N2 - Photometrie eines offenen Sternhaufens (klassisch)

Hinweis: Dieser Artikel schildert die klassische Auswertung des N2-Versuchs bei der Flatfields und Darkframes explizit reduziert sowie alle Verschiebungen von Hand ausgemessen und manuell in eine Routine eingegeben werden. Die moderne Version, bei der die genannten Schritte in einer halbautomatischen Routine erfolgen, wird im Artikel Photometrie eines offenen Sternhaufens beschrieben.

Aufgabe

Photometrieren sie einen offenen Sternhaufen (ggf. auch einen Kugelsternhaufen) und fertigen sie ein Farben-Helligkeits-Diagramm (Hertzsprung-Russell-Diagramm) dieses Haufens an. Ziel der Arbeit ist zusätzlich die Altersbestimmung (Abschätzung) des beobachteten Sternhaufens.

Beobachtung

Der Versuch erfordert eine Nachtbeobachtung am 70cm-Teleskop des AIP oder dem OST der Uni Potsdam. Zur Vorbereitung sollte die Checkliste: Nachtbeobachtung beachtet werden. Am AIP werden in der Dämmerung (oder gegen eine weiße gleichmäßig beleuchtete Wand) Flatfields mit den entsprechenden Filtern, sowie bei geschlossener Abdeckung (also null Belichtungszeit) Bias-Aufnahmen gemacht. Am OST können die Flatfields nach der Beobachtung mit einer Flatfielfolie aufgenommen werden. Bias-Aufnahmen sind nicht nötig, da die Kamera diesen Fehler automatisch korrigiert. Auf jeden Fall sollten außerdem Darkframes aufgenommen werden. Diese werden später zur Datenreduktion benötigt. Bei vollständiger Dunkelheit werden dann Aufnahmen des bzw. der Sternhaufen durchgeführt. Um einerseits eine ausreichende Zahl an Counts zu erhalten und andererseits auch leuchtschwächere Sterne aufzunehmen, sollten die Aufnahmen mehrere Minuten dauern. Bei unzureichender Genauigkeit der Nachführung kann man auch mehrere Aufnahmen (beispielsweise von nur einer Minute Dauer) machen und später aufaddieren.

Vorbereitung der Datenauswertung

Der Versuch kann mit IDL /GDL ausgewertet werden. Eine Installationsanleitung für die freie Variante GDL ist hier zu finden. In dieser Anleitung werden IDL und GDL synonym benutzt.

Ansehen der Bilder

Einloggen im Praktikumspool. Kopieren der Beobachtungsdaten (FITS-Files), inklusive Flatfield- und Bias/Darkframe-Aufnahmen aus dem Verzeichnis `~/daten/<datum>` ins eigene Verzeichnis `~/data_reduction/` . Es gibt verschiedene Tools, um die im FITS-Format abgelegten

zweidimensionalen CCD-Bilder (Images) zu betrachten. Beispielsweise ds9 (neueste Version des früheren saoimage).

```
ds9 filename.fit
```

öffnet das Bild mit ds9. Durch Bewegen des Cursors bei gedrückter rechter Maustaste kann man Helligkeit und Kontrast variieren. Weitere Optionen (Zoom, Falschfarbendarstellungen, Drehen, Spiegeln usw.) sind per Buttons zu erreichen. Die Koordinaten der aktuellen Cursor-Position werden links oben angezeigt. Man kann auch mehrere Bilder gleichzeitig öffnen (Buttons Frame > new frame, dann weiteres File mit File > open öffnen). Mit der Blink-Option (Buttons Frame > blink) kann man mehrere Bilder besonders gut vergleichen.

Alternativ können alle Bilder gleichzeitig geöffnet werden. Dazu wird

```
ds9 *
```

eingeben. In ds9 können die Frames wie oben betrachtet werden oder jeder Frame einzeln hintereinander (Button Frame > Singe Frame). Dabei kann zwischen den Frames durch Drücken der Tabulatortaste umgeschaltet werden.

Es sind die `brauchbaren` Beobachtungen für die weitere Bearbeitung auszuwählen, in denen die Sterne als runde Scheibchen zu sehen sind. Bilder mit ovalförmigen Sternen sind nicht zu verwenden.

