

検出器のステータス：98年1月

本原顕太郎

1998年1月22日

1 検出器の現状

誤って切断してしまったコドラント4のボンディングワイヤーの修理が完了した。

98/01/10~01/13にかけて CISCO にこの修理の完了した science grade の検出器を取り付け、各種の試験を行なった。

2 読み出しノイズ

2.1 条件

読み出しノイズは以下の5つの条件のもとでフレームを20枚、あるいは21枚取得して算出した。

filename	suffix	温度	積分時間	状況
/data/980109/dark	0001~0020	常温	2 sec	シールド結線せず
/data/980109/dark	0021~0040	常温	10	シールド結線せず
/data/980113/dark	0054~0073	77K	5	シールド結線済 冷え切った直後
/data/980113/dark	0143~0163	77K	5	シールド結線済 box に蓋をした
/data/980113/dark	0164~0184	77K	5	シールド結線済 CH off にした

2.2 結果

2.2.1 1次元化したイメージ

図1, 2に単純にフレームを縦横方向に1次元化したグラフを、図3, 4に平均フレームを引いて1次元化したグラフを示す。

まず、単純に1次元化したフレームであるが、

- 常温で横方向に1次元化すると2段に分かれているのが分かる。常温ではフレームに1ピクセル毎の縞が現れ、それがこのように見えている。平均フレームを引いたあともこの縞が少し残っている(図2)ところを見ると、常温での安定度も1ピクセル毎に違うらしい。岩室さんによると NICMOS 系の検出器はそうように作られているらしい。(本当か?)
- 冷却した状態での読み出しは(ボックスに蓋をすると)、以前よりもより安定したようだ。ただし、フレームの途中で段がついて、さらにフレームに傾きがつく症状は相変わらず。また、その傾きが安定するまでには3フレームくらいかかる。(実際の観測では空読みを3フレーム程度入れる必要がある。)
- Amp/ADC ボックスに蓋をすることにより、読み出しが極めて安定するようになったのが分かる。蓋は重要。

2.2.2 読み出しの揺らぎ1

いつものように、20 枚の stddev フレームを作る。生のままのフレームと、一部領域を切り出してその傾きを fit したフレームの stddev を求める。

切り出した領域は [663:1012,151:500], [151:500,151:500], [151:500,663:1012], [663:1012,663:1012] (それぞれ順に、コドラント 1, 2, 3, 4 に対応する)。

求めた stddev フレームを IRAF の imstat コマンドにかけた。但し、おかしいピクセルの影響を除くため、upper=20 で統計をとるピクセルに制限をかけている。その結果を表 1 に示す。それぞれ、上の段が生フレームの mode、下の段がフィットしたフレームの mode。誤差は stddev の値である。

2.2.3 読み出しの揺らぎ2

次に、いつもとは違った方法で読み出しの揺らぎを求めてみる。

通常は 20 フレームを取得して、それぞれのピクセルについて stddev を求めている。しかしながら、この方法では手軽にノイズを見積もることができない。そこで今、それぞれのピクセルの読み出しの性質の違いは読み出し回路のノイズに比べて十分に小さいと考える。(少なくとも今のノイズレベルではこれは悪くない近似である。) そうすると、全ピクセルを母集団として統計をとることにより、読み出しノイズを求めることができる。

では実際にどのようにこのノイズを定義したかというと

1. 2 枚のフレームの同じ領域を切り出してフレームを 2 枚作る。
2. 2 枚を引き算する
3. それを imstat して stddev を求める
4. stddev を $\sqrt{2}$ で割った値を読み出しノイズと定義

この作業を 77K で連続して取得した 20 枚の隣同士のフレームについて行なった。結果を図 5 に示す。

大体以前の方法と同じくらいの値であるが、全体的に大きい値を返している。また、コドラント 1 だけが比較的不安定で、大きめの値になっているのが分かる。フレームの一部をフィットする以前の方法だと他のコドラントと変わらないことを考えると、変なピクセルが多いことを反映しているのか？ 良く分からない。

