

検出器からの読み出しの conversion factor 2

本原顕太郎

1998 年 7 月 30 日

概要

CISCO の読み出し回路の conversion factor を求める目的で、さまざまな露出レベルでの stddev と count の関係を求めた。

結果、求まった conversion factor は算出の方法によって違いが出た。平均フレームとノイズフレームそれぞれの平均値から導くと $\sim 4.0e^-/\text{ADU}$ 、conversion factor フレームを作り、その平均値から導くと $\sim 3.3e^-/\text{ADU}$ となる。

1 データ

CISCO の入射窓に白い紙を貼り、その前面の壁面を懐中電灯で照らした。さらに、DC レベルのオフセットをちゃんと見積もるために、スリットを半分閉じて光の入らない領域を作った (図 1)。

用いたフィルタは H と J 、積分時間を変えることにより入射フラックスの量を変化させている。データは以下の通り。それぞれのフラックスで 20 枚ずつフレームを取得した。

No.	Filter	積分時間	count
0601~0620	H	2.0	~ 8000
0621~0640	H	4.0	~ 16000
0641~0660	H	6.0	~ 23000
0661~0680	H	1.5	~ 6000
0681~0700	H	1.5	~ 900
0701~0720	H	3.0	~ 1700
0721~0740	H	4.5	~ 2400
0741~0760	H	6.0	~ 3000
0761~0780	H	10.0	~ 5000
0781~0800	H	20.0	~ 10000
0801~0820	z	1.5	~ 40
0821~0840	z	2.0	~ 50
0841~0860	z	3.0	~ 80
0861~0880	z	6.0	~ 160
0881~0900	z	10.0	~ 260
0901~0920	z	20.0	~ 530

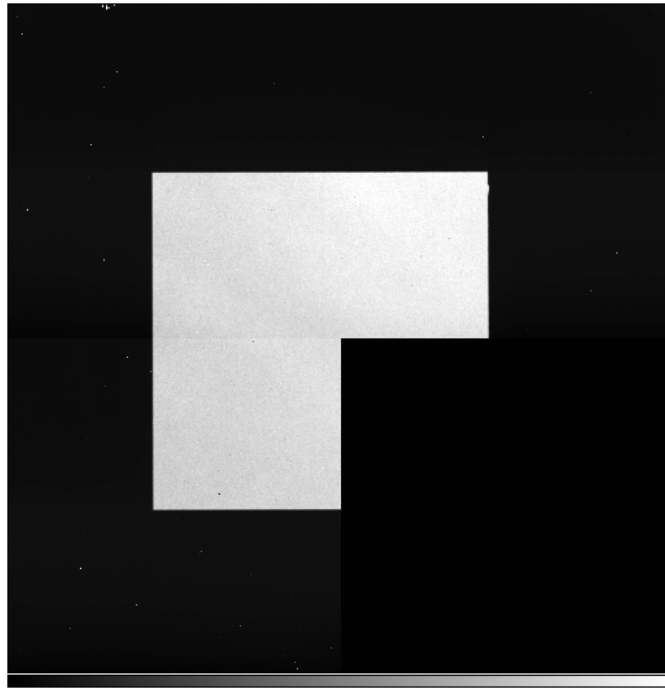


図 1: フレーム 0601。 $z_1=0$, $z_2=8890$ で表示してある。DC レベルのオフセットをちゃんと見積もるためにスリットを半分まで閉じて積分している。コドラント 1 はおそらくファンアウトボードの不調により読み出せていない。

2 イメージの処理

各々のフラックスでとったイメージ組について、領域 $[385:512, 385:512]$ の average フレームと stddev フレームを以下の方法で作った。

2.1 average フレーム

1. DC offset の量を評価するため、各フレームの領域 $[1:128, 385:512]$ を x 方向に 0 次、 y 方向に 2 次の Legendre 関数で fit したフレームを作る。
2. 領域 $[385:512, 385:512]$ と fit したフレームとの差をとり、それを平均して average フレームとした。

