February 1969

PERSPECTIVE TRANSFORMATIONS

By:

Peter E. Hart
Richard O. Duda

Artificial Intelligence Group
Technical Note No. 3

SRI Project No. 7494

This research was sponsored by the Advanced Research Projects Agency and the Rome Air Development Center under Contract F30602-69-C-0056.
INTRODUCTION

The purpose of this technical note is to set down the mathematics of perspective transformations, the natural first-order approximation to the picture-taking process. There seem to be just two general questions: (1) What is the image of a point in space seen by a camera, and (2) What is the ray in space corresponding to a given image point? The answers to these two questions, together with a number of applications, are given below.

DIRECT TRANSFORMATION

A perspective transformation is not linear, but it can be put in a linear form through the device of homogeneous coordinates. These are defined as follows (all vectors are column vectors, so the row vectors shown all have invisible transpose signs).

Let \( v = (x, y, z) \) be a real-world, i.e., three-dimensional vector. Then \( V = (X, Y, Z, W) = (Wx, Wy, Wz, W) \) represents the same physical point in homogeneous coordinates, where \( W \) is an arbitrary scale factor. Obviously, the real-world coordinates are obtained from the homogeneous coordinate vector simply by dividing the first three components by the last. Thus, the homogeneous vector \( kV = (kWx, kWy, kWz, kW) \) corresponds to the same real-world point as \( V \) itself. This, plus a judicious choice of coordinate systems, provides the necessary magic to express the perspective transformation in linear form.

* References are listed at the end of this technical note.
Figure 1 shows the geometry of a single point $v$ projected onto a picture plane. The distance $f$ is measured from the pinhole lens to the image plane (and is not the focal length of the lens). Notice that the coordinate system is attached to the camera so that a real-world point is expressed in camera coordinates. Later we will need a change of coordinates so that real-world points will be expressed in room coordinates (or some other global frame of reference) while picture points will still be expressed in camera coordinates.

The following linear transformation $P$ takes a world-point expressed in homogeneous camera coordinates to a picture point expressed in homogeneous camera coordinates, i.e., $P: \mathbf{V} \rightarrow \mathbf{V}_p$, where $\mathbf{V}_p$ is the image point expressed in homogeneous camera coordinates.

$$
P = \begin{bmatrix}
1 & & \\
& 1 & \\
& & 1 \\
& & 1/f
\end{bmatrix}
$$

(using column vectors)

For a given world point (in homogeneous camera coordinates) $\mathbf{V} = (Wx, Wy, Wz, W)$, so $P\mathbf{V} = (Wx, Wy, Wz, Wy/f + W) = \mathbf{V}_p$. Hence, the image point $\mathbf{v}_p$, in ordinary camera coordinates is

$$
\mathbf{v}_p = \left( \frac{fx}{f+y}, \frac{fy}{f+y}, \frac{fz}{f+y} \right).
$$

Since the picture plane is the plane $y = 0$, the middle coordinate $fy/(f+y)$ is slightly mysterious. We will ignore it for the moment and obtain the picture coordinates $(x_p, z_p)$ of an arbitrary point $(x, y, z)$ as

$$
x_p = \frac{fx}{f+y}; \quad z_p = \frac{fz}{f+y}.
$$
INVERSE TRANSFORMATION

Each point in the image plane corresponds to a ray in space determined by the point and the lens. The equation of this ray is found by using the inverse transformation \( P^{-1} \) in the following way.

\[
P^{-1}: \quad V_p = V
\]

\[
P^{-1} = \begin{bmatrix} 1 & 1 \\ -\frac{1}{f} & 1 \\ \end{bmatrix}
\]

For a given picture point \( V_p = (Wx_p, Wy_p, Wz_p, W) \)

\[
P^{-1}V_p = (Wx_p, Wy_p, Wz_p, -\frac{Wy_p}{f} + W) = V
\]

Thus

\[
v = \left( \frac{fx_p}{f-y_p}, \frac{fy_p}{f-y_p}, \frac{fz_p}{f-y_p} \right)
\]

