Orthogonal projection of an n-dimensional space into an m-dimensional subspace

An orthogonal projection must fulfill these two elemental equations:

$$0 \leq m \leq n, \qquad m, n \in \mathbb{N}_0, \qquad \mu_i \in \mathbb{R}, \qquad \vec{v}_i, \vec{p}(\vec{x}), \vec{x} \in \mathbb{R}^n$$

$$\sum_{i=1}^{m} \mu_i \vec{v}_i = \vec{p}(\vec{x})$$

$$\sum_{i=1}^{m} |\langle \vec{p}(\vec{x}) - \vec{x}, \vec{v}_i \rangle| = 0$$

With $\vec{p}(\vec{x})$ as the projection vector, $\vec{x}$ as the input vector that should be projected, $\vec{v}_i$ as the input span vectors of the subspace of the $\mathbb{R}^n$.

The last sum can be broken down into m equations.

We can see, that the first equation can be easily replaced inside the second one, so we get:

$$\sum_{i=1}^{m} \left| \langle \sum_{j=1}^{m} \mu_j \vec{v}_j - \vec{x}, \vec{v}_i \rangle \right| = 0$$

For better understanding we now break down this term into several equations:

$$\begin{gathered}
\langle \mu_1 \vec{v}_1 + \cdots + \mu_m \vec{v}_m - \vec{x}, \vec{v}_1 \rangle = 0 \\
\langle \mu_1 \vec{v}_1 + \cdots + \mu_m \vec{v}_m - \vec{x}, \vec{v}_2 \rangle = 0 \\
\vdots \\
\langle \mu_1 \vec{v}_1 + \cdots + \mu_m \vec{v}_m - \vec{x}, \vec{v}_m \rangle = 0
\end{gathered}$$

This linear system can be transformed into the following one:

$$\Rightarrow \begin{gathered}
\langle \mu_1 \vec{v}_1 + \cdots + \mu_m \vec{v}_m, \vec{v}_1 \rangle = \langle \vec{x}, \vec{v}_1 \rangle \\
\langle \mu_1 \vec{v}_1 + \cdots + \mu_m \vec{v}_m, \vec{v}_2 \rangle = \langle \vec{x}, \vec{v}_2 \rangle \\
\vdots \\
\langle \mu_1 \vec{v}_1 + \cdots + \mu_m \vec{v}_m, \vec{v}_m \rangle = \langle \vec{x}, \vec{v}_m \rangle
\end{gathered}$$

$$\Rightarrow \begin{matrix} \langle \mu_1 \vec{v}_1, \vec{v}_i \rangle + \cdots + \langle \mu_m \vec{v}_m, \vec{v}_1 \rangle = \langle \vec{x}, \vec{v}_1 \rangle \\ \langle \mu_1 \vec{v}_1, \vec{v}_i \rangle + \cdots + \langle \mu_m \vec{v}_m, \vec{v}_2 \rangle = \langle \vec{x}, \vec{v}_2 \rangle \\ \vdots \\ \langle \mu_1 \vec{v}_1, \vec{v}_i \rangle + \cdots + \langle \mu_m \vec{v}_m, \vec{v}_m \rangle = \langle \vec{x}, \vec{v}_m \rangle \end{matrix}$$

$$\Rightarrow \begin{pmatrix} \langle \vec{v}_1, \vec{v}_1 \rangle & \langle \vec{v}_2, \vec{v}_1 \rangle & \cdots & \langle \vec{v}_m, \vec{v}_1 \rangle \\ \langle \vec{v}_1, \vec{v}_2 \rangle & \langle \vec{v}_2, \vec{v}_2 \rangle & & \langle \vec{v}_m, \vec{v}_2 \rangle \\ \vdots & & \ddots & \vdots \\ \langle \vec{v}_1, \vec{v}_m \rangle & \langle \vec{v}_2, \vec{v}_m \rangle & \cdots & \langle \vec{v}_m, \vec{v}_m \rangle \end{pmatrix} \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_m \end{pmatrix} = \begin{pmatrix} \langle \vec{x}, \vec{v}_1 \rangle \\ \langle \vec{x}, \vec{v}_2 \rangle \\ \vdots \\ \langle \vec{x}, \vec{v}_m \rangle \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} \langle \vec{v}_1, \vec{v}_1 \rangle & \langle \vec{v}_2, \vec{v}_1 \rangle & \cdots & \langle \vec{v}_m, \vec{v}_1 \rangle \\ \langle \vec{v}_1, \vec{v}_2 \rangle & \langle \vec{v}_2, \vec{v}_2 \rangle & & \langle \vec{v}_m, \vec{v}_2 \rangle \\ \vdots & & \ddots & \vdots \\ \langle \vec{v}_1, \vec{v}_m \rangle & \langle \vec{v}_2, \vec{v}_m \rangle & \cdots & \langle \vec{v}_m, \vec{v}_m \rangle \end{pmatrix}^{-1} \begin{pmatrix} \langle \vec{x}, \vec{v}_1 \rangle \\ \langle \vec{x}, \vec{v}_2 \rangle \\ \vdots \\ \langle \vec{x}, \vec{v}_m \rangle \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_m \end{pmatrix}$$

Now we have a Gram-matrix that is our first step to the n-dimensional projection matrix.

