IPM School on Gravitational Lensing Techniques July 2008

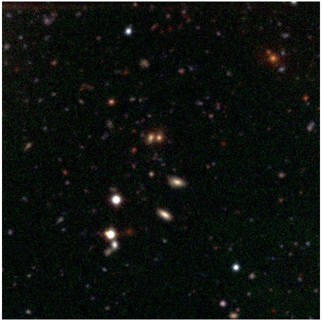
Lecture II: Image simulations

Thomas Erben
Argelander-Institute for Astronomy / Bonn

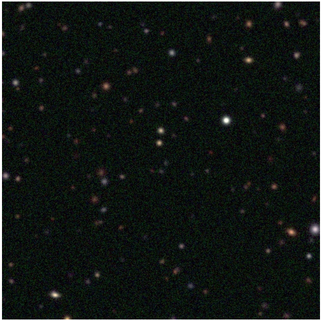
03. July 2008

Image Simulations with E. Bertins `Skymaker`

WFI 'empty' region (3.′0 × 3.′0)



`Stuff/Skymaker` simulation



`Stuff/Skymaker` (developed by E. Bertin/Terapix) allow the creation of *realistic* (ground-based) simulations to train yourself in the fields of catalogue creation, photometric redshift estimation and weak lensing studies.

Unfortunately no official documentation is available! Most of the following is my understanding from investigating the source codes!

Features of Stuff/Skymaker

Stuff and Skymaker are intended to mimic the outcome of *real* ground-based CCD observations

- You mainly specify parameters of the telescope, camera and observing conditions (observed passbands, telescope mirror size, seeing, sky-brightness)
- Stuff closely mimics basic galaxy properties (distribution of spectral types, properties of galaxy bulge and disk components, observed magnitudes)
- Skymaker models galaxies with simple analytical profiles (de Vaucouleurs and elliptical)
- Skymaker includes a state-of-the-art modeling of the telecope PSF (Airy Pattern of the telescope entrance pupil, atmospheric seeing, optical aberrations, ...) (currently no variable PSF)
- Skymaker mimics various relalistic noise contributions (blooming effects on saturated pixels, sky brightness, pixel noise)

Motivation for Image Simulations

The combination of `Stuff` and `Skymaker` allows us to create photometric, multi-colour object catalogues and FITS images mimicing observation with a telescope.

- We know exactly what goes into simulated catalogues and images (redshifts of galaxies, photometry, galaxy elliptícties and associated errors, PSF of images, modelisation of galaxies and stars, noise in FITS images etc.)
- We can train ourselfs in the application of scientific analysis software (object detection, photometric redshifts, shear extraction algorithms, image processing algorithms)
- We can validate and test programs and algorithms under development
- The availablility of catalogues and images allows a progressive testing/training of the influence of various noise contributions (testing step by step and quick isolation of potential problems)

Stuff/Skymaker Philosophies

- Stuff and Skymaker are stand-alone C-executables controlled via ASCII-text configuration files (such as other programs from E. Bertin: SExtractor, Swarp)
- They can easily be integrated in shell script wrappers and pipelines
- Stuff is repsonsible for the creation of a galaxy object catalogue in ASCII format. It tries to mimic a simple *field* galaxy population (magnitudes, galaxy sizes, ellipticities, spectral types etc.)
- Skymaker creates optical image simulations produced by Stuff or *other* programs; optimised to test photometric properties under *realistic* observing conditions (instrument properties, PSF)

The separation of catalogue and image creation allows us to modify the input catalogue before FITS images are produced (effects of gravitational lensing)

The Stuff Program (version 1.11)

The user *needs* to specify:

- The angular extent of the simulation and basic telescope properties:

```
CATALOG_NAME      U38.list,...
IMAGE_WIDTH       2048  # (in pix)
IMAGE_HEIGHT      2048  # (in pix)
PIXEL_SIZE        0.238 # (in arcsec)
(COLLECT_AREA      2.8   # in m^2)
(GAIN              2.4   # (in e-/ADU))
```

- The observed passbands and the used magnitude system:

```
PASSBAND_OBS      wfi/WFI_U38,wfi/WFI_B,...
SED_CALIB         Vega  # Vega or AB
```

There are many other parameters which, in my opinion, do not need to be touched for most applications!

