Subject: Re: Image warping in IDL
Posted by Wox on Wed, 08 Nov 2006 14:50:58 GMT
View Forum Message <> Reply to Message

On Wed, 8 Nov 2006 07:01:29 -0700, David Fanning <news@dfanning.com>
wrote:

> ...but I'll have
> to study the question for a few more days to understand
> just what you want here. :-)

The problem is not straitforward. Let me try again:

1. I have two arrays with the same dimensions:
- input image with pixels [Xi,Yi]: this contains my image
- output image with pixels [Xo,Yo]: this "will" contain the warped
image

2. There are two ways of warping:
a. Inverse warping: You have two surfaces (Xo,Yo)->Xi and (Xo,Yo)->Yi
where (Xo,Yo) are irregular and non-integer. First these two surfaces
are evaluated for the pixels of the output image, i.e. regular-integer
(Xo,Yo). Now we know where each output pixel is located in the input
image (hence the name "inverse" warping). These locations are
non-integer, so we have to INTERPOLATE (bilinear, cubic, whatever...).

=> gridding + interpolation

b. Forward warping: You have two surfaces (Xi,Yi)->Xo and (Xi,Yi)->Yo
where (Xi,Yi) are irregular and non-integer. First these two surfaces
are evaluated for the pixels of the input image, i.e. regular-integer
(Xi,Yi). Now we know where each input pixel is located in the output
image (hence the name "forward" warping). These locations are
non-integer, so we have to RESAMPLE.

=> gridding + resampling

Forward warping is slower because the resampling, as I implemented it,
loops over all colums and row (and the for each pixel in the
row/column). I use forward warping, because I only have the surfaces
(Xi,Yi)->Xo and (Xi,Yi)->Yo (actually I have the coefficients of two
2D splines).

Now the question again:
1. Can I make the resampling faster in IDL (avoid the looping)?

2. Can I avoid the resampling completely by somehow converting the 2 spline surfaces to (Xo,Yo)->Xi and (Xo,Yo)->Yi, so I can use inverse warping.

---