

ctools - Change request #2498

Make cssrcdetect more robust

06/07/2018 11:52 AM - Knödlseeder Jürgen

Status:	Closed	Start date:	06/07/2018
Priority:	Normal	Due date:	
Assigned To:	Cardenzana Josh	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	1.6.0		
Description			
Make cssrcdetect more robust, for example by			
<ul style="list-style-type: none">• smoothing the input map to a given size• determining the excess counts locally and not globally			
The robustness should be tested for example with a EGAL or GPS skymap.			

History

#1 - 08/22/2018 02:19 PM - Cardenzana Josh

- File *ds9_deg01.reg* added
- File *ds9_deg05.reg* added
- File *ds9_deg10.reg* added
- File *ds9_devel.reg* added
- File *models_deg01.xml* added
- File *models_deg05.xml* added
- File *models_deg10.xml* added
- File *models_devel.xml* added
- File *gps_cssrcdetect_comparison.png* added
- Status changed from New to Pull request

The applied changes include two new steps:

- Smoothing of the input map
- Bin-independent computation of mean and standard deviation taking into consideration only those pixels within a certain distance. This is done using a convolution with a disk kernel.

As a result of the second change, the 'maximum pixel' definition has been changed:

- Old definition: Pixel with the highest value in the map
- New definition: Pixel with the largest number of standard deviations above its respective mean.

The two definitions are the same if the radius used for computing the mean and stddev. is set to cover the entire skymap.

There are three new parameters:

- `smoothkernel`: Kernel to use for pre-smoothing the skymap (possible values are NONE, GAUSSIAN and DISK)
- `smoothparam`: Parameter for the smoothing kernel (sigma for GAUSSIAN or radius for DISK, ignored if `smoothkernel=NONE`)
- `momentradius`: All pixels within this distance of a given pixel will be used for computing its respective mean and standard deviation.

Comparing the computation time between the previous method and the updated method are as follows (note that the lack of smoothing does not dramatically impact the computation time):

Parameters:

```
inmap= counts map of the GPS survey (7200 x 280 pixels)
maxsrcs=100
threshold=3
```

Devel version:

```
real 19m14.410s
user 19m0.748s
```

2498 changes (smoothkernel=NONE, momentradius=1 deg):

```
real 26m4.866s
user 25m37.019s
```

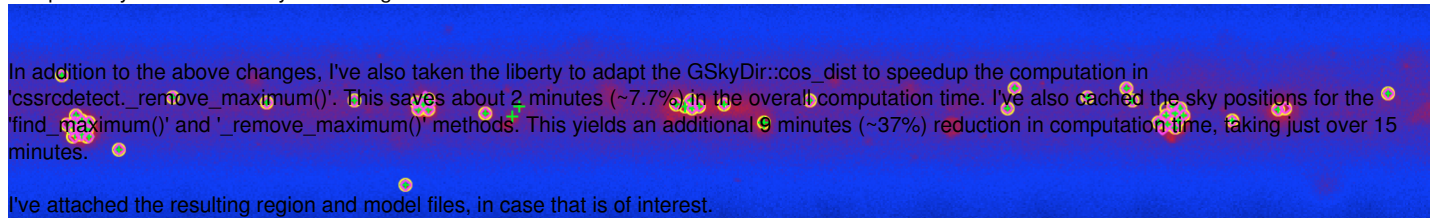
2498 changes (smoothkernel=NONE, momentradius=5 deg):

```
real 26m45.997s
user 26m13.691s
```

2498 changes (smoothkernel=NONE, momentradius=10 deg):

```
real 26m18.184s
user 25m42.642s
```

Here's an image comparing the results for the two methods showing the detected source positions for the devel branch (green crosses) and the changes branch with momentradius=5 deg (magenta diamonds) and 10 deg (yellow circles). There appear to still be some missed sources, but these can probably be recovered by increasing the value of maxsrcs.



