Useful Notes on Vectors, Matrices and Transformations

Marcelo Kallmann

This is a quick summary of selected basic and key concepts. For full details and explanations, please consult our textbook.

Matrices

We start with a reminder of some basic notations and operations. A matrix is a set of elements, organized into rows and columns:

$$m \text{ lines} \left\{ M = \overbrace{\begin{pmatrix} e_{11} & e_{12} & \cdots & e_{1n} \\ e_{21} & e_{22} & \cdots & e_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ e_{m1} & e_{m2} & \cdots & e_{mn} \end{pmatrix}}^{n \text{ columns}} \right.$$

Operations

Addition: $$A+B=\begin{pmatrix} a_{11}+b_{11} & a_{12}+b_{12} & \cdots & a_{1n}+b_{1n} \\ a_{21}+b_{21} & a_{22}+b_{22} & \cdots & a_{2n}+b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ e_{m1}+b_{m1} & a_{m2}+b_{m2} & \cdots & a_{mn}+b_{mn} \end{pmatrix}$$

Multiplication: $$AB=\begin{pmatrix} \sum_{i=0}^{n} a_{1i}b_{i1} & \sum_{i=0}^{n} a_{1i}b_{i2} & \cdots & \sum_{i=0}^{n} a_{1i}b_{ip} \\ \sum_{i=0}^{n} a_{2i}b_{i1} & \sum_{i=0}^{n} a_{2i}b_{i2} & \cdots & \sum_{i=0}^{n} a_{2i}b_{ip} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{n} a_{mi}b_{i1} & \sum_{i=0}^{n} a_{mi}b_{i2} & \cdots & \sum_{i=0}^{n} a_{mi}b_{ip} \end{pmatrix}, A_{m\times n}, B_{n\times p}.$$

Example of operations with 2x2 matrices:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}+\begin{pmatrix} e & f \\ g & h \end{pmatrix}=\begin{pmatrix} a+e & b+f \\ c+g & d+h \end{pmatrix},$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} - \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} a-e & b-f \\ c-g & d-h \end{pmatrix},$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{pmatrix}.$$

$\rightarrow$ Addition and Subtraction operate on matrices of same dimension.
$\rightarrow$ Multiplication is only defined when a $m\times n$ matrix multiplies a $n\times p$ matrix, yielding a $m\times p$ matrix as result. *Multiplication is not commutative.*
$\rightarrow$ Matrices preserve linear combinations: $A(a\mathbf{u}+b\mathbf{v})$=aA**u**+bA**v**.

Transpose

$$M = \begin{pmatrix} e_{11} & e_{12} & \cdots & e_{1n} \\ e_{21} & e_{22} & \cdots & e_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ e_{m1} & e_{m2} & \cdots & e_{mn} \end{pmatrix}_{m\times n}, \quad M^T = \begin{pmatrix} e_{11} & e_{21} & \cdots & e_{m1} \\ e_{12} & e_{22} & \cdots & e_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ e_{1n} & e_{2n} & \cdots & e_{mn} \end{pmatrix}_{n\times m}.$$

Ex: $A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$, $A^T = \begin{pmatrix} a & d & g \\ b & e & h \\ c & f & i \end{pmatrix}$; $\quad \mathbf{v} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$, $\mathbf{v}^T = \begin{pmatrix} x & y & z \end{pmatrix}$.

$A^{T^T} = A$
Transpose of a Product: $(AB)^T = B^T A^T$
Transpose of an inverse: $(A^{-1})^T = (A^T)^{-1}$

A is symmetric if $A^T = A$ (symmetric matrices produce scalings)

Identity

$$I : AI\text{=}A \Rightarrow I = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

Inverse

The inverse of a square matrix A is $A^{-1} : AA^{-1} = I$

Common way to compute inverse: solve the related linear system, for example for a 3x3 matrix the linear system will be:

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} a' & b' & c' \\ d' & e' & f' \\ g' & h' & i' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

A popular method is Gauss-Jordan elimination (with pivoting). Start with the extended matrix of the linear system:

$$\begin{pmatrix} a & b & c & 1 & 0 & 0 \\ d & e & f+0 & 1 & 0 \\ g & h & i & 0 & 0 & 1 \end{pmatrix};$$

and reduce the extended matrix until the first 3 columns become identity. The last 3 columns will then contain the inverse matrix.

