

## (H $\alpha$ )RGB – Blending nach Mischa Schirmer

19.02.2012 Gerd Althoff

Als Hobby-Astrofotograf mit besonderer Leidenschaft für H $\alpha$ -Regionen bin ich seit Jahren auf der Suche nach einer wissenschaftlich-technisch fundierten Methode mit der (L-)RGB-Daten und Schmalband-H $\alpha$ -Aufnahmen gemischt werden können ohne die Farben im Sternfeld jenseits der H $\alpha$ -Regionen zu verändern. Fündig geworden bin ich mit Hilfe von Frank Sackenheim, der mich auf eine Methode von Mischa Schirmer, dem Programmierer von Theli, aufmerksam gemacht hat.

Mit Kenntnis dieser Methode ist für mich die Erstellung kalibrierter und normierter (H $\alpha$ )RGB-Bilder jetzt hinreichend beschrieben und mit Hilfe eines Hilfsprogramms (siehe Beschreibung am Ende) direkt in Theli umsetzbar. Die Methode funktioniert nach Augenschein hervorragend und vereinfacht die Datenreduktion und die Erstellung von H $\alpha$ -RGB-Daten, weil nur noch ein Programm benötigt wird: Theli!

Ich zitiere hier zunächst meine Informationsquellen zum (H $\alpha$ )RGB – Blending nach Mischa Schirmer, dann folgen ein paar theoretische Betrachtungen zu den Grenzen der Methode und eine Beschreibung, wie ich diese Methode anwende und die Beschreibung des Programms „Image Calculator“.

### Mischa Schirmer schreibt an Frank Sackenheim (aus: Forum auf Astronomie.de):

*„um die farben "echt" zu erhalten, dient folgender gedanke:*

*-- mein R-Filter hat ca 60nm durchlass, mein h-alpha-filter 6nm.  
-- d.h., bei einem kontinuumstrahler (stern) registriert der halpha-filter 1/10 des lichtes des r-filters.  
-- d.h. bei einem h-alpha strahler registrieren beide 100% des lichtes da sie vollen durchlass bei 656nm haben*

*wenn ich nun also das R-bild um 10% reduziere (\*0.9), und darauf das unmodifizierte ha-bild addiere, dann habe ich:*

*-- in den sternern wieder die urspruengliche 100% helligkeit (0.9+0.1=1)  
-- bei h-alpha-strahlern die 1.9-fache helligkeit (0.9+1.0=1.9)*

*ein faktor 1.9 ist eine starke betonung aller h-alpha-komponenten im bild. anschliessend laesst du mit dem so erzeugten neuen rotkanal wie ueblich die B-V kalibration laufen, etc etc.*

*wenn du moechtest, kannst du h-alpha auch noch staerker betonen:*

*R-kanal \* 0.8 (sternfluss um 20% reduziert), dann muesste ich den h-alphakanal mit 2 multiplizieren um den verlust im stern wettzumachen (0.8+0.2=1)  
ein h-alphastrahler waere dann: 0.8+2.0 = 2.8 mal so hell im vergleich zum originalen rotkanal.“*

### Weiterhin schreibt Mischa Schirmer im Forum zur Frage zur Normierung der R- und H $\alpha$ -Summenbilder auf die gleiche Belichtungszeit (aus: Forum auf Astronomie.de):

*„Wie man R und Ha normiert, ist egal, Hauptsache sie haben die gleiche normierte Belichtungszeit. Wenn du z.B. den Mittelwert oder den Median fuer das Summenbild verwendest, dann koennte man das Summenbild durch die Belichtungszeit der Einzelbilder dividieren. Die Summenbilder haetten dann also eine effektive Belichtungszeit von 1s.*

*Falls du unterschiedliche Belichtungszeiten fuer den gleichen Filter verwendest, musst du rausfinden, wie (und hoffentlich dass) deine Software die Bilder reskaliert.*

*Der Himmelshintergrund sollte sowieso immer abgezogen werden. Ob man das aber vorher auf FITS-Ebene macht, oder nachher in Photoshop (meistens unbewusst, indem man den Hintergrund neutral grau einstellt), ist egal.“*

### Frank Sackenheim schreibt mir, dass Bildberechnungen direkt „in Theli“ mit dem Hilfsprogramm „Image Calculator“ durchgeführt werden kann:

*„Sobald du die coaddition in Theli durchgeführt hast öffnest du create colour picture. Wenn du nun "get coad pictures" drückst entsteht ein Ordner Color Theli. Du öffnest nun eine Konsole und wechselst in diesen*

