

KOOLS 試験観測 (2006/03/31–04/04) レポート

岩田 生 (国立天文台 岡山天体物理観測所)

2006/07/06

1 観測内容

- 2006/03/31 学生実習用バックアップデータ (M67 撮像), NGC4395(low-luminosity Sey1) 分光
- 2006/04/01 (雨天) VPH グリズム試験, ドームフラット
- 2006/04/02 分光標準星, Xe ランプ試験
- 2006/04/03 学生実習, 坂本フィールド撮像
- 2006/04/04 (雨天) ドームフラット (線形性試験)
- 2006/04/11 LED による室内線形性試験

2 VPH グリズム試験

2.1 スリット位置の変更によるピーク波長変化

4月1日に VPH683 を用いてスリット位置をずらしながらドーム内照明を使って連続光の波長特性の変化を調べた。

図1に示すようにスリットの光学系に対する位置を変えて撮像。光源は不安定なので効率の変化は不明。図2のように、スリット位置の変化に応じて peak 位置は変動した。表1におおよそのピーク波長を示す。位置によって、ターゲット波長である 683nm 前後で変動している。

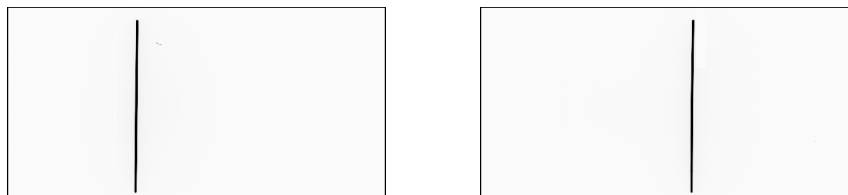


図 1: kls003999(slit2 の位置)+0(左) と+3000(右)

pulse	slit pos.	λ_{peak}	kls00
0	440	7200	3998
+1000		7000	3989
+2000	529	6800	3988
+2500	552	6675	3992
+3000	574	6600	4002

表 1: Relative positions from “slit2” (motor pulse numbers), slit positions (CCDX), peak wavelengths and file numbers.

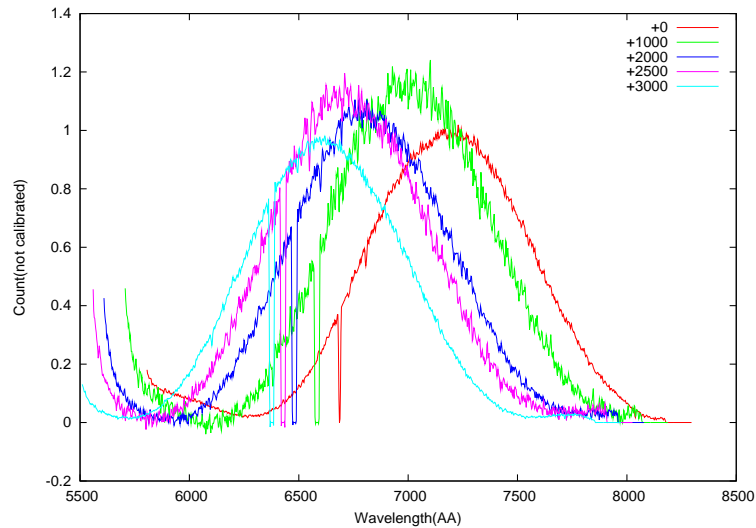


図 2: スリット位置を変更した場合の連続光のカウント

2.2 効率測定

3月31日、4月2日に分光標準星の観測を試みたが、天気が悪く効率測定はできなかった。

2.3 NGC4395 試験観測

Low-luminosity AGN である NGC4395 の分光を VPH495, VPH683 で行った。スリット幅は 1.3"。VPH495 は 1,800sec×1, VPH683 は 1,800sec×2 の積分時間。FeNe ランプで波長較正。分光標準星の観測ができなかったため、flux calibration はできていない。図 3 に 2pixel のスムージングをかけたスペクトルを示す。二つの VPH グレーティングで狙った、H β , OIII, H α , SII といったラインが適切にとれている。次回試験観測で効率を測定したい。

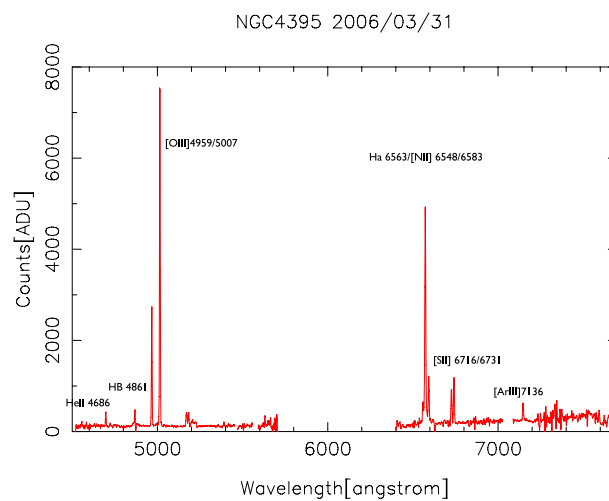


図 3: spectrum of NGC4395. left: VPH495, right: VPH683

3 Xe ランプ試験

ISLE 用に購入された Xe ランプを撮影してみた。AG ユニット部分にランプを入れて、手で保持しながら撮像。図 4 に VPH495 を用いたスペクトル (180sec 積分) を示す。図 5 に見られるような連続光成分がある。強い line はあるが、全体をカバーできるように弱い line もとるとなると、かなり時間がかかる。波長同定を試みたが、 $rms=1.34\text{\AA}$ と悪かった。正しくない波長同定が混じっているかもしれない。

参考までに FeNe の場合のスペクトルを 6 に示した。こちら積分時間 180sec。4 次で fit すると $rms=0.34\text{\AA}$ 。こちら 5000 \AA より短波長側に強い line がないのが問題。

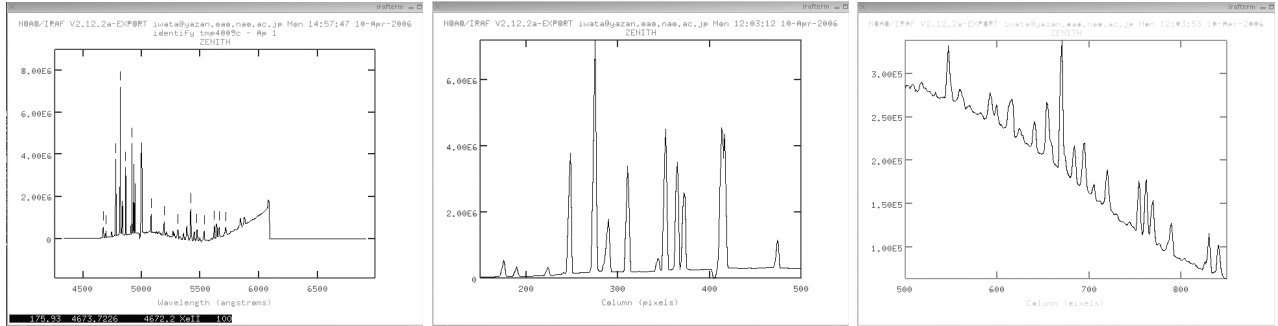


図 4: Xe ランプを VPH495 で分光したスペクトル. (左) 全体, (中) 4580–5100 \AA 付近, (右) 5200–5600 \AA 付近.

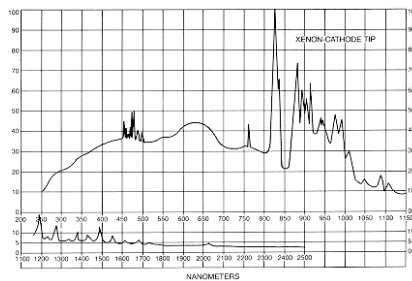


図 5: Xe ランプのスペクトル. 2,500–11,500 \AA .

