Three dimensional coordinates into two dimensional coordinates transformation

Edward Gerhold

June 28, 2015

Version 0.1.6 (very drafty paper)
The formulas are correct. The text itself is in the first stage. And i have to learn LaTeX for the first time.
June 25, 2015

Remark On a piece of paper you see three coordinate axis pointing into three directions in space. In reality these vectors are two dimensional. Because they point into three directions on the paper, and not into the real space.

[[missing: Picture of a 3-D coordinate system with ijk-vectors on the axis pointing into three directions. See [1] for introduction.]]

In this document we will design a basis for the coordinate transformation. A basis is multiplied with the value of the coordinate to move to the correct new point.
In the case of cosines and sines, we move left and right and up and down, to tell you directly, what happens, when we multiply the coordinates with the matrix.

1 Definitions

Definition Let φ_n be the set of axis angles, one for each axis. The angles start at the same place, at the number zero. You have to arrange the x, y, and z axes like on a piece of paper around the unit circle by giving them the appropriate angles. All three angles start at the default at zero.

$$\varphi_n := \{\varphi_x, \varphi_y, \varphi_z\}$$

Example The function rad converts degrees to radians, its useful for computer functions taking radians.

$$rad(\phi) := \frac{\pi}{180} \times \phi, \phi \in R$$

Here is an example of three angles. The three axis have an angle of 120 degrees beetween each.

$$\varphi_x = rad210, \varphi_y = rad330, \varphi_z = rad90$$

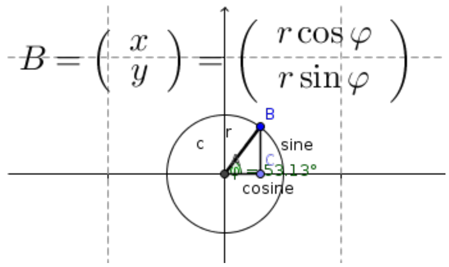
Definition Let e_n be the set of three two dimensional unit base vectors, namely $\vec{e}_x$,$\vec{e}_y$ and e_z. Other names are i, j or k for example, like on the picture of the coordinate system mentioned. The three vectors point into the three directions of the three axis. On a piece of paper they are two dimensional, because they point into three directions on the paper. The space being shown is what our brain completes, seeing three correct axis. The three base vectors represent exactly one unit of each axis.

$$e_n := \{\vec{e}_x, \vec{e}_y, \vec{e}_z\}$$

This is the set of three unit vectors in set notation. To guess no numbers, its easier for us, for each vector, to go around the unit circle by the angles we already defined and to use cosine and sine for the correct x-distance and y-distance. For help, you should remember this parametrization of x and y from the unit circle. [1]

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r\cos\varphi \\ r\sin\varphi \end{pmatrix}$$

$$(x, y) = (r\cos\varphi, r\sin\varphi)$$



Definition

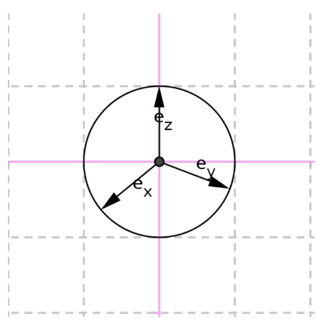
Modeling the three two dimensional base vectors with this information, we get the following three two dimensional base vectors.

$$\vec{e}_x := (r_x\cos(\varphi_x), r_x\sin(\varphi_x))^T = \begin{pmatrix} r_x\cos(\varphi_x) \\ r_x\sin(\varphi_x) \end{pmatrix}$$

$$\vec{e}_y := (r_y\cos(\varphi_y), r_y\sin(\varphi_y))^T = \begin{pmatrix} r_y\cos(\varphi_y) \\ r_y\sin(\varphi_y) \end{pmatrix}$$

$$\vec{e}_z := (r_z\cos(\varphi_z), r_z\sin(\varphi_z))^T = \begin{pmatrix} r_z\cos(\varphi_z) \\ r_z\sin(\varphi_z) \end{pmatrix}$$

One for each component of (x, y, z) By multiplying with, we move the points into the right pieces of direction. On the plane we use to point into three directions.



