

Perspective Mappings

David Eberly

Geometric Tools, LLC

<http://www.geometrictools.com/>

Copyright © 1998-2012. All Rights Reserved.

Created: August 16, 2011

Last Modified: November 3, 2012

Contents

1	Introduction	2
2	Perspective Mapping from a Quadrilateral to a Square	2
3	Perspective Mapping from a Cuboid to a Cube	5
4	Perspective Mapping from a Hypercuboid to a Hypercube	8

1 Introduction

This document combines—and supersedes—three others

- *Perspective Mapping Between Two Convex Quadrilaterals* (file Perspective.pdf). A brief summary of the fractional linear transformation that map a convex quadrilateral to a unit square. The inverse of such a transformation is easy to compute, so in fact the extension to a mapping between two convex quadrilaterals is simple to construct.
- *Perspective Mappings Between Quadrilaterals* (file Perspective2.pdf). A 3D geometric motivation for the perspective projection of one convex quadrilateral onto another. The idea is to place one quadrilateral in the xy -plane and set z -values to zero. One of the vertices is translated to the origin. The other quadrilateral has a vertex translated to the origin. The quadrilateral is placed on some plane through the origin. This plane and an eyepoint for the perspective projection of the rotated quadrilateral onto the other is constructed, which leads to a 2D-to-2D mapping fractional linear transformation between the quadrilaterals.
- *Perspective Mappings Between Cuboids* (file Perspective3.pdf). A 4D geometric motivation for the perspective projection of one cuboid (convex polyhedron with 6 planar faces) onto another cuboid. This is the extension of the idea discussed in *Perspective Mappings Between Quadrilaterals*.

The current document simplifies the constructions. It also shows the extension to higher dimensions (perspective projection of hypercuboids to hypercuboids). It is sufficient to construct the mapping from the canonical hypercube (axis-aligned, vertex at the origin, edge of length one) to a hypercuboid. The mapping is a fractional linear transformation, which is easy invertible, so you easily obtain the mapping from the hypercuboid to a hypercube.

2 Perspective Mapping from a Quadrilateral to a Square

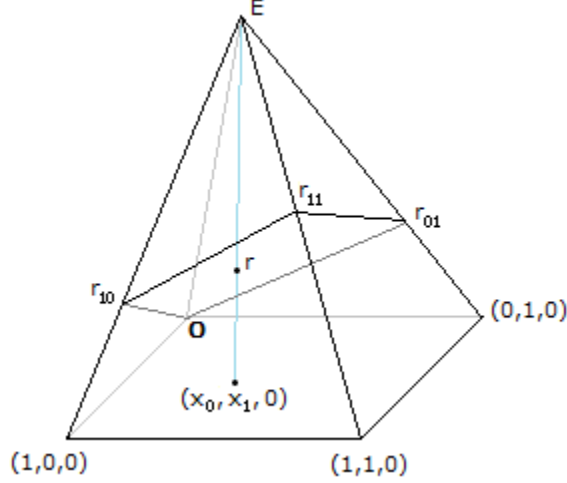
Let the square vertices be $(0,0)$, $(1,0)$, $(1,1)$, and $(0,1)$. The corresponding quadrilateral vertices are \mathbf{q}_{00} , \mathbf{q}_{10} , \mathbf{q}_{11} , and \mathbf{q}_{01} . To map the quadrilateral to the square, consider the problem as a perspective projection in three dimensions. The view plane is $y_2 = 0$ and the eyepoint is $\mathbf{E} = (e_0, e_1, e_2)$ where $e_2 \neq 0$ (the eyepoint is not on the view plane). The perspective projection of a point $\mathbf{r} = (r_0, r_1, r_2)$ to the view plane is the point $\mathbf{p} = (x_0, x_1, 0)$, which lies on the ray originating at \mathbf{E} and contains \mathbf{r} . The intersection of the ray with the plane is the projection point: $\mathbf{p} = \mathbf{E} + t(\mathbf{r} - \mathbf{E})$ for some t . Some algebra leads to $t = e_2/(e_2 - r_2)$, $x_0 = (e_2 r_0 - e_0 r_2)/(e_2 - r_2)$, and $x_1 = (e_2 r_1 - e_1 r_2)/(e_2 - r_2)$.