For shorter equations we now set:

$$G = \begin{pmatrix} \langle \vec{v}_1, \vec{v}_1 \rangle & \langle \vec{v}_2, \vec{v}_1 \rangle & \cdots & \langle \vec{v}_m, \vec{v}_1 \rangle \\ \langle \vec{v}_1, \vec{v}_2 \rangle & \langle \vec{v}_2, \vec{v}_2 \rangle & & \langle \vec{v}_m, \vec{v}_2 \rangle \\ \vdots & & \ddots & \vdots \\ \langle \vec{v}_1, \vec{v}_m \rangle & \langle \vec{v}_2, \vec{v}_m \rangle & \cdots & \langle \vec{v}_m, \vec{v}_m \rangle \end{pmatrix}$$

The very first equations can be transformed into:

$$\sum_{i=1}^{m} \mu_i \vec{v}_i = \vec{p}(\vec{x}) = \langle \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_m \end{pmatrix}, \begin{pmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vdots \\ \vec{v}_m \end{pmatrix} \rangle$$

And we can replace the coefficient vector with our previous equation:

$$\Rightarrow \vec{p}(\vec{x}) = \langle G^{-1} * \begin{pmatrix} \langle \vec{x}, \vec{v}_1 \rangle \\ \langle \vec{x}, \vec{v}_2 \rangle \\ \vdots \\ \langle \vec{x}, \vec{v}_m \rangle \end{pmatrix}, \begin{pmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vdots \\ \vec{v}_m \end{pmatrix} \rangle$$

With further transformation we now get:

$$\Rightarrow \vec{p}(\vec{x}) = \begin{pmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vdots \\ \vec{v}_m \end{pmatrix}^T * G^{-1} * \begin{pmatrix} \langle \vec{x}, \vec{v}_1 \rangle \\ \langle \vec{x}, \vec{v}_2 \rangle \\ \vdots \\ \langle \vec{x}, \vec{v}_m \rangle \end{pmatrix} = \begin{pmatrix} \vec{v}_1 & \vec{v}_2 & \cdots & \vec{v}_m \end{pmatrix} * G^{-1} * \begin{pmatrix} \langle \vec{x}, \vec{v}_1 \rangle \\ \langle \vec{x}, \vec{v}_2 \rangle \\ \vdots \\ \langle \vec{x}, \vec{v}_m \rangle \end{pmatrix}$$

$$\Longrightarrow \vec{p}(\vec{x}) = \begin{pmatrix}\vec{v}_1 & \vec{v}_2 & \cdots & \vec{v}_m\end{pmatrix} * G^{-1} * \begin{pmatrix}\vec{v}_1^T \\ \vec{v}_2^T \\ \vdots \\ \vec{v}_m^T\end{pmatrix} * \vec{x}$$

Now we can shorten the term with:

$$A = \begin{pmatrix}\vec{v}_1 & \vec{v}_2 & \cdots & \vec{v}_m\end{pmatrix}$$

$$\Longrightarrow \vec{p}(\vec{x}) = A * G^{-1} * A^T * \vec{x}$$

The Gram-Matrix can be represented with A:

$$G = A^T A$$

And we get our desired transformation matrix:

$$\boldsymbol{\vec{p}(\vec{x}) = A\left(A^T A\right)^{-1} A^T * \vec{x}}$$

$$\boldsymbol{P = A(A^T A)^{-1} A^T}$$

Projection into the $\vec{v}_i$ space (getting the μ_i coefficients)

The next interesting part is not to get the 3-dimensional vector of the projection, we want the projection relative to our given $\vec{v}_1, \ldots, \vec{v}_2$ base:

$$\vec{p}(\vec{x}) = A(A^T A)^{-1} A^T * \vec{x} = \sum_{i=1}^{m} \mu_i \vec{v}_i$$

$$\Longrightarrow A(A^T A)^{-1} A^T * \vec{x} = \mu_1 \vec{v}_1 + \mu_2 \vec{v}_2 + \cdots + \mu_m \vec{v}_m$$

$$\Longrightarrow A(A^T A)^{-1} A^T * \vec{x} = \begin{pmatrix}\vec{v}_1 & \vec{v}_2 & \cdots & \vec{v}_m\end{pmatrix} \begin{pmatrix}\mu_1 \\ \mu_2 \\ \vdots \\ \mu_m\end{pmatrix} = A * \vec{\mu}$$

$$\Longrightarrow A^{-1} A(A^T A)^{-1} A^T * \vec{x} = \vec{\mu}$$

$$\Longrightarrow \boldsymbol{\left(A^T A\right)^{-1} A^T * \vec{x} = \vec{\mu}}$$