The above process is usually faster when A is in the LU decomposition form *A*=*LU*, where:

$$L = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{m1} & l_{m2} & \cdots & 1 \end{pmatrix}, \; U = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{1mn} \end{pmatrix}.$$

Inverse of a product: $(AB)^{-1} = B^{-1}A^{-1}$

A simple proof: Let $C = AB \Rightarrow B = A^{-1}C, A = CB^{-1}$

$$\Rightarrow C = CB^{-1}A^{-1}C \Rightarrow B^{-1}A^{-1}C = I \Rightarrow B^{-1}A^{-1} = C^{-1}$$

Inverse of a 2D matrix: $\begin{pmatrix} a & c \\ b & d \end{pmatrix}^{-1} = \dfrac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$

Inverse of “small order” matrices can be computed analytically:

$$A^{-1} = \frac{1}{|A|}(C_{ij})^T = \frac{1}{|A|}\begin{pmatrix} C_{11} & \cdots & C_{1n} \\ \vdots & \ddots & \vdots \\ C_{m1} & \cdots & C_{mn} \end{pmatrix}^T,$$

where C is the cofactor matrix, $C_{ij} = (-1)^{i+j} M_{ij}$.

Determinant

Laplace's formula: $|A| = \sum_{i=1}^{k} a_{ij} C_{ij}$

2D: $\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc.$

3D: $\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = aei + bfg + cdh - afh - bdi - ceg$

$\rightarrow$ You can memorize the multiplications by drawing the diagonal lines
$\rightarrow$ A square matrix A has inverse $\Leftrightarrow$ det(A)$\neq$0
$\rightarrow$ If A has inverse $\Rightarrow$ A is nonsingular

A determinant can be found by multiplying the diagonal elements of a LU decomposition, and setting the sign according to row permutations (Gauss-Jordan elimination can also be used).

Vectors

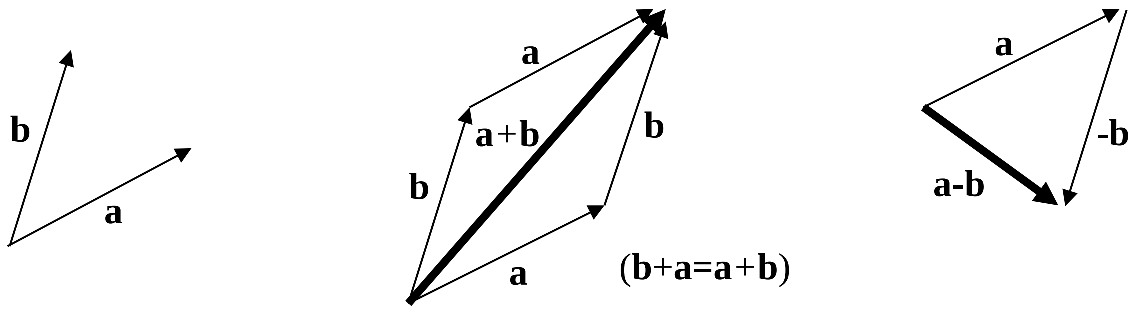
Vector notation: $\mathbf{v} = (v_1, v_2, \ldots, v_n) \in \Re^n$ (use lower case bold letters for vectors)

Column matrix notation: $\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$

Orthonormal Basis

Orthonormal basis: a set of vectors defining a space (such that a point in the basis will be a *linear combination* of the basis vectors)
Properties:
a vector is orthogonal to any other: dot product is zero,
each vector is normalized: magnitude is one.
Orthonormal means: orthogonal and normal.

Addition and Subtraction



Euclidean norm

The Euclidean norm of a vector **v** is

$$\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \ldots + v_n^2} = \sqrt{\sum_{i=0}^{n} v_i^2}$$

It gives the length (or magnitude) of the vector.

→ A unit (or normalized) vector has norm 1. A vector is normalized by dividing it by its norm.
The normal vector to a surface (or curve) is a vector perpendicular to the surface (or curve).

Triangle inequality: $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$

Dot Product

The dot product between **a** and **b** is $\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n \in \Re$

Also known as the inner product (notation: <**a**,**b**>), or the scalar product.

