

Fortgeschritten-Praktikum 1  
Sommersemester 2023

Versuch A003  
Eigenschaften optischer CCDs



Norbert Przybilla  
Institut für Astro- und Teilchenphysik  
E-mail: norbert.przybilla@uibk.ac.at  
Raum: 08/27  
Tel: +43 512 507-52100  
Abhaltung: 08/36

**Titelbild:** Detektorarray der ESO-VST/OmegaCAM bestehend aus 32+4 CCDs  
(<https://www.eso.org/sci/facilities/paranal/instruments/omegacam/inst.html>)

Versuchsbeschreibung: Veronika Schaffenroth, Stefan Kimeswenger, Norbert Przybilla

# Kapitel 1

## Einleitung

In der Astrophysik wird das Licht eines Objektes mit einem Teleskop gesammelt und danach physikalisch untersucht. Dafür muss das gesammelte Licht von einem Detektor aufgezeichnet werden. Erste Untersuchungen wurden noch mit dem bloßen Auge durchgeführt. Darauf basiert auch die in der Astronomie verwendete logarithmische Magnitudenskala, da das Auge eine logarithmische Empfindlichkeit besitzt. Genaue Messungen sind jedoch mit dem Auge nicht möglich. Die Erfindung der Photographie Mitte/Ende des 19. Jahrhunderts erlaubte das erste Mal einen Detektor zu entwickeln, der unabhängig vom Beobachter ist und der es erlaubt, Licht über einen längeren Zeitraum zu integrieren. Außerdem war es erstmals möglich, Messungen aufzuzeichnen und später auszuwerten.

Eine weitere Revolution in der Astrophysik stellte das Charge Coupled Device (CCD) da, das 1969 von W. S. Boyle und G. E. Smith von den Bell Laboratories erfunden wurde, das Patent für das CCD wurde 1974 erteilt. Im Jahr 2009 haben sie dafür den Nobelpreis erhalten. Einen Überblick über die Entwicklungsgeschichte des CCD gibt [McLean \(1997\)](#). Im Gegensatz zur Photoplatte hat es mehrere entscheidende Vorteile. Zum einen ist die Empfindlichkeit, beim CCD Quanteneffizienz genannt, mehr als eine Größenordnung höher als beim Auge oder der Photoplatte. Zum anderen ist ein CCD über einen großen Bereich linear zwischen Signal und aufgefangener Lichtmenge, weshalb es sich vor allem für photometrische<sup>1</sup> Anwendung eignet. Ein weiterer Vorteil ist, dass ein CCD eine digitale Aufnahme liefert, die direkt und mit hoher Genauigkeit mit dem Computer quantitativ ausgewertet werden kann. Aber schlussendlich ist auch ein Vorteil die Reproduzierbarkeit. Während Photomaterial nur einmal belichtet werden konnte, und je nach Behandlung mit wärmeren und kälteren Entwicklerlösungen, Luftstrom, Dicke der Geleeschicht etc. andere Empfindlichkeiten zeigte, kann ein CCD am Ort der Belichtung reproduzierbar geeicht werden. CCDs hatten auch einen durchschlagenden Erfolg in der Photographie, so dass eine Zeit lang fast ausschließlich Digitalkameras mit einem CCD Chip verwendet wurden. Auch in Handys wurden CCDs benutzt. Vor ein paar Jahren wurden CCD Sensoren in Handys durch den CMOS (Complementary metal-oxide-semiconductor) Sensor abgelöst, der billiger zu produzieren und energiesparender ist, dafür aber höheres Bildrauschen besitzt und nicht so stabil bezüglich seiner Reproduzierbarkeit der Eichungen ist wie ein CCD. Für die meisten wissenschaftlichen Zwecke werden daher immer noch fast ausschließlich CCD Sensoren benutzt.

CCDs sind Halbleiterdetektoren, die aus einer zweidimensionalen Anordnung von Siliziumdioden (Pixeln) bestehen. Das Funktionsprinzip beruht auf dem inneren Photoeffekt. Es folgt eine kurze Erinnerung und Einführung zu Halbleitern und dem pn-Übergang, auf dem das Funktionsprinzip eines CCD beruht.

### 1.1 Halbleiter

Aufgrund der räumlichen Nähe vieler Atome und deren Elektronen entstehen in einem Festkörper durch die Überlagerung der einzelnen Energieniveaus der Atome Energiebänder. In Abbildung 1.1 ist schematisch das durch die einzelnen Atomröhre entstehende Potential im so genannten „Bändermodell“

---

<sup>1</sup>photometrisch = absolute quantitative Messung der Strahlung in physikalischen Einheiten

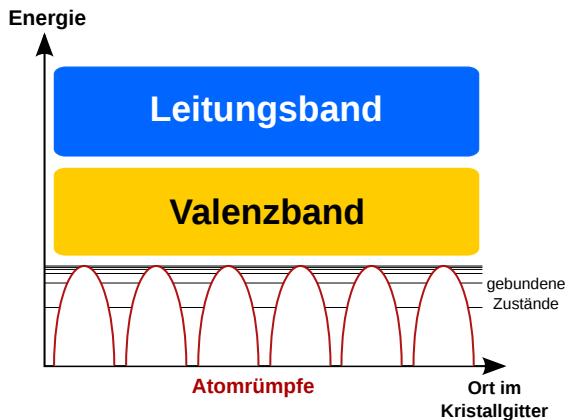


Abbildung 1.1: Zustände im Bändermodell.

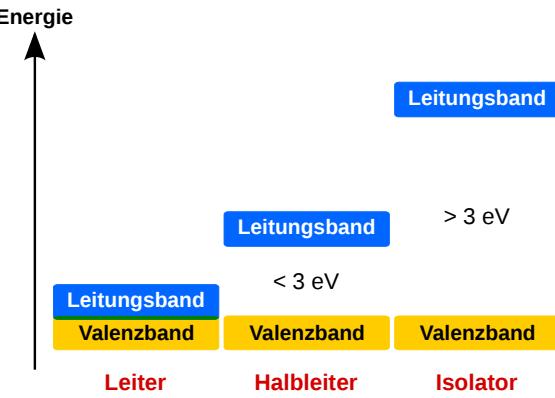


Abbildung 1.2: Leitfähigkeit im Bändermodell.

dargestellt. Energetisch nahe bei den Atomrümpfen sind die gebundenen Zustände. Darüber entsteht durch Überlappung der Potentiale ein quasi-kontinuierliches Band, das Valenzband. Im Valenzband sind alle Energieplätze von Elektronen besetzt. Ein Elektron kann sich nur dann bewegen, wenn ein anderes Elektron die komplementäre Bewegung durchführt. Folglich kann ein solches Band nicht zur Leitung beitragen. Soll ein Elektron zur Leitung beitragen, muss es vom Valenzband ins Leitungsband, das nicht voll besetzt ist, gebracht werden. Im Leitungsband kann es sich nun bewegen. Der Übergang vom Valenz- ins Leitungsband ist nur durch Energiezufuhr möglich.

Im Bändermodell besteht der Unterschied zwischen Isolatoren, Leitern und Halbleitern nur in der Größe der Energielücke zwischen Valenz- und Leitungsband. Abbildung 1.2 veranschaulicht diese Unterschiede. Bei einem Leiter überlappen sich Valenz- und Leitungsband; d.h. bei beliebig kleinen Temperaturen können Elektronen ohne zusätzliche Energiezufuhr vom Valenz- ins Leitungsband gebracht werden. Steigt die Temperatur an, so können aufgrund der thermischen Energiezufuhr noch mehr Elektronen vom Valenz- ins Leitungsband gebracht werden. Dies trägt zur Verbesserung der Leitfähigkeit bei. Auf der anderen Seite stoßen die freien Elektronen aufgrund stärkerer Gitterschwingungen häufiger mit den Atomen zusammen, was zur Abnahme der Beweglichkeit der Elektronen führt und die Leitfähigkeit des Materials reduziert. Der letztgenannte Effekt überwiegt bei höheren Temperaturen. Ein Isolator zeichnet sich durch eine so große Energielücke zwischen den Bändern aus, dass keine Elektronen mehr vom Valenz- ins Leitungsband wechseln können. Als Halbleiter werden Stoffe bezeichnet, die bei  $T = 0 \text{ K}$  als Isolator wirken, aber bei höherer Temperatur zu leiten beginnen. So beträgt die Energielücke zwischen dem Valenz- und Leitungsband beim reinen Halbleiter Silizium 1.115 eV bei 0 K. Bei höheren Temperaturen können nun thermisch angeregte Elektronen vom Valenzband ins Leitungsband vordringen und dort zur Leitung beitragen. Die Dotierung (s. u.) verringert (geplant und gezielt) den Bandabstand in Zonen von Grenzschichten.

**Aufgabe 0:** Diskutieren Sie warum (welche) Elektronen die Hürde von 1.115 eV überwinden können, wo doch die mittlere Energie der Elektronen durch die thermische Bewegung nur  $E_{\text{kin}} = 3/2 k T$  ist.

## 1.2 pn-Übergang

Silizium besitzt in der äußeren Schale vier Elektronen. Wird in die Kristallstruktur ein Element eingebaut (Dotierung), das in der äußeren Schale fünf Elektronen besitzt, z.B. ein Phosphor-Atom, so ist im Gitter ein Elektron überzählig und folglich nur noch schwach an sein Atom gebunden. Es kann jetzt leicht vom Valenz- ins Leitungsband wechseln und dort zur Leitung beitragen, daher wird das Fremdatom als Elektronendonator bezeichnet, während die so zustande kommende Leitung *Elektronen-Leitung* oder *n-Leitung* (negative-Leitung; Abb. 1.3, links) genannt wird. Für die Dotierung ist nur ein sehr kleiner Anteil an Donatoren notwendig – in n-dotiertem Si wird beispielsweise nur jedes  $10^6$ -te Si-Atom ersetzt.

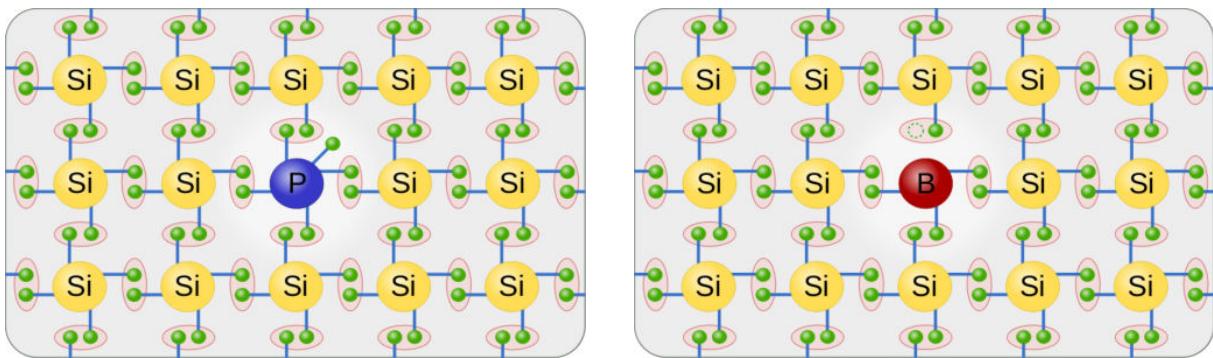


Abbildung 1.3: links: n-Leitung durch Dotierung mit Phosphor; rechts: p-Leitung durch Dotierung mit Bor, jeweils am Beispiel von Silizium.

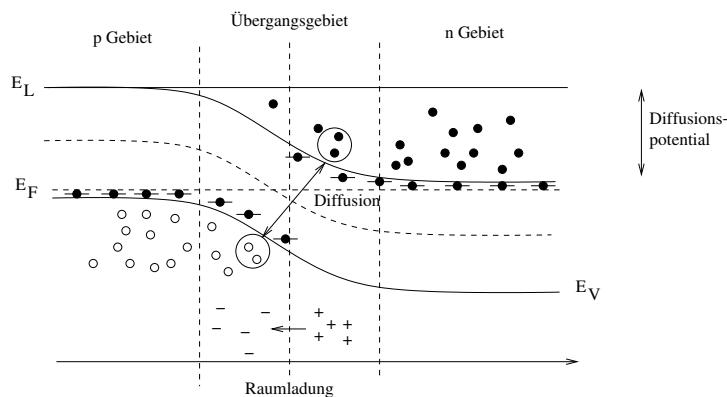


Abbildung 1.4: Schematische Darstellung eines pn-Übergangs.  
 • symbolisiert negative und  
 ○ positive Ladungsträger.  
 $E_V$  bezeichnet die Energie des Valenzbandes,  
 $E_L$  die Energie des Leitungsbandes und  
 $E_F$  die Fermi-Energie.

Wird als Fremdatom jedoch ein Element verwendet, das in der äußeren Schale drei Elektronen hat, wie z.B. Bor, dann fehlt im Gitter ein Elektron. Es ist nun für das Bor-Atom einfach, ein Elektron aus dem Valenzband anzulagern. Folglich entsteht eine Elektronenfehlstelle. Das Fremdatom wird als Elektronenakzeptor bezeichnet und die Leitung als *Defektelektronen-Leitung*, *Löcherleitung* oder *p-Leitung* (positive-Leitung, Abb. 1.3, rechts).

Werden ein p-Halbleiter und ein n-Halbleiter zusammengebracht, entsteht eine typische pn-Übergangszone (Abb. 1.4). In unmittelbarer Nähe der Trennschicht werden die frei beweglichen Elektronen und Löcher aufgrund der hohen Konzentrationsunterschiede zu Diffusionsströmen gezwungen und erhalten dadurch eine andere Verteilung. Elektronen des n-Gebiets diffundieren in das p-Gebiet und füllen dort die Löcher auf. Dadurch wird das p-Gebiet in der Nähe der Trennschicht negativ aufgeladen. Die im n-Gebiet zurückbleibenden Löcher erzeugen dort eine positive Raumladung. Es entsteht ein vom n- ins p-Gebiet gerichtetes elektrisches Feld. Die Umgebung des pn-Übergangs wird als Verarmungszone oder Depletionszone bezeichnet, weil hier keine freien Ladungsträger mehr vorhanden sind: im p-Gebiet sind die Löcher durch Elektronen besetzt, und im n-Gebiet fehlen die ins p-Gebiet hinübergewanderten Elektronen.