Embed the square vertices in the view plane as $(i_0, i_1, 0)$ for i_0 and i_1 in $\{0,1\}$. Embed the quadrilateral vertices as $(\mathbf{q}_{i_0 i_1} - \mathbf{q}_{00}, 0)$, which translates one of the vertices to the origin. Rotate the embedded quadrilateral vertices so that they lie in a plane containing the origin $\mathbf{0}$ and having normal vector $\mathbf{N} = (n_0, n_1, n_2)$. Let the rotated points be denoted $\mathbf{r}_{i_0 i_1} = R(\mathbf{q}_{i_0 i_1} - \mathbf{q}_{00}, 0)$, where R is a rotation matrix corresponding to the normal; thus, $\mathbf{N} \cdot \mathbf{r}_{i_0 i_1} = 0$. The problem is to construct \mathbf{E} and \mathbf{N} so that

$$\mathbf{E} + t_{i_0 i_1} (\mathbf{r}_{i_0 i_1} - \mathbf{E}) = (i_0, i_1, 0), \quad i_j \in \{0,1\} \quad (1)$$

for some scalars $t_{i_0 i_1}$. Equation (1) says that the embedded quadrilateral vertices are projected perspectively to the square vertices. Figure 2.1 shows the geometric configuration.

Figure 2.1 Projection of a quadrilateral onto a square. The rotated quadrilateral points $\mathbf{r}_{i_0 i_1}$ are projected to the square points $(i, j, 0)$. The figure shows a rotated quadrilateral interior point projected to a square interior point.



The vector $\mathbf{q}_{11} - \mathbf{q}_{00}$ can be represented as a linear combination of two quadrilateral edges, namely,

$$\mathbf{q}_{11} - \mathbf{q}_{00} = a_0 (\mathbf{q}_{10} - \mathbf{q}_{00}) + a_1 (\mathbf{q}_{01} - \mathbf{q}_{00}) \quad (2)$$

The coefficients a_0 and a_1 are constructed as the solution of two linear equations in two unknowns. These coefficients show up in the equations for the perspective projection. The convexity of the quadrilateral guarantees that $a_0 \geq 0$, $a_1 \geq 0$, and $a_0 + a_1 > 1$. Rotation preserves the relative positions of the vertices, so we know that

$$\mathbf{r}_{11} = a_0 \mathbf{r}_{10} + a_1 \mathbf{r}_{01} \quad (3)$$

Equation (1) contains four relationships,

$$\begin{aligned} \mathbf{E} + t_{00} (\mathbf{r}_{00} - \mathbf{E}) &= (0, 0, 0) \\ \mathbf{E} + t_{10} (\mathbf{r}_{10} - \mathbf{E}) &= (1, 0, 0) \\ \mathbf{E} + t_{01} (\mathbf{r}_{01} - \mathbf{E}) &= (0, 1, 0) \\ \mathbf{E} + t_{11} (\mathbf{r}_{11} - \mathbf{E}) &= (1, 1, 0) \end{aligned} \quad (4)$$

It must be that $t_{00} = 1$, because $\mathbf{r}_{00} = \mathbf{0}$. Dotting the last three equations with the normal vector leads to

$$(1 - t_{10})\mathbf{N} \cdot \mathbf{E} = n_0, \quad (1 - t_{01})\mathbf{N} \cdot \mathbf{E} = n_1, \quad (1 - t_{11})\mathbf{N} \cdot \mathbf{E} = n_0 + n_1$$

Geometrically, it is clear that $\mathbf{N} \cdot \mathbf{E} \neq 0$, so these three equations imply

$$t_{11} = t_{10} + t_{01} - 1 \quad (5)$$

Substituting equations (3) and (5) into the last line of equation (4) produces

$$(2 - t_{10} - t_{01})\mathbf{E} + (t_{10} + t_{01} - 1)(a_0\mathbf{r}_{10} + a_1\mathbf{r}_{01}) = (1, 1, 0) \quad (6)$$

Adding the second and third lines of equation (4),

$$(2 - t_{10} - t_{01})\mathbf{E} + t_{10}\mathbf{r}_{10} + t_{01}\mathbf{r}_{01} = (1, 1, 0) \quad (7)$$

