

Photometry using IRAF (ver. 2. 5. 1)

(IRAF를 이용한 구경 측광 및 PSF 측광)

2012. 3. 25

박 근 홍, 류 진 혁, 장 인 성, 황 호 성

0. 들어가는 글

2009년 처음으로 천문관측 및 실험 1의 조교를 맡았습니다. 그간 천문학과에서 널리 사랑을 받아온 선배님들의 안내서들¹⁾(ver. 1. 0)을 바탕으로 하여 내용을 추가, 구조를 재편하여 두 개의 안내서²⁾(ver. 2. 0)를 만들었습니다.

그로부터 3년이 지났습니다. 그 사이 천문학과는 전산 실습실을 갖췄고, 덕분에 학생들은 관측 자료 처리를 그간 사용되어왔던 astro10 서버가 아닌 전산 실습실에서 할 수 있게 되었습니다. 그래서 astro10에 깔려 있던 cl(command language)기반의 구버전 IRAF가 아닌, ecl(enhanced command language)기반의 최근 IRAF를 기준으로 IRAF 안내서를 다시 손보았습니다. 특히 이명균 선생님 팀원들의 도움을 받아 이전 버전에서 상대적으로 부실했던 표준화에 대한 설명을 강화하였고, 공간적인 표준화에 관한 내용을 추가할 수 있었습니다.

표준화 부분의 수정 및 검수를 맡아준 류진혁 학생, 구정보정 부분의 수정과, 공간적인 표준화 부분을 작성해준 장인성 학생 및 본 안내서에 기탄없는 의견과 오류를 지적해주신 김성진 선배님께 지면을 빌어 감사의 말씀을 드립니다. 해당 안내서의 오류 및 건의 사항이 있으면 khpark@astro.snu.ac.kr로 언제든지 알려주시면 수정 혹은 적극 반영토록 하겠습니다. 부족하지만 측광 작업 입문자에게 도움이 되길 바랍니다.

2012년 3월, 서울대학교에서
박 근 홍

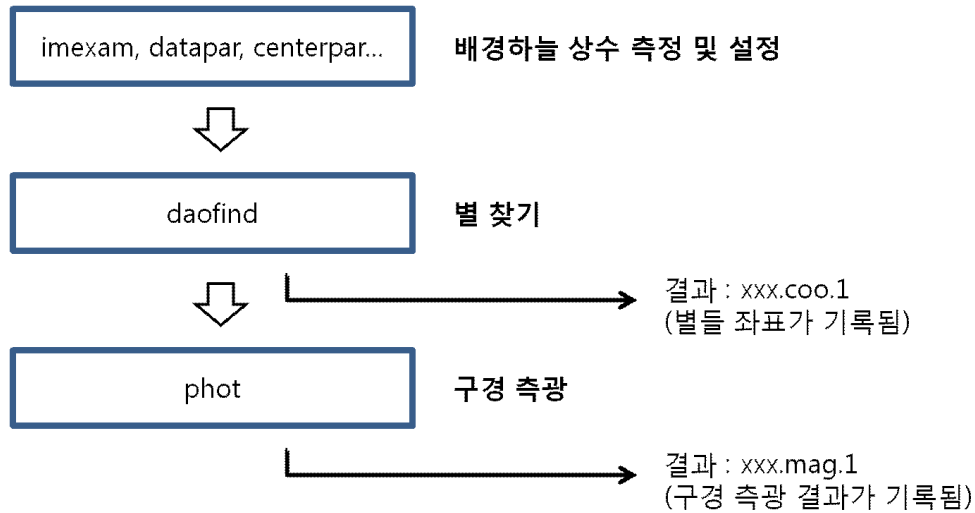
(※ SAGA 홈페이지³⁾에서 가장 최근 버전의 안내서를 받을 수 있습니다.)

1) IRAF ABC (황호성, 이명균, 2004), DAOPHOT을 이용한 PSF측광 (황호성, 이명균, 2004), IRAF/PHOT을 이용한 변광성 측광 (황호성, 이명균, 2006), PHOTCAL Tutorial (이명균, 이준협, 김상철, 2002)

2) Linux 명령어 및 IRAF를 이용한 전처리 (황호성, 이명균, 박근홍, 이광호, 이상각, 박홍수, 류진혁, 2009), IRAF를 이용한 구경측광, PSF측광 그리고 표준화 (황호성, 이명균, 이준협, 박근홍, 이광호, 이상각, 박홍수, 류진혁, 2009)

3) <http://astro.snu.ac.kr/~grad>

I. 구경측광



[그림 1] 구경 측광 작업의 순서

측광(photometry)은 빛의 세기를 측정하여, 별의 밝기를 구하는 작업이다. 구경측광이란, 별의 중심에서 일정 반경 안에 있는 빛의 세기를 측정하는, 가장 기초적인 측광 방법이다.

구경측광 및 PSF측광에는 DAOPhot task를 사용한다. 이 때 DAOPhot은 '다오팩'으로 발음하는 것이 아니라 '디에이오팩'으로 발음해야한다. IRAF를 실행 한 뒤 다음과 같은 명령어로 DAOPhot mode를 실행한다.

```

ec1> noao.digi
      apphot.  daophot.  photcal.  ptools.
digiphot> da
      addstar  daotest   nstar    pexamine   psf
      allstar  datapars@ pcalc    pfmerge    psort
      centerpars@ findpars@ pconcat  phot       pstselect
      daoedit  fitskypars@ pconvert photpars@ seepsf
      daofind  group     pdump    prenumber  setimpars
      daopars@ grpselect peak      pselect   substar
daophot>
  
```

tip1, IRAF에서 바로 이전 mode로 이동하고 싶을 때는 bye를 쓰면 된다.

```

daophot> bye
      apphot.  daophot.  photcal.  ptools.
digiphot> bye
      dataio.  language.  obsolete.  softtools.
      dbms.    lists.     plot.       system.
      images.  noao.      proto.      utilities.
ec1>
  
```

1) 배경하늘 상수 구하기

별을 찾고, 등급을 구하기 위해서는 배경하늘의 값과 요동, 그리고 별의 반치폭(FWHM)을 알아야 한다. daophot 명령어로 위 값들을 쉽게 구할 수 있다. 명령어 순서는 daoedit (파일명) 이다. 이 경우 xgterm 상에 바로 데이터를 출력해주고, 따로 텍스트 파일의 형식으로 저장하기를 원하면 뒤에 > (저장할 파일명)을 붙여주면 된다. daoedit를 실행하면 ds9창으로 커서가 이동하고 주위에 비교적 밝고 주위에 다른 별이 없는 별 위에 커서를 올린 후 **[r]**키를 눌러서 별의 밝기 profile을 본다. saturation이 일어나지 않았고 별다른 이상한 점이 없으면 **[a]**키를 눌러서 별을 선택한다. 5-10개 정도의 별을 선택하고 난 뒤 **[q]**키를 눌러 빠져나온다.

```
daophot> disp M50b 1 fit+
z1=306.7036 z2=403.3371
daophot> daoedit M50b
Warning: Graphics overlay not available for display device.
```

#	XCENTER	YCENTER	SKY	SKYSIGMA	FWHM	COUNTS	MAG
383.37	349.07	338.8	8.51	3.34	4908.1	-9.227	
294.98	100.20	336.2	7.11	3.38	17286.2	-10.594	
33.00	359.15	338.9	11.11	3.18	2830.5	-8.630	
178.06	366.29	344.4	8.34	3.63	2386.4	-8.444	
249.48	167.75	339.7	7.21	4.65	1216.2	-7.713	

즉 예시의 경우 배경하늘의 평균은 339.6, 요동값은 8.46, 반치폭은 3.63 이다.