2.2 stddev フレーム

1. DC レベルの傾きや入射フラックスの時間変動成分を除去するため、領域 $[385:512, 385:512]$ を x , y 方向に各々 2 次の Legendre 関数で fit し、その残余のフレームを作る。
2. その残余のフレームで stddev フレームを作った。

3 解析結果

検出器のピクセルに x ADU の電荷が溜って、それを読み出した時の値の揺らぎが y ADU、さらに読み出しノイズが a ADUrms/frame で conversion factor が ke^-/ADU だとすると、これらの間には

$$(yk)^2 = xk + (ak)^2 \quad (1)$$

の関係が成り立つ。

この関係式を用いて解析を行なう。

3.1 解析その1

average フレームと stddev フレームのピクセル値の平均の組をグラフ上にプロットし、それに式 (1) を重ね書きした。

結果を図 2 に示す。Conversion factor は $4.0 e^-/\text{ADU}$ でもっとも良く合うことがわかる。

3.2 解析その2

ここで、average フレームと stddev フレームから以下のように新たなフレーム conversion を作る。

$$\text{conversion} = \frac{\text{average}}{\text{stddev}^2} \quad (2)$$

先の式 (1) で、 x が十分に大きければ

$$k = \frac{x}{y^2}$$

となるから、十分にカウントの大きい conversion フレームは、まさに各ピクセルの conversion factor を保持しているはずである。各々の組の conversion フレームのヒストグラムを図 4, 5 に示す。

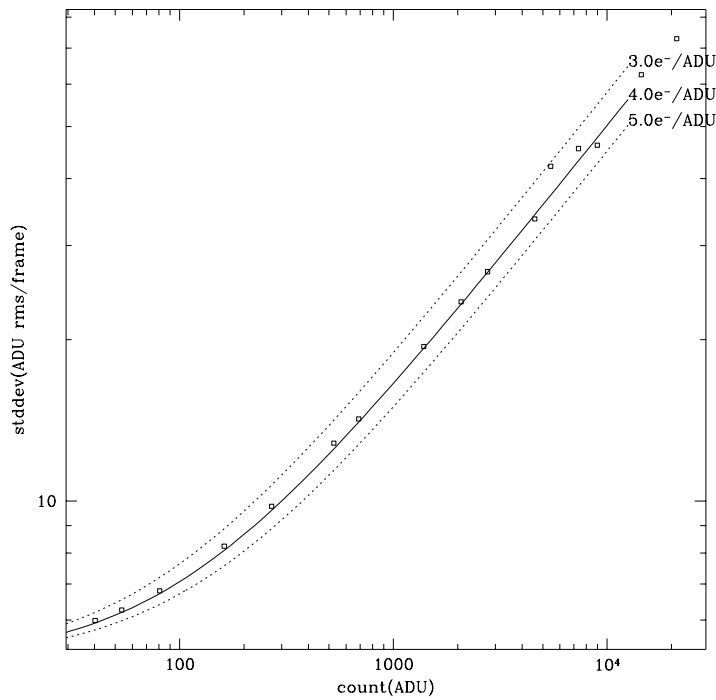


図 2: Count-Noise 関係。読み出しノイズ a はすべて 5.0 ADU である。

また、conversion フレームのピクセル値を z とすると、式 (1) から

$$z = \frac{x}{y^2} = \frac{xk}{x + ka^2} \quad (3)$$

となる。そこで、average フレームの average と conversion フレームの mode をグラフ上にプロットし、これに式 (3) を重ねたのが図 3 である。これから見る限りでは、conversion factor は $4.0 e^-/\text{ADU}$ ではなく、せいぜい $3.3 e^-/\text{ADU}$ である。

4 まとめ

二通りの解析を行なった結果は、一致しなかった。Average-Stddev 関係からは $4.0 e^-/\text{ADU}$ 、Average-Average/Stddev² 関係からは $3.3 e^-/\text{ADU}$ という値が出てきた。

この値の違いがなぜ生じるのかはまだ良くわからないが、Average-Average/Stddev² 関係の方がよりピクセルの個性の違いの影響を排除していることを考えると $3.3 e^-/\text{ADU}$ の値の方を信用すべきではないかと思う。

