what follows, we treat position vectors as columns with transformation matrices as factors on the left, though some writers do the reverse.

In general, only some portion of display space which lies inside a certain frustum of a pyramid is required to fall within the picture. The pyramid may be thought of as having the observer's eye at its vertex, with a rectangular base corresponding to the picture area. This volume is called a *window*. A transformation, $\boldsymbol{U}$, which takes *display-space* coordinates as input and generates vectors (X, Y, Z, W) for which X/W and $Y/W = \pm 1$ for points on the left, right, top and bottom boundaries of the window and for which Z/W takes particular values on the front and back planes of the window, is said to be a *windowing transformation*. In machines for which Z/W controls intensity depth cueing, the range of Z/W corresponding to the window is likely to be 0 to 1 rather than -1 to 1. Coordinates obtained by multiplying display-space coordinates by $\boldsymbol{U}$ are termed *picture-space* coordinates. Mathematically, $\boldsymbol{U}$ is a 4×4 matrix like any other, but functionally it is special. Declaring a transformation to be a windowing transformation implies that only resulting points having $|X|, |Y| < W$ and positive $Z < W$ are to be plotted. Machines with clipping hardware to truncate lines which run out of the picture perform clipping on the output from the windowing transformation.

Finally, the picture has to be drawn in some rectangular portion of the screen which is allocated for the purpose. Such an area is termed a *viewport* and is defined in terms of *screen coordinates* which are defined absolutely for the hardware in question as $\pm n$ for full-screen deflection, where n is declared by the manufacturer. *Screen coordinates* are obtained from picture coordinates with a *viewport transformation*, $\boldsymbol{V}$.*

To summarize, screen coordinates, $\mathbf{S}$, are given by



3.3.1.3.2. *Translation*

The transformation

$$\begin{pmatrix} N\boldsymbol{I} & \mathbf{V} \\ \mathbf{0}^T & N \end{pmatrix} \begin{pmatrix} \mathbf{X} \\ W \end{pmatrix} = \begin{pmatrix} \mathbf{X}N + \mathbf{V}W \\ NW \end{pmatrix} \simeq \begin{pmatrix} \mathbf{X} + \mathbf{V}W/N \\ W \end{pmatrix}$$
$$\simeq \begin{pmatrix} \mathbf{X}/W + \mathbf{V}/N \\ 1 \end{pmatrix}$$

evidently corresponds to the addition of the vector $\mathbf{V}W/N$ to the components of $\mathbf{X}$ or of $\mathbf{V}/N$ to the components of $\mathbf{X}/W$. ($\boldsymbol{I}$ is the identity.) Displacements may thus be affected by expressing the required displacement vector in homogeneous coordinates with any suitable choice of N (commonly, $N = W$), with $\mathbf{V}$ scaled to correspond to this choice, and loading the 4×4 transformation matrix as indicated above.

* In recent years it has become increasingly common, especially in two-dimensional work, to apply the term 'window' to what is here called a viewport, but in this chapter we use these terms in the manner described in the text.

3.3.1.3.3. *Rotation*

Rotation about the origin is achieved by

$$\begin{pmatrix} N\boldsymbol{R} & \mathbf{0} \\ \mathbf{0}^T & N \end{pmatrix} \begin{pmatrix} \mathbf{X} \\ W \end{pmatrix} = \begin{pmatrix} N\boldsymbol{R}\mathbf{X} \\ NW \end{pmatrix} \simeq \begin{pmatrix} \boldsymbol{R}\mathbf{X} \\ W \end{pmatrix},$$

in which $\boldsymbol{R}$ is an orthogonal 3×3 matrix. $\boldsymbol{R}$ necessarily has elements not exceeding one in modulus. For machines using integer arithmetic, therefore, N would be chosen large enough (usually half the largest possible integer) for the product $N\boldsymbol{R}$ to be well represented in the available word length. Characteristically, N affects resolution but not scale.

3.3.1.3.4. *Scale*

The transformation

$$\begin{pmatrix} SN\boldsymbol{I} & \mathbf{0} \\ \mathbf{0}^T & N \end{pmatrix} \begin{pmatrix} \mathbf{X} \\ W \end{pmatrix} = \begin{pmatrix} SN\mathbf{X} \\ NW \end{pmatrix} \simeq \begin{pmatrix} S\mathbf{X} \\ W \end{pmatrix}$$

scales the vector ($\mathbf{X}$, W) by the factor S. For integer working and $|S| < 1$, N should be set to the largest representable integer. For $|S| > 1$ the product SN should be the largest representable integer, N being reduced accordingly.

3.3.1.3.5. *Windowing and perspective*

It is necessary at this point to relate the discussion to the axial system inherent in the graphics device employed. One common system adopts X horizontal and to the right when viewing the screen, Y vertically upwards in the plane of the screen, and Z normal to X and Y with $+Z$ into the screen. This is, unfortunately, a left-handed system in that $(\mathbf{X} \times \mathbf{Y}) \cdot \mathbf{Z}$ is negative. Since it is usual in crystallographic work to use right-handed axial systems it is necessary to incorporate a matrix of the form

$$\begin{pmatrix} W & 0 & 0 & 0 \\ 0 & W & 0 & 0 \\ 0 & 0 & -W & 0 \\ 0 & 0 & 0 & W \end{pmatrix}$$

either as the left-most factor in the matrix $\boldsymbol{T}$ or as the right-most factor in the windowing transformation $\boldsymbol{U}$ (see Section 3.3.1.3.1). The latter choice is to be preferred and is adopted here. The former choice leads to complications if transformations in display space will be required. Display-space coordinates are necessarily referred to this axial system.