Durch Anlegen einer äußeren Spannung kann ein pn-Übergang verschiedene elektronische Funktionen erfüllen, z.B. als Diode: Wird an die p-Schicht der Pluspol und an die n-Schicht der Minuspol angeschlossen, so wird der Übergang in "Durchlassrichtung" betrieben. Das heißt, die Bandverschiebung wird nahezu aufgehoben und das Bauteil leitet elektrischen Strom. Wird hingegen die äußere Spannung entgegengesetzt angelegt, so wird der Übergang in "Sperrrichtung" betrieben. Dabei wird die Potentialdifferenz noch vergrößert und es kann von selbst kein Strom fließen.

Eine weitere wichtige Anwendung des pn-Übergangs bei Halbleitern ist die Photodiode. Hierbei handelt es sich um einen in Sperrrichtung gepolten pn-Übergang. Durch Absorption elektromagnetischer Strahlung werden im Depletionsgebiet Elektron-Loch-Paare erzeugt. Das Gleichgewicht zwischen Driftstrom und Diffusionsstrom wird dadurch gestört, es fließt kurzzeitig ein Strom. Strahlungsabsorption ist nur möglich, wenn die Energie  $h\nu$  der einfallenden Photonen größer ist als der Bandabstand des bestrahlten Materials.

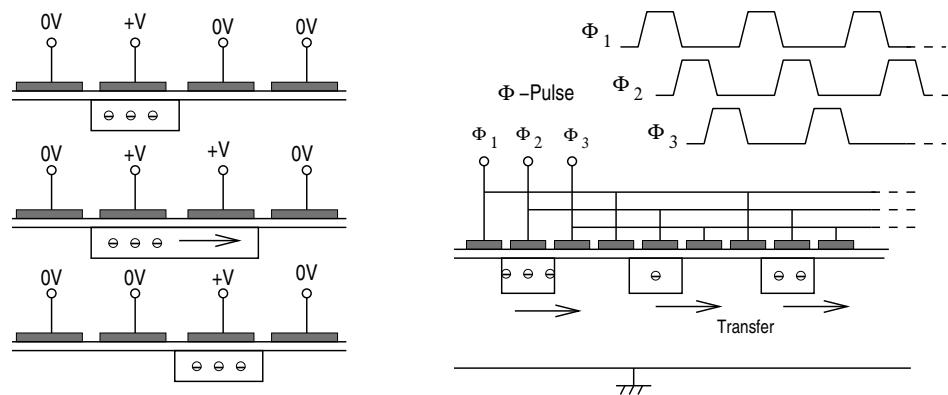


Abbildung 1.5: Schematischer Aufbau und Prinzip des Ausleseverfahrens bei einem dreiphasigen CCD. Die unter den Elektroden angesammelten Ladungen werden durch stufenweise Änderung der an den Pixeln anliegenden Potentiale verschoben (nach McLean, 1997, Fig. 6.9).

### **1.3 Charged Coupled Device (CCD)**

Treffen Photonen also auf einen pn-Übergang, so werden Elektronen freigesetzt, die dann durch eine Ladungsmessung detektiert werden können. Werden nun viele pn-Übergänge in einer zweidimensionalen Struktur kombiniert, so ist es möglich, auch noch den Ort des Einfalls der Photonen zu messen; es entsteht ein abbildender Detektor – ein CCD.

Um eine räumliche Auflösung des CCDs zu erreichen, d.h. um Abbildung zu ermöglichen, muss die an jedem Ort auf dem CCD entstandene Ladung auch dort gesammelt werden. Dies kann z.B. dadurch erreicht werden, dass über das CCD ein Potentialgradient erzeugt wird, in dem die Elektronen an einer der Oberflächen des CCDs driften. An dieser Oberfläche werden die Elektronen dann in Potentialtopfen gesammelt. Jeder dieser Potentialtopfe entspricht einem Pixel. Die Potentialtopfe werden dadurch erzeugt, dass sich für jedes Pixel drei Kontakte auf der CCD-Oberfläche befinden, von denen der Mittlere auf einem leicht anderen Potential gehalten wird. Diese Potentialtopfe definieren die Pixelstruktur in Spaltenrichtung. Zur Erzeugung einer zweidimensionalen Abbildung werden einzelne solcher CCD-Spalten durch sogenannte "Channelstops", Ionenimplantate im CCD-Material, in Zeilen getrennt.

Sind in einem Potentialtopf durch starke Belichtungen sehr viele Elektronen bereits vorhanden, gleichen diese natürlich (teilweise) das elektrische Feld aus. Man spricht vom erreichen der *full well capacity*. Schon kurz vor der Komplettfüllung werden Elektronen, welche durch Photonen entstanden sind, nicht mehr komplett eingefangen. Daher entsteht nahe der Überbelichtung eine Nichtlinearität der Empfindlichkeit. Bei optischen CCDs befindet sich die Pixelstruktur normalerweise auf der dem Teleskop zugewandten Seite. Dies liegt daran, dass bei optischen CCDs aufgrund der hohen Wechselwirkungswahrscheinlichkeit die photoempfindliche Zone in einer sehr geringen Tiefe unter der Oberfläche des Halbleiters liegt. Nachteil dieser Konstruktion ist, dass die zu detektierende Strahlung erst durch die Pixelstruktur mitsamt der Kontaktierung dringen muss. Dies limitiert vor allem die Empfindlichkeit zu hohen Energien hin (UV Licht).

Nach einer gewissen Integrationszeit, während derer die Elektronen in den Potentialöpfen gesammelt werden, wird das CCD ausgelesen. Das dabei verwendete Verfahren ist in Abbildung 1.5 dargestellt. Durch Anlegen verschiedener Spannungen an die Elektroden ( $\phi$ -Pulse) lassen sich die Elektronen in den Potentialöpfen zur Ausleseelektronik hin verschieben. In der Ausleseelektronik werden die Photoelektronen zunächst in einem Vorverstärker verstärkt. Danach werden die Signale in einem Analog-Digital-Konverter digitalisiert und an die Elektronik weitergeleitet.

# Kapitel 2

## Durchführung

### 2.1 Vorbemerkung

Machen Sie sich schon vor der Versuchsdurchführung Gedanken über die theoretischen Fragen, insbesondere die Ermittlung von Rauschen und dem *gain* beschäftigen sich mit Prinzip, Aufbau und Wirkungsweise eines CCD. Planen Sie anhand der folgenden Versuchsbeschreibung die Durchführung des Versuchs bevor Sie mit dem Versuch anfangen. Durch eine effiziente Versuchsdurchführung können Sie sich viel Zeit sparen.

Wie in jedem Laborversuch ist das Führen von Protokollen nötig um später die Auswertungen machen zu können. Sie sollten während des Versuchs die Dateinamen und Typ der Aufnahme notieren. Überlegen Sie sich eine sinnvolle Benennung der Dateien, da man sich so bei der Auswertung viel Zeit sparen kann (mathematische Operatoren wie '+', '-', '/' etc. sind im Dateinamen sind zu vermeiden!). Da das CCD die Dateien im FITS-Format<sup>1</sup> speichert, das nur mit speziellen Programmen lesbar sind, werden diese während der Praktikumsdurchführung mit Munich Image Data Analysis System, MIDAS der Europäischen Südsternwarte ESO ausgewertet. Dazu stehen die Rechner im neuen Computerlehrsaal des Instituts<sup>2</sup> direkt neben dem Raum in welchem Sie messen zur Verfügung, ebenso wie die Linux Maschinen unter ZID Betreuung in den Rechnerräumen am Campus<sup>3</sup>.

Munich Image Data Analysis System, MIDAS: eine Einführung und die wichtigsten Befehle dazu sind in Kapitel C zu finden, zusammen mit einem Beispielskript. Machen Sie sich bei jeder Aufgabe klar welche Messwerte Sie bestimmen müssen. Die Messwerte sollten Sie notieren und diese dann außerhalb der Praktikumszeit auswerten. Teilen Sie sich die Aufgaben untereinander auf. Die benötigte Zeit für die Versuche sind auf die ECTS Arbeitsleistung der Summe von 3-er Gruppen abgestimmt. Im Protokoll sollte als Einleitung eine kurze Einführung zu CCDs und dem dahinter liegenden Prinzip gegeben werden. Außerdem soll auf alle in der Anleitung vorkommenden Fragen eingegangen werden. Dieses bitte auf OLAT hochladen.

---

<sup>1</sup>Flexible Image Transport System FITS – ein ursprünglich aus der Astronomie stammendes, und heute auch viel in der Technik oder auch in der Archivierung von digitalisierten Büchern in Bibliotheken verwendetes, Bildformat. Im Gegensatz zu reinen Bildformaten kann im *header* der Datei der großzügig andere Information mit gespeichert werden. Ebenso bewahrt es nach z.B. Anpassungen von Helligkeit und Kontrast in der Darstellung die ursprünglichen Daten. Eine Eigenschaft welche JPG, PNG, TIFF, GIFF und Co. nicht haben, welche aber für die Reproduzierbarkeit in der Wissenschaft unerlässlich ist. Die Wartung hat heute das NASA GSFC übernommen (<https://fits.gsfc.nasa.gov/>). Die weit verbreiteten Programme Photoshop und Gimp sind wohl in der Lage FITS Dateien zu lesen, wandelnd diese aber in JPEG um und verlieren so komplett die wertvollen Eigenschaften. Daher ist dies nicht empfehlenswert.

<sup>2</sup>Der Betreuer gibt Ihnen Zugangsdaten da hier Ihre ZID Accounts nicht funktionieren

<sup>3</sup>An den ZID Rechnern können Sie Ihr normalen Account csXXnnnn verwenden, stoßen bei unserer Datenmenge jedoch oft auf Hindernisse durch zusätzliche Verwendung von z.B. USB Medien

## 2.2 Theoretische Grundlagen

Die Anzahl der in einem Pixel freigesetzten Elektronen ist nach der Poisson-Statistik verteilt.<sup>4</sup> Der Mittelwert  $N^e$  und die Standardabweichung  $\sigma^e$  der Elektronenzahl hängen somit folgendermaßen zusammen:

$$\sigma^e = \sqrt{N^e} \quad (2.1)$$

Der tatsächliche Messwert  $N^{\text{ADU}}$ , der von einem CCD-Pixel erhalten wird, ist jedoch in (mehr oder weniger willkürlichen) Einheiten (ADU = Analog-Digital-Units) gegeben, die mit dem Gain Faktor  $g$  in die Anzahl der Elektronen pro Pixel umgerechnet werden können:

$$N^e = g \cdot N^{\text{ADU}} \quad (2.2)$$

Neben dem statistischen Fehler wird eine Aufnahme vom Dunkelstrom und Auslese-Bias beeinflusst: Der Dunkelstrom entsteht durch thermische Erzeugung zusätzlicher Elektronen im Chip, der mittlere  $\langle bias \rangle$  ist ein konstanter ADU-Offset, der beim Auslesen des CCDs entsteht. Er kann systematische Strukturen über das Bild hinweg aufweisen. Der Dunkelstrom hat im wesentlichen die selbe Ursache aber zwei physikalisch unterschiedliche Quellregionen. Im reinen Silizium des Chip-Körpers gelten die Energieabstände des reinen Siliziums. Dieser Effekt dominiert bei größerer Kühlung. Bei wärmeren Chips beginnen die dotierten Regionen (welche volumensmäßig einen geringeren Anteil haben) zu dominieren, da durch die "Verunreinigung" der Dotierung die Bandabstände dort verändern<sup>5</sup>.

Das totale Signal des Rohbildes,  $S_{\text{total}}^{\text{ADU}}$ , das in ADU Einheiten umgerechnet wurde ist:

$$S_{\text{total}}^{\text{ADU}} = \frac{N^e}{g} + \frac{N_d^e}{g} + b^{\text{ADU}} \quad (2.3)$$

Dabei ist  $N^e$  die detektierte Photonenzahl in Elektronen,  $N_d^e$  der Dunkelstrom in Elektronen und  $b^{\text{ADU}}$  der Bias in ADUs. Das totale Rauschen im Rohbild,  $\sigma_{\text{total}}^{\text{ADU}}$ , in ADU ist gegeben durch:

$$\sigma_{\text{total}}^{\text{ADU}} = \frac{1}{g} \sqrt{(\sigma^e)^2 + (\sigma_d^e)^2 + (\sigma_{\text{ron}}^e)^2} \quad (2.4)$$

mit  $\sigma^e$  dem Photonerauschen,  $\sigma_d^e$  dem Dunkelstromrauschen und  $\sigma_{\text{ron}}^e$  dem Ausleserauschen (jeweils in Einheiten von Elektronen).

Durch die zusätzliche Aufnahme von **Calibration frames** lassen sich diese systematischen Fehler reduzieren:

**Bias frames** werden mit der kürzest möglichen Belichtungszeit und ohne Lichteinfall aufgenommen (geschlossener Verschluss). Sie enthalten daher neben systematischer Strukturen, da kein Fluss (keine Beleuchtung) und kein Dunkelstrom (Belichtungszeit  $t_{\text{exp}} = 0$  s) vorhanden ist nur das Ausleserauschen aus Gleichung 2.4. Jedoch ist der  $\langle bias \rangle^{\text{ADU}}$  (aus Gleichung 2.3) selbst geringfügig ortsabhängig. Solche Strukturen erkennt man oft erst richtig, wenn man das Ausleserauschen "unterdrückt". Dies geschieht indem man viele (10+) Bias frames mittelt (siehe Abb. 2.1). Daher sind diese systematischen Muster (ohne Ausleserauschen) zu ermitteln und von allen Bildern zu subtrahieren. Subtrahiert man ein solch ermitteltes mittleres  $\langle bias \rangle$  von einem einzelnen Bias frame bleibt das Ausleserauschen selbst übrig. Dabei gilt in der Umwandlung zwischen Einheiten von Elektronen und ADUs näherungsweise<sup>6</sup>

$$\sigma_{\text{ron}}^e = g \cdot \sigma_{\text{ron}}^{\text{ADU}} \quad (2.5)$$

Die Umwandlung zwischen zwei Systemen ganzer Zahlen (Elektronen  $\longleftrightarrow$  ADU) ergibt statistisch ein geringes weiteres Rauschen, welches wir hier aber vernachlässigen können.