For a fixed picture point \((x_p, z_p)\), the world-point \(v\) traces out some curve in space as \(y_p\) varies. The form of this curve is easily found by eliminating the parameter \(y_p\) from the three parametric equations

\[
x = \frac{fx_p}{f-y_p}
\]

\[
y = \frac{fy_p}{f-y_p}
\]
\[ z = \frac{fz_p}{f-y_p} \]

to obtain
\[ x = \frac{x_p(y+f)}{f} = \frac{x_z z_p}{z_p} \]

the equation of a straight line. What interesting points are on this line? We can notice that the lens point \((0, -f, 0)\) and the picture point \((x_p, 0, z_p)\) both satisfy the equations. Hence, the inverse perspective transformation gives the equation of the ray from the lens through the picture point and on into space. This is probably the clearest justification that the linear transformation \(P\) performs as advertised. We can also notice that \(y_p\) is monotonic in \(y\). Hence, for display purposes \(y_p\) can be used to solve the hidden line problem.

**CHANGE OF COORDINATES**

The perspective transformation itself is clearest when the world-point is expressed in a coordinate frame attached to the camera. We therefore need a change of coordinates to go between some convenient "world coordinate" system and the camera coordinates. This coordinate transformation can be expressed as a linear transformation in homogeneous coordinates, but for our purposes a less elegant method will suffice. The transformation is straightforward, but a little bothersome, so we'll do it in several steps. We'll first transform from world coordinates to "robot coordinates," where the robot coordinate system is attached to the center of rotation of the head assembly, but is always oriented parallel to the world system.
In other words, it is a pure translation of the world system to the head of the robot. The second step will be to rotate the robot system into what we will call the "gimbal coordinate system," which will be oriented so that its y-axis will be parallel to the optical axis of the camera. We then translate the gimbal system so its center is at the center of the image plane of the camera, and thus obtain the camera coordinate system in which the perspective transformation is defined.

We will use the same letter with primes to denote the same physical point in different coordinate systems, according to the following convention:

- unprimed: world coordinates
- one prime: camera coordinates
- two primes: robot coordinates
- three primes: gimbal coordinates

**Step 1: Room Coordinates to Robot Coordinates**

In Fig. 2, suppose that the robot is at position \( d \) (measured in room coordinates attached to the floor) and let the height of the center of rotation of the head assembly be \( h \) above the floor. Then if the location of a point is \( v \) (unprimed...therefore world coordinates), its coordinates with respect to the robot system are given by the vector equation:

\[
v'' = v - (d + h)
\]  

**Step 2: Robot Coordinates to Gimbal Coordinates**

This is a pure rotation, so we can set the rotation matrix
\[ R = [\text{Tilt}] [\text{Pan}] \]

\[
R = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \phi & \sin \phi \\
0 & -\sin \phi & \cos \phi \\
\end{bmatrix} \begin{bmatrix}
\cos \theta & \sin \theta & 0 \\
-\sin \theta & \cos \theta & 0 \\
0 & 0 & 1 \\
\end{bmatrix}
\]

where angles are measured in the usual counterclockwise sense. That is, first we turn the robot through a pan angle \( \theta \), and then we tilt up through an angle \( \phi \). Multiplying out, we obtain

\[
R = \begin{bmatrix}
\cos \theta & \sin \theta & 0 \\
-\cos \phi \sin \theta & \cos \phi \cos \theta & \sin \phi \\
\sin \phi \sin \theta & -\sin \phi \cos \theta & \cos \phi \\
\end{bmatrix}
\]

and the transformation from robot (double primed) coordinates to gimbal (triple primed) coordinates is

\[ v''' = Rv'' \quad . \] (2)

**Step 3: Gimbal Coordinates to Camera Coordinates**

The camera coordinates do not coincide with the gimbal coordinates because of two reasons. First, the lens is not at the center of rotation of the head, and second, we want the camera coordinate system to be centered at the image plane and not at the lens. We therefore have two offsets: the vector \( b''' \) is the displacement of the lens from the center of the gimbal coordinates, and the vector \( f''' \) is the displacement from the lens to the center of the image plane. Hence, the camera coordinates of the point are obtained from the gimbal coordinates by
\begin{equation}
  v' = v'' - (\ell'' + \bar{f}'') \quad .
\end{equation}

From an inspection of the robot itself, it appears that the vectors \( \ell'' \) and \( \bar{f}'' \) are collinear with the \( y'' \) axis, so \( \ell'' = (0, \ell, 0) \) and \( \bar{f}'' = (0, \bar{f}, 0) \).