2.2.4 読み出しの揺らぎ3

さらにもう一つの方法に、単純にフレームの傾きを imsur で取り除いてその stddev を読み出しの揺らぎと定義する方法もある。しかしながら、現在使っている reset_scan というクロックではそれぞれの列

Quad	0001 ~ 0020	0021 ~ 0040	0054 ~ 0073	0143 ~ 0163	0164 ~ 0184
1	6.026 ± 1.262	5.336 ± 0.830	7.986 ± 1.397	7.709 ± 1.531	7.025 ± 1.427
	6.062 ± 0.9149	4.349 ± 0.7373	6.900 ± 1.324	6.668 ± 1.234	6.216 ± 1.175
2	5.372 ± 0.8355	7.736 ± 0.9625	9.091 ± 1.557	7.689 ± 1.539	7.015 ± 1.446
	4.639 ± 0.7584	4.313 ± 0.742	7.311 ± 1.467	6.641 ± 1.233	6.338 ± 1.136
3	5.216 ± 1.004	7.824 ± 0.8871	11.530 ± 1.501	7.585 ± 1.460	6.702 ± 1.307
	4.221 ± 0.7376	4.001 ± 0.6738	6.160 ± 1.212	6.299 ± 1.155	5.862 ± 1.093
4	5.591 ± 0.9748	7.743 ± 0.9704	13.700 ± 1.668	7.873 ± 1.423	7.320 ± 1.320
	4.847 ± 0.8084	4.167 ± 0.7423	6.835 ± 1.262	6.706 ± 1.220	6.488 ± 1.152

表 1: 各種条件での読み出しの揺らぎ。IRAF の imstat upper=20 で得られた結果。それぞれの上の段が単純 stddev フレームの mode、下の段がそれぞれのコドラントの [151:500,151:500] を 2 次のスプラインでフィットしたもので作った stddev フレームの mode。誤差は stddev。

の reset をするごとに他の列の電荷量に影響を与えている疑いがあり（後述）、この方法を用いる意味が余りなさそうなので、今回はこの解析は行なわない。

改良したクロックを用いたときに試してみる価値がありそうである。

3 読み出しの方法の良不

以前からダークカレントを測定する時などに気付いてはいたが、読み出しの方法で reset_scan - scan - scan として読み出して、前半の 2 フレームを引き算して得られたフレームと後半の 2 フレームを引き算して得られたフレームではイメージが違う。

すなわち、reset_scan-scan というクロックで得られたイメージには書くコドラントの下の方に横に大きく線が入ってしまうのである。

以前は、これがコドラント 4 が死んでいる生であると考えていたが、修理しても直っと思っていたがそうではなく、結局、原因は reset_scan というクロックにあった。即ち、この方法では各行ごとにリセットをかけた後読み出しているが、この「リセットをかける」という行為によって、リセットをかけた行に隣接する行のピクセルに貯められた電荷の量に影響が出てしまうらしい（McLean のゼミ本）。

実際にこの違いを見るために行なった読み出しを図 6 に示す。いずれも積分時間は 20 秒、温度 77K、フィルタ K'、スリットを半分だけ空けて 273K の恒温槽を見せている。

上の画像は reset_scan-scan で得たフレーム。各コドラント途中で線が入って、さらに画面が 10ADU 程度傾いている。また、20 枚連続してとった時には 2 枚目以降のフレームの傾きがはるかに大きくなって、100ADU に達してしまう (図 2)。

下の画像は reset_scan-scan-scan とクロックを送って、後の 2 枚を使ってフレームを作ったもの。ただしそれぞれのクロックを送る間隔は等しく 20 秒にしてある。この場合、先ほどのような線は見事に消えている。このことから、reset をかけることによってたまっている電荷が変わることがわかる。

