国立天文台報　第 8 巻, 15－20（2005）

JASMINE プロジェクトのためのデータ圧縮技術

山田良透 *1，上田誠治 *2，奥村晴彦 *3，矢野太平，郷田直輝

（2004年 9 月30日受理）

A Data Compression Algorithm Suitable for JASMINE

Yoshiyuki YAMADA, Seiji UEDA, Haruhiko OKUMURA, Taihei YANO, Naoteru GOUDA

Abstract

We investigate data compression algorithm for astronomical satellite mission JASMINE. We are required to use lossless compression algorithms for scientific data. We cannot use lossy compression schemes which recently develop with the advance of internet technologies. Astronomical data is mainly image data, and it is binary data. Consequently, we must use lossless binary compression algorithms. Furthermore, CPU power is very limited in satellite mission. We conclude that combination of Golomb-Rice codes and Karhunen-Loeve transformation is suitable for JASMINE data.

1. はじめに

近年の衛星プロジェクトは，大型化や観測装置の進歩によって，取り扱うデータ量が増大しており，これに対応して大量のデータを地上へ転送しなければならない．一方で、人工衛星プロジェクトに特有の制限として，データ転送量の制限とデータ処理能力を考慮する必要がある．データ転送量に関して，地上のネットワークでは10Gbps 程度の高速通信が可能だが，衛星からのデータ転送は衛星の軌道や使用するアンテナに依存して通信速度は数Mbps から数十Mbps 程度に制限される．また，宇宙では放射線環境が厳しいため，使用可能なCPU は非常に処理能力の低いものに制限される．圧縮に関して言えば，地上のパソコンではほんの数秒程度で可能な数倍程度の圧縮でも，衛星機上で行なうのは容易なことではない．本稿では，CPU の処理能力が低いと言う前提のもとで，すばやく大量のデータを転送するために必要なデータ圧縮手法について，具体的にJASMINE 計画[1] での検討に沿って議論する．

JASMINE（高精度赤外線位置天文観測衛星計画）やSPICA（大型宇宙赤外線望遠鏡計画）などにおいては，さまざまな観点から地球－太陽が作るL2点周りでの観測が望ましい．ここは地球から150万kmと遠方のであることと，JASMINE の高度な姿勢安定要求のため電気的に指向性を変化させるフェーズドアレイアンテナの使用を検討してことから，通信帯域は0.7（旧NASDA の評価[2]）～3Mbps 程度（GAIA グループの評価[3]）となる．一方JASMINE の最新の仕様による必要なデータ転送量は，転送方法によりおよそ1.2Mbps から9Mbps 程度となり[4]，このデータをさらに 2 ～ 3 倍程度に圧縮しなければ，転送不可能という結論が得られる（表 1 参照）．

表 1．データ伝送量の制限と必要な通信量．データ量の評価は本文参照．

生データ	3.5Gbps
パラボラアンテナ転送限界	30Mbps
2D データ転送量(平均値)	9Mbps
1D データ転送量(平均値)	1.19Mbps
フェーズドアレイアンテナ転送限界	0.7～3Mbps

衛星機上データ処理量に対する制限も，人工衛星プロジェクトでは非常に厳しい．集積度の高い電子デバイスは放射線に弱いことから，ASTRO-F 衛星やSELENE 衛星のデータ処理系は1-

*1 京都大学
*2 総合研究大学院大学,現所属(株)ドリームメカニズム
*3 三重大学

2MIPS 程度，JASMINE のミッション提案の時期までに十分な放射線対策の目処がたつと見込まれる範囲のCPU の処理能力は，MIPS 値で最大200MIPS 程度と予想される．データ転送量の制限からはデータの圧縮が必要だが，このデータ圧縮を非常に限れたCPU 資源で実現するアルゴリズムを考えなければならない．