Ausmessen der Verschiebung

Da die verschiedenen Aufnahmen wahrscheinlich nicht ganz exakt das gleiche Bildfeld zeigen, ist zunächst (mit ds9) die relative Verschiebung der Aufnahmen untereinander ermitteln. Dazu sucht man auf allen Bildern ein und denselben, markanten Stern (nicht zu hell, und etwa in der Mitte des Gesichtsfeldes) heraus und notiert für jedes Bild seine exakte Pixel-Position (ds9 zeigt oben im Fenster die Position des Maus-Cursors an, nachdem man die Maus auf dem Stern platziert hat, kann man mittels der Richtungstasten auf der Tastatur pixelweise über den Stern gehen, um die Position genau festzulegen). Zur Kontrolle wiederholt man die Prozedur für einen zweiten Stern. Desweiteren ist es für die spätere Arbeit nötig die Halbwertsbreite bzw. FWHM eines typischen Sterns des Sternhaufens zu kennen. Dies kann ebenfalls mit ds9 ausgemessen werden (vielleicht den Mittelwert aus einigen FWHM bilden).

Sternfeld finden

Zur Eichsternidentifikation gibt es mehrere Möglichkeiten:

Am besten (und daher empfohlene Variante) geht das mit der Simbad-Datenbank. Man gibt unter `basic search` den Objekt-Namen (also den des Sternhaufens) ein und wählt unter `Plots and Images` dann den `Aladin Previewer`. Dann muss man das aufgenommene Feld wiedererkennen. Zum Vergleich betrachtet man dazu eine unserer Aufnahmen mit ds9

```
ds9 filename.fit
```

Unter Umständen muss unsere Aufnahme gedrehet und / oder gespiegelt werden.

Mögliche Eichsterne identifizieren

Hat man das Feld identifiziert, suche man sich möglichst 3-4 Vergleichssterne in der Mitte des Feldes, für die B- und V-Helligkeit bekannt sind und notiere deren Helligkeiten für später.

- Dafür kann man z.B. in Simbad nach der `basic search` unter der Option `query around` (with radius...) eine Übersicht aller Objekte im Sternfeld bekommen. Wählt man weiterhin die Option `plot this list of objects`, erscheint eine schematische Karte in der die dort markierten Sterne anklickbar sind. Wenn bekannt, bekommt man dann deren Namen und B- und V -Werte angegeben.
- Einfacher geht das gleiche mit dem `Aladin Java Applet`, das jedoch leicht abstürzt und leider nicht auf allen unserer Maschinen gleich gut brauchbar ist. Mit dem Applet können die Koordinaten ausgegeben werden, um sie anschließend bei `Query and Plot around` anzuklicken. Wer von einem Rechner aus dem Pool oder von zuhause arbeitet, sollte diese Variante zunächst ausprobieren.
- Alternativ (ohne Simbad) vergleiche man die am besten belichtete visuelle Aufnahme mit dem Digitized Sky Survey der betreffenden Gegend und identifiziere den Bildausschnitt. Sodann lässt man sich als Overlay Katalogsterne der wichtigsten Kataloge (Bonner Durchmusterung und Henry Draper (HD) Catalogue) identifizieren (man beachte die Scrollbalken in den entsprechenden Menufenstern!). Hat man glücklicherweise Katalogsterne in dem aufgenommenen Feld, kann man mit Hilfe der Simbad-Datenbank vielleicht deren B- und V-Magnituden in Erfahrung bringen, die dann zur Eichung verwendet werden können.

In jedem Fall sollte darauf geachtet werden, ob es sich bei den Magnituden für die Eichsterne um scheinbare oder absolute Helligkeiten handelt. Ist ersteres der Fall, ist noch die Entfernung des Haufens zur Erde bzw. einem dieser entsprechendes Distanzmodul in Erfahrung zu bringen.

IDL / GDL–Datenreduktion

Es empfiehlt sich, alle folgende Schritte nicht auf der Kommandozeile von IDL bzw. GDL auszuführen, sondern ein kleines Programm name.pro zu schreiben. Dazu einen Texteditor mit der gewünschten Programmdatei öffnen, z.B.:

```
kate name.pro &
```

In diese Datei müssen nun mindestens die folgenden zwei Zeilen geschrieben werden, damit IDL / GDL es als gültiges Programm identifiziert:

```
pro programmname
end
```

Dieses wird nach dem Aufruf von IDL durch den Befehl `idl` mit

```
.rnew name.pro
```

übersetzt. GDL wird analog durch den Befehl `gdl` aufgerufen und Programme mittels

```
.compile name.pro
```

compiliert. So übersetzte Programme können in beiden Fällen dann durch Eingabe des Programmnamens ausgeführt werden:

```
programmname
```

Zwischen den beiden bisher vorhandenen Befehlen in der Datei folgen nun alle Anweisungen zum Einlesen und Bearbeiten der Daten. Es können (und sollten) `help`-Kommandos eingestreut werden, um den Erfolg einzelner Arbeitsschritte zu überprüfen.

