Rotating vectors in 3D

Jarek Rossignac

1 Problem

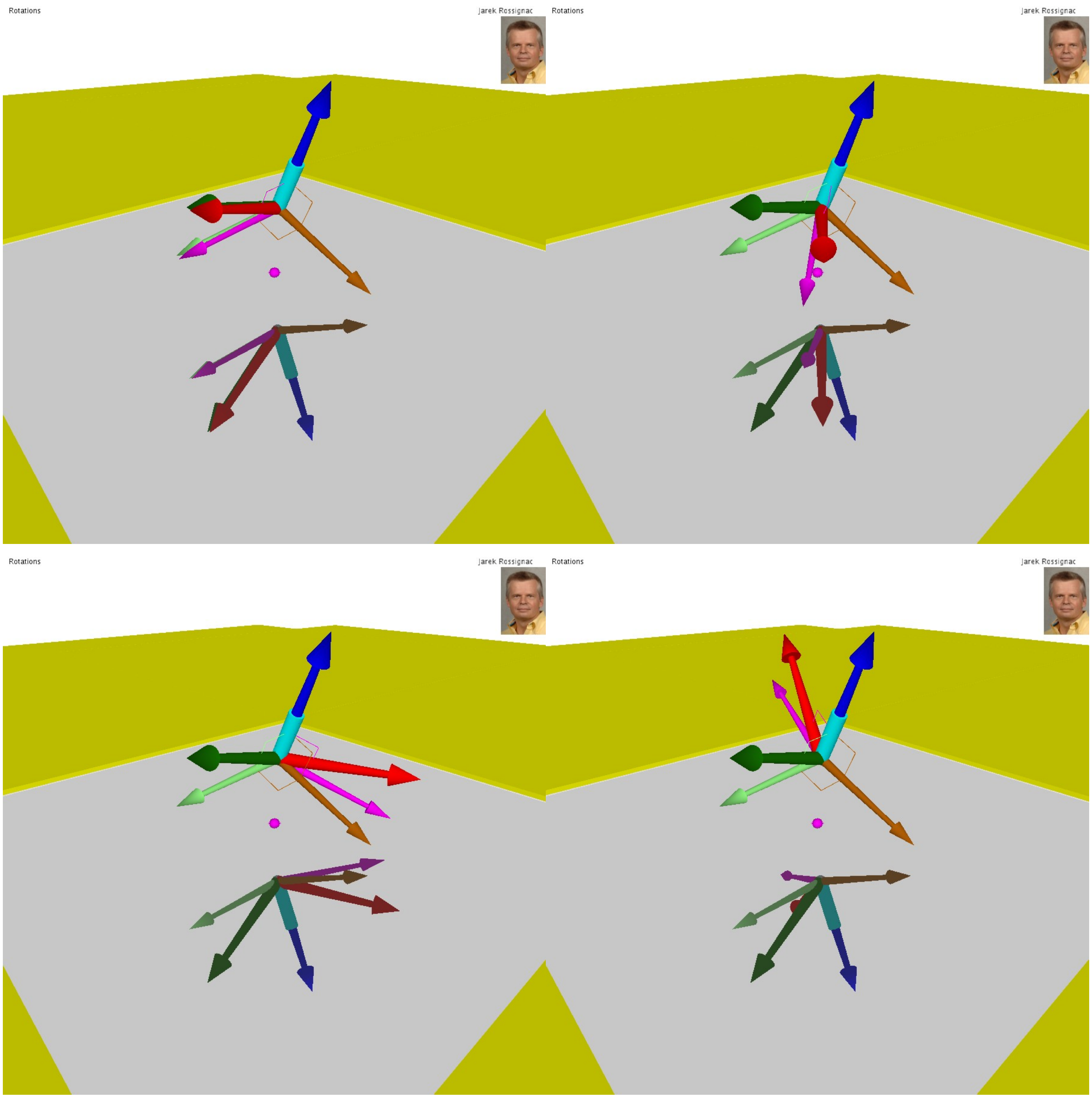
We are given two vectors, U and V, and an angle α. (V is a unit vector.)