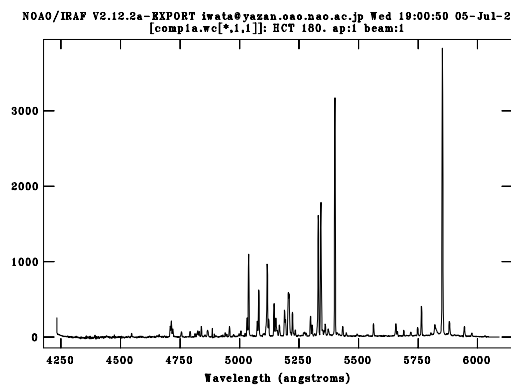


図 6: FeNe ランプを VPH495 で分光したスペクトル.

4 線形性試験

4.1 ドームフラットによる試験

4月4日に積分時間を変えながらドームフラットを撮影する方法で線形性の試験を行った。観測に参加して下さった和歌山大学の富田さんから詳細なレポートを送って頂いた。Appendix A 参照。

要点は

- 1bin data では、bias level より上 7000 count くらいまでは linear で、そこまでの gain は 2.07 [e/ADU]。ただし光源の時間安定性に問題があるようなので、確定的ではない。
- 2bin data では、bias level より上 25000 count くらいまで linear で、そこまでの gain は 2.23 [e/ADU]。ただし光源の時間安定性に問題があるようなので、確定的ではない。
- Read noise は 3.2 [ADU]、およそ 7 [e] 相当

図7では、同じ積分時間の dome flat 画像 (overscan 差し引き済み) を割ったもの。線形性に問題が生じると、flat pattern に異常が見られる。図7では 140sec 積分の binning ありのデータ、600sec の binning なしのデータで中央付近のカウントが高いところに異常がみられる。いずれも、もとのカウントでは 10^4 ADU を超えたあたりである。

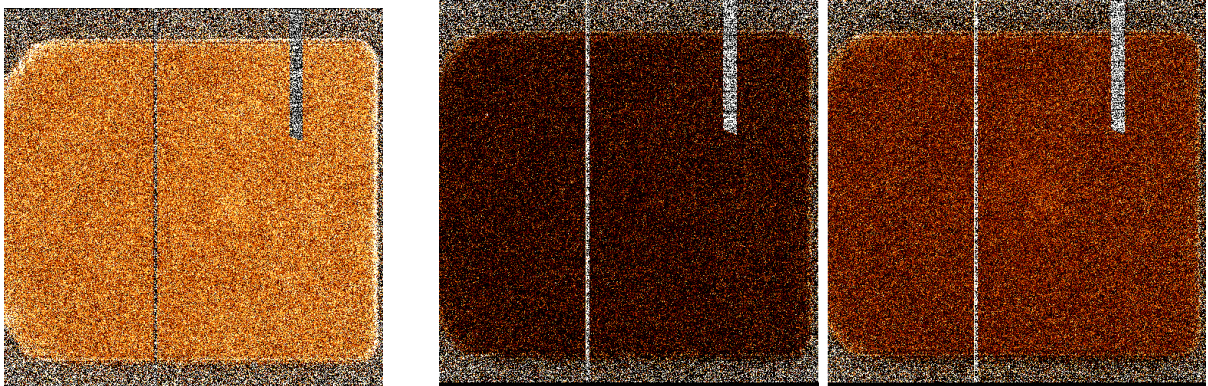


図7: 同じ積分時間の dome flat を割った画像. (左) 2×2 binning, 140sec 積分. (中) binning なし, 400sec 積分. (右) binning なし, 600sec 積分.

4.2 Shutterless Photon Transfer 法による試験

4月11日に組立調整室内でLEDを用いてShutterless Photon Transfer法による線形性試験を行った。これは、定光源を照射したまま、メカニカルシャッターを強制的に開いた状態で平行方向にoverscanを行うことで、columnごとのカウントの分散からphoton noiseを得るものである。

この実験の際、Messiaのクロックパターン(cpg.ascii, spv.asciiに記述; Appendix B参照)を変更して、ゲインを変更した場合も試してみた。具体的には、BIAS, SIGの繰り返し回数が通常10(operation_type 2)であるのに対し5(operation_type 10)にしている。

通常の読み方の場合はread_spt, read_spt.2bin, ゲインを変更した場合はread_spt.gain2, read_spt.gain2.2binを使用。

図8にbinningなしの場合の結果を示した。通常のゲインの場合は服部君が以前試したときとほぼ同様の結果で、 10^4 カウントあたりで変化している。クロックパターンを変更した場合、ゲインは大きくなっている。この場合、 6×10^3 カウントあたりで変化が起きている。

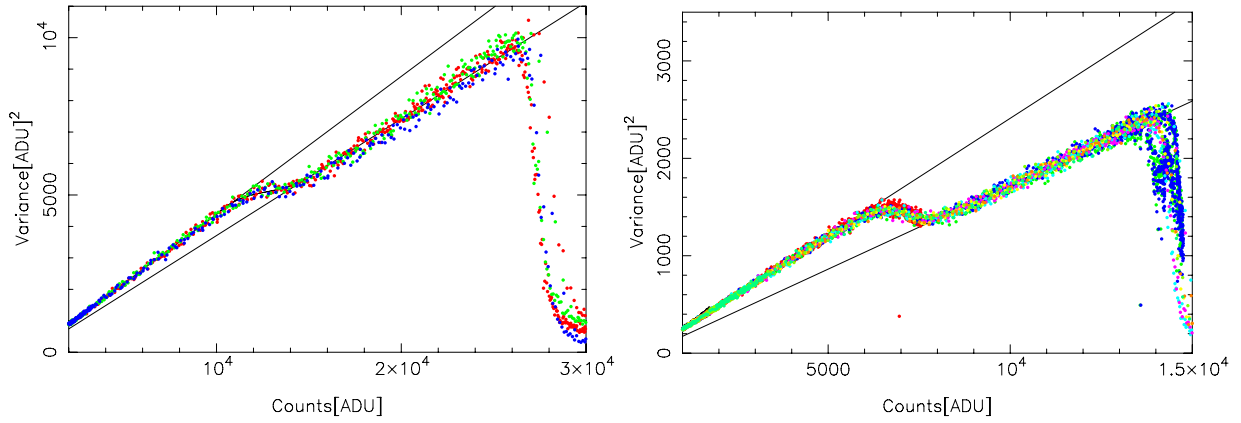


図 8: (左) spt full 上の線は $\text{gain}=2.28e^-/\text{ADU}$, 下の線は $2.63e^-/\text{ADU}$. (右) spt full (gain2) 上の線は $\text{gain}=4.15e^-/\text{ADU}$, 下の線は $5.80e^-/\text{ADU}$