ただ、ヒストグラムを見てもわかるように Average-Average/Stddev² 関係では conversion フレームの値のバラツキが大き過ぎる。このバラツキは stddev フレームのバラツキを反映しており、より正確な stddev フレームを作る必要がある。そのために、次の試験では、一組当たりのフレーム数をより大きくしてデータを取得する必要があると考えている。

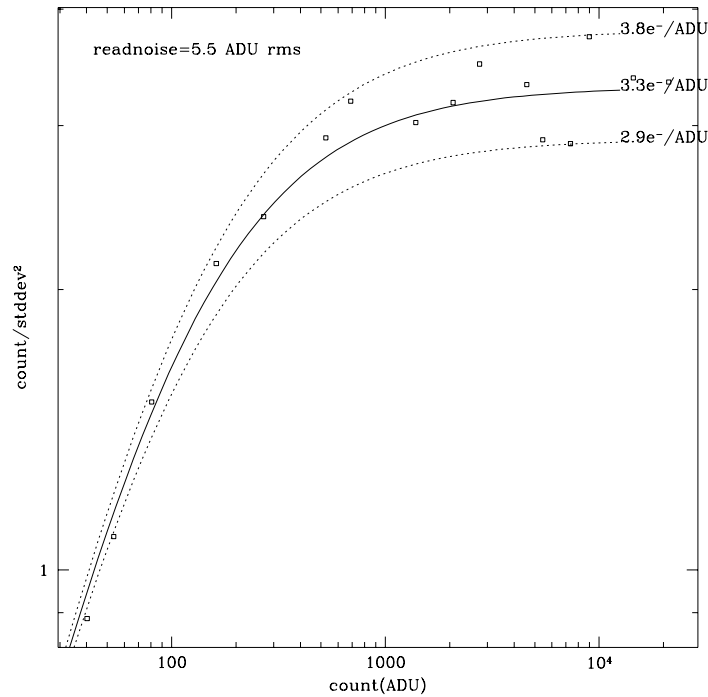


図 3: Count-Count/Noise² 関係。

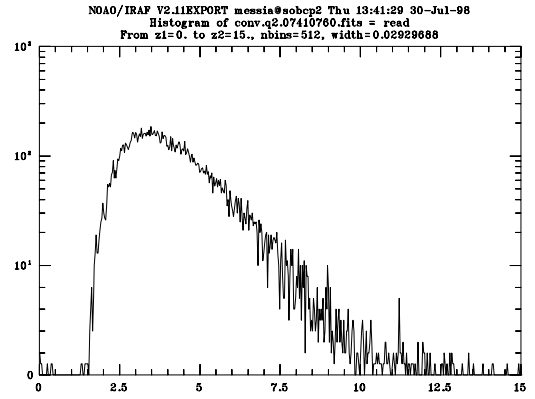