Subtracting equation (7) from (6), we have

$$[a_0(t_{10} + t_{01} - 1) - t_{10}]\mathbf{r}_{10} + [a_1(t_{10} + t_{01} - 1) - t_{01}]\mathbf{r}_{01} = \mathbf{0} \quad (8)$$

The linear independence of \mathbf{r}_{10} and \mathbf{r}_{01} guarantees that the coefficients in equation (8) must both be zero. We now have two linear equations in the two unknowns t_{10} and t_{01} . These are easily solved, and together with equation (5) produce

$$t_{00} = 1, \quad t_{10} = \frac{a_0}{a_0 + a_1 - 1}, \quad t_{01} = \frac{a_1}{a_0 + a_1 - 1}, \quad t_{11} = \frac{1}{a_0 + a_1 - 1} \quad (9)$$

It turns out that computing t_{11} explicitly in terms of a_0 and a_1 is not necessary to construct the fractional linear transformations.

At this time we may attempt to construct \mathbf{E} and \mathbf{N} , but this is not necessary to actually construct the perspective projection. Instead, consider a rotated quadrilateral point \mathbf{r} that is projected to a square point $(x_0, x_1, 0)$, as illustrated in Figure 2.1. We may write \mathbf{r} as a linear combination of two rotated quadrilateral edges,

$$\mathbf{r} = y_0\mathbf{r}_{10} + y_1\mathbf{r}_{01} \quad (10)$$

The ray equation for the pair of points is

$$(x_0, x_1, 0) = \mathbf{E} + t(\mathbf{r} - \mathbf{E}) = \mathbf{E} + t(y_0\mathbf{r}_{10} + y_1\mathbf{r}_{01} - \mathbf{E}) \quad (11)$$

for some scalar t . Substituting our now known values for $t_{i_0i_1}$ into the second and third lines of equation (4), solving for \mathbf{r}_{10} and \mathbf{r}_{01} , replacing in equation (11), and grouping like terms produces

$$\left(1 - \frac{ty_0(1 - t_{10})}{t_{10}} - \frac{ty_1(1 - t_{01})}{t_{01}} - t\right)\mathbf{E} + \left(\frac{ty_0}{t_{10}} - x_0\right)(1, 0, 0) + \left(\frac{ty_1}{t_{01}} - x_1\right)(0, 1, 0) = (0, 0, 0) \quad (12)$$

By linear independence of \mathbf{E} , $(1, 0, 0)$, and $(0, 1, 0)$, the coefficients in equation (12) must all be zero. This leads to the fractional linear transformation that maps the quadrilateral to the square

$$(x_0, x_1) = \frac{(a_1(a_0 + a_1 - 1)y_0, a_0(a_0 + a_1 - 1)y_1)}{a_0a_1 + a_1(a_1 - 1)y_0 + a_0(a_0 - 1)y_1} \quad (13)$$

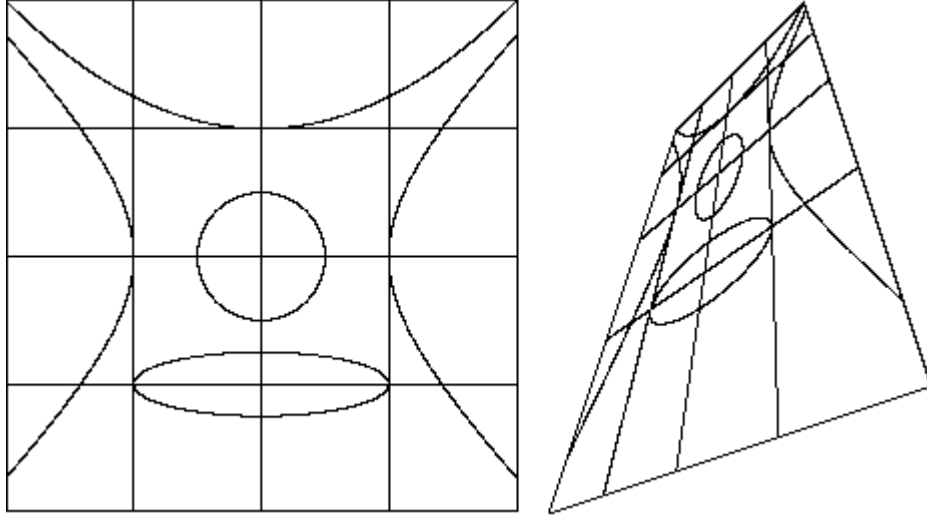
The fractional linear transformation that maps the square to the quadrilateral is

$$(y_0, y_1) = \frac{(a_0x_0, a_1x_1)}{(a_0 + a_1 - 1) + (1 - a_1)x_0 + (1 - a_0)x_1} \quad (14)$$

