

Руководство по обработке эшелле-спектров с помощью IRAF
(Image Reduction and Analysis Facility)

Три золотых правила любой работы с данными:

1. Сохраняйте исходники
2. Делайте бэкапы
3. Читайте хэлп и мануал

С какими данными мы работаем?

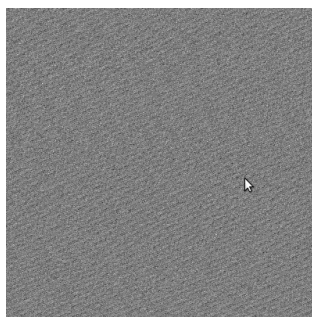
Мы работаем с ПЗС-кадрами. Они бывают нескольких типов в зависимости от того, что на них изображено.

Самые важные — изображения спектра изучаемого объекта (звезды, туманности, планеты и т. п.). Эти изображения предстоит анализировать. Но без калибровочных кадров они не представляют никакой научной ценности. К калибровочным кадрам относят:

1. Байес (bias) — кадр с изображением подложки, данные считывают без экспонирования, подложка является составляющей любого другого кадра.
2. Дарк (dark) — темновой кадр, который отражает вклад темнового тока в изображение. Его снимают с закрытым затвором, с такой же экспозицией и температурой чипа, как и у спектра объекта. Поскольку в дарке содержится и подложка, то, при наличии дарка, нет необходимости снимать байес. Если чип охлаждают до очень низких температур ($\sim 100^\circ$), вклад темнового тока становится пренебрежимо мал. В таких случаях дарк не снимают (только байес).
3. Флэт (flat) — изображение спектра лампы плоского поля. Лампа плоского поля имеет непрерывный гладкий спектр. Чувствительность пикселей в матрице не одинакова, спектр плоского поля получают, чтобы учесть неоднородность чувствительности разных пикселей и фринги (fringe) — интерференционные полосы, возникающие в красной части спектра.
4. Спектр сравнения. Спектр сравнения служит для точной привязки спектра объекта к длинам волн. В спектре сравнения должно быть много узких линий, равномерно распределенных по всему спектру, имеющих известную постоянную длину волны. Обычно используют спектры ламп с металлическим катодом, наполненных инертным газом (HeNeAr, ThAr, FeAr, Xe, Kr, Ar, CuAr). Они все имеют линейчатые спектры излучения. Также используют спектр утреннего или вечернего неба, по сути спектр солнечного света, рассеянного атмосферой. Он имеет вид обычного спектра звезды класса G с узкими линиями поглощения Солнца и земной атмосферы.

Состояние камеры и спектрографа постоянно меняется, поэтому все калибровочные изображения снимают сериями (обычно по 5 кадров каждого типа), а потом усредняют, чтоб исключить случайные ошибки и выбросы.

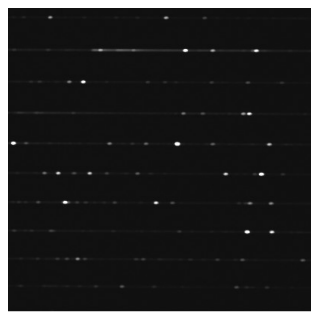
Фрагменты изображений разного типа



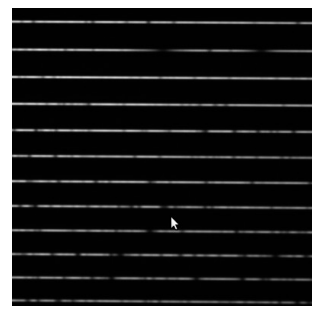
Байес



Флэт



Лампа ThAr



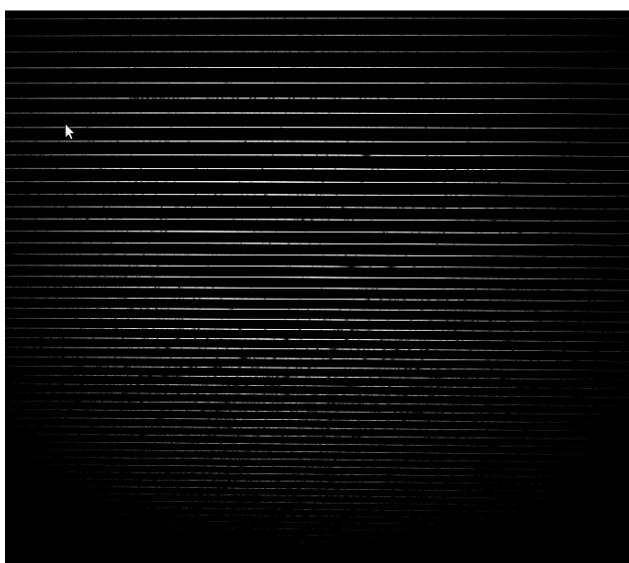
Звезда Альдебаран

Порядок работы с кадрами

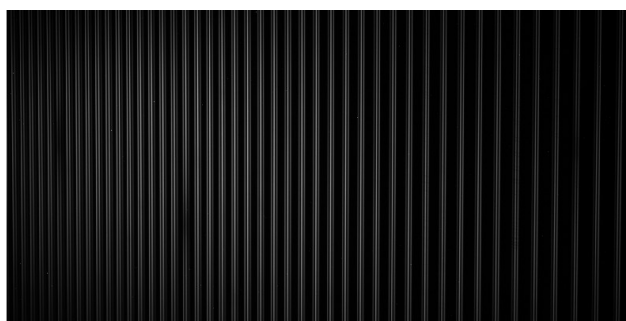
Порядок работы с кадрами таков:

1. Делаем резервную копию всех исходных файлов.
2. Кадры одной ночи сортируют по времени получения. Далее рекомендуется разделить их по времени на столько папок, сколько было снято полных серий калибровок и работать с каждой папкой в отдельности.
3. Просматриваем все кадры на предмет резких выбросов, отклонений, если есть плохие кадры — удаляем их.
4. Обрезаем у всех кадров оверскан и поворачиваем так, что длина волны увеличивается снизу слева вверх направо при любой ориентации порядков (вертикально или горизонтально).
5. Комбинируем супер-байес из серии байесов и вычитаем его из всех остальных изображений (объект, флэт, спектр сравнения). Если есть не байес, а дарк, то делаем то же самое с дарком. Ставшие ненужными кадры убираем в архив.
6. Комбинируем суперфлэт из серии флэтов и спектр сравнения из серии спектров сравнения. Лишние кадры убираем в архив.
7. Экстрагируем флэт и спектр сравнения. Лишние кадры убираем в архив.
8. Внимательно смотрим на спектр объекта. Если есть битые пиксели — корректируем. Если есть космики (обычно при длительных экспозициях) — убираем. Убираем рассеянный свет, опорное изображение для поиска апертур — флэт. Лишние кадры — в архив.
9. Экстрагируем спектр объекта, опорное изображение — флэт. Все лишнее — в архив.
10. Нормализуем флэт: исключаем инструментальную кривую и форму спектра плоского поля, остаются только фринги и попиксельная неоднородность.
11. Делим спектр объекта на нормализованный флэт. Все лишнее в архив.
12. Нормализуем спектр объекта.
13. Строим дисперсионную кривую по спектру сравнения.
14. Указываем спектр сравнения как опорный для всех кадров с объектами и привязываем спектры объектов к шкале длин волн.
15. Все, спектр объекта готов к дальнейшему анализу. При необходимости еще можно учесть барицентрическую поправку.

Правильная ориентация спектра



Порядки вдоль строк



Порядки вдоль колонок

Начало работы в IRAF

IRAF запускается непосредственно из терминала Linux, но графический терминал IRAF работает только из-под терминала *xgterm*.

После входа в Linux запускается терминал и командой

```
> xgterm&
```

запускается терминал *xgterm*. Тер **&** позволяет закрыть терминал, не прерывая работы *xgterm*.

Узнать, что находится в папке, где Вы сейчас находитесь, можно с помощью команды

```
> ls
```

Перемещение вниз по дереву папок осуществляется с помощью команды

```
> cd name (name – имя папки)
```

Например:

```
krussh@krussh-laptop:~$ ls
Desktop Downloads IRAF Pictures Templates
Documents examples.desktop Music Public Videos
krussh@krussh-laptop:~$ cd IRAF
krussh@krussh-laptop:~/IRAF$ ls
20100501 login.cl login.cl~ uparm
krussh@krussh-laptop:~/IRAF$
```

Переход вверх по дереву папок на один уровень командой

```
> cd ..
```

Скорее всего, после запуска *xgterm* Вы окажетесь в своей домашней папке (*/home/user*), запуск IRAF возможен только из директории IRAF.



```
krussh@krussh-laptop: ~/IRAF
krussh@krussh-laptop:~$ cd IRAF
krussh@krussh-laptop:~/IRAF$
```

Переходим в эту директорию и запускаем IRAF командой

```
> cl
```

Мы видим приветственную страницу IRAF и список основных пакетов:

```

krussh@krussh-laptop: ~/IRAF
NOAO/IRAFNET PC-IRAF Revision 2.14 Fri Nov 30 15:27:05 MST 2007
This is the RELEASED version of IRAF V2.14 supporting PC systems.

Welcome to IRAF. To list the available commands, type ? or ??. To get
detailed information about a command, type `help <command>'. To run a
command or load a package, type its name. Type `bye' to exit a
package, or `logout' to get out of the CL. Type `news' to find out
what is new in the version of the system you are using.

Visit http://iraf.net if you have questions or to report problems.

The following commands or packages are currently defined:

  dataio.    language.  obsolete.  softtools.  tables.
  dbms.     lists.     plot.      stsdas.     utilities.
  images.   noao.     proto.     system.

ecl> █

```

В IRAF можно выполнять команды Linux, для этого перед командой пишут символ **!**, например Вы можете запустить программу просмотра фитс-файлов DS9 следующим образом:

```
ecl> !ds9&
```

перемещаться по дереву папок с помощью команды **cd**, просматривать содержимое папок командой **ls**.

Символ **#** обозначает начало комментария, конец комментария — конец строки

Некоторые команды IRAF не привязаны к пакетам, и Вы можете их вызвать в любое время, а некоторые входят в конкретные пакеты и работают только внутри них. Для входа в пакет просто наберите его имя, например:

```

  dataio.    language.  obsolete.  softtools.  tables.
  dbms.     lists.     plot.      stsdas.     utilities.
  images.   noao.     proto.     system.

```

```
ecl> noao
```

```

  artdata.    digiphot.    nobsolete.    onedspec.
  astcat.     focas.      nproto.      rv.
  astrometry. imred.      observatory  surfphot.
  astutil.    mtlocal.    obsutil.     twodspec.

```

Некоторые пакеты содержат в себе другие, это часто обозначается точкой после названия пакета (т.е. images.imutil).

Чтобы выйти из пакета, наберите команду **bye**. Названия пакетов, команд и их параметров необязательно писать полностью, достаточно столько первых символов в названии, чтоб можно было однозначно отличить одну команду от другой, например: **bu** вместо **bye**, **ech** вместо **echelle**, и т. п.

В IRAF очень простая и полная справка по всем командам. Чтобы увидеть список команд/пакетов, принадлежащих пакету, в котором вы находитесь, служит команда **?** Чтобы увидеть список пакетов со списком команд, доступных в каждом из них - **??** Чтобы увидеть список пакетов с кратким описанием их функционала — **help** Чтобы увидеть справку по конкретной команде или пакету — **help <имя команды/пакета>**. Вся справка приведена на английском языке. Наиболее полные описания и справка на русском языке есть здесь:

<http://www.astronet.ru/db/msg/1216409> .

Для работы с файлами:

С большим числом файлов удобно работать двумя способами: с помощью маски и с помощью списка. Маска выглядит, например, так: ***00.txt**. Здесь ***** - отличающаяся часть в

имени файла, 00.txt - общая. Можно создать список файлов (текстовый) и использовать его как аргумент команды @list, здесь @ обозначает, что действие должно быть применено ко всему списку.

```
>del *trash.fits # удалить файлы по маске  
>ls *.fts > list # создать список файлов  
>imstat name.fits # сведения о файле  
>imhead name.fits # показать фитс-хедер файла
```

Создать список файлов с их параметрами из фитс-хедера, можно ввести условие (здесь только кадры в фильтре R) :

```
>hselect @filelist $I '@"FILTER"=="R"' > filterlist  
>imrename @oldnames @newnames # если нужно переименовать много файлов  
>rename *.fts fits field=extn # изменить имя/расширение большого числа файлов
```

Арифметические действия с файлами, вторым аргументом может быть как файл, так и число:

```
>imarith name.fits + name2.fits new.fits
```

Файлы обязательно должны иметь одинаковую размерность!

Для работы с командами IRAF:

```
>unlearn <команда/пакет> #установить для пакета/команды параметры по  
умолчанию  
>epar <команда/пакет> # редактировать параметры команды/пакета
```

Начальная обработка изображений

Включает в себя следующие процедуры:

1. Делаем резервную копию всех исходных файлов.
2. Кадры одной ночи сортируют по времени получения. Далее рекомендуется разделить их по времени на столько папок, сколько было снято полных серий калибровок и работать с каждой папкой в отдельности.
3. Просматриваем все кадры на предмет резких выбросов, отклонений, если есть плохие кадры — удаляем их.
4. Обрезаем у всех кадров оверскан и поворачиваем так, что длина волны увеличивается снизу слева вверх вправо при любой ориентации порядков (вертикально или горизонтально).
5. Комбинируем супер-байес из серии байесов и вычитаем его из всех остальных изображений (объект, флэт, спектр сравнения). Если есть не байес, а дарк, то делаем то же самое с дарком. Ставшие ненужными кадры убираем в архив.
6. Комбинируем суперфлэт из серии флэтов и спектр сравнения из серии спектров сравнения. Лишние кадры убираем в архив.