Einlesen der Bilder

Die IDL-Funktion `readfits` liest ein FITS-File ein und speichert das Bild in einem Array. Diesen Arrays sollten für die weitere Ansprache in IDL / GDL handliche Namen gegeben werden (z.B. b3 für das dritte Bild im blauen Filter):

```
b3 = float(readfits('N188B1.fit'))
```

wobei offensichtlich der Name des aktuell einzulesenden Bildes (z. B. N188B1.fit) zu verwenden ist. Diese Prozedur wiederholt man mit allen brauchbaren Aufnahmen, sowie den Darkframe- und Flatfield-Daten. Anmerkung: Der Datentyp der Bilder sollte zur Sicherheit in `float` geändert werden, da bei der Addition besonders heller Bilder die Größe des Datentyps `int` nicht ausreicht. Falls die Bilder nicht im selben Verzeichnis wie die pro-Datei liegt, muss der relative Pfad angegeben werden. Liegt das im Beispiel verwendete Bild z.B. im Unterverzeichnis `data/blue`, so lautet der korrekte Aufruf von readfits nun:

```
  b3 = float(readfits('data/blue/N188B1.fit'))
```

Liegen die Bilder nicht in einem Unterverzeichnis des aktuellen Verzeichnisses, kann mittels `../` der Pfad auch auf jeweils auf die nächsthöhere Ebene verweisen. Im Normalfall gibt es nach erfolgreichem Einlesen der FITS-Dateien eine Ausgabe in der Konsole. Während dies am Anfang hilfreich sein kann, wirkt es mit zunehmender Dateizahl oft störend und kann daher durch den Parameter `/SILENT` unterdrückt werden:

```
  b3 = float(readfits('data/blue/N188B1.fit',/SILENT))
```

Abspeichern von FITS-Dateien

Schon bearbeitete Bilder können wieder als FITS-Dateien abgespeichert werden mit dem Kommando

```
writefits, 'bildname.fit', <Bildname in IDL / GDL>
```

```
z.B. writefits, 'blue3.fit', b3
```

Abziehen der Darkframes

Die Darkframe-Bilder zeigen ein gleichmäßiges Rauschen, das bei der Aufnahme durch den Dunkelstrom verursacht wird. Um das Rauschen und thermische Effekte zu minimieren, wird ein Masterdark erstellt. Dazu werden die jeweiligen Darks eingelesen, z.B.

```
dark1 = readfits('pfad/zum/Bild/dark1_60s.fit',/SILENT)
dark2 = readfits('pfad/zum/Bild/dark2_60s.fit',/SILENT)
...
```

und gemittelt

```
masterdark60s = (dark1+dark2+...)/n
```

wobei n die Anzahl der Darkframes ist. Das Ergebnis kann wieder in eine FITS-Datei geschrieben werden:

```
writefits, 'masterdark60s.fit', masterdark60s
```

Die so erstellten Masterdarks müssen von allen Aufnahmen - auch den Flatfields(!) - abgezogen werden. Dabei sind die jeweiligen Belichtungszeiten zu beachten. Als Beispiel:

```
b3  = b3 - masterdark60s
```

Flatfield-Korrektur

Die einzelnen Flatfield-Aufnahmen für jeden Filter werden jetzt (zur Verringerung des Rauschens) aufaddiert, z.B. für den v-Bereich:

```
ffv = ffv1 + ffv2 + ffv3 ...
```

Um im Weiteren bei handlichen Zahlen zu bleiben, empfiehlt sich eine Normierung der Flatfields auf das Maximum (etwa in der Bildmitte):

```
ffv = ffv / max(ffv)
```

Jetzt kann an jedem Bild die Flatfield-Korrektur angebracht werden, wobei natürlich die Blau-Aufnahmen mit dem blauen Flatfield ffb, die visuellen mit dem visuellen Flatfield ffv usw. korrigiert werden, also z.B.:

```
b3ff = b3 / ffb
```

Hinweis: Mit dem writefits-Befehl kann das resultierende Bild abgespeichert und mit ds9 betrachtet werden. Der Helligkeitsabfall zu den Ecken hin sollte nun verschwunden sein, und der Hintergund nicht mehr so hoch.