<sup>4</sup>Die Poisson-Statistik mit Mittelwert  $\mu$  hat die Verteilung  $P(n) = \exp(-\mu) \cdot \mu^n / n!$ . Somit ist  $\sigma^2 = \sum_n (n - \mu)^2 \cdot P(n) = \mu$ .

<sup>5</sup>siehe z.B. Abb. 5 in Ralf Widenhorn, et al. "Temperature dependence of dark current in a CCD", Proc. SPIE 4669, Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications III, (24 April 2002); <https://doi.org/10.1117/12.463446>. Dieses Journal ist kostenpflichtig, aber Sie finden freie Kopien leicht im Internet mittels des Titels

<sup>6</sup>[http://www.uclick.org/~bolte/AY257/s\\_n.pdf](http://www.uclick.org/~bolte/AY257/s_n.pdf)

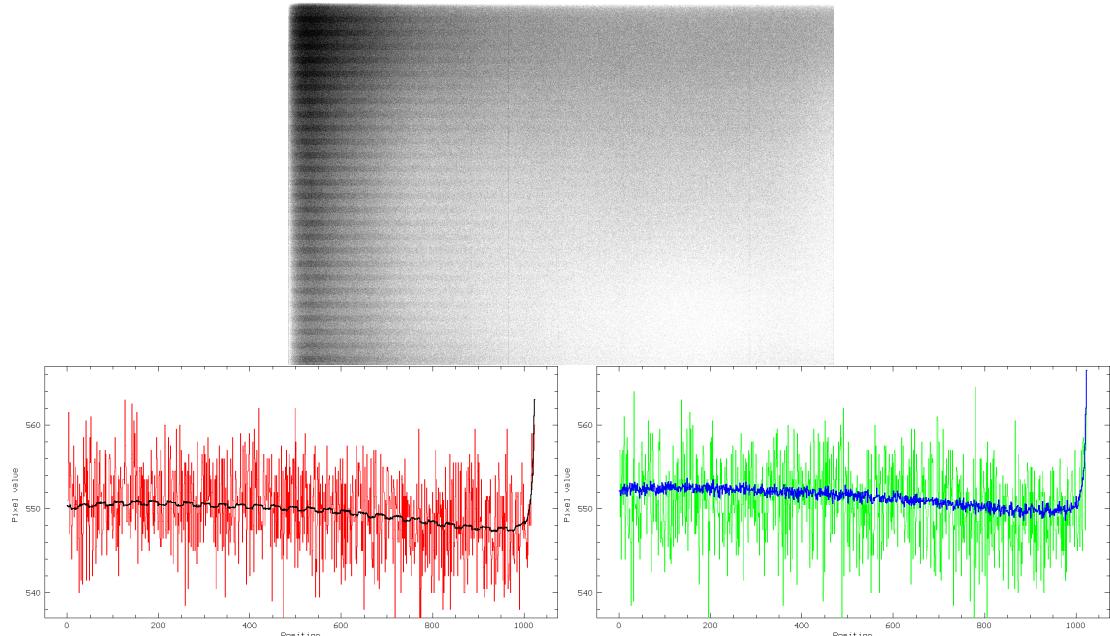


Abbildung 2.1: Der  $\langle bias \rangle$  aus der Mittlung von 50 Bias frames vom 23. Feb. 2020, aufgenommen mit der in unserem Versuch verwendeten Moravian Kamera. Deutlich ist eine streifenartige Struktur zu sehen. Im unteren Plot ist eine einzelne Spalte (links Spalte Nr. 40 und rechts Spalte Nr. 800) gezeigt. Neben den kleinen, relativ zum Ausleserauschen wohl vernachlässigbaren Streifenstrukturen sind generelle globale Verläufe erkennbar.

**Dark frames** werden mit geschlossenem Shutter (Verschluss) aufgenommen. Die Integrationszeit sollte dabei der Belichtungszeit des eigentlichen Bildes entsprechen oder – eine häufiger verwendete Technik – relativ lange sein um ein gutes Signal-zu-Rausch-Verhältnis zu bekommen und hernach auf den Dunkelstrom pro Sekunde normiert werden. Der Dunkelstrom variiert von Bildpunkt zu Bildpunkt erheblich und steigt linear mit der Belichtungszeit ( $N_d^e \propto t_{exp}$ ). Es gibt oft einige echte sogenannte 'hot pixels' welche sehr hohe Werte aufweisen. Je weniger diese sind desto höher ist die Einstufung durch den Hersteller ('science grade' genannt) aber somit auch der Preis bei ansonsten identischen Chips. Bei den längeren Aufnahmen ist außerdem die Wahrscheinlichkeit, dass hochenergetische Teilchen der kosmischen Strahlung den Chip treffen und aufladen erheblich. Daher werden immer mehrere Aufnahmen zur Eliminierung dieser statistisch im Ort verschieden verteilter Ereignisse gemacht (Abb.2.2).

**Flat fields** sind Aufnahmen einer möglichst homogen ausgeleuchteten Fläche. Tatsächlich gelingt dies nicht perfekt. Ebenso hat die verwendete Optik immer Variationen und eventuell Verschmutzungen. All diese Effekte korrigiert man mit den Flat frames ebenfalls. Für die Statistik kann man aber immer nur kleine, durchschnittlich konstant belichtetet Bereiche verwenden. Ein weiteres Problem stellt die ungleichmäßige Empfindlichkeit der einzelnen Pixel dar. Diese Schwankungen sind wohl im klassischen Sinne kein Rauschen da sie herstellungsbedingt sind und immer am selben Ort gleich auftreten. Aber in der Betrachtung der Statistik verhalten sie sich wie Rauschen. Damit erweitert sich die Gleichung 2.4 um einen Term

$$(\sigma_{\text{total}}^{\text{ADU}})^2 = \frac{1}{g^2} \left[ (\sigma_e^e)^2 + (\sigma_d^e)^2 + (\sigma_{\text{ron}}^e)^2 + (\sigma_{\text{FF}}^e)^2 \right] \quad (2.6)$$

Dabei stellt  $\sigma_{\text{FF}}$  die Schwankung der Empfindlichkeit von Bildpunkt zu Bildpunkt dar. Diese setzt sich aus einem "prozentuellen" Wert der Schwankung (welchen man mit einer Konstanten  $k_\lambda$  darstellen kann) und dem Signal durch den Fluss ( $N^{\text{ADU}}$ ) dar<sup>7</sup>. Je höher der Fluss desto höher ist diese Schwankung.

<sup>7</sup>[https://www.mirametrics.com/tech\\_note\\_ccdgain.php](https://www.mirametrics.com/tech_note_ccdgain.php)

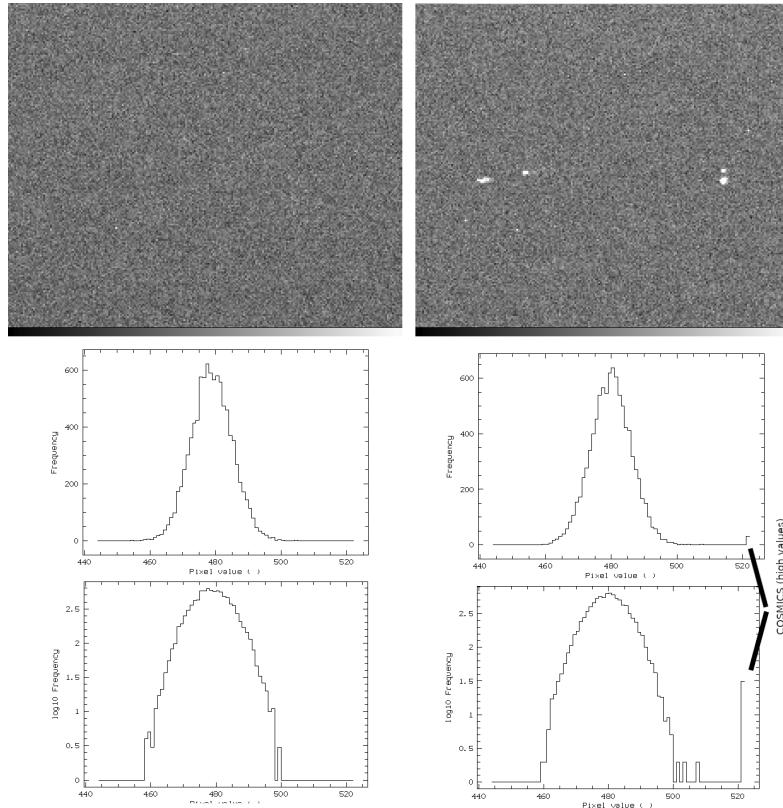


Abbildung 2.2: Ausschnitt aus den Aufnahmen zweier Dark frames welche direkt hintereinander am 27.02.2020 aufgenommen wurden. Der rechte Dark frame wurde von gleich mehreren Cosmics getroffen. Darunter finden sie die Häufigkeitsverteilung (Histogramm) der Daten in seiner linearen und seiner logarithmischen Form. Ein reines Rauschen würde eine Gauss-Verteilung erstellen. Die Cosmics enthalten sehr hohe Belichtungen außerhalb der Statistik.

$$(\sigma_{\text{FF}}^{\text{ADU}})^2 = (k_\lambda \cdot N^{\text{ADU}})^2 \quad (2.7)$$

$$(\sigma_{\text{FF}}^e)^2 = (k_\lambda \cdot N^e)^2 = g^2 \cdot k_\lambda^2 \cdot (N^{\text{ADU}})^2 \quad (2.8)$$

Wie der Index schon erkennen lässt ist diese Konstante abhängig von der Wellenlänge  $\lambda$  (der Farbe) des verwendeten Lichtes. Suchen Sie in der Erläuterung Ihrer Einleitung im Protokoll mittels Literatur (im Netz) den physikalischen Grund für diese Eigenschaft (Zitat/Quelle anfügen nicht vergessen!).

Bias oder Dark frames können von den aufgenommenen Bildern subtrahiert werden. Um vollständig kalibrierte CCD-Aufnahmen zu erhalten, dividiert man schließlich durch die entsprechenden Flat Fields, um die unterschiedliche Sensitivität einzelner Pixel zu berücksichtigen. Bei diesen Berechnungen ist auch die Fortpflanzung der jeweiligen Messfehler zu beachten.

**Aufgabe 1:** Versuchen Sie im Protokoll zu erläutern, warum a) die Staubkörner 6-eckige Strukturen hinterlassen und b) warum diese kleinen Körner verschieden große, aber nicht 100% Licht abschirmende Strukturen schaffen.

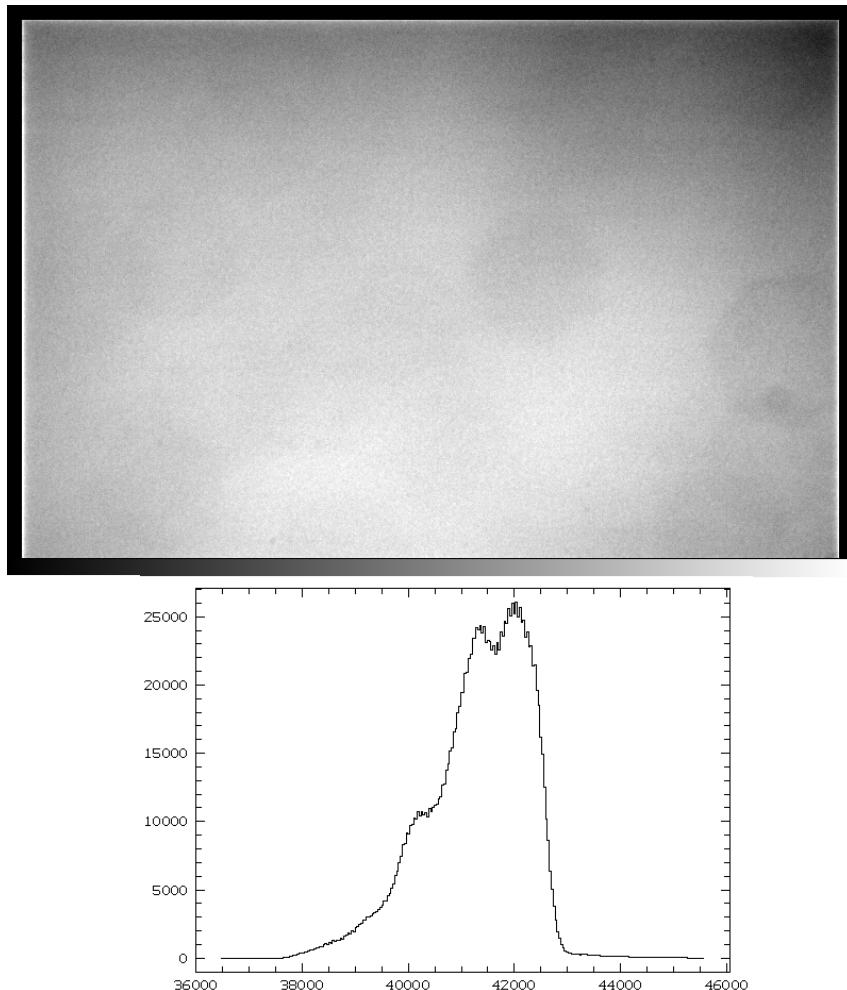


Abbildung 2.3: Flatfield frame vom 25.02.2020 mit  $t_{\text{exp}} = 25$  s. Deutlich sind Schwankungen der Ausleuchtung zu den Ecken hin ebenso wie dunkle Zonen durch Staubkörnern in der Optik zu sehen. Das Histogramm des Gesamtgebiets verhält sich daher nicht wie eine Gauss Verteilung.