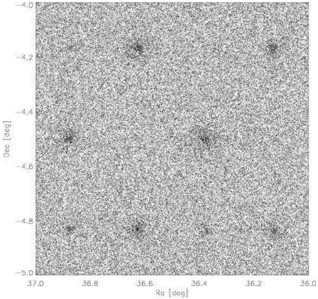
The `Stuff` Program (II)

A simulated field galaxy catalogue is created:

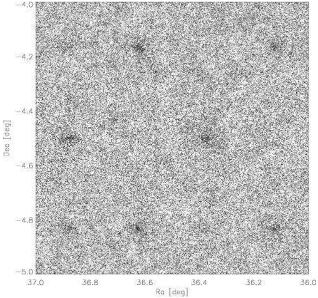
- Galaxies are distributed randomly in a *Poissonian way* in redshift slices from $0 < z < 20$ (no galaxy clustering!)
- The number and absolute luminosity of the galaxy types E,S0,Sab,Sbc,Scd,Sdm are obtained from a Schechter luminosity function
- galaxy profiles are modeled as superposition of de Vaucouleurs (bulge coponent) and exponential (disk component) profiles: $I = I_0 \exp\left(-7.67\left[(r/r_0)^{1/4} - 1\right]\right)$; $I = I_0 \exp(r/r_e)$ r_0 and r_e are calculated from scaling relations with the absolute magnitude
- The apparent magnitude of each galaxy is calulated taking into account: distance modulus, observed passbands, galaxy templates, K-correction, disk extinction
- Output is an ASCII file with the galaxy entries

The `Stuff` Program: Uniform Galaxy Distributions

uniform galaxy distribution



clustered galaxy distribution



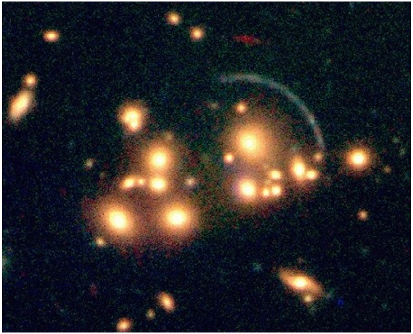
(Figures from Olsen et al. 2006)

`Stuff` produces a uniform galaxy distrbution; it should be *clustered* for applications such as investigating the efficiency of cluster search algorithms (see e.g. Soneira & Peebles 1978).

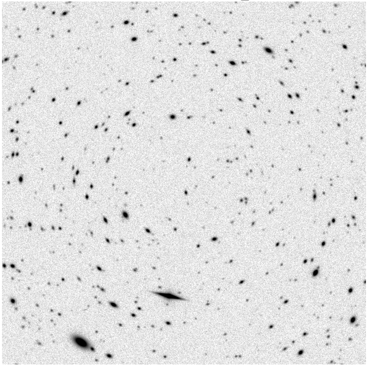
The `Stuff` Program: Redshift Slice Galaxy Distributions

The *simple* redshift slice distribution from high to low redshifts does not allow to include strong lensing features. For this, ray tracing would be necessary.

Giant Arc in CL2244



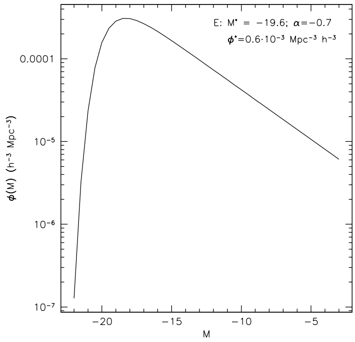
Giant 'Arcs' in `Skymaker`

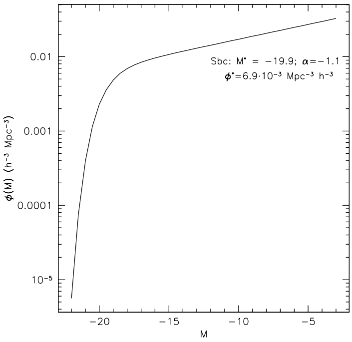


The `Stuff` Program: The Distribution of Galaxy Types

The number and absolute luminosity of galaxies is taken from a Schechter luminosity function in the blue band:

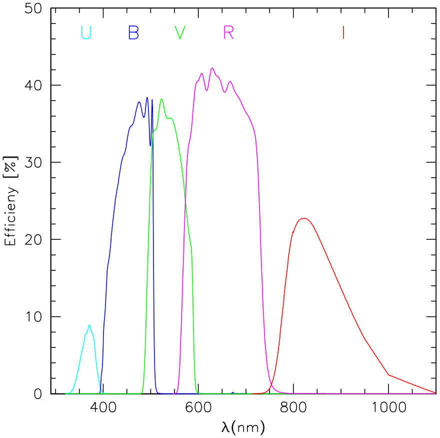
$$\phi(L)dL = \phi^* \left(\frac{L}{L^*}\right)^{\alpha} \exp\left(-\left(\frac{L}{L^*}\right)\right) d\left(\frac{L}{L^*}\right)$$



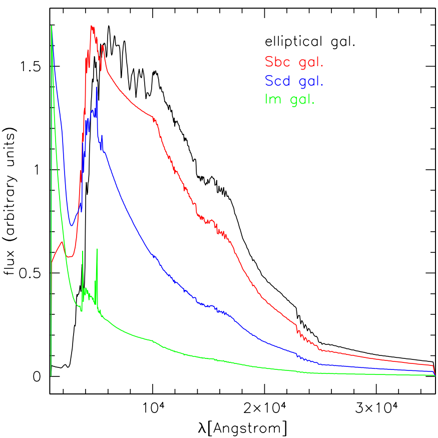


The `Stuff` Program: SEDs and Instrument Filter Curves

WFI filter curves



`Stuff` SEDs



Arbitrary filter curves and SEDs can be given to `Stuff`. As SEDs it uses the Coleman et al. (1980) templates.

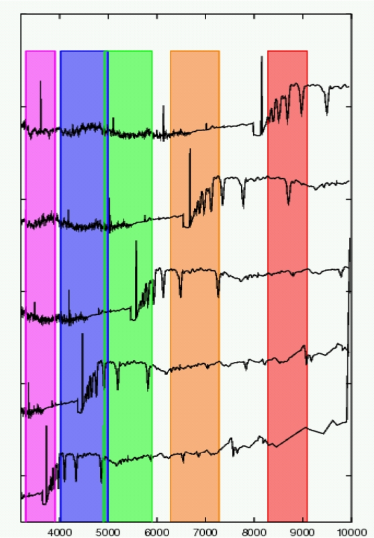
The `Stuff` Program: Photometry

Given is the total luminosity M_B in a *reference band*, here the blue. Together with the knowledge of the spectral energy distributions and the redshift of a galaxy the *observed* flux m in any passband can be calculated:

$$m = M_B + DM(z) + K(z),$$

with $DM(z)$ the distance modulus and $K(z)$ the *k-correction* (no extinction included in the formula!).

Redshifted Templates
(from Mellier 2006)



The Output `Stuff` Catalogue

`Stuff` can produce ASCII catalogues for an arbitrary number of observed filters at the same time (multi-colour catalogues):

```
200   611.634 1421.220  24.6608 ......   3.50661
200   717.635  -40.110  26.720  ......   3.47838
200   986.630  -50.237  25.2722 ......   2.95611
```

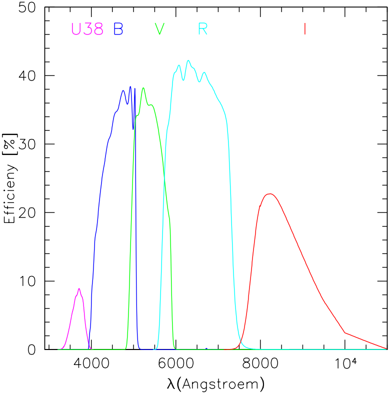
Important Columns for us (Stuff version 1.11!):