Сортировка кадров

Кадры одной ночи сортируют по времени получения. Далее рекомендуется разделить их по времени на столько папок, сколько было снято полных серий калибровок и работать с каждой папкой в отдельности.

Если файлы имеют неподходящее расширение, его можно быстро поменять.

```
>rename *.fts fits field=extn
```

Также посмотреть/поменять глобальные настройки можно в файле login.cl, которым запускается IRAF:

```
#set   cmbuflen      = 512000
#set   min_lenuserarea = 64000
set   imtype        = "fits"
#set   imextn        = "oif:imh fxf:fits,fit,mt,fts plf:pl qpf:qp stf:hhh,??h"
```

Сделать список файлов в папке с указанием любых ключевых полей, в данном случае получаем файл datelist, в котором указаны название, тип изображения, время начала экспозиции, датаа начала экспозиции, продолжительность экспозиции для каждого файла:

```
>hselect *.fits $I,IMAGETYPE,TIME-OBS,DATE-OBS,EXPTIME > datelist
```

imrename @oldnames @newnames # если нужно переименовать кучу файлов

Есть очень полезная команда ccdlist в пакете noao-imred-ccdred, чтобы посмотреть, какие файлы лежат в рабочей папке. Аргумент команды - список файлов или маска *.fits

```
ccdred> ccdlist
CCD images to listed: @list
31t.fits[2068,2072][real][object][:ThAr
32t.fits[2068,2072][real][object][:ThAr
33t.fits[2068,2072][real][object][:ThAr
35o.fits[2068,2072][real][object][:AlphaBoo
38o.fits[2068,2072][real][object][:P_Cyg
39o.fits[2068,2072][real][object][:89_Her
40t.fits[2068,2072][real][object][:ThAr
41t.fits[2068,2072][real][object][:ThAr
42t.fits[2068,2072][real][object][:ThAr
sflat.fits[2068,2072][real][object][:FlatField
ccdred>
```


Обрезку области оверскана и разворот для всех кадров

можно произвести с помощью команд **imcopy** и **rotate**. Необходимо чтобы изображения спектра (и всех калибровочных кадров) были ориентированы следующим образом – покраснение вверх и вправо.

Например:

```
ecl> imcopy name[1:2048,1:2048] new_name  
//копируем часть изображения name с размерами [1:2048,1:2048] и создаем новое  
изображение с именем new_name уже без оверскана//
```

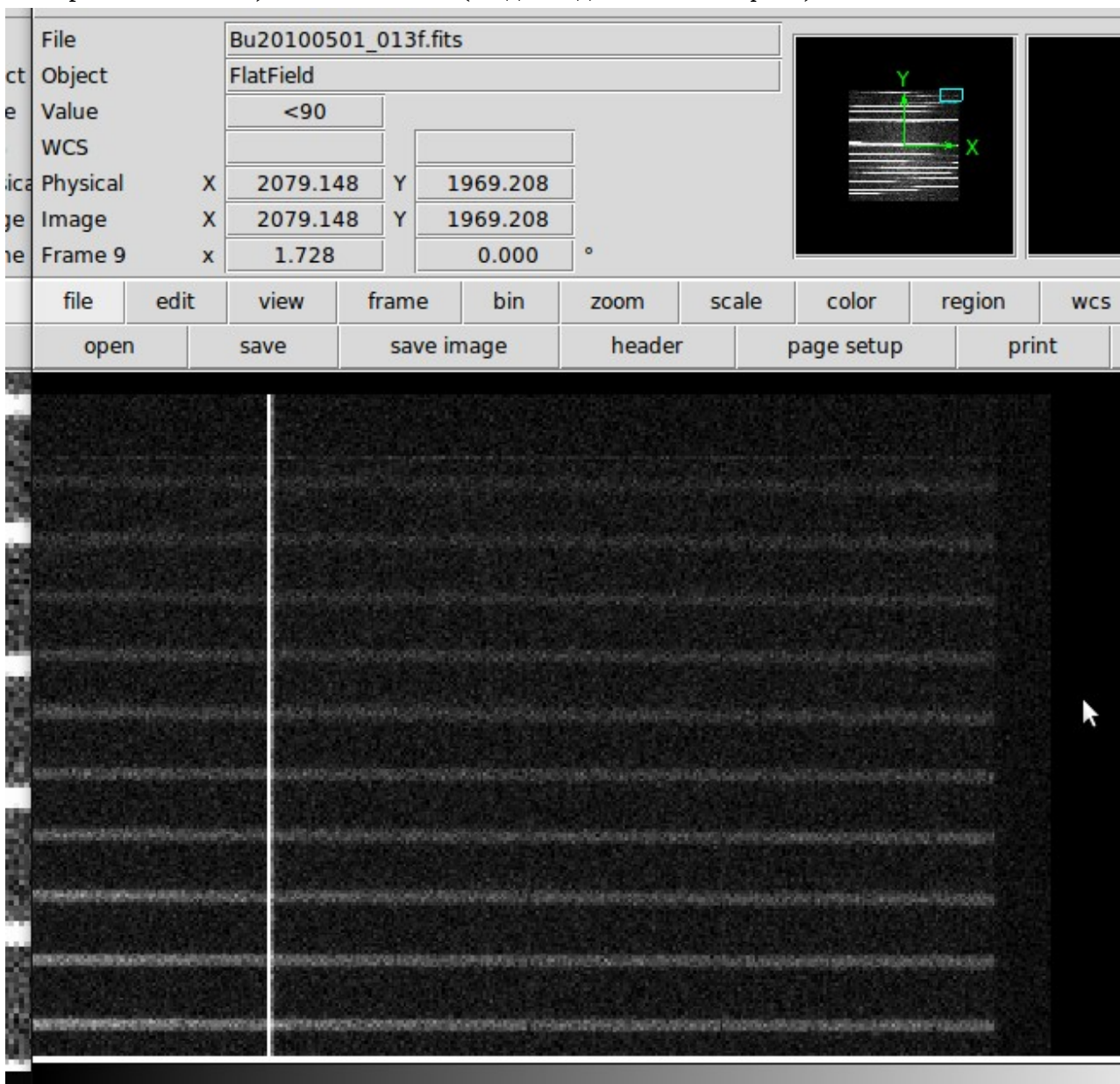
Изображение можно повернуть на указанное число градусов (здесь 180):

```
ecl> rotate name new_name 180
```

Изображение можно отразить по оси X или Y (здесь по Y):

```
ecl> imcopy name[*,-*] new_name
```

Во всех случаях name и new_name могут совпадать (тогда изображение перезаписывается) или отличаться (тогда создается новый файл).



Верхний правый кадр кадра (пока ориентированного неправильно) с областью оверскана (однородная темная полоса справа и сверху кадра). Белая полоса слева — колонка битых пикселей.

Обрабатывать большое количество изображений проблематично. Вы можете создать список входных файлов для потоковой обработки и использовать его. Для этого необходимо перейти в папку с Вашими изображениями используя команду **cd**. Для формирования списка файлов используйте команду

```
> ls *.fits > filelist.txt
```

Имена всех файлов по маске *.fits будут записаны в файл filelist.txt. Скопируйте filelist.txt в output.txt

```
> cp filelist.txt output.txt
```

и отредактируйте новый файл в любом редакторе. Можно просто поменять имя выходного файла, а можно попросить IRAF записать выходные файлы в подпапку, предварительно создав ее командой **mkdir name**

Например:

<i>file1.fits</i>	→	<i>name/sfile1.fits</i>
<i>file2.fits</i>	→	<i>name/sfile2.fits</i>
<i>file3.fits</i>	→	<i>name/sfile3.fits</i>
<i>file4.fits</i>	→	<i>name/sfile4.fits</i>

Проще можно создать папку в рабочей папке:

```
>mkdir new_folder
```

и скопировать туда файлы по маске:

```
>copy *.fits new_folder/
```

Вот как будет выглядеть команда для поворота изображений из списка filelist.txt. Изображения с новыми именами будут записаны в подпапку указанную в *output.txt*.

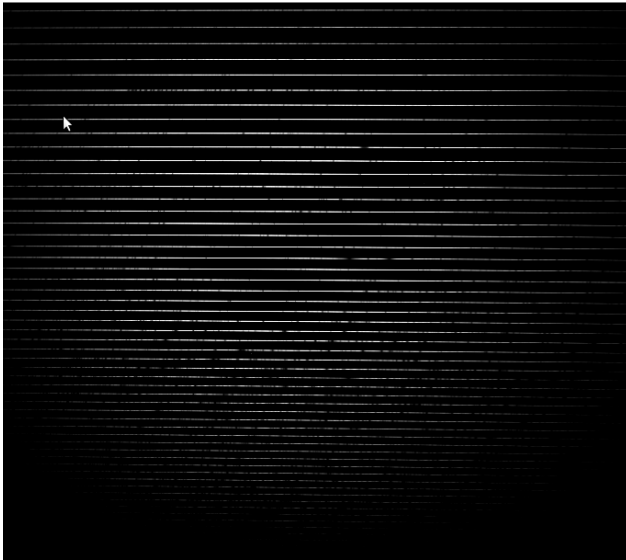
```
ecl> rotate @filelist.txt @output.txt 180
```

Команда **rotate** может перезаписывать файлы и работает по маске.

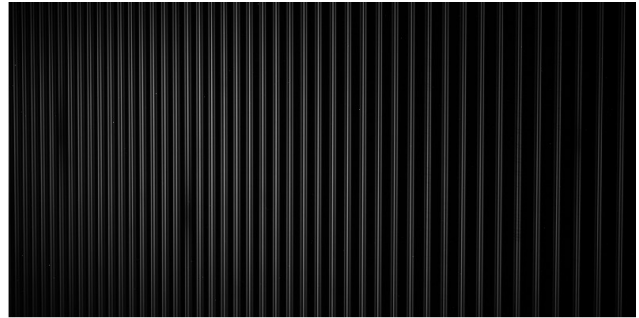
Списки файлов и маски можно использовать и с другими командами IRAF.

Апертуры на кадре могут располагаться как вертикально, так и горизонтально. Если кадр не квадратный, то поворот на 90 градусов даст только квадрат с апертурами минимального размера, кусочек с черным полем, остальные данные (по длинной стороне кадра) будут безвозвратно потеряны. Чтобы этого избежать - в *poao.imred* пишем *epar echelle*, находим параметр *disraxi*, его значение 1 - апертуры горизонтальные (вдоль строки), 2 - вертикальные (вдоль клонки).

Правильная ориентация спектра



Порядки вдоль строк



Порядки вдоль колонок

Комбинирование кадров *bias*

Проще всего скомбинировать кадры подложки (как и любые другие) используя команду **imcombine**. Параметры и список входных файлов можно задать непосредственно в командной строке IRAF, или же отредактировав параметры команды. Лучше выбрать последнее.

Вызовите список параметров:

```
ecl> epar imcombine
```

Вы увидите вот это (некоторые параметры могут иметь другие значения).

```

                                I R A F
                                Image Reduction and Analysis Facility

PACKAGE = immatch
TASK = imcombine

input =                @bias_in List of images to combine
output =                sbias List of output images
(headers=                ) List of header files (optional)
(bpmsk=                  ) List of bad pixel masks (optional)
(rejmask=                ) List of rejection masks (optional)
(nrejmas=                ) List of number rejected masks (optional)
(expmask=                ) List of exposure masks (optional)
(sigmas =                ) List of sigma images (optional)
(logfile=                STDOUT) Log file
(combine=                median) Type of combine operation
(reject =                none) Type of rejection
(project=                no) Project highest dimension of input images?
(outtype=                real) Output image pixel datatype
(outlimi=                ) Output limits (x1 x2 y1 y2 ...)
(offsets=                none) Input image offsets
(masktyp=                none) Mask type
(maskval=                0.) Mask value
(blank =                 0.) Value if there are no pixels
(scale =                 none) Image scaling
(zero =                  none) Image zero point offset
(weight =                none) Image weights

```

```

(statsec=          ) Image section for computing statistics
(expname=         ) Image header exposure time keyword
(lthresh=        INDEF) Lower threshold
(hthresh=        INDEF) Upper threshold
(nlow =          1) minmax: Number of low pixels to reject
(nhigh =         1) minmax: Number of high pixels to reject
(nkeep =         1) Minimum to keep (pos) or maximum to reject (neg)
(mclip =         yes) Use median in sigma clipping algorithms?
(lsigma =        3.) Lower sigma clipping factor
(hsigma =        3.) Upper sigma clipping factor
(rdnoise=        0.) ccdclip: CCD readout noise (electrons)
(gain =          1.) ccdclip: CCD gain (electrons/DN)
(snoise =        0.) ccdclip: Sensitivity noise (fraction)
(sigscal=       0.1) Tolerance for sigma clipping scaling correction
(pclip =        -0.5) pclip: Percentile clipping parameter
(grow =          0.) Radius (pixels) for neighbor rejection
(mode =         ql)

```

Первые две строки – имена входного и выходного файлов. В качестве входного используется список *@bias_in*. Для формирования кадра *bias* нужно использовать медианное усреднение, для этого параметр *combine* должен быть равен *median*. Но можно получить и сумму *combine=sum* или среднее *combine=average*. Медианное усреднение позволяет достаточно хорошо избавиться от следов космических частиц. Если кадров подложки, участвующих в усреднении достаточно много (>3...5), то использовать остальные параметры сложения, позволяющие избавиться от «мусора» на изображении, не обязательно и можно их значения оставить без изменений.

Для выхода из редактора параметров нажмите **ctrl+c** если не желаете сохранять изменения, или **ctrl+d** если желаете сохранить правки.