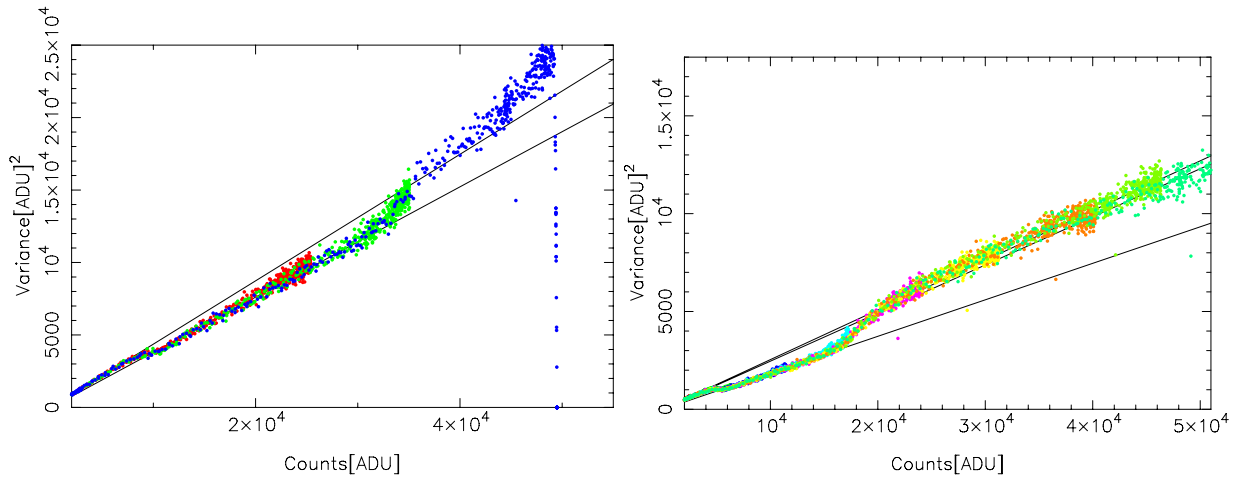


図 9: (左) spt 2bin 上の線は $\text{gain}=2.29e^-/\text{ADU}$, 下の線は $2.63e^-/\text{ADU}$. (右) spt 2bin (gain2) $< 4,000\text{ADU}$ は $\text{gain}=4.07e^-/\text{ADU}$, $6,000\text{--}16,000\text{ADU}$ は $\text{gain}=5.36e^-/\text{ADU}$, $25,000\text{--}50,000\text{ADU}$ は $3.94e^-/\text{ADU}$.

図 9 にはビンングした場合の結果を示した。通常のゲインの場合、 8×10^3 カウントあたりで変化が起こり、 3×10^4 カウントあたりで再び変化している。ゲインを変更した場合、 3×10^3 カウントあたりで変化し、 1.6×10^4 カウントあたりで再び変化している。2 回目の変化では元のゲインに近くなるが、完全に同じではないように見える。

ビンングしても同程度のカウントでゲインに異常がみられる。このデータが正しいとすれば、通常のゲインの場合 2×2 ビンングで 8×10^3 カウントあたりのゲインの変化を無視すると、14.8%の誤差となる。線形性に問題が起きない範囲で積分を行おうとすると、例えば 2×2 ビンングではビンングなしに比べて $1/4$ くらいの積分時間しかとれないことになってしまう。

4177		4	
4178		6	
4179		10	
4180		15	
4181		25	
4182		40	
4183		60	
4184		100	
4185		120	
4186		140	
4187		250	
4188		400	
4189		600	
4190	bias	0	

4192	bias	0	1bin	3つのseriesの2つ目
4193	flat	4		
4194		6		
4195		10		
4196		16		
4197		24		
4198		40		
4199		60		
4200		100		
4201		160		
4202		240		
4203		400		
4204		600		
4205		1000		
4206	bias	0		

4207	bias	0	2bin	1-600secで行う3つのseriesの3つ目
4208	flat	1		
4209		1.5		
4210		2.5		
4211		4		
4212		6		
4213		10		
4214		15		
4215		25		
4216		40		
4217		60		
4218		100		
4219		120		
4220		140		
4221		250		
4222		400		
4223		600		
4224	bias	0		

4225	bias	0	1bin	3つのseriesの3つ目
4226	flat	4		
4227		6		
4228		10		
4229		16		
4230		24		
4231		40		
4232		60		
4233		100		
4234		160		

```

4235 |      240 |
4236 |      400 |
4237 |      600 |
4238 |     1000 |
4239 bias      0 |
-----
4240 bias      0 2bin 100-350sec で行う 2 つの series の 1 つ目
4241 flat      1 |
4242 |      100 |
4243 |      110 |
4244 |      120 |
4245 |      130 |
4246 |      140 |
4247 |      150 |
4248 |      160 |
4249 |      180 |
4250 |      200 |
4251 |      220 |
4252 |      250 |
4253 |      300 |
4254 |      350 |
-----
4258 bias      0 2bin 100-350sec で行う 2 つの series の 2 つ目
4259 flat      1 |
4260 |      100 |
4261 |      110 |
4262 |      120 |
4263 |      130 |
4264 |      140 |
4265 |      150 |
4266 |      160 |
4267 |      180 |
4268 |      200 |
4269 |      220 |
4270 |      250 |
4271 |      300 |
4272 |      350 |
=====

```

```

=====
2. カウントの統計の取り方
=====

```

領域

```

1bin: light region: [410:750,110:870]
      overscan region: [1010:1040,110:870]
2bin: light region: [205:375,55:435]
      overscan region: [510:540,55:435]

```

-> light region: flat lamp がちゃんと (まんべんなく) あたっているところで、
大きな pixel defects のないところ。

-> 1bin と 2bin で、おなじ領域になるように取っている。
overscan は binning によらず 50 pixel の幅をもっている。

この領域の imstat を取って、mean, stddev を調べていく。

```

=====
3. 線形性を調べる方法
=====

```


方法 1 :

単純に、横軸露出時間、縦軸カウントを取った図から読み取る、という方法がある。
光源の強度が時間的に安定していると、そのまま線形性を見るグラフになる。

方法 2 :

似たカウントを持つ 2 枚のフレーム (同じ露出時間のフレームを使う場合が多い) を
引き算し、引き算後のフレームでのカウントの標準偏差を利用する方法がある。
横軸に元のカウント (2 枚のフレームのカウントの平均値を採用)
引き算後のフレームでのカウントの標準偏差を σ と表記して縦軸に $(1/2) \sigma^2$ を取る。
この図上のプロット点を直線でフィットすると、その直線の傾きの逆数が [e/ADU] の
単位で見た gain になる。

方法 2 の補足

を作る原因として、以下の 3 つが考えられる。

- (a) 入射光の photon noise
- (b) read noise
- (c) CCD pixel の defects によるもの

(a) が Poisson noise を持つとすると、ここから入射 photon number を推定することが
できる。(c) は、bad pixel というほどのものではなくても、pixel-to-pixel の
感度ムラの大きなものを含めて考えている。

似たカウントを持つ 2 枚のフレームを引き算して得られたフレームの σ では (c) の成分が
cancel されていると考えられる。(b) は一般には (a) に比べて十分小さいから、
引き算後のフレームの σ は、基本的には (a) が支配していると考えてもよい。

元の 1 枚のフレーム分でのカウントの標準偏差 (a の成分に対応するもの) は
 $\sqrt{1/2} \cdot \sigma$ になる。これが photon noise だから、photon number は
 $(1/2) \sigma^2$ になる。

方法 2 では、縦軸に入射光強度の時間積分、横軸にカウントを取ったものになる。
プロットが直線上に乗るかで線形性を見ることができると同時に、gain もわかる。
また似たカウントのフレームどうしの pair を作って引き算すればよく、
この pair が見つかる限りにおいては、光源の時間安定性は必要でなくなる。
方法 2 では、横軸が bias count のところの縦軸の値の sqrt が read noise を
示すことにもなる。

```
=====
4. imstat 出力を編集した text data (gnuplot 向け)
=====
```

```
1bin_raw.data: readmode が 1bin のもので、引き算前の frames の統計
2bin_raw.data: readmode が 2bin のもので、引き算前の frames の統計
1bin_sub.data: readmode が 1bin のもので、引き算後の frames の統計
2bin_sub.data: readmode が 2bin のもので、引き算後の frames の統計
```

いずれも、もと 2 つのファイルだったものを横に貼り合わせている。
行の左半分が light region の統計、右半分が overscan region の統計。

```
=====
5. 方法 1 のグラフ
=====
```

< 1bin data > 3 つの試行がある

* 引き算前の (普通の) frames に対する exp time - count のグラフ

```
data: 1bin_raw.data
gnuplot script: figure01.gp
PS figure: figure01.eps
```