2) 상수 설정.

위에서 구한 값을 가지고 각 작업의 상수를 수정해준다.

```
daophot> datapars
```

- fwhmpsf : PSF의 반치폭값. 위에서 계산한 FWHM값을 넣는다.
- sigma : 배경하늘의 요동값. 위에서 계산한 sigma값을 넣는다.
- datamax : 포화(saturation)가 일어나는 값. 해당 CCD의 최대 용량(Fullwell Capacity)보다 약간 작은 값을 넣는다. 학교 CCD등 60K인 경우가 일반적이다.
- ccdread, readnoi : Readout Noise 값을 설정하는 항목. ccdread에는 헤더의 키워드, readnoi는 숫자를 입력한다. 둘 중 하나만 넣어도 된다. (둘 다 넣을 경우 헤더의 값을 우선시 한다.)
- gain, epadu: Gain 값을 설정하는 항목. gain에는 헤더의 키워드, epadu는 숫자를 입력한다. 둘 중 하나만 넣어도 된다. (둘 다 넣을 경우 헤더의 값을 우선시 한다.)
- exposur, itime: 노출 시간을 설정하는 항목. exposur에는 헤더의 키워드, itime에는 숫자를 입력한다. 둘 중 하나만 넣어도 된다. (둘 다 넣을 경우 헤더의 값을 우선시 한다.)

Notice 1. 이미지 합침 과정에서 Gain과 RDnoise가 제대로 바뀌었나 체크한다. 이 두 수치가 제대로 입력이 되어야 등급 오차를 제대로 계산할 수 있다. 두 수치가 바뀌지 않았다면 gain과 RDnoise값을 변경해주도록 한다. 이미지 합성방식에 따라 바꾸는 방식이 다른데

Sum으로 N개의 이미지를 합쳤다면 Gain은 그대로, RDnoise는 \sqrt{N} 배,

Average로 N개의 이미지를 합쳤다면 Gain은 N배, RDnoise는 \sqrt{N} 배,

Median으로 N개의 이미지를 합쳤다면 Gain은 $\frac{2}{3}N$ 배, RDnoise는 $\sqrt{\frac{2}{3}N}$ 배이다.

daophot>centerpars

-cbox: 천체의 중심을 잡는 범위폭을 의미한다. 5나 1.5*FWHM 중 큰 값으로 수정한다

daophot>fitskypars

-annulus: 배경하늘 맞춤(skyfitting)을 하는 고리모양의 지역의 안쪽 경계를 말한다.

4*FWHM (표준성을 처리한다면 7"에 해당하는 픽셀 개수)을 넣어주면 된다.

-dannulus: 배경하늘 맞춤을 하는 고리모양의 지역의 폭을 말한다. 천체들의 밀집된 정도에 따라 FWHM의 2~3배의 값을 넣어준다.

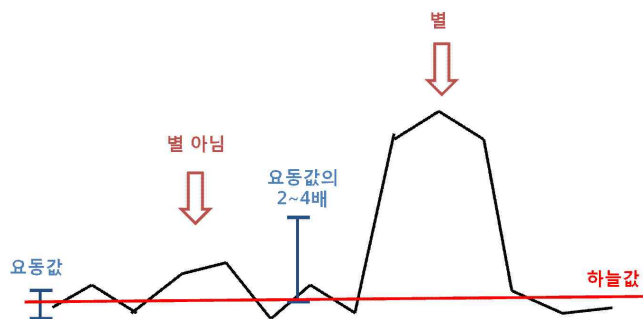
-skyvalu: 배경하늘 값. 위에서 계산한 Sky 값을 입력한다.

daophot>epar photpars

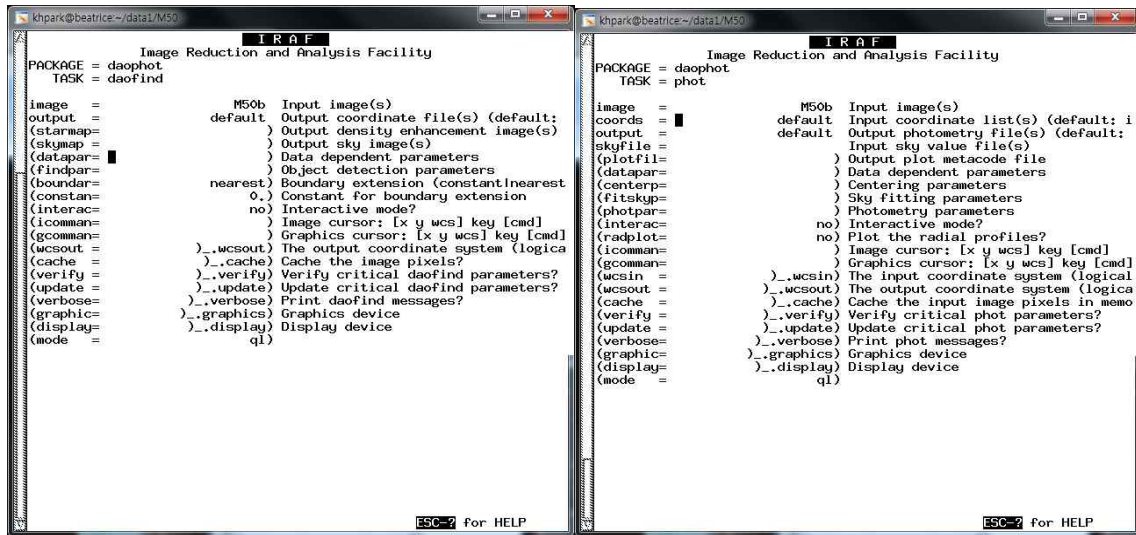
-aperture: 구경측광 시 구경의 크기. 연구의 목적에 따라 적당하게 주어주되 일반적으로 1*FWHM 또는 3 중 큰 값으로 입력한다. (표준성을 처리한다면 7"에 해당하는 픽셀 개수)

tip2. 구경값을 FWHM의 1배 일 때 신호 대 잡음비가 가장 좋다. 별에서 나오는 모든 광자를 잡고 싶을 때는 FWHM의 3배를 지정해주면 된다.

3) 별 찾기



[그림 2] 별 찾기의 원리



[그림 3] daofind를 이용해 별 찾기

[그림 4] 구경 측광 하기

이미지에서 별이 있는 곳의 좌표를 찾아주는 작업이다. daofind명령을 통해 자동으로 이미지에서 별을 찾을 수 있다. [그림 3]과 같이 epar설정 후 실행을 하거나, 혹은 아래와 같이 명령창에 입력을 하여 별을 찾을 수 있다. output 파일을 default로 설정하면 자동으로 (파일명).coo.1 이라는 별의 좌표가 기록된 파일을 생성한다.

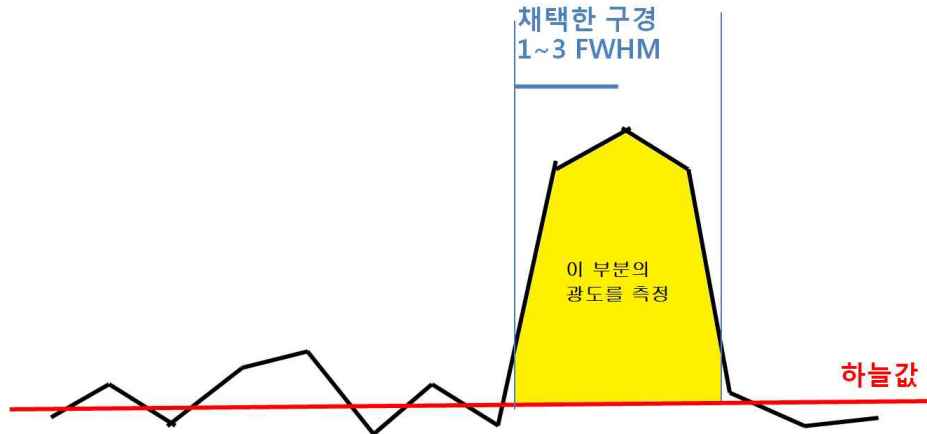
```
daophot> daofind M50b
```

실행하면 threshold값을 묻는데, 이는 배경의 요동 값의 몇 배 이상의 값을 별로 간주하겠다는 뜻이다. 2~4 정도의 값을 준다. (PSF 측광을 할 때는 처음에는 3.5 정도로 높게 주자.)