## 2.3 Versuchsaufbau

In aufgebautem Zustand sollte die CCD Kamera mit dem Objektiv so aussehen wie in Abb. 2.4. Um den Fokus zu verstellen, drehen Sie am äußersten schwarzen, geriffelten Griff. Die aktuelle Einstellung in Metern ist dabei gegenüber des weißen Strichs abzulesen. Der geriffelte Ring dahinter ist zum Einstellen des Zooms. Der innerste Ring lässt die Blende einstellen. Die Blende beschreibt das Verhältnis von Brennweite zur Größe der Eintrittspupille. Das heißt, je größer die Zahl, desto mehr Strahlen werden vom Rand des Lichtbündels abgeschnitten und desto größer wird die Tiefenschärfe. Dafür wird aber auch die Menge des einfallenden Lichts reduziert. Eine kleine Zahl hingegen bedeutet eine maximal geöffnete Eintrittspupille mit minimaler Tiefenschärfe und maximaler Lichtausbeute. Da die Öffnung der Eintrittspupille quadratisch in die Menge des einfallenden Lichtes eingeht, bedeutet eine durch  $\sqrt{2}$  geteilte Blendenzahl doppelt so viel Licht auf dem CCD. Die eingestellte Blende ist gegenüber des grünen Punktes abzulesen.

Das derzeit verwendete Objektiv ermöglicht die Blenden 4, 5.6, 8, 11, 16 und 22. Wie Sie sehen ergibt dies eine Folge von jeweils diesem Faktor  $\sqrt{2}$ .

Um optimale Ergebnisse für die verschiedenen Aufgaben zu erhalten, sollten Sie Schärfe, Zoom und Blende justieren, bis Sie gute Einstellungen gefunden haben und diese dann entsprechend beibehalten. Dazu sollte die Blende so gewählt werden, dass die Überbelichtung<sup>8</sup> bei Belichtungszeiten von 20 bis

<sup>8</sup>die “digitale Sättigung” liegt bei  $(2^{16} - 1 = 65\,535)$  ADU



Abbildung 2.4: CCD-Kamera mit Ansatz für das Objektiv (links), die Flatfield Lampe (mitte) und die komplett aufgebaute Anlage (rechts).

35 s liegt. Beginnen Sie, um Zeit zu sparen, mit einer relativ kurzen Belichtungszeit und nutzen Sie die Linearität des CCDs und die oben angegebene Zusammenhänge mit der Blende um dann nach einer ersten Testaufnahme durch Rechnung schnell ans Ziel zu kommen. Dies gilt jeweils für den blauen und den grünen Filter unterschiedlich da diese verschiedenen Durchlässigkeiten haben.

Sie Software des Kameraherstellers zum Lesen der Daten wird SIPS genannt. Eine Kurzanleitung dazu ist im Kapitel A gegeben. Wir beginnen mit der Minimaltemperatur von  $-20^{\circ}\text{C}$ . Achten Sie darauf, dass sich diese immer vollständig "einpendelt" hat nachdem Sie Änderungen der Parameter vorgenommen haben. Sie können jedoch während die Kühlung läuft, die Einstellungen zum optimalen Fokus, Zoom und Blende bereits durchführen, da sie diese Testbilder sowieso verwerfen werden. Sie sollten aber keine Bilder welche mit nicht-stabiler Kühlung gewonnen wurden für die Analyse selbst verwenden.

## 2.4 Bias frames

Beginnen Sie das Experiment mit dem Aufbau und schalten Sie die Kühlung auf die minimale Temperatur von  $-20^{\circ}\text{C}$  (bei normalen Bedingungen im Labor von  $20\text{--}25^{\circ}\text{C}$  – sollten wir im Sommer wesentlich höhere Außentemperaturen bekommen und so diese Temperatur nicht erreichbar erscheint, stellen Sie den Versuch auf  $-15^{\circ}\text{C}$  um).

**Aufgabe 2:** Nachdem die Minimaltemperatur stabil erreicht ist messen Sie eine Serie von mindestens 23 Bias frames (die ersten 3 Aufnahmen ignorieren Sie, da der Bias erst nach ein paar Aufnahmen stabile Werte annimmt). Bestimmen Sie daraus das Ausleserauschen  $\sigma_{\text{ron}}^{\text{ADU}}$  und bestimmen Sie mittels Histogrammen sowohl den mittleren  $\langle \text{bias} \rangle$  selbst wie auch das Ausleserausches allein aus den statistischen Eigenschaften. Schließen Sie dazu immer die äußersten 50 Zeilen und Spalten des CCDs aus.

## 2.5 Bestimmung des Gain Faktors $g$ und der Linearität des CCDs

Der Gain Faktor kann aus einer Flat field Aufnahme abgeleitet werden: Bei einer solchen haben viele Pixel annähernd den gleichen Wert  $N^{\text{ADU}}$ ; damit kann auch dessen statistische Streuung  $\sigma_{\text{total}}^{\text{ADU}}$  ermittelt werden. Wie zuvor gezeigt sind dabei systematische Fehler zu beachten. Verwenden Sie daher nur ein Gebiet von etwa  $200 \times 200$  Pixeln nicht zu nahe zum Rand, wo Sie eine homogene Ausleuchtung (ohne Staubstrukturen) sehen. Mittels des Histogramms (vor allem der logarithmischen Form) können Sie dies prüfen (bitte Histogramm ins Protokoll einfügen und erläutern). Im Protokoll muss unbedingt erwähnt werden, welches Gebiet verwendet worden ist.<sup>9</sup>

Verwenden Sie dazu die ans stabilisierte Netzteil angeschlossene Flatfieldfolie (kreisrunde Lichtquelle). Vorsicht - den Wert dort NICHT über 12 V erhöhen. Normalerweise sollten die voreingestellten 11.5 V verwendet werden.

---

<sup>9</sup>Stichwort: Reproduzierbarkeit in der Wissenschaft.

**Aufgabe 3:** Wie kann  $g$  aus den Messgrößen  $\sigma_{\text{ADU}}^{(\text{stat})}$  und  $N_{\text{ADU}}$  mit Hilfe statistischer Beziehungen berechnet werden? Nutzen Sie dazu die Definition des Gains (Gleichung 2.2) als Ausgangspunkt. Nehmen Sie ein Flatfield mit einer Belichtungszeit von etwa 1 s auf und ziehen einen  $\langle bias \rangle$  ab. Verwenden Sie dazu sowohl den grünen wie auch den blauen Farbfilter.

Diese einfache Bestimmung des Gain ist nur richtig, wenn es keine anderen Rauschquellen auf dem CCD gibt, so dass die Anzahl der Photonen exakt bei der Photonenstatistik gegeben ist, der Dunkelstrom vernachlässigbar und außerdem das Flatfield perfekt gleichmäßig ist. Da dass wie wir bereits wissen in der Realität nicht gegeben ist, ist das keine genaue Methode den Gain zu bestimmen. Eine bessere Möglichkeit ist es Flatfields mit verschiedenen Signalstärke zu benutzen.

**Aufgabe 4:** Bevor Sie mit diesem Versuch beginnen, sollten Sie erst die geeignete Einstellung des Zooms, Fokus und der Blende wie oben beschreiben für die jeweilige Folie (blau und grün) bestimmen. Nun bestimmen Sie eine Serie mit einer Farbfolie ohne irgendwelche Bewegungen am System mit 0.5, 1, 2, 3, 4, 5, 6, 7.5, 10, 15, 20 s, ... und wechseln etwa 10 s VOR der von Ihnen zuvor getesteten Überbelichtung wieder in den 2.5 und 1 s Takt. Danach bestimmen Sie analog die Serie für die zweite Farbe.

Zwischen den Serien nehmen Sie bitte jeweils 8 Bias frames (die ersten 3 bitte wieder ignorieren) zur Kontrolle der Stabilität des Systems auf. Ziehen Sie von den Bildern zuerst das zuvor gewonnene Bild des mittleren  $\langle bias \rangle$  ab (siehe Abbildung 2.1).

Messen Sie in einem kleinen Gebiet konstanter Beleuchtung (wie oben – in allen Belichtungen das identische Gebiet!) das Totalrauschen  $\sigma_{\text{total}}^{\text{ADU}}$  und das Signal  $N_{\text{mean}}^{\text{ADU}}$ .

Bestimmen Sie die Linearität des Systems indem Sie den Werte  $N_{\text{mean}}^{\text{ADU}} / t_{\text{exp}}$  gegen die Belichtungszeit  $t_{\text{exp}}$  auftragen. Im mittleren Teil des Diagramms erhalten Sie einen horizontalen Bereich. Bestimmen Sie dort einen Mittelwert  $N_{\text{mean}}^{\text{ADU}}$ . Wiederholen Sie letzten Schritt mit der Y-Achse  $N_{\text{mean}}^{\text{ADU}} / (t_{\text{exp}} \cdot N_{\text{mean}}^{\text{ADU}})$  für den endgültigen Graphen. Es gilt dann für den linearen Bereich

$$\frac{N_{\text{mean}}^{\text{ADU}}}{t_{\text{exp}} \cdot N_{\text{mean}}^{\text{ADU}}} = a_0 + a_1 \cdot t_{\text{exp}} \quad (2.9)$$

$$a_0 \approx 1.0 \quad (2.10)$$

$$a_1 \approx 0.0 \quad (2.11)$$

**Frage:** Warum dürfen wir den Dunkelstrom bei dieser Untersuchung ignorieren (Sie benötigen dazu das Ergebnis aus dem letzten Versuchsteil). Diskutieren Sie das Ergebnis.

Bisher “arbeiteten” wir in Einheiten von ADU. Nunmehr wollen wir von diesen willkürlichen Einheiten auf die physikalischen Einheiten der Elektronen wechseln.

**Aufgabe 5:** Bestimmen Sie mittels Gleichung 2.6 den Gain und die Werte für  $k_\lambda$  indem Sie das gemessene Totalrauschen  $(\sigma_{\text{total}}^{\text{ADU}})^2$  gegen das Signal  $N_{\text{mean}}^{\text{ADU}}$  auftragen. Dies ergibt eine nach oben offene Parabel<sup>10</sup>. Als Nebenergebnis erhalten Sie auch noch eine verbesserte Angabe für den Oberrand des linearen Bereiches. Verwenden Sie für den Fit nur die Werte unter jener Grenze.

Fitten Sie einmal die Parabel mit den 3 freien Parametern (klassisch:  $y = a_0 + a_1 \cdot x + a_2 \cdot x^2$ ) und einmal mit festgelegtem Achssabschnitt  $a_0$  welchen sie über das Ausleserauschen  $\sigma_{\text{ron}}^{\text{ADU}}$  aus Aufgabe 2 festlegen. Bestimmen Sie weiter mittels dem verbesserten Wert für das Ende des linearen Bereichs von hier die maximale Elektronenanzahl (full well capacity) des Chips und vergleichen Sie es mit Datenblättern der Hersteller (Chip ist ein KAF1603ME).

**Frage:** Wie bestimmen wir den  $\langle bias \rangle$  Wert mittels dieser Messungen ohne zuvor Bias frame Serien zu ermitteln? Warum verwenden wir diese Technik normalerweise nicht?<sup>11</sup>

**Frage:** Der untersten Teil der Kurve aus Aufgabe 4 weicht ebenfalls von der Linearität ab. Dieser Effekt ist von der Physik des CCDs her nicht zu verstehen. Er variiert auch sicher stark von Gruppe zu Gruppe. Was könnte die Ursache sein und wie können Sie dies in der Auswertung in der Formel 2.9 einfach korrigieren.<sup>12</sup>

**Aufgabe 6:** Bestimmen Sie das Ausleserauschen in Elektroneneinheiten

<sup>10</sup> siehe flatfield-errors in Abb. 30 von Jansen, 2006 – achten Sie darauf, dass der Autor andere Achsen wie wir hier verwendet!

<sup>11</sup> Hinweis: Fig. 24 in Jansen, 2006

<sup>12</sup> Hinweis: Ein mechanischer Effekt in der Anlage.

## 2.6 Blooming

Werden Aufnahmen überbelichtet, kann es zu zwei interessanten Effekten kommen, dem “Blooming” und dem “Smear”. Der “Smear” ist nur relevant wenn man keinen Shutter besitzt, da der CCD noch während des Auslesens belichtet wird und kann daher von unserem CCD nicht demonstriert werden.

Benutzen Sie für diese Aufgabe die Glühlampe und schließen die Blende des Objektivs so weit wie möglich. Stellen Sie das Bild scharf. Schatten Sie die Bereiche, welche die Lampe außerhalb der Blende beleuchtet weitestgehend ab um nicht dort später Überbelichtungen zu erhalten. Beginnen Sie dabei mit einem Bild, dass nicht überbelichtet ist und erhöhen Sie dann die Belichtungszeit jeweils um das Doppelte (e.g. 0.1, 0.2, 0.4, 0.8 .... Sekunden) um den Effekt zu demonstrieren.

**Aufgabe 7:** Erklären Sie den Blooming Effekt und visualisieren Sie ihn anhand einer Aufnahme der Glühlampe mit und ohne Blooming.