また、フレームの傾きもほとんどなくなり、コドラントの境目を見つけるのも難しいくらいになっている。しかしながら、この傾きはクロックを送る間隔を等間隔にしないと依然として残る。この原因は特定されていないが、温度変化なのだろうか？さらに先ほどのように 20 枚連続してとると 2 枚目以降のフレームの傾きがどのようになるかを調べる必要がある。

いずれにせよ reset を分離した読み出しのクロックを至急作る必要がある。

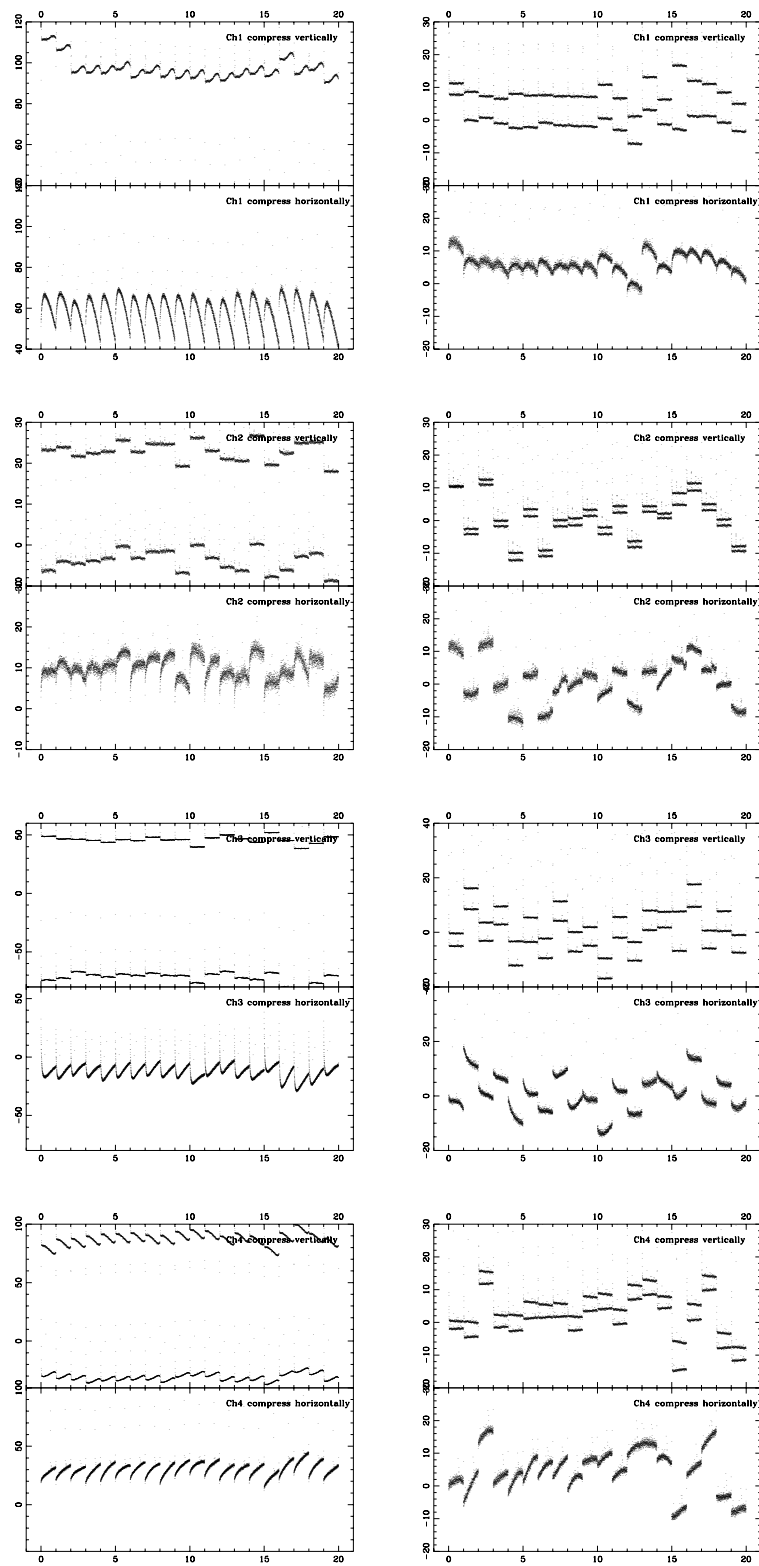


図 1: 常温で取得した全フレームを縦方向と横方向に2次元化して並べたもの。上段から順に、コドラント 1, 2, 3, 4。左が 0001 ~ 0020, 右が 0021 ~ 0040。