Let L, R, T, B, N and F be the left, right, top, bottom, near and far boundaries of the windowed volume $(L < R, T > B, N < F)$, S be the Z coordinate of the screen, and C, D and E be the coordinates of the observer's eye position, all ten of these parameters being referred to the origin of display space as origin, which may be anywhere in relation to the hardware. L, R, T and B are to be evaluated in the screen plane. All ten parameters may be referred to their own fourth coordinate, V, meaning that the point (X, Y, Z, W) in display space will be on the left boundary of the picture if $X/W = L/V$ when $Z/W = S/V$. V may be freely chosen so that all eleven quantities and all elements of $\boldsymbol{U}$ suit the word length of the machine. These relationships are illustrated in Fig. 3.3.1.1.

Since

$$(X, Y, Z, W) \simeq \left(\frac{XV}{W}, \frac{YV}{W}, \frac{ZV}{W}, V\right),$$

XV/W is a display-space coordinate on the same scale as the window parameters. This must be plotted on the screen at an X coordinate (on the scale of the window parameters) which is the weighted mean of XV/W and C, the weights being $(S - E)$ and

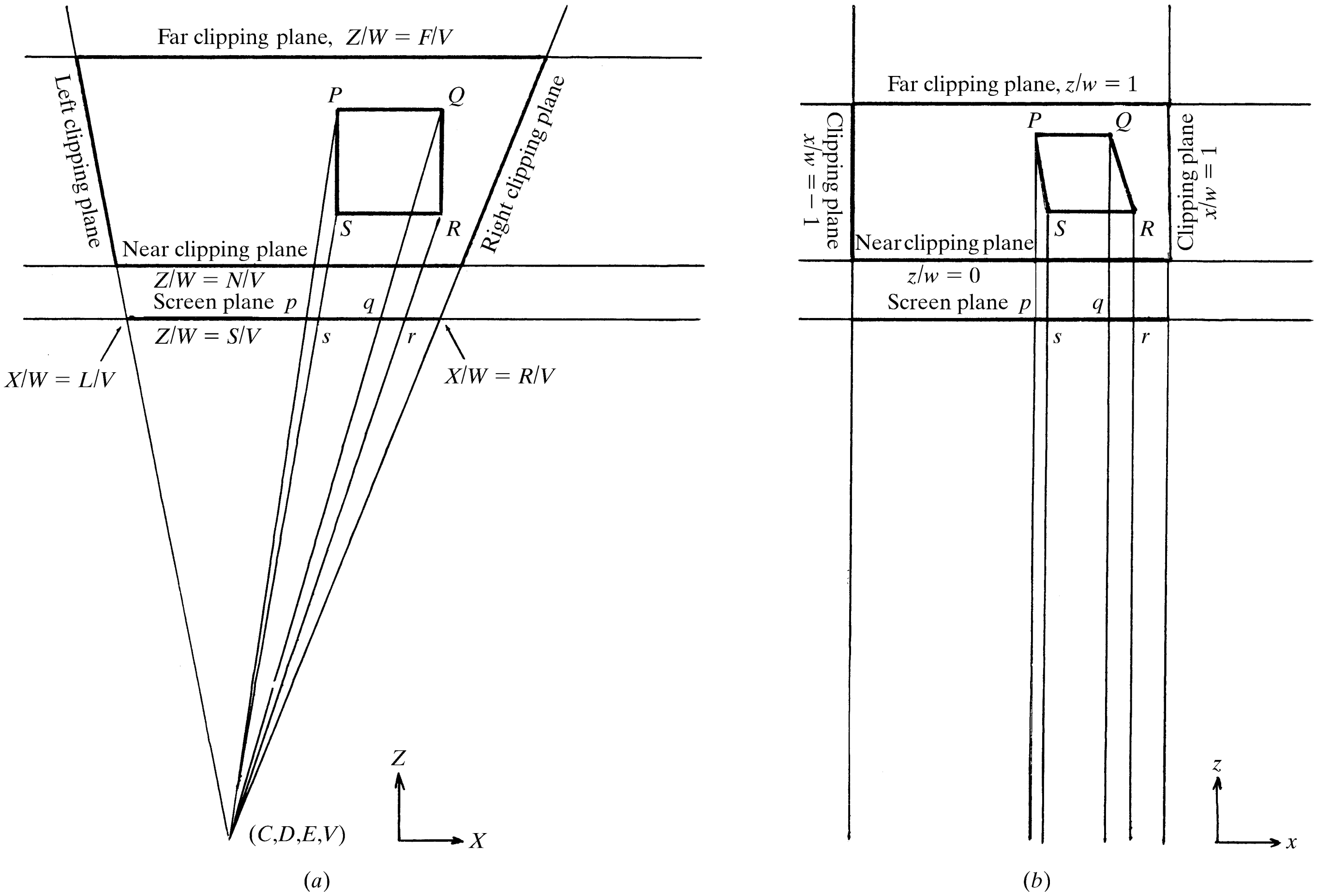


Fig. 3.3.1.1. The relationship between display-space coordinates (X, Y, Z, W) and picture-space coordinates (x, y, z, w) derived from them by the window transformation, $\boldsymbol{U}$. (a) Display space (in X, Z projection) showing a square object P, Q, R, S for display viewed from the position (C, D, E, V). The bold trapezium is the window (volume) and the bold line is the viewport portion of the screen. The points P, Q, R and S must be plotted at p, q, r and s to give the correct impression of the object. (b) Picture space (in x, z projection). The window is mapped to a rectangle and all sight lines are parallel to the z axis, but the object P, Q, R, S is no longer square. The distribution of p, q, r and s is identical in the two cases. Note that z/w values are not linear on Z/W, and that the origin of picture space arises at the midpoint of the near clipping plane, regardless of the location of the origin of display space. The figure is accurately to scale for coincident viewport positions. The words 'Left clipping plane', if part of the scene in display space, would currently be obscured, but would come into view if the eye moved to the right, increasing C, as the left clipping plane would pivot about the point L/V in the screen plane.

$(ZV/W - S)$, respectively. This provides perspective because the weighted mean is at the point where the straight line from (X, Y, Z, W) to the eye intersects the screen. This then has to be mapped into the L-to-R interval, so that picture-space coordinates (x, y, z, w) are given by