Осталось только запустить команду

```
ecl> imcombine @list sbias
```

которая сформирует изображение super-bias (*sbias*) из файлов, входящих в список *list*. Также можно выполнить комбинирование по маске; параметры можно задавать непосредственно после команды, не меняя параметры:

```
>imcombine bias*.fits sbias combine=median
```

Значения остальных параметров не изменятся.

Вычитание bias из изображений объекта, Th-Ar лампы и лампы плоского поля

Опять же формируем список изображений для обработки командой

```
> ls *.fits > filelist.txt
```

(Самого кадра *sbias* в списке не должно быть!)

в терминале Linux (находясь в папке с изображениями), редактируем его и создаем список *clear* с именами выходных файлов.

Командой

```
ecl> imarith @filelist - sbias @clear
```

вычитаем *sbias* из изображений перечисленных в файле *filelist* и сохраняем их под именами из списка *clear*.

Комбинирование кадров flat-field и ThAr лампы

Осуществляется так же, как и комбинирование кадров *bias*. Можно использовать список или же перечислить изображения непосредственно в параметрах команды:

```
ecl> imcombine flat0,flat1,flat2 sflat combine=median  
ecl> imcombine thar* ThAr combine=average
```

Так мы сформировали медианное среднее *sflat* из изображений *flat0,flat1,flat2* и среднее изображение лампы спектра сравнения.

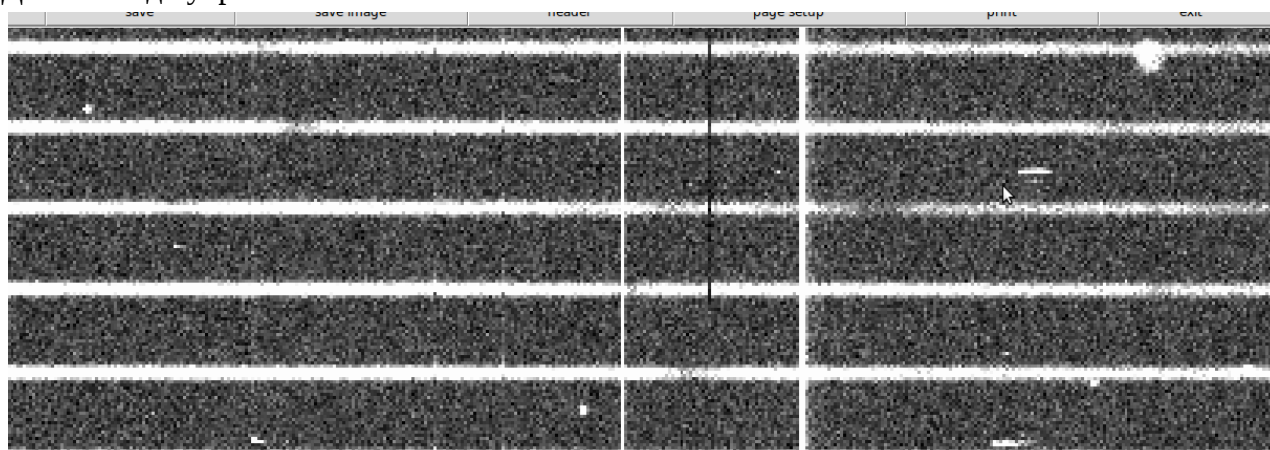
Теперь мы имеем следующий набор необходимых для дальнейшей работы изображений:

- 1) Кадры со спектром объекта с вычтенным *super-bias*;
- 2) Один кадр со спектром калибровочной лампы, полученный усреднением нескольких, с вычтенным *super-bias*;
- 3) Один кадр плоского поля, полученный методом медианного усреднения из нескольких отдельных кадров, с вычтенным *super-bias*.

Подготовка изображения к экстракции спектра

Первый этап подготовки уже пройден — мы вычли подложку.

Дальше надо убрать плохи пиксели и космики.



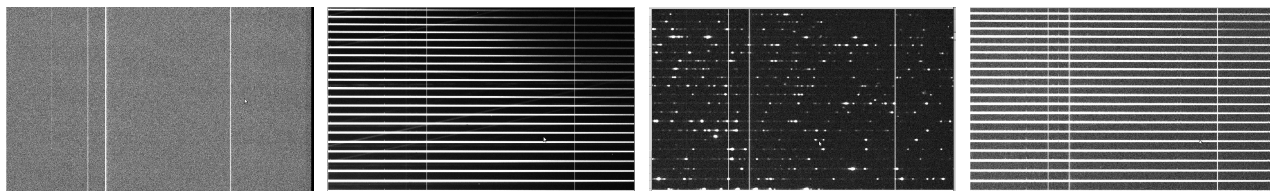
Это фрагмент спектра объекта. Белые и темные колонки — это колонки плохих пикселей. Яркие точки разной формы — это следы космических частиц, космики. Число космиков заметно растет с увеличением экспозиции (у этого кадра 1200 сек — долгая экспозиция, космиков много). Есть еще отдельные битые пиксели. Их можно отличить от космиков, посмотрев на калибровочные кадры, битые пиксели есть на всех кадрах.

Убираем плохие пиксели

Важно уметь избавляться от плохих пикселей, иногда это целые полосы через весь кадр. Для этого в пакете proto есть команда fixpix. Да! На всякий случай делайте бэкапы, поскольку команда перезаписывает файл и если что не так, прошлого не воротить. Итак, смотрим наш файл, находим плохие пиксели и записываем их координаты в отдельный файл badpix по такому принципу: xbegin xend ybegin yend, то есть если есть колонка плохих пикселей по x 380 через весь кадр и плохой пиксель с координатами (265,456), файл будет выглядеть так:

```
380 380 1 2070
265 456
```

Посмотреть координаты плохих пикселей можно в ds9. Битые колонки видно на всех кадрах. fixpix @pixlist badpix запускаем команду и на выходе получаем чистенький файл.



Байес

Флэт

Спектр сравнения

Звезда (Вега)

Избавляемся от космиков

Чтобы избавиться от космиков есть очень хороший скрипт L.A.Cosmic (Laplacian Cosmic Ray Identification By Pieter G. van Dokkum (Yale)). Вот здесь всё про него написано и код тоже можно скачать <http://www.astro.yale.edu/dokkum/lacosmic/> Для спектров и фотометрии версии разные. Также они разные для IRAF и IDL, скачайте нужную. Как запустить скрипт там в примере <http://www.astro.yale.edu/dokkum/lacosmic/notes.html> подробно описано, но я всё же

повторю алгоритм здесь.

Берём отсюда http://www.astro.yale.edu/dokkum/lacosmic/lacos_spec.cl код, копируем его в gedit и сохраняем как lacos_im.cl в рабочую папку. Запускаем IRAF, заходим в пакет stsdas (в него можно попасть из любого пакета), если его нет - установите.

Пишем task lacos_im = path/lacos_im.cl где path/ - путь к папке, в которой лежит скрипт.

erac lacos_im редактируем параметры, жмём CTRL-D, чтобы сохранить изменения.

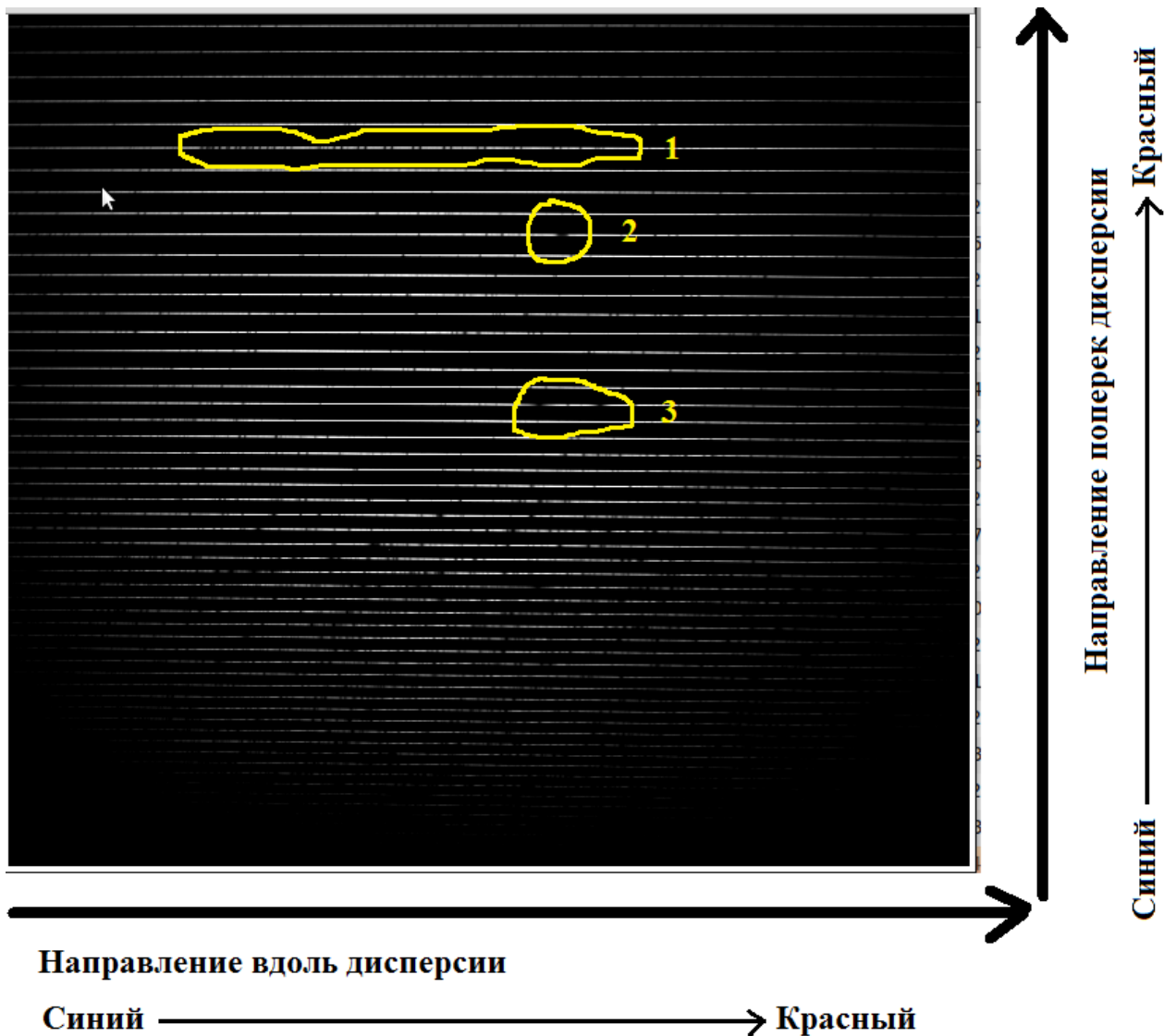
lacos_im input output outmask запускаем скрипт. Периодически отвечаем на промежуточные вопросы и всё, изображение чистое. Все этот скрипт очень хвалят, видимо он и правда очень хороший.

Обработка эшелле-спектров

Проводится в несколько этапов:

- 1) Определение оптимальной апертуры;
- 2) Определение следа апертуры для экстракции порядков спектра;
- 3) получаем нормализованное плоское поле и делим спектры на него;
- 4) избавляемся от рассеянного света;
- 5) проводим экстракцию спектра;
- 6) идентифицируем линии в спектре калибровочной лампы и выполняем калибровку по длинам волн.

До сих пор мы работали с эшелле-кадрами как с любыми другими, не учитывая того, что на них изображены спектры. При дальнейшей работе с кадрами это надо учитывать.



Эшелле спектр звезды Альдебаран, полученный на спектрографе UFES. 1 — атмосферные линии воды, 2 — звездная линия водорода Н-альфа, 3 — дублет звездного натрия.

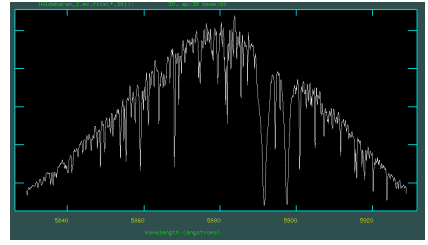
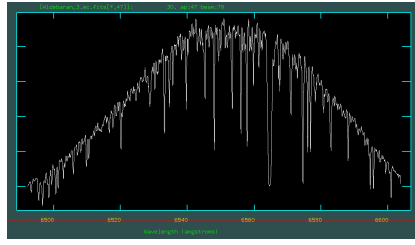
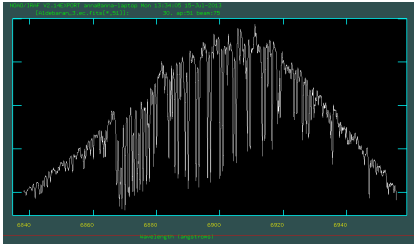
На рисунке приведен спектр звезды позднего спектрального класса (это нетрудно понять по множеству линий поглощения в спектре). На рисунке отмечены характерные линии, которые могут помочь в работе с кадром.

1. Атмосферные линии ($\lambda \approx 6860\text{--}6940 \text{ \AA}$). Эти линии есть на всех спектрах небесных объектов и выглядят всегда одинаково, тем они и удобны (атмосфера-то у Земли одна). Линий много, но выделенная на рисунке серия наиболее характерная. По ней можно ориентироваться для правильного поворота спектра. "Пунктир" – синий, "штрих-пунктир" – красный, так выглядит эта серия на кадре.

2. Главная линия водорода – первая линия серии Бальмера – Н-альфа ($\lambda \approx 6563 \text{ \AA}$). Эта линия наблюдается в спектрах большинства звезд, имеет разную форму в зависимости от спектрального класса и класса светимости. Чаще всего это линия поглощения, иногда – излучения, иногда и то, и другое сразу. У спектрально-двойных звезд она (как и другие звездные линии) раздваивается.

3. Дублет натрия ($\lambda \approx 5890 \text{ \AA}$) – сильные линии в желтой части спектра.

