
Subject: Re: particle detection - a way to speed up things?

Posted by dcleon@gmail.com on Wed, 28 Nov 2007 18:42:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Nov 28, 6:40 am, Ingo von Borstel <newsgro...@planetmaker.de> wrote:

> Hi there,

>

> I run an algorithm which tries to detect particles on image sequences.

> The most time consuming operation (more than half of the processing

> time) is to find the centre of all detected particles. I calculate the

> centre of each particle separately by supplying an image where only the

> i-th particle is present to "schwerpunkt2". Is there a faster way to do

> this? I put the outline of the calling routine and "schwerpunkt2" below

> for reference.

>

> Best regards,

> Ingo

>

> PRO schwerpunkt2, image, xpos, ypos, img_total=img_total, dims=dims

>

> ; Procedure returns the centre of weight (xpos, ypos) of a

> ; 2D-array (image). In order to speed up calculation, the total of the

> ; supplied 2D array (img_total) and its dimensions (dims) can be

> ; supplied, should they already be known.

>

> IF SIZE(image,/N_DIMENSIONS) NE 2 THEN BEGIN

> MESSAGE, "Number of dimensions must be exactly two.",/CONTINUE

> xpos = 0

> ypos = 0

> RETURN

> ENDIF

>

> IF NOT KEYWORD_SET(dims) THEN \$

> dims = SIZE(image,/DIMENSIONS)

>

> IF NOT KEYWORD_SET(img_total) THEN \$

> img_total = TOTAL(image)

>

> xs = dims[0] & ys = dims[1]

>

> xvec = indgen(xs)

> yvec = indgen(ys)

>

> xpos = TOTAL(xvec * TOTAL(image,2))/img_total

> ypos = TOTAL(yvec * TOTAL(image,1))/img_total

>

> END ; schwerpunkt2

```

>
> PRO detect_particles, filename, area, pos, brightness, minintbriht,
> maxsize, minsize
> image = READ_IMAGE(filename)
>      ; Now do proper noise reduction and particle enhancement using edge
> detection, and filtering with proper structuring elements such that
> particles most probable don't overlap anymore and are separated by zeros
> in the image.
> gray_image = enhance_image(image)
>
> particle_image =
> WATERSHED(255-gray_image,CONNECTIVITY=8,/LONG,nregions=n_particles)
> dims =SIZE(particle_image,/DIMENSIONS)
>
> pos = DBLARR(n_particles,2)
> area = DBLARR(n_particles)
> brightness = DBLARR(n_particles)
>
>      ; Now determine properties of all detected particles
> FOR i=0,n_particles-1 DO BEGIN
>     bin_thisparticle = particle_image EQ i
>     gray_thisparticle = particle_image * bin_thisparticle
>     xpos = 0 & ypos = 0
>     area[i-1] = TOTAL(bin_thisparticle)
>     brightness[i-1] = TOTAL(gray_thisparticle)
>     schwerpunkt2, gray_thisparticle, xpos,
> ypos,img_total=brightness[i-1],dims=dims
>     pos[i-1,0] = xpos
>     pos[i-1,1] = ypos
>     IF area[i-1] LT minsize OR area[i-1] GT maxsize OR brightness[i-1] LT
> minintbriht THEN BEGIN
>         real_particle[i-1] = 0
>         gray_image *= particle_image NE i
>     ENDIF
> ENDFOR
> END ;detect_particles
>
> --
> Ingo von Borstel          <newsgro...@planetmaker.de>
> Public Key:http://www.planetmaker.de/ingo.asc
>
> If you need an urgent reply, replace newsgroups by vgap.

```

If I'm reading your program correctly, you have a big image consisting (presumably) of a lot of empty space and numerous particles that you have identified in some way.

My guess is that the main reason your program is slow is that for each

particle you
are summing over the entire image.

I can see two ways to speed things up:

- 1) Create subsets of the image for each particle and only sum only over the subset containing the particle.
- 2) Use something like HISTOGRAM or a multi-dimensional histogram with the REVERSE_INDICES keyword (or equivalent) to get the indices associated with each particle and sum over those. The histogram command would be applied to your particle_image field with a binsize of 1 and starting at 0. See http://www.dfanning.com/tips/histogram_tutorial.html for ideas on how to approach this problem using histograms.

Of these approaches, 1 is more straightforward while 2 has the potential to speed things up more dramatically.

cheers
dave