$$\begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = \begin{pmatrix} \dfrac{2(S-E)V}{(R-L)} & 0 & \dfrac{(2C-R-L)V}{(R-L)} & \dfrac{(R+L)E-2SC}{(R-L)} \\ 0 & \dfrac{2(S-E)V}{(T-B)} & \dfrac{(2D-T-B)V}{(T-B)} & \dfrac{(T+B)E-2SD}{(T-B)} \\ 0 & 0 & \dfrac{(F-E)V}{(F-N)} & \dfrac{-N(F-E)}{(F-N)} \\ 0 & 0 & V & -E \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix}$$

which provides for $|x/w|$ and $|y/w|$ to be unity on the picture boundaries, which is usually a requirement of the clipping hardware, and for $0 < z/w < 1$, zero being for the near-plane boundary. Even though z/w is not linear on Z/W, straight lines and planes in display space transform to straight lines and planes in picture space, the non-linearity affecting only distances. Thus vector-drawing machines are not disadvantaged by the introduction of perspective.

Note that the dimensionality of X/W must equal that of S/V and that this may be regarded as length or as a pure number, but that in either case x/w is dimensionless, consistent with the stipulation that the picture boundaries be defined by the pure number ± 1.

The above matrix is $\boldsymbol{U}$ and is suited to left-handed hardware systems. Note that only the last column of $\boldsymbol{U}$ (the translational part) is sensitive to the location of the origin of display space and that if the eye is on the normal to the picture centre then $C = \frac{1}{2}(R+L)$, $D = \frac{1}{2}(T+B)$ and simplifications result. If C, D and E can be continuously monitored then dynamic parallax as well as perspective may be obtained (Diamond *et al.*, 1982).

If data space is referred to right-handed axes, the viewing transformation $\boldsymbol{T}$ involves only proper rotations and the hardware uses a left-handed axial system then elements in the third column of $\boldsymbol{U}$ should be negated, as explained in the opening paragraph.

To provide for orthographic projection, multiply every element of $\boldsymbol{U}$ by $-K/E$ and then let $E \rightarrow -\infty$, choosing some positive K to suit the word length of the machine [see Section 3.3.1.1.2 (iii)]. The result is

$$U' \simeq \begin{pmatrix} \dfrac{2KV}{(R-L)} & 0 & 0 & \dfrac{-K(R+L)}{(R-L)} \\ 0 & \dfrac{2KV}{(T-B)} & 0 & \dfrac{-K(T+B)}{(T-B)} \\ 0 & 0 & \dfrac{KV}{(F-N)} & \dfrac{-KN}{(F-N)} \\ 0 & 0 & 0 & K \end{pmatrix},$$

which is the orthographic window.

It may be convenient in some applications to separate the functions of windowing and the application of perspective, and to write

$$U = U'P,$$

where U and U' are as above and P is a perspective transformation given by

$$P = (U')^{-1}U \simeq \begin{pmatrix} S-E & 0 & C & -SC/V \\ 0 & S-E & D & -SD/V \\ 0 & 0 & F-E+N & -NF/V \\ 0 & 0 & V & -E \end{pmatrix},$$

which involves F and N but not R, L, T or B. In this form the action of P may be thought of as compressing distant parts of display space prior to an orthographic projection by U' into picture space.