Example. Intuitively, perspective transformations map lines to lines. They also map conic sections to conic sections. The proof is straightforward. Let the perspective transformation be $y_0 = (a_0x_0 + a_1x_1 + a_2)/(c_0x_0 + c_1x_1 + c_2)$ and $y_1 = (b_0x_0 + b_1x_1 + b_2)/(c_0x_0 + c_1x_1 + c_2)$. The inverse transformation is of the same form, so

it suffices to show a conic section in (y_0, y_1) coordinates satisfying $Ay_0^2 + By_0y_1 + Cy_1^2 + Dy_0 + Ey_1 + F = 0$ is mapped to a conic section in (x_0, x_1) coordinates satisfying $\bar{A}x_0^2 + \bar{B}x_0x_1 + \bar{C}x_1^2 + \bar{D}x_0 + \bar{E}x_1 + \bar{F} = 0$. Substituting the formulas for y_0 and y_1 into the quadratic equation, multiplying by $(c_0x_0 + c_1x_1 + c_2)^2$, expanding the products, and grouping the appropriate terms yields a quadratic in x_0 and x_1 . The left image in Figure 2.2 shows a square containing several conic sections. The top curve is a parabola, the left and right curves are hyperbolas, the center curve is a circle, and the bottom curve is an ellipse. The grid consists of straight lines. The right image in Figure 2.2 shows a perspective mapping of these conic sections.

Figure 2.2 Projection of conic sections in a square to conic sections in a quadrilateral.



3 Perspective Mapping from a Cuboid to a Cube

The ideas of the previous section extend to 3D. A cuboid is a convex polyhedron that has 6 planar faces. For example, a typical view frustum is a cuboid. A cube is a special case of a cuboid, all six faces are squares, and opposite faces are parallel. Let the vertices of the cuboid be denoted $\mathbf{q}_{i_0i_1i_2}$ where $i_j \in \{0, 1\}$. The canonical cube vertices are (i_0, i_1, i_2) .

To project perspectively the cuboid to the cube, we can embed the problem in 4D. The view hyperplane is $y_3 = 0$ and the eyepoint is $\mathbf{E} = (e_0, e_1, e_2, e_3)$ where $e_3 \neq 0$. The cuboid is translated to the origin and then rotated into a hyperplane with normal $\mathbf{N} = (n_0, n_1, n_2, n_3)$. The rotation matrix is R and the rotated and translated vertices are $\mathbf{r}_{i_0i_1i_2} = R(\mathbf{q}_{i_0i_1i_2} - \mathbf{q}_{000}, 0)$. Our goal is that

$$\mathbf{E} + t_{i_0i_1i_2} (\mathbf{r}_{i_0i_1i_2} - \mathbf{E}) = (i_0, i_1, i_2, 0), \quad i_j \in \{0, 1\} \quad (15)$$

for some scalars $t_{i_0i_1i_2}$.

The vector $\mathbf{q}_{111} - \mathbf{q}_{000}$ can be represented as a linear combination of three cuboid edges, namely,

$$\mathbf{q}_{111} - \mathbf{q}_{000} = a_0 (\mathbf{q}_{100} - \mathbf{q}_{000}) + a_1 (\mathbf{q}_{010} - \mathbf{q}_{000}) + a_2 (\mathbf{q}_{001} - \mathbf{q}_{000}) \quad (16)$$

The coefficients a_0 , a_1 , and a_2 are constructed as the solution of three linear equations in three unknowns. These coefficients show up in the equations for the perspective projection. The convexity of the cuboid guarantees that $a_0 \geq 0$, $a_1 \geq 0$, $a_2 \geq 0$, and $a_0 + a_1 + a_2 > 1$. Rotation preserves the relative positions of the vertices, so we know that

$$\mathbf{r}_{111} = a_0 \mathbf{r}_{100} + a_1 \mathbf{r}_{010} + a_2 \mathbf{r}_{001} \quad (17)$$