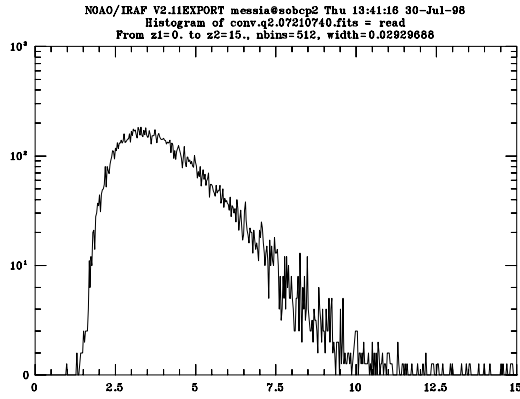
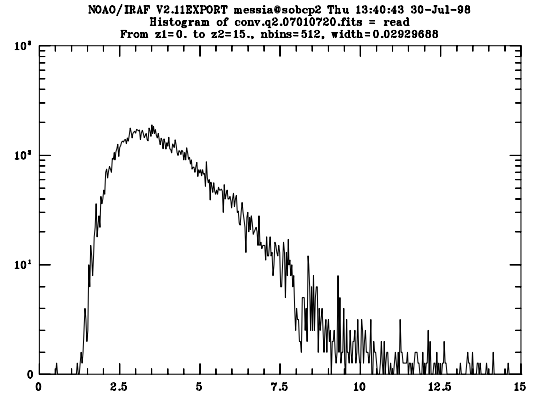
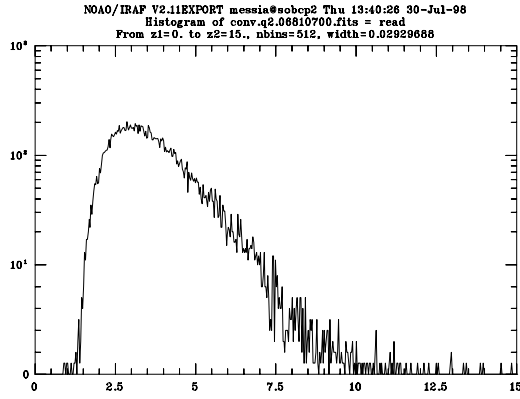
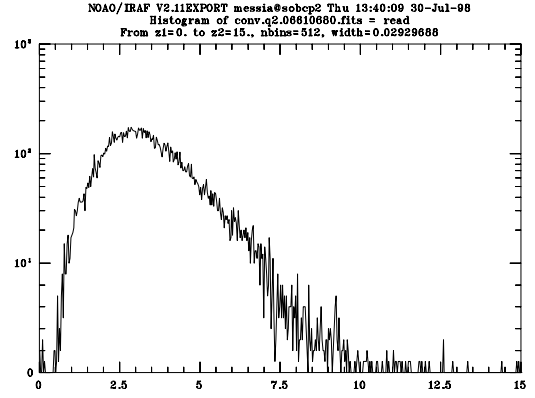
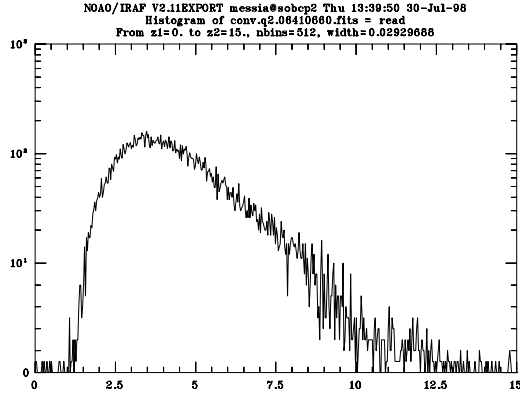
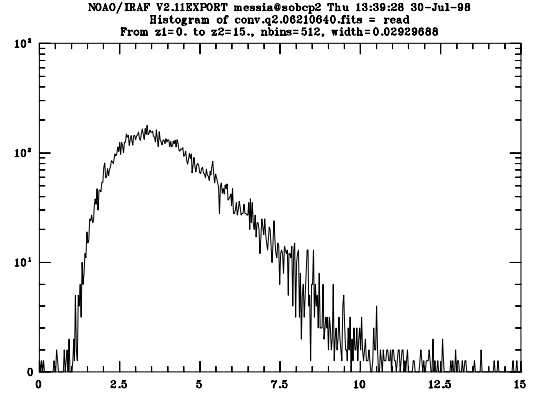
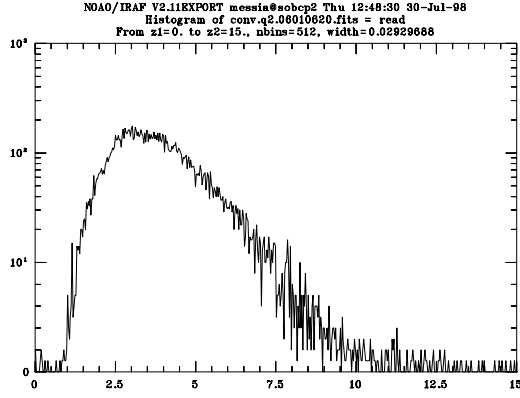