## 2.7 Temperaturabhängigkeit des Dunkelstroms

Wie schon erwähnt, werden Ladungen thermisch angeregt durch die Temperatur des umgebenen Mediums. Dies kann im CCD ein Signal erzeugen, den sogenannten Dunkelstrom. Die theoretische Abhängigkeit des Dunkelstroms kann von der Fermi Statistik abgeleitet werden und hat die folgende Form, wobei  $E_g$  die jeweilige Bandlücke ist<sup>13</sup>:

$$I = \text{const} \cdot T^{\frac{3}{2}} e^{-\frac{E_g}{2k_B T}} \quad (2.12)$$

**Aufgabe 8:** Nehmen Sie bei jeder Temperatur 8 Bias frames (da der  $\langle bias \rangle$  temperaturabhängig ist; die ersten 3 bitte wieder ignorieren) und mindestens 3 Dark frames für Temperaturen im Bereich von der ja zuvor schon eingestellten maximalen Kühlung der Kamera bis zur Raumtemperatur (in Schritten zu  $8^\circ\text{C}$ ). Zu Empfehlen ist hierbei eine Belichtungszeit von 180 Sekunden für  $T < -10^\circ\text{C}$ , 120 Sekunden für  $T < +10^\circ\text{C}$  und 60 Sekunden für die höheren Temperaturen. Beachten Sie, dass sich nicht nur die Temperatur erst auf einen konstanten Wert einpendeln muss, sondern der ganze elektronische Regelkreis (siehe Abbildung A.1), bevor Sie mit der Messung beginnen. Bestimmen Sie die Bandlücke durch das Fitten einer geeigneten Funktion in einem geeigneten Plot unter Berücksichtigung der Fehler. Wie verhält sich die Bandlücke im Vergleich zum theoretischen Wert von Silizium?

**Tipp:** Berücksichtigen Sie die Form der theoretischen Temperaturabhängigkeit des Dunkelstroms. Achten sie darauf den Dunkelstrom in einer geeigneten Einheit anzugeben, Sie benötigen ebenfalls das Ergebnis der *gain* Aufgabe! Da sich der Ursprung des Dunkelstroms mit der Temperatur ändert, müssen Sie das Ergebnis in einem geeigneten Diagramm mit zwei Geraden unterschiedlicher Steigung (d.h. unterschiedlicher Bandlücke) nähern.

---

<sup>13</sup>In der Literatur finden Sie oft die vereinfachte Form/Näherung, welche nur den Exponentialterm enthält.

## 2.8 Ablauf

Treffpunkt ist 14:00 Uhr das Büro 08/27 (nordwestliche Ecke des Viktor-Franz-Hess-Hauses), der Versuch findet im Raum 08/36 statt. Nach der Aktivierung der Kühlung findet die Vorbesprechung statt, bis die Ziel-Temperatur des CCDs erreicht wird. Bitte schalten Sie die Kamera nie aus, wenn die Temperatur des CCD unter dem Gefrierpunkt liegt. Dies kann sonst zu teuren und im Rest des Semesters kaum mehr behebbaren Schäden am System führen.

Legen Sie am Desktop ein eigenes Verzeichnis mit dem Gruppennamen für Ihre Daten an. Die Messungen sollten in der Reihung wie sie in diesem Skriptum angegeben sind in etwa 2 h durchzuführen sein. Arbeiten Sie konzentriert und dennoch zügig. Das Praktikum endet im Normalfall pünktlich.

Arbeiten Sie im Dunklen. Schalten Sie während der Aufnahmen die Tastaturbeleuchtung ab und reduzieren Sie die Bildschirmhelligkeit auf ein Minimum (im dunklen Raum ist das immer noch sehr hell). Drehen Sie den Bildschirm schräg, damit er nicht direkt auf die Kamera leuchtet. Diese CCDs sind auf die Detektion geringer Lichtmengen ausgelegt. Geringes Streulicht kann Ihre Messung schädigen.

Legen Sie beim FLATfield die Farbfolien (grün und blau) so zwischen Licht und Optik, dass sie die ganze Optik bedecken. Verknittern Sie die Farbfolien nicht - sie machen Ihre Messungen sonst unzuverlässig. Überwachen Sie mittels der Histogrammdarstellung in SIPS Ihre Daten bei den Aufnahmen. Bei Zweifel öffnen Sie die Datei direkt in Ihrem Datenordner mit dem NASA FITSview tool (siehe Anhang B). In sehr seltenen Fällen kann es zu Hardwarefehlern kommen. Informieren Sie dann umgehend den Betreuer.

Die Auswertungen können Sie sowohl an den Rechnern im daneben befindlichen Rechnerraum des Instituts (Zugangsdaten gibt Ihnen der Betreuer) erfolgen, als auch an den Rechnern in den Rechnerräumen am ZID. Für die Auswertung müssen Sie die Daten auf ein LINUX System übertragen. Dies geschieht am Einfachsten mittels des Windows Programms WinSCP direkt über das Netzwerk. Entscheiden Sie sich für die ZID Rechner ist es am einfachsten sich mit zid-gpl.uibk.ac.at und Ihrer ZID Studentenkennung zu verbinden. Beachten Sie dass der Versuch etwa 600 - 700 MByte an Daten produziert. Dies ist oft zu viel für Ihren UNI Account. Wir haben auch keine Rechte dort, Software-Installationen zu warten. Für Softwareprobleme ist am ZID Herr Mag. Wolfgang Jais zu kontaktieren.

Daher empfehlen wir die Rechner in unserem Rechnerraum. Verbinden Sie sich mit einem der Rechner pc39-c706.uibk.ac.at bis pc51-c706.uibk.ac.at. Es ist egal welchen Sie nutzen, da die \$HOME Verzeichnisse vereint sind. Diese Rechner sind via ssh -X pcNN-c706 vom UNI-Netz aus erreichbar (nutzen Sie VPN von zu Hause). Von zu Hause aus müssen Sie erst über VPN eine Verbindung ins UNI-Netz aufbauen, Details dazu unter

<https://www.uibk.ac.at/zid/netz-komm/vpn/>

Sie können Sie sowohl remote auf diesen arbeiten, als auch Ihre Daten dorthin transferieren. Auf unseren Rechnern können Sie Softwareunterstützung durch uns bekommen (was normalerweise nicht nötig sein wird, da die relevante Software existiert und ständig nach jedem Update des Betriebssystems getestet wird). Weiter Information unter

<http://www.uibk.ac.at/astro/intranet/rechnerraum>

Wir empfehlen die Auswertung mit MIDAS (Anhang C). Es steht Ihnen aber frei jedes Ihrer Tools zu verwenden. Sicherlich ist die Bearbeitung z.B. unter AstroPython möglich. Sollten Sie aber nicht bereits sehr erfahren in der Nutzung dieses Systems sein, werden Sie wahrscheinlich zur Programmierung der Tools, welche MIDAS schon fertig anbietet, viel Zeit verwenden müssen. Die Aufgabe wurde bezüglich des Workloads (ECTS) so erstellt, dass eine solche Entwicklungszeit nicht eingerechnet ist, da sie nicht die Zielsetzung des Lernzieles dieses Versuchs darstellt. Zur Anpassung von Funktionen und Erzeugung Graphen verwenden Sie am besten den Spreadsheets Ihrer Präferenz. Hier installiert ist Open/LibreOffice. Beginnen Sie mit Auswertungen nur, wenn Sie absehen können, dass sie mit der Serie der Messungen ansonsten zeitlich zu Rande kommen. Dazu ist sicher auch die Organisation innerhalb der Gruppe relevant. Löschen Sie Ihre Daten auf dem (Windows) Versuchsrechner nicht, da ansonsten etwaige Fehler und Irrtümer für den Betreuer bei der Bewertung nicht nachvollziehbar sind.

Bei Zitaten im Protokoll erwarten wir die Angabe auf welchen Bereich eines Buches oder Artikels Bezug genommen wird. Beispiel (wie oben): *Siehe Abbildung 24 aus [3] oder (Seiten 44-52 aus [4]) oder Kapitel 3 aus [5]*.

## Literaturverzeichnis

- Berry, R. and Burnell, J. (2005). *The handbook of astronomical image processing*. Willmann-Bell, Richmond, VA.
- McLean, I. S. (1997). *Electronic Imaging in Astronomy*. Wiley-Praxis, Chichester, New York.

## Weiterführende Literatur

Steve B. Howell (2006). *Handbook of CCD Astronomy*. Cambridge University Press

Dieses Buch ist am Campus (und via VPN) in der 2. etwas modernisierten Version (Juni 2012) auch als elektronischer Volltext verfügbar.

Nick Waltham. *CCD and CMOS sensors* (<http://www.issibern.ch/forads/sr-009-23.pdf>)

Claudio Cumani. *Introduction to CCDs* (<http://www.eso.org/~ccumani/doc/2004-CCDs-Introduction.ppt>)

R. A. Jansen *Astronomy with Charged Coupled Devices* (<http://www.public.asu.edu/~rjansen/ast598/2006ACCD.ebook...1J.pdf>)

## Anhang A

# Auslesesoftware SIPS

Der Start der Software öffnet automatisch einige Fenster. Neben der Darstellung des Bildes sind 2 Unterfenster für uns wichtig:

Das Histogramm der Belichtungsverteilung und das Steuerfenster.

Im Steuerfenster hat es etliche Untergliederungen (Tabs). In Abbildung A.1 sehen Sie die Temperatursteuerung. In der Abbildung sehen sie neben der Temperatur des CCDs in rot die Temperatur der Kamera am Außenkörper (gelb) und den Stromverbrauch der Kühlung in blau. besonders letzteres zeigt Ihnen, wann das System einen stabilen Zustand erreicht hat ('Einschwingen'). Alleinig, dass die Temperatur selbst den Zielwert erreichte ist keine hinreichende Bedingung.

In der Abbildung A.2 sehen Sie die Steuerung der Belichtungen. Oben links wählen Sie den Typ

- BIAS = keine Belichtung und Schutter bleibt zu;
- DARK = längere Belichtung und Schutter bleibt zu;
- LIGHT = Aufnahme mit Licht. Verwenden Sie letztere auch für unsere Flat frames. Es gibt wohl eine eigene Einstellung FLAT. Diese führt aber zu automatischen Bearbeitungen der Rohdaten. Dies ist hier unerwünscht, da es die Statistik verändert.

Darunter können Sie die Belichtungsdauer einstellen sowie die Anzahl der Wiederholungen.

Die Parameter direkt darunter (preflash + clean) bitte auf "OFF" setzen.

Der Button 'Start Exp.' startet dann die Belichtung.

Rechts geben sie einen "prefix" des Dateinamens an und ein Verzeichnis (Folder). Darunter läuft eine "ordinal number" automatisch hoch. Sie können bei Bedarf diese händisch ändern. Die Datei hat dann den Namen "prefix" "ordinal number".fits.

Um beim Beispiel aus der Abbildung zu bleiben - die nächste Datei heißt dann dm070113.fits denn die "ordial number" wird immer 4-stellig verwendet. Das system zählt die "ordinal number" automatisch hoch. Sie können deshalb im Prinzip Serien von Aufnahmen machen nachdem Sie nur am Anfang einen (sinnvollen) "prefix" eingegeben haben. Dennoch unbedingt im Protokoll mit notieren. Vorsicht: Bei der ersten Änderung des Typs (z.B. BIAS → DARK) "vergisst" dieses Programm die Einstellung des Dateipfades (Folder). Diese daher immer kontrollieren!

Ein weiteres Problem stellt bei dem Programm vorauselnde Änderungen von Parametern dar. Wenn Sie in einer Serie – während noch immer Aufnahmen laufen – Parameter ändern, ist das Verhalten des Programms etwas "stochastisch". Manchmal wendet es diese bereits auf den Rest der noch laufenden Serie an und manchmal startet es ungefragt die nächste Serie mit einem Teil der alten Parameter. Daher bitte die Parameter für die nächste Serie immer erst nach Beendigung der Serie einstellen.

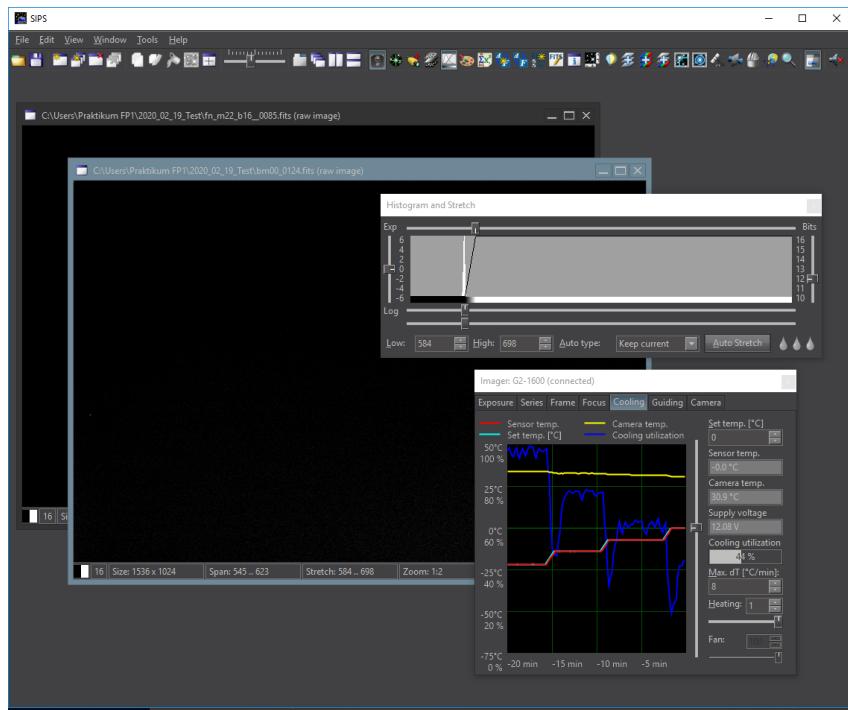


Abbildung A.1: Die Kamerasteuerungssoftware SIPS mit der Temperaturregelung und dem Fenster für die Werte-verteilung (Histogramm).