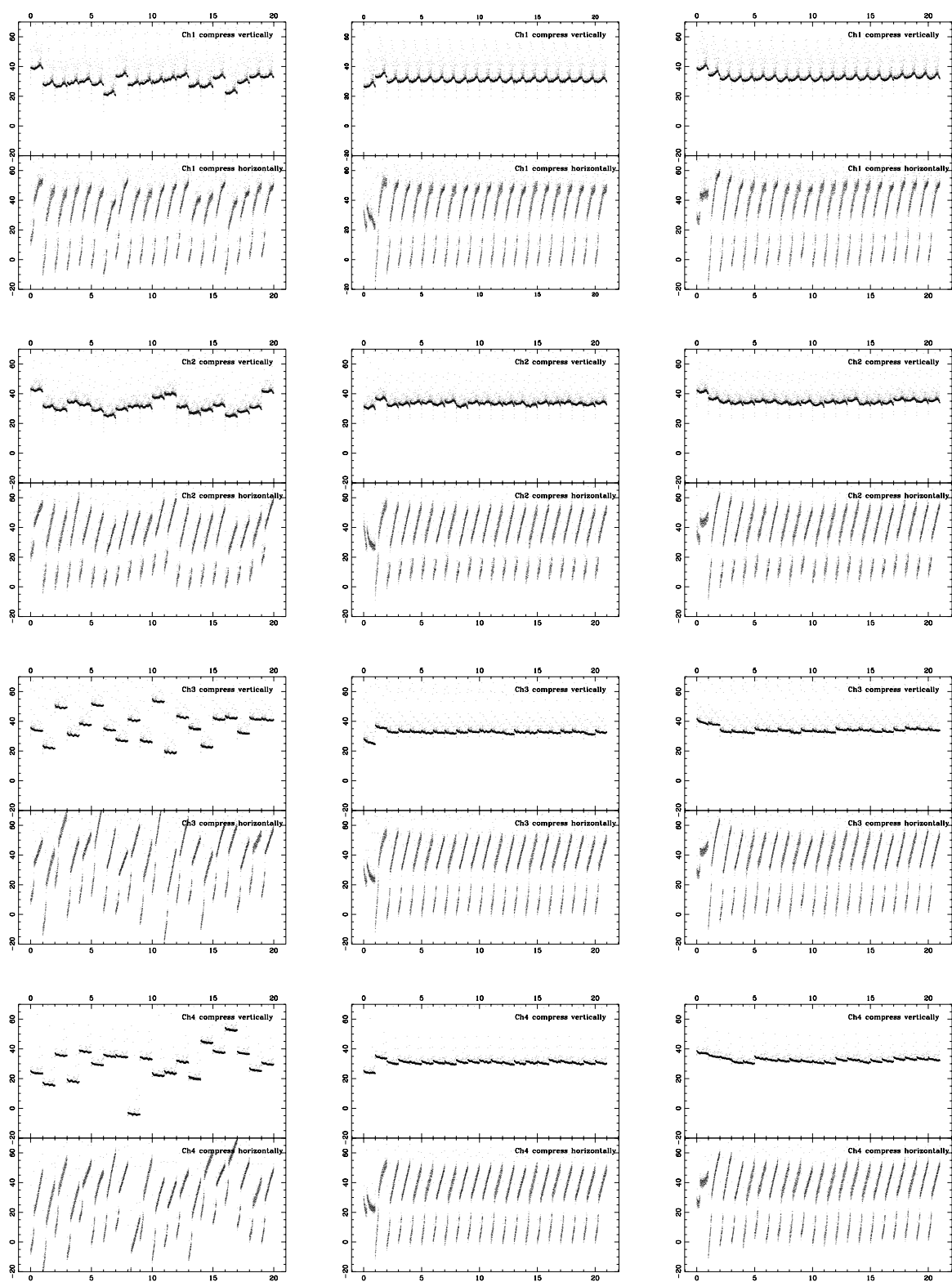


図 2: 77K で取得した全フレームを縦方向と横方向に 2 次元化して並べたもの。上段から順に、コドラント 1, 2, 3, 4。左から順に 0054 ~ 0073, 0143 ~ 0163, 0164 ~ 0184。