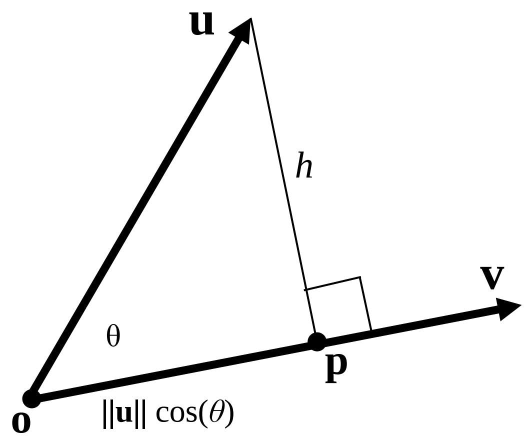
$$\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T \mathbf{b} = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = a_1 b_1 + a_2 b_2 + \ldots + a_n b_n \in \Re$$

Therefore, the square of the norm has a dot product form: $\|\mathbf{v}\|^2 = \mathbf{v}^T \mathbf{v}$

Properties

If **u**, **v** are perpendicular, from the triangle rectangle **u**, **v**, and **v-u** we have:
$\|\mathbf{v}-\mathbf{u}\|^2 = \|\mathbf{v}\|^2 + \|\mathbf{u}\|^2 \Rightarrow \cdots \Rightarrow \mathbf{u}\cdot\mathbf{v} = 0$

So, if **u** and **v** are perpendicular $\Rightarrow \mathbf{u}\cdot\mathbf{v} = 0$
Acute angle: $\cos(\theta) > 0 \Rightarrow \mathbf{u}\cdot\mathbf{v} > 0$
Obtuse angle: $\cos(\theta) < 0 \Rightarrow \mathbf{u}\cdot\mathbf{v} < 0$



Dot product and the angle between two vectors:

$$\cos(\theta) = \frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$$

$h = \|\mathbf{u}\|\sin(\theta)$, $\mathbf{p} = (\|\mathbf{u}\|\cos(\theta))\dfrac{\mathbf{v}}{\|\mathbf{v}\|}$

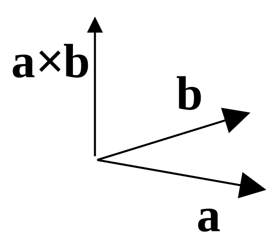
Observing that $0 \le \cos(\theta)^2 \le 1 \Rightarrow$
$(\mathbf{u}\cdot\mathbf{v})^2 \le \|\mathbf{u}\|^2\|\mathbf{v}\|^2$ Cauchy-Schwartz inequality

- **Important**: note that **p** is the projection of **u** on vector **v**.
- If **u** and **v** are unit vectors, the length of the projection is simply the dot product: $\|\mathbf{o}-\mathbf{p}\| = \mathbf{u}\cdot\mathbf{v}$ (Note that the norm of **o-p** gives the distance between **o** and **p**.)
- If only **v** is a unit vector, then the dot product **u · v** is |**u**| cos(θ), i.e. the magnitude of the projection of **u** in the direction of **v**, with a minus sign if the direction is opposite (this is the scalar projection of **a** onto **b**).

Cross product

The cross product between **a** and **b** is

$$\mathbf{v} = \mathbf{a}\times\mathbf{b} = \begin{vmatrix} \hat{\mathbf{i}} & \hat{\mathbf{j}} & \hat{\mathbf{k}} \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix} = \begin{pmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{pmatrix}$$



v is perpendicular to **a** and **b**, and has length $\|\mathbf{v}\| = \|\mathbf{a}\times\mathbf{b}\| = \|\mathbf{a}\|\|\mathbf{b}\|\sin(\theta)$, theta being the angle between **a** and **b**. The length is 2 times the area of the triangle with sides **a** and **b**.

The direction of **v** follows the right hand rule, assuming we are working on a right-hand coordinate system.

⇨ **Important**: the cross product operator is one of the most used tools in computer graphics!

Matrix notation $\mathbf{a}^{\times}$:

Cross product operator: $\mathbf{a}\times\mathbf{b} = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix}\begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix}$, $\mathbf{a}^{\times} = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix}$.