そこで，我々は，JASMINE のデータの特徴を検討し，既存の圧縮技術について比較検討した．圧縮にはモデル化と符号化の二段階があるが，モデル化にKarhunen-Loeve 変換を，符号化にGolomb-Rice 符号を用いる組合せがJASMINE の星像データに適していることを見いだしたので，報告する．Karhunen-Loeve 変換は，複数のデータからのパターンの抽出を線形操作で行なうことができる点において，制限された処理能力のものとで行なえる優れたモデル化手法であり，多くのサイエンスミッションへの応用が期待される．また，Golomb-Rice 符号化は，一般に小さな数の出現頻度が高く大きな数の出現頻度が低い場合に適用される．天文画像などは背景レベルは低く，シグナルレベルが高いため，この符号化が適している．また，Golomb-Rice 符号化は数値の出現頻度が指数関数的に分布する場合には効率良い圧縮が行なえるという特徴を持っている．科学データは，自身の出現頻度がこれに従っていなくても，順次近似関数を差し引いた残差列は指数関数的分布になることが多く，適当なモデル化のもとでは高効率な圧縮が期待できる手法である．さらに，一般の圧縮における符号テーブルに相当する符号化パラメータはGolomb-Rice 符号においては4bit であり，小さいデータでも効率良く圧縮可能である点で優れている．小さなデータを効率良く圧縮できる特徴は通信エラーにたいする堅牢性を与え，Golomb-Rice 符号化に要する処理は非常に軽く，特に衛星プロジェクト一般への応用が期待される．第 2 章では，本検討の基礎となる既存の圧縮技術に関するサーベイを行なう．第 3 章では，星のCCD 画像などに適した符号化の手法として，Golomb-Rice 符号化について述べる．第 4 章ではモデル化技術としてKarhunen-Loeve 変換を紹介する．第 5 章は、議論と今後の課題についてまとめる．

2. 既存の圧縮手法の検討

JASMINE に要求される圧縮の要件を整理すると，以下のようになる．

●可逆圧縮であること
●比較的小さなデータでも高効率で圧縮可能であること
●圧縮に必要なデータ処理量が比較的小さいこと

最初に，ここで用いる圧縮率の定義を述べておく．本稿では，

(圧縮率)=(１－(圧縮後の容量)/(圧縮前の容量))

を圧縮率と呼ぶことにする．つまり，圧縮する前と圧縮した後でデータの容量が違わなければ圧縮率は0%，データが半分になれば50%，データが1/10になれば90%と表すことにする．この表記では，数字が大きいほどより効率良く圧縮していることになる．文献によっては

(圧縮率)=(圧縮後の容量)/(圧縮前の容量)

を圧縮率と表現している場合もあるので，他の文献のデータと数値を比較する場合には、圧縮率の定義に注意を要する．

圧縮には可逆圧縮と非可逆圧縮があるが，科学データでは可逆圧縮が必要となる．インターネットや携帯電話の発達で，画像データや音声データの非可逆圧縮技術は近年急速に進歩した．非可逆圧縮の場合，圧縮率も90%から95%と非常に高い圧縮率を示すこともある．そもそも非可逆圧縮は，もともとアナログデータである音声や画像などのデータを離散化誤差以上の精度でデータを正確に復元する必要は無いという着想で考えられたものである．連続的なアナログデータを離散化してデジタルデータにしたという事情は天文学で扱う画像でも同じように思えるが，事情は大きく異なる．天体写真を含む多くの科学データは通常複数回データ取得を行なう．これらから統計処理を行なう際に，ノイズの性質にまで立ち入って処理を行なっている．また，さまざまなシステム誤差も，データの数値から計算される．一見ノイズに思えるようなデータも，処理をする上で科学的に意味があるデータとなり得るのである．そのため，科学データの圧縮を考える際には，可逆圧縮は是非とも必要である．

近年コンピューターの実行可能プログラムを配布する技術として，バイナリーデータの可逆圧縮が再び進歩している．この手法では，バイナリーデータが60%程度で圧縮できる方法が利用されている．しかしながら，これらの方法では，数MB