Other factorizations of U are possible, for example

$$U = U''P'$$

with

$$U'' \simeq \begin{pmatrix} \dfrac{2KV}{R-L} & 0 & 0 & \dfrac{-K(R+L)}{(R-L)} \\ 0 & \dfrac{2KV}{T-B} & 0 & \dfrac{-K(T+B)}{(T-B)} \\ 0 & 0 & \dfrac{KV(N-E)(F-E)}{E^2(F-N)} & \dfrac{KN(F-E)}{E(F-N)} \\ 0 & 0 & 0 & K \end{pmatrix}$$

$$P' \simeq \begin{pmatrix} S-E & 0 & C & -SC/V \\ 0 & S-E & D & -SD/V \\ 0 & 0 & -E & 0 \\ 0 & 0 & V & -E \end{pmatrix},$$

which renders P' independent of all six boundary planes, but U'' is no longer independent of E. It is not possible to factorize U so that the left factor is a function only of the boundary planes and the right factor a function only of eye and screen positions.

Note that as $E \to -\infty$, $U'' \to U'$, P and $P' \to -IE \simeq I$.

3.3.1.3.6. *Stereoviews*

Assuming that left- and right-eye views are to be presented through the same viewport (next section) or that their viewports are to be superimposed by an external optical system, *e.g.* Ortony mirrors, then stereopairs are obtained by using appropriate eye coordinates in the U matrix of the previous section. However, U may be factorized according to

$$U = U'''S$$

in which U''' is the matrix U obtained by setting (C, D, E, V) to correspond to the point midway between the viewer's eyes and

$$S = \begin{pmatrix} 1 & 0 & c/(S-E) & -cS/(S-E)V \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\simeq \begin{pmatrix} V & 0 & cV/(S-E) & -cS/(S-E) \\ 0 & V & 0 & 0 \\ 0 & 0 & V & 0 \\ 0 & 0 & 0 & V \end{pmatrix}$$

in which $(c, 0, 0, V)$ is the position of the right eye relative to the mean eye position, and the left-eye view is obtained by negating c.

Stereo is often approximated by introducing a rotation about the Y axis of $\pm\sin^{-1}[c/(S-E)]$ to the views or $\sin^{-1}[2c/(S-E)]$ to one of them. The first corresponds to

$$S = \begin{pmatrix} \sqrt{1-\sigma^2} & 0 & \sigma & 0 \\ 0 & 1 & 0 & 0 \\ -\sigma & 0 & \sqrt{1-\sigma^2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

with $\sigma = c/(S-E)$. The main difference is in the resulting Z value, which only affects depth cueing and z clipping. The X translation which arises if $S \neq 0$ is also suppressed, but this is not likely to be noticeable. σ is often treated as a constant, such as $\sin 3^\circ$.

The distinction in principle between the true S and the rotational approximation is that with the true S the eye moves relative to the screen and the displayed object, whereas with the approximation the eye and the screen are moved relative to the displayed object, in going from one view to the other.

Strobing of left and right images may conveniently be accomplished with an electro-optic liquid-crystal shutter as described by Harris *et al.* (1985). The shutter is switched by the display itself, thus solving the synchronization problem in a manner free of inertia.

A further discussion of stereopairs may be found in Johnson (1970) and in Thomas (1993), the second of which generalizes the treatment to allow for the possible presence of an optical system.

3.3.1.3.7. *Viewports*

The window transformation of the previous two sections has been constructed to yield picture coordinates (X, Y, Z, W) (formerly called x, y, z, w) such that a point having X/W or $Y/W = \pm 1$ is on the boundary of the picture, and the clipping hardware operates on this basis. However, the edges of the picture need not be at the edges of the screen and a viewport transformation, V, is therefore needed to position the picture in the requisite part of the screen.

$$V = \begin{pmatrix} (r-l)/2 & 0 & 0 & (r+l)/2 \\ 0 & (t-b)/2 & 0 & (t+b)/2 \\ 0 & 0 & n & 0 \\ 0 & 0 & 0 & n \end{pmatrix},$$

where r, l, t and b are now the right, left, top and bottom boundaries of the picture area, or viewport, expressed in screen coordinates, and n is the full-screen deflection value. Thus a point with $X/W = 1$ in picture space plots on the screen with an X coordinate which is a fraction r/n of full-screen deflection to the right. Z/W is unchanged by $\boldsymbol{V}$ and is used only to control intensity in a technique known as depth cueing.

It is necessary, of course, to arrange for the aspect ratio of the viewport, $(r-l)/(t-b)$, to equal that of the window otherwise distortions are introduced.

3.3.1.3.8. *Compound transformations*

In this section we consider the viewing transformation $\boldsymbol{T}$ of Section 3.3.1.3.1 and its construction in terms of translation, rotation and scaling, Sections 3.3.1.3.2–4. We use $\boldsymbol{T}'$ to denote a new transformation in terms of the prevailing transformation $\boldsymbol{T}$.