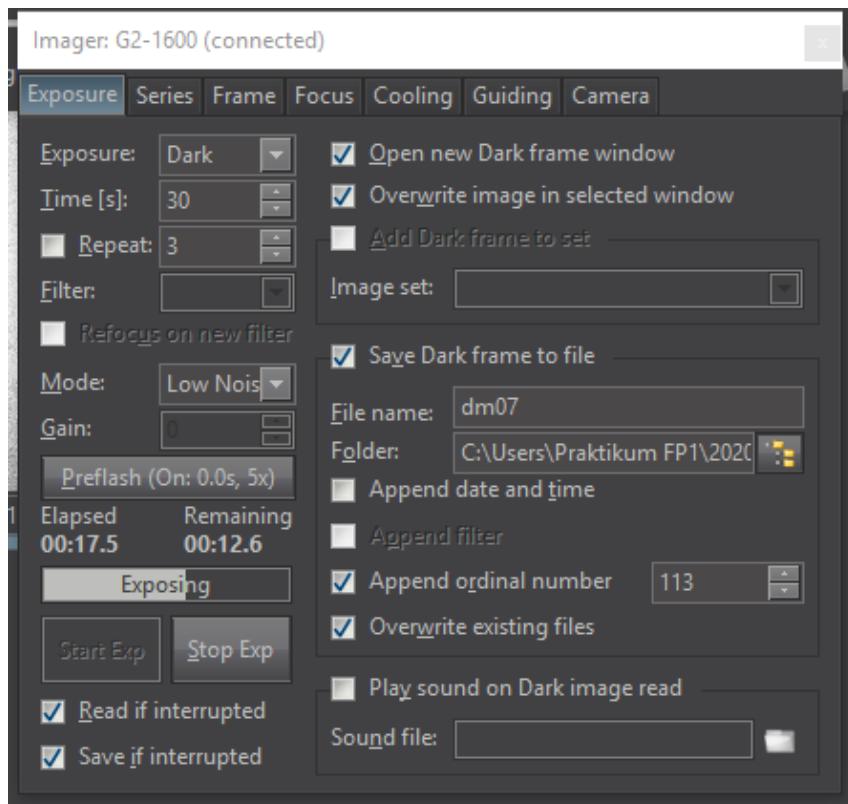


Abbildung A.2: Steuerungsfenster der Belichtungen der SIPS Software

## Anhang B

# Quick pre-View mit dem NASA Tool FitsView.

Das NASA HEASARC Tool fv ist geeignet, schnell Information aus FITS Files zu bekommen. Es erlaubt neben der Darstellung selbst auch erste schnelle Betrachtungen mit geringer statistischer Genauigkeit. Es reicht aber gut aus, um am Versuchsrechner Ihre Messungen zu prüfen.

Es ist komplett freeware und für fast alle Betriebssystem(-varianten) verfügbar.

<https://heasarc.gsfc.nasa.gov/docs/software/ftools/fv/>

Die Abbildung B.1 zeigt die prinzipiellen Bereiche. Der Aufruf öffnet zuerst nur die oberen beiden linken threads. Hier können Sie den Header einsehen und auch heraus kopieren (z.B. für das Protokoll). Mittels dem Button Image öffnet sich die Darstellung. In den verschiedenen Pulldowns des Menüs können Sie verschiedenste Operationen durchführen. So öffnet z.B. Colors->Rescale Image das Histogramm der Datenverteilung wo Sie dann interaktiv die Helligkeit und den Bereich der Darstellung einstellen können. Mittels Tools->Image Probe können Sie das Statistik wie in der Abbildung öffnen. Dabei lässt sich der Testbereich im Bild interaktiv mit der Maus verschieben. Das Programm steht Ihnen auf unseren Linux Maschinen unter dem Kommando fv ebenso zur Verfügung nachdem sie das modul load fv geladen haben.

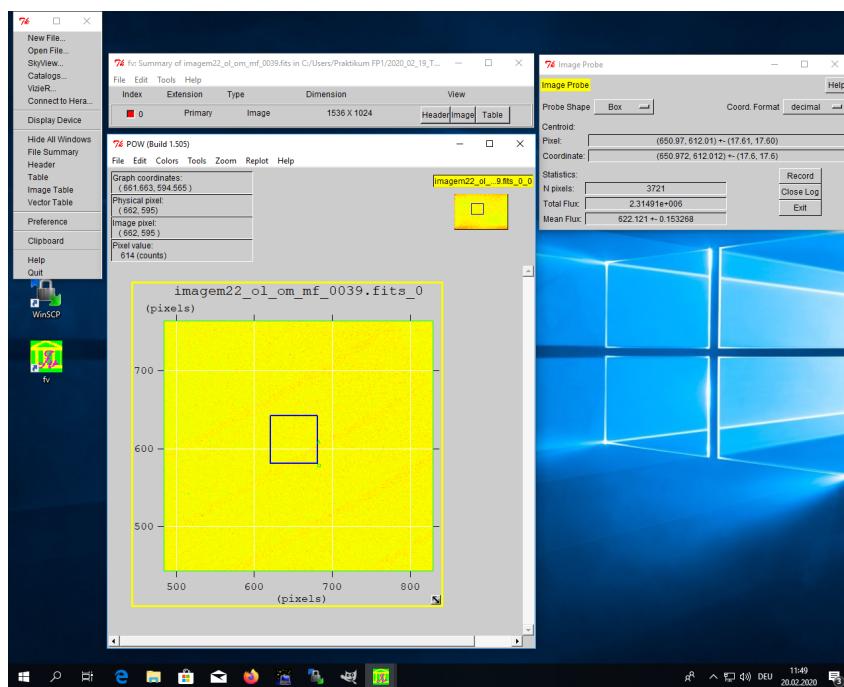


Abbildung B.1: Arbeiten mit dem NASA FITSview

## Anhang C

# Auswertung mit MIDAS

### C.1.: Einführung

Das Munich Image Data Analysis System (MIDAS) der Europäischen Südsternwarte (ESO) ist ein seit Jahrzehnten gewartetes Community Produkt zur Auswertung von astronomischen Aufnahmen. Auch wenn keine größeren Weiterentwicklungen mehr durchgeführt werden, wird es immer noch voll gewartet. Alternative zu MIDAS ist AstroPy (derzeit Version 5.0.1). Es enthält sicherlich alle benötigten Bausteine. Aber es müsste erst mittels größerer Programme für den jeweiligen Versuch nutzbar aufbereitet werden. Da hier die Physik des Halbleiterbildgebers CCD und nicht das Programmieren im Vordergrund des Lernziels steht, wird empfohlen die fertigen Tools von MIDAS zu verwenden.

Die Anleitung ist identisch für die Nutzung an den Rechnerräumen des ZID wie an unserem Institutsrechnerraumes. Mittels dem module Befehl kann man die Verfügbarkeit von Software steuern. Im gegenständlichen Fall gibt man

```
module load midas/19FEB
```

ein. Mittels `module unload` kann man es deaktivieren um z.B. die Version zu ändern. Mittels `module avail` kann man sich eine Liste der so verfügbaren Programme holen. So ist z.B. das CERN Statistikpaket `root`<sup>1</sup> für Spezialisten in der Auswertung (Fitting, Fehler, Fehlerfortpflanzung, ...) für viele Versuche interessant.

Man startet die MIDAS shell mittels dem Kommand

```
inmidas
```

Sollte man unter dem selben Benutzer mehrerer Sitzungen machen wollen (auch wenn es auf verschiedenen Rechnern unseres Raumes ist) muss man die Parallel Option nutzen

```
inmidas -p NN           NN = 00,01,02, ... 99
```

Zum Verlassen der shell nutzen sie eines der Kommandos `bye`, `exit` oder `quit`. Diese shell erlaubt die Wiederholung von Kommandos mit einem einfachen '.' (daher Vorsicht mit dem Punkt im Aufruf von UNIX Kommandos). Ebenso kann man mit dem Cursor auf und ab wandern und frühere Befehle editieren. Um ein UNIX Kommando aufzurufen setzen sie ein '\$' davor. Vorsicht: es werden dennoch die MIDAS shell-spezifischen Wildcards wie z.B. der Punkt oder die Bedeutung von geschwungenen Klammern interpretiert. Wollen sie das vermeiden müssen Sie ein doppeltes ' \$\$ ' voran setzen. Beachte Sie, dass die Kommandos in einer Untershell (in UNIX heißt es child task) ausgeführt werden und daher eventuell keinen Einfluss auf die als 'parent' laufende MIDAS shell haben. z.B. sind Kommandos wie `setenv`, `export` und `cd` somit wirkungslos. Das ist kein Problem von MIDAS, sondern eine Feature von UNIX/LINUX selbst. Um letzteres zu ersetzen gibt es daher dass MIDAS Kommando `change`.

Damit sie, zwecks Vorbereitung des Versuches und Einarbeitung an den ZID Rechnern, arbeiten können, stehen in der Datei A003\_Demo.zip im OLAT einige Bilddateien und der unten gezeigte demo batch zur Verfügung. Der wichtigste Befehl (von den etwa 3000 MIDAS Befehlen) ist

```
help [praefix[/postfix]]
```

Hier zur folgenden "Nomenklatur" - eckige Klammern bezeichnen (mit einer Ausnahme) optionale

---

<sup>1</sup><https://root.cern.ch/>

Parameter. Groß- und Kleinschreibung ist nicht wichtig (außer bei Filenamen da hier die UNIX-Regeln gelten). Praktisch alle MIDAS Kommandos bestehen aus zwei Teilen.

**praefix/postfix**

Dabei muss nur so viel vom Wort gegeben werden, dass es klar ist was gemeint ist. So ist CO unzureichend da es COPY aber auch COMPUTE bedeuten könnte. Dahinter kommen Optionen. Möchte man einen optionalen Parameter nicht nutzen, aber einen anderen weiter hinten in der Reihe, so ist ein ? an der Stelle zu setzen. z.B. um Bilder in einer mathematischen Operation zu verarbeiten geben Sie

COMPUTE/IMAGE a = b + c \* LOG10(d/e)

Es werden .fits Dateien als Input verwendet, der Output erfolgt ebenfalls in .fits Dateien.

Sie können eigene Serien von Befehlen zu batches mittels eines Editors Ihrer Wahl erstellen (z.B. gedit). Sie sollten im Filenamen die Endung .prg haben und werden dann mit zwei [at] Zeichen davor aufgerufen.

z.B.               @@ bias

ruft die Befehle in der Datei bias.prg auf.

## C.2.: Wichtige Kommandos im Schnellüberblick

An sich erhalten Sie mit help genaue Beschreibungen. Einige wenige für uns hier sehr wichtige Kommandos sind auch im nächsten Abschnitt vollständig gelistet. Hier werden sie kurz mittels Beispielen eingeführt.

Öffnen und Schliessen von Grafikfenstern für Plots:

CRE/GRA N

DEL/GRA N

Dabei ist N eine Zahl von 0 bis 9 – zerstören Sie Fenster NIE mit dem 'X' des Window Managers. Sie können sie mit den '\_' zu Icons verkleinern. Zerstören stört aber die Session – Sie müssen dann MIDAS verlassen und neu starten. Größenanpassungen funktionieren auch meist nur VOR dem ersten Plot richtig.

Zuweisen eines Grafikfenstern für Plots:

ASSI/GRA g,N

ASSI/GRA POSTSCRIPT

die folgenden Grafikbefehle werden entweder auf das Grafikfenster Nummer N oder auf eine Postscript Datei Namens postscriptN.ps geleitet. Ich verwende das Umleiten auf Postscript selten - meist mache ich Hardcopy mittels des LiNUX Grafikprogrammes GIMP.

Öffnen, Schliessen und Zuweisen von Fenstern für Bilder:

CRE/DIS N

DEL/DIS N

ASSI/DIS d,N

ASSI/DIS POSTSCRIPT

Anders wie bei den Grafikfenstern ist hier die Größenänderung möglich (aber ebeno kein Zerstören!). Im Wesentlichen muss man den letzten Ladebefehl aber wiederholen.

Bilder laden:

LOAD/IMAGE a [cuts=min,max] [center=@x,@y] [scale=factor]

lädt ein Bild. Optional kann man den Wertebereich für die Skala/Grauwerte steuern (cuts), das Zentrum definieren (center) oder die Skalierung. Dabei sind nur ganzzahlige Werte möglich. Positive Werte bedeuten zoom, negative un-zoom.

Darstellung invertieren:

LOAD/ITT NEG

dreht die Werteskala im Display um. Möglich sind RAMP (normal), NEG (umgekehrt), LOG (normal logarithmisch – hebt schwächere Werte hervor), NEGLOG = NEG+LOG

Darstellung in Farbskala:

LOAD/LUT rainbow

Ersetzt die Grauskala mit einer Farbtafel. Möglich sind RAMP (normal grau), RAINBOW, SMOOTH, FIRE, HEAT, RANDOM, ....

Statistik über ein Gebiet einer Datei - einer unserer wichtigsten Befehle für diesen Versuch:

STATISTICS/IMAGE a.fits [@100,@200:@500,@300] 1.0 -20.0,+60.0

Erstellt die Statistik des Bildes a.fits in einem Subgebiet von der linken unteren Ecke (100,200) bis zur oberen rechten Ecke (500,300) – diese Koordinatenangabe ist die zuvor erwähnte Ausnahme der eckigen Klammer. Hier ist es nicht als optional gedacht sondern die Ortsangabe. Wird der Ortsparameter nicht gegeben wird das ganze Bild genommen. Die beiden weiteren Parameter geben die Statistikgebiete (für ein späteres Histogramm) an. Hier würde ein Histogramm mit binsize 1.0 entstehen und nur die Werte zwischen -20 und +60 verwendet. Das Programm gibt am Ende an, wie viele % der Pixel verwendet werden konnten. An Stelle des Filenamens+Gebietes kann man den Text CURSOR stellen, wenn das Bild zuvor in ein Displayfenster geladen wurde. Dann kann man interaktiv eine Statistikbox mit der Maus fahren - jeder Klick mit der linken Maustaste erzeugt dort eine Statistik, mit der rechten Maustaste beendet man die Serie.