Ordner. Sobald du in diesem bist gibst du folgendes ein:

```
ic '%1 0.9 * %2 +' r_cropped.fits ha_cropped.fits > rha_cropped.fits
```

Wobei die Dateinamen der Bilder natürlich zu ersetzen sind durch deine eigenen, ebenso die Gewichtung. Wenn du dann das Create Colour Picture Fenster schliesst und erneut öffnest sollte unten auch dein künstlicher Kanal erscheinen. Damit machst du dann eben weiter.

ic steht für Image Calculator und ist ein wichtiger Bestandteil des Theli Pakets. Ich habe Mischa vorgeschlagen doch eine kleine Routine in die GUI ein zu bauen mit der man dann direkt solche Kanäle basteln kann. Mal sehen wann er das realisiert:-)"

## **Berechnung der Multiplikatoren für den Rot- und den H $\alpha$ -Kanal und der Verstärkung des H $\alpha$ -Signals**

Allgemein ausgedrückt ergibt sich für beliebige Durchlassbreiten der verwendeten Filter:

Rotkanal \* x + H $\alpha$ -Kanal \* y = Rotkanal(modifiziert)

wobei x und y die Faktoren sind, mit denen die Kanäle vor der Addition multipliziert werden.

Damit die Farbbilanz des fertigen Bildes nicht verändert wird, muss der modifizierte Rotkanal beim RGB-Komposit den gleichen Beitrag leisten wie der unmodifizierte Rotkanal. x und y sind dafür wie folgt voneinander abhängig:

$$y = (1 - x) * Q \text{ oder auch}$$

$$x = 1 - (y / Q)$$

wobei Q der Quotient der Durchlassbreiten (oder auch Halbwertsbreiten) der verwendeten Filter ist:

$$Q = \Delta\lambda(\text{Rot}) / \Delta\lambda(\text{H}\alpha)$$

Bei einem Rotfilter mit beispielsweise 60nm Halbwertsbreite und einem H $\alpha$ -Filter mit 12nm Halbwertsbreite nimmt Q den Wert 5 an, bei 6nm Halbwertsbreite des H $\alpha$ -Filters entsprechend 10.

Der zulässige Wertebereich für x liegt zwischen 1 und 0, mit den Grenzwerten

x = 1 Rotkanal = Rotkanal(modifiziert) (= keine H $\alpha$ -Beimischung) sowie

x = 0 Rotkanal = H $\alpha$ -Kanal

Im letzteren Fall wird also der Rotkanal vollständig durch den H $\alpha$ -Kanal ersetzt.

Soweit die Betrachtung im Kontinuum für ein unverfälschtes RGB-Komposit. Die Verstärkung V des H $\alpha$ -Signals ergibt sich aus der Addition von x und y zu:

$$V = x + (1 - x) * Q$$

Die maximale Verstärkung ergibt sich, wenn der Rotkanal durch den H $\alpha$ -Kanal ersetzt wird, also für x = 0:

$$V = Q$$

Das bedeutet, je schmalbandiger der H $\alpha$ -Filter ist, um so höher ist die mögliche Verstärkung. Für die oben genannten Beispiele (Rotfilter 60nm Halbwertsbreite) ist also maximal eine 5fache (H $\alpha$ -Filter 12nm Halbwertsbreite) oder eine 10fache (H $\alpha$ -Filter 6nm Halbwertsbreite) Verstärkung erreichbar.

Durch Umstellen der oben genannten Formel ergibt sich:

$$x = (V - Q) / (1 - Q)$$

Daraus errechnen sich beispielsweise für eine Verstärkung des H $\alpha$ -Signals von 3 und einen Q-Wert von 5 die Werte x = 0,5 und (oben eingesetzt) y = 2,5.

## Beispiel

Ein Rot-Filter mit 60nm Durchlass und ein H $\alpha$ -Filter mit 12nm Durchlass ergeben  $Q = 5$ . Beim einem angenommenen idealen Kontinuumstrahler (Stern) gehen also durch den H $\alpha$ -Filter 20% des Lichtes vom Rotfilter. Wird der Rotkanal mit 0.4 multipliziert muss der H $\alpha$ -Kanal vor der Addition mit 3 multipliziert werden, um auf die gleiche Helligkeit zu kommen ( $0.4 + 3 * 0.2 = 1$ ). Das H $\alpha$ -Signal wird bei diesem Beispiel jedoch um  $0.4 + 3 = 3.4$ fach verstärkt.