2 万 count 中程あたりで予想通り折れているように見える。

それより、3回の試行で系統が違う。光源の強度が時間安定していないのだろう。
1回のシリーズでは明るさに違いはない、としてもいいだろうが、ここまで系統が違うと方法1を素直に受け入れていいのか、ちょっと不安である。
特に1回目は光源強度が低いようで、まだランプが暖まっていなかったのだろう。
2回目と3回目はかなり似ているので、方法2ではこの後二者の比較になるだろう。

* 引き算前の(普通の)framesに対する mean - std dev のグラフ

```
data: 1bin_raw.data
gnuplot script: figure02.gp
PS figure: figure02.eps
```

bias levelとして15484 countとし、count above bias levelとstd devを見た。
15484 countは、bias frames(6枚)のlight regionでsampleしたmeanの平均値。
光源の強度安定性の問題によらず、3系統ともほぼ重なっている(予想通り)。
std devがmeanに対してsqrt的ではなくlinearに近い。
gain(e/ADUの単位で)がものすごく小さい値ならともかく(それは考えにくいので)の原因(a)(b)(c)のうち、(c)の影響が大きいことを物語っている。

* 念のため、overscan領域の統計を見ておく

```
data: 1bin_raw.data
gnuplot script: figure03.gp
PS figure: figure03.eps
```

light regionとしてsampleした領域のbias levelは、overscan regionでの値と(たまたま)系統的にずれているようである。
light regionでのcountが高いと、わずかではあるがoverscan regionでの値が上がるのが気になる。1系統目から3系統目にかけての変化が見られないことから、これはCCD特性の時間変化によるものとは考えにくい。
最初の読み出しでphotonを吸い出しきっていないのだとすると、overscanの時にその残骸が出てしまう、という可能性もある。
しかし、あまり気にしなくてもいいような気がする。

< 2bin data > 1-600secで行った3つの試行と、100-350secで行った2つの試行がある

* 引き算前の(普通の)framesに対する exp time - count のグラフ

```
data: 2bin_raw.data
gnuplot script: figure04.gp
PS figure: figure04.eps
```

1binの場合と同じく、2万count中程あたりで折れているように見え... ないな。
結構素直な直線だ。
1回目の試行だけcountがかなり低い。他の系統は互いに似たcountになっている。
光源の強度が最初は暗く、しばらくして安定していったのか。

* 引き算前の(普通の)framesに対する mean - std dev のグラフ

```
data: 2bin_raw.data
gnuplot script: figure05.gp
PS figure: figure05.eps
```

bias levelとして16039.5 countとし、count above bias levelとstd devを見た。
16039.5 countは、bias frames(10枚)のlight regionでsampleしたmeanの平均値。
しかし1binと2binでbias levelが微妙に違うのはなぜだろう。差は500 count強だし...
ここでもstd devがmeanに対してsqrt的ではなくlinearに近い。
原因(a)(b)(c)のうち、(c)の影響が大きいことを物語っている。
この図では、大きなcountのところでは上側に反り始めている。
これはphoton numberの非常に大きな場合にCCD countの上昇が鈍ることに対応している

のだろう（つまり線形性が悪くなっている）。

* 念のため、overscan 領域の統計を見ておく

```
data: 2bin_raw.data
gnuplot script: figure06.gp
PS figure: figure06.eps
```

light region として sample した領域の bias level は、overscan region での値と 2bin でも（たまたま）系統的にずれているが、差はずっと小さくなっている。1bin のデータで見られた、light region での count が高いと、わずかではあるが overscan region での値が上がる、ということは見られない（むしろ逆か）。最初の読み出しで photon を吸い出しきっていないのだとすると、overscan の時にその残骸が出てしまう、という可能性もなさそうだ。結局、あまり気にしなくてもいいようだ。

元の 4 pixel まとめてだから、2bin データの方が落ち着いている、のだろう。

=====

6. 方法2のグラフ

=====

overscan region は、以下では気にしないことにする（せっかく統計をとっているが、とりあえず）。

< 1bin data >

3つの試行があるので、差の取り方として：

1 - 2、1 - 3、2 - 3

の3つがある。ただし、1回目は光源強度が弱かったようなので注意。

* average count - std dev のグラフ

```
data: 1bin_sub.data
gnuplot script: figure07.gp
PS figure: figure07.eps
```

average count としては、ここからさらに bias level の 15484 を引いてある。std dev として挙げたものは、 $(1/2) * \sigma^2$ のことである。やはり (1 - 2)、(1 - 3) は「変」である。(2 - 3) をとりあえず信用することにしよう。

横軸右から3点目まで、よく linear になっている。

この点は、average count = 22453（ここから bias level of 15484 を引く）

std dev = 82.09 $\rightarrow (1/2) * \sigma^2 = 3369.4$

ここから gain = 2.07 [e/ADU] となる。

Bias level より 7000 count くらいまで gain = 2.07 で linear、それを超えると gain が大きくなるような感じ（感度が鈍る sense）になる。

服部氏のデータ：

<http://www.oao.nao.ac.jp/support/staff/hattori/kools/ccd/spt1.png>
と比べると、

linearity の折れ曲がり早い（服部氏のデータでは 10000 count くらい）。

15000 count くらいまでは「遷移」状態のところなので、

方法2のグラフでは傾きがかなり小さくなることになる。この点は再現。

また服部氏のデータでは、10000 count 以下では gain = 2.25 と記されており、この値も若干違う（違いはわずか、ではないな）。

光源の時間安定性の問題もあるので、服部氏のデータと inconsistent とはいえない。きっと consistent な結果なのだろう。

< 2bin data >

露出時間のセットが2種類あり、それぞれ3つ、2つの試行がある。
差の取り方として：

1 - 2、1 - 3、2 - 3、4 - 5

の4つがある。ただし、1回目は光源強度が弱かったようなので注意。

data: 2bin_sub.data
gnuplot script: figure08.gp
PS figure: figure08.eps

average count としては、ここからさらに bias level の 16039.5 を引いてある。
std dev として挙げたものは、 $(1/2) * \sigma^2$ のことである。
(1 - 2) (1 - 3) が「変」なことは分かっているので、最初から除外。
(2 - 3) (4 - 5) だけ表示した。

「折れ曲がり」は見えなくもないが、はっきりしない。
あるとすると 10000 count より小さめ、5000-7000 count あたりとも読める。

25000 count くらいまでそれなりに素直な直線。
average count (bias level を引く前の値) = 41851
std dev (引き算後の frame での値) = 152.3
の点まで直線だとすると、gain = 2.23 になる。
服部氏が出していた、低い count での gain の値 2.25 とほぼ一致。う~む。

=====
7. Read noise
=====

< 1bin data >

Bias frame すべてを使う。

* average count - std dev のグラフ

data: 1bin_sub.data
gnuplot script: figure09.gp
PS figure: figure09.eps

average count としては、ここからさらに bias level の 15484 を引いてある。
std dev として挙げたものは、 $(1/2) * \sigma^2$ のことである。
特に奇異な特徴はない。
縦軸の値の単純平均を取ると 13.34、その平方根 3.65 [ADU] が read noise に相当。
KOOLS web site にあげられている 2.5 - 3 [ADU] より大きい...
経年変化かな。

念のため、overscan region の sample で同じものを作成。

data: 1bin_sub.data
gnuplot script: figure10.gp
PS figure: figure10.eps

縦軸の値の単純平均を取ると 10.55、その平方根 3.25 [ADU] が read noise に相当。
上で求めた値より若干小さい。
Bias frame とはいえ、前の exposure の electron 残りがあって、overscan の時には
「完璧に」bias になる、のかな。そうするとこちらの統計の方が信頼性大かも。
いずれにしても、KOOLS web site での値より大きいことには違いない。

< 2bin data >

こちらでも bias frame すべてを使う。

* average count - std dev のグラフ

```
data: 2bin_sub.data
gnuplot script: figure11.gp
PS figure: figure11.eps
```

average count としては、ここからさらに bias level の 16039.5 を引いてある。
std dev として挙げたものは、 $(1/2) * \sigma^2$ のことである。
妙なグラフである。