Dotting the relationships in equation (15) with the normal vector produces

$$\begin{aligned} (1 - t_{100})\mathbf{N} \cdot \mathbf{E} &= n_0, & (1 - t_{110})\mathbf{N} \cdot \mathbf{E} &= n_0 + n_1 \\ (1 - t_{010})\mathbf{N} \cdot \mathbf{E} &= n_1, & (1 - t_{101})\mathbf{N} \cdot \mathbf{E} &= n_0 + n_2 \\ (1 - t_{001})\mathbf{N} \cdot \mathbf{E} &= n_2, & (1 - t_{011})\mathbf{N} \cdot \mathbf{E} &= n_1 + n_2 \\ & & (1 - t_{111})\mathbf{N} \cdot \mathbf{E} &= n_0 + n_1 + n_2 \end{aligned}$$

Because $\mathbf{N} \cdot \mathbf{E} \neq 0$, we obtain

$$t_{110} = t_{100} + t_{010} - 1, \quad t_{101} = t_{100} + t_{001} - 1, \quad t_{011} = t_{010} + t_{001} - 1, \quad t_{111} = t_{100} + t_{010} + t_{001} - 2 \quad (18)$$

Substituting the relationship for t_{111} from equation (18) into the equation of (15) involving t_{111} leads to

$$(3 - t_{100} - t_{010} - t_{001})\mathbf{E} + (t_{100} + t_{010} + t_{001} - 2)(a_0 \mathbf{r}_{100} + a_1 \mathbf{r}_{010} + a_2 \mathbf{r}_{001}) = (1, 1, 1) \quad (19)$$

Adding the relationships of equation (15) involving t_{100} , t_{010} , and t_{001} produces

$$(3 - t_{100} - t_{010} - t_{001})\mathbf{E} + t_{100}\mathbf{r}_{100} + t_{010}\mathbf{r}_{010} + t_{001}\mathbf{r}_{001} = (1, 1, 1) \quad (20)$$

Subtracting equation (20) from (19), we have

$$\begin{aligned} [a_0(t_{100} + t_{010} + t_{001} - 2) - t_{100}]\mathbf{r}_{100} &+ [a_1(t_{100} + t_{010} + t_{001} - 2) - t_{010}]\mathbf{r}_{010} \\ &+ [a_2(t_{100} + t_{010} + t_{001} - 2) - t_{001}]\mathbf{r}_{001} = \mathbf{0} \end{aligned} \quad (21)$$

The linear independence of \mathbf{r}_{100} , \mathbf{r}_{010} , and \mathbf{r}_{001} guarantees that the coefficients in equation (21) must all be zero. We now have three linear equations in three unknowns t_{100} , t_{010} , and t_{001} . These are easily solved, and together with previous equations produce

$$\begin{aligned} t_{000} &= 1, & t_{110} &= \frac{+a_0+a_1-a_2+1}{a_0+a_1+a_2-1} \\ t_{100} &= \frac{2a_0}{a_0+a_1+a_2-1}, & t_{101} &= \frac{+a_0-a_1+a_2+1}{a_0+a_1+a_2-1} \\ t_{010} &= \frac{2a_1}{a_0+a_1+a_2-1}, & t_{011} &= \frac{-a_0+a_1+a_2+1}{a_0+a_1+a_2-1} \\ t_{001} &= \frac{2a_2}{a_0+a_1+a_2-1}, & t_{111} &= \frac{+a_0+a_1+a_2+1}{a_0+a_1+a_2-1} \end{aligned} \quad (22)$$

It turns out that computing t_{110} , t_{101} , t_{011} , and t_{111} explicitly in terms of a_0 , a_1 , and a_2 is not necessary to construct the fractional linear transformations.