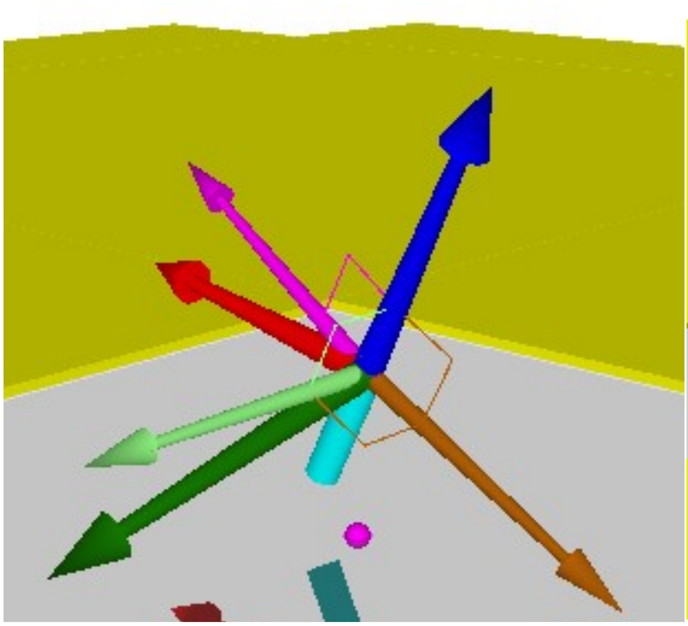
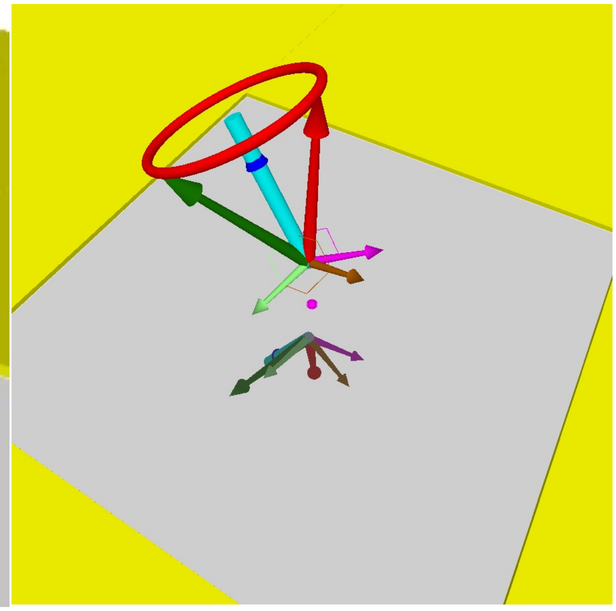
We want to compute the vector W that is obtained by rotating vector U clockwise by angle α around V.

Hence, we expect that |W|==|U| and that V^U==V^W.

2 One solution

There are many solutions. I propose one that does not require matrices or quaternions. Just dot and cross products.



2.1 U and $\underline{V}$

U is shown by the dark green thick arrow and $\underline{V}$ the blue thick arrow.

2.2 T

$T=(\underline{V}\bullet U)\underline{V}$, shown as a cyan cylinder, is the tangential component of U with respect to $\underline{V}$. T and $\underline{V}$ are always parallel. In the image above, $\underline{V}\bullet U<0$, hence, in this configuration, T and $\underline{V}$ have opposite orientations.

2.3 N

N=U-T, shown as a light green arrow, is the normal (orthogonal) component of U with respect to $\underline{V}$. N is defined by U=T+N.

2.4 M

$M=N\times\underline{V}$, shown as a brown arrow, is orthogonal to N, $\underline{V}$, and U (as indicated by the brown right-angle thin lines)). Because $|\underline{V}|=1$, $|M|=|N|$.

2.5 R

$R = \cos\alpha\ N + \sin\alpha\ M$, shown as a magenta arrow, is the rotation of N by α around $\underline{V}$, since both N and M are orthogonal to $\underline{V}$.

2.6 W

$W = R + T$, shown as a red thick arrow, is the rotation of N by α around $\underline{V}$, since both N and M are orthogonal to $\underline{V}$.

2.7 Summary

$T=(\underline{V}\bullet U)V$

$N=U-T$

$M=N\times\underline{V}$

$R = \cos\alpha\ N + \sin\alpha\ M$

$W = R + T$

2.8 A variant

We can also compute M as $\underline{V}\times U$ and write

$W = \cos\alpha\ (U-T) + \sin\alpha\ M + T$

Replacing M and T as defined above:

$W = \cos\alpha\ U + \sin\alpha\ \underline{V}\times U + (1-\cos\alpha)\ (\underline{V}\bullet U)\underline{V}$, which is known as the Rodriguez formula.

Using $\cos\alpha=\mathrm{cois}^2(\alpha/2) - \sin^2(\alpha/2)$, $1-\cos\alpha= 2\sin^2(\alpha/2)$ and $\sin\alpha= 2\cos(\alpha/2)\sin(\alpha/2)$:

$W = (1-2\sin^2(\alpha/2))U + 2\cos(\alpha/2)\sin(\alpha/2)\underline{V}\times U + 2\sin^2(\alpha/2)(\underline{V}\bullet U)\underline{V}$

Setting $c=\cos(\alpha/2)$, $s=\sin(\alpha/2)$ and $X=s\underline{V}$:

$W = 2cX\times U + 2(X\bullet U)X + (1-2s^2)U$

Rearranging

$W = 2cX\times U + 2(X\bullet U)X - 2s^2U + U,$

Using $U\times(V\times W) = (U\bullet W)V - (U\bullet W)W$:

$W = 2cX\times U + 2(X\bullet U)X - 2(X\bullet X)U + U,$

Using $U\times(V\times W) = (U\bullet W)V - (U\bullet V)W$, https://en.wikipedia.org/wiki/Cross_product

$W = 2cX\times U + 2X\times(X\times U) + U$.

2.9 Implementation

```
vec Cross(vec U, vec V) {return V( U.y*V.z-U.z*V.y, U.z*V.x-U.x*V.z, U.x*V.y-U.y*V.x); };
vec RotateAround(vec U, vec D, float a)
 {
 float c=cos(a/2);
 vec X = V(sin(a/2),U(D));
 vec N = Cross(X,U);
 return V(1.,U,2,Cross(X,N),2.*c,N);
 }
```

2.10 Comparing to other approaches

A classic approach is to find the minimum angle rotation R that brings V to the K=<0,0,1> axis and to compute a rotation matrix that combines R with the rotation by α around K and with the inverse of R (to put V back where it was). Hence, it performs a temporary change of coordinate system to one in which they know how to perform the rotation by α using a matrix.