Bias frame は全部で 10 枚ある。露出のシリーズが 5 回、それぞれの最初と終わりに 1 枚ずつ入れている。

どうもこのうち 2 枚で、たまたま変な pixel があったようだ。

それは kls004240 と kls004254 のようで、これと pair を作ってしまったもの
合計 17 枚分のデータが上方に「飛んで」いる。

下に沈んでいるものが使えるものになるが、bias level に時間変動があるようで、
5 つのグループに分かれてしまっている。

これはこれでおいておいて、overscan region の sample で同じものを作成。

```
data: 2bin_sub.data
gnuplot script: figure12.gp
PS figure: figure12.eps
```

ここでも 1 枚、妙なのがある。原因は kls004207 のようだ。

それと pair になっている 9 枚を除いて、縦軸の値の単純平均を取ると 9.80、
その平方根 3.13 [ADU] が read noise に相当。こんなもんかな。

Read noise は、overscan region で sample した値を使えば、1bin でも 2bin でも
ほぼ一致する値になった（そうなら困る）。

Read noise = 3.2 [ADU] というところだろう。

gain = 2.25 [e/ADU] (服部氏による 10000 count 以下での値を採用) とすると、
read noise は 7.2 [e] ということになる。

8. 積み残し

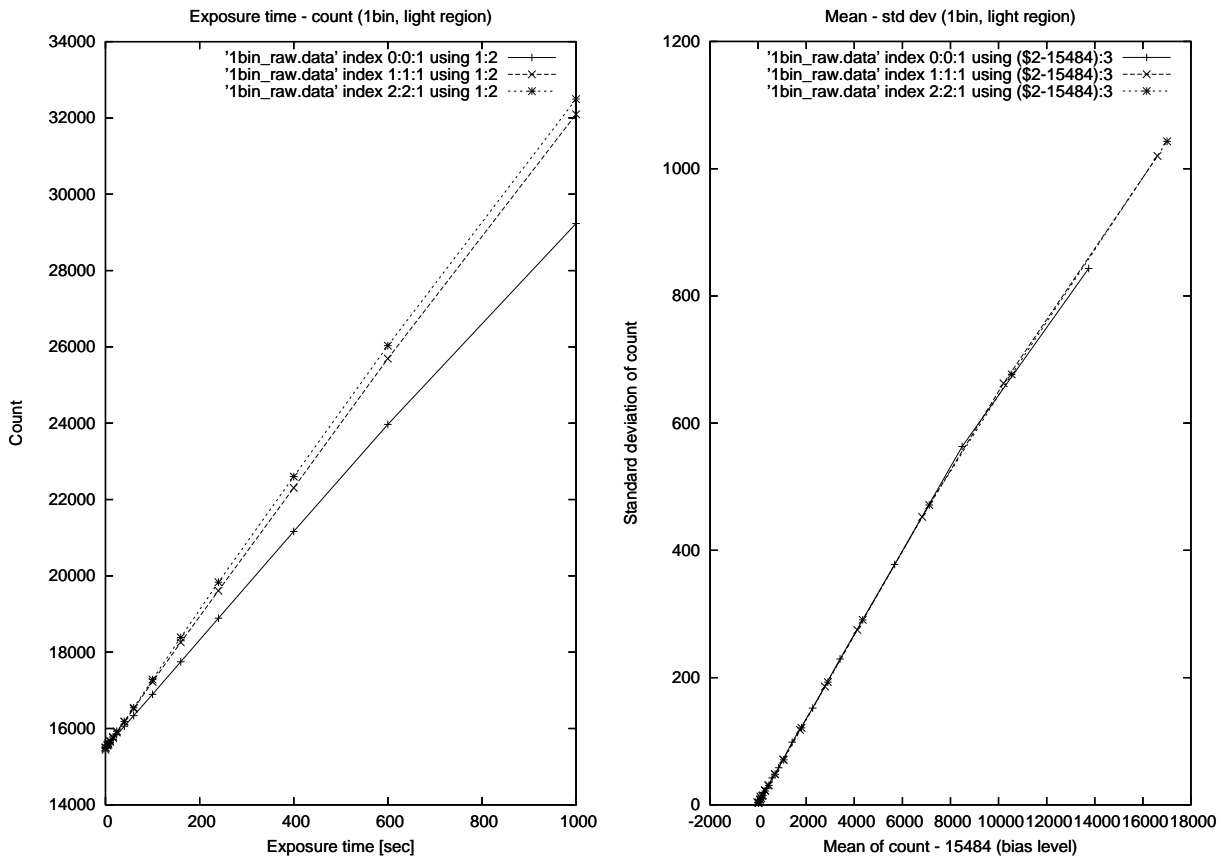
グラフ上で直線と読み取るところがあれば、直線からのずれを計算して、ずれのグラフを作成
すべきだが、そこまで作業しなかった（横着）。

9. まとめ

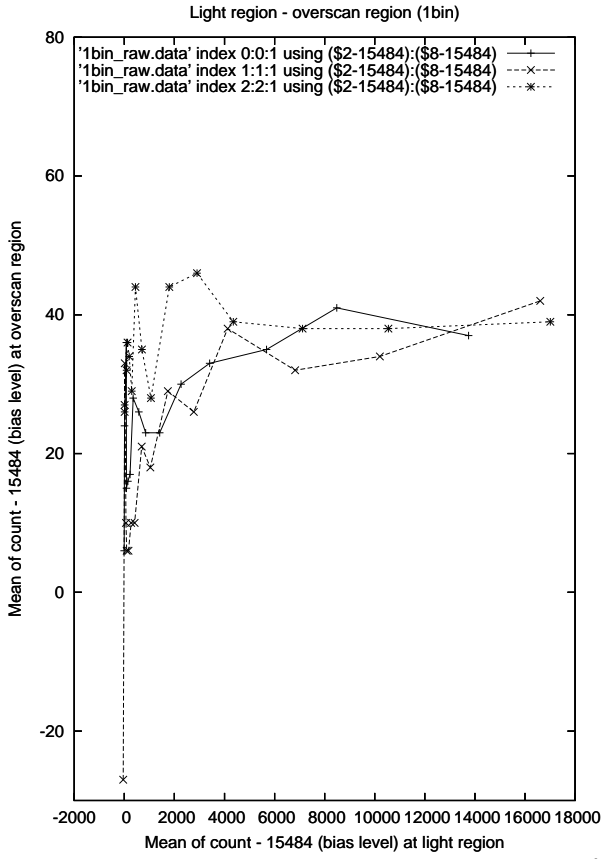
1. 1bin data では、bias level より上 7000 count くらいまでは linear で、
そこまでの gain は 2.07 [e/ADU]。
ただし光源の時間安定性に問題があるようなので、確定的ではない。
2. 2bin data では、bias level より上 25000 count くらいまで linear で、
そこまでの gain は 2.23 [e/ADU]。
ただし光源の時間安定性に問題があるようなので、確定的ではない。
3. 2bin data は元の pixel で見ると $2 \times 2 = 4$ pixels 分の binning なので、
linearity を保証してくれる count は、1bin の場合の 4 倍、でいいのかな。
そうすると、7000 の約 4 倍=25000 で勘定があう。
もしそうなら、dynamic range を大きく使うという点では、
1bin での観測はかなり損である。
3bin なら、ほぼ 16 bit の幅全部が使える。ただし、1 pixel = 1 arcsec