Speicherplatz freigeben (!)

Beim Einlesen der Bilder wird im Rechner eine beträchtliche Menge Speicher reserviert. Um Probleme zu vermeiden ist es daher nötig, diesen Speicher (auch zwischenzeitlich) wieder freizugeben, wenn die Daten nicht mehr gebraucht werden. Das kann zum Beispiel mit allen Einzelflatfields und -darkframes passieren, wenn sie bereits aufaddiert wurden. Dazu werden die Variablen, die die Daten enthalten einfach auf 0 gesetzt:

```
dark1 = 0
dark2 = 0
...

ffv1 = 0
ffv2 = 0
...

b1 = 0
b2 = 0
...
```

Dieser Schritt gilt nicht nur für die darkframes und flatfields. Auch die ursprünglichen Aufnahmen des Sternhaufens können und sollten nach dem Schneiden (sofern man hierzu eine andere Variable genutzt hat) wieder freigegeben werden. Andernfalls kann es dazu kommen, dass der Rechner “hängen bleibt” und neu gestartet werden muss. Damit gehen dann nicht gespeicherte Daten möglicherweise auch von anderen Gruppen, die gleichzeitig auf dem Rechner arbeiten, verloren.

Übereinanderlegen der Bilder

Schneiden der Bilder

Um die brauchbaren Aufnahmen mit gleichem Filter aufaddieren zu können, und um später die Sterne zwischen den verschiedenen Aufnahmen zu identifizieren, müssen die im 1.Kapitel festgestellten Verschiebungen korrigiert werden. Die linke Seite (x = 0) wird durch dasjenige Bild definiert, in dem der vermessene Stern die kleinste x-Koordinate aufweist (analog für y). Angenommen, dies ist x_7 (auf Aufnahme 7). Dann haben die anderen Bilder i einen x-Versatz von $\Delta x_i = x_{i} - x_{7}$. Der größte auftretende Versatz sei Δx_max. Das Bild habe vorher $N \times M$ Pixel. Klar, dass das Bild dann in x-Richtung auf $N - \Delta x_{\mathrm{max}}$ Pixel verkürzt werden muss. Man beachte, dass in IDL die Indizierung mit 0 beginnt.

Ausführen der Verschiebung bei jedem Bild mit den entsprechenden Δx_i, Δy_i, z. B.:

```
b3cut = b3ff[dx_i:N − dx_max + dx_i, dy_i:M − dy_max + dy_i]
```

also beispielsweise:

```
b3cut = b3ff[13:322, 0:324]
```

Man sollte den Erfolg des Schneidens mit einem help überprüfen. In der Ausgabe der Variablen sollten die geschnittenen Bilder nun Arrays gleicher Größe sein.

Aufaddieren der Bilder

Wenn alles getan ist, kann man die Bilder jedes Filterbereiches für sich aufaddieren, zum Beispiel in b wie blau:

```
badd = b1cut + b2cut + ...
```

Da wir schlussendlich ein Farben-Helligkeitsdiagramm erstellen wollen, muss das Zurechtschneiden der Bilder dazu führen, dass auch das blaue Bild über das visuelle passt. Dazu muss der Versatz der fertigen Bilder (visuell und blau) nochmals überprüft werden und die Bilder ggf. nochmals geschnitten werden.

Nun kann das finale v- und b-Bild z. B. mit

```
writefits, 'vadd.fit', vadd
```

im FITS-Format gespeichert werden. Eines davon wird zur Identifikation der Eichsterne benötigt. Die Bilder sollten ausserhalb IDL mit `ds9` geöffnet werden und scharfe, runde Sterne zeigen. Beim `Blinken` der Bilder sollten die Sterne nicht hin- und herhüpfen.