Так выглядят спектры порядков с указанными линиями



Атмосферные линии ($\lambda \approx 6860$ - 6940 \AA) Н-альфа ($\lambda \approx 6563 \text{ \AA}$)

Дублет натрия ($\lambda \approx 5890 \text{ \AA}$)

Убираем рассеянный свет

После того, как мы исключили байес, плохие пиксели и космики, убираем рассеянный свет. Для этого в `noao.imred.echelle` есть команда `apscatter`. Там куча разных параметров, которые можно менять, в том числе число итераций и порядок функций приближения. Есть ещё параметры `apscat1` и `apscat2`, у них тоже есть свои параметры, там-то и надо установить число итераций и порядок функций. Большую часть параметров можно оставить по умолчанию кроме того самого числа итераций (по умолчанию 0) и порядка функций.

`>apscatter input output # убираем рассеянный свет.`

I R A F

Image Reduction and Analysis Facility

PACKAGE = echelle

TASK = apscatter

```
input = polaris020.fits List of input images to subtract scattered light # входной файл
output = polaris_20 List of output corrected images # выходной файл
(apertur= 60) Apertures # число апертур (порядков) на кадре
(scatter= ) List of scattered light images (optional)
(referen= flat) List of aperture reference images # опорное изображение, для которого в
database есть файл с записанными позициями (в пикселях) апертур

(interac= yes) Run task interactively? # команда может выполняться автоматически с
заданными параметрами (no) или в интерактивном режиме, параметры можно менять в ходе
работы и контролировать результат
(find = yes) Find apertures? # автоматический поиск апертур
(recente= yes) Recenter apertures? # автоматическое/ручное центрирование апертур
(resize = yes) Resize apertures? # автоматическое/ручное изменение размера апертур
(edit = yes) Edit apertures? # автоматическое/ручное изменение (любое) апертур
(trace = yes) Trace apertures? # автоматическая/ручная трассировка апертур
(fittrac= yes) Fit the traced points interactively? # возможность в интерактивном
режиме искать лучшую функцию для трассировки апертуры
(subtrac= yes) Subtract scattered light? # вычитание рассеянного света
(smooth = yes) Smooth scattered light along the dispersion? # сглаживание
поверхности рассеянного света вдоль дисперсии
(fitscat= yes) Fit scattered light interactively? # возможность в интерактивном
режиме искать лучшую функцию для рассеянного света

(fitsmoo= yes) Smooth the scattered light interactively? # возможность в
интерактивном режиме сглаживать функцию для рассеянного света

(line = INDEF) Dispersion line
(nsum = 10) Number of dispersion lines to sum or median
```

(buffer = 1.) Buffer distance from apertures
 (apscat1= apscat1) Fitting parameters across the dispersion # параметры функции поперек дисперсии
 (apscat2= apscat2) Fitting parameters along the dispersion # параметры функции вдоль дисперсии
 (mode = ql)

I R A F
 Image Reduction and Analysis Facility

PACKAGE = echelle

TASK = apscat1

(functio= spline3) Fitting function # приближающая функция
 (order = 3) Order of fitting function # порядок приближающей функции
 (sample = *) Sample points to use in fit # интервал точек, используемых для построения функции
 (naverag= 1) Number of points in sample averaging # число точек внутри интервала, по которым проводят усреднение
 (low_rej= 5.) Low rejection in sigma of fit # нижний порог выбросов в ско
 (high_re= 2.) High rejection in sigma of fit # верхний порог выбросов в ско
 (niterat= 5) Number of rejection iterations # число итераций для устранения выбросов
 (grow = 1.) Rejection growing radius in pixels # радиус области вокруг выброса, внутри которой все точки также считают выбросами, в пикселях
 (mode = ql)

Для списка параметров apscat2 все то же самое.

3.1 Определение оптимальной апертуры

Теперь, когда всё, кроме плоского поля, исключено, можно приступить к экстракции спектра. Для этого в том же пакете echelle есть команда apall. Экстракцию лучше проводить в интерактивном режиме; сначала корректируем апертуры для всех порядков, затем выполняем трассировку и, наконец, экстракцию. Если что-то не получается, всегда можно вернуться и всё переделать.

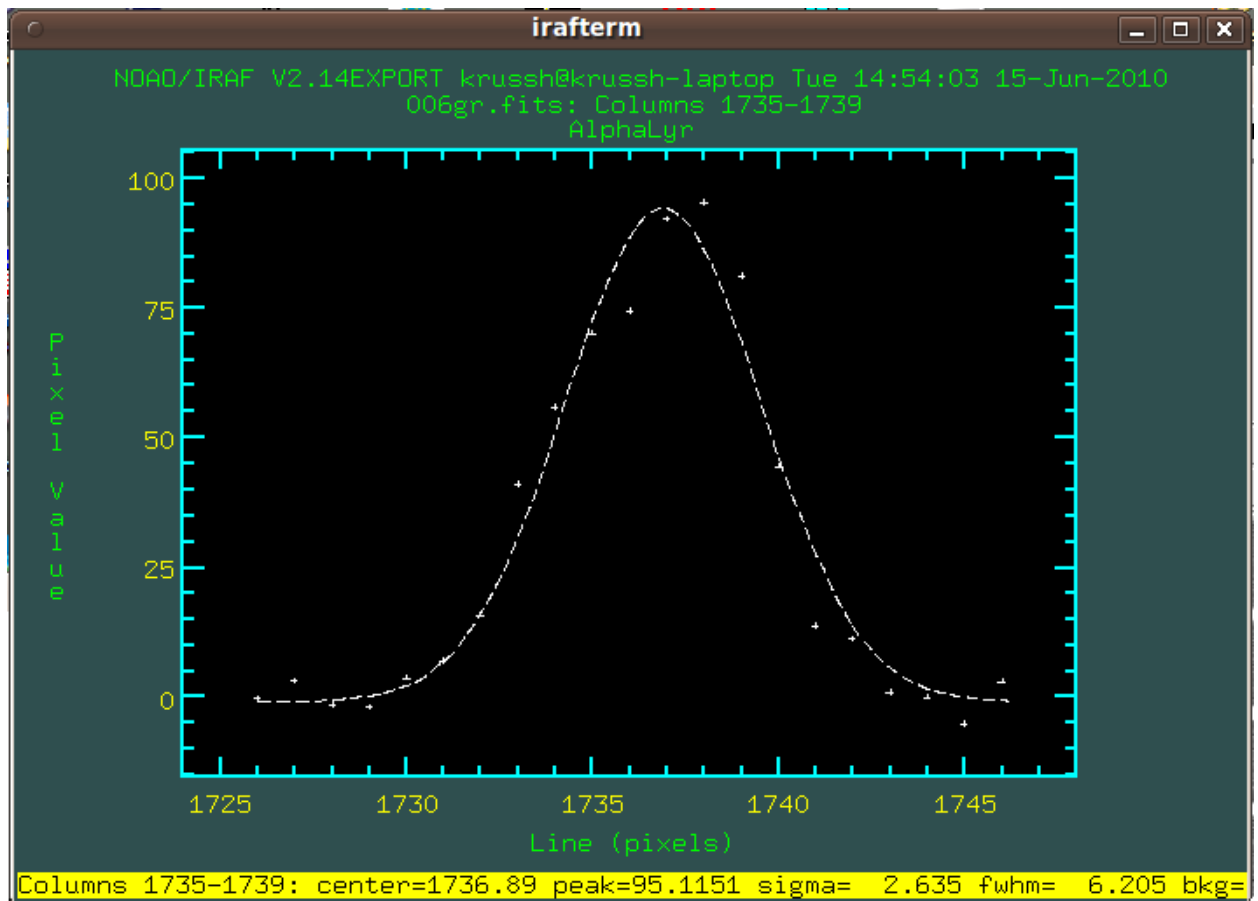
Перед началом работы стоит посмотреть на спектры и определить оптимальную ширину апертуры. Запустите DS9:

ecl> !ds9&

Для анализа используйте команду:

ecl> imexam filename

Наведите указатель в активном окне DS9 на один из порядков спектра и нажмите клавишу **k**, в открывшемся окне irafterm Вы увидите профиль порядка и оценку FWHM.



Для оптимальной экстракции порядка можно использовать апертуру с шириной около 2 FWHM. Параметр FWHM может меняться по кадру, запомните или запишите его значение для разных порядков и скорректируйте его значение позже, при интерактивной работе в **apall**. Для выхода из **imexam** нажмите **q** в активном окне DS9.

3.2 Определение следа апертуры

Определение следа апертуры для экстракции порядков спектра производится в интерактивном режиме команды **apall**.

Для начала следует переместиться в пакет **noao/imred/echelle** и проверить параметры задания *echelle*

```
echelle> epar echelle
```

необходимо установить параметр **dispaxis=1** (для порядков примерно параллельных строкам). Не забываем, что мы развернули все кадры именно таким образом.

В пакет *echelle* входит множество заданий (task) со своим набором параметров. Но мы воспользуемся пактом *apall*, это процедура включает в себя несколько отдельных процедур и позволяет повести их редактируя все параметры одновременно.

Как всегда начнем с редактирования параметров:

```
echelle> epar apall
```

I R A F
Image Reduction and Analysis Facility

```
PACKAGE = echelle  
TASK = apall
```

```
input = H4998b List of input images
```

```

(output =                H4998b_ec) List of output spectra
(apertur=                ) Apertures
(format =                echelle) Extracted spectra format
(referen=                ) List of aperture reference images
(profile=                ) List of aperture profile images
(interac=                yes) Run task interactively?
(find =                  yes) Find apertures?
(recente=                yes) Recenter apertures?
(resize =                yes) Resize apertures?
(edit =                  yes) Edit apertures?
(trace =                 yes) Trace apertures?
(fittrac=                yes) Fit the traced points interactively?
(extract=                yes) Extract spectra?
(extras =                no) Extract sky, sigma, etc.?
(review =                yes) Review extractions?
(line =                  INDEF) Dispersion line
(nsum =                  20) Number of dispersion lines to sum or median

# DEFAULT APERTURE PARAMETERS

(lower =                  -5.) Lower aperture limit relative to center
(upper =                  5.) Upper aperture limit relative to center
(apidtab=                ) Aperture ID table (optional)

# DEFAULT BACKGROUND PARAMETERS

(b_func=                 chebyshev) Background function
(b_order=                1) Background function order
(b_sampl=                -10:-6,6:10) Background sample regions
(b_naver=                -3) Background average or median
(b_niter=                0) Background rejection iterations
(b_low_r=                3.) Background lower rejection sigma
(b_high_=                3.) Background upper rejection sigma
(b_grow =                0.) Background rejection growing radius

# APERTURE CENTERING PARAMETERS

(width =                  10.) Profile centering width
(radius =                 10.) Profile centering radius
(thresho=                0.) Detection threshold for profile centering

# AUTOMATIC FINDING AND ORDERING PARAMETERS

nfind =                   59 Number of apertures to be found automatically
(minsep =                 10.) Minimum separation between spectra
(maxsep =                 100.) Maximum separation between spectra
(order =                  decreasing) Order of apertures

# RECENTERING PARAMETERS

(aprecen=                 ) Apertures for recentering calculation
(npicks =                 INDEF) Select brightest peaks
(shift =                  yes) Use average shift instead of recentering?

# RESIZING PARAMETERS

(llimit =                 -10.) Lower aperture limit relative to center
(ulimit =                 10.) Upper aperture limit relative to center
(ylevel =                 0.1) Fraction of peak or intensity for automatic wid
(peak =                   yes) Is ylevel a fraction of the peak?
(bkg =                    no) Subtract background in automatic width?
(r_grow =                 0.) Grow limits by this factor
(avglimi=                 yes) Average limits over all apertures?

# TRACING PARAMETERS

(t_nsum =                 10) Number of dispersion lines to sum
(t_step =                 10) Tracing step
(t_nlost=                 3) Number of consecutive times profile is lost bef
(t_func=                 legendre) Trace fitting function
(t_order=                 3) Trace fitting function order
(t_sampl=                 *) Trace sample regions

```

```

(t_naver=          1) Trace average or median
(t_niter=          0) Trace rejection iterations
(t_low_r=         3.) Trace lower rejection sigma
(t_high_=         3.) Trace upper rejection sigma
(t_grow =         0.) Trace rejection growing radius

# EXTRACTION PARAMETERS

(backgro=         none) Background to subtract
(skybox =         1) Box car smoothing length for sky
(weights=         none) Extraction weights (none|variance)
(pfit =          fit1d) Profile fitting type (fit1d|fit2d)
(clean =         no) Detect and replace bad pixels?
(saturat=        INDEF) Saturation level
(readnoi=        8) Read out noise sigma (photons)
(gain =          1.) Photon gain (photons/data number)
(lsigma =        4.) Lower rejection threshold
(usigma =        4.) Upper rejection threshold
(nsubaps=        1) Number of subapertures per aperture
(mode =         ql)

```

Выше представлены параметры для работы со спектрами, полученными на UFES в диапазоне 4000-8000Å. Особое внимание следует обратить на **выделенное полужирным шрифтом**. Подробнее о каждом из параметров можно прочитать здесь: <http://iraf.noao.edu/scripts/irafhelp?apall>.

Запускаем процедуру **apall** в интерактивном режиме:

```
echelle> apall
```

отвечаем на несколько простых вопросов. Обычно просто можно нажать enter или ввести новое значение параметров. Обратите внимание на число апертур, которое iraf ищет автоматически. Лучше указать верное число (для UFES это 60). После открывается окно irafterm в котором можно корректировать положение апертур.