図 4: Conversion factor frame のヒストグラム。

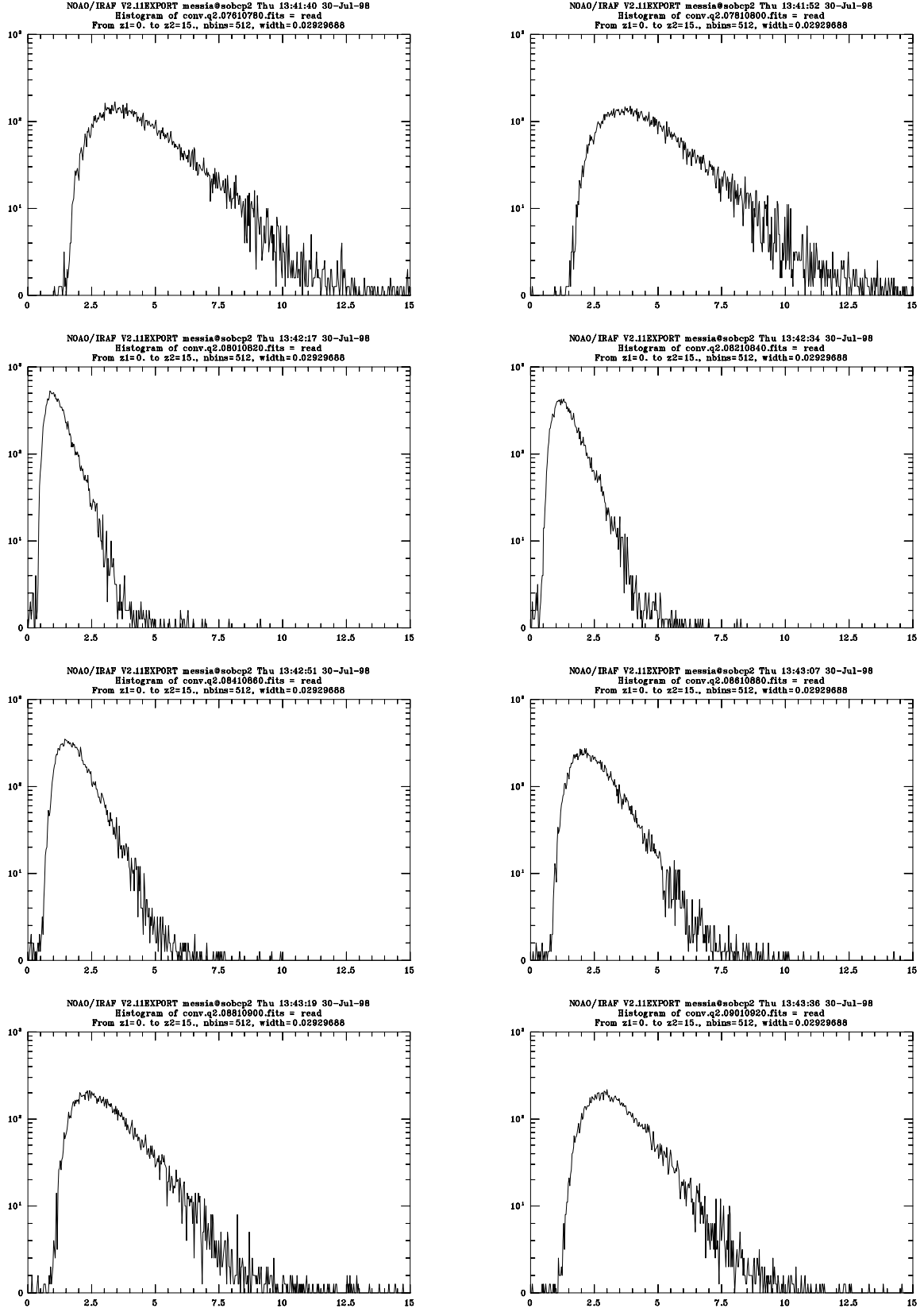


図 5: Conversion factor frame のヒストグラム。