Identifikation der Sterne

Finden der Sterne

Die Identifikation der Sterne z.B. im Bild `badd` und das Ausmessen ihrer Helligkeit erfolgt in IDL durch:

```
find, badd, xb, yb, fluxb
```

Dabei sind `xb`, `yb` und `fluxb` Namen indizierter Variablen, die hier definiert werden und unter denen später die Resultate angesprochen werden können. Der Name des Bildes und die Indizes müssen für das jeweilige Bild angepasst werden. Es folgen verschiedene Anfragen, mit denen man die Kriterien, nach denen die Sterne identifiziert werden sollen, spezifizieren muss. Zum einen ist dies die Halbwertsbreite, die man sich bereits zu Beginn notiert hat (z.B. 3 Pixel). Den Wert für die Hintergrundintensität kann man jeweils im zuvor aufaddierten und gespeicherten Bild mit *ds9* bestimmen. Für die restlichen Parameter können die default-Werte verwendet werden (einfach mit Enter bestätigen). Diese Prozedur liefert eine Tabelle, in der unter anderem die x- und y-Position der Sterne (xb, yb), sowie deren Fluss (`fluxb`) stehen. Es sollten möglichst viele Sterne, jedoch kein "Dreck" identifiziert werden. Sind zu wenig Sterne gefunden worden, so muss man die find-Prozedur wiederholen und die Identifikationskriterien verändern. Hat man z.B. einen zu großen Intensitätswert für den Hintergrund angegeben, so sind schwächere Sterne nicht identifiziert worden. Bei einer zu großen Angabe für die Standardabweichung können mehrere Sterne als ein einzelner identifiziert worden sein; bei einer zu kleinen Standardabweichung kann es dazu kommen, dass auch "Schmutzeffekte" als Sterne identifiziert werden. Man muss also die Parameter etwas verändern und gegebenenfalls die Prozedur wiederholen. – Das Verfahren wird auf das finale blaue und visuelle Bild

angewandt.

Hat man gute Werte für die find-Parameter gefunden, ist es oftmals lästig diese bei jedem Programmdurchlauf neu eingeben zu müssen. Es bietet sich daher dann an diese mit in den Befehlsaufruf zu schreiben und den Befehl zusätzlich mit der silent-Option zu versehen. Die vollständige Syntax des `find`-Befehls lautet:

```
find, image, x, y, flux, sharpness, roundness, hmin, fwhm, roundlim,
sharplim, PRINT=print, SILENT=silent
```

Dabei sind `sharpness` und `roundness` Arrays, die (genau wie die Sternpositionen selbst) vom `find`-Befehl erst gefüllt werden, erst danach folgen die sonst über die Kommandozeile eingegebenen Parameter. Ein typisches Nutzungsbeispiel lautet:

```
find, badd, xb, yb, fluxb, sharpb, roundb, 300, 4.0, [-2.0,2.0], [0.2,1.0],
/silent
```

In diesem Fall wäre 300 der Schwellwert und 4.0 der FHWM-Wert. Die weiteren Werte können als Intervall eingegeben werden, so daß hier einfach die internen Standardkriterien weiter verwendet wurden.

Hinweis: Es hat sich als günstig erwiesen, vor der *find*-Prozedur am aufaddierten Bild eine Background-Korrektur zu machen. Dabei wird zunächst das Minimum ermittelt:

```
bmin = min(badd)
```

Dieses wird dann mit einer kleinen Verschiebung nach oben abgezogen. Hat man so zum Beispiel für das Minimum einen Wert von 360 bestimmt, dann könnte die Korrektur wie folgt aussehen:

```
badd = badd - 400.
```

In die *find*-Routine kann nun ein kleinerer Wert als Hintergrundintensität (z.B. 200.) angegeben werden. Da es hierbei aber immer Schwankungen geben wird, sollte dieser nicht auf 0 gesetzt werden.

Umrechnung in Magnituden

Da im folgenden in Magnituden gearbeitet wird, müssen die Flüsse entsprechend umgerechnet werden. Man beachte, dass die Magnituden nur bis auf eine additive Konstante bestimmt sind, solange noch keine Eichung durchgeführt wurde.

```
magb = flux2mag(fluxb)
magv = flux2mag(fluxv)
```

Identifikation der Farbbilder

Nun müssen diejenigen Sterne herausgesucht werden, die in beiden Spektralbereichen identifiziert worden sind. Dies geschieht mit `srcor`, wobei die vorher erstellten Tabellen nach gleichen x- und y-Koordinaten durchsucht werden. Als Parameter muss die “Ungenauigkeit” der Koordinatenbestimmung angegeben werden. Im unteren Beispiel wurde eine Ungenauigkeit von 2 Pixeln gew¨ahlt, d. h. wenn die x- bzw. y-Koordinaten eines Sterns im blauen und im visuellen Bild um maximal 2 voneinander abweichen, wird der Stern noch als dasselbe Objekt identifiziert.

```
srcor, xb, yb, xv, yv, 2, ib, iv
```