In addition to the above changes, I've also taken the liberty to adapt the `GSkyDir::cos_dist` to speedup the computation in `'cssrcdetect_remove_maximum()'`. This saves about 2 minutes (~7.7%) in the overall computation time. I've also cached the sky positions for the `'find_maximum()'` and `'_remove_maximum()'` methods. This yields an additional 9 minutes (~37%) reduction in computation time, taking just over 15 minutes.

I've attached the resulting region and model files, in case that is of interest.

Pull branch:

Josh Cardenzana/ctools: 2498-cssrcdetect_local_selection

#2 - 09/10/2018 05:20 PM - Knödseder Jürgen

- Assigned To set to Cardenzana Josh

I'm about to merge in your changes. In which GammaLib branch are the changes in GSkyDir::cos_dist?

I'm also wondering whether we should switch the new parameters to hidden, at least the ones for smoothing. They all affect more the inner workings of the source detection and are there to tune the result. What about the following?

```
momentradius = 1.0  
smoothkernel = "GAUSSIAN"  
smoothparam = 0.05
```

Maybe we should also change names to make them a bit shorter. We have for example exclrad for the exclusion radius, could therefore also have momentrad? I think in HESS they call this radius also the "correlation radius", which could then become corrad. I cannot come up with better names for the smoothing parameters, but maybe smooth_kernel and smooth_radius would be more in line with other parameters (such as fit_pos and fit_shape)?

#3 - 09/10/2018 10:37 PM - Knödseder Jürgen

- % Done changed from 0 to 100

Here is finally what I implemented

```
avgrad, r, h, 1.0,,,"Averaging radius (deg) for significance computation"  
corr_rad, r, a, 0.1,,,"Correlation kernel radius (deg)"  
corr_kern, s, h, DISK,NONE|DISK|GAUSSIAN,,"Correlation kernel to apply to input sky map"
```

The correlation radius is in fact the smoothing kernel radius, which I set by default to 0.1 deg. This is the only parameter that will be queried. The smoothing kernel is by default set to DISK, which avoids additional tails in sources. I set the averaging radius to 1 deg which is the typical scale of the counts variations over the field of view. Hopefully with this also sources at the edge of the field of view will be correctly detected.

#4 - 09/11/2018 11:20 AM - Cardenzana Josh

These changes all look good. I wasn't exactly sure what to call the parameters. I figured you would have a better idea of proper names so I tried to make them descriptive enough that you would understand what they're used for when you did your evaluation of the pull request.

I actually didn't make any changes to GammaLib, I just changed one of the computations in cssrcdetect from computing 'dist' to computing 'cos_dist' for speed. So there's no need to merge any changes from GammaLib.

#5 - 09/11/2018 01:56 PM - Knödseder Jürgen

- Status changed from Pull request to Closed

Code is integrated in devel branch.

#6 - 09/11/2018 01:56 PM - Knödseder Jürgen

user#223 wrote:

These changes all look good. I wasn't exactly sure what to call the parameters. I figured you would have a better idea of proper names so I tried to make them descriptive enough that you would understand what they're used for when you did your evaluation of the pull request.

I actually didn't make any changes to GammaLib, I just changed one of the computations in cssrcdetect from computing 'dist' to computing 'cos_dist' for speed. So there's no need to merge any changes from GammaLib.

Great, thanks for clarifying.

Files

ds9_deg01.reg	9.62 KB	08/22/2018	Cardenzana Josh
ds9_deg05.reg	10 KB	08/22/2018	Cardenzana Josh
ds9_deg10.reg	9.84 KB	08/22/2018	Cardenzana Josh
ds9_devel.reg	9.64 KB	08/22/2018	Cardenzana Josh
models_deg01.xml	60.4 KB	08/22/2018	Cardenzana Josh
models_deg05.xml	60.4 KB	08/22/2018	Cardenzana Josh
models_deg10.xml	60.4 KB	08/22/2018	Cardenzana Josh
models_devel.xml	60.4 KB	08/22/2018	Cardenzana Josh
gps_cssrcdetect_comparison.png	1.28 MB	08/22/2018	Cardenzana Josh