Combining Steps 1, 2, and 3, we have
\begin{equation}
  v' = Rv'' - (\ell'' + \bar{f}'')
\end{equation}

\text{pic. coord. from world coord.}
\begin{equation}
  v' = R[v - (d+h)] - (\ell'' + \bar{f}'') \quad .
\end{equation}

Inverting
\begin{equation}
  v = R^{-1}[v' + \ell'' + \bar{f}''] + d + h \quad .
\end{equation}

Note that \( R^{-1} = R^t \) since pure rotation operators are orthonormal. Equations 4 and 5, together with the direct and inverse perspective transformations, provide everything necessary to go from a world point to a picture point or from a picture point to a ray in the world.

**WORLD POINT TO PICTURE POINT: GENERAL CASE**

We are given a point in the world \( v = (x, y, z) \) expressed in world coordinates, and want to find the point \( v'_p = (x'_p, y'_p, z'_p) \) of the corresponding point in the image plane expressed in camera coordinates. (We can usually ignore the \( y' \) coordinate, as we saw earlier.) To do this, we first express the world point \( v \) in camera coordinates \( v' \) and then apply the perspective transformation. We assume the following displacement vectors:
\[ d = (d_1, d_2, 0) \]
\[ h = (0, 0, h) \quad \text{(through a small abuse of notation)} \]
\[ \ell = (0, \ell, 0) \]
\[ f = (0, f, 0) \]

Then
\[ v' = R[v - (d + h)] - (\ell'' + f'') \]

\[
\begin{bmatrix}
\cos \theta & \sin \theta & 0 \\
-\cos \varphi \sin \theta & \cos \varphi \cos \theta & \sin \varphi \\
\sin \varphi \sin \theta & -\sin \varphi \cos \theta & \cos \varphi
\end{bmatrix}
\begin{bmatrix}
x - d_1 \\
y - d_2 \\
z - h
\end{bmatrix}
- 
\begin{bmatrix}
0 \\
\ell + f \\
0
\end{bmatrix}
\]

\[
\begin{bmatrix}
(x-d_1)\cos \theta + (y-d_2)\sin \theta \\
-(x-d_1)\cos \varphi \sin \theta + (y-d_2)\cos \varphi \cos \theta + (z-h)\sin \varphi - (\ell+f) \\
(x-d_1)\sin \varphi \sin \theta - (y-d_2)\sin \varphi \cos \theta + (z-h) \cos \varphi
\end{bmatrix}
\]

Now, applying the perspective transformation,
\[
v' _{p} = \left( \frac{fx'}{fy'}, \frac{fy'}{fy'}, \frac{fz'}{fy'} \right)
\]

and ignoring \( y'_{p} \), we obtain

\[
x'_{p} = f\left(\frac{(x-d_1)\cos \theta + (y-d_2)\sin \theta}{-(x-d_1)\cos \varphi \sin \theta + (y-d_2)\cos \varphi \cos \theta + (z-h)\sin \varphi - \ell}\right) \tag{6}
\]

\[
z'_{p} = f\left(\frac{(x-d_1)\sin \varphi \sin \theta - (y-d_2)\sin \varphi \cos \theta + (z-h)\cos \varphi}{-(x-d_1)\cos \varphi \sin \theta + (y-d_2)\cos \varphi \cos \theta + (z-h)\sin \varphi - \ell}\right) \tag{7}
\]
Equations 6 and 7 give the picture coordinates of the image of an arbitrary point in space for an arbitrary position of the robot camera.

**IMAGE POINT TO RAY IN THE WORLD: GENERAL CASE**

We are given an image point \((x'_p, y'_p, z'_p)\) expressed in camera coordinates and want the corresponding ray in the real world. We saw earlier that the inverse perspective transformation gives this most naturally as a vector that is parametric in \(y'_p\), the fictitious third component of the picture point. This vector is simply transformed back to world coordinates to obtain the solution.