程度のまとまったデータを圧縮しなければ，高効率な圧縮はできない．一方JASMINE の星像データは，2 次元情報を転送する場合は90バイト，1 次元情報のみを転送する場合は12バイト程度となる．地上で用いられているLZ圧縮技術（例えばgzip）やBlock Sorting 圧縮技術（例えばbzip2）を用いて100バイト程度のデータを圧縮すると，大抵は圧縮率が負，即ちもとのデータよりも大きくなってしまう．さらに，必要な星像データをいくつか連結した大容量のデータを圧縮しようとすると60%程度の圧縮率を得ることができるが，CPU パワーも必要となる．一般に，大きなデータを圧縮する場合，データ量n に対してn^2 からn^3 程度の処理量が必要となる．

表 2. 既存の圧縮手法による場合．連続した 1 個から1000個の星像データをbzip2 およびgzip で圧縮した結果．上の星像 1 個から1000個の 4 つのデータは全て中心pixel がfull well で露出する場合，下の10個から1000個の 3 つのデータはJASMINE の観測で期待できるLuminosity 分布を持つ場合．

	圧縮率	
星像数	gzip	bzip2
1	−36%	−92%
10	−4%	−27%
100	0.6%	−5%
1000	1%	0.06%
10	31%	40%
100	43%	54%
1000	45%	59%

実際に，模擬的なpixel データを作り，bzip2とgzipで圧縮を行ってみた．我々の星像Windowは90バイトのデータであるが，星像一つでは圧縮後の容量は200バイト程度，即ち二倍以上に増えてしまっている．これらの圧縮技術に用いられる符号化技術であるHuffmann 符号化あるいは算術符号化では，符号化の単位は通常8bit である．圧縮データの先頭8bit で表現される256種類のデータの符号化テーブルを付加するため，もとのデータより圧縮処理を行ったデータの容量が増えることになる．複数の星像をまとめて転送する場合は，数がある程度大きくなれば圧縮効率は向上する．具体的な数値実験結果は，表 2 に示す．また，100個の星像をまとめて圧縮するための処理に，Pentium4 1.5GHz のパソコンで0.3秒程度，1000個の場合は0.6秒を要した．星像は平均で 1 秒辺り5000個程度観測される[4] ので，100個の星像を0.02秒，1000個の星像を0.2秒以内で処理しなければならず，Pentium4 クラスのCPU でも実時間では処理不可能である．圧縮の単位となるbit 数を小さくしても，出現頻度に偏りが少なくなるため圧縮効率は上がらない．

このように，近年発展してきた圧縮技術は汎用性の高い圧縮技術である一方，画像や音声の圧縮に用いられるものの多くは非可逆であり科学データに適さない．また，バイナリーデータを高効率で圧縮する技術も大量のデータを同時に圧縮しなければ高効率を実現できない．大量のデータを同時に処理するためには大きな処理能力が必要となるが，衛星搭載可能なプロセッサの処理能力が低いことから，科学衛星プロジェクトには不向きである．科学衛星プロジェクトの場合，汎用性を犠牲にしても良いので，このセクションの最初で述べた三つの条件を満たすような圧縮手法を考慮することが必要になる．

3. Karhunen-Loeve 変換

データ圧縮はモデル化と符号化から構成される．通常まずモデル化によって符号化しやすいデータに変換し，その後に符号化を行なう．この処理の順序に従って，まずJASMINE のデータ圧縮でモデル化として採用できそうな技術に関して述べる．

PSF（Point Spread Function）の形状は光学系によって定まるため，PSF を離散化したものと考えられるpixel データからAiry 関数のパラメータを推定することが出来れば，これがもっとも良いモデル化となる．しかしながら，この変換は極度に非線形であり，計算量が膨大となる可能性が大きい．さらに，打ち上げの際の光学系の微妙な変形などによって推定関数が変化する可能性もあり，変換に必要なパラメータを地上実験から予め確定することも難しい．そこで，線形計算で行なえ，データから変換に必要なパラメータを推定できる、主成分分析という方法を応用することを考える．