- two and three: pixel position in simulated image
- four: apparent magnitude
- seven/eight: bulge axis ratio and position angle
- ten/eleven: disk axis ratio and position angle
- twelve: galaxy redshift

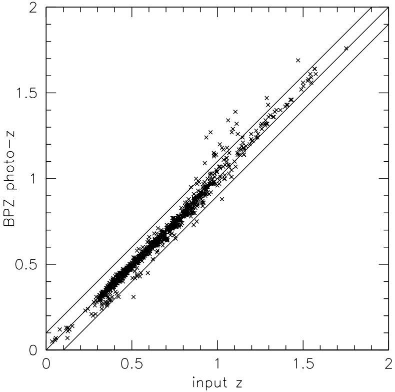
Note that `Stuff` catalogues contain *only* galaxies. A modeling of stellar sources is currently not implemented in `Stuff` but this is probably added in the future.
Note that bulge and disks have, in general, different ellipticities (shear simulations)

Photo-z Simulations with `Stuff` Catalogues

WFI filter curves



BPZ photo-z estimates



A catalogue with the characteristics of our A226 data was simulated and photo-z were estimated with `Bayesian Photometric Redshifts (BPZ; Benitez 2000)`

The `Skymaker` program (I)

!! Many internals of this program are not yet clear to me !!

The `Skymaker` program takes the `Stuff` catalogues and creates FITS images. The user *needs to specify* (the absolute minimum):

- The extent of the simulation, basic telescope and exposure properties (needs to match parameters for `Stuff`!):

```
IMAGE_NAME        field.fits
IMAGE_SIZE        2000,2000
GAIN              2.40          # in (e-/ADU)
EXPOSURE_TIME     620.0         # (in s)
MAG_ZEROPOINT     26.0          # (in ADU/s)
PIXEL_SIZE        0.238         # (in arcsec)
SEEING_FWHM       0.7           # (in arcsec)
BACK_MAG          20.0          # (in mag/arcsec^2)
```

- Details on the stellar counts:

```
STARCOUNT_ZP      3e4       # stars /deg2 <= MAG_LIM
STARCOUNT_SLOPE   0.2       # (dexp/mag)
MAG_LIMITS        17.0,26.0 # magnitude range
```

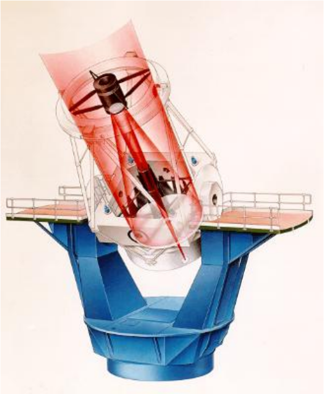
The `Skymaker` program (II)

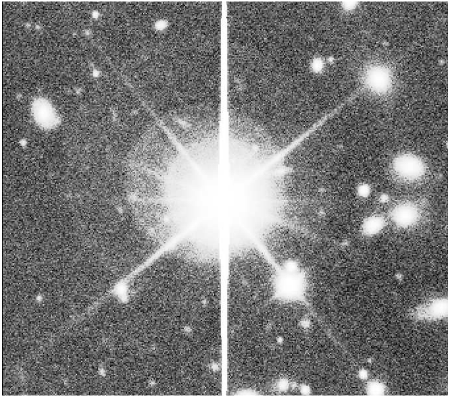
A simulated FITS image is created:

- The PSF is internally constructed taking into account: telescope aperture, atmospheric seeing, optical aberrations, telescope jittering
- *Stars* are distributed *randomly* with a magnitude distribution of a simple power law (no realistic colour information in case of multi-colour exposures); we can add stars to the `Stuff` catalogues ourselfs
- Galaxies are modeled as superposition of de Vaucouleurs and exponential disks, convolved with the PSF and put to the pixel grid
- Noise and blooming effects are added
- Result is an output catalogue (listing positions of galaxies and stars) and a FITS image with `Skymaker` configuration information in the FITS header

`Skymaker` is very careful about pixelisation effects (PSF sampling etc.)