Histogramm:

**PLOT/HISTO a.fits ? ? LOG**

Erstellt das Histogramm mit logarithmischer Skala (4. Parameter - daher die beiden nicht verwendeten dazwischen ein Fragezeichen). Der Befehl benötigt den Aufruf von stati/image und die Erzeugung eines Grafikfensters zuvor. Lässt man den LOG Parameter weg, wird eine lineare Y-Achse verwendet.

Messen von Einzelpunkten:

**GET/CURSOR**

**GET/GCURSOR**

**CLEAR/OVERLAY**

Misst im Bild (CURSOR) oder im Grafikfenster (GCURSOR). Die Messung hinterlässt Markierungen wo gemessen worden ist. Oft nützlich wenn man ein neues Bild lädt und die alten Messorte noch darüber markiert haben will. Der letzte dieser drei Befehle löscht diese Markierungen.

Abfragen von FITS Header Information:

**READ/DESCRIPTOR myfile.fits [NAME]**

Wird kein bestimmter NAME angegeben werden die wichtigsten Grundparameter (Größe, Datentype etc.) ausgegeben. Wildcard (\*) ist zulässig im NAME. So zeigt

**READ/DESCRIPTOR myfile.fits \***

alles aus dem Header an.

Erzeugen eines Katalogs von Bildern (für andere Kommandos).

**cre/icat tmp\_bias.cat tmp\_bias\_????.fits**

erzeugt eine Katalogdatei aus dem Bildern welche dahinter angegeben wurden. Dabei gelten die Regeln der UNIX Wildcards für \* und ? (hier also forciert eine 4 Stellige Eingabe – es können aber auch Buchstaben sein – nicht notwendigerweise 4 Ziffern).

Ein weiteres Sehr wichtiges und mächtiges Kommando ist

**average/image result = tmp\_bias.cat ? ? MINIMUM,5**

Es mittelt die untersten Werte aus 6 Dateien (Minimum + 5) aus dem zuvor erstellten Katalog. wenn man z.B. 8 Bilder hat fallen immer die beiden höchsten Werte eines Pixels an der Position (x,y) weg. Damit kann man vermeiden dass z.B. Cosmic events mit in die Mittelwerte eingehen. Da die Sortierung für jeden Bildpunkt extra gemacht wird ist es egal in welchem Bild diese auftreten. Andere Varianten sind MEDIAN,3,2 = Mittelwert der Bilder um den Median Wert herum (3 drunter + median + 2 drüber = 6), MEAN,3,2 und MAXIMUM,2. Das verwendet man am besten für die BIAS und DARK Aufnahmen.

Einfache Bildoperationen erstellt man mit

**comp/ima result = (tmp\_dark.fits - BIAS.fits) / 10.0**

Es sind fast alle mathematischen Operationen definiert (siehe unten). Vorsicht die Zeilenlänge in der Shell ist auf etwa 90 Zeichen limitiert. Daher muss man schon mal Zwischenergebnisse rechnen.

### C.3.: Wichtige komplizierteren Kommandos im Detail

Purpose:

Calculate statistics of a frame.

Syntax:

```
STATISTICS/IMAGE [frame] [area] [bins] [lo,hi_exc] [option]
                  [outtab] [plotflg] [format]
frame = input data,
(a) name of image frame
(b) table_name,column (as :col_label or #col_no);
defaulted to displayed image
area = window within image to work on;
      (a) subframe      or      subframe1,subframe2, ...
          subframe(s) in the usual MIDAS syntax: [xs:xe] or
          [xs,ys:xe,ye] or [xs,ys,zs:xe,ye,ze],
          see the Help of READ/IMAGE for more info
      (b) complete frame, indicated by '+' (the plus sign);
      (c) CURSOR for cursor defined windows, size may be adjusted
          interactively;
          or CURSOR,cmax for max. 'cmax' cursor inputs;
      (d) ROW for processing of all rows of 2-dim input frame, or
          ROW,i,j,... to process rows no. i, j, ... of frame;
      (e) COLUMN for processing all columns of 2-dim input frame, or
          COLUMN,i,j,... to process columns no. i, j, ... of frame;
      (f) PLANE for processing all planes of 3-dim input frame, or
          PLANE,i,j,... to process planes no. i, j, ... of frame;
      (g) table_name for table defined subimages, which are defined
          in the (real data) columns labeled :XSTART, :XEND, :YSTART
          and :YEND. For 3-dim frames also :ZSTART and :ZEND are used;
      (h) [xpix*ypix] fixed sized intervals/subframes
          as many subframes of 'xpix' * 'ypix' pixels are used as
          needed to cover the full 1- or 2-dim image, if NPIX(1) is
          not a multiple of 'xpix' the last interval gets the size to
          fit (the same applies to y-direction)
          for 1-dim frames [xpix*1] has to be given
          defaulted to (b);
bins = bin size for histogram calculation or no. of bins;
      to distinguish between the two options, the character # is used
      for no. of bins; defaulted to #256;
      if the parameter lo,hi_exc is given, two excess bins are added.
      Bins are half-open intervalls [a,b[, i.e. all values v with
      a <= v < b contribute to this bin.
lo,hi_exc = optional low and high excess values,
      i.e. all values less than lo_exc contribute to the low excess
      bin (1st bin) and all values greater or equal to hi_exc
      contribute to the high excess bin (last bin) in the histogram.
      The resulting histogram is stored in the descriptor HISTOGRAM
      and can be converted to image or table format by the
      application procedure @a histogram (see HELP/APPLIC histogram);
option = 4 chars flag for: statistics_level (1), display_flag (2),
                           rect_draw_flag (3), merge_flag (4);
statistics_level: F/G or V/W or R/X or S or H or M;
```

---

```

F(ull) Statistics => compute statistical quantities (1) - (4)
                  (as explained in the Notes below)
G is like Full, but exact median is obtained,
V is like Full, but also stat. quantity (5) is calculated
W is like V, but exact median is obtained and used for (5)
R(educed) Statistics => only (1) -> (3) are computed
X like Reduced, but exact median is obtained,
S(hort) Statistics => only (1) + (2) are computed,
H(istogram) => (1) + (4) are computed,
M(inmax) => only (1) is calculated
display_flag: F(ull), S(hort), X(short) or N(o) display of
              results;
rect_draw_flag: Y(es) or N(o) for drawing rectangles around
                 subframes (only applied to area option );
merge_flag: M(erge) or N(o Merge) - not applicable to area
              choices (b) and (h);
              if set to Merge, the area of all subframes specified are
              used to gather the required statistics;
              defaulted to GFYN
outtab = optional output table (or image)
          for table output the table columns hold same contents as the
          descr. STATISTIC (cf. Note);
          enter 'outtab,A' if results should be appended to existing table
          'outtab', otherwise a new table 'outtab' will be created
          enter 'outtab,image' if all results are written into a 2-dim
          image 'outtab', each row holding contents like descr. STATISTIC
          enter 'outtab,image,stat_val' if only one specific statistical
          value should be written into a 1-dim image 'outtab'
          stat_val may be any of: Minimum, Maximum, Mean, Stddev, Mom3,
          Mom4, Total, Median, Mode1 or Mode
plotflg = P(lot) or N(oplot) for plotting the histogram (with some
          consequences for area option (c) -> (g));
          defaulted to N
format = Format string used for displaying minimum, maximum, mean and
          std_dev of frame in Fortran notation;
          defaulted to E15.6

```

See also:

STATISTICS/TABLE, PLOT/HISTOGRAM, FIND/MINMAX, FIND/PIXEL  
 SHRINK/IMAGE, REBIN/STATISTICS, HELP/APPLIC histogram

Note:

The parameters may also be referenced via  
 FRAME=, AREA=, BINSIZE=, EXCESS=, OPTION=, OUTTAB=, PLOT=, FORMAT=

The following statistical quantities are computed:

- 1) minimum + maximum
- 2) mean value, standard deviation
- 3) dimensionless 3rd, 4th moment + intensity (sum of pixels)

formulas used for 3rd, 4th moment are:

$$\text{3rd moment} = 1/N * 1/\text{std}^{**3} * \text{SUM of } (x - \text{mean})^{**3}$$

$$\text{4th moment} = 1/N * 1/\text{std}^{**4} * \text{SUM of } (x - \text{mean})^{**4}$$

(with std = standard deviation) - they are also known as

moment coefficients of "skewness" and "kurtosis"

4) histogram + median + first mode + mode of data values.

5) mean abs. deviation ( $m_a_d$ ) of mean + of median

formulas used are:

$m_a_d$  of mean =  $1/N * \text{SUM of } \text{abs}(x - \text{mean})$

$m_a_d$  of median =  $1/N * \text{SUM of } \text{abs}(x - \text{median})$

**Purpose:**

Calculate the average of images - size of result frame is either intersection or union of all image sizes.

**Syntax:**

```
AVERAGE/IMAGES out = in_specs [merge] [null] [av_option] [dat_intval]
out = result frame
in_specs = specification of input frames, either
    in1,in2,...inj    or
    catalog.cat = name of catalog containing the input frames    or
    3-dim image cube, averaging will be done over all xy-planes
merge = M (for Merge), size of result frame will be set to the union
       of all input frames, else the size will be the intersection of all
       input frames;
       defaulted to N (No merge)
null = value for undefined pixels, also the real keyword NULL(2) will
      be updated accordingly;
      if set to '+', undefined pixels are set to the value of the
      pixel calculated before;
      defaulted to the actual contents of keyword NULL(2)
av_option = av_spec,av_low,av_hi,av_switch;
    av_spec - Average (a), Minimum (b), Maximum (c) or Median (d),
    (a) take average of of all valid pixels;
    (b) take minimum of of all valid pixels;
    (c) take maximum of of all valid pixels;
        for (b) + (c) av_low may be set to indicate that the
        average (instead of the min/max) should be taken of
        minimum/maximum and 'av_low' next higher/lower pixels
    (d) take median of all valid pixels;
        for (d) av_low, av_hi may be set to indicate that the
        average (instead of the median) should be taken of all the
        pixels in an interval around the median pixel,
        to indicate if the interval refers to the indices (order) of
        the pixels or their values, av_switch is set to INDEX or
        DATA, if 'av_switch' not given, it is used as INDEX;
    except for (d) with av_switch = DATA, 'av_low' and 'av_hi' must
    be non-negative values;
    defaulted to AV
dat_intval = rlow,rhigh - optional interval for input pixels;
            if omitted (the default) all pixels in the input frames are
            used for the calculations
```

**See also:**

AVERAGE/WEIGHTS, AVERAGE/ROW, AVERAGE/WINDOW

**Note:**

Undefined pixels result from non-overlapping areas when using the Merge option or from the constraints specified in the average options and data interval.

Overlapping areas will be determined via the world coordinates of the frames, i.e. via descriptors START and STEP.

The input frames are all opened simultaneously to speed up the averaging process. Thus, the no. of frames we can work on depends on a system variable (FOPEN\_MAX on Unix systems, stored in stdio.h).

In ‘averag.exe’ a max. of 300 frames is foreseen. This is an upper limit for the internal structures. In practice the max. number of frames opened will be limited by FOPEN\_MAX and the available memory.

If you use FITS files directly, e.g. in your catalog, please, keep in mind, that FITS files are always mapped totally into memory, so you could end up allocating a lot of memory - it may be worth to convert the input images to the Midas internal format first.

If the keyword MID\$SPEC contains the string ‘DEBUG’, a MIDAS image named ‘averdummy.dum’ will be created containing the valid pixel count for each x,y position of the result frame.

If av\_option=median, the index of the median pixel is calculated as nopix/2 for even nopix (nopix = no. of pixels used) and (nopix+1)/2 for odd nopix.

Examples:

```
AVERAGE/IMAGES av1 = ccd01,ccd02,ccdrbv,flata,flatb,flatc,flatd
Compute average of given input frames on their overlapping area.
Size of result frame ‘av1.fits’ will be the overlapping part of all
input frames.
If the above input frames had entries in the image catalog
‘ccdfat.flat.cat’, and only these,
“AVERAGE/IMAGES av1 = ccdfat.flat.cat” would do the same...
If the above input frames had been all stored as the planes of a
3-dim image cube ‘bigcube.fits’,
“AVERAGE/IMAGES av1 = bigcube” would do the same...
AVERAGE/IMAGES av1 = ccdfat.flat.cat ? -99.9 min 0.3,7.23
Take minimum of input frames, but use only pixels in the
interval [0.3,7.23] for finding the minimum. If for a pixel all
corresponding input pixels are outside the given interval, the
pixel is set to -99.9.
AVERAGE/IMAGES av1 = ccdfat.flat.cat ? -99.9 min,2 0.3,7.23
As above, but resulting pixel will be the average of the 3 lowest
pixels. Less than 3 pixels may be used for the averaging depending
upon how many valid pixels exist at a given x,y position of the
result frame.
For example, if the first pixels in the 7 input frames had the
values 1.0, 4.4, 2.1, 3.0, 2.0, 4.0, 5.6 then the first pixel in
the result frame would be calculated as (1.0 + 2.0 + 2.1)/3 .
AVERAGE/IMAGES av1 = ccdfat.flat.cat ? -99.9 max,4 0.3,7.23
As above, but resulting pixel will be the average of the 5 highest
pixels.
AVERAGE/IMAGES av1 = ccdfat.flat.cat ? + max,4 0.3,7.23
As above, but undefined pixels are set to the value of the pixel
calculated before. If the very first result pixel is an undefined
pixel, it is set to 0.0 .
AVERAGE/IMAGES av1 = ccdfat.flat.cat ? + median,1,2 0.3,7.23
As above, but resulting pixel will be the average of (if possible)
the 4 pixels in the ordered interval [median-1,median+2]. Using
the numbers from the example above, the first result pixel would be
(2.1 + 3.0 + 4.0 + 4.4)/4 = 3.375      (median is 3.0).
AVERAGE/IMAGES av1 = ccdfat.flat.cat ? + median,1,2,data 0.3,7.23
As above, but resulting pixel will be the average of all pixels
```

within the interval [median\_val-1.0,median\_val+2.0]. Using the same numbers as above, the first result pixel would be  
 $(2.0 + 2.1 + 3.0 + 4.0 + 4.4)/5 = 3.10$

AVERAGE/IMAGES av2 = ccdflat.cat M 0.33

Compute average of given input frames on their combined area.  
Size of result frame ‘av2.fits’ will be the union of all frames with entries in image catalog ‘ccdflat.cat’. Undefined pixels are set to 0.33

Here only pixels at x,y-positions with no corresponding input pixels are undefined, since parameters ‘av\_option’ and ‘dat\_intval’ are not given.