ここで用いるモデル化は，情報科学ではKarhunen-Loeve 変換と呼ばれる変換だが，これは数学の用語で言えば N 個のデータを N 次元空間の点と見なして，N 次元データ点に対して寄与率の高い軸に直交変換する操作である．JASMINE の星像データは9×5 = 45pixel のデータとなるが，個別の星像を表す45次元の点を$\mathbf{x} = \{x_k\}$ と表す．i 番目の星像$\mathbf{x}_i$ の各pixel 値は$= x_{ki}$ と表すことにする．今，45個の直交主軸ベクトル$\mathbf{w}_j =$

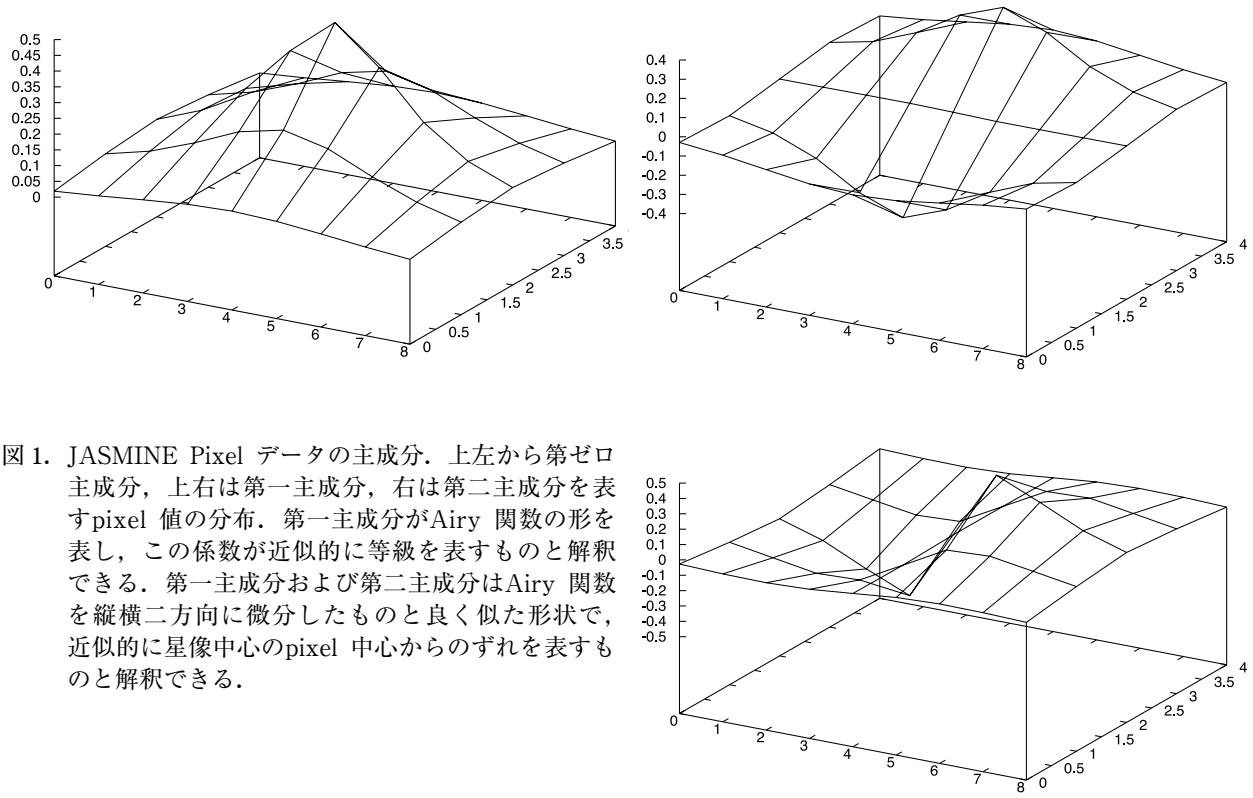

図 1. JASMINE Pixel データの主成分．上左から第ゼロ主成分，上右は第一主成分，右は第二主成分を表すpixel 値の分布．第一主成分がAiry 関数の形を表し，この係数が近似的に等級を表すものと解釈できる．第一主成分および第二主成分はAiry 関数を縦横二方向に微分したものと良く似た形状で，近似的に星像中心のpixel 中心からのずれを表すものと解釈できる．

w_{jk} が与えられていれば，

$$x_{ki} = \sum_{j=0}^{44} c_{ji} w_{jk}$$

と書いた時の係数c_{ji} は

$$c_{ji} = \sum_{k=0}^{44} w_{jk} w_{ki} = \mathbf{w}_j \bullet \mathbf{x}_i$$

で求められる．

今，データx から直交主軸ベクトルw_{jk} は，M 個のデータ列$\mathbf{x}_i$;（$i = 0 \cdots \mathrm{M}$）の分散共分散行列

$$s_{lm} = \frac{1}{M} \sum_{i=0}^{M} (x_{li} - \overline{x}_{li})(x_{mi} - \overline{x}_{mi})$$

の固有ベクトルとして求めることが出来る．固有値が寄与率となるため，固有値の大きい物から順に$\mathbf{w}_0, \mathbf{w}_1, \cdots$とする．