## Umsetzung in Theli

Im Folgenden wird beschrieben, wie ich diese Prozedur in Theli und mit Hilfe des Hilfsprogramms „Image Calculator“ umsetze. Ich habe einen Satz Rohdaten für RGB und für H $\alpha$ , am besten alle mit Binning 1x1.

### Vergleichsbild ohne Verstärkung des H $\alpha$ -Signals

Nach der Coaddition in Theli gehe ich zunächst für ein Vergleichsbild ohne Verstärkung des H $\alpha$ -Signals ins Menu „Prepare color Picture“ und führe „get coadded images“ durch. Diese Bilder sind grundsätzlich auf 1s Belichtungszeit normiert (siehe Theli Forum). Danach erstelle ich aus allen (H $\alpha$ )RGB-Daten einen künstlichen Luminanzkanal (Mischa Schirmer führt auf seiner Homepage zum Thema Bildbearbeitung (RGB vs LRGB) aus, dass ein [RGB]-RGB-Bild deutlich rauschärmer als ein einfaches RGB-Komposit ist).

Theli bietet die Möglichkeit im Menu „Prepare color pictures“ diesen künstlichen Luminanzkanal zu erzeugen (mit dem Namen „elum\_chisquare“). Für ein Vergleichsbild kann „elum\_chisquare“ mit und ohne H $\alpha$ -Kanal gebildet werden (je nach Geschmack, man sollte sich nur merken was man getan hat). Jetzt führe ich die photometrische Kalibrierung durch und wandele die (E-chi)RGB-Daten zur weiteren Bearbeitung in Tiffs um.

### Verstärkung des H $\alpha$ -Signals durch einmischen der H $\alpha$ -Daten

Ich öffne das Menu „Prepare color pictures“ erneut und führe wieder „get coadded images“ durch. Der vorhandene Ordner „color\_theli“ wird als „color\_theli\_backup ...“ abgelegt und ein neuer Ordner „color\_theli“ automatisch angelegt.

In diesem Ordner sollen die modifizierte Daten abgelegt werden. Dafür öffne ich ein neues Terminal und ergänze den Pfad um das Verzeichnis in dem der Image Calculator liegt (kann global eingestellt werden):

```
PATH=$PATH:~/theli/theli/bin/Linux
```

Ich wechsele in den Ordner „color\_theli“, zum Beispiel:

```
cd 2012/01_17/NGC896/all/color_theli
```

und lasse den Image Calculator für eine 1.8fache Verstärkung des H $\alpha$ -Signals folgende Berechnung durchführen:

```
ic '%1 0.8 * %2 +' Red_cropped.fits H-alpha_cropped.fits > RH18.fits
```

Eine höhere Verstärkung erhält man zum Beispiel durch

```
ic '%1 0.6 * %2 2.0 * +' Red_cropped.fits H-alpha_cropped.fits > RH26.fits  
ic '%1 0.4 * %2 3.0 * +' Red_cropped.fits H-alpha_cropped.fits > RH34.fits  
ic '%1 0.2 * %2 4.0 * +' Red_cropped.fits H-alpha_cropped.fits > RH42.fits
```

Die Verstärkung beträgt bei den oben genannten Beispielen  $0.8 + 1.0 = 1.8$ fach,  $0.6 + 2.0 = 2.6$ fach,  $0.4 + 3.0 = 3.4$ fach oder  $0.2 + 4.0 = 4.2$ fach. Die maximale Verstärkung von 5fach ergibt sich durch

```
ic '%1 5.0 *' H-alpha_cropped.fits > RH50.fits
```

Dann verschiebe ich die entstandene Datei in den Ordner „Coadd Red“ und lösche den Ordner „color\_theli“.

Die Datei „coadd.fits“ im Ordner „coadd\_Red“ benenne ich zunächst in „coadd\_org.fits“ um (nicht löschen) und anschließend die Datei „RHxx.fits“ in „coadd.fits“. Dann starte ich „Prepare color pictures“ in Theli neu und erzeuge einen künstlichen Luminanzkanal (diesmal mit H $\alpha$ ).

Thehi nimmt nun den modifizierten Rotkanal zusammen mit der Wichtung des Original-Rotkanals für die weitere Bearbeitung und photometrische Kalibrierung.