So suppose we have a point \((x'_p, y'_p, z'_p)\) of a point in the picture. A point \(v'\) on the ray is

\[
v' = \left( \frac{f x'_p}{f-y'_p}, \frac{f y'_p}{f-y'_p}, \frac{f z'_p}{f-y'_p} \right)
\]

Then in world coordinates,

\[
v = R^{-1} (v' + \lambda'''+ f''') + d + h
\]

\[
v = R^{-1} \left[ v' + \begin{pmatrix} 0 \\ \ell \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \ell \\ 0 \end{pmatrix} \right] + \begin{pmatrix} d_1 \\ d_2 \\ h \end{pmatrix}
\]

\[
v = R^{-1} \left( \frac{f x'_p}{f-y'_p}, \frac{f y'_p}{f-y'_p} + \ell + f, \frac{f z'_p}{f-y'_p} \right) + (d_1, d_2, h)
\]
Using

\[
R^{-1} = \begin{bmatrix}
\cos \theta & -\cos \varphi \sin \theta & \sin \varphi \sin \theta \\
\sin \theta & \cos \varphi \cos \theta & -\sin \varphi \cos \theta \\
0 & \sin \varphi & \cos \varphi
\end{bmatrix}
\]

we obtain

\[
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix} = \frac{1}{f-y_p} \begin{bmatrix}
x_p' \cos \theta - \left(f^2 + f\ell - k_y p' \right) \cos \varphi \sin \theta + f z_p' \sin \varphi \sin \theta \\
x_p' \sin \theta + \left(f^2 + f\ell - k_y p' \right) \cos \varphi \cos \theta - f z_p' \sin \varphi \cos \theta \\
(f^2 + f\ell - k_y p') \sin \varphi + f z_p' \cos \varphi
\end{bmatrix} + \begin{bmatrix}
d_1 \\
d_2 \\
h
\end{bmatrix}
\]

This equation gives the coordinates of the real-world vector, expressed in world coordinates, of a ray through the picture point \((x_p', z_p')\). It is parametric in \(y_p'\). Two particular values of \(y_p'\) are of special interest, \(y_p' = -\infty\) and \(y_p' = 0\). If we let \(y_p'\) approach minus infinity, we obtain the world coordinates of the lens center,

\[
v_{\ell} = \begin{bmatrix}
x_{\ell} \\
y_{\ell} \\
z_{\ell}
\end{bmatrix} = \begin{bmatrix}
d_1 - \ell \cos \varphi \sin \theta \\
d_2 + \ell \cos \varphi \cos \theta \\
h + \ell \sin \varphi
\end{bmatrix}
\]

For \(y_p' = 0\), we obtain the world coordinates of the picture point \((x_p', z_p')\),

\[
v_p = \begin{bmatrix}
x_p \\
y_p \\
z_p
\end{bmatrix} = \begin{bmatrix}
d_1 + x_p' \cos \theta - (f+\ell) \cos \varphi \sin \theta + z_p' \sin \varphi \sin \theta \\
d_2 + x_p' \sin \theta + (f+\ell) \cos \varphi \cos \theta - z_p' \sin \varphi \cos \theta \\
h + (f+\ell) \sin \varphi + z_p' \cos \varphi
\end{bmatrix}
\]

Then a point \(v\) on the ray from the lens center through the picture point can be written more transparently as

\[v = \text{v}_{\ell} + \frac{d}{f-y_p} \left( \begin{bmatrix}
x_p \\
y_p \\
z_p
\end{bmatrix} - \text{v}_{\ell} \right)\]
\[ v = v_\perp + \lambda \ v_r \]

where

\[
v_r = v_p - v_\perp = \begin{bmatrix}
x' \cos \theta + z' \sin \varphi \sin \theta - f \cos \varphi \sin \theta \\
x' \sin \theta - z' \sin \varphi \cos \theta + f \cos \varphi \cos \theta \\
z' \cos \varphi + f \sin \varphi
\end{bmatrix}
\]

and \( \lambda \), the parameter for the ray, is related to the mysterious \( y'_p \) by

\[
\lambda = \frac{f}{f - y'_p}
\]

or

\[
y'_p = f \frac{\lambda - 1}{\lambda}
\]