PSF Realisation in `Skymaker`





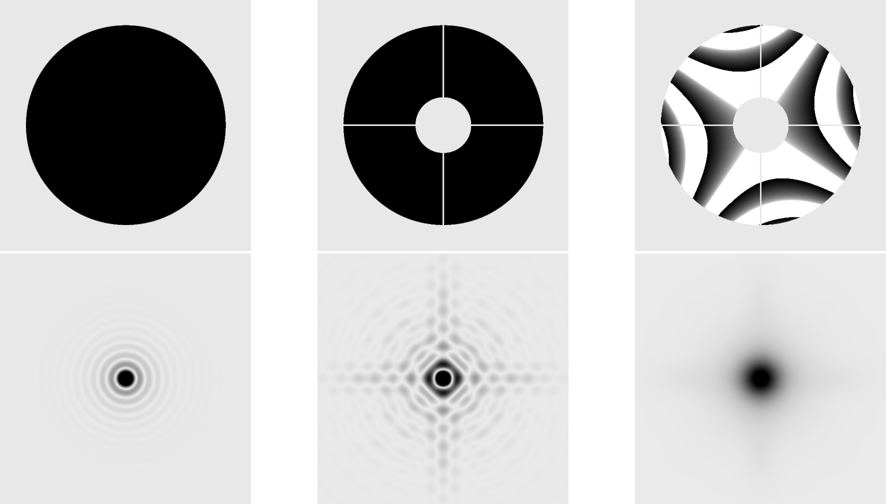
The intrinsic light distribution is smeared with the PSF P (Earth atmosphere, telescope optics):

$$I^{\text{obs}}(x) = \int d^2y \, I^{\text{intr}}(y) P(x - y)$$

The Construction of `Skymaker` PSFs

`Skymaker` has very interesting check images!

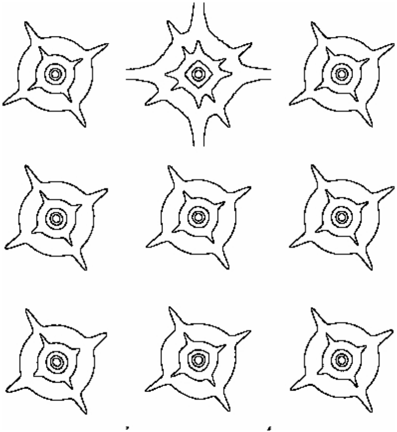
mirror aperture | sec. mirror; spider arms | Astigmatism

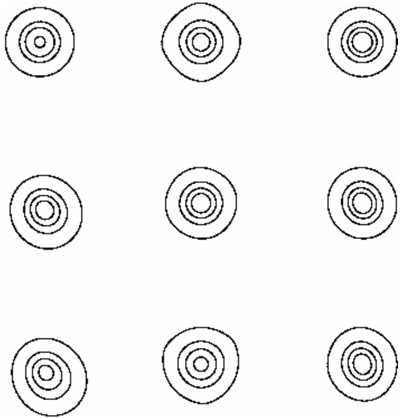


Airy pattern | modified Airy | final PSF

Shape of `Skymaker` PSFs

Contours for `Skymaker` PSFs with various optical aberrations (used for weak shear simulations)