Purpose:

Perform arithmetic operations on constants and images, using world coordinates.

Subject:

Arithmetic.

Syntax:

COMPUTE/IMAGE [outspec =] expression

outspec = frame to get result of expression.

It will be created with the size of the common area of all images involved in the expression. The result frame will be created using real data format, no matter what data format the individual input frames have.

If "outspec =" is omitted, the expression should contain only numerical constants, the result of the computation is shown on the terminal and stored in real keyword OUTPUTR(1).

If descriptor copying is enabled (which is the default, see also the command SET/MIDAS\_SYSTEM DSCCOPY=..),

all non-standard descriptors of the 1. input frame in the expression are copied to the result frame unless you enter 'frame,dscname' as outspec with dscname the name of a frame in the expression;

then, the non-standard descriptors of that frame are copied instead of the ones of the 1. frame in the expression

expression = arithmetic expression with up to 21 operands, which may be functions and frames or/and constants in the usual algebraic notation.

Note:

OJO: Images must have same stepsize within 0.0001 of a step and the overlapping start values must be within 0.01 of a step. Computations are done on the common area of all images only!

Spaces before and after the "=" sign are required. If the name of frame begins with a digit or contains a special char. like, e.g. '+', '-' ,... the name (with filetype!) has to be enclosed in double quotes.

Inside the expression may be spaces or not. However, if your expression is longer than 100 chars. (the current max. token length) without spaces, you should include spaces between some operands, then the limit is 256 chars. for the complete command line.

If the result frame has the same name as a frame appearing in the expression, no new frame is created, but the result frame is overwritten (also in VMS!). This option should be used with care, since computations are only done on the overlapping area of all frames in the expression!

Furthermore, since no new result frame is created, the final data is reconverted to the original data format when closing the result frame and you could get truncation errors!

The operations +, -, \*, / and \*\* are supported with the same precedence as in FORTRAN. Parentheses may be used to change that order as well as to nest operations.

The functions

SQRT(a), EXP(a), EXP10(a), LN(a), LOG10(a), SIN(a) ASIN(a), COS(a), ACOS(a), TAN(a), ATAN(a), INT(a), ABS(a), ATAN2(a1,a2), MAX(a1,a2), MIN(a1,a2) and MOD(a1,a2)

are implemented as in FORTRAN. Except, that INT(a) will return the nearest integer of "a" converted to real. The trigonometric functions expect arguments in degrees!

Results of illegal operations (e.g. division by zero) are set to the "null value" stored in the real keyword NULL.

This command uses single precision (floating point) calculations, if you need higher precision, use COMPUTE/DIMAGE instead.

See also:

COMPUTE/PIXEL, COMPUTE/ROW, COMPUTE/COLUMN, COMPUTE/XYPLANE  
COMPUTE/DIMAGE, COMPUTE/PLANE, COPY/II, REPLACE/IMAGE  
@a func2d, SET/MIDAS\_SYS

Examples:

COMPUTE/IMA r = sqrt(c+5.-log10(b))+abs(aa)

Compute resulting image 'r.fits' from the expression involving the frames 'c.fits', 'b.fits' and 'aa.fits'.

If descriptor copying is enabled, all non-standard descriptors of 'c.fits' (the 1. frame in the expression) are copied to 'r.fits'.

COMPUTE/IMA r,aa = sqrt(c+5.-log10(b)) + abs(aa)

As above, but if descriptor copying is enabled, all non-standard descriptors of 'aa.fits' are copied to 'r.fits'.

COMPUTE/IMA EXP(25./3.4) + SIN(1.1)

Compute the expression and store result in keyword OUTPUTR(1).

COMPUTE/IMA 10.3\*{OUTPUTR(1)}+4.5

Use result from computation above.

COMPUTE/IMA r = sqrt(c) + qso+22.fits"

Compute resulting image 'r.fits' from the expression involving the frames 'c.fits' and 'qso+22.fits'.

COMPUTE/IMA a = a+LOG10(a)

Also in VMS just modify frame 'a.fits', do not create a new file.

#### C.4.: Die Programmierung der Befehlssprache

MIDAS erlaubt auch die Erstellung von Programmen. An sich ist darin jeder MIDAS Befehl (auch der Aufruf eines UNIX Kommandos mit \$) erlaubt. Hinzu kommen Kommandos welche eine Ablaufsteuerung und Variablen erlauben. **Es ist NICHT notwendig diesen Versuch damit auszuwerten.** Es ist aber für Personen mit Programmierkenntnissen wahrscheinlich erheblich schneller, da sich Teile immer wieder wiederholen. Ansonsten gehen Sie weiter zum nächsten Abschnitt.

Alles hinter einem Rufzeichen ist ein Kommentar (egal ob in einer Zeile ohne Kommando oder hinter einem Kommando).

Definition einer Variablen

```
define/local name/typ/von/bis [default]
```

Dabei sind zulässige Typen I (integer), R (real/float), D (double) und C (character). Die Längenangabe ist verpflichtend, auch wenn Sie keine Felder definieren. Einzelvariablen sind somit /1/. Definiert man ganze Felder wird entweder mittels Klammern angegeben welcher Wert verwendet werden soll (z.B. OUTPUTR(8)) oder es wird bei Zuweisungen immer bei 1 automatisch begonnen. D.h. das, wenn Feld a seinerseits 5 Elemente und Feld b seinerseits 8 Elemente hat, bewirkt die Anweisung

```
define/local a/r/1/5 3.14,2.17,3.2,4.1,10.2
define/local b/i/1/8 0 ! fuellt alles mit Nullen
b = a
```

dass die ersten 5 Elemente von b mit den Werten von a überschrieben werden. Es sind auch Bereiche (from:to) möglich. Die Zuweisung verhält sich wie in C und FORTRAN (keine Änderung des Typs – daher wird nur der Integer Anteil übergeben) und nicht wie in Python (welches den Typ des Ziels jenem der Quelle anpasst)! Wie das Kommando schon sagt, sind diese Variablen LOKAL. Somit ist sowohl der rekursive Selbstauftrag, wie auch die die Verwendung selber Namen in anderen Routinen zulässig.

Sie können mit diesen Variablen direkt in Kommandozeilen rechnen, sie aber ebenso in Kommandos als Textbausteine einsetzen. Für letzteres setzen Sie sie zwischen geschwungene Klammern:

```
! definition der Variablen - Namens I und Typ Integer und einem Anfangswert von 15
define/local i/i/1/1 15
! und zwei Textvariablen
define/local pref/c/1/20 "bias_"
define/local file/c/1/30
! und rechnen damit
i = i + 10
! und einsetzen
file = {pref}{i}.fits      ! erzeugt "bias_0025.fits"
! oder direkt verwenden
comp/ima tmp = {pref}{i}.fits / {i}.0
! erzeugt comp/ima tmp = bias_0025.fits / 0025.0
! achten Sie beim Dividieren (wie in C) darauf, dass Sie
! keine Integeroperation erzeugen
```

Default ist, dass z.B. Integervariablen 4-stellig mit führenden Nullen ausgegeben werden, real und double zahlen als Exponentenzahlen, usw. Mittels set/format können Sie alle Formate ändern (sehen Sie help set/format).

Reservierte (globale) Variablen sind definiert, um umfangreiche Eingaben und Ausgaben an/von Befehlen zu übergeben. Diese beginnen mit INPUTx und OUTPUTx mit je 20 Elementen – dabei steht an der Stelle

x der Typ (I,R,D) und INPUTC / OUTPUTC mit 60 Zeichen. Vorsicht – jedes Kommando darf diese Felder nutzen – wenn sie z.B. die Ergebnisse der Statistik (aus OUTPUTR) verwenden wollen, müssen sie dies sofort danach tun oder auf lokale Variable umspeichern. Weiter gibt es noch einen Satz von Variablen P1, ..., P8 für Parameter. Auch hier gilt die Regel – sofort Sie auf eigene Variablen transferieren (siehe Beispiele in C.5.)

Parameter an einen batch:

```
define/param P1 ? C "prefix ? "
Erwartet den Parameter Nr 1 - Typ sollte Text
sein (alternativ N für Number wobei kein Unterschied von I,R und D gemacht wird). Da als zweiter
Parameter ein Fragezeichen kommt, kennt dieser Parameter keinen Defaultwert (ansonsten Defaultwert
dort an der Stelle). Wird der Parameter beim Aufruf nicht angegeben erscheint der Text am Bildschirm
und die Eingabe wird abgewartet.
```

Die wichtigste Verzweigung ist die `if then else endif` (else block ist optional).

```
if i .NE. j then
    ! do something
else
    ! do something different
endif
```

Die Bedingungen sind wie in FORTRAN (der damals einzigen international ANSI-genormten Programmiersprache als MIDAS entwickelt worden ist – selbst C hatte damals noch keine Norm) definiert und sind durch die Punkte zu markieren

.NE.	not equal	.EQ.	equal
.GT.	greater than	.LT.	less than
.GE.	greater or equal than	.LE.	less or equal than
.NOT.	inverting the following condition		

Klammerungen um größere Boolesche Ausdrücke zu schaffen sind möglich.

Die wichtigste Schleife ist die DO/FOR Schleife

```
do i = {from} {to} [{step}]
    ! do something
enddo
```

Default step size ist 1. Anders wie in FORTRAN ist es hier erlaubt die Schleifen-Variable innerhalb der Schleife zu verändern.

Abfrage ob eine Datei existiert:

```
if M$EXIST("{prefix}{1}.fits") .EQ. 1 THEN
```

Abfrage eines FITS Header Parameters (copy descriptor -> key)

```
define/local temp/d/1/1
copy/dk dark_0001.fits CCD-TEMP/d/1/1 temp
```

Drucken:

```
write/out so ein Text Nummer 1 hallo Mittelwert mean ist sozusagen ein einfacher printf befehl
```

### C.5.: Beispiel batch files

Der erste File erstellt ein mittleres bias und macht etwas Statistik hernach. Dabei werden keine Parameter übergeben, sondern alle Parameter sind explizit im Batch zu setzen/ändern:<sup>2</sup>

```
! define a character variable (max length 60 - from 1 to 60) for
! the first part of the filename
define/local prefix/C/1/60 "bm22_"

! and a name for the result file
define/local result/C/1/60 "bias_m22"

! define two integer variables with one element (from 1 to 1)
! with the files to be used
define/local from/I/1/1 59
define/local to/I/1/1 68

! and a local loop variable
define/local l/i/1/1
define/local n/i/1/1
define/local m/i/1/1

! set the integer format to have 4 digits forced (= leading 0's)
set/format I4

! make an image catalogue from the FITS files
cre/icat tmp_bias.cat {prefix}{l}.fits

! average the bias images - but take only the lowest
! (m-1) if we have m images
m = to - from + 1
n = m - 1
average/image {result} = tmp_bias.cat ? ? MINIMUM,{n}

! and write it back as normal FITS file
outdisk/sfits {result} {result}.fits B
```

---

<sup>2</sup>Einrückungen bei bedingten Blöcken und Schleifen sind, anders wie bei Python, nicht vorgeschrieben. Aber sie erhöhen die Übersichtlichkeit.

Die weitere Variante nutzt noch mehr von der Befehlssprache (nur die ersten Zeilen sind wiedergegeben macht an sich etwa das gleiche, übernimmt aber die Parameter von der Kommandozeile. Will man z.B. die Dateien bias\_0014.fits bis bias\_0027.fits zu einem bias\_t20.fits mitteln ruft man diesen mit

@@ bias\_p bias\_ 14 27 bias\_t20

auf (Vorsicht – die Version hier und am OLAT ist nur ein kleiner Teil. Sie müssen dies mit anderen und Ihren Teilen kombinieren.)

```
define/param P1 ? C "prefix ? "
define/param P2 ? N "from ? "
define/param P3 ? N "to ? "
define/param P4 ? C "result name ? "
! define a character variable (max length 60 - from 1 to 60) for
! the first part of the filename
define/local prefix/C/1/60 {P1}

! and a name for the result file
define/local result/C/1/60 {P4}

! define two integer variables with one element (from 1 to 1)
! with the files to be used
define/local from/I/1/1 {P2}
define/local to/I/1/1 {P3}

! ... hier fehlen viele Zeilen

average/image {result} = tmp_bias.cat ? ? MINIMUM,{n}
outdisk/sfits {result} {result}.fits B

! ... hier fehlen viele Zeilen

sum = sum / ({count}.0)
write/out mean rms of differences = {sum}
statistic/image {result} [@650,@450:@750,@550] ? ? FN
write/out mean/median BIAS {outputr(3)} {outputr(8)} {outputr(3)}
```

Und hier ein Abschnitt, welchen man für die Statistik (z.B. Ausleserauschen) adaptieren kann.

```
! create (if not yet done) a graphics window
create/gra 0
assign/gra g,0

! now do some statistics
do l = {from} {to}
    ! make the difference of each individual one and the mean -
    compute/image tmp = tmp_bias_{l}.fits - {result}
    ! do statistics between -20 and 20 in steps of 1.0
    statistic/image tmp ? 1.0 -20,20
    plot/histogram tmp
    $$ sleep 1  ! sleep a moment - otherwise the computer
                 ! will be faster than you look on it
enddo
```