`ib` und `iv` geben die Indexwerte an, die die Sterne in der mit find erstellten Tabelle f¨ur das blaue bzw. visuelle Bild hatten. Anhand dieser Indizes müssen die Positionen und Magnituden aus den ursprünglichen Tabellen ausgelesen werden. Sind z.B. 226 gemeinsame Sterne gefunden worden, so muss ein Feld (array) vom Typ `float` mit 226 Einträgen angelegt werden, in welches die Magnituden geschrieben werden können.

```
bmag=fltarr(226)
vmag=fltarr(226)
for i=0, 225 do bmag[i]=magb[ib[i]]
for i=0, 225 do vmag[i]=magv[iv[i]]
```

Diese Dimensionierung lässt sich auch mittels Variablen automatisieren. Um die Länge eines Arrays herauszufinden, gibt es den IDL-Befehl `size`, welcher allerdings wiederum ein Array mit verschiedenen Informationen über das übergebene Array liefert. Der vollständige Befehl, um die Anzahl der gemeinsam gefundenen Sterne zu ermitteln und in einer Variable `anz` zu speichern, lautet daher

```
anz = (size(ib,/dimensions))(0)
```

Ob dabei wie hier `ib` oder `iv` genutzt wird, ist natürlich egal, da beide Arrays gleich dimensioniert sein sollten. Statt der Zahl 226 oben kann nun stets anz benutzt werden, bzw. anz-1 für den in den Schleifen benötigten Wert 225.

Vorläufiges Farben-Helligkeits-Diagramm

Nun kann das FHD geplottet werden, also die visuelle Magnitude über der Differenz aus blauer und visueller Magnitude. Das FHD an sich wird mit einem gnuplot-Skript geplottet, für das man die Ergebisse von IDL in eine Datei fhd.dat ausgeben lässt:

```
openw, 1, 'fhd.dat'
for i=0,225 do printf,1,i,bmag[i],vmag[i]
close, 1
```

Die 1 steht für eine frei wählbare Kanalnummer, die dabei eindeutig der zu öffnenden Datei (hier 'fhd.dat') zugeordnet ist. Sollte diese Nummer für Probleme beim aufruf des IDL/GDL-Programms sorgen, ist sie möglicherweise schon anderweitig vergeben und es sollte eine andere Nummer gewählt werden. 225 ist hier die Zahl der gefundenen gemeinsamen Sterne minus 1 (IDL-Zählung!). Um nun das vorläufige FHD zu plotten kopiert man sich aus dem Ordner ~/skripte/n2 die Datei “plot_fhd.plot” in sein aktuelles Arbeitsverzeichnis. Dieses sollte man dann mit dem Editor *kate* öffnen und zum Beispiel den Plot-Bereich für die x- und y-Koordinate, sowie Name der ausgegebenen Datei

und des Plots anpassen. Nach dem Speichern dieser Datei, kann sie von der Konsole aus mit dem Befehl

```
gnuplot plot_fhd.plot
```

ausführen. Dabei sollte man sich auf der Konsole im gleichen Ordner befinden wie die Datei. Bei Erfolg sollte das vorläufige FHD dann in diesem Ornder als Grafik zu finden sein.

Alternative: Man kann sich den Plot auch mit IDL erstellen: Mit yrange bzw. xrange kann ein Plotausschnitt gewählt werden. Dies ist nötig, um die y-Achse umzudrehen (größere Magnitude bedeutet kleinere Helligkeit!). Mit psym kann das Symbol der Datenpunkte bestimmt werden.

```
plot, bmag-vmag, vmag, psym=1, yrange=[-17,-25]
```

Kalibration

Die Helligkeiten (Magnituden) sind bislang nur bis auf eine Verschiebung (additive Konstante) bestimmt. Die Eichung (Kalibration) stellt ein erhebliches Problem dar. Es bieten sich im Prinzip zwei Möglichkeiten an:

1. Man findet auf der Aufnahme einen Stern, dessen b- und v-Magnituden bekannt sind. Dann muss man diesen Stern in den Tabellen indentifizieren und die Werte der gesamten Tabelle so verschieben, dass der identifizierte Stern die Sollwerte erhält.