Thus, if we follow a ray from the lens center (\( \lambda = 0 \)) through the picture plane (\( \lambda = 1 \)) and on out into space, \( y'_p \) increases monotonically from \( -\infty \), being negative for points behind the picture plane, zero for points on the picture plane and positive for points in front of the picture plane. In the limit as \( \lambda \) approaches infinity, \( y'_p \) approaches \( f \). Summarizing, given an image point \((x'_p, z'_p)\) expressed in camera coordinates, the corresponding ray in world coordinates is given by

\[
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix} = \begin{bmatrix}
d_1 - \ell \cos \varphi \sin \theta \\
d_2 + \ell \cos \varphi \cos \theta \\
h + \ell \sin \varphi
\end{bmatrix} + \lambda \begin{bmatrix}
x' \cos \theta + z' \sin \varphi \sin \theta - f \cos \varphi \sin \theta \\
x' \sin \theta - z' \sin \varphi \cos \theta + f \cos \varphi \cos \theta \\
z' \cos \varphi + f \sin \varphi
\end{bmatrix}
\]
ILLUSTRATIVE APPLICATIONS

A. Locations of Objects: Floor Boundaries

A standard problem for a robot vision system is to provide the location of an object point in space given its image point. We saw earlier that a point in an image corresponds to a ray in space. In general we know only that the object point lies somewhere on this ray, so additional information is needed. This information can take several forms. We could, for example, specify the range of the object point from the camera lens. A more commonly used approach is to select an object point one of whose coordinates is already known. Typically, such an object point is one known to be on the floor, e.g., the lower end of a vertical edge. Let us assume that we have selected such an object point and that the world coordinates are taken with the X-Y plane coincident with the floor. Then in (8) we need only set the z-coordinate equal to zero, solve for the parameter \( \lambda \), and use this parameter value in the expressions for the X and Y components of the ray. If we do this, we obtain

\[
\lambda = - \frac{h + l \sin \varphi}{z_p' \cos \varphi + f \sin \varphi}.
\]

Substituting this in the other two equations in (8) yields

\[
x = d_1 - \frac{(h + l \sin \varphi) x' \cos \theta + ((l + h \sin \varphi) z'_p - hf \cos \varphi) \sin \theta}{z_p' \cos \varphi + f \sin \varphi}
\]

\[
y = d_2 - \frac{(h + l \sin \varphi) x' \sin \theta - ((l + h \sin \varphi) z'_p - hf \cos \varphi) \cos \theta}{z_p' \cos \varphi + f \sin \varphi}
\]
B. Vertical Lines: Perspective Distortion

A commonly noticed occurrence is that vertical lines often do not appear vertical in the picture. At times this so-called "perspective distortion" can be severe enough to make the identification of vertical lines difficult. We can, of course, never obtain a sufficient condition for a straight line in a picture to correspond to a vertical line in the real world, but we can derive a very easily implemented necessary condition.

Suppose in (6) and (7) we take the point \((x_0, y_0, z)\) to be the generic point on a vertical line through \((x_0, y_0)\). By eliminating \(z\) between these equations we obtain the equation of the image of the vertical line. The interesting result of this exercise is that the vertical axis intercept of the line \((Z'-\text{intercept in our notation})\) is \(f(\cot \varphi)\). In other words, the \(Z'-\text{intercept}\) depends only upon the scale factor \(f\) and the tilt angle \(\varphi\) of the camera, both of which are usually known. As shown in Fig. 3, the image of a vertical line anywhere in the real world passes through this point, the "vertical vanishing point" known to draftsmen and artists. A simple geometric interpretation can also be given, but we will stop short of this and remark only that the existence of the vertical vanishing point gives us an easily computable necessary condition that images of vertical lines must satisfy.