As in the 2D problem, we can construct the fractional linear transformations for the projection without explicitly constructing \mathbf{E} and \mathbf{N} . We may write a rotated interior cuboid point \mathbf{r} as

$$\mathbf{r} = y_0 \mathbf{r}_{100} + y_1 \mathbf{r}_{010} + y_2 \mathbf{r}_{001} \quad (23)$$

The ray equation for the pair of points is

$$(x_0, x_1, x_2, 0) = \mathbf{E} + t(\mathbf{r} - \mathbf{E}) = \mathbf{E} + t(y_0\mathbf{r}_{100} + y_1\mathbf{r}_{010} + y_2\mathbf{r}_{001} - \mathbf{E}) \quad (24)$$

for some scalar t . Substituting the values for $t_{i_0i_1i_2}$ into equation (15); solving for \mathbf{r}_{100} , \mathbf{r}_{010} , and \mathbf{r}_{001} ; replacing in equation (24), and grouping like terms produces

$$\begin{aligned} & \left(1 - \frac{ty_0(1-t_{100})}{t_{100}} - \frac{ty_1(1-t_{010})}{t_{010}} - \frac{ty_2(1-t_{001})}{t_{001}}\right) \mathbf{E} \\ & + \left(\frac{ty_0}{t_{100}} - x_0\right) (1, 0, 0, 0) + \left(\frac{ty_1}{t_{010}} - x_1\right) (0, 1, 0, 0) + \left(\frac{ty_2}{t_{001}} - x_2\right) (0, 0, 1, 0) = (0, 0, 0, 0) \end{aligned} \quad (25)$$

By linear independence of \mathbf{E} , $(1, 0, 0, 0)$, $(0, 1, 0, 0)$, and $(0, 0, 1, 0)$, the coefficients in equation (25) must all be zero. This leads to the fractional linear transformation that maps the cube to the cuboid,

$$(y_0, y_1, y_2) = \frac{2(a_0x_0, a_1x_1, a_2x_2)}{d_0x_0 + d_1x_1 + d_2x_2 + d_3} \quad (26)$$

where $d_0 = a_0 - a_1 - a_2 + 1$, $d_1 = -a_0 + a_1 - a_2 + 1$, $d_2 = -a_0 - a_1 + a_2 + 1$, and $d_3 = a_0 + a_1 + a_2 + 1$. The fractional linear transformation that maps the cube to the cuboid is

$$(x_0, x_1, x_2) = \frac{d_3(a_1a_2y_0, a_0a_2y_1, a_0a_1y_2)}{2a_0a_1a_2 - a_1a_2d_0y_0 - a_0a_2d_1y_1 - a_0a_1d_2y_2} \quad (27)$$

Example. When drawing a view volume in 3D graphics, one typically creates a view frustum with near plane, far plane, left plane, right plane, bottom plane, and top plane. The vertices of geometric primitives are transformed from model space to world space, then from world space to view space (camera space). The view-space vertices are finally transformed to clip space (homogeneous coordinates) by a 4×4 perspective projection matrix. The rasterizer does the clipping in homogeneous coordinates and then performs the perspective divide to obtain the normalized window coordinates $(x, y) \in [-1, 1]^2$ and normalized depth $z \in [0, 1]$ (for DirectX; OpenGL maps depth to $[-1, 1]$). The viewport is the full rectangular window, but it may be chosen to be an subwindow specified as an axis-aligned rectangle.

It is possible to specify a viewport that is a convex quadrilateral (nonrectangular). Let this viewport have vertices in camera coordinates, $\mathbf{q}_{i_0i_1}$ for $i_j \in \{0, 1\}$, and let them be counterclockwise oriented. This polygon acts as the near face of a cuboidal view volume. The quadrilateral may be extruded toward the far plane of the view frustum, thus constructing the far face of the cuboidal view volume. Let the camera forward direction be \mathbf{D} . Let n be the near-plane distance from the eyepoint and let f be the far-plane distance from the eyepoint. The viewport vertex $\mathbf{q}_{i_0i_1}$ is on the near plane, so $\mathbf{D} \cdot \mathbf{q}_{i_0i_1} = n$. The corresponding far-plane vertex is $\mathbf{q}_{i_0i_1} = (f/n)\mathbf{q}_{i_0i_1}$ so that $\mathbf{D} \cdot \mathbf{q}_{i_0i_1} = f$.