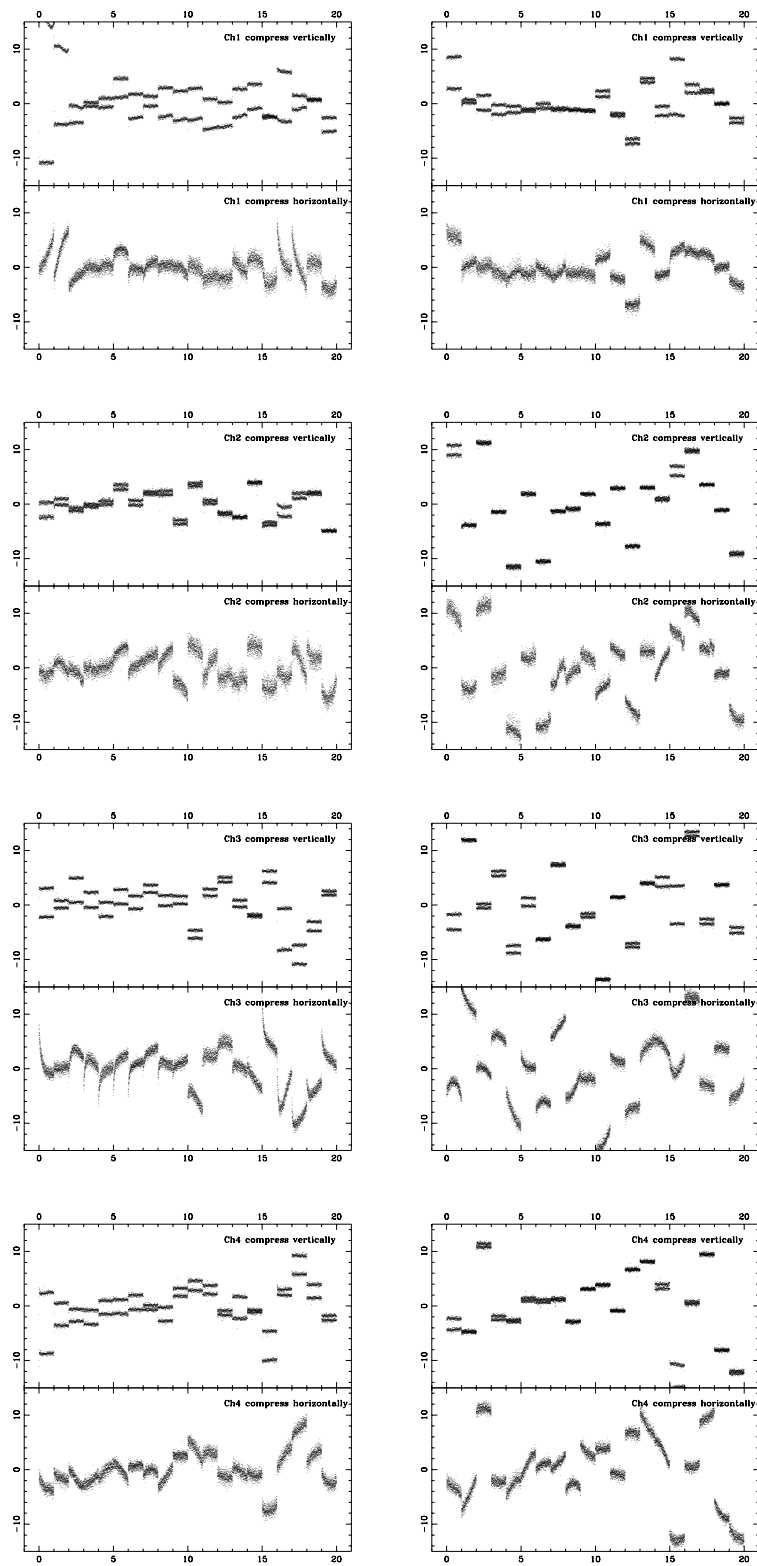


図 3: 常温で取得した全フレームを平均フレームで引いて、縦方向と横方向に 2 次元化して並べたもの。上段から順に、コドラント 1, 2, 3, 4。左が 0001 ~ 0020, 右が 0021 ~ 0040。