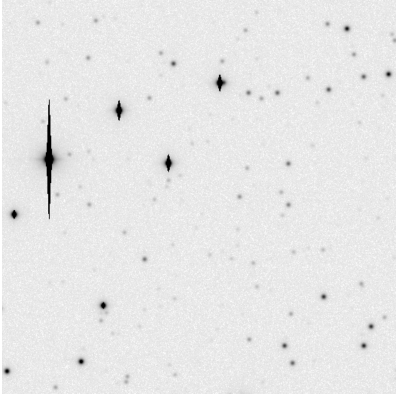


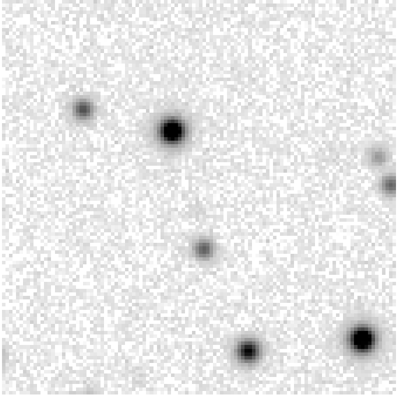


The cores of the PSFs contain more than 80% of the light!

Noise and Artefacts in `Skymaker`

Blooming effects occur if the number of electrons in a pixel exceeds the `WELL_CAPACITY`; sky noise is Poissonian





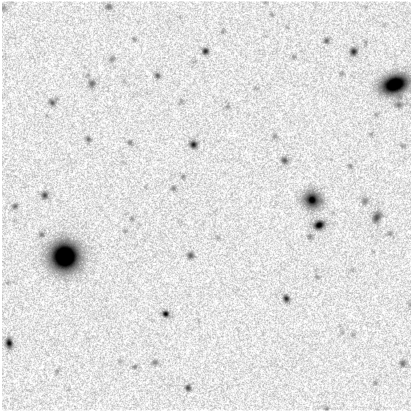
The level of noise is determined by the provided sky brightness (Typical values (Unit is $mag/('')^2$): U: 21.6; B: 22.3; V: 21.2; R: 20.3; I: 19.2)

The output `Skymaker` Image

FITS Header

```
NAXIS1   =                 2000
NAXIS2   =                 2000
.
.
GAIN     =             9600.000
WELLCAPA=          2.400000E+06
SATLEV   =              65535.0
RON      =                  4.8
EXPOTIME=                 500.0
MAGZERO =                  23.1
PIXSIZE =                   0.2
.
.
```

Image Pixel Data



Two pitfalls with `Skymaker`: The *LONG* exposure

To simulate a *LONG* exposure it is *NOT* a good idea to just put a very high exposure time! All pixels would be saturated as in a real exposure.
Instead we simulate a long exposure as a sequence of *N* single frames with a small exposure time. To this the `GAIN` (and the `WELL_CAPACITY`) have to be multiplied by according factors:

1. An exposure of 500s (intrinsic `GAIN` is 2.4):
   `GAIN`: 2.4; `EXPOSURE_TIME`: 500 This mimics one exposure with an exposure time of 500s.
2. An exposure of 50000s (intrinsic `GAIN` is 2.4):
   `GAIN`: 240; `EXPOSURE_TIME`: 500
   This mimics an exposure of 50000s composed of 100 images a 500s.

The `MAG_ZEROPOINT` confusion

This is a general source for confusion, not only with `Skymaker`

The provided zeropoint `MAG_ZEROPOINT` for `Skymaker` is normalised for an exposure of 1 second, i.e. the unit of the zeropoint is `ADU/s`. However, the resulting `Skymaker` image is normalised to the exposure time of *N* seconds. Hence, to obtain correct magnitudes the zeropoint which needs to be given to `SExtractor` is:

$$\mathtt{MAGZP}_{\mathtt{SExtractor}} = \mathtt{MAG\_ZEROPOINT} + 2.5 \log(N)$$

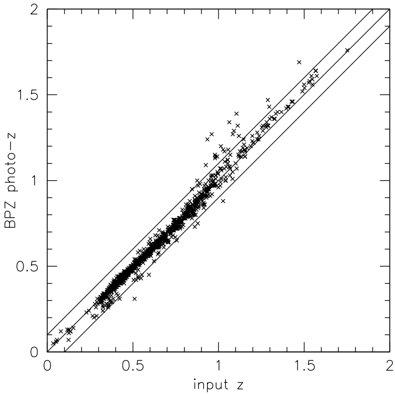
Alternatively, you can normalise the `Skymaker` image to an exposure time of 1s and avoid recalculating the zeropoint:

1. Subtract the Sky-background
2. Divide the pixel values by the exposure time

All resulting co-added images from the `THELI` pipeline have the 1s normalisation and provided zeropints can directly be used.

