International Tables for Crystallography (2006). Vol. B, Section 3.3.1.3.7, pp. 370–371.

result is

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

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

$$\boldsymbol{U} = \boldsymbol{U}'\boldsymbol{P},$$

where $\boldsymbol{U}$ and $\boldsymbol{U}'$ are as above and $\boldsymbol{P}$ is a perspective transformation given by

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

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

Other factorizations of $\boldsymbol{U}$ are possible, for example

$$\boldsymbol{U} = \boldsymbol{U}''\boldsymbol{P}'$$

with

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

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

which renders $\boldsymbol{P}'$ independent of all six boundary planes, but $\boldsymbol{U}''$ is no longer independent of E. It is not possible to factorize $\boldsymbol{U}$ 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 $E \to -\infty$, $\boldsymbol{U}'' \to \boldsymbol{U}'$, $\boldsymbol{P}$ and $\boldsymbol{P}' \to -\boldsymbol{I}E \simeq \boldsymbol{I}$.

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 $\boldsymbol{U}$ matrix of the previous section. However, $\boldsymbol{U}$ may be factorized according to

$$\boldsymbol{U} = \boldsymbol{U}'''\boldsymbol{S}$$

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

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

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

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

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

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 $\sin 3°$.

The distinction in principle between the true $\boldsymbol{S}$ and the rotational approximation is that with the true $\boldsymbol{S}$ 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, $\boldsymbol{V}$, is therefore needed to position the picture in the requisite part of the screen.

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

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

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

or

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

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,

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

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 $\boldsymbol{R}^T\boldsymbol{R}_1\boldsymbol{R}$