Remark.The values of r_x, r_y and r_z decide, how long one unit into each direction is. To preserve affine graphical transformations all three axes should have the same unit length, which can generally be enlarged or made smaller than unit length. By default the resulting vector of the cos and sin Terms has unit length, if you dont multiply with r_x, r_y and r_z.

Remark The length of r can be determined by pulling the root out of the sum of the squares of the vector. This is also known as euclidean norm, or the root of the inner product of the vector with itself. Like real

[1] Interested people can read about parametrization of x and y, the unit circle, polar coordinates and cosine and sine for example in the books of [1].

fans of sines and cosines, we know that $\sin^2 + \cos^2 = 1$ and what the root of 1 is. If we pull the root out of the products of cosine and sine multiplied with $r \neq 1$, we get the length of r again.

The other help we take The other lemma we need is the theorem for multiplying with the orthogonal bases. The sum of the basis multiplied with the coordinates is nothing new. But Literature and lecture scripts just tell how to multiply same dimensions, giving no clue about the easy 3-D to 2-D conversions.

$$\vec{x} = \sum_{i=1}^{n} \vec{x}_i \vec{e}_i$$

With $\vec{x}_i$ as the coordinate components and $\vec{e}_i$ as the corresponding base vector component and $\vec{v}$ as the resulting new vector.

This is also equal to, which also explains, what the ijk-Notation means, if it is not used for determining determinants, but to describe a vector.

$$\vec{x} = x\vec{i} + y\vec{j} + z\vec{k}$$

Dont forget, our i, j, k is two dimensional, because we draw on the screen or the paper.

A good book to read is [2] or the linear algebra lecture scripts. Where i also did not find this coordinate transformation.

Finishing the matrix

Each (x, y, z) coordinate has to be multiplied for the new (x', y') with its corresponding term of the unit vectors in the matrix. That means, to sum the products with (x, y, z) and the cos terms up for x' and to sum the products of (x, y, z) and the sin terms up for y'. This is the same as imagining walking left and right with $\cos\varphi$ and up and down with $\sin\varphi$. Or mathematically adding positive or negative values.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} xr_x \cos(\varphi_x) + yr_y \cos(\varphi_y) + zr_z \cos(\varphi_z) \\ xr_x \sin(\varphi_x) + yr_y \sin(\varphi_y) + zr_z \sin(\varphi_z) \end{pmatrix}$$

2 Theorem

Definition Let A be the matrix containing the three two dimensional unit vectors in order, one each column. You get a 2x3 matrix[2] $A : R^3 \to R^2$. With the base vectors $\begin{pmatrix} \cos\varphi_n \\ \sin\varphi_n \end{pmatrix}$ in the three columns.

$$A := \begin{pmatrix} \vec{e}_x & \vec{e}_y & \vec{e}_z \end{pmatrix} = \begin{pmatrix} r_x \cos(\varphi_x) & r_y \cos(\varphi_y) & r_z \cos(\varphi_z) \\ r_x \sin(\varphi_x) & r_y \sin(\varphi_y) & r_z \sin(\varphi_z) \end{pmatrix}$$

Theorem *(Fundamental Theorem of converting 3-D Points into 2-D Points)*:

If you multiply A, the matrix of the three two-dimensional unit vectors, with the three-coordinate points (x, y, z), the result is a two coordinate point, (x', y'). This point (x', y') is the correct point on the two dimensional plane, representing the point from the three dimensional coordinate system, you would like to display.

$$A \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix}$$

[2] A 2x3 Matrix, which i call the Gerhold Projection Matrix to distinguish it from other matrices, making sure it contains the three two dimensional and trigonometric base vectors.