Photo-z Simulations with `Stuff/Skymaker` Catalogues

`Stuff` photo-z estimates



`Skymaker` photo-z estimates

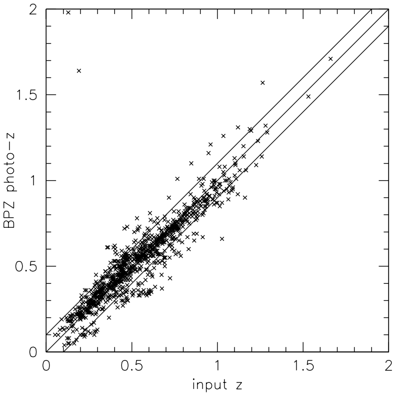
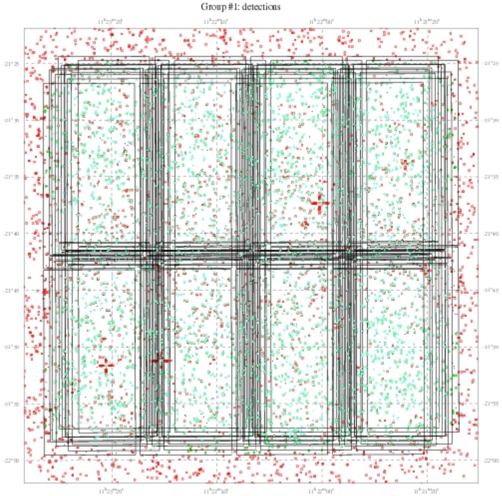


Photo-z were estimated directly from the `Stuff` catalogue (left) and from sources extracted from `Skymaker` images (right).
Main additional error contributions: Magnitude errors, colour estimates

Test of Photometric Pipelines (I)

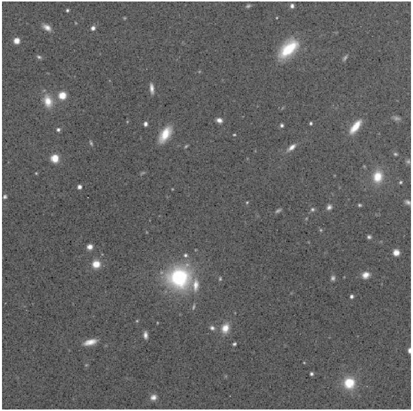
Several dithered observations of a target are *combined* to obtain deep images.



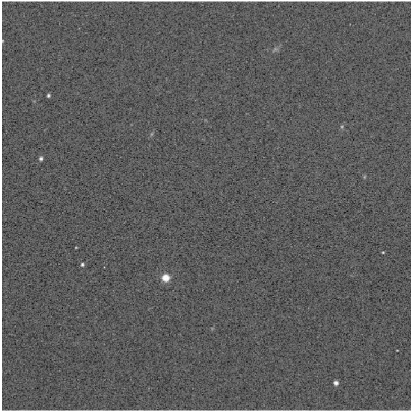
During long campaigns not all of them have good photometric

Test of Photometric Pipelines (II)