We note first that any 4×4 matrix of the form

$$\begin{pmatrix} U\boldsymbol{R} & \mathbf{V} \\ \mathbf{0}^T & W \end{pmatrix},$$

with U a scalar, may be factorized according to

$$\begin{pmatrix} U\boldsymbol{R} & \mathbf{V} \\ \mathbf{0}^T & W \end{pmatrix} \simeq \begin{pmatrix} U\boldsymbol{I} & \mathbf{0} \\ \mathbf{0}^T & W \end{pmatrix}\begin{pmatrix} U\boldsymbol{I} & \mathbf{V} \\ \mathbf{0}^T & U \end{pmatrix}\begin{pmatrix} U\boldsymbol{R} & \mathbf{0} \\ \mathbf{0}^T & U \end{pmatrix}$$

and also that multiplying

$$\begin{pmatrix} U\boldsymbol{R} & \mathbf{V} \\ \mathbf{0}^T & W \end{pmatrix}$$

by an isotropic scaling matrix, a rotation, or a translation, either on the left or on the right, yields a product matrix of the same form, and its inverse

$$\begin{pmatrix} W\boldsymbol{R}^T & -\boldsymbol{R}^T\mathbf{V} \\ \mathbf{0}^T & U \end{pmatrix}$$

is also of this form, *i.e.* any combination of these three operations in any order may be reduced by the above factorization to a rotation about the original origin, a translation (which defines a new origin) and an expansion or contraction about the new origin, applied in that order.

If

$$\begin{pmatrix} N\boldsymbol{R} & \mathbf{0} \\ \mathbf{0}^T & N \end{pmatrix}$$

is a rotation matrix as in Section 3.3.1.3.3, its application produces a rotation about an axis through the origin defined only in the space in which it is applied. For example, if

$$\boldsymbol{R} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\boldsymbol{T}'\begin{pmatrix} \mathbf{X} \\ W \end{pmatrix} = \boldsymbol{T}\begin{pmatrix} N\boldsymbol{R} & \mathbf{0} \\ \mathbf{0}^T & N \end{pmatrix}\begin{pmatrix} \mathbf{X} \\ W \end{pmatrix}$$

rotates the image about the z axis of data space, whatever the prevailing viewing transformation, $\boldsymbol{T}$.

Forming

$$\begin{pmatrix} N\boldsymbol{R} & \mathbf{0} \\ \mathbf{0}^T & N \end{pmatrix}\boldsymbol{T}\begin{pmatrix} \mathbf{X} \\ W \end{pmatrix}$$

rotates the image about the z axis of display space, *i.e.* the normal to the tube face under the usual conventions, whatever the prevailing $\boldsymbol{T}$. Furthermore, if this rotation is to appear to be about some chosen position in the picture, *e.g.* the centre, then the window transformation $\boldsymbol{U}$, Section 3.3.1.3.5, must place the origin of display space there by setting $F > S = R + L = T + B = 0 > N$, in the notation of that section.

If a rotation is to be about a point

$$\begin{pmatrix} \mathbf{V} \\ N \end{pmatrix}$$

then

$$\boldsymbol{T}' = \begin{pmatrix} N\mathbf{I} & \mathbf{V} \\ \mathbf{0}^T & N \end{pmatrix}\begin{pmatrix} N'\boldsymbol{R} & \mathbf{0} \\ \mathbf{0}^T & N' \end{pmatrix}\begin{pmatrix} N\boldsymbol{I} & -\mathbf{V} \\ \mathbf{0}^T & N \end{pmatrix}\boldsymbol{T}$$
$$\simeq \begin{pmatrix} N\boldsymbol{R} & \mathbf{V}-\boldsymbol{R}\mathbf{V} \\ \mathbf{0}^T & N \end{pmatrix}\boldsymbol{T}$$

or

$$\boldsymbol{T}' = \boldsymbol{T}\begin{pmatrix} N\boldsymbol{I} & \mathbf{V} \\ \mathbf{0}^T & N \end{pmatrix}\begin{pmatrix} N'\boldsymbol{R} & \mathbf{0} \\ \mathbf{0}^T & N' \end{pmatrix}\begin{pmatrix} N\boldsymbol{I} & -\mathbf{V} \\ \mathbf{0}^T & N \end{pmatrix}$$
$$\simeq \boldsymbol{T}\begin{pmatrix} N\boldsymbol{R} & \mathbf{V}-\boldsymbol{R}\mathbf{V} \\ \mathbf{0}^T & N \end{pmatrix}$$

according to whether $\boldsymbol{R}$ and $\mathbf{V}$ are both defined in display space or both in data space. If the rotation is defined in display space and the position of the centre of rotation is defined in data space, then the first form of $\boldsymbol{T}'$ must be used, in which $\mathbf{V}$ is first computed from

$$\begin{pmatrix} \mathbf{V} \\ N \end{pmatrix} = \boldsymbol{T}\begin{pmatrix} \mathbf{U} \\ W \end{pmatrix}$$

for a rotation centre at

$$\begin{pmatrix} \mathbf{U} \\ W \end{pmatrix}$$

in data space.