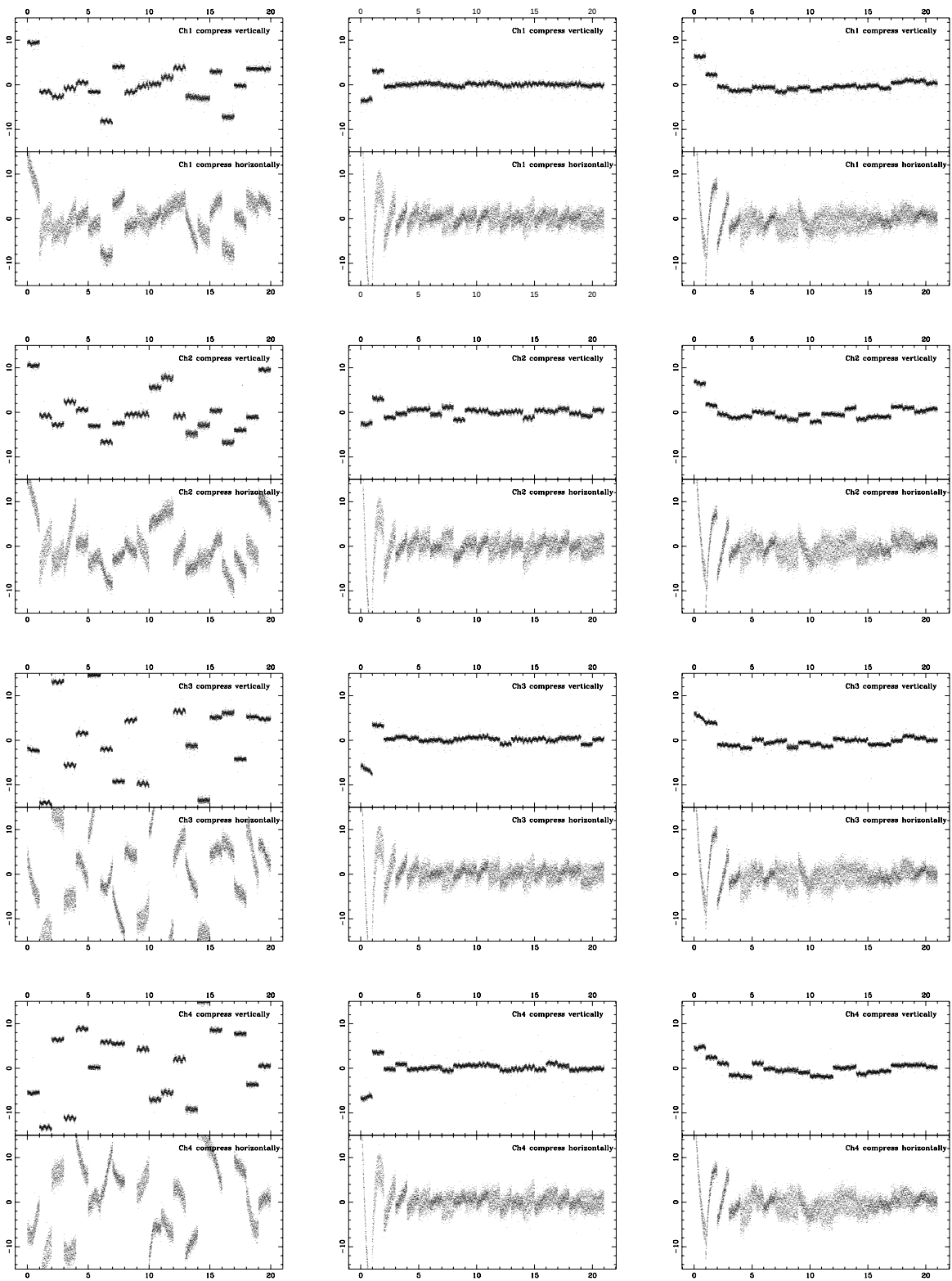


図 4: 77K で取得した全フレームを平均フレームで引いて、縦方向と横方向に 2 次元化して並べたもの。上段から順に、コドラント 1, 2, 3, 4. 左から順に 0054 ~ 0073, 0143 ~ 0163, 0164 ~ 0184.