Важно знать, что в интерактивном режиме можно менять параметры процедуры не прекращая её работы. Для этого достаточно набрать в активном окне irafterm :<параметр> <значение>, например:

:function legendre

здесь также необязательно полностью писать имя параметра и его значение, достаточно однозначно определяющая их комбинация первых символов, например:

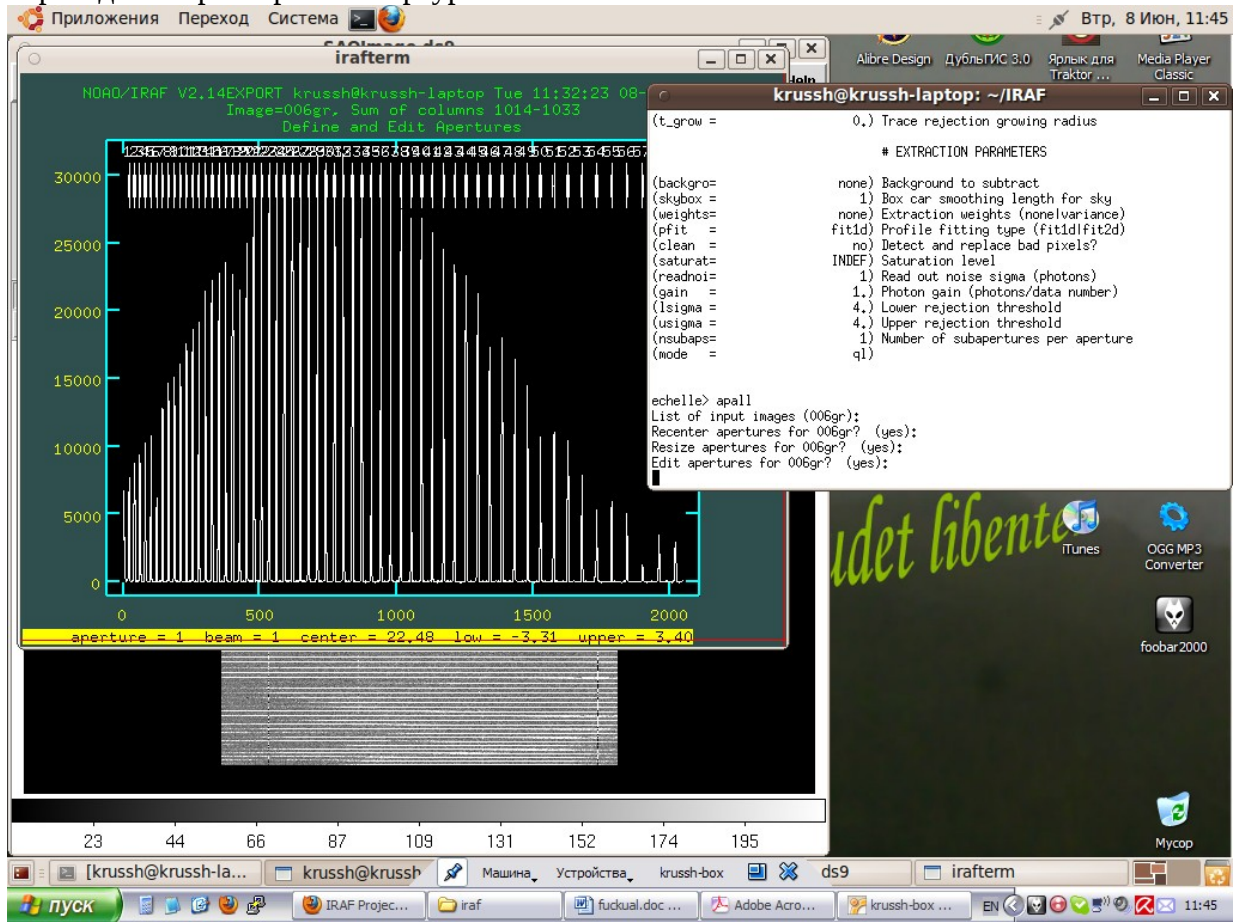
:func legen

Не забывайте про двоеточие!

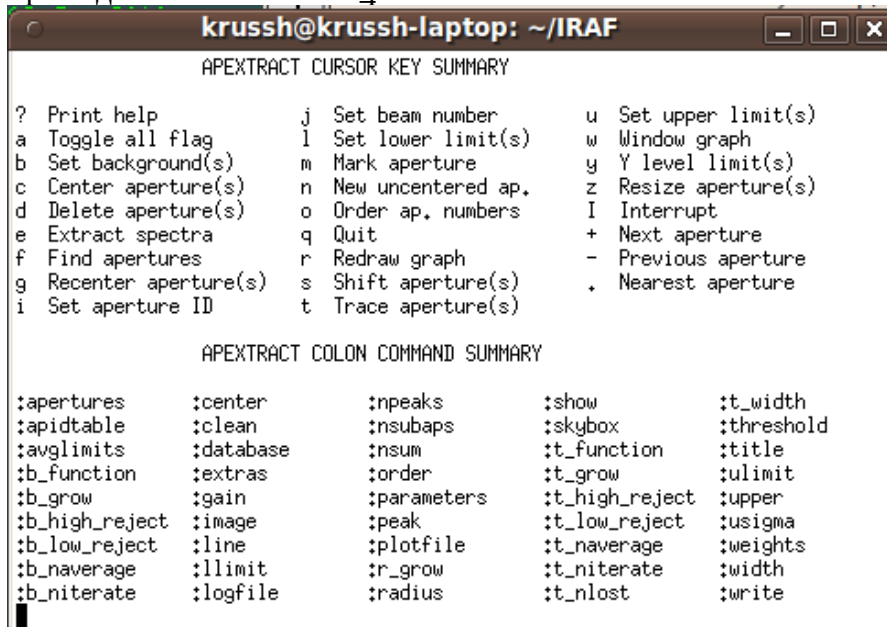
В общем-то у спектров с хорошим уровнем сигнала apall автоматически подбирает оптимальную апертуру (ширину и след). Если спектр очень слабый, лучше его экстрагировать по опорному спектру (обычно это ближайший по времени флэт).

Сначала apall показывает разрез спектра поперек дисперсии, примерно посередине кадра (номер колонки указан в параметрах apall) и устанавливает ширины апертур, их можно менять. Важно проверить, чтобы все порядки были отмечены, а всё лишнее — не отмечено. Масштаб окна можно менять (порядков так много, что в маленьком окне они сливаются). Для этого жмем **w** и затем клавишей **e** отмечаем левый нижний и правый верхний углы области окна, которую нужно увеличить. Вернуть прежний масштаб можно нажав **w** и **a**. Особое внимание обратите на крайние (синие и красные) порядки. Во-первых, в зависимости от спектрального класса объекта сигнал в синей или красной части может быть слабым. Во-вторых, синие порядки расположены очень близко друг к другу, а в красной части между порядками возникают интерференционные полосы, которые часто тоже распознаются как порядки. Лишние апертуры можно удалить клавишей **d**, добавить апертуру можно клавишей **m**, также можно менять номера апертур, их ширину, центр и т. д., подсказка открывается в основном окне IRAF по нажатию **?**. После того, как все порядки нашли и отметили, жмем **q** и

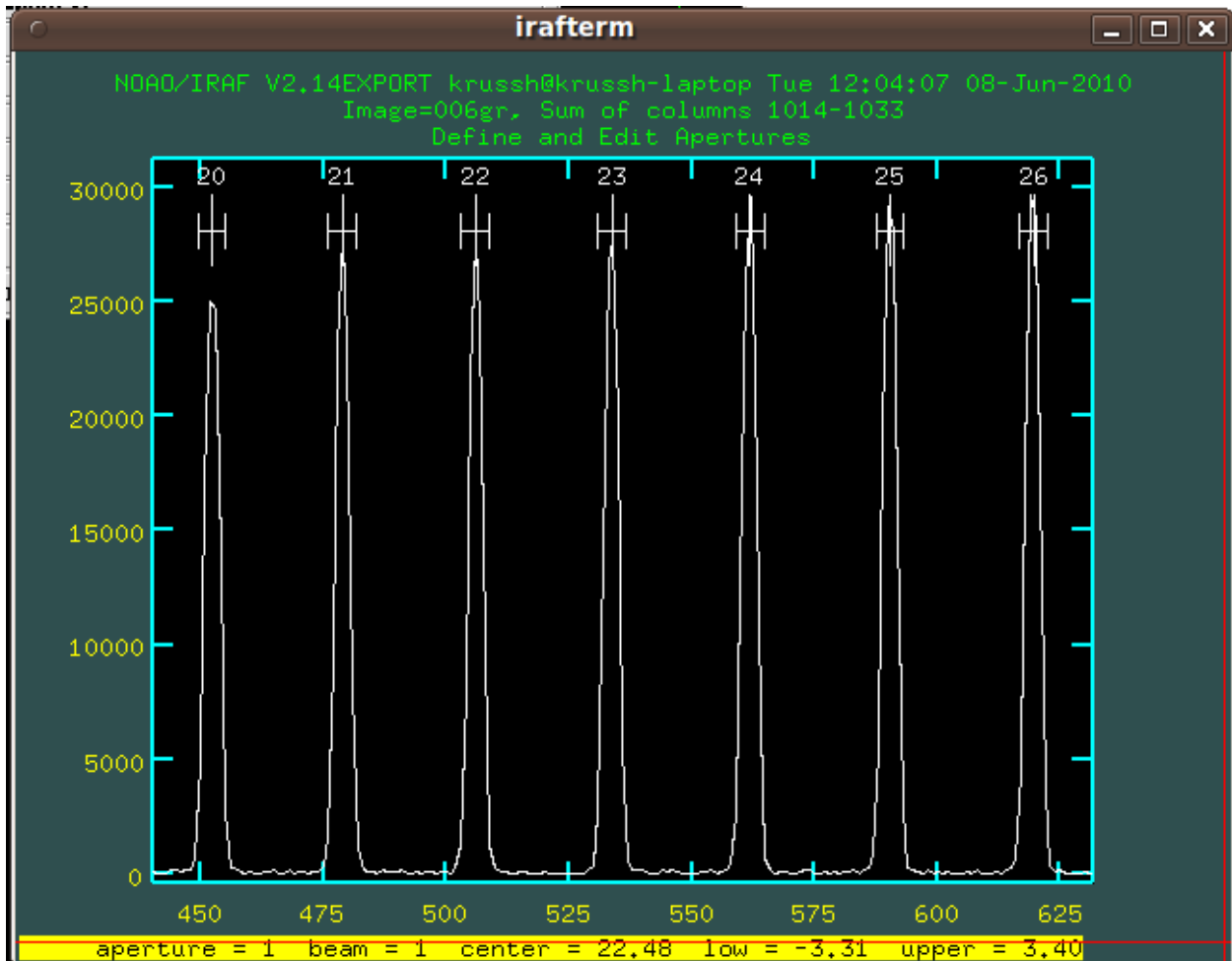
переходим к трассировке апертур.



Вы можете вызвать справку по клавишам управления, нажав shift+?. Выход из просмотра подсказок - клавиша q.



Для управления просмотром необходимо перейти в режим окна – сделайте irafterm активным, щелкнув на нем мышкой, и нажмите w. Выберите левый нижний угол участка для просмотра и нажмите e, выберите правый верхний угол и опять нажмите клавишу e. Участок изображения будет увеличен. Для возвращения обратно наберите w и a.



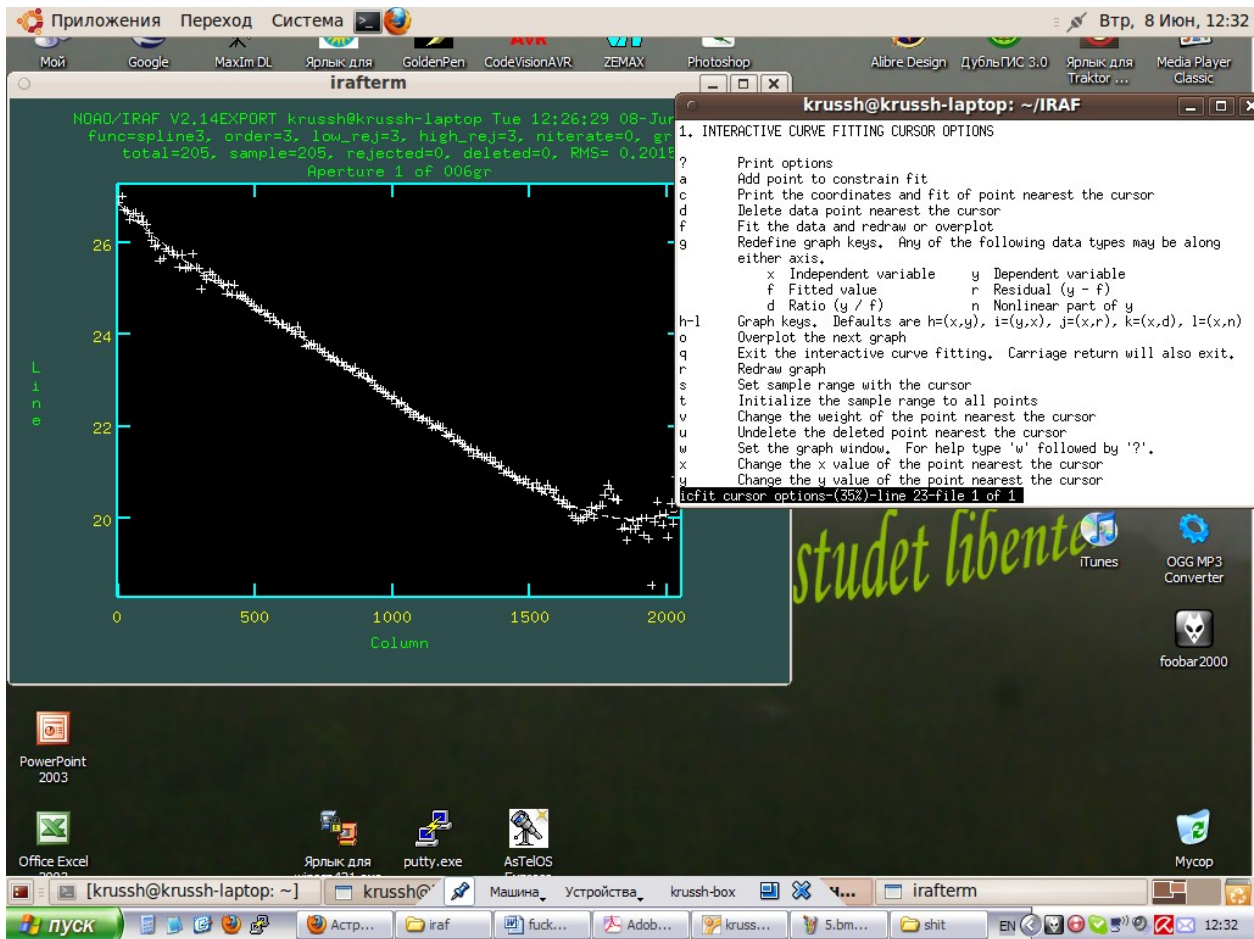
Мы можем определить центр аперттуры, используя курсор (навести его на центр, и нажать c), или использовать его как стартовую позицию для центрирования с помощью IRAF (навестись на центр и нажать m). Можно использовать кнопку d, чтобы удаль предыдущие значения аперттуры (они будут нумероваться 1,2,3 и т.д.). Ширина аперттуры может быть определена с помощью кнопки w. Она задает ширину аперттуры, равной расстоянию от центра до местоположения курсора (поэкспериментируйте, чтобы понять, как это работает). Наоборот, вы можете задать верхнюю и нижнюю границы аперттуры, используя кнопки u и l (для верхней и нижней соответственно).