になるので、この点できつい。Seeingの努力をしてきた岡山の特長をつぶしてしまう。

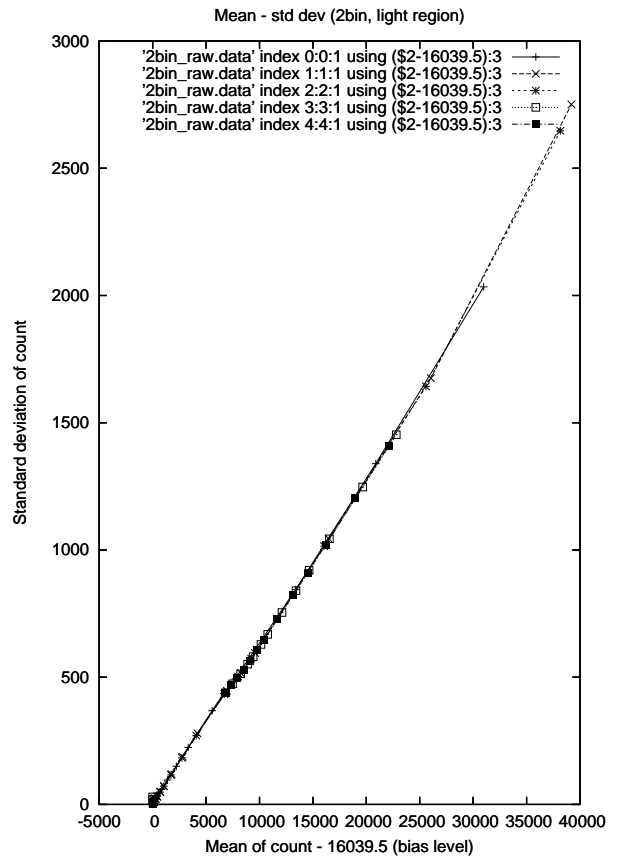
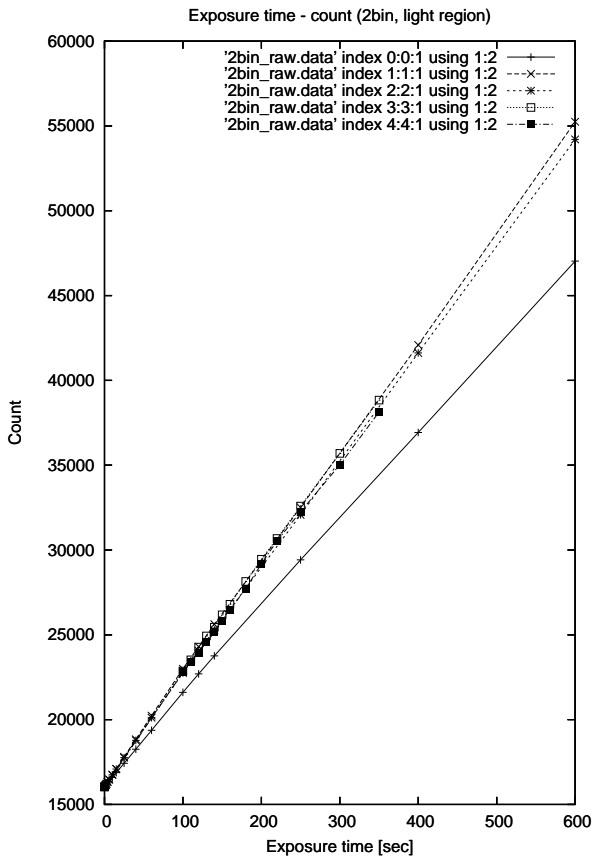
- 1bin dataと2bin dataでgainが違うのは(?)なので、実際は同じ値を返してくるのでしょう。2bin dataにやや重みを置くと、2.2 [e/ADU] あたりか。
- Read noiseは3.2 [ADU]、およそ7 [e] 相当、だろう。



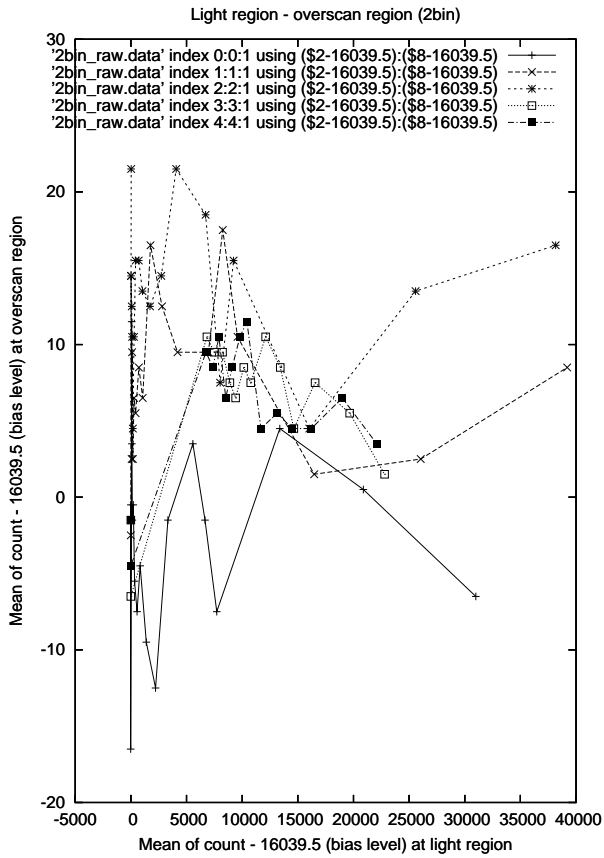
富田レポート:figure01.eps, figure02.eps



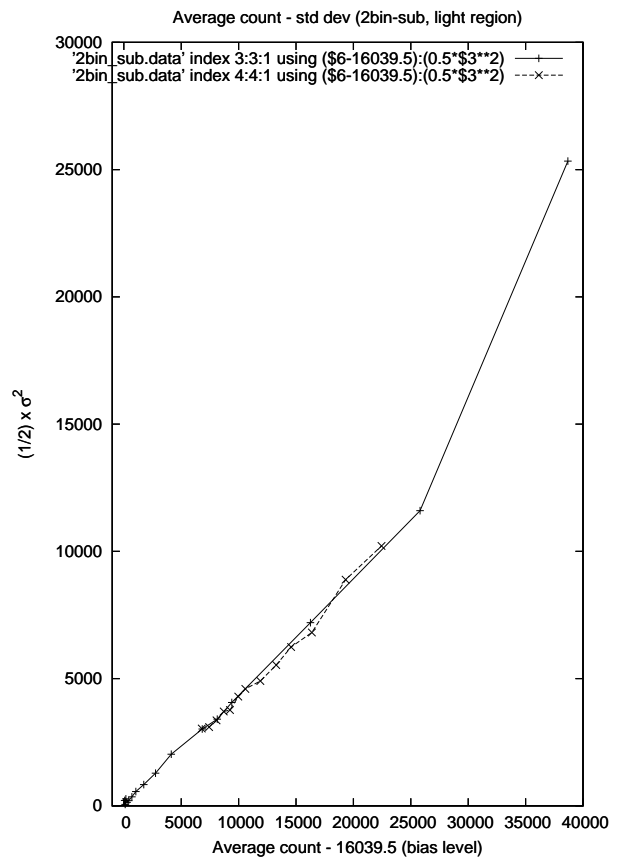
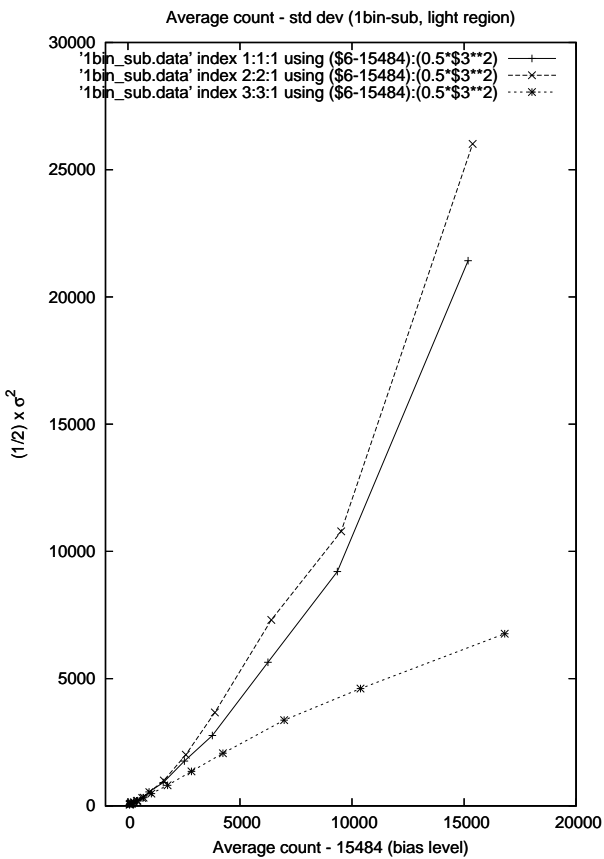
富田レポート:figure03.eps