한 편, 변광성이나 외계행성과 같이 광도 변화를 보는 자료를 처리할 때, 만약 비교성·점검성이 어떤 별인지 알고 있다면 굳이 모든 별의 광도를 다 구할 필요 없이, 대상별과 알고 있는 몇 개의 비교성·점검성의 광도만 필요하다. 따라서 이 경우는 daofind를 사용하지 않고, 배경하늘 값을 측정할 때 사용했던 daoedit를 이용해서 별의 좌표를 찾아준다. 위치 정렬이 완료되어 있다면 하나의 좌표 파일을 가지고 모든 이미지에 적용할 수 있다. (그리고 이 경우, 매 출력파일마다 별의 순서가 똑같기 때문에 작업이 훨씬 수월하다)

```
daophot> daoedit M50b > M50.coo.1
```

4) 구경측광하기



[그림 5] 구경측광의 원리

[그림 4]와 같이 epar 설정 및 실행을 하거나, 혹은 아래와 같이 명령어를 입력하여 구경측광을 한다.

```
daophot> phot M50b
```

실행 후 엔터키를 몇 번 눌러주면 구경측광이 완료되고 결과는 (이미지).mag.1 파일로 출력된다. 출력된 파일을 열어보면 알아보기 어려울 정도로 너무 많은 정보가 있는 것을 볼 수 있다. 우리가 필요한 이 많은 정보 중 결국 우리가 필요한 정보는 별의 ID, 등급, 등급오차이므로 (X,Y좌표도 경우에 따라서 필요할 때가 있다.) txdump 명령어로 mag.1 파일에서 해당 정보만 뽑아낸다.

```
daophot> txdump('M50b.mag.1','ID,MAG,MAGERR','yes',>>'M50b.dat')
```

위 명령어는 M50b.mag.1파일에서 ID, MAG, MERR정보만 찾아서 M50b.dat에 기록하라는 명령이다. 뽑아낸 파일을 열고 확인해보자.

tip3. 변광성을 처리할 때 원래는 배경하늘 값이 시간에 따라 변하기 때문에 매 파일의 배경하늘 parameter들을 계산해서 수정해야하지만, 짧은 시간 범위 내에서는 크게 변하지 않는다고 가정하고 관측 중간의 이미지에서 정한 값을 그대로 써도 무방하다, 그리고 이 경우에는 아래와 같이 스크립트란 것을 짜서 편하게 여러 이미지의 구경측광을 한 번에 할 수 있다, 스크립트파일의 실행 명령은 ci < (스크립트 파일명) 이다, 즉,

```
daophot> ci < phot.ci
```

이라고 입력하면 아래와 같이 짜여진 phot.ci이란 스크립트 파일을 실행하여 모든 이미지에 대해서 구경측광이 이뤄지고 ID,등급,등급오차를 (이미지)_apc 파일에, 헤더의 JD 정보를 (이미지)_apc.jd 파일에 기록을 하는 것을 볼 수 있다, (관측장소에 따라 헤더에 JD정보가 없을

수 있다는 점에 유의하자.) 아래 스크립트 파일을 찬찬히 뜯어보고 자신의 필요에 따라 수정·응용해서 사용하도록 하자.

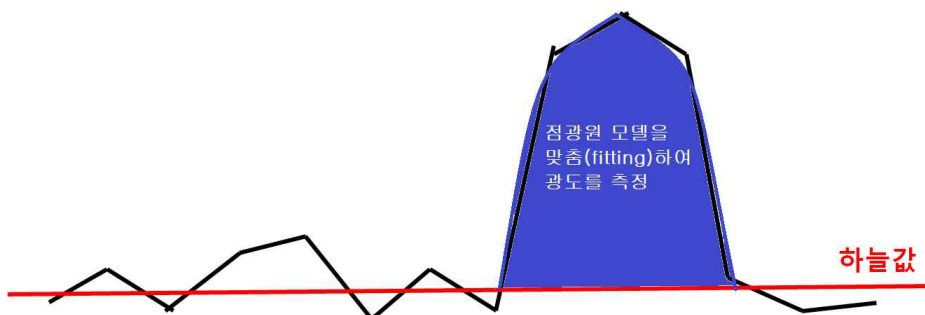
```
#START phot.cl ; 앞에 #이 붙으면 주석으로 간주한다.
!ls *fit* > phot.lis ; !는 리눅스 명령어를 IRAF에서 실행할 때 사용한다. fit 파일을
                                ; 모조리 phot.lis란 파일에 넣어둔다

!sed s/. fits//g phot.lis > phot.lst ; phot.lis파일 안의 .fits를 공백으로 바꿔서 phot.lst에 저장
list= "phot.lst" ; phot.lst를 list로 정의
while( fscan(list, s1) !=EOF) { ;list 안의 문자열을 s1으로 읽어서 끝줄까지 {}안에 있는 내용을 실행
s2="M50b.coo.1" ; s2를 M50b.coo.1로 정의
s3=s1/"_mag.1" ; s1뒤에 _mag.1을 붙여 s3에 저장
#disp (s1,frame=1,fi+)
phot (s1,s2,s3,inter=no,veri=no) ; s1,s2,s3를 input으로 하여 구경측광을 한다
txdump(s1/"_mag.1","ID,MAG,MERR","yes",>>s1/"_apc") ;mag.1파일에서 ID,등급,등급오차
                                ;정보를 뽑아서 s1_apc파일에 저장

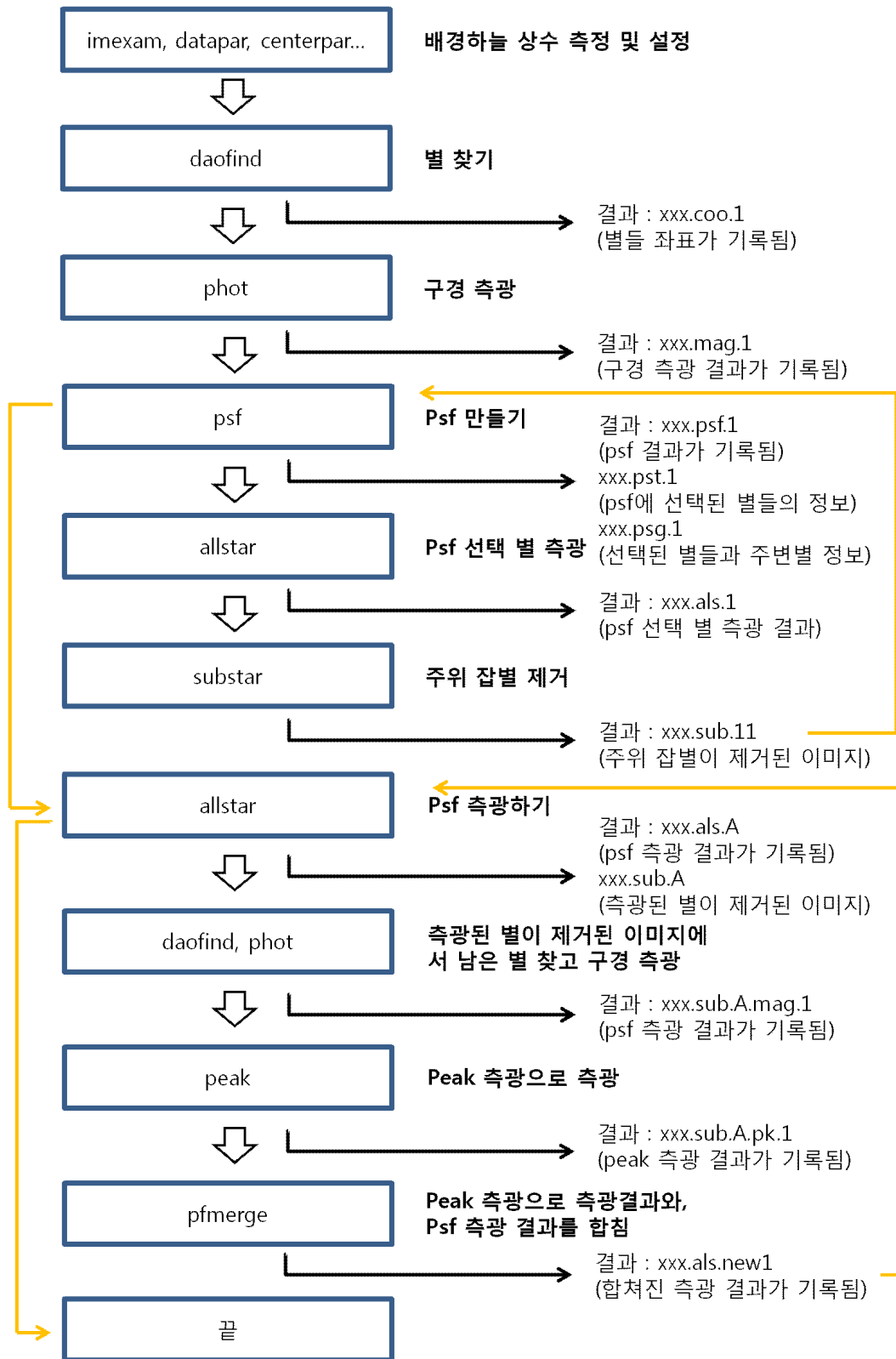
hselect (s1,"JD","yes", >> s1/"_apc.jd") ; s1의 헤더에서 JD를 뽑아 s1_apc.jd에 저장
#!sum_f s1/"_apc" s1/"_jd" s1/"_dat"
!rm -rf phot.lis
!rm -rf phot.lst
!rm -rf *mag*
}
#END
```