Закончив центрирование всех аперттур нажмите клавишу q для перехода к следующему шагу – трассировке порядков. После того, как Вы ответите на несколько простых вопросов, в окне irafterm появится изображение.

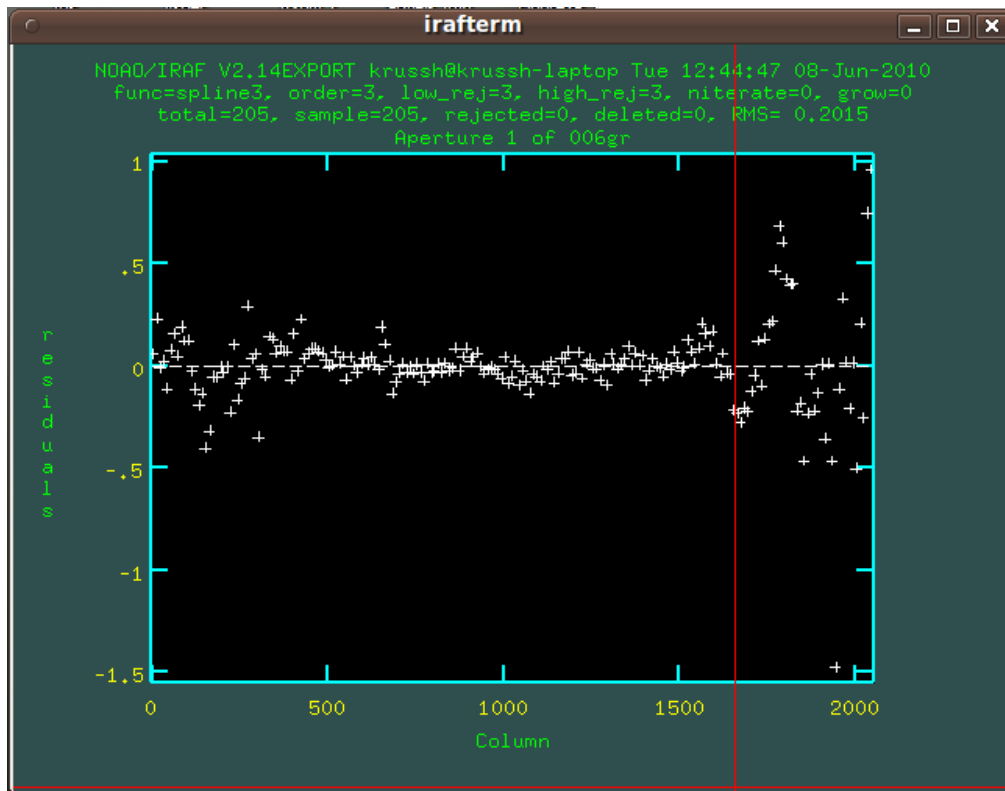
На первый взгляд кажется, что порядки расположены на кадре почти горизонтально по прямой, но это не так. Дисперсионная кривая — это обычно полином высокого порядка, например, для UFES порядок кривой 6 — поперек дисперсии и 8 — вдоль дисперсии. На этапе трассировки точно подбирать полином не требуется. Надо только подобрать кривую, повторяющую форму порядка на кадре, обычно достаточно 2-3 степени кубического сплайна. Так называемый след порядка — X и Y координаты пикселей, содержащих порядок спектра, на кадре. На этом этапе можно менять отображение следа порядка и приближающей функции: собственно след порядка, отношение следа к функции и разность следа и функции.

Здесь точки с данными (+) представляют след спектра, штриховая линия показывает приближение. Разница выглядит огромной, пока вы не заметите, что отклонение по оси Y от штриховой линии составляет около 3 пикселей. Чтобы изменить порядок приближающего полинома используйте команду **:order** (не забудьте двоеточие). После **:order** введите нужный порядок полинома. Чтобы обновить картинку используйте **f**. Также можно поменять тип функции, возможные опции – *spline1*, *spline3*, *legendre*, *chebyshev* – линейный и кубический сплайны, полиномы Лежандра и полиномы Чебышева.

Как и раньше Вы можете вызвать справку по клавишам управления, нажав shift+?.



Нажав клавишу `j` можно перейти в режим просмотра разницы между приближением (штриховая линия) и следом спектра (+).



Если Вы вполне удовлетворены результатом, нажмите `q` и переходите к следующему

порядку спектра.

Все апертуры, которые мы только что определили, сохраняются в файл с именем *database/filename* в текущей папке. Так что вы можете вернуться к этой картинке позже и продолжить определение апертуры.

Для просмотра полученного спектра используйте команду

```
ecl> implot filename
```

Важные клавиши для этого окна:

- 1) l – профиль строки
- 2) c – профиль колонки
- 3) j, k – переход по строкам вверх и вниз
- 4) space – координаты и отсчет в пикселе

Подробности можно прочитать здесь: <http://iraf.noao.edu/scripts/irafhelp?implot>.

Альтернативный вариант просмотра и анализа:

```
ecl> splot filename
```

Нажав клавишу w, вы перейдете в режим окна и сможете увеличить часть спектра, наводясь в левый нижний угол и нажимая e и, соответственно в верхний правый и опять же e. Для возврата к неувеличенному изображению используйте клавишу c. Для перехода к другому порядку спектра нажмите shift+# и введите номер порядка который хотите увидеть. Команда **splot** имеет много возможностей для отображения и анализа спектра, более подробно смотрите здесь: <http://iraf.noao.edu/scripts/irafhelp?splot>.

После определения апертур и порядков файл, для которого была сделана эта работа, используется в качестве опорного (**referen** в параметрах **apall**). В этом случае экстракция спектров будет проходить одинаково для всех кадров.

Из полученного output файла с помощью **apall** экстрагируем спектр. В интерактивном режим более нет необходимости, поэтому пишем

```
apall obj.fits ref=ref.fits interac=no find=no edit=no trace=no fittrace=no
```

Далее надо учесть плоское поле и откалибровать по длинам волн.

3.3 Учет плоского поля

Производится с помощью команды **apflatten**. Для начала, как всегда, редактируем параметры:

```
echelle> epar apflatten
```

```

                                     IRAF
                               Image Reduction and Analysis Facility

PACKAGE = echelle
TASK = apflatten

input =                               flat List of images to flatten
output =                              flatn List of output flatten images
(apertur=                             ) Apertures
(referen= H4998b) List of reference images
(interac= yes) Run task interactively?
(find = no) Find apertures?
(recente= yes) Recenter apertures?
(resize = yes) Resize apertures?
(edit = yes) Edit apertures?
(trace = no) Trace apertures?
(fittrac= no) Fit traced points interactively?
(flatten= yes) Flatten spectra?
```

```

(fitspec=          yes) Fit normalization spectra interactively?
(line =           INDEF) Dispersion line
(nsum =          10) Number of dispersion lines to sum or median
(thresho=       10.) Threshold for flattening spectra
(pfit =         fit1d) Profile fitting type (fit1d|fit2d)
(clean =         no) Detect and replace bad pixels?
(saturat=       INDEF) Saturation level
(readnoi=       0.) Read out noise sigma (photons)
(gain =         1.) Photon gain (photons/data number)
(lsigma =       4.) Lower rejection threshold
(usigma =       4.) Upper rejection threshold
(funcnio=       spline3) Fitting function for normalization spectra
(order =        19) Fitting function order
(sample =       *) Sample regions
(naverag=       1) Average or median
(niterat=       5) Number of rejection iterations
(low_rej=       3.) Lower rejection sigma
(high_re=       3.) High upper rejection sigma
(grow =         0.) Rejection growing radius
(mode =         ql)

```

Параметры аналогичны параметрам задания **appall**, нам необходимо получить спектр плоского поля по тому же пути, что и для спектра объекта. Поэтому параметр **referen** устанавливаем равным имени файла для которого выполнялась процедура **appall**, а параметры **find, trace, fittrac = no**.

Дополнительно следует изменить параметры **ylevel** и **peak** в задании **apresize**:

I R A F Image Reduction and Analysis Facility

```

PACKAGE = echelle
      TASK = apresize
input =          List of input images
(apertur=       ) Apertures
(referen=       ) Reference images
(interac=       no) Run task interactively?
(find =         yes) Find apertures?
(recente=       no) Recenter apertures?
(resize =       yes) Resize apertures?
(edit =         yes) Edit apertures?
(line =         INDEF) Dispersion line
(nsum =         1) Number of dispersion lines to sum or median
(llimit =      -20.) Lower aperture limit relative to center
(ulimit =       20.) Upper aperture limit relative to center
(ylevel =      0.4) Fraction of peak or intensity for automatic wi
(peak =         yes) Is ylevel a fraction of the peak?
(bkg =         no) Subtract background in automatic width?
(r_grow =       0.) Grow limits by this factor
(avglimi=      yes) Average limits over all apertures?
(mode =         ql)

```

Запустите **apflatten**

```
echelle> apflatten
```

если необходимо, введите имя входного и выходного файла и приступайте к корректировке апертур.

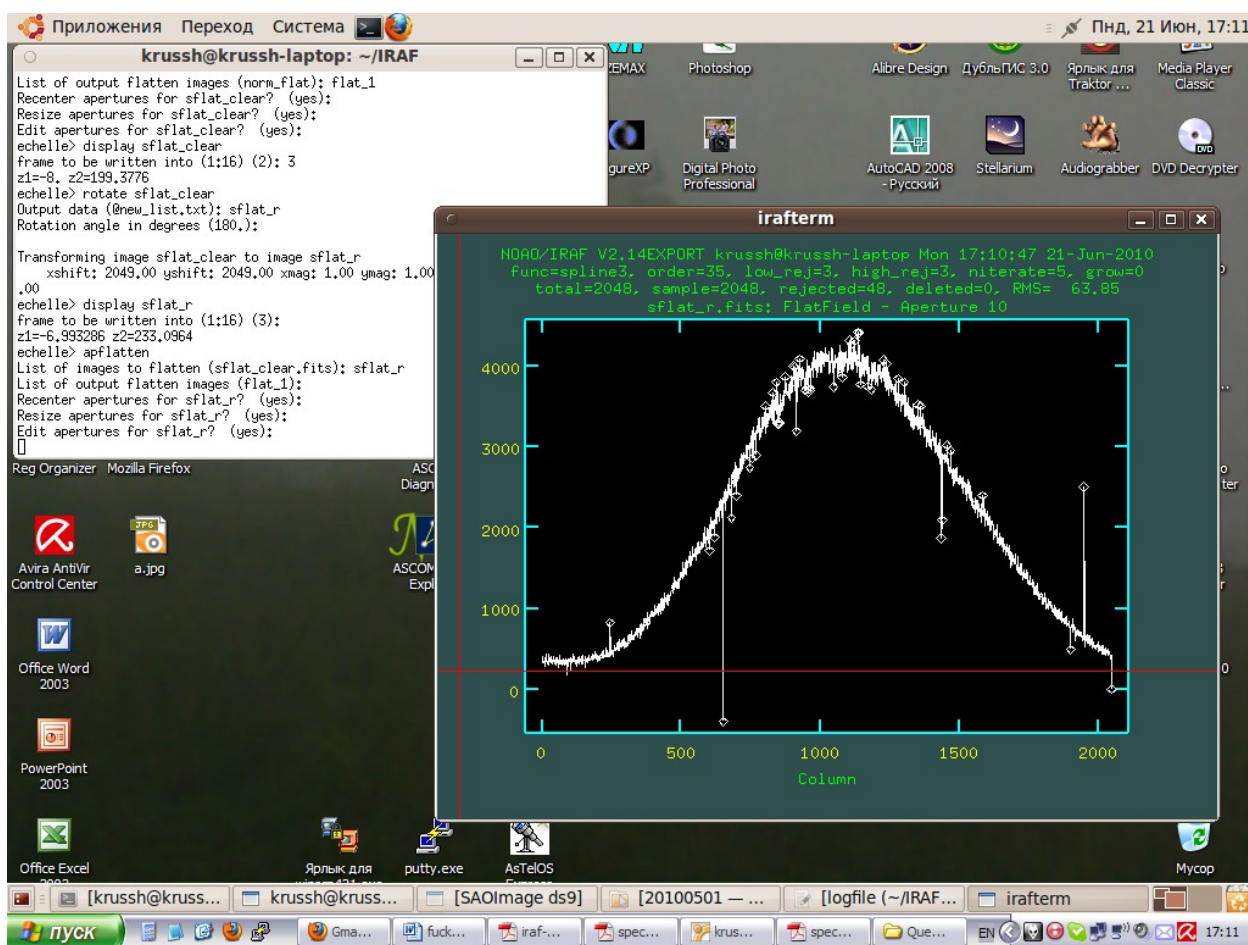
Первая часть, а именно коррекция положения и ширины апертур не обязательна (даже не желательна). По завершении нажмите **q**, для перехода к следующей процедуре – получение полинома (лучше сплайна), описывающего плоское поле. Необходимо учесть только функцию блеска и виньетирование, итоговая кривая должна быть плавной. Соответственно, фринги остаются и вносят свой вклад в плоское поле. Это связано с тем, что

для спектров высокого разрешения, где фон неба пренебрежимо мал, фринги возникают при интерференции света от объекта в матрице ПЗС и, по сути, являются вариациями чувствительности матрицы. Они являются **компликативной** составляющей и на них надо делить. Для спектров низкого разрешения фринги возникают при интерференции фона неба и являются аддитивной составляющей, ее надо вычитать.