Operations with triangles

The normal of a triangle is therefore easily computed with a cross product. Given triangle with vertices **a**, **b**, **c**, its normal **n** is $(\mathbf{b}-\mathbf{a})\times(\mathbf{c}-\mathbf{a})$.
Normal vectors are often normalized (length 1). When a vector **u** is normalized (it has length 1), a common notation to indicate that is: $\hat{\mathbf{u}}$

If the following determinant is >0, the three 2D vertices **a**, **b**, **c**, are in CCW (counter-clockwise) order:

$$\begin{vmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \\ 1 & 1 & 1 \end{vmatrix} = a_x b_y + b_x c_y + c_x a_y - b_x a_y - c_x b_y - a_x c_y$$

Line Representations:

Parametric: $\mathbf{l}(t) = (1-t)\mathbf{p} + t\mathbf{q}$, where **p** and **q** are points in the line.

Implicit: $\mathbf{n} \cdot (\mathbf{x}-\mathbf{p}) = 0$ (**p** is a point in the line, **n** is normal)

Transformations

T is a linear transformation if: $T(a\mathbf{x}+\mathbf{y}) = aT(\mathbf{x})+T(\mathbf{y})$

Linear transformations (or maps) can be achieved with matrix multiplications, for ex:
$T(\mathbf{x}) = M\,\mathbf{x}$

2D rotation transformation: $R_\theta = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$

Let $\mathbf{b} = R_\theta \mathbf{a}$. It can be verified that $\|\mathbf{a}\|=\|\mathbf{b}\|$, and angle($\mathbf{a}$,$\mathbf{b}$)=θ, therefore we have a rigid transformation which rotates a vector by angle theta.

Affine Maps

Linear transformations with translations: $\mathbf{y} = A\mathbf{x}+\mathbf{p}$

Can be encoded with homogeneous coordinates

2D example: $\mathbf{v} = \begin{pmatrix} x \\ y \end{pmatrix}$, $\underline{\mathbf{v}} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$

Given $\underline{\mathbf{v}} = \begin{pmatrix} x \\ y \\ w \end{pmatrix}$, the Cartesian coordinates are $\mathbf{v} = \begin{pmatrix} x/w \\ y/w \end{pmatrix}$

Therefore a family of homogeneous vectors (all scalings) is equivalent to **v**.

Geometrical interpretation: projective geometry.

Basic 3D Transformations in Homogeneous Coordinates

$$T(a,b,c) = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$S(r,s,t) = \begin{pmatrix} r & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & t & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

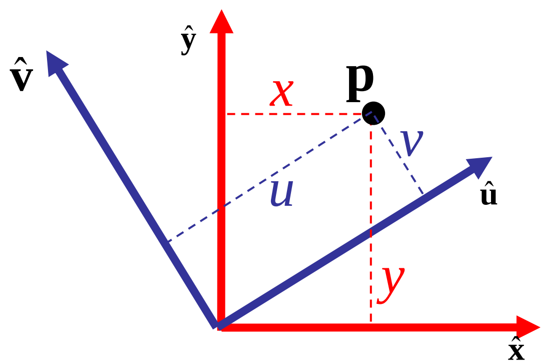
$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_y(\theta) = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_z(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Change of Frame of Reference or Change of Basis

Can be easily encoded with a transformation matrix, 2D example:



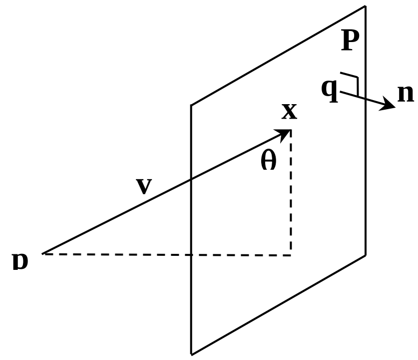
$$\begin{aligned}
&\mathbf{p}_{uv} = (u, v) = u\hat{\mathbf{u}} + v\hat{\mathbf{v}} \\
&\hat{\mathbf{u}} = a\hat{\mathbf{x}} + b\hat{\mathbf{y}} \\
&\hat{\mathbf{v}} = c\hat{\mathbf{x}} + d\hat{\mathbf{y}} \\
&\Rightarrow \mathbf{p} = u(a\hat{\mathbf{x}} + b\hat{\mathbf{y}}) + v(c\hat{\mathbf{x}} + d\hat{\mathbf{y}}) \\
&\Rightarrow \mathbf{p} = (ua + vc)\hat{\mathbf{x}} + (ub + vd)\hat{\mathbf{y}} \\
&\mathbf{p}_{xy} = (x, y) = x\hat{\mathbf{x}} + y\hat{\mathbf{y}} = (ua + vc, ub + vd) \\
&\Rightarrow \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & c \\ b & d \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} \Rightarrow \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}
\end{aligned}$$