So our projection matrix for the μ-coefficients is now:

$$\mathbf{M} = \boldsymbol{(A^T A)^{-1} A^T}$$

Computaional informatics: The “Z”-coordinate

In 3D-applications it’s common to use the last component as the “depth”, the distance from the projection point to the original projected one. This component is needed for rendering images, because the renderer must know how the objects should be arranged on screen (for example what object is nearer to the camera than another).

So we need to reconstruct our final matrix, we need to append another row-dimension. For M counts:

$$M \in \mathbb{R}^{n \times m}$$

With n as the space dimension, and m as the desired subspace dimension.

For the current matrix we adjust it a little bit:

$$M_M \in \mathbb{R}^{n \times m+1}$$

But we get a problem: the equation of the (Euclidean) distance is not linear:

$$d_2 = |\vec{p}(\vec{x}) - \vec{x}|_2 = \sqrt{\langle \vec{p}(\vec{x}) - \vec{x}, \vec{p}(\vec{x}) - \vec{x} \rangle}$$

So this term doesn't fit into a matrix.

But we can help out with a trick: We use the 1-norm.

$$d_1 = |\vec{p}(\vec{x}) - \vec{x}|_1 = \sum_{i=1}^{n} \vec{p}(\vec{x})_i - \vec{x}_i$$

This norm doesn't return the real distance between the projection point and $\vec{x}$, but it's a good approximation and the distance is linear, what means the right order of objects is guaranteed.

So we need to append a $\mathbb{R}^{1 \times n}$ matrix that performs this calculation and stores the result in the last component of the vector.
To make the formula as good as possible, we can calculate the distance with orthonormal vectors (and in addition we get the advantage of a shorter equation):

$$\vec{x} - \vec{p}(\vec{x}) = \sum_{i=1}^{j} d_i \vec{n}_i\,, \qquad d_i \in \mathbb{R}$$

$$\Rightarrow \vec{x} - \vec{p}(\vec{x}) = \langle \begin{pmatrix} \vec{n}_1 \\ \vec{n}_2 \\ \vdots \\ \vec{n}_j \end{pmatrix}, \vec{d} \rangle = \begin{pmatrix} \vec{n}_1 & \vec{n}_2 & \cdots & \vec{n}_j \end{pmatrix} * \vec{d}$$

We shorten the $\vec{n}$ vector:

$$N = \begin{pmatrix} \vec{n}_1 & \vec{n}_2 & \cdots & \vec{n}_j \end{pmatrix}$$

Further transformation results in:

$$\Rightarrow \vec{x} - A(A^T A)^{-1} A^T * \vec{x} = N * \vec{d}$$

$$\Rightarrow (E - A(A^T A)^{-1} A^T) * \vec{x} = N * \vec{d}$$

$$\Rightarrow N^{-1}(E - A(A^T A)^{-1} A^T) * \vec{x} = \vec{d}$$

To get the distance we need to sum up the components of $\vec{d}$:

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix} = E_s, \qquad E_s \in \mathbb{R}^{1xn}$$

$$\Rightarrow \boldsymbol{E_s N^{-1} \left(E - A\left(A^T A\right)^{-1} A^T\right) * \vec{x} = d}$$

To simplify the computation, you can choose as the orthonormal base the standard bases:

$$N = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

$$\Longrightarrow N = N^{-1} = E$$

$$\Longrightarrow E_s E = E_s$$

$$\Longrightarrow \left(\boldsymbol{E_s} - \boldsymbol{E_s A}\left(\boldsymbol{A^T A}\right)^{-1} \boldsymbol{A^T}\right) * \vec{\boldsymbol{x}} = \boldsymbol{d}$$

The disadvantage of this method is the accuracy. Calculated vectors to the projection subspace orthonormal span normal result in the shortest possible way (from the 1-norm point of view). Imagine a projection from a 3-dimensional space onto a 2-dimensional plane. If you don’t take the normal plane vector, every projected point is subdivided into standard normals. That results in much inaccurate values for the distance. But if you take the normal vector, you then get the exact distance (but this only happens, when the subspace $n - 1$ dimensions).