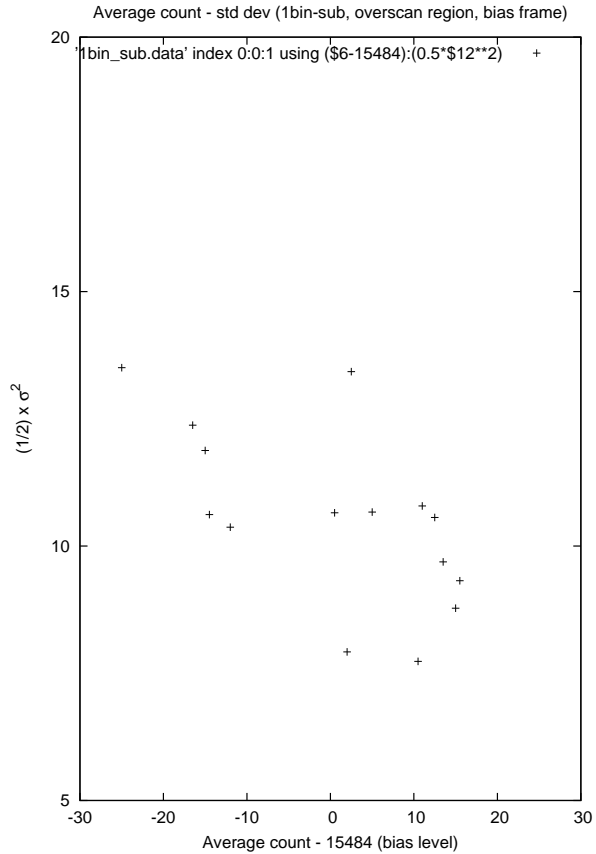
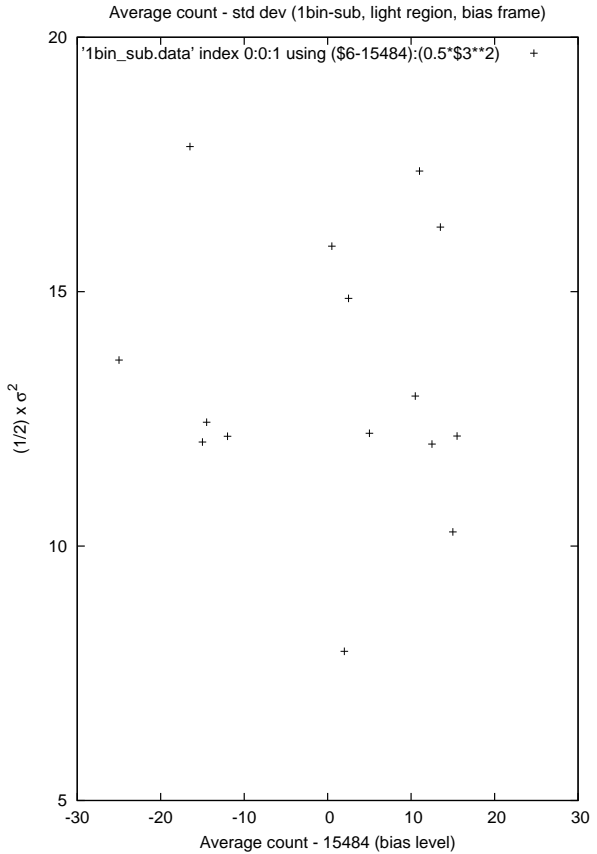
富田レポート:figure04.eps, figure05.eps



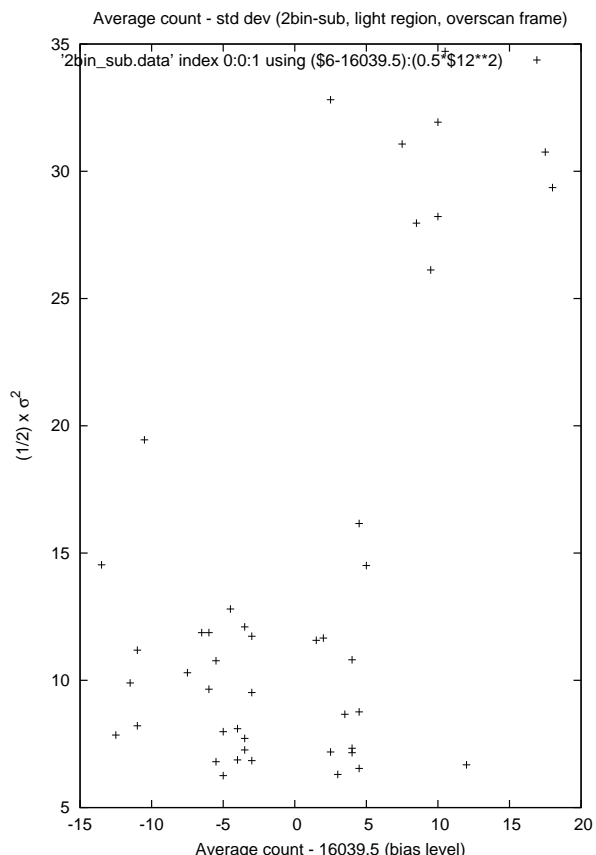
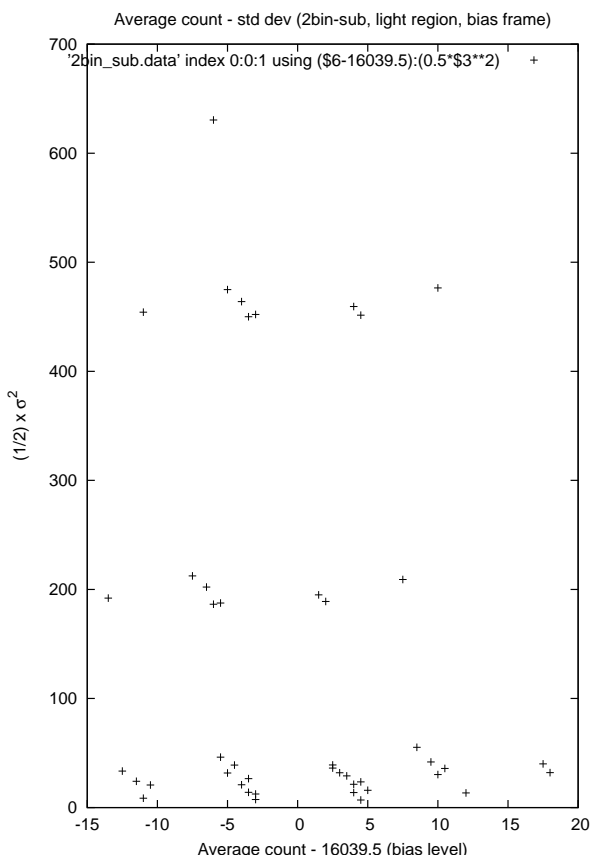
富田レポート:figure06.eps