II. PSF(Point Spread Function; 점 광원 분포 함수) 측광

PSF 측광은 이미지상의 별의 밝기 분포에 주어진 점광원 분포 함수 맞춤을 한 후, 그 구해진 맞춤 함수를 기반으로 광도를 측정하는 과정이다. 각 픽셀에 포함된 밝기를 세기만 하면 되는 구경측광에 비해, 맞춤 함수를 구하는 과정으로 인해 좀 더 시간이 오래 걸리는 단점이 있다. 하지만 인접한 여러 별의 경우 구경 측광은 이 별들을 구분해서 측광하기 어렵지만, PSF 측광에서는 함수 맞춤 작업으로 각 별을 ‘분리’하기 때문에 구경 측광에 비해 더 정확한 결과를 얻을 수 있다. 따라서 많은 별들이 밀집해서 분포하는 성단을 측광할 때는 구경측광이 아니라 PSF 측광을 사용한다.



[그림 7] PSF 측광의 원리



[그림 8] psf 측광 과정 순서

Photometry using IRAF

1) 배경하늘 값 측정하기부터 구경 측광까지

PSF측광은 구경측광이 선행되어야 한다. 따라서 psf측광의 앞부분은 구경측광의 방법과 일치한다. 다만 상수 설정에서 psf 맞춤(fitting)에 영향을 주는 하나의 설정을 더 추가한다.

```
daophot> epar daopars
```

-psfrad : PSF model이 정의되는 원의 반지름. 4*FWHM을 입력한다.

-fitrad : fitting을 수행하는 반지름. FWHM을 입력한다.

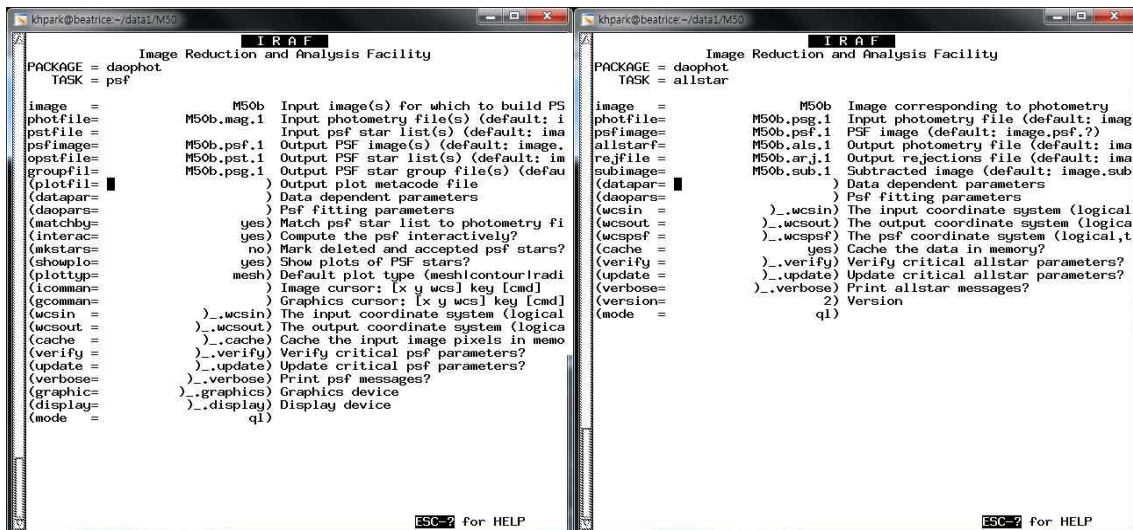
daofind와 phot명령어를 통해서 별 찾기와 구경측광하기를 완료한다.

2) PSF 구할 별 정하기

```
daophot> epar psf
```

[그림 9]과 같이 설정한다. 구경 측광에서는 출력항목들에 default를 많이 썼지만, PSF 측광부터는 수동으로 지정한다.

image는 분석하고자 하는 이미지, photfile은 구경 측광한 결과 파일, pstfile은 이전에 선택한 psf 선택 별들의 정보파일인데, 첫 작업에는 이전에 선택한 별들이 없으므로 당연히 빈칸으로 둔다. psfimage는 만들어진 psf가 저장될 파일이고 (이 작업에서 얻고자 하는 결과물이다.), opstfile은 이 작업에서 선택한 psf 구할 별들을 저장하고자 하는 파일이다 (다음 번 맞춤 단계 시 pstfile로 들어갈 파일이다), groupfil은 psf 구할 별 및 그 근처의 별들을 저장하고 있는 파일이다. 실행하면 몇가지 질문이 뜨는데, 가장 먼저 맞춤을 수행할 analytic function을 선택하라고 한다. 첫 번째 수행에서는 자동(auto)으로 두자, 차수는 처음에는 0차로 한다. (analytic function은 보통 penny1, penny2 함수가 가장 잘 들어맞고, 흐린 날씨에 관측한 이미지의 경우에는 moffat15, moffat25가 잘 들어맞는다.)



