

Specificații tehnice pentru pachetul de programe Umbrella

1. Acronime și componente

a) Pentru datele obținute avem următoarele acronime:

- BFD -- bias, flat, dark (darkflats, etc.). Imagini de calibrare
- RAWSCI -- raw science. Imaginile brute obținute de diverse instrumente montate pe diferite telescoape.
- REDSOI -- reduced science (BFD, plate solving, field correction). Imaginile calibrate: a) corecția de bias și dark, b) corecția de flat, c) rezolvarea soluției astrometrice, d) corecția distorsiunilor câmpului.
- ImageDetection -- structura de date pentru o detecție (sursa luminoasă) pe o imagine.
- Tracklet -- structura de date pentru un (candidat de) obiect observat pe mai multe imagini; conține o listă de ImageDetection.

b) Pentru componentele pachetului de programe Umbrella se folosește următoarea terminologie (acestea sunt în diferite stadii de testare și dezvoltare conform planificării proiectului)

1. Umbrella -- suita de programe pentru detecție de asteroizi.
2. Umbrella2 -- biblioteca open-source (<https://gitplanet.astro.ro/>).
3. Webrella -- interfața web.
4. IPP – (Image Processing Pipeline) subprogramul de reducere BFD + apelat AstrOmatic pentru plate solving și field correction.
5. STU – (Synthetic Tracking Unit) subprogramul pentru aplicarea metodei de synthetic tracking.
6. Pipeline Umbrella (v4) - o serie de subprograme (denumite componente în terminologia Umbrella) conectate între ele, care procesează un set de fișiere (de obicei imagini) într-un rezultat (de obicei imagini sau rapoarte de detecție); termenul de v4 se referă că acestea au fost introduse în versiunea 4 a suitei Umbrella (dezvoltată majoritar sub proiectul Parasol).
7. Pipeline Host - tip de program ce încarcă și assemblează subprograme Umbrella v4 în pipeline-uri pe care ulterior le rulează.
8. BasicHost - un program de tip Pipeline Host, generic, pentru a fi folosit acolo unde nu este necesară dezvoltarea unui Pipeline Host specific, mai avansat.

2. Pachet installer Umbrella

Am instalat un sistem de update pentru Umbrella pe sistemul parasol1. Acesta poate rula pe sistemele de operare ArchLinux și Ubuntu. Pentru a-l folosi aveți nevoie de un access token de la:

[https://gitplanet.astro.ro/-/profile/personal_access_tokens\(scope_read_api\)](https://gitplanet.astro.ro/-/profile/personal_access_tokens(scope_read_api))

după configurare puteți download-a componente Umbrella direct pe sistem fără a mai fi nevoie să intrați cu browser-ul pe GitLab. Dacă vreți să îl instalați pe PC-ul personal sau un alt PC , pachetele de instalare pentru Ubuntu și ArchLinux se găsesc aici:

<https://gitplanet.astro.ro/umbrella/umbrella-common/-/packages>

Pentru a instala BasicHost, proiectul/componenta e umbrella-common și subcomponenta este basichost. Pentru instrucțiuni:

umbrella-update help [comandă]

Exemple:

umbrella-update upstream add gitbrella https://gitplanet.astro.ro/

umbrella-update upstream fetch

umbrella-update install --local bh umbrella-common v0.2.2-rc1 basichost -u gitbrella

umbrella-update install --local bh pipeline-stu v0.3.0 -u gitbrella

3. Explicații setări STU

Conținut redactat: Parametrii algoritmului nu sunt publici la acest moment. Aceștia vor fi disponibili după ce articolul științific va fi publicat.

4. Cum se rulează IPP

IPP se poate rula local dacă există instalat AstrOmatic și Astrometry.net. Se rulează ca (în linie de comandă Linux):

```
$ mono --debug <cale>/Umbrella2.ImagePipe.exe <input> <output> < folder temporar>
```

unde:

- *<input>* este folder-ul cu flat, dark și raw science (light)
- *<output>* este folder-ul unde să salveze fișierele finale
- *< folder temporar>* este folder-ul pentru fișiere temporare

Exemplu:

```
mono --debug ./bin/Umbrella2.ImagePipe.exe ./in/2100 ./out/2100/ ./work/2100/
```

Structura de foldere este definită în fișierele de configurare. IPP-ul creează master bias, dark, flat. În această versiune folderele trebuie să fie toate prezente (bias, dark, flat).