For continuous rotations defined in display space it is usually worthwhile to bring the centre of rotation to the origin of display space and leave it there, *i.e.* to omit the left-most factor in the first expression for $\boldsymbol{T}'$. Incremental rotations can then be made by further rotational factors on the left without further attention to $\mathbf{V}$. When continuous rotations are implemented by repeated multiplication of $\boldsymbol{T}$ by a rotation matrix, say thirty times a second for a minute or so, the orthogonality of the top-left partition of $\boldsymbol{T}$ may become degraded by accumulation of round-off error and this should be corrected occasionally by one of the methods of Section 3.3.1.2.3.

It is sometimes a requirement, depending on hardware capabilities, to affect a transformation in display space when access to data space is all that is readily available. In such a case

$$\boldsymbol{T}' = \boldsymbol{T}_1\boldsymbol{T} = \boldsymbol{T}\boldsymbol{T}_2,$$

where $\boldsymbol{T}_1$ is the required alteration to the prevailing viewing transformation $\boldsymbol{T}$ and $\boldsymbol{T}_2$ is the data-space equivalent,

$$\boldsymbol{T}_2 = \boldsymbol{T}^{-1}\boldsymbol{T}_1\boldsymbol{T} = \begin{pmatrix} U\boldsymbol{R} & \mathbf{V} \\ \mathbf{0}^T & W \end{pmatrix}^{-1}\begin{pmatrix} U_1\boldsymbol{R}_1 & \mathbf{V}_1 \\ \mathbf{0}^T & W_1 \end{pmatrix}\begin{pmatrix} U\boldsymbol{R} & \mathbf{V} \\ \mathbf{0}^T & W \end{pmatrix}$$
$$\simeq \begin{pmatrix} UU_1\boldsymbol{R}^T\boldsymbol{R}_1\boldsymbol{R} & \boldsymbol{R}^T(U_1\boldsymbol{R}_1\mathbf{V} + W\mathbf{V}_1 - W_1\mathbf{V}) \\ \mathbf{0}^T & UW_1 \end{pmatrix}.$$

An important special case is when $\boldsymbol{T}_1$ is to effect a rotation about the origin of display space without change of scale, so that $\mathbf{V}_1 = 0, U_1 = W_1 = W$, for then

$$\boldsymbol{T}_2 \simeq \begin{pmatrix} U\boldsymbol{R}^T\boldsymbol{R}_1\boldsymbol{R} & \boldsymbol{R}^T(\boldsymbol{R}_1 - \boldsymbol{I})\mathbf{V} \\ \mathbf{0}^T & U \end{pmatrix}.$$

If $\mathbf{r}$ is the required axis of rotation of $\boldsymbol{R}_1$ in display space then the axis of rotation of $\boldsymbol{R}^T\boldsymbol{R}_1\boldsymbol{R}$ in data space is $\mathbf{s} = \boldsymbol{R}^T\mathbf{r}$ since $\boldsymbol{R}^T\boldsymbol{R}_1\boldsymbol{R}\mathbf{s} = \mathbf{s}$. This gives a particularly simple result if $\boldsymbol{R}_1$ is to be a primitive rotation for then $\mathbf{s}$ is the relevant row of $\boldsymbol{R}$, and $\boldsymbol{R}^T\boldsymbol{R}_1\boldsymbol{R}$

can be constructed directly from this and the required angle of rotation.

3.3.1.3.9. *Inverse transformations*

It is frequently a requirement to be able to identify a feature or position in data space from its position on the screen. Facilities for identifying an existing feature on the screen are in many instances provided by the manufacturer as a 'hit' function which correlates the position indicated on the screen by the user (with a tablet or light pen) with the action of drawing and flags the corresponding item in the drawing internally as having been hit. In other instances it may be necessary to be able to indicate a position in data space independently of any drawn feature and this may be done by setting two or more non-parallel sight lines through the displayed volume and finding their best point of intersection in data space.

In Section 3.3.1.3.1 the relationship between data-space coordinates and screen-space coordinates was given as

$$\mathbf{S} = \boldsymbol{VUT}\mathbf{X};$$

hence data-space coordinates are given by

$$\mathbf{X} = \boldsymbol{T}^{-1}\boldsymbol{U}^{-1}\boldsymbol{V}^{-1}\mathbf{S}.$$

A line of sight through the displayed volume passing through the point

$$\begin{pmatrix} x \\ y \end{pmatrix}$$

on the screen is the line joining the two position vectors

$$\boldsymbol{S} = \begin{pmatrix} x & x \\ y & y \\ o & n \\ n & n \end{pmatrix}$$

in screen-space coordinates, as in Section 3.3.1.3.7, from which the corresponding two points in data space may be obtained using

$$\boldsymbol{V}^{-1} \simeq \begin{pmatrix} \dfrac{2n}{r-l} & 0 & 0 & \dfrac{-(r+l)}{(r-l)} \\ 0 & \dfrac{2n}{t-b} & 0 & -\dfrac{(t+b)}{(t-b)} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and