Projections

Parallel projection of **p** by direction **v** into plane **P (q,n)**, **q** a point, **n** the normal.
If **θ=90 => orthographic projection, otherwise oblique projection**.

From the plane equation: $(\mathbf{x}-\mathbf{q})\cdot\mathbf{n}=0$ (1)

From the ray definition: $\exists t : \mathbf{x}=\mathbf{p}+t\mathbf{v}$ (2)



=> replacing (2) in (1):

$$((\mathbf{p}+t\mathbf{v})-\mathbf{q})\cdot\mathbf{n}=0 \Rightarrow t=\frac{(\mathbf{q}-\mathbf{p})\cdot\mathbf{n}}{\mathbf{v}\cdot\mathbf{n}}$$

$$\Rightarrow \mathbf{x}=\mathbf{p}+\frac{(\mathbf{q}-\mathbf{p})\cdot\mathbf{n}}{\mathbf{v}\cdot\mathbf{n}}\mathbf{v} \qquad (3)$$

Rewriting as an affine map:

$$\mathbf{x}=\mathbf{p}+\frac{\mathbf{q}\cdot\mathbf{n}}{\mathbf{v}\cdot\mathbf{n}}\mathbf{v}-\frac{\mathbf{p}\cdot\mathbf{n}}{\mathbf{v}\cdot\mathbf{n}}\mathbf{v}=\mathbf{p}-\frac{\mathbf{n}^{\mathrm{T}}\mathbf{p}}{\mathbf{v}\cdot\mathbf{n}}\mathbf{v}+\frac{\mathbf{q}\cdot\mathbf{n}}{\mathbf{v}\cdot\mathbf{n}}\mathbf{v}=\mathbf{p}-\frac{\mathbf{v}\mathbf{n}^{\mathrm{T}}}{\mathbf{v}\cdot\mathbf{n}}\mathbf{p}+\frac{\mathbf{q}\cdot\mathbf{n}}{\mathbf{v}\cdot\mathbf{n}}\mathbf{v}=\left(\mathbf{I}-\frac{\mathbf{v}\mathbf{n}^{\mathrm{T}}}{\mathbf{v}\cdot\mathbf{n}}\right)\mathbf{p}+\frac{\mathbf{q}\cdot\mathbf{n}}{\mathbf{v}\cdot\mathbf{n}}\mathbf{v}$$

We have now decomposed the equation in two terms: a linear transformation and a translation, and we can now write the corresponding transformation matrix:

$$\mathbf{x}=\begin{pmatrix} \begin{pmatrix} \mathbf{v}\cdot\mathbf{n} & 0 & 0 \\ 0 & \mathbf{v}\cdot\mathbf{n} & 0 \\ 0 & 0 & \mathbf{v}\cdot\mathbf{n} \end{pmatrix} - \left(\mathbf{v}\mathbf{n}^{\mathrm{T}}\right) & (\mathbf{q}\cdot\mathbf{n})\mathbf{v} \\ 0 \quad 0 \quad 0 & \mathbf{v}\cdot\mathbf{n} \end{pmatrix}\begin{pmatrix}\mathbf{p}\\ 1\end{pmatrix}$$

For a **perspective projection**, direction **v** now becomes **-p**, which is the line from **p** to the coordinate system origin.