Applying the operator[3] performs the following operation

$$A\begin{pmatrix} x \\ y \\ z \end{pmatrix} = x\vec{e}_x + y\vec{e}_y + z\vec{e}_z$$

$$= \begin{pmatrix} xr_x\cos(\varphi_x) + yr_y\cos(\varphi_y) + zr_z\cos(\varphi_z) \\ xr_x\sin(\varphi_x) + yr_y\sin(\varphi_y) + zr_z\sin(\varphi_z) \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix}$$

Proof

$$A\begin{pmatrix} x \\ y \\ z \end{pmatrix} = x\vec{e}_x + y\vec{e}_y + z\vec{e}_z = \begin{pmatrix} x' \\ y' \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} xr_x\cos(\varphi_x) + yr_y\cos(\varphi_y) + zr_z\cos(\varphi_z) \\ xr_x\sin(\varphi_x) + yr_y\sin(\varphi_y) + zr_z\sin(\varphi_z) \end{pmatrix}$$

Example
The following is an EcmaScript 6 (JavaScript) Arrow Function returning the new point. It is not optimized for speed. Means, to repeat the sin and cosine calls.
Example for Programmers let projection = ([x,y,z]) =¿ [x*r*Math.cos(xAxisAngle) + y*r*Math.cos(yAxisAngle) + z*r*Ma

3 Corollary

The theorem can be extended into more dimension to go down to any other dimension. The generic case is known to linear algebra, i found it lately after beginning this document and will include the information within the next versions.

Corollary *(Converting any Dimensions down to less dimensions)*:

The theorem can be extended to more dimensions, for example can four two-dimensional vectors represent a 4-D space on the 2-D plane. They get converted into the correct 2-D points. For Example, if you use a 2x4 matrix and convert all points at each instance of t you have a moving object into the direction of the fourth base vector.

$$A := \begin{pmatrix} \vec{e}_x & \vec{e}_y & \vec{e}_z & \vec{e}_t \end{pmatrix} = \begin{pmatrix} r_x\cos(\varphi_x) & r_y\cos(\varphi_y) & r_z\cos(\varphi_z) & r_t\cos(\varphi_t) \\ r_x\sin(\varphi_x) & r_y\sin(\varphi_y) & r_z\sin(\varphi_z) & r_t\sin(\varphi_t) \end{pmatrix}$$

Here the basis is four times of two dimensions. A 2x4 matrix.

$$A\begin{pmatrix} x \\ y \\ z \\ t \end{pmatrix} = \sum_n \vec{e}_n\vec{x}_n = \begin{pmatrix} x' \\ y' \end{pmatrix}$$

Proof:

[3]The *Gerholdian operator* is my favorite nickname for this matrix

$$A \begin{pmatrix} x \\ y \\ z \\ t \end{pmatrix} = \begin{pmatrix} xr_x \cos(\varphi_x) + yr_y \cos(\varphi_y) + zr_z \cos(\varphi_z) + zr_t \cos(\varphi_t) \\ xr_x \sin(\varphi_x) + yr_y \sin(\varphi_y) + zr_z \sin(\varphi_z) + zr_t \sin(\varphi_t) \end{pmatrix}$$

$$= x\vec{e}_x + y\vec{e}_y + z\vec{e}_z + t\vec{e}_t = \sum_n \vec{e}_n \vec{x}_n = \begin{pmatrix} x' \\ y' \end{pmatrix}$$

The same method can be used to convert any other number of dimensions to the xy-plane. But it can also be used in a generic m by n case[4], to convert from n dimension down to m, if you know the basis for the destination.

4 Quick Summary

Quick overview

1. Lay out the three base vectors around a circle and write down the angles φ_n. Programmers have to write down a variable for anyways.
2. Write down the base vectors $\vec{e}_n$ as $\cos\varphi$ and $\sin\varphi$ (two dimensional) with a unit length of 1, or multiplied with $r_n \neq 1$.
3. Put the three base vectors $\vec{e}_n$ into a matrix A. Programmers can directly code the two lines multiplication and forget the formal rest.
4. Iterate over your points and multiply each (x, y, z) with the matrix A, which acts as our linear operator, and put (x', y') into your new set.

References:

References

[Corr09] *Michael Corral, Schoolcraft College*, Vector Calculus, GNU Free Documentation License, http://mecmath.net (about ijk-Vector Notation)

[0] [1] *Michael Corral, Schoolcraft College*, Trigonometry, GNU Free Documentation License, http://mecmath.net (about sines and cosines)

[2] *Gilbert Strang*, Linear Algebra and its Applications. Introductory Book. A whole part covers the orthogonality.

[4] http://de.wikipedia.org/wiki/Abbildungsmatrix, also found in lecture scripts, but not anyone explaining me this matrix or the topic. Is it too obvious? Or isnt it obvious?