$$\boldsymbol{U}^{-1} \simeq \begin{pmatrix} \dfrac{R-L}{2(S-E)} & 0 & \dfrac{-C(F-N)}{(F-E)(N-E)} & \dfrac{(R+L)(N-E)-2C(N-S)}{2(N-E)(S-E)} \\ 0 & \dfrac{T-B}{2(S-E)} & \dfrac{-D(F-N)}{(F-E)(N-E)} & \dfrac{(T+B)(N-E)-2D(N-S)}{2(N-E)(S-E)} \\ 0 & 0 & \dfrac{-E(F-N)}{(F-E)(N-E)} & \dfrac{N}{(N-E)} \\ 0 & 0 & \dfrac{-V(F-N)}{(F-E)(N-E)} & \dfrac{V}{(N-E)} \end{pmatrix}$$

in the notation of Section 3.3.1.3.5, and $\boldsymbol{T}^{-1}$ was given in Section 3.3.1.3.8. If orthographic projection is being used $(E = -\infty)$ then $\boldsymbol{U}^{-1}$ simplifies to

$$\boldsymbol{U}'^{-1} \simeq \begin{pmatrix} \dfrac{R-L}{2} & 0 & 0 & \dfrac{R+L}{2} \\ 0 & \dfrac{T-B}{2} & 0 & \dfrac{T+B}{2} \\ 0 & 0 & F-N & N \\ 0 & 0 & 0 & V \end{pmatrix}.$$

Each of these inverse matrices may be suitably scaled to suit the word length of the machine [Section 3.3.1.1.2 (iii)].

Having determined the end points of one sight line in data space the viewing transformation $\boldsymbol{T}$ may then be changed and the required position marked again through the screen in the new orientation. Each such operation generates a pair of points in data space, expressed in homogeneous form, with a variety of values for the fourth coordinate. Each such point must then be converted to three dimensions in the form $(X/W,\ Y/W,\ Z/W)$, and for each sight line any (three-dimensional) point $\mathbf{p}_A$ on the line and the direction $\mathbf{q}_A$ of the line are established. For each sight line a rank 2 projector matrix $\boldsymbol{M}_A$ of order 3 is formed as

$$\boldsymbol{M}_A = \boldsymbol{I} - \mathbf{q}_A\mathbf{q}_A^T/\mathbf{q}_A^T\mathbf{q}_A$$

and the best point of intersection of the sight lines is given by

$$\Big(\sum_a \boldsymbol{M}_a\Big)^{-1}\Big(\sum_a \boldsymbol{M}_a\mathbf{p}_a\Big),$$

to which three-vector a fourth coordinate of unity may be applied.

3.3.1.3.10. *The three-axis joystick*

The three-axis joystick is a device which depends on compound transformations for its exploitation. As it is usually mounted it consists of a vertical shaft, mounted at its lower end, which can rotate about its own length (the Y axis of display space, Section 3.3.1.3.1), its angular setting, φ, being measured by a shaft encoder in its mounting. At the top of this shaft is a knee-joint coupling to a second shaft. The first angle φ is set to zero when the second shaft is in some selected direction, *e.g.* normal to the screen and towards the viewer, and goes positive if the second shaft is moved clockwise when seen from above. The knee joint itself contains a shaft encoder, providing an angle, ψ, which may be set to zero when the second shaft is horizontal and goes positive when its free end is raised. A knob on the tip of the second shaft can then rotate about an axis along the second shaft, driving a third shaft encoder providing an angle θ. The device may then be used to produce a rotation of the object on the screen about an axis parallel to the second shaft through an angle given by the knob. The necessary transformation is then

$$\begin{aligned} \boldsymbol{R} = &\begin{pmatrix} \cos\varphi & 0 & -\sin\varphi \\ 0 & 1 & 0 \\ \sin\varphi & 0 & \cos\varphi \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{pmatrix} \\ &\times \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{pmatrix} \\ &\times \begin{pmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{pmatrix} \end{aligned}$$

which is

$$\begin{pmatrix} c^2\psi s^2\varphi + (1-c^2\psi s^2\varphi)c\theta & -s\psi c\psi s\varphi(1-c\theta) - c\psi c\varphi s\theta & -c^2\psi s\varphi c\varphi(1-c\theta) + s\psi s\theta \\ -s\psi c\psi s\varphi(1-c\theta) + c\psi c\varphi s\theta & s^2\psi + c^2\psi c\theta & s\psi c\psi c\varphi(1-c\theta) + c\psi s\varphi s\theta \\ -c^2\psi s\varphi c\varphi(1-c\theta) - s\psi s\theta & s\psi c\psi c\varphi(1-c\theta) - c\psi s\varphi s\theta & c^2\psi c^2\varphi + (1-c^2\psi c^2\varphi)c\theta \end{pmatrix}$$

in which cos and sin are abbreviated to c and s, which is the standard form with $l = -\cos\psi\sin\varphi$, $m = \sin\psi$, $n = \cos\psi\cos\varphi$.