From (3): $\mathbf{x}=\mathbf{p}+\dfrac{(\mathbf{q}-\mathbf{p})\cdot\mathbf{n}}{\mathbf{v}\cdot\mathbf{n}}\mathbf{p}=\mathbf{p}+\dfrac{\mathbf{q}\cdot\mathbf{n}}{\mathbf{p}\cdot\mathbf{n}}\mathbf{p}-\dfrac{\mathbf{p}\cdot\mathbf{n}}{\mathbf{p}\cdot\mathbf{n}}\mathbf{p}\Rightarrow\mathbf{x}=\dfrac{\mathbf{q}\cdot\mathbf{n}}{\mathbf{p}\cdot\mathbf{n}}\mathbf{p}$

And the resulting transformation is:

$$\mathbf{x}=\begin{pmatrix} \begin{pmatrix} \mathbf{q}\cdot\mathbf{n} & 0 & 0 \\ 0 & \mathbf{q}\cdot\mathbf{n} & 0 \\ 0 & 0 & \mathbf{q}\cdot\mathbf{n} \end{pmatrix} & 0 \\ 0 \quad 0 \quad 0 & \mathbf{p}\cdot\mathbf{n} \end{pmatrix}\begin{pmatrix}\mathbf{p}\\ 1\end{pmatrix}$$

Note that the previous matrix obtained still depends on **p**. Fortunately, we can further manipulate its elements to achieve the following equivalent result:

$$\mathbf{x} = \begin{pmatrix} \begin{pmatrix} \mathbf{q}\cdot\mathbf{n} & 0 & 0 \\ 0 & \mathbf{q}\cdot\mathbf{n} & 0 \\ 0 & 0 & \mathbf{q}\cdot\mathbf{n} \end{pmatrix} & 0 \\ \mathbf{n}^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{p} \\ 1 \end{pmatrix}$$

As you can see the last line of a projection matrix results in a division term when the matrix is applied to a point. The division term, only possible with a homogeneous matrix, is always needed to achieve a projection transformation.

Some Important Properties of a 3D Rotation Matrix R

- R preserves angles and lengths
- R has nine elements and as a rotation has 3 degrees of freedom, the elements of R must obey 6 constraints, which are: each column is a unit vector, and the columns are orthogonal to each other (the same is true for the rows)
- R is orthogonal, i.e., the inverse of R is equal to the transpose of R
- Det (R) = 1

Transforming Points versus Transforming Normal Vectors

Transformation matrices are usually applied to points but they can also be applied to normal vectors. However, a normal vector transformed by a non-rigid transformation may not lead to a correct transformed normal vector.

To show this let's consider the problem of transforming the entire plane orthogonal to a normal vector n. The plane equation is:
$n \cdot (x-p) = 0$, where x and p are points in the plane.

The transformed plane by M will be defined by:
$n' \cdot (Mx–Mp) = 0$, where we want to determine the new normal n'.

We can then write:
$n' \cdot (Mx–Mp) = n \cdot (x-p)$,
$n'^T (Mx–Mp) = n^T (x-p)$,
$n'^T M(x–p) = n^T (x-p)$,
$n'^T M = n^T$,
$n'^T = n^T M^{-1}$,
$n' = (n^T M^{-1})^T = (M^{-1})^T n$.

The normal of a point in an object transformed by M should thus be transformed by $(M^{-1})^T$.

To see what this is doing, consider the singular value decomposition:

$M = R_1 S R_2$,

where R_1 is an orthogonal matrix, thus encoding rotations or reflections, which are rigid transformations, and R_2 is the transpose of R_1.

We can then write:
$(M^{-1})^T = [(R_1 S R_2)^{-1}]^T = (R_2^{-1} S^{-1} R_1^{-1})^T = (R_1^{-1})^T (S^{-1})^T (R_2^{-1})^T = R_1 S^{-1} R_2$.

Therefore when we transform normal vectors by the inverse-transpose of M we leave the rigid transformations unchanged while inverting the scaling transformation.

<u>Example:</u> Let p be the midpoint of the segment defined by endpoints a=(1,0) and b=(0,1), and let M be a non-uniform scaling transformation of 4 along the x-axis and 1 along the y-axis. The normal vector of the segment at point p is $n=(1/\sqrt{2}, 1/\sqrt{2})$, however $Mn = (4/\sqrt{2}, 1/\sqrt{2})$, which is clearly not the normal vector of Mp. Draw a sketch in a piece of paper ($\sqrt{2} \cong 1.4142$). The normal vector is deformed in the wrong orientation and to achieve the correct result we should instead multiply n by the inverse scaling matrix to compensate for the deformation caused by M.

<u>Note:</u> applying $(M^{-1})^T$ to a unit vector may not lead to another unit vector and so a re-normalization is often needed.