Für diesen Schritt steht ein *Python*-Skript (plot_stars.py) im Verzeichnis ~/skripte/n2 zur Verfügung. Dieses sollte für die Nutzung in den Ordner kopiert werden, in dem auch das IDL/GDL-Programm zur Auswertung des jeweiligen Sternhaufens liegt. Danach kann man es zum Beispiel mit dem Text-Editor *kate* öffnen und im oberen Bereich anpassen. Hierbei müssen die Pfade zu einem der addierten und gespeicherten Bilder im FITS-Format (z.B. vadd.fit), die Dimensionen dieses Bildes (welche man mit ds9 bestimmen kann) und der Pfad zu einer Datei angegeben werden, in der die Sternpositionen und die jeweiligen Helligkeiten im V- und B-Band angegeben sind. Dies erreicht man im IDL/GDL - Skript, indem man wie im vorigen Abschnitt, diese Größen in eine Datei schreibt. Diese könnte zum Beispiel “stars.dat” heißen und sollte den folgenden Inhalt haben:

```
openw, 2, 'stars.dat'
for i=0,anz do printf,2,i,xv[iv[i]],yv[iv[i]],bmag[i],vmag[i]
close, 2
```

Nach dem Speichern des Python-Skriptes, kann man dieses mit folgendem Befehlt von der Konsole aus ausführen:

```
./plot_stars.py
```

Dabei sollte man sich in der Konsole im gleichen Ordner befinden, in den man das Skript kopiert hat. In diesem Ordner findet sich nun eine Grafik mit dem Namen “starmap.png”. In dieser sind das Sternfeld des aufaddierten Bildes im Negativ, sowie alle identifizierten Sterne als Markierung zu erkennen. Dieses Bild kann man nun mit den anfangs erwähnten Eichsternen vergleichen. Hat man im Bild einen Stern identifiziert, zu dem man in Simbad die Helligkeiten im V- und B-Band feststellen

konnte, so kann man die Nummer der Markierung im Bild “starmap.png” notieren und in der Datei “stars.dat” die entsprechenden Helligkeiten des Sterns in den eigenen Aufnahmen unter dieser Nummer finden.

Aus der Differenz dieser Werte eines Sterns mit den Simbad-Daten ergibt sich die Eichkonstante. Dieser Vorgang wird für 3-4 Sterne wiederholt und anschließend wird der Mittelwert der Eichkonstanten gebildet, wobei die Streuung 0.1 mag nicht übersteigen sollte.

2. Alternativ: Man hat auf separaten Aufnahmen, jedoch in möglichst guter zeitlicher und räumlicher Nähe einen Vergleichsstern aufgenommen, dessen Magnituden bekannt sind. Dann müssen diese Aufnahmen in genau der gleichen Weise ausgewertet werden wie die Sternfeldaufnahmen. Zusätzlich ist in diesem Fall zu beachten, dass die Magnituden auf eine einheitliche Belichtungszeit bezogen werden. Hat man also die aufaddierten visuellen Bilder insgesamt 440 Sekunden belichtet, dividiert man fluxv durch 440 (vor der Umrechnung in Magnituden) und bezieht so den Fluss auf 1 Sekunde (fluxb = fluxb / 440). Das gleiche geschieht mit den Belichtungen des Vergleichssterns. Am Ende wird wieder die Tabelle der Magnituden so um additive Konstanten verändert, dass sich für den Vergleichsstern die Sollwerte für bmag und vmag ergeben.

Nachdem nun die Eichwerte für das FHD bestimmt wurden, kann man diese im IDL/GDL-Programm von den berechneten vorläufigen Magnituden abziehen:

```
magbkal = magb - beich
magvkal = magv - veich
```

Mit diesen Werten könnt ihr bereits ein FHD plotten. Auf der Abszisse steht dort allerdings die Differenz der Magnituden, also ergänzen wir noch die Zeile:

```
magbminusv = magbkal - magvkal
```

Beim Vergleich mit der Literatur wird euch allerdings auffallen, dass die Hauptreihe verschoben scheint. Dieser entsteht durch die interstellare Materie, die sich auch zwischen den Sternen unserer Milchstrasse befindet. Wie alle andere Materie auch, kann sie durch Licht angeregt werden. Diese Energie gibt sie später wieder ab, allerdings nicht auf derselben Wellenlänge, sondern niederenergetischer, also auf der roten Seite des Spektrum. Man spricht daher bei diesem Effekt auch von Rötung, nicht zu verwechseln mit der Rotverschiebung:

$(B-V)_{0} = (B-V) - E_{(B-V)}$

$V_{0} = V - A_{V}$

Die Rötung wird mathematisch durch den Ausdruck $E_{(B-V)}$ beschrieben, den man auch als Farbexzess bezeichnet. Er beschreibt exakt den Unterschied zwischen dem hier gemessenen $(B-V)$ und dem korrigierten, vergleichbaren Wert $(B-V)_{0}$. Die Rötung betrifft allerdings auch die Ordinate, wo der Korrekturterm mit A_{V} bezeichnet wird. Allerdings sind A_{V} und $E_{(B-V)}$ nicht unabhängig voneinander, sondern lassen sich über ein sogenanntes Rötungsgesetz der Form

$A_{V} = R_V \cdot E_{(B-V)}$

beschreiben. Eines der bekanntesten Rötungsgesetze stammt von Seaton (1979) und setzt R_V = 3,1. Damit muss also nur noch der Wert für $E_{(B-V)}$ bekannt sein, um die nötigen Korrekturen komplett durchzuführen, indem man die obigen Formeln in IDL/GDL-Code übersetzt. Entsprechende

Werte findet ihr via Simbad über die mit einem Objekt assoziierten Paper (Veröffentlichungen) oder direkt über Datenbanksuche bei VizieR. Vergesst in jedem Fall nicht, den benutzten Wert und die genaue Quelle (d.h. in der Regel das direkt benutzte oder zu einem VizieR-Eintrag gehörende Paper) in eurem Protokoll anzugeben.

Um nun das endgültige FHD mit dem gnuplot-Skript zu plotten, ist es erneut nötig, die Daten in eine Datei zu schreiben. Dazu kann die bereits verwendete “fhd.dat” überschrieben werden:

```
openw, 2, 'stars.dat'
for i=0,anz do printf,2,i,magbkal[i],magvkal[i],magbminusv[i]
close, 2
```

FHD plotten

Ausgabe mit Gnuplot

Wie oben bereits beschrieben, kann die geänderte Datei “fhd.dat” mit dem gnuplot-Skript wie folgt geplottet werden:

```
gnuplot plot_fhd.plot
```

Dabei sollte man in der Datei “plot_fhd.plot” eventuell noch einmal die x- und y-Bereiche des Plots neu angeben.

Alternative mit IDL

Plotten mit

```
plot, magbkal-magvkal, magvkal, psym=1, yrange=[18,12]
```

Anpassung des y-Ranges und der Beschriftung des Plots durch weitere Parameter möglich, die dem obigen Kommando angefügt werden können, z.B.:

```
title="FHD von NGC 9999", xtitle="B-V in mag", ytitle="V in mag"
```

Wenn man mit dem Diagramm zufrieden ist, kann es in ein ps-file umgewandelt werden.

```
set_plot, 'ps'
device, filename='FHD.ps'
plot, magbkal-magvkal, magvkal, psym=1, yrange=[18,12]
device, /close
```

Um wieder das Plotfenster auf dem Bildschirm zu aktivieren:

```
set_plot, 'x'
```

In einem neuen Fenster kann man sich das ps-file mit *gs FHD.ps* ansehen und über den print-Befehl ausdrucken.

Auswertung

Legen sie dar, wie sich Kugelsternhaufen von offenen Sternhaufen unterscheiden, mit besonderer Betonung auf den Charakter des beobachteten Objektes. Gehen sie darauf ein, wie sich insbesondere Sternhaufen von allgemeinen Sternansammlungen unterscheiden. Erklären sie kurz den Lebensweg eines Stern und gehen sie dabei auf die Darstellung des Hertzsprung- Russell-Diagramms und des Farben-Helligkeits-Diagrammes ein. Erläutern sie die Altersbestimmung eines Sternhaufens mittels Analyse des Abknickpunktes und Isochronen und vergleichen sie ihr FHD mit bekannten Diagrammen. Ermitteln sie das Alter des beobachteten Sternhaufens mittels Abknickpunkt und/oder Vergleich mit Isochronen. Es ist ein praktikumsübliches Protokoll anzufertigen.

Übersicht: Praktikum

From:
https://polaris.astro.physik.uni-potsdam.de/wiki/ - **OST Wiki**

Permanent link:
https://polaris.astro.physik.uni-potsdam.de/wiki/doku.php?id=de:praktikum:photometrie-klassisch

Last update: **2020/03/09 15:46**