## Programmbeschreibung Image Calculator (ic) aus der Terminalhilfe

### PROGRAMNAME

ic -- image calculator

### SYNOPSIS

ic [options....] rpnexpr fitsfile...

where options are:

- u # print usage
- c N1 N2 # create an image of this size
- p pixtype # specify output pixtype
- s seed # seed rand num generator
- m magicsub # substitute magic value
- h name val # add header comment 'name = val'
- b BSCALE BZERO # add scaling header information

### DESCRIPTION

'ic' does arithmetic on one or more images according to the reverse-polish notation expression 'rpnexpr'. Images are referred to in rpnexpr as '%1', '%2'.... and must all have the same size. If the fitsfilename is given as '-' then that file will be read from stdin. If fitsfilename is given as 'command |' then we read from a pipe executing that command.

'ic' operates on images a line at a time, and so can be used on very large images. A reference to an input image such as '%1' causes a line of that image to be pushed onto a stack. Single argument math functions operate on the line at the top of the stack and multi-argument functions pop lines as necessary and then push the resultant line.

If any of the input images is flagged as MAGIC (as defined 'fits.h') then the result will be MAGIC also (though with '-m' option 'ic' will output 'magicsub' in place of the usual SHRT\_MIN)

Fits header comments are inherited from the first image - other comments are discarded. You may add extra comments with the -h option; see below.

The functions supported include all of the standard C math library (including bessel functions  $j_0(x)$ ,  $j_1(x)$ ,  $j_n(n,x)$ ,  $y_0(x)$ ,  $y_1(x)$ ,  $y_n(n,x)$ ) plus the operators '+', '-', '\*', '/', and 'mult' is provided as a synonym for '\*' to avoid potential problems if you invoke ic from a script. There are the logical operations '>', '<', '>=', '<=', '!=', '==', 'and', 'or' and the negation operator '!'. There is a random number generator 'rand' which generates a uniform random number on the range 0.0-1.0 and 'grand' which generates a zero-mean, unit variance normal variate. There are two functions 'xp', 'yp' to get the horizontal and vertical pixel positions respectively, and two further functions 'x', 'y' which return the position in units of the length of the side of the image. There is a constant MAGIC (defined in magic.h - and currently set to -32768) which is a flag for bad data. There is a function 'if' (a.k.a. '?') which mimics the C language '(c ? t : f)' which returns 't' or 'f' respectively depending on the truth or falseness of the condition 'c'. The rpn syntax for this expression is 't f c'

'?' in which '?' pops the condition 'c' followed by 'f' and then 't' and pushes 't' or 'f' as appropriate. The condition 'c' will of course most likely be a compound logical expression. There are functions 'max' and 'min' which pop two values and pushes the maximum or minimum respectively.

There is also a function 'enter' which duplicates the top value of the stack, and a function 'swap' which unsurprisingly swaps the top two values on the stack.

Use -c option (with no input images) to generate an image from scratch.

Use -p option to specify output pixtype which can be one of

8	# 1-byte unsigned char
16	# 2-byte signed integer
32	# 4-byte signed int
-32	# 4-byte floating point
-64	# 8-byte floating point

Otherwise the output will have same pixtype as that of the first input image, or, with -c option, will have pixtype -32.

Use the -b option to apply scaling of pixel values on output and record the BSCALE, BZERO values in the header. Otherwise the BSCALE, BZERO values (if any) are inherited from the (first) input image. The definition of BSCALE and BZERO is such that the internal values are computed from disk values as:

$$f\_internal = BZERO + BSCALE * f\_disk$$

Output goes to stdout.

The '-h' option can be used to add new header values or to modify existing ones. If an existing name is provided then the existing value will be overwritten with the new value. Otherwise, or if the name is 'HISTORY' or 'COMMENT', the new header line will be appended. The -h option can be given repeatedly to insert a number of comments.

## EXAMPLES

Subtract b.fits from a.fits:

```
ic '%1 %2 -' a.fits b.fits
```

Take sqrt of image to be read from stdin; output as float:

```
ic -p -32 '%1 sqrt' -
```

Create a 512 x 512 image with a linear horizontal ramp and multiply by 10:

```
ic -c 512 512 'x 10 *'
```

Replace all pixels in a.fits with value < 10 with MAGIC:

```
ic '%1 MAGIC %1 10 > ?' a.fits
```

Filter to clip image at fmin = 0.25 fmax = 0.75:

```
ic '%1 0.25 max 0.75 min' -
```

## AUTHOR

Original version:

Nick Kaiser: kaiser@hawaii.edu

THELI version maintained by

Thomas Erben: terben@astro.uni-bonn.de