[그림 9] psf 구할 별 정하기

[그림 10] 잡별 제거를 위한 allstar 측광하기

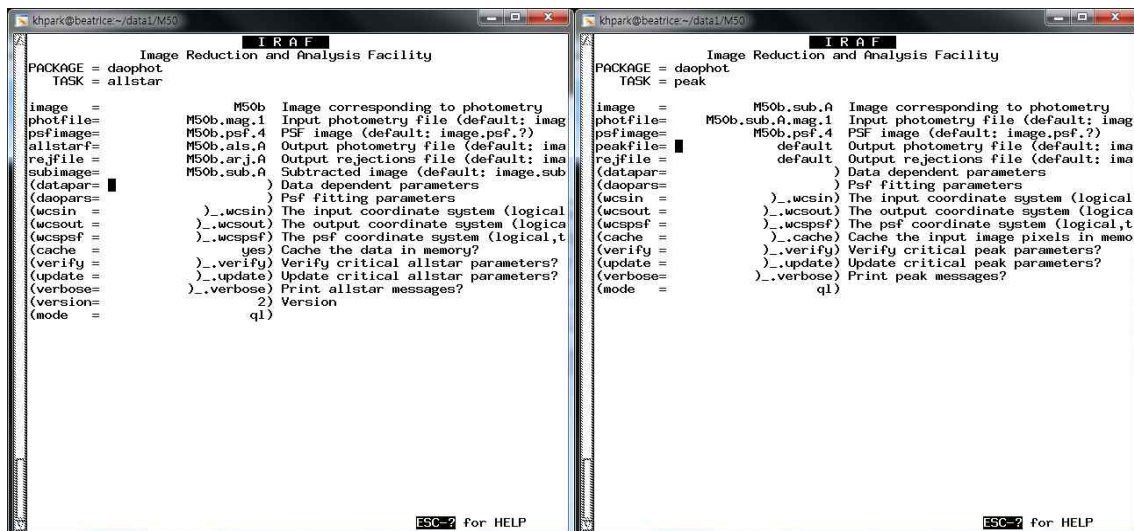

```
daophot>disp M50b.sub.1 2 fit+
```

이제 [그림 11]과 같이 substar명령을 통해 psf 구할 별 주위의 잡별들을 제거한다. psf.1을 바탕으로 앞서 구한 psf 측광 파일인 als.1 안에 들어간 별들(psf구할 별+잡별)을 원래 이미지에서 빼되, 그 중 pst.1(psf 선택 별)은 남겨두고 그 결과는 sub.11에 저장한다. 실행시키면 몇 개의 후보 군 중 몇 개의 별이 제거 되었다는 메시지가 나타난다. sub.11.파일을 디스플레이하여 psf 선택 별은 남아 있고, 그 주위의 별들이 빠진 것을 확인하자.

```
daophot> disp M50b.sub.11 3 fit+
z1=306.7036 z2=403.0552
```

이제 주위의 잡별이 제거되어, 주위의 다른 별에 의해 영향을 받지 않은 psf를 결정할 수 있다. 다시 [그림 12]과 같이 psf를 실행하도록 하자. 앞의 psf와 달라진 것은 우선 이미지가 잡별이 제거된 sub.11.이미지를 사용하고, 앞서 공백으로 두었던 psf star list항목에는 pst.1 파일이 들어갔다. 그리고 출력파일들의 끝번호가 하나씩 올라 간 것을 알 수 있다. 명령을 실행하면 지금까지 선택된 psf 구할 별들이 하나씩 디스플레이 된다. 선택하고 싶으면 **a**를 누르고, 선택하고 싶지 않으면 **d**를 눌러서 목록에서 제거한다. pst명단의 별이 모두 등장한 후에는 앞서 했듯이 더 추가 하고 싶은 별 위에 커서를 두고 **a**를 눌러 별의 모양을 확인하고 **a**나 **d**를 눌러 선택 혹은 거부한다. 실행할 때 함수와 차수를 물어보는데, 함수는 앞에서 계산 했던 함수와 동일한 함수로, 차수는 선택한 별의 개수가 적을 때는 0차에서, 많을 때는 2차를 주는데, 점점 별의 개수가 늘어나게 되므로 자연스럽게 차수를 올려주면 된다.

이제 다시 allstar, substar로 psf 구할 별의 주위 잡별들을 다시 제거하고 다시 psf를 수행한다. 별이 밀집되지 않은 지역이나 산개성단이면 이 작업을 약 2~3회, 구상성단이라면 이 작업을 4~5회 반복한다. 파일 번호는 매 작업마다 숫자를 하나씩 늘려나간다. (xxx.1 -> xxx.2 , sub.1 -> sub.2 이런 식으로..) 주위의 잡별이 더 이상 제거 되지 않을 때까지 이 작업을 반복하고, 이 정도면 psf 계산이 충분하다 생각된다면 psf 작업까지만 수행한다. 완성된 psf는 seepsf 명령어로 확인 할 수 있다.



[그림 13] psf측광을 위한 allstar 측광

[그림 14] peak 측광

```
daophot> seepsf M50b.psf.4 sM50b.psf.4
daophot> disp sM50b.psf.4 2 fit
```

4) psf 측광하기

최종 psf이미지가 완성되었으므로 이를 이용해서 전체별을 측광하자. [그림 13]과 같이 allstar 명령을 실행하자. 앞서 잡별 제거하는 단계에서의 allstar에서는, psf 선택 별과 그 주위 잡별에 대해서만 측광을 했지만, 이제부터는 모든 별을 psf측광할 것이므로 측광파일에 앞서 구경 측광의 결과인 mag.1.파일이 들어오고, 앞의 별빼기의 결과와 구분하기 위해 결과파일의 확장자를 영어로 사용한다. 모든 별에 대해 맞춤 함수를 수행하여 측광하므로, 이미지나 컴퓨터의 성능에 따라 약간의 시간이 걸릴 수도 있다. als.A 파일에는 측광 결과가 들어가고, sub.A파일에는 측광된 별들을 빼고 남은 이미지가 만들어진다. sub.A를 ds9을 통해 열어보면, 측광되지 못하고 남아있는 별들이 있다. 이와 같이 남아 있는 별들을 찾는 작업을 몇 차례 반복하여 수행하게 된다. 먼저 sub.A 파일에 daofind 명령어를 수행하여 별 찾기를 하고 (이 때 threshold값은 앞서 주었던 값보다 조금 낮은 값을 준다) phot을 이용해서 다시 구경측광을 한다.

```
daophot> daofind M50b.sub.A
daophot> phot M50b.sub.A
```

peak 명령은 psf 측광 방법 중 하나이다. allstar와 다른 점은 별의 최대값을 이용해서 측광하므로 측광 결과는 allstar에 비해 부정확하지만, 처리속도가 빠르다는 장점이 있기 때문에, 앞서 측광에 빠졌다가, 다시 찾아진 별들을 psf 측광하는 단계에서 사용한다. (물론 결과의 정확도는 떨어진다. 하지만 이후 어차피 allstar로 다시 측광하게 되므로, 측광결과는 의미가 없고, 빠르게 psf 측광을 할 수 있다는 장점만 남는다.) [그림 14]와 같이 입력하고 실행하면 측광된 별은 sub.A.pk.1에 저장되고, 거부된 별은 sub.A.rej.1에 저장된다. allstar와 달리 새로운 이미지를 생성하지는 않는다.