このような変換を行なった係数c_{ji} とx_{ki} はともに同じ数の数値だが，寄与率が高いものは大きいビット数での表現が必要だが，寄与率が低い物については少ないビット表現で足りるため，この変換を予め行なうことは圧縮効率を高める可能性がある．さらに，第三番目までの主成分に対応する固有ベクトルを，pixel 値として表現した物を示すと，図 1 のようになる．これを見ると，第ゼロ主成分はPSF の形状を示し，第一主成分はPSFの中心の5pixel 側へのずれ，第二主成分はPSFの中心の9pixel 側へのずれを表す．従って，こう言ったデータを計算しておくことは，圧縮のためだけでなく衛星上で等級を求めたり，sub pixel 精度での星像中心を評価するのにも役立つ．

主成分を計算するためには内積だけで良いが，変換行列を求める計算はコストがかかる．しかしながら，これは頻繁に行なう必要がある計算ではないので，ある程度データがたまった段階で計算し，光学系や衛星などの安定度をチェックする必要がある頻度で再計算すれば良い．従って，変換行列の計算コストは全体としては問題にならない．更に，可逆データを地上に転送しているので，この計算は地上の計算機を用いて行ない，変換行列のみをアップロードすれば，衛星機上の処理系を使う必要は無い．

4. Golomb(-Rice)符号化

第二章で見ように，現在広く用いられているバイナリーデータの可逆圧縮技術に用いられる符号化は符号化テーブルが大きいため，人工衛星のデータの処理には不向きである．この問題点を解決

するためには，単純にはテーブルそのものを固定とし，これを符号化側と復号で記憶させておいて使用する方法が考えられる．しかし，この方法では柔軟性が無い．本質的には，参照テーブルが少ないパラメータで表される方法があれば良いことになる．

そこで，次のような符号化（Golomb-Rice 符号化）を考える[5, 6]．この符号化は，0 に近いほど頻繁に現れ，0 から離れると出現頻度が下がるようなデータの符号化に適している．今，データ列 $e_i\,(i = 1, \cdots, n)$ を圧縮する場合に

$$a = \sum_{i=1}^{n} |e_i|$$

に対して

$$2^k < a$$

を満たす最大の k を計算する．この計算は，

for（k = 0 ; (n << k) < a ; k++);