Good photometric conditions



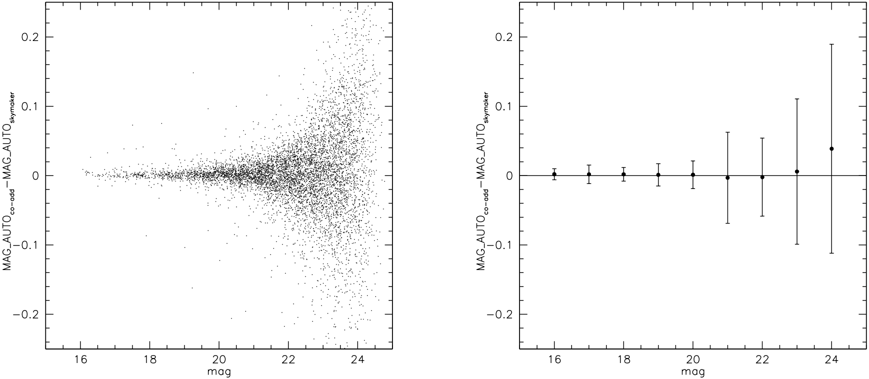
Bad photometric conditions



In a stack of many images we need to estimate flux scaling factors to bring all images to the same photometric level. `Skymaker` can help us to evaluate the accurcay of our scaling algorithms.

Test of Photometric Pipelines (III)

Magnitude comparison of 28 simulated, pipeline processed `Skymaker` images vs. one *raw* `Skymaker` output frame:



`Skymaker` tests whether our pipeline processing (sky-background subtraction, selection of objects, image co-addition) has a significant influence on magnitude estimates. The studies helped us to optimise the selection of objects

Catalogue Handling

- Most applications/programs work with formatted ASCII catalogues/tables (standard UNIX format, easy to handle by editors and script languages); good in astronomy if only a small number of columns is present/if the number of objects is not too large
- Cumbersome to handle for very large data sets
  - Slow read/write access in programs
  - catalogue entries can only be accessed via column numbers which makes writing of general purpose programs difficult (the meaning of columns may change from catalogue to catalogue)
  - One file typically only holds one object table; it is difficult to combine several tables associated with one entity

In `THELI` and for our own applications we typically use the FITS binary table `LDAC` catalogue format for object catalogues and other large tables

The `LDAC` Catalogue Format (I)

- A *close-to* FITS table standard binary format; fast read/write access in programs
- The access to data is realised by a large number of utility programs (`ldactoasc`: convert `LDAC` catalogue to ASCII format; `asctoldac`: convert an ASCII catalogue to an `LDAC` table; `ldacfilter`: filter object entries according to given conditions; `ldaccalc`: calculate new catalogue entries out of existing ones etc.)
- `LDAC` catalogues are organised in `TABLES` and `KEYS`; a `TABLE` represents a traditional ASCII table and a `KEY` can be thought of as its columns.
- One `LDAC` catalogue can host different tables!
- All catalogue entries are accessed via *meaningful* names (`ldactoasc -i ldac.cat -t OBJECTS -k Xpos Ypos`)

`LDAC`: advanced applications

Several *fast* `LDAC` utility programs for frequently used and more complex applications are available:

- association of object lists and the creation of tables with entries from both catalogues (e.g. we have catalogues of the same field observed with different cameras and we want a catalogue with common sources and all avilable magnitudes)
- Transfer of keys from one catalogue to another is possible (transform an `LDAC` table to ASCII for a photo-z program; estimate photo-zs and transform the photo-z estimates to `LDAC`; transfer new quantities to old catalogue)
- Transfer of object mask information in the form of `ds9`/`saoimage` region files to object catalogues (see tomorrows lecture)

Some basic LDAC applications

Nearly all LDAC programs share some command line options: `-i` (input catalogue) `-o` (output catalogue) `-t` (table name) `-k` (key name(s))

- `ldactoasc -i ldac.cat -t OBJECTS` (list all keys in table `OBJECTS`)
  `ldactoasc -i ldac.cat -t OBJECTS -k Xpos Ypos` (list keys `Xpos` and `Ypos`)
- `ldacfilter -i ldac.cat -o ldac_out.cat -t OBJECTS -c "(Xpos>2000)AND(Xpos<5000);"` (filter all objects with `Xpos` between 2000 and 5000)
- `ldaccalc -i ldac.cat -o ldac_out.cat -t OBJECTS -c "(mag_B-mag_I)" -n BmI "B minus I colour index" -k FLOAT` (calculate a colour index out of two magnitudes)