Versiunile noi de IPP sunt rulate conform secțiunii 5 din acest document.

5. Cum se rulează subprogramele Umbrella v4 folosind Basic Host

Sistemul de subprograme din Umbrella v4

Pipeline-urile din Umbrella v4 (ce folosesc interfața de pipelines din umbrella-common) sunt distribuite în biblioteci sub forma unor componente. Acestea necesită o aplicație gazdă care să

le asambleze într-un pipeline și să le ruleze. O astfel de gazdă este Basic Host-ul din umbrella-common.

Apelatul Basic Host

Comenzile următoare folosesc ca proces gazdă aplicația Basic Host. Pentru a putea folosi aceste comenzi, Basic Host trebuie apelat corect.

Următoarele convenții sunt folosite pentru comenzi:

- `$ cmd` reprezintă începutul comenzii `cmd` pe linia de comandă sub linux
- `> cmd` similar, dar sub Windows. Caracterele de la început reprezintă prompter-ul pentru comandă și nu trebuie introdus ca parte din comandă.

Pentru Linux, Basic Host se apelează cu subcomanda/argumentele `subcmd` prin:

- `$ mono --debug <cale>/BasicHost.exe subcmd`

Pe calculatoarele unde alias-ul este configurat (calculatoarul parasol), comanda se poate prescurta la

- `$ ubh subcmd`

Sub Windows, comanda este

- `> C:\<cale>\BasicHost.exe subcmd`

Pe calculatoarele unde Basic Host este adăugat la variabila PATH, comanda poate fi prescurtată la

- `> BasicHost.exe subcmd`

În acest document este folosit `$ ubh`, urmând ca utilizatorii să înlocuiască această comandă cu cea potrivită în funcție de configurația sistemului pe care lucrează.

Comanda help

Pentru mai multe detalii despre folosirea Basic Host se poate folosi comanda `help` care oferă informații suplimentare despre celelalte comenzi ale Basic Host, deoarece acesta este un ghid ce explică în detaliu doar opțiunile necesare pentru a rula subprogramele.

Exemplu:

```
$ ubh help
```

```
$ ubh help info
```

```
$ ubh help run
```

Pregătirea unui pipeline

➔ Instalare

Pentru a instala un pipeline acesta trebuie dezarhivat (păstrând numele pipeline-ului) în folder-ul `pipes` din folder-ul unde este instalat Basic Host. De exemplu, pentru STU, zip-urile `pipeline-stu-*.zip` trebuie dezarhivate în `pipes` astfel încât fișierele pentru STU să fie în folder-ul `pipes/STU`.

➔ Asamblare

Primul pas pentru a rula un pipeline este configurarea acestuia. Aceasta se realizează prin intermediul unui fișier de configurare. Implicit, acest fișier de configurare se numește `config.txt` și trebuie să se afle în folder-ul curent, dar poate fi specificat din linia de comandă. În acest fișier de configurare trebuie să existe o secțiune Pipeline, unde sunt specificate componentele din care este asamblat pipeline-ul.

Exemplu secțiune Pipeline

```
[Pipeline]
Helpers=Gpu
DetectionMechanism=STU
TrackletFilters=
ReportGenerator=BasicReporter
```

Pentru a afla ce elemente de pipeline sunt instalate și pot fi folosite în mod curent, se utilizează următoarea comandă:

```
$ ubh info list-pipelines
```

➔ **Apelatul pipeline-ului**

Basic Host este foarte configurabil iar o comandă minimală de a rula un pipeline (pe linux) este:

```
$ ubh /cale/la/input
```

O comandă mai completă, echivalentă cu cea de deasupra, este:

```
$ ubh run -c config.txt /cale/la/input -o out/ -w work/
```

Pentru pipeline-urile de procesare de imagini (IPP), mai sunt disponibile și alte opțiuni.

Exemplu comandă rulat STU

```
$ ubh run -c config.txt /cale/la/input -o out/ -w work/
```

Notă, pentru a putea rula trebuie să existe fișierul `config.txt` în care este specificat că se rulează STU

Comanda de help

```
$ ubh help run
```

Configurarea unui pipeline

Pe lângă asamblarea pipeline-ului, elementele individuale se pot configura. Pentru a afla ce setări sunt disponibile se pot obține aceste informații chiar de la pipeline. Comanda pentru a face acest lucru este:

```
$ ubh info inspect-element elem, unde elem este elementul pentru care vrem să aflăm  
configurarea
```