Three dimensional coordinates into two dimensional coordinates transformation

Edward Gerhold

June 29, 2015

Version 0.1.9 (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.
Has to be overworked a few more times.
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 axes 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 axes have an angle of 120 degrees beetween each.

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

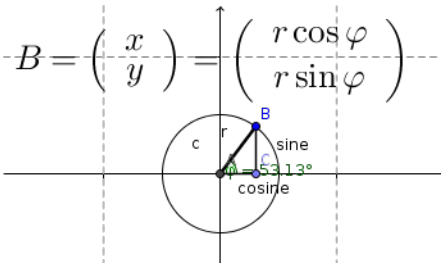
Definition Let e_n be the set of three two dimensional basis vectors, namely $\vec{e}_x$,$\vec{e}_y$ and e_z. Other names are i, j or k for example, like on the picture of the coordinate system mentioned. The three vectors point into the three directions of the three axes. On a piece of paper they are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

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

This is the set of three basis 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

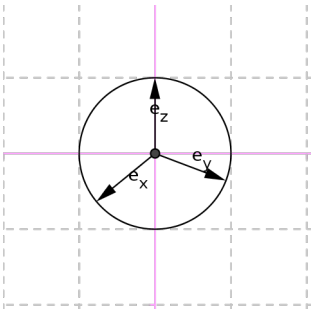
Modeling the three two dimensional basis vectors with this information, we get the following three two dimensional basis vectors.

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

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

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

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 length, to represent same distance for each coordinate unit, which can generally be enlarged or made smaller than *unit length*, which is a length of 1.

[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 [1], [2], [5].

By default the resulting vector of the cosine and sine Components has *unit length*, if you dont multiply the cosine and the sine with r_x, r_y and r_z in each basis vector.

Remark On the other side, the length of r can be determined from existing basis vectors, by pulling the square root out of the sum of the squares of the vector components. This is also known as euclidean norm, or the root of the inner product of the vector with itself. Like real fans of sines and cosines, we know that $\sin^2\varphi + \cos^2\varphi = 1$ 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. $r = \sqrt{\vec{x}\vec{x}} = \left(\Sigma_{i=1}^{n}\vec{x}_i^2\right)^{\frac{1}{2}}$.

Now about the multiplication of the coordinates with the basis.

The other help we take The other lemma we need is the theorem for multiplying with the orthogonal basis.

In our case the 90-orthogonality rules dont count, we arrange for example with about 120 beetween each around the origin on a a plane. The point is, the generic formula still holds.

Remark 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. I can not speak for all, and do not believe, that no one told no one about this, but i have not met any one in the last twenty years telling me about this easy transformation and computer graphics went another, more complicated way, over homogeneous coordinates, 4 by 4 matrices and a final viewport division, to get the points back to two dimensions.

The formula for multiplying a vector with a base to get a new vector is this.

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

It is done componentwise for each row of the vector.
With $\vec{x}_i$ as the coordinate components and $\vec{e}_i$ as the corresponding basis vector component and $\vec{v}$ as the resulting new vector.

This is also equal to

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

what also explains, what the ijk-Notation means. If you dont use it already for determining determinants for calculating cross products. Its for describing a vector. Dont forget, our i, j, k basis is two dimensional, because we draw on a 2-D plane like the computer screen or a piece of paper.

With a 3x3 basis the vector $x\vec{i}+y\vec{j}+z\vec{k}$ is equal to $\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}$. But with a 2x3 basis the vector $x\vec{i}+y\vec{j}+z\vec{k}$ is becoming $\begin{pmatrix} x' \\ y' \end{pmatrix}$

Finishing the matrix

Each (x, y, z) coordinate has to be multiplied for the new (x', y') with its corresponding term of the basis 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 1 *Let A be the matrix containing the three two dimensional unit vectors in order, one each column. You get a 2x3 matrix*[2] $A \in R^{2x3} : R^3 \to R^2$. *With the basis vectors* $\begin{pmatrix} r_n \cos\varphi_n \\ r_n \sin\varphi_n \end{pmatrix}$ *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 (The Fundamental Theorem of transforming 3-D Points into 2-D Points) 1 *:*

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 transform.*

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

Applying the operator[3] *performs the following operation*

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

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

(1)

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 example code for varius computer systems.
$x\_ = x * r * cos(alpha) + y * r * cos(beta) + z * r * cos(gamma)$
$y\_ = x * r * sin(alpha) + y * r * sin(beta) + z * r * sin(gamma)$

Example
The following is an EcmaScript 6 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]) =⟩[$x*r*Math.cos(xAxisAngle)+y*r*Math.cos(yAxisAngle)+z * r * Math.cos(zAxisAngle), x * r * Math.cos(xAxisAngle) + y * r * Math.sin(yAxisAngle) + z * r * Math.cos(zAxisAngle)$];

[2] A 2x3 Matrix, which i proudly gave the nickname Gerhold Projection Matrix to distinguish it from other matrices, making sure it contains the three two dimensional and trigonometric basis vectors, which allow arranging the projection vectors just by going around the circle instead of guessing wild numbers.
[3] The *Gerholdian projection operator* is my favorite nickname for this matrix

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 basis 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:

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

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

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 Summary

Summary of all neccessary steps

1. Lay out the three basis vectors around a circle and write down the angles φ_n. Programmers have to write down a variable for anyways.
2. Write down the basis vectors $\vec{e}_n$ as $r_n\cos\varphi_n$ and $r_n\sin\varphi_n$ (two dimensional). Dont multiply with r_n for a unit length of 1 or multiply with r_n to change the length of the basis vector.
3. Put the three basis vectors $\vec{e}_n$ into a matrix A. Programmers can directly code the two lines of 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.

[4] http://de.wikipedia.org/wiki/Abbildungsmatrix, also found in lecture scripts, but not anyone explaining me this matrix here or the topic of the from-three-to-two-points conversion. Is it too obvious? Or isnt it obvious?

Remark
About the word *unit*. I am not really sure, if i have to use *base vector* for a vector of any length and *unit vector* only for the *unit length* of 1. Because of the misleading mismatch with the *unit* of the thought *coordinate axes*, which the *base vector* defines, i tend in the first versions to misuse the word *unit vector* for both. If you find this, or any other formal mistake, be sure, it is not wanted :-) I will try to remove more of these spelling errors in the next versions.

References

Michael Corral, Schoolcraft College, Vector Calculus, GNU Free Documentation License, http://mecmath.net

Michael Corral, Schoolcraft College, Trigonometry, GNU Free Documentation License, http://mecmath.net

Michael Corral, Schoolcraft College, Latex Mini Tutorial, http://mecmath.net

Gilbert Strang, MIT, Linear Algebra and its Applications. Fourth Edition.

Gilbert Strang, MIT, Calculus. MIT OpenCourseWare Supplemental Resources. http://ocw.mit.edu

Manuela Jürgens, Thomas Feuerstack LaTeX, eine Einführung und ein bisschen mehr..., Fernuniversität Hagen, a026_latex_einf.pdf