[그림 15] 다시 allstar로 측광하기

이제 별의 위치가 기록된 측광 파일은 앞서 allstar에서 얻은 als.A 파일과 peak에서 얻은 .sub.A.pk.1 파일 두 개가 존재한다. 이 파일들을 다음과 같이 pfmerge 명령으로 하나로 합치고 별들의 prenumber로 id정보를 정렬해준다.

```
daophot> pfmerge M50b.sub.A.pk.1,M50b.als.A M50b.als.new1
Merging photometry file M50b.sub.A.pk.1 into M50b.als.new1
Merging photometry file M50b.als.A into M50b.als.new1
daophot> prenumber M50b.als.new1
```

이제 [그림 15]와 같이 다시 allstar 작업으로 돌아간다. 측광 목록은 als.new1 파일을 적고, 결과물들의 A파일들을 B로 올린다. 다시 sub.B 이미지에 대해 daofind, phot, peak 그리고 그 결과물과 allstar의 결과물을 pfmerge, prenumber해주는 작업을 2~3회 반복하면 많은 별들까지 측광할 수 있다.

다만 threshold값을 지나치게 낮추면 결과의 신뢰도가 감소하므로 큰 욕심을 부리지 않도록 한다. 위 과정을 2~3회 반복했음에도 남는 별이 있다면 손으로 별을 찾아줄 수 밖에 없다. allstar이후 daofind로 별을 찾아주는 대신 다음과 같은 명령으로 수동으로 coo.1 파일을 만들어 준다.

```
daophot> display pM67_V3_10.sub.C 1 fit+
daophot> tvmark 1 pM67_V3_10.sub.C.coo.1 inter+
```

그 뒤 ds9상에서 자신이 별이라고 생각되는 좌표로 커서를 옮긴 후 **a**자를 누르면 현재 위치가 pM67_V3_10.coo.1 파일에 저장된다. 모든 별을 찾았다고 생각되면 **q**를 눌러서 결과를 저장하고 빠져나오도록 한다. 이 생성된 coo.1 파일에 대해 다시 구경측광, peak측광 및 allstar결과와 병합, 정렬을 하고, 다시 allstar로 psf측광을 해주도록 하자. 이러한 과정을 반복하면서 찾을 수 있는 별을 모두 찾았다고 생각될 때, allstar 명령까지만 하고 작업을 마치도록 한다.

최종적으로 남겨둘 파일은

- ① 원본 이미지 파일 pM67_V3_10.fits
- ② 최종 측광 결과 pM67_V3_10.als.Z
- ③ 최종적으로 별을 빼고 남은 이미지 pM67_V3_10.sub.Z.fits
- ④ PSF파일 pM67_V3_10.psf.4.fits

네 가지이고, 그 외의 파일 중 마지막 단계(Z) 파일들은 보관해 두되, 그 이전 단계의 파일들은 필요가 없으므로 지워주도록 하자. 또한 최종 결과물인 als.Z 파일 역시 필요 이상으로 많은 정보가 포함되어 있기 때문에 txdump 명령어로 원하는 정보만 따로 뺀다.

Notice 2. 서버 용량도 하나의 자원이다. 따라서 필요 없는 파일은 반드시 삭제하도록 하자.

5) 구경 보정(aperture correction)

이론적으로 광원의 radial profile은 무한히 멀리 퍼져나간다. 하지만 측광에서는 유한한 크기의 구경을 사용하기 때문에 필연적으로 광량의 손실이 발생하게 된다. 이 같은 광량의 손실을 보정하는 과정이 바로 구경보정이다. psf측광을 통해 얻은 등급의 구경보정과정은 아래와 같다.

우선 주위에 다른 별이 없는 외판 별 A에 대해 구경을 점차 늘려가며 구경측광을 실시한다. 어느 구경 이상이 되면 더 이상 등급이 증가하지 않는데 (일반적으로 FWHM의 3~5배 정도), 이때의 등급이 광원의 전체 등급이 된다. 구경보정량은 PSF등급과 광원의 전체 등급간의 차이를 의미한다. 최종적으로 PSF측광을 통해 얻은 모든 별의 등급에 구경보정량을 더해주면(즉, 등급에서 빼주면) 우리는 모든 별의 전체 광량을 얻을 수 있다.

III. 참고 문헌

- 이명균,이준협,김상철, 'PHOTCAL Tutorial', 2002
- 황호성, 이명균, 'DAOPHOT을 이용한 PSF 측광', 2004
- 황호성, 이명균, 'IRAF/PHOT을 이용한 변광성 측광', 2006

Appendix A. 표준성 측광 및 표준화

“수십 년 동안 매일 작은 망원경을 가지고 표준성만을 관측한 천문학자가 있습니다. 그 이름 란돌트.” -M. G. Lee, 2008

1) 표준성 측광 시 주의사항

다른 표준성 목록이 있는지는 모르겠다. 그러나 란돌트만큼 널리 알려져 있고 신뢰도가 높은 목록을 대체할 수 있을 거라고는 생각할 수 없다. 현재 이 목록의 최신판은 Landolt(2009) 이다.

표준성을 측광할 때 주의해야 하는 점은, 14"의 구경으로 **구경측광**을 해야 한다는 것이다. 이는 란돌트가 사용한 방식과 동일하게 측광을 해서 그 등급을 비교하기 위한 것이다. IRAF에서 구경을 설정할 때는 pixel radius를 넣게 되어 있으므로, 7"에 해당하는 픽셀 값을 계산하여 넣어주면 된다. 직접 해 보면 알겠지만, 14"라는 구경은 이미지에서 보이는 별보다 매우 크다.

별이 밝다면 구경 내의 대부분의 빛은 별에서 기인한 것이므로 이 문제를 생각하지 않아도 되지만 어두운 경우가 문제가 된다. 이러한 경우에는, 배경하늘 값이 별빛에 비할 수 있을 정도가 되어 등급이 정확하게 측정되지 않을 수 있다. 배경하늘 값과 그에 의한 노이즈가 구경안으로 과다하게 들어갔기 때문이다. 이때는 적절히 구경을 줄이는 것이 도움이 되는데, 일단은 12"를 권장한다. 이 이상으로 줄여야 한다면 아마도 너무 어두워서 표준화에 사용하기에 좋지 않거나, 날씨가 좋지 않아서 표준화가 힘들 가능성이 클 것이다.

2) 표준화에 대한 이해

우주로부터 오는 빛은 지구 대기를 통과하면서 소광(extinction)을 겪는다. 관측천문학에서 가장 중요한 것은 빛의 양을 정확하게 측정하는 것이므로, 대기를 뚫고 도달한 빛에서 관측된 기기등급은 말하자면 손상된 자료이다. 거기다 얼마만큼의 대기층을 뚫고 도달했는지에 따라 그 등급이 달라지고, 그 정도도 파장에 따라 각기 다르다. 정확한 빛의 양을 알기 위해서 대기에 의한 소광 효과 등을 보정해 줘야 하는데, 이것을 통칭하여 표준화(standard calibration, NOT standardization)라 한다.

표준화를 하기 위해 알아야 하는 요소들은 몇 가지가 있는데, 식을 보면서 이야기 하도록 하자.

①

$$V = v + a(b-v) + k'_v X + k''_v (b-v)X + zero_v$$

$$(B-V) = c(b-v) + k'_{bv} X + k''_{bv} (b-v)X + zero_{bv}$$

②

$$v = V + a(B - V) + k'_V X + k''_V (B - V) X + zero_V$$

$$(b - v) = c(B - V) + k'_{BV} X + k''_{BV} (B - V) X + zero_{BV}$$

①과 ②의 식은 모양은 똑같고, 다만 표준등급(B, V)과 기기등급(b, v)의 위치만 다를 뿐이다. 각각은 등급에 대한 표준화, 색에 대한 표준화 식이다. 표준화 식에는 필터시스템의 차이를 고려하기 위한 색항, 대기량을 고려하기 위해 X 라는 값으로 정의된 1차 소광항(airmass, $\equiv \sec Z$), 색과 대기량 모두 동시에 고려하는 2차 소광항, 그리고 영점 차이를 고려하기 위해 영점항이 들어간다.