の一行のコードで計算できる．これに対して，$k$bit より上位のビットはunary コードで，下位のビットはそのまま無変換で符号化する．unary コードとは，整数値 x を x 個の 0 並びにstop コード 1 を付加した符号を割り当てる方法である．0 と 1 の役割は完全に逆転しても良い．unary コードによる木構造は，整数 $x(\geqq 0)$ の出現頻度が 2^{-x-1} である場合には最小冗長符号を与えるHuffman 木[7] の木構造と一致することが知られている．Golomb-Rice 符号化の場合は上位ビットと下位ビットを分けていることから，底が 2 でない一般の指数関数的な分布をする場合にもほぼ最小冗長符号となる．

星像のpixel データはPSF で決まり，中心が大きく周辺部が小さい．これを主成分に分解すると，頻度分布はほぼ指数関数的になる．さらに，星の光度の数分布もほぼ指数関数的になる．したがって，データは二重の意味で指数関数的に分布することが期待され，Golomb-Rice 符号化で効率良く圧縮できることが期待できる．数値的に実験したところ，表 3 に示す通りほぼ1/3程度に圧縮可能であることがわかる．表の計算は，星像中心が縦横とも1pixle 以内で一様に分布するとし，PSF に類似する関数として 2 次元のガウス分布を用いて光子が検出器上に落ちる位置を確率的に計算し，pixel 単位で離散化した数値を計算することにより作った，JASMINE の疑似星像のサンプルを用いて解析した結果である．表は10000サンプルでの平均値で，主成分数 0 の（すなわちGolomb-Rice 符号化だけを適用する）場合は，サンプル数は統計的な意味を持ち，それ以外の場合は主成分を適切に取り出すことが出来るかどうかが関係する．同時に1000サンプルの場合の数値実験を行なったが結果は 1 %程度しか違わなかった．

表 3．Golomb-Rice 符号化した場合の必要なbit 数．もとのデータは720bit．10000サンプルでの平均値．主成分を取り出さなくても，Golomb-Rice 符号化だけで52%程度の圧縮が可能，もっとも効率が良い第三主成分まで取り出した場合では66%程度の圧縮が可能となっている．

主成分数	残差計	主成分	k	計
0	338.527	0	4	342.527
1	247.013	12.247	4	263.260
2	222.109	22.737	4	248.846
3	207.794	31.814	4	243.608
4	203.974	39.257	4	247.231
5	202.063	45.703	4	251.766
6	200.866	51.495	4	256.361
7	199.738	57.001	4	260.739
8	198.485	62.497	4	264.982

JASMINE のPSF については論文[8] に掲載されているとおりだが，ビーム混合鏡により視野を二分割しているため，PSF は縦横比がほぼ1:2 に延びた形状となる．圧縮の効率はPSF の動径方向の関数の詳細にはよらず，pixel 値の頻度分布が適切に表現されていれば良い．星像は本来円形の光学系のAiry 関数として知られるAiry 関数を用いて，中心部に穴の空いた円形の光学系ののPSF として知られるAiry を組み合わせた形の関数で表現される．しかしながら，星像は5 × 9pixel 程度の狭い領域に分布するため，この離散化度でのデータの頻度分布は十分良くガウス関数で近似できるので，今回は縦横の分散を1:2 とした非等方の 2 次元ガウス関数を，PSF の近似関数として用いている．

Golomb-Rice 符号化は，条件判断がはいるので実際の処理量は星像の性質に依存するが，計算量はおよそ以下のように見積もられる．パラメータ k の算出に星像あたりの最悪値でデータ数（JASMINE の場合は45）の 2 倍と1pixelのビット数（JASMINE の場合は16）の和，これを符号にするためにはおよそpixel あたりで数operation，最悪値でbit 数程度となる。そこで，JASMINE の星像の場合はノミナル値で200程度，最悪値でも1000以下となる。1 秒あたりの星像の数が平均で