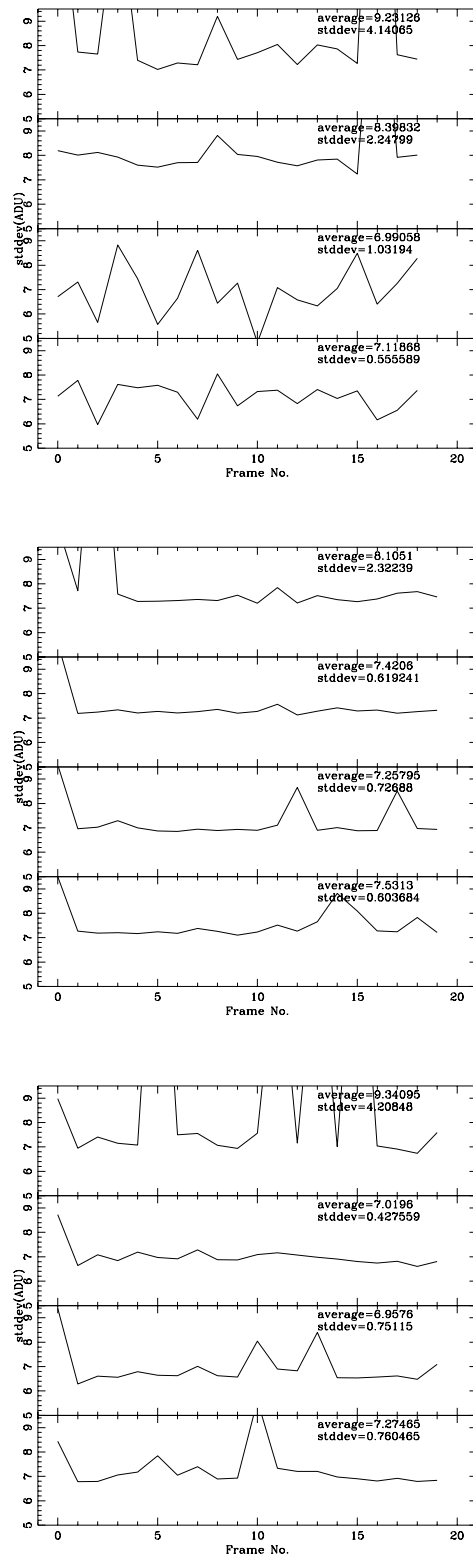


図 5: 20 枚連続フレームの隣合う 2 フレームの差のフレームの stddev の値。コドラント 1 が他のコドラントに比べて不安定なのが見える。また、これで求めた「ノイズ」はこれまでの方法よりも若干大きめに出現している。