표준화를 한다는 것은 기기등급으로부터 표준등급을 구하는 것이므로 ①의 식이 자연스럽게 느껴진다. 그러나 실제로 사용하는 것은 보통 ②의 형식이다. 이쪽이 더 정확하기 때문이다. 좀 더 자세하게 이유를 살펴보면, (1)기기등급보다 표준등급이 더 믿을만하기 때문에, 알려진 값을 좌변에 놓는 ①보다는 ②가 낫다. (2)특히 색 오차는 두 개 필터에서 측광된 등급의 오차로부터 계산되는 것인데, 기기등급 b, v 는 불확실하고 따라서 $b-v$ 도 불확실하다. 반면 표준등급 B, V 는 더 믿을만하므로 이에서 얻은 $B-V$ 와, 그 오차도 더 믿을만하다. (3)또한 색에 대한 표준화 식의 경우에, ②의 형식이라면 다른 $(B-V)$ 를 가지는 표준성이라도 기기등급상으로는 같은 $(b-v)$ 를 가질 수 있다는 것을 의미하는데, 이는 마음에는 들지 않지만 가능한 일이다. 그러나 ①의 형식은 하나의 $(B-V)$ 를 가진 표준성에 대해 관측된 색 $(b-v)$ 가 동시에 여러 가지로 나타난다는 것을 의미하는데, 이는 불가능한 일이다.

표준화를 수행한다는 것은 위 식에서 계수의 값을 구한다는 말과 같다. 즉 $a, c, k'_V, k'_B, k''_V, k''_B, zero_V, zero_B$ 를 구하는 것인데, 이러한 값들은 모두 물리적으로 의미 있는 범위에 존재해야 표준화가 제대로 되었다고 할 수 있다. $zero$ 는 얼마가 나오든 전혀 중요하지 않고, 등급에 대한 표준화에서 색항 계수 a 는 0에 가까이($< \pm 0.1$), 색에 대한 표준화에서 색항 계수 c 는 1에 가까이 나와야 한다. 1차 소광 계수는 필터마다 그 값이 다르게 나와야 하는데, U필터에서는 0.4, B에서는 0.2, V에서는 0.1 정도의 값이 나와야 한다. 즉 파장이 긴 필터일수록 그 값이 작아지는 경향을 보여야 하고, 그 값도 대략 저 정도에 위치해야 한다. 2차 소광계수는 0에 가까워야 한다. 그러나 날씨가 정말로 완벽한 경우가 아니면 이 항을 도입하여 결과가 더 안 좋아지는 경우가 많기 때문에 보통은 사용하지 않는다. (사용하지 않으면 구해야 하는 것은 색항, 1차 소광, 영점 계수 3가지인데, 이렇게만 써도 표준화가 개관이 되는 경우도 흔하다.) 또한 fitting error(rms)는 0.05보다 작아야 쓸 만하며, 0.03 이하가 되면 좋다.

필터시스템이 란돌트가 사용한 것과 완벽히 동일하다는 확신이 있다면 색항을 표준화에서 제외해도 좋다. 색항은 필터시스템의 차이에서 도입되는 항이기 때문이다. 가장 단순화된 형태의 표준화는 영점만 조정하는 것인데, 이런 방식을 사용할 수 있는 경우는 survey catalog가 이미 존재하고 있는 영역에 대해 이미지를 재측광한다거나(i.e. SDSS 이미지를 SDSS catalog를 사용해서 표준화 할 때), 동일한 기기를 사용하여 새로 이미지를 찍은 경우(i.e. UKIRT로 측광을 하고 UKIDSS를 이용해서 표준화를 할 때)로 한정된다.

표준화 식은 multi-linear regression method(다중선형회귀방법)로 풀어야한다. 이것이 가능한 프로그램은 IDL등 여러 가지가 있지만 여기에서는 IRAF의 photcal 패키지의 fitparam task를 통해서 표준화 계수를 구하는 방법에 대해서 설명한다.

3) 표준화 계수 구하기

```

ec1> noao
      artdata.      digiphot.      nobsolete.      onedspec.
      astcat.       focas.         nproto.       rv.
      astrometry.   imred.         observatory   surfphot.
      astutil.     mtlocal.       obsutil.      twodspec.

noao> digi
      apphot.      daophot.      photcal.      ptools.

digiphot> phot
      apfile      fitparams      mkcatalog      mknobsfile      obsfile
      chkconfig   invertfit      mkconfig      mkobsfile
      evalfit     mkapfile      mkimsets      mkphotcors
    
```

표준화 계수를 구하기 위해서 필요한 파일은 3가지가 있는데, 첫 번째는 표준성의 정보파일, 두 번째는 표준성의 관측 파일, 그리고 설정 파일이다. 아래는 설정 파일의 예시이다.

```

catalog
V 2
B 4

observation
X 2
v 5
b 3
verr 6
berr 4

transformation
fit k1v=0.1301, a=-0.0537, zv=-0.2476, k2v=0.0193
fit k1b=0.2427, c=-0.1143, zb=0.0622, k2b=0.0288
FITv: v=V+k1v*X+a*(B-V)+zv+k2v*(B-V)*X
FITb: b=B+k1b*X+c*(B-V)+zb+k2b*(B-V)*X
    
```

우선 아래의 transformation에는 ‘우리가 구하고 싶은 계수들의 초기 값’과 Fitting을 수행할 식을 적어둔다. 위에 catalogue와 observtion 항목은 catalogue파일과 observation파일로부터 가지고 올 변수와 해당파일에 변수가 적혀있는 컬럼 번호를 적어준다. 이 때 transformation에 사용된 문자들은 반드시 구하고 싶은 계수, 혹은 외부 파일에서 가지고 오는 변수로 선언이 되어 있어야 한다.

아래는 관측파일(좌측부터 별-대기량-b기기등급-b기기등급오차-v기기등급-v기기등급오차), 카탈로그(좌측부터 별-V표준등급-(B-V)-B표준등급)의 예시이다. 이때 카탈로그, 관측파일은 서로 같은 수의 행으로 구성되어야 한다.

a1	1.000	9.122528	0.000229	8.989002	0.000305
a2	1.000	10.309300	0.000396	9.889766	0.000461
a3	1.000	11.840774	0.000801	10.852436	0.000719
a4	1.000	13.129350	0.001450	11.757645	0.001090
a5	1.000	14.765177	0.003077	12.925320	0.001866
a1	1.300	9.188298	0.000216	9.026200	0.000283
a2	1.300	10.375729	0.000373	9.929123	0.000429
...					

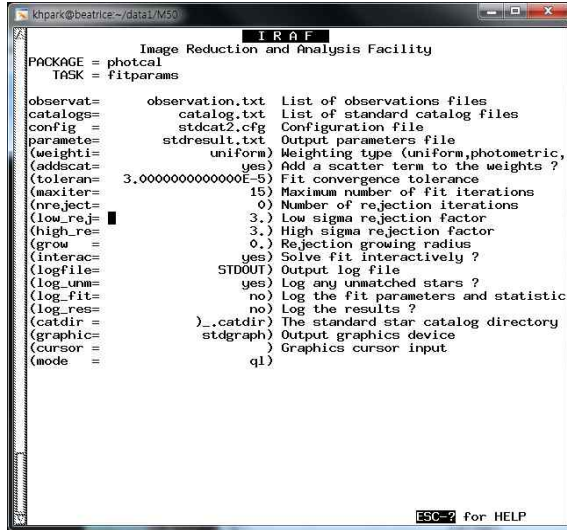
a1	9.100	-0.300	8.800
a2	10.000	0.000	10.000
a3	11.000	0.600	11.600
a4	11.900	1.000	12.900
a5	13.100	1.500	14.600
a1	9.100	-0.300	8.800
a2	10.000	0.000	10.000
...			