Чтобы увеличить порядок приближающего полинома используйте команду **:order** (не забудьте двоеточие). После **:order** введите нужный порядок полинома.

:order nn,

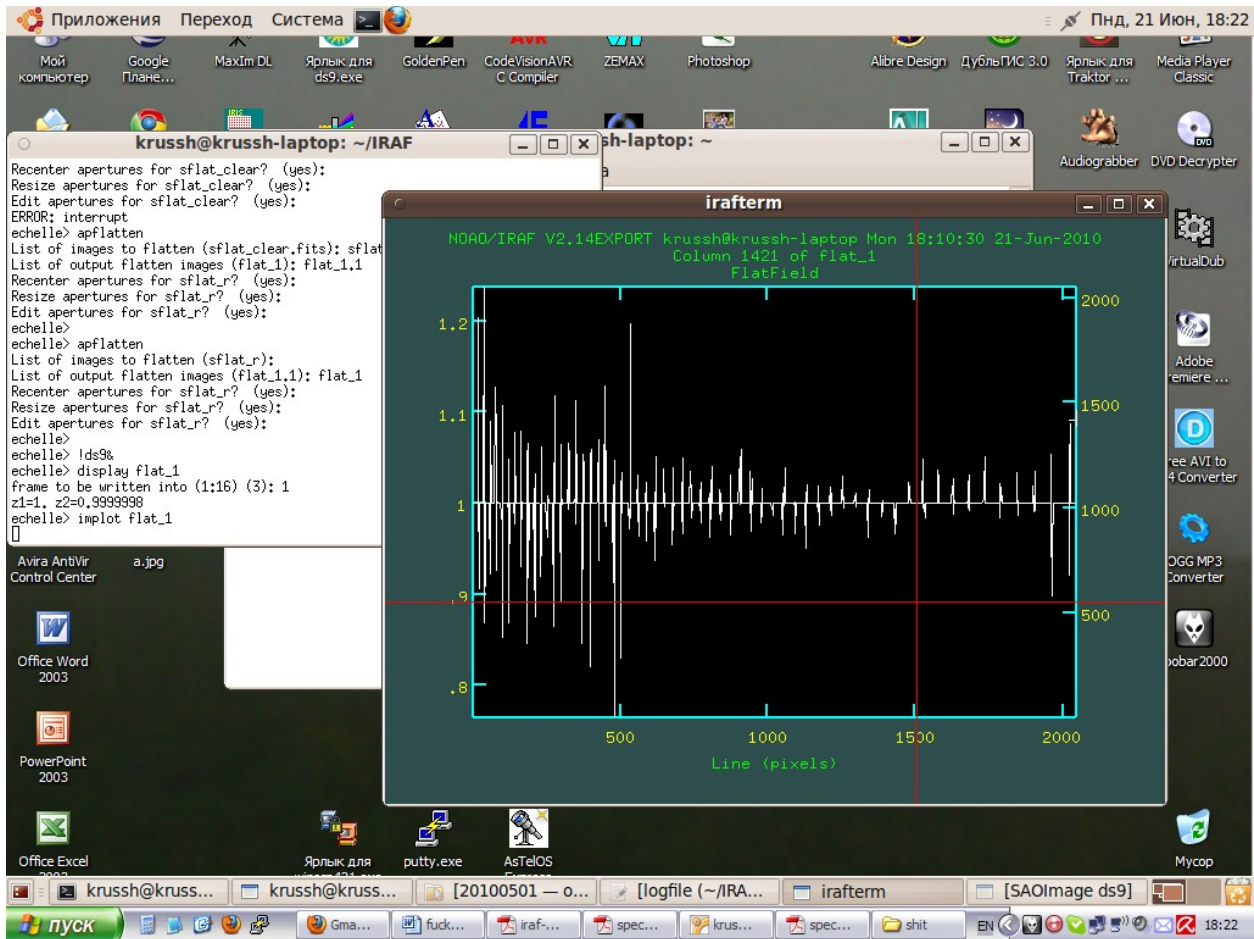
где **nn** – порядок. Для удаления точек, имеющих большие выбросы, наводите на них курсором и нажмите **d**, Чтобы обновить картинку используйте **f**. Приблизить участок можно так же, как и в **apall**, используйте клавиши **w**, **e** и **a**. Вызов справки: **shift + ?**. Переход дальше – **q**.



После завершения работы вы можете посмотреть разрез полученного кадра плоского поля, используя команду

echelle> implot flat

Для построения разреза по колонкам нажмите **c**. Должно получиться что-то похожее на вот это:



Между порядками значения равны единице, в порядках близки к единице.
Остается поделить изображение спектра объекта на плоское поле:

```
echelle> imarith object / flat
```

Еще один вариант учесть плоское поле — экстрагировать его спектр по уже существующей опорной маске (по ней же мы экстрагировали спектр)

```
apall flat.fits ref=Flat.fits interac=no find=no edit=no trace=no fittrace=no
```

Нормировать извлеченный спектр плоского поля на 1, параллельно исключив инструментальную функцию и наклон непрерывного спектра лампы плоского поля с помощью команды `continuum`.

Задаем нужные параметры:

```
continuum flat.ec.fits flat_c.fits type=ratio
```

Проводим знакомую процедуру: надо найти функцию, приближающую порядки спектра плоского поля, чтобы убрать только низкочастотную составляющую, приближающая функция должна быть гладкой. С помощью клавиш `k`, `l`, `j`, `h` можно менять визуализацию. Все, что остается после деления на приближающую функцию — это неоднородность чувствительности пикселей и некоторые оптические эффекты, которые частично или полностью исключаются именно делением на нормированное плоское поле (например, так называемые фринги — интерференционные полосы в красной части кадра).

Сочетание этих двух команд по сути то же самое, что и `apflattten`.

Дальше делим спектр объекта на плоское поле:

```
echelle> imarith object / flat
```

Важно поделить спектр объекта на плоское поле до привязки к шкале длин волн. Дело в том, что для удобства и экономии машинного времени при работе со спектром, в процессе привязки к шкале длин волн дисперсионную кривую линеаризуют, что приводит к небольшому смещению значений потока внутри порядка. В этом случае после такого смещения деление на флэт будет некорректным.

Построение дисперсионной кривой

Находим усредненный спектр лампы ThAr. Проводим его экстракцию с помощью arall по опорному спектру Flat

```
arall thar.fits ref=Flat.fits interac=no find=no edit=no trace=no fittrace=no
```

Дальше строим дисперсионную кривую с помощью интерактивной команды eidentify.

Сначала редактируем ее параметры:

```
epar eciid
```

I R A F

Image Reduction and Analysis Facility

```
PACKAGE = echelle
```

```
TASK = eidentify
```

```
images =      Thar.ec.fits Images containing features to be identified # наш спектр лампы ThAr
(databas=     database) Database in which to record feature data # название папки с базовыми
данными, она лежит в рабочей папке и уже создана автоматически командой arall
(coordli=     linelists$thar.dat) User coordinate list # файл со списком линий, по умолчанию iraf
его ищет в /iraf/iraf/noao/lib/linelists
(units =      angstroms) Coordinate units # единицы измерения, в которых строится дисп.
кривая
(match =      0.008) Coordinate list matching limit in user units # максимальная разница
между текущим значением длины волны линии и значением из опорного списка для
автоматического поиска
(maxfeat=     4000) Maximum number of features for automatic identif # максимальное
число линий, которые iraf может найти автоматически. Это число есть смысл ограничивать
только при работе со спектрами среднего и низкого разрешения
(zwidth =     10.) Zoom graph width in user units
(ftype =      emission) Feature type # тип линий. Эмиссионные – в случае лампы с
линейчатым спектром, абсорбционные в случае привязки по линиям Солнца или атмосферы
(такое иногда практикуют)
(fwidth =     4.) Feature width in pixels # ширина линии в пикселях. Она у всех линий
одна и равна толщине инструментального контура FWHM
(cradius=     5.) Centering radius in pixels
(thresho=     1000.) Feature threshold for centering # порог обнаружения линии по
величине потока (не абсолютное значение, а за вычетом окружающей подложки)
(minsep =     4.) Minimum pixel separation # минимальная разница между пиками
линий в пикселях
(funcnio=     chebyshev) Coordinate function # тип функции – дисперсионной кривой
(xorder =     8) Order of coordinate function along dispersion # порядок кривой вдоль
дисперсии
(yorder =     6) Order of coordinate function across dispersion # порядок кривой поперек
дисперсии
(niterat=     0) Rejection iterations # число итераций
(lowreje=     3.) Lower rejection sigma
More
```

ESC-? for HELP

I R A F

Image Reduction and Analysis Facility

```
PACKAGE = echelle
```

```
TASK = eidentify
```

```
More
```

```
(highrej=     3.) Upper rejection sigma
```

```
(autowri=     yes) Automatically write to database? # автоматическая запись в базу
```

данных

(graphic= stdgraph) Graphics output device # графическое отображение в интерактивном режиме (графический терминал iraf)

(cursor =) Graphics cursor input

(mode = ql)

ecidentify thar.fits

Затем запускаем команду. По атласу отмечаем несколько линий (около 10). Эти линии должны быть равномерно распределены по спектру, лучше отметить пары - одни и те же линии на соседних порядках. Затем ждем f, смотрим предварительный результат. Кривая будет построена по указанным 10 точкам (линиям). Если результат похож на желаемый, выходим из просмотра дисперсионной кривой с помощью клавиши q, нажимаем l – автоматический поиск линий, снова нажимаем f. Неправильно отмеченные линии можно удалить клавишей d. Центрировать линии можно клавишей c. Центрировать все отмеченные линии можно сочетанием клавиш ac. Процедуру можно прервать на любом шаге. I - немедленно остановить процедуру без записи данных в базу. q – выйти, данные будут записаны в базу. После выхода из процедуры к ней можно вернуться и начать с того, на чем закончили. Если результат последней "сессии" неудовлетворителен, можно открыть текстовый файл с дисперсионной кривой в папке database и удалить последнюю сохраненную запись (она начинается с даты и времени завершения процедуры).

Важно, чтоб апертуры правильно отождествились с абсолютным значением порядка. Это видно в шапке в списке параметров в окне irafterm. Если открыт какой-то порядок, там написано aperture=*** order=*** эти значения должны соответствовать значениям в атласе для данного спектрографа. Если открыта дисперсионная кривая, там же в шапке есть параметр offset=***, это абсолютный номер крайнего порядка +1. Бывает, что изображение немного сдвигается, один порядок уходит, другой добавляется. Тогда соответствие номера апертуры и порядка поменяется, но абсолютный номер одного и того же порядка, пусть даже сбился его номер на кадре, не меняется никогда. Порядки легко друг от друга отличить по расположению эмиссионных линий.