富田レポート:figure07.eps, figure08.eps



富田レポート:figure09.eps, figure10.eps



富田レポート:figure11.eps, figure12.eps

B messia クロックパターン記述ファイル

B.1 cpq.ascii(抜粋)

```
#####
set_clock_tick 1 # numebr of 400nsec clock ticks in one time bin
operation_type 2 # Serial shift
# M-Front Channel name
# S S S S D P P P P S I R B S R C T
# 1 2 3 X G 1 2 3 X W P G I I S O R
# C A G T N G
# S V
start 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 31
t 1 ] | | | | ] ] | | | | ] | | | | ]
t 1 ] | | | | ] ] | | | | ] | | | | |
t 14 ] | | | | ] ] | | | ] | | | | |
t 1 ] | ] | | ] ] | | | ] | | | | |
t 1 | | ] | | ] ] | | | ] | | | | |
t 1 | ] ] | | ] ] | | ] ] | | | | |
t 1 | ] | | | ] ] | | ] ] | | | | |
t 1 ] ] | | | ] ] | | ] ] | | | | |
t 4 ] | | | | ] ] | | ] ] | | | ] | |
t 1 ] | | | | ] ] | | ] | | | | | |
t 10 ] | | | | ] ] | | ] | | | | |
t 1 ] | | | | ] ] | | ] | | | | |
t 3 ] | | | | ] ] | | | | | | | |
t 10 ] | | | | ] ] | | | | | ] | |
t 1 ] | | | | ] ] | | | | | | | |
t 1 ] | | | | ] ] | | | | | | | ] |

#####
set_clock_tick 1 # numebr of 400nsec clock ticks in one time bin
operation_type 7 # Serial shift without dizitization. start binning
# M-Front Channel name
# S S S S D P P P P S I R B S R C T
# 1 2 3 X G 1 2 3 X W P G I I S O R
# C A G T N G
# S V
start 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 31
t 1 ] | | | | ] ] | | | | ] | | | | ]
t 1 ] | | | | ] ] | | | | ] | | | | |
t 10 ] | | | | ] ] | | | ] | | | | |
t 1 ] | ] | | ] ] | | | ] | | | | |
t 1 | | ] | | ] ] | | | ] | | | | |
t 1 | ] ] | | ] ] | | ] ] | | | | |
t 1 | ] | | | ] ] | | ] ] | | | | |
t 1 ] ] | | | ] ] | | ] ] | | | | |
t 1 ] | | | | ] ] | | ] ] | | | | |
end

#####
set_clock_tick 1 # numebr of 400nsec clock ticks in one time bin
operation_type 9 # Serial shift. end of binning
# M-Front Channel name
# S S S S D P P P P S I R B S R C T
# 1 2 3 X G 1 2 3 X W P G I I S O R
# C A G T N G
# S V
start 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 31
t 1 ] | | | | ] ] | | ] ] ] | | | | ]
```

```

t 1 ] | | | | ] ] | | ] ] | | | | | |
t 10 ] | | | | ] ] | | ] ] | | | | | |
t 1 ] | ] | | | ] ] | | ] ] | | | | | |
t 1 | | ] | | | ] ] | | ] ] | | | | | |
t 1 | ] ] | | | ] ] | | ] ] | | | | | |
t 1 ] ] | | | | ] ] | | ] ] | | | | | |
t 4 ] | | | | ] ] | | ] ] | | | | ] | |
t 1 ] | | | | ] ] | | ] | | | | | | |
t 10 ] | | | | ] ] | | ] | | ] | | | |
t 1 ] | | | | ] ] | | ] | | | | | | |
t 3 ] | | | | ] ] | | | | | | | | | |
t 10 ] | | | | ] ] | | | | | | | ] | |
t 1 ] | | | | ] ] | | | | | | | | | |
t 1 ] | | | | ] ] | | | | | | | | ] |
end

```

```

#####
set_clock_tick 1 # numebr of 400nsec clock ticks in one time bin
operation_type 10 # Serial shift, bias,sig time=5
# M-Front Channel name
# S S S S D P P P P S I R B S R C T
# 1 2 3 X G 1 2 3 X W P G I I S O R
# C A G T N G
# S V
start 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 31
t 1 ] | | | | ] ] | | | | ] | | | | ]
t 1 ] | | | | ] ] | | | | | | | | | |
t 14 ] | | | | ] ] | | | | ] | | | | |
t 1 ] | ] | | | ] ] | | | | ] | | | | |
t 1 | | ] | | | ] ] | | | | ] | | | | |
t 1 | ] ] | | | ] ] | | | | ] | | | | |
t 1 | ] | | | | ] ] | | | | ] | | | | |
t 1 ] ] | | | | ] ] | | | | ] | | | | |
t 4 ] | | | | ] ] | | | | ] | | | | ] | |
t 1 ] | | | | ] ] | | | | ] | | | | | |
t 5 ] | | | | ] ] | | | | ] | | ] | | | |
t 1 ] | | | | ] ] | | | | ] | | | | | |
t 3 ] | | | | ] ] | | | | | | | | | |
t 5 ] | | | | ] ] | | | | | | | ] | | | |
t 1 ] | | | | ] ] | | | | | | | | | |
t 1 ] | | | | ] ] | | | | | | | | ] |

```

```

#####
set_clock_tick 1 # numebr of 400nsec clock ticks in one time bin
operation_type 11 # Serial shift. end of binning, bias,sig time=5
# M-Front Channel name
# S S S S D P P P P S I R B S R C T
# 1 2 3 X G 1 2 3 X W P G I I S O R
# C A G T N G
# S V
start 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 31
t 1 ] | | | | ] ] | | ] ] ] | | | | ]
t 1 ] | | | | ] ] | | ] ] | | | | | |
t 10 ] | | | | ] ] | | ] ] | | | | | |
t 1 ] | ] | | | ] ] | | ] ] | | | | | |
t 1 | | ] | | | ] ] | | ] ] | | | | | |
t 1 | ] ] | | | ] ] | | ] ] | | | | | |
t 1 | ] | | | | ] ] | | ] ] | | | | | |

```

```

t 1 ] ] | | | ] ] | | ] ] | | | | | |
t 4 ] | | | | ] ] | | ] ] | | | ] | |
t 1 ] | | | | ] ] | | ] | | | | | | |
t 5 ] | | | | ] ] | | ] | | ] | | | |
t 1 ] | | | | ] ] | | ] | | | | | | |
t 3 ] | | | | ] ] | | | | | | | | | |
t 5 ] | | | | ] ] | | | | | | ] | | |
t 1 ] | | | | ] ] | | | | | | | | | |
t 1 ] | | | | ] ] | | | | | | | | ] |
end

```

B.2 spv.ascii(spt 用のみ抜粋)

```

begin read_spt
  loop2_start 8192
  ccd_operation 0 1
  loop3_start 25
  ccd_operation 0 3
  loop3_continue
  loop1_start 2098
  ccd_operation 0 2
  loop1_continue
  loop2_continue
end

```

```

begin read_spt.2bin
  loop2_start 4096
  ccd_operation 0 1 2
  ccd_operation 0 3 25
  loop1_start 1074 #1024+overscan50
  ccd_operation 0 7 #benning start
  ccd_operation 0 9 #benning end
  loop1_continue
  loop2_continue
end

```

```

begin read_spt.gain2
  loop2_start 8192
  ccd_operation 0 1
  loop3_start 25
  ccd_operation 0 3
  loop3_continue
  loop1_start 2098
  ccd_operation 0 10
  loop1_continue
  loop2_continue
end

```

```

begin read_spt.gain2.2bin
  loop2_start 4096
  ccd_operation 0 1 2
  ccd_operation 0 3 25
  loop1_start 1074 #1024+overscan50
  ccd_operation 0 7 #benning start
  ccd_operation 0 11 #benning end
  loop1_continue
  loop2_continue
end

```