위의 세 파일이 준비 되었다면 [그림 A.1]과 같이 fitparams 명령어로 표준화 계수를 구해준다. observat, catalogs, config항에 해당하는 파일 명을 적어 준 후 결과가 인쇄될 파일명을 paramete란에 적어준다.

fitparam을 실행하면 그래프 창이 뜨는데, 이 창을 보며 데이터를 취사 선택하여 표준화 계수를 얻을 수 있다. 이 때 주의해야할 점은 너무 많은 데이터를 제거하면 표준화 결과는 그럴듯하더라도, 실제로는 의미가 없는 값일 가능성이 높아진다.

- d: 근처의 점을 삭제한다.
- u: 근처의 삭제한 점을 되살린다.
- f: 현재까지의 선택을 반영하여 다시 fitting해준다.
- c: 가까운 점의 정보를 타나낸다.
- h,j,k,l: 그래프 축의 내용을 변경한다.
- w: 창을 확대할 수 있다.
 - w,x: x축으로 방향으로 확대
 - w,y: y축으로 방향으로 확대
 - w,z: xy축 모든 방향으로 확대
 - w,a: 다시 원래대로 돌아간다.

만족스럽게 fitting이 되었다면 :error를 입력해서 값을 확인한다. 만족스러운 값이 나왔으면 q 키를 누른 후 저장 여부를 선택한 뒤, next를 선택해서 그 다음 함수 fitting으로 넘어간다. 작업을 마친 후 parameter에 입력했던 파일을 열어보면 표준화 계수가 구해진 것을 볼 수 있다.



[그림 A.1] fitparams로 표준화 계수구하기

4) 표준화 계수를 써서 기기등급을 표준등급으로 바꾸기

표준화 계수를 구한 시점에서 사실상 표준화는 다 끝났다고 봐도 되지만, 우리가 필요한 것은 표준등급이지 표준화 계수가 아니다. ①형식으로 표준화를 했다면 우변에 계수와 기기등급들을 다 넣어서 계산해낼 수 있지만, ②의 경우는 그리 간단하지 않다. 표준화 식은 선형이기 때문에 해석적으로 이를 ①과 같은 형태로 바꿀 수는 있다. 그러면 정확하게 계산할 수 있으나, 이는 권장하지 않는 방법이다. 변환된 식은 매우 길기 때문에, 손으로 적어서 풀고 코드로 옮겨 적는 과정에서 분명히 실수가 개입될 여지가 크다. 따라서 조금은 덜 정확하지만 numerical한 방법을 쓰는 것이 좋을 것이다.

②의 표준화 식을 행렬로 표현하면 다음과 같이 쓸 수 있다.

$$\begin{pmatrix} v \\ b-v \end{pmatrix} = \begin{pmatrix} a & k'_B \\ c & k'_{BV} \end{pmatrix} \begin{pmatrix} B-V \\ X \end{pmatrix} + \begin{pmatrix} V \\ 0 \end{pmatrix} + \begin{pmatrix} zero_V \\ zero_{BV} \end{pmatrix}$$

여기에서, 가운데 들어있는 2×2 행렬의 역행렬을 numerical하게 구해서 좌변에 곱해줄 수 있다. 사실 이 예처럼 2개의 등급만 변환할 때에는 해석적으로 구하는 것이 오히려 편하고 정확하다. 수치적 방법을 동원하는 것은 multi-linear regression에 사용되는 식의 수가 3,4,5로 클 때 유용하다. 이 때 행렬식의 형태는 위와는 조금 달라지겠지만, $m \times n$ 형태의 행렬에서 좌역/우역행렬을 구하는 것은 수치적으로도 쉽지 않은 일이기 때문에 square matrix 형태를 만들어 주어야 역행렬을 구할 수 있다. IDL에서는 invert를 비롯한 여러 내부 함수가 역행렬을 구해준다.

표준화가 끝났으면 드디어 측광에서 귀찮은 전처리가 다 끝난 겁니다. 어때요, 참 쉽죠?

Appendix B. 공간적인 표준화 Wregister

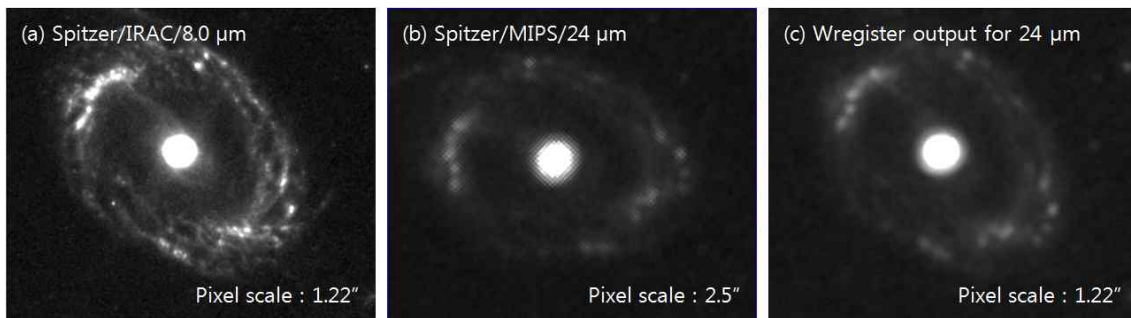
아래와 같은 상황을 가정해보자.

오늘도 새벽까지 연구하는 판넨막내 대학원생 홍근이는 내일아침까지 두 개의 이미지를 측광해서 차이점을 발표하라는 교수님의 엄명을 받았다. 두 개의 이미지는 NGC 1512를 관측한 자료로, 오늘밤 은하의 중심부와 원반부의 플럭스를 알아내야 한다. 그런데 첫 번째 이미지는 픽셀스케일이 1.22초, 두 번째 이미지는 픽셀스케일이 2.5초였다. 픽셀스케일만 다르게 아니다. 화각(Field of View) 상에서 관측된 방향 (Orientation)도 다르다. 자 이걸 어떻게 측광할 것인가?

이런 상황에서 아주 유용하게 쓸 수 있는 IRAF task가 있다. 바로 Wregister! Wregister는 헤더에 기록된 WCS정보를 이용하여(즉, 헤더 정보에 픽셀과 R.A./Dec. 사이의 관계가 계산되어 있어야 한다.) 좌표를 맞춰주는 도구이다. 쉽게 말해 픽셀스케일과 방향각이 다른 두 영상을 하나로 통일해 준다는 것이다. 아래 그림을 보자.

패널 (a)와 (b)는 홍근이가 받은 첫 번째와 두 번째 이미지이다. 한눈에 봐도 함께 다루긴 어려워 보인다. Wregister의 Reference, Input 이미지에 각각 패널 (a) 와 (b)를 넣으면 패널 (c)와 같은 Output 이미지를 얻는다. 픽셀스케일과 방향각이 통일되었다. 픽셀스케일이 달라졌지만 전체 플럭스는 변하지 않았으니 걱정하지 말 것(옵션에서 fluxconserve=yes로 선택할 수 있다).

이미지의 공간적인 표준화를 해냈으니 한결 쉬워졌다. 별지와 원반부의 영역을 선택한 후, 그 좌표를 모든 이미지에 똑같이 적용하면 되니까. Multi-Wavelength자료를 쓸때에는 서로 다른 기기를 이용하여 관측된 이미지를 다루는 경우가 많은데 이때 사용하면 아주 좋다.



[그림 B.1] (a) : Reference image, (b) : Input image, (c) : Output image