5000個程度であることから，1MIPS 程度の処理能力でほぼ十分であり，最悪値を考えても5MIPS 程度の処理系を考えれば十分であることが分かる．

5. まとめ

JASMINE の星像を転送するための圧縮手法を検討した．モデル化にKarhunen-Loeve 変換を，符号化にGolomb-Rice符号を用いる組合せが，JASMINE の星像データに適していることが分かった．星像はPSF により表現されるが，データからPSF のパラメータを逆算する計算は極度に非線形であり，処理量が膨大となる．さらに，装置の経年劣化などがあるため検出器の応答まで含めたPSF は時間的に変化し，これを予想することは困難である．Karuhnen-Loeve 変換は，パターンの抽出を線形操作で行なうことができる点において，制限された処理能力のものとで行なえる優れたモデル化手法である．実際，JASMINE の星像データからKaruhnen-Loeve 変換の変換行列を求めると，寄与率が高い順に等級，星像の5pixel 側へのずれ，星像の9pixel 側へのずれを表す成分を取り出すことが出来ることが示された．この操作において，あらかじめPSF に関する情報を必要とせず，純粋にデータから主要な 3 成分が取り出せたことは，他の多くのミッションへの応用可能性を示唆するものである．

また、Golomb-Rice 符号化は，一般に小さな数の出現頻度が高く大きな数の出現頻度が低い場合に適用され，数値の出現頻度が指数関数的に分布する場合には効率良い圧縮が行なえる．科学データは，データ自身の出現頻度がこれに従っていなくても，近似に従ってテンプレートとなる関数を差し引いた残差列は指数関数的分布になることが多く，こういったデータの圧縮に適した方法である．さらに，一般の圧縮における符号テーブルに相当する符号化パラメータは4bit であり，小さいデータでも効率良く圧縮可能である点で優れている．小さなデータを効率良く圧縮できる特徴は通信エラーにたいする堅牢性を与え，Golomb-Rice 符号化に要する処理は非常に軽く，特に衛星プロジェクト一般への応用が期待される．

我々の数値実験の結果，JASMINE の星像に相当する720bit のバイナリーデータを効率良く1/3程度に圧縮出来ることが分かった．現在，コンパイルされた計算機プログラムの配布などに用いられている優れた可逆圧縮手法であるBlock Sorting やLZ 圧縮技術を用いた場合，この圧縮率を得るには1MB 以上のまとまったデータを高速のCPUで処理する必要が有るが，Golomb-Rice 符号化を用いれば90バイト程度のデータでも十分にこれらに匹敵する圧縮率が得られることになる．

この手法によると，等級に相当するデータ，星像中心のサブピクセルレベルでのずれを表すデータが取得できる．こう言ったデータは，衛星姿勢の把握などに用いることが出来ることが期待できる．この使用可能性を評価するため，今後はこう言ったデータの信頼性や衛星搭載機器の機械的なずれや経年変化による性能変化による影響などを評価することが必要となる．

参考文献

1) N. Gouda et al: SPIE, 4850, 1161 (2003).
2) A. Noda et al: private communications.
3) M. Perryman: Gaia in 2003 (http://www.rssd.esa.int/gaia/Assets/Paper/Gaia_in_2003.pdf).
4) 郷田直輝他 JASMINE WG:「光学赤外線天文学将来計画検討会報告書」(JASMINE 部分) (http://www.jasmine-galaxy. org/pub/future-report-0407.pdf).
5) S. W. Golomb: Run-length encodings, *IEEE Transactions on Information Theory*, **IT-12**(3), 399–401.
6) Robert F. Rice: Some Practical Universal Noiseless Coding Techniques, *JPL Publication*, **79**-22, (1979).
7) D. A. Huffman: A method for the construction of minumum-redundancy codes. *Proceedings of the Institute of Radio Engineers*, Vol. **40** (No. 9), pp. 1098–1101 (1952).
8) 矢野太平，高遠徳尚，小林行泰，郷田直輝: 国立天文台報 **7**, 9－14(2004).