Subject: Polywarp order
Posted by wlandsman on Thu, 09 Dec 2010 14:39:35 GMT
View Forum Message <> Reply to Message

I have used POLYWARP/POLY_2D successfully for many years.   It seems to be robust and simple method to align images using grid points (or in astronomy to align images with different world coordinate system information).

But I have always been bothered by the definition of order in POLYWARP

Xi = sum over i and j of: Kx[i,j] * Xo^j * Yo^i
Yi = sum over i and j of: Ky[i,j] * Xo^j * Yo^i

so for second order Xi = Kx[0,0] + Kx[0,1]*Y0 + Kx[0,2]*Y0^2 + Kx[1,0]*X0 + Kx[1,1]*X0*Y0 + Kx[1,2]*X0*Y0^2 + Kx]2,0]*X0^2 +
Kx[2,1]*X0^2*Y0 + Kx[2,2]*X0^2*Y0^2

To me this is not a second order equation because it has cubic terms (Kx[1,2]*X0*Y0^2) and a quartic term (Kx[2,2]*X0^2*Y0^2).   So while POLYWARP works, I wonder if it requires more control points than necessary.   I have successfully fit a "true" second order, such as

  Xi = Kx[0,0] + Kx[0,1]*Y0 + Kx[0,2]*Y0^2 + Kx[1,0]*X0 + Kx[1,1]*X0*Y0 + Kx]2,0]*X0^2   .

using mpfit2dfun to determine the Kx[i,j] and then using POLY_2d with the "extra" terms such as Kx[2,2] set to zero to transform the image.   But I am not sure in what way, if any, this is a "better" solution.   (My google searching suggests that both definitions are used, but that a "true" second order is more common.)   --Wayne