We may construct a projection matrix that maps the cuboidal view volume to the cube $[-1, 1]^2 \times [0, 1]$. Define $\mathbf{U}_0 = \mathbf{q}_{100} - \mathbf{q}_{000}$, $\mathbf{U}_1 = \mathbf{q}_{010} - \mathbf{q}_{000}$, and $\mathbf{U}_2 = \mathbf{q}_{001} - \mathbf{q}_{000}$. Define the matrix $M = [\mathbf{U}_0 \ \mathbf{U}_1 \ \mathbf{U}_2]$ whose columns are the specified vectors. Define \mathbf{a} to be the column vector whose components are the a_i in the fractional linear transformations. Then

$$\mathbf{a} = M^{-1}(\mathbf{q}_{111} - \mathbf{q}_{000})$$

In homogeneous block form,

$$\begin{bmatrix} \mathbf{a} \\ 1 \end{bmatrix} = \begin{bmatrix} M & \mathbf{q}_{000} \\ \mathbf{0}^T & 1 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{q}_{111} \\ 1 \end{bmatrix} = \begin{bmatrix} M^{-1} & -M^{-1}\mathbf{q}_{000} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{q}_{111} \\ 1 \end{bmatrix} = \begin{bmatrix} M^{-1}(\mathbf{q}_{111} - \mathbf{q}_{000}) \\ 1 \end{bmatrix}$$

Define H to be the homogeneous matrix whose upper-left block is the matrix M and whose translation component is the vector \mathbf{q}_{000} .

The projection of equation (27) maps the cuboidal volume to the canonical cube $[0, 1]^3$. We need to perform one additional scaling and translation to map (x_0, x_1) from $[0, 1]^2$ to $[-1, 1]^2$. With this additional transformation, the cuboid-to-cube mapping may be written in homogeneous form as

$$P = \begin{bmatrix} \frac{a_2}{a_0} (2d_3 + d_0) & \frac{a_2}{a_1} d_1 & d_2 & -2a_2 \\ \frac{a_2}{a_0} d_0 & \frac{a_2}{a_1} (2d_3 + d_1) & d_2 & -2a_2 \\ 0 & 0 & d_3 & 0 \\ -\frac{a_2}{a_0} d_0 & -\frac{a_2}{a_1} d_1 & -d_2 & 2a_2 \end{bmatrix}$$

where the output (y_0, y_1, y_2, w) is processed by the rasterizer in the usual manner, performing the perspective divide by w to obtain the cube points.

The projection matrix from camera space to clip space is therefore PH^{-1} . The matrix H^{-1} maps camera coordinates to cuboid coordinates relative to three linearly independent edges of the cuboid. These points are then perspectively projected by P into the cube.

4 Perspective Mapping from a Hypercuboid to a Hypercube

The ideas in the previous sections generalize to higher dimensions. Let the d -dimensional hypercuboid have vertices \mathbf{q}_I , where $I = (i_1, i_2, \dots, i_d)$ is a multiindex with components $i_j \in \{0, 1\}$. The canonical hypercube vertices are \mathbf{I} , with bold face used to distinguish between vectors and multiindices.

The view hyperplane is $y_d = 0$ and the eyepoint is $\mathbf{E} = (e_1, e_2, \dots, e_d)$ with $e_d \neq 0$. The hypercuboid is translated to the origin and then rotated into a hyperplane with normal $\mathbf{N} = (n_1, n_2, \dots, n_d)$. Let R be the corresponding rotation matrix and define the rotated hypercuboid vertices by $\mathbf{r}_I = R(\mathbf{q}_I - \mathbf{q}_O)$, where O is the multiindex of all zeros. Our goal is that

$$\mathbf{E} + t_I(\mathbf{r}_I - \mathbf{E}) = (\mathbf{I}, 0) \quad (28)$$

for some scalars t_I . Let U be the multiindex of all ones. The vector $\mathbf{q}_U - \mathbf{q}_O$ may be written as

$$\mathbf{q}_U - \mathbf{q}_O = \sum_{j=1}^d a_j (\mathbf{q}_{B_j} - \mathbf{q}_O) \quad (29)$$

where B_j is the multiindex with 0 in all components except for a 1 in component j . The coefficients are nonnegative and $\sum_{j=1}^d a_j > 0$. By linearity of the rotation matrix,

$$\mathbf{r}_U = \sum_{j=1}^d a_j \mathbf{r}_{B_j} \quad (30)$$

Let $I = (i_1, i_2, \dots, i_d)$ be a multiindex with at least one nonzero component. Suppose that I has m components that are 1; it is the case that $1 \leq m \leq d$. Let those components be i_{j_1} through i_{j_m} . Dotting the ray

equation with the normal vector, we have

$$(1 - t_I)\mathbf{N} \cdot \mathbf{E} = \sum_{k=1}^m n_{i_k} \quad (31)$$

Whenever $m \geq 2$, the t_I variable is related to t_{B_j} variables by summations. Specifically, $I = \sum_{k=1}^m i_k B_{i_k}$ and

$$(1 - t_I)\mathbf{N} \cdot \mathbf{E} = \sum_{k=1}^m n_{i_k} = \sum_{k=1}^m (1 - t_{B_{i_k}})\mathbf{N} \cdot \mathbf{E} \quad (32)$$