C. Horizontal Lines: Slope and Orthogonality Condition

Suppose we consider the image of a line known to lie in the plane of the floor. For simplicity let us orient our world coordinate system so that the camera pan angle \(\theta\) is zero, and suppose that the equation of the line in the floor is \(y = mx + b\) \((z = 0)\). If we use these values of \(x, y,\) and \(z\) in (6)
and (7) and eliminate \( x \) between the two equations, we obtain as the equation of the image of the straight line

\[
z'_p = \frac{-m(h + \ell \sin \varphi) x'_p + f(b \sin \varphi + h \cos \varphi)}{h \sin \varphi - b \cos \varphi + \ell}
\] (11)

There is nothing particularly simple about either the slope or intercepts of this line, unlike the case of vertical lines. However, instead of considering the intercept of the line with the picture coordinate axes let us examine the intercept of the line with the horizon. The horizon line is found by taking a ray from the lens parallel to the plane of the floor and finding where it pierces the image plane. As can be seen from Fig. 4, any such ray pierces the image plane at \( z'_h = -f \tan \varphi \). If we set (11) equal to \(-f \tan \varphi\) and solve for \( x'_p \), we find the horizon intercept \( x'_h \) to be

\[
x'_h = \frac{f}{m \cos \varphi}
\]

This result could also have been obtained by setting \( y = mx + b \) in (6) and letting \( x = \infty \), i.e., we let the object point recede to infinity and observe its image approach the horizon. In any event, we can find the slope of the line as

\[
m = \frac{f}{x'_h \cos \varphi}
\] (12)

This expression is simple enough in itself (and will be used in a slightly more general form in the next section), but we can also derive a simple necessary condition for two lines in the floor to be orthogonal. Let the slopes of the two lines be \( m_1 \) and \( m_2 \), and let the respective horizon
intercepts be $x_{h1}'$ and $x_{h2}'$. Then since $m_1 m_2 = -1$ we immediately have

$$x_{h1}' x_{h2}' = -\frac{f^2}{\cos^2 \varphi}. \quad (13)$$

This simple test can be used to resolve such ambiguities as cube/wedge-on-side or corner-of-room/door-ajar.

D. **Robot Reorientation**

So far we have assumed that the position and orientation of the robot, given by the parameters $d_1$, $d_2$ and $\theta$, are known. As the robot moves about, our knowledge of these parameters may degrade as small errors corrupt dead-reckoning calculations. However, if the locations of some distinctive points in the room are known, together with their corresponding picture coordinates, we can use this information to reorient the robot. In the sequel, we shall see how the location of one known floor point can be used to fix $d_1$ and $d_2$, and how the slope of a known line in the floor (probably a wall-floor boundary) can be used to fix $\vartheta$, though, of course, other procedures can also be used.

Let the known floor point $(x, y, 0)$ image at $(x_p', z_p')$. Then we have at once from (9) and (10)

$$d_1 = x + \frac{(h + \ell \sin \varphi) x_p' \cos \vartheta + (\ell + h \sin \varphi) z_p' - hf \cos \varphi) \sin \vartheta}{z_p' \cos \varphi + f \sin \varphi} \quad (14)$$

$$d_2 = y + \frac{(h + \ell \sin \varphi) x_p' \sin \vartheta - (\ell + h \sin \varphi) z_p' - hf \cos \varphi) \cos \vartheta}{z_p' \cos \varphi + f \sin \varphi} \quad (15)$$
These formulas are easy to evaluate, provided that $\theta$, the pan angle of the camera with respect to world coordinates, is known. Typically, our knowledge of $\theta$ is subject to at least as much error as $d_1$ and $d_2$, but we can obtain $\theta$ by an independent calculation. To compute $\theta$, we use a known line in the floor, described by $(x, mx + b, 0)$. Using the methods of the previous section, we obtain the horizon intercept

$$x_h' = \frac{f}{\cos \varphi} \cdot \frac{m \sin \theta + \cos \theta}{m \cos \theta - \sin \theta}$$

which shows that all lines having the same slope go through the same vanishing point $(x_h', z_h')$. This equation can be further simplified by the substitution

$$m = \tan \psi$$

which yields a generalization of (12),

$$x_h' = \frac{f}{\cos \varphi} \cdot \frac{1}{\tan (\psi - \theta)}$$

This in turn can be solved for $\theta$ to yield

$$\theta = \psi - \tan^{-1} \frac{f}{x_h' \cos \varphi} \tag{16}$$

Summarizing, from a floor line of known slope $m = \tan \psi$, we can measure the horizon intercept of its image $x_h'$ and compute $\theta$ from (16). This value of $\theta$ can then be used in (14) and (15) to find $d_1$ and $d_2$ from the image of a known floor point.
REFERENCES


FIG. 3