Первый раз построение дисперсионной кривой может занять значительное количество времени. Для следующего сета наблюдений (за эту же ночь или следующую) строить дисперсионную кривую значительно проще. Для этого в IRAF есть команда ecreidentify. Чтобы ее использовать, надо поместить в database текущего сета файл с дисперсионной кривой предыдущего сета. Обычно он называется ес<имя файла со спектром лампы без .fits>. Далее редактируем параметры команды ecreidentify

IRAF

Image Reduction and Analysis Facility

PACKAGE = echelle

TASK = ecreidentify

images = Thar.ec.fits Spectra to be reidentified # спектр лампы в текущем сете

referenc= thar.ec Reference spectrum # имя файла, по которому построена опорная кривая

(shift = INDEF) Shift to add to reference features

(cradius= 1.) Centering radius

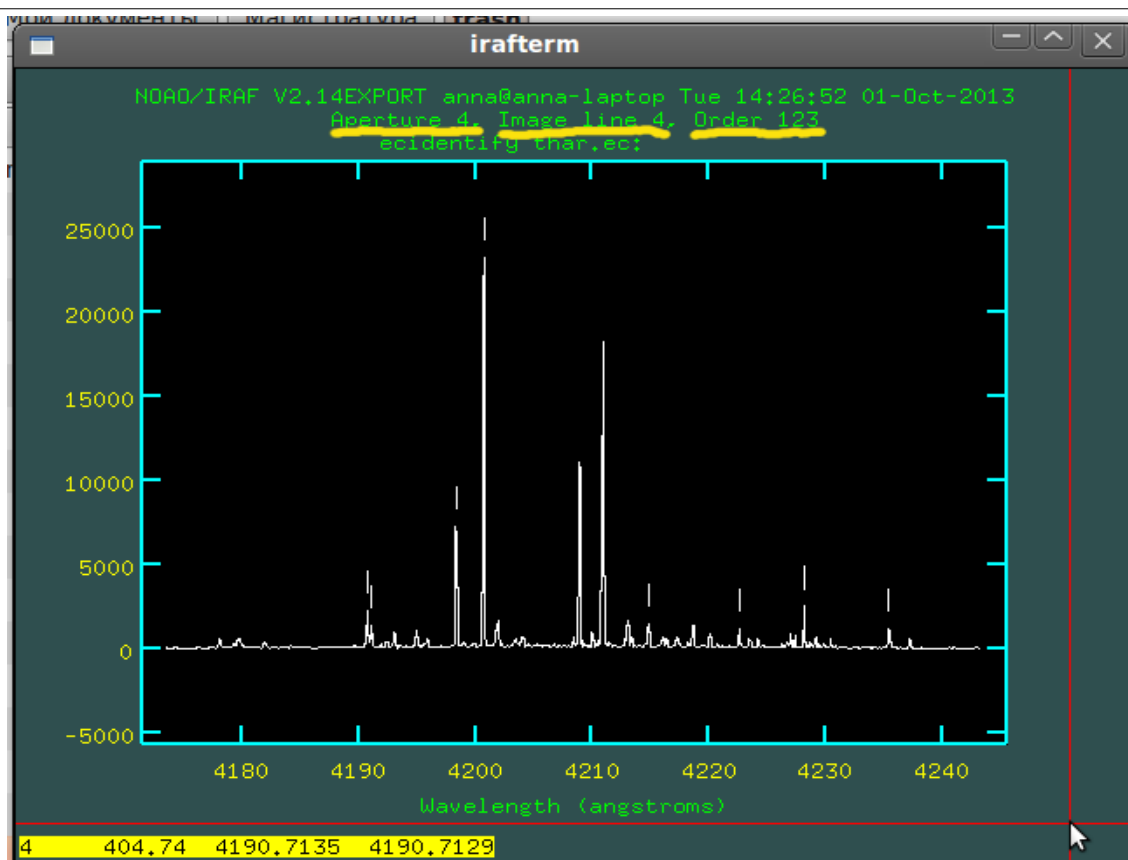
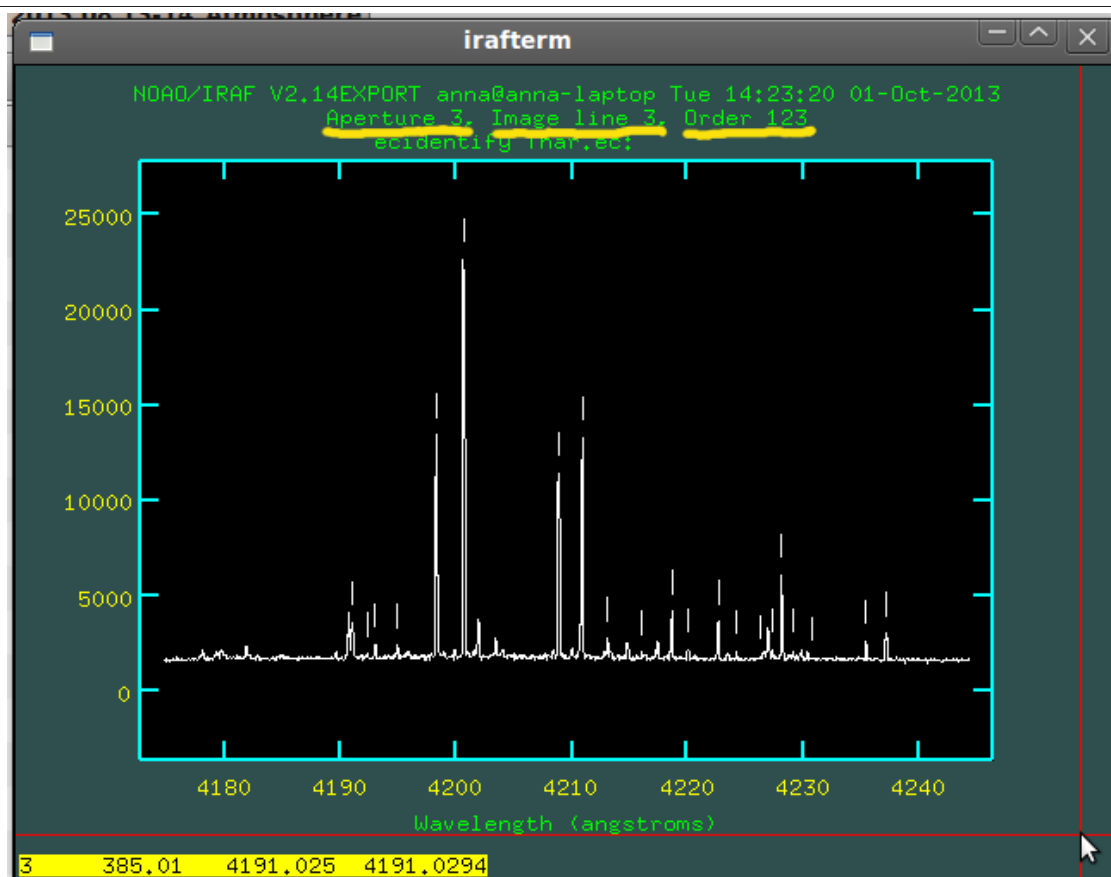
(thresho= 200.) Feature threshold for centering

(refit = yes) Refit coordinate function?

(databas= database) Database

(logfile= STDOUT,logfile) List of log files

(mode = ql)



Эти спектры ThAr сняты с разницей примерно год. Нумерация апертур в кадре (aperture) не совпадает из-за сдвига; абсолютная нумерация порядков (order) совпадает. Отождествить порядки можно по характерному расположению сильных линий.

Редактируем, сохраняем (Ctrl+d) и запускаем. >ecrid Thar.ec.fits Команда не интерактивная, так что после того, как она отработает, запускаем >ecid Thar.ec.fits. Тут повторяется описанная ранее процедура с той разницей, что большая часть линий (примерно 100) автоматически найдена, остается только подкорректировать параметры, чтоб найти максимальное число линий (для спектров с UFES это число примерно 1500). Дисперсионная кривая должна быть построена очень точно, как в целом (это можно проследить по ее среднеквадратической ошибке), так и в отдельных частях порядка (это надо проверять по конкретным линиям, насколько их длина волны отличается от значения в атласе или опорном списке). У хорошо съюстированного спектрографа точность дисперсионной кривой должна быть близка к теоретической, то есть той, которую позволяет конструкция. В частности, для UFES ско составляет 0,003-0,005 ангстрем. Нет смысла строить кривую с ско меньше теоретической ошибки. Также нет смысла достигать меньшей ско в ущерб точности кривой на отдельных участках. Пусть лучше средняя ошибка будет больше, но кривая будет построена по большому числу точек, равномерно распределенному по кадру.

Привязка спектров к шкале длин волн

Кривая построена, можно привязать спектры объектов к этой шкале. Сначала каждому спектру надо указать опорный спектр – спектр ThAr лампы. Для этого есть команда refspectra. Как всегда, сначала редактируем параметры >epar refspectra

I R A F

Image Reduction and Analysis Facility

PACKAGE = echelle

TASK = refspectra

input = sky*.fits List of input spectra # имена файлов, к которым надо привязать шкалу можно задать маской, списком или просто по отдельности
(referen= thar.ec) List of reference spectra # имя файла со спектром лампы
 (apertur=) Input aperture selection list
 (refaps =) Reference aperture selection list
 (ignorea= yes) Ignore input and reference apertures?
 (select = interp) Selection method for reference spectra
 (sort =) Sort key
 (group =) Group key
 (time = no) Is sort key a time?
 (timewra= 17.) Time wrap point for time sorting
 (overrid= yes) Override previous assignments?
 (confirm= yes) Confirm reference spectrum assignments?
 (assign = yes) Assign the reference spectra to the input spectr
 (logfile= STDOUT,logfile) List of logfiles
 (verbose= no) Verbose log output?
 answer = yes Accept assignment?
 (mode = ql)

Редактируем и сохраняем (Ctrl+d). Запускаем команду >refspec

Теперь у каждого файла есть опорный спектр (у всех один и тот же). Надо привязать шкалу, для этого есть команда dispcor. Редактируем параметры >epar dispcor

I R A F

Image Reduction and Analysis Facility

PACKAGE = echelle

TASK = dispcor

input = sky_26.fits List of input spectra # имя файла, которому нужна привязка
output = sky_26.fits List of output spectra # имя нового файла со шкалой длин волн

(linearize= yes) Linearize (interpolate) spectra? # линейризацию проводить очень рекомендуется, т.к. иначе операции с файлом будут очень громоздкими – решать линейные уравнения всегда значительно проще, а изменения координат по сравнению с полиномом незначительные

(database= database) Dispersion solution database # папка с файлом echar.ec

(table =) Wavelength table for apertures

(w1 = INDEF) Starting wavelength

(w2 = INDEF) Ending wavelength

(dw = INDEF) Wavelength interval per pixel

(nw = INDEF) Number of output pixels

(log = no) Logarithmic wavelength scale?

(flux = no) Conserve flux? # разрешает менять значение потока (значение потока есть смысл сохранять только для спектрофотометрии), если его сохранить, форма континуума в порядке значительно усложнится

(blank = 0.) Output value of points not in input

(samedis= no) Same dispersion in all apertures?

(global = no) Apply global defaults?

(ignorea= no) Ignore apertures?

(confirm= no) Confirm dispersion coordinates?

(listonl= no) List the dispersion coordinates only?

More

ESC-? for HELP

I R A F

Image Reduction and Analysis Facility

PACKAGE = echelle

TASK = dispcor

More

(verbose= yes) Print linear dispersion assignments?

(logfile=) Log file

(mode = ql)

Редактируем и сохраняем параметры (Ctrl+d) и запускаем команду >dispcor
Теперь спектры объектов имеют свою шкалу длин волн (уравнения для каждого порядка записаны в фитс-хедере). Полюбоваться на обработанный спектр можно, например, с помощью команды splot. Тут же можно провести некоторый анализ спектра.
Но это уже совсем другая история :)

Вывод графической информации на печать

Вывести картинку из irafterm в файл/на печать:

```
:.snap [device]
```

По умолчанию [device] - то, что указано в файле login.cl в поле stdplot

Возможные варианты:

lw, lpr - черно-белая печать на принтер, установленный по умолчанию

eps - ч/б eps (encapsulated postscript file) файл, альбомная ориентация изображения, каждая запись в новый файл в рабочую папку psXXXX.ps, где XXXX - ряд чисел

epsf - ч/б eps (encapsulated postscript file) файл, портретная ориентация изображения, каждая запись в новый файл в папку tmp, /tmp/psXXXX.ps, где XXXX - ряд чисел

epsfl - ч/б eps (encapsulated postscript file) файл, альбомная ориентация изображения, каждая запись в новый файл в папку tmp, /tmp/psXXXX.ps, где XXXX - ряд чисел

psdump - ч/б ps (standard postscript) файл с альбомной ориентацией изображения, каждая запись в новый файл в папку tmp, /tmp/psXXXX.ps, где XXXX - ряд чисел

psidump - цветной ps (standard postscript) файл с альбомной ориентацией изображения, все записи одной сессии делаются в один файл в папку tmp, /tmp/psXXXX.ps, где XXXX - ряд чисел. Отличие от irafterm - черный фон меняется на белый, белые буквы - на черные.

psidumprr - цветной ps (standard postscript) файл с портретной ориентацией изображения, все записи одной сессии делаются в один файл в папку tmp, /tmp/psXXXX.ps, где XXXX - ряд чисел. Отличие от irafterm - черный фон меняется на белый, белые буквы - на черные.

g-gif - судя по всему, можно делать анимированные гифки размером 640*480, до 8 изображений в файле. Подробно см. исходную страничку

<http://www.astro.yale.edu/ast255/iraf/iraf-login.cl.html>

Собрать атлас калибровочной лампы в ps/pdf:

Включить identify/ecidentify для нужного файла лампы, дисперсионную кривую уже построили

Отметить линии длинами волн :labels [units]

[units] могут быть none, index, pixel, user, coord, both

Надо выбрать user, coord или both тогда у отождествленных линий появятся значения длин волн и элементы

Вид подписи можно улучшить с помощью :txset [параметр]=[значение]

Параметры следующие: <http://iraf.net/irafhelp.php?val=cursors&help=Help+Page>

keyword	values	default
---------	--------	---------

up	degrees counterclockwise, zero = +x 90	
size	character size scale factor	1.0
path	left, right, up, down	right
hjustify	normal, center, left, right	left
vjustify	normal, center, top, bottom	bottom
font	roman, greek, italic, bold	roman
quality	normal, low, medium, high	normal
color	integers greater than one	1

Возможно, это работает только для текста, вписываемого в поле с клавиатуры (такое тоже возможно). Для автоматических подписей работает только :txq h - высокое качество текста, это позволяет развернуть символы в подписях так, чтоб они читались вертикально снизу вверх

Итак, включаем подписи к линиям :labels coord, приводим в порядок текст :txq h, пишем картинку в файл одним из указанных выше способов. Если это :.snap psidump, получится

единый ps файл, по сути готовый атлас. Можно перегнуть его в pdf, например, он-лайн-сервисом ps2pdf.com. Если делаем ч/б атлас `..snar psdump`, получим несколько/много ps файлов. Их можно собрать в один ps файл линуксовой командой `psmerge -o<out.ps> files.ps` например, `psmerge -oatlas.ps *.ps`

Надо помнить, что таким образом объединить можно ps файлы единообразные, одинаково созданные одним и тем же устройством (чтоб шапка была одинаковая)

Этот ps файл опять-таки можно перегнуть в pdf.