3.3.1.3.11. *Other useful rotations*

If rotations in display space are to be controlled by trackerball or tablet then there are two measures available, an x and a y, which can define an axis of rotation in the plane of the screen and an angle θ. If x and y are suitably scaled coordinates of a pen on a tablet then the rotation

$$\begin{pmatrix} \dfrac{y^2+x^2c}{x^2+y^2} & \dfrac{-xy(1-c)}{x^2+y^2} & x\sqrt{x^2+y^2} \\ \dfrac{-xy(1-c)}{x^2+y^2} & \dfrac{x^2+y^2c}{x^2+y^2} & y\sqrt{x^2+y^2} \\ -x\sqrt{x^2+y^2} & -y\sqrt{x^2+y^2} & c \end{pmatrix}$$

with $c = \sqrt{1-(x^2+y^2)^2}$ is about an axis in the xy plane (*i.e.* the screen face) normal to (x, y) and with $\sin\theta = x^2 + y^2$. Applied repetitively this gives a quadratic velocity characteristic. Similarly, if an atom at (x, y, z, w) in display space is to be brought onto the z axis by a rotation with its axis in the xy plane the necessary matrix, in homogeneous form, is

$$\begin{pmatrix} \dfrac{x^2z+y^2r}{x^2+y^2} & \dfrac{-xy(r-z)}{x^2+y^2} & -x & 0 \\ \dfrac{-xy(r-z)}{x^2+y^2} & \dfrac{x^2r+y^2z}{x^2+y^2} & -y & 0 \\ x & y & z & 0 \\ 0 & 0 & 0 & r \end{pmatrix}$$

with $r = \sqrt{x^2+y^2+z^2}$.

3.3.1.3.12. *Symmetry*

In Section 3.3.1.1.1 it was pointed out that it is usual to express coordinates for graphical purposes in Cartesian coordinates in ångström units or nanometres. Symmetry, however, is best expressed in crystallographic fractional coordinates. If a molecule, with Cartesian coordinates, is being displayed, and a symmetry-related neighbour is also to be displayed, then the data-space coordinates must be multiplied by

$$\begin{pmatrix} \mathbf{W} & \mathbf{T} \\ \mathbf{0}^T & W \end{pmatrix} \begin{pmatrix} \boldsymbol{M} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathscr{S} \begin{pmatrix} \boldsymbol{M}^{-1} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{W} & -\mathbf{T} \\ \mathbf{0}^T & W \end{pmatrix},$$

where

$$\begin{pmatrix} \mathbf{T} \\ W \end{pmatrix}$$

are the data-space coordinates of the crystallographic origin, $\boldsymbol{M}$ and $\boldsymbol{M}^{-1}$ are as in Section 3.3.1.1.1 and $\mathscr{S}$ is a crystallographic symmetry operator in homogeneous coordinates, expressed relative to the same crystallographic origin.

For example, in $P2_1$ with the origin on the screw dyad along **b**,

$$\mathscr{S} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and

$$\begin{pmatrix} \boldsymbol{M} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathscr{S} \begin{pmatrix} \boldsymbol{M}^{-1} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{1}{2}b \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

$\mathscr{S}$ comprises a proper or improper rotational partition, $\boldsymbol{S}$, in the upper-left 3×3 in the sense that $\boldsymbol{MSM}^{-1}$ is orthogonal, and with the associated fractional lattice translation in the last column, with the last row always consisting of three zeros and 1 at the 4, 4 position. See *IT* A (1983, Chapters 5.3 and 8.1) for a fuller discussion of symmetry using augmented (*i.e.* 4×4) matrices.

3.3.1.4. *Modelling transformations*

The two sections under this heading are concerned only with the graphical aspects of conformational changes. Determination of such changes is considered under Section 3.3.2.2.

3.3.1.4.1. *Rotation about a bond*

It is a common requirement in molecular modelling to be able to rotate part of a molecule relative to the remainder about a bond between two atoms.

If four atoms are bonded 1–2–3–4 then the dihedral angle in the bond 2–3 is zero if the four atoms are *cis* planar, and a rotation in the 2–3 bond is, by convention (IUPAC–IUB Commission on Biochemical Nomenclature, 1970), positive if, when looking along the 2–3 bond, the far end rotates clockwise relative to the near end. This is valid for either viewing direction. This sign convention, when applied to the $\boldsymbol{R}$ matrix of Section 3.3.1.2.1, leads to the following statement.

If one of the two atoms is selected as the near atom and the direction cosines are those of the vector from the near atom to the far atom, and if the matrix is to rotate material attached to the far atom (with the reference axes fixed), then a positive rotation in the foregoing sense is generated by a positive θ.

Rotation about a bond normally involves compounding $\boldsymbol{R}$ with translations in the manner of Section 3.3.1.3.8.

3.3.1.4.2. *Stacked transformations*

A flexible molecule may require to be drawn in any of a number of conformations which are related to one another by, for example, rotations about single bonds, changes of bond angles or changes of bond lengths, all of which changes may be brought about by the application of suitable homogeneous transformations during the drawing of the molecule (Section 3.3.1.3.8). With suitable organization, this may be done without necessarily altering the coordinates of the atoms in the coordinate list, only the transformations being manipulated during drawing.

The use of transformations in the manner shown below is straightforward for simply connected structures or structures containing only rigid rings. Flexible rings may be similarly handled provided that the matrices employed are consistent with the consequential constraints as described in Section 3.3.2.2.1, though this requirement may make real-time folding of flexible rings difficult.

Any simply connected structure may be organized as a tree with a node at each branch point and with an arbitrary number of sites of

references