Therefore, once we determine the relationships between t_{B_i} and a_j , the t_I with 2 or more 1-valued indices are determined. It turns out that for the construction of the fractional linear transformations, we only need to know t_U explicitly in terms of the a_j ,

$$t_U = \sum_{i=1}^d t_{B_i} - (d - 1) \quad (33)$$

Substituting the relationship for t_U into equation (28) leads to

$$\left(d - \sum_{i=1}^d t_{B_i}\right)\mathbf{E} + \left(\sum_{j=1}^d t_{B_j} - (d - 1)\right)\left(\sum_{i=1}^d a_i \mathbf{r}_{B_i}\right) = \mathbf{U} \quad (34)$$

Adding the relationships of equation (28) involving the t_{B_i} produces

$$\left(d - \sum_{i=1}^d t_{B_i}\right)\mathbf{E} + \sum_{i=1}^d t_{B_i} \mathbf{r}_{B_i} = \mathbf{U} \quad (35)$$

Subtracting the two equations, we have

$$\sum_{i=1}^d \left[a_i \left(\sum_{j=1}^d t_{B_j} - (d - 1) \right) - t_{B_i} \right] \mathbf{r}_{B_i} = \mathbf{0} \quad (36)$$

The linear independence of the \mathbf{r}_{B_i} guarantees that all the coefficients in this equation are zero. This leads to an invertible linear system of d equations in the d unknowns t_{B_1} through t_{B_d} .

As in dimensions 2 and 3, we need not construct \mathbf{E} and \mathbf{N} explicitly. We may write a rotated interior hypercuboid point \mathbf{r} as

$$\mathbf{r} = \sum_{i=1}^d y_i \mathbf{r}_{B_i} \quad (37)$$

The ray equation for the pair of points is

$$(x_1, \dots, x_d, 0) = \mathbf{E} + t(\mathbf{r} - \mathbf{E}) = \mathbf{E} + t \left(\sum_{i=1}^d y_i \mathbf{r}_{B_i} - \mathbf{E} \right) \quad (38)$$

for some scalar t . Substituting the values for t_{B_i} into equation (28), solving for \mathbf{r}_{B_i} , replacing in equation (38), and grouping like terms produces

$$\left(1 - \sum_{i=1}^d \frac{ty_i(1 - t_{B_i})}{t_{B_i}}\right)\mathbf{E} + \sum_{i=1}^d \left(\frac{ty_i}{t_{B_i}} - x_i\right)(\mathbf{B}_i, 0) = \mathbf{0} \quad (39)$$

By linear independence of \mathbf{E} and the $(\mathbf{B}_i, 0)$, the coefficients in equation (39) must all be zero. This leads to the fractional linear transformation that maps the hypercube to the hypercuboid,

$$y_i = \frac{(d-1)a_i x_i}{\sum_{k=1}^d c_k x_k + c_{d+1}}, \quad 1 \leq i \leq d \quad (40)$$

where

$$c_k = \sum_{j=1}^d b_{kj} a_j + 1, \quad 1 \leq k \leq d; \quad c_{d+1} = \sum_{j=1}^d a_j - 1; \quad b_{kj} = \begin{cases} d-2, & k = j \\ -1, & k \neq j \end{cases} \quad (41)$$

The fractional linear transformation that maps the hypercuboid to the hypercube is

$$x_i = \frac{\left[\prod_{j=1}^d a_j \right] c_{d+1} (y_i / a_i)}{\left[\prod_{j=1}^d a_j \right] \left((d-1) - \sum_{j=1}^d c_j (y_j / a_j) \right)}, \quad 1 \leq i \leq d \quad (42)$$

The products of the a_j appear as if they cancel, but if any of the a_j are zero, you need to formally multiply through by the product and then evaluate the expression using the a_j .