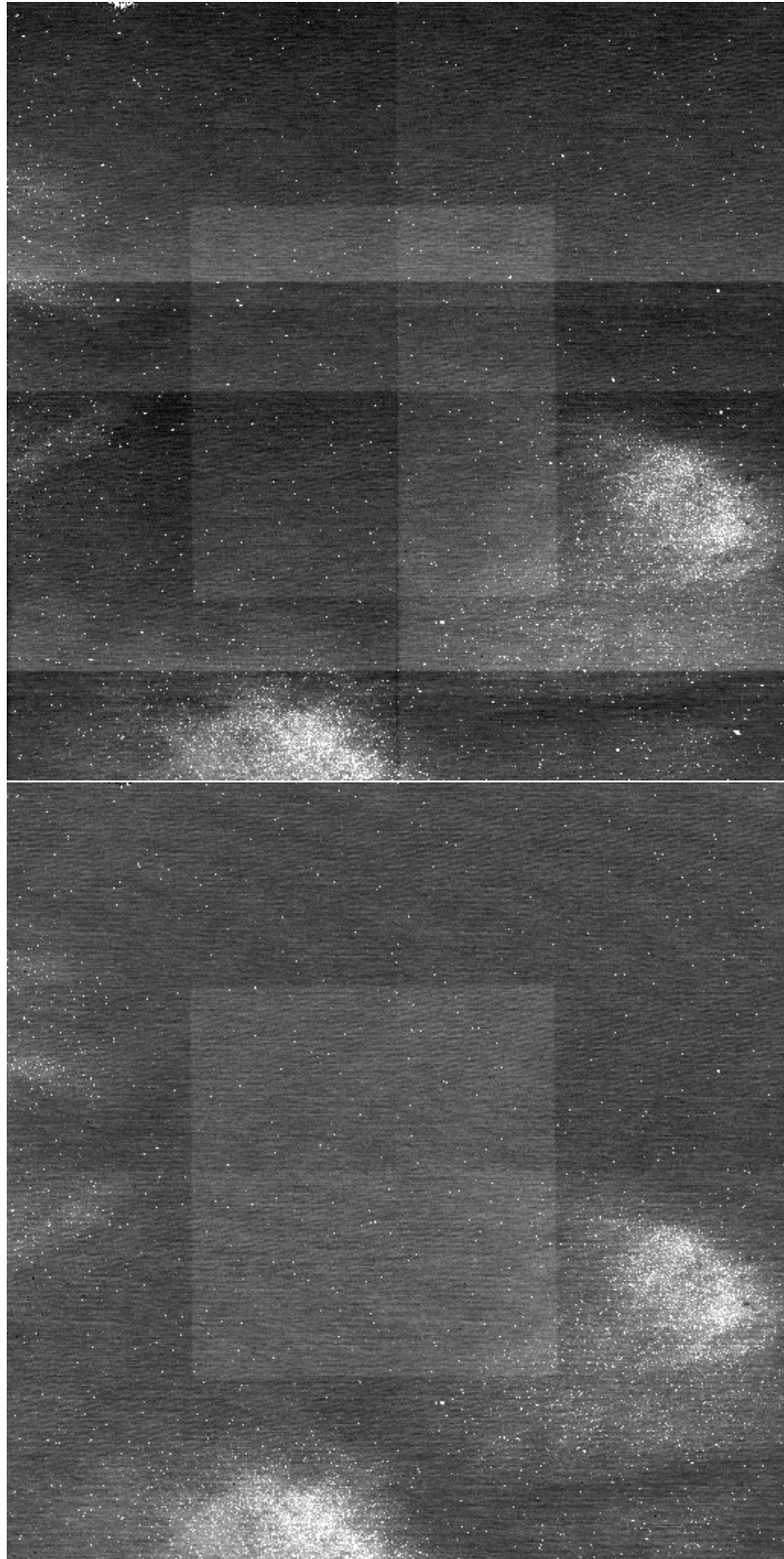


図 6: クロックによるイメージの違い。上は reset_scan-scan で得たもの。下は reset_scan-scan-scan の後の 2 枚で得たもの。