Subject: rot and poly_2d question
Posted by Helder Marchetto on Mon, 10 Aug 2015 16:52:59 GMT
View Forum Message <> Reply to Message

Hi,
I've been using rot() since ages and I would like to understand the use of this function a bit more.
Rot uses poly_2d to rotate images. To do that, the 2 times 4 coefficients of the equations are calculated:

x' = P0 + P1*x + P2*y + P3*x*y
y' = Q0 + Q1*x + Q2*y + Q3*x*y

In the case of rot(), the coefficients are quite complicated. I have expressed them using LaTex here: http://idl.marchetto.de/rot-transformation-coeffiecients/

I now come to the question... how were these coefficients calculated in the first place? I would have expected them to be something like this:

x' = translationX + M*cos(theta)*x - M*sin(theta)*y + 0*x*y
y' = translationY + M*sin(theta)*x + M*cos(theta)*y + 0*x*y

[I omitted here the magnification for simplicity]

Can someone who understands the math explain me why the two are different?

Seeing it from another point of view, given an affine transformation matrix as a result of translation+scaling+rot matrix (T*S*R):
| M*cos(theta)   -M*sin(theta)   0 |
| M*sin(theta)    M*cos(theta)   0 |
|      tr_x            tr_y      1 |

where tr_x and tr_y are:

tr_x =  tx M cos(theta) + ty M sin(theta)
tr_y = -tx M sin(theta) + ty M cos(theta)

(the matrix is also shown in the link)

How can I relate the above matrix to the equations for x' and y'?

Since an explanation is tooooo long, I would really be happy if you could point me to some good resource (internet or book) to grasp these things a bit better.

Thanks,
Helder

PS: I know of polywarp, but that is rather to determine coefficients when these cannot be calculated (here they should be pretty simple/straight forward).