

# 卒業論文

## 題目

ビームスポット列画像を対象とする圧縮アルゴリズムと  
そのソフトウェアデコーダーの開発

## 指導教員

近藤 利夫

2013年

三重大学 工学部 情報工学科  
計算機アーキテクチャ研究室

水野 篤 (410854)

## 内容梗概

レーザープリンタの製造において、治具を用いることによりレーザービームの微調整作業の効率化が期待されている。当研究室で開発の支援を行っている全領域治具は、レーザーの照射状況を18個の単眼センサで画像データとしてキャプチャし、そのキャプチャ画像をまとめて1台のコンピュータに転送し、そこで画像の解析を行う。キャプチャ画像が大量で、既存の高速転送手段の上限速度を超えるため、リアルタイムでキャプチャ画像を圧縮伸張ができるコーデックシステムの実現が要求される。

そこで、本研究では、センサから出力された画像データを対象とした圧縮アルゴリズムの確立と、高速なソフトウェアデコーダの開発を行った。

本研究で確立したアルゴリズムは、全領域治具で扱う対象画像の圧縮に特化したもので、既存の圧縮コーデックよりも高圧縮を可能としている。また、そのアルゴリズムに準じたソフトウェアデコーダもまた、既存の圧縮コーデックのデコーダよりも非常に高速で動作するものを実装した。

# Abstract

In the laser printer production line, a jig capturing whole irradiated area is required for efficient fine adjustment of the laser beam focus.

In this jig system, the whole irradiated area is captured by 18 monocular-vision sensors, and the captured image data are transferred to a single computer for laser focusing analysis.

Since the capturing image data rate is too high, compression and decompression are required for real-time transfer from the jig to the computer.

The purpose of this study was to development of a compression algorithm and its software decoder for laser beam spot array image.

The algorithm is specialized to the target image. Therefore the algorithm has higher compression capability than existing compression systems. Also the decoding speed of the software decoder is faster than the current decoders.

# 目次

<b>1</b>	<b>まえがき</b>	<b>1</b>
1.1	背景	1
1.2	研究目的	3
<b>2</b>	<b>対象画像の概要と圧縮転送の必要性</b>	<b>4</b>
2.1	レーザープリンタのレーザービーム	4
2.2	製造時のレーザービームの微調整	5
2.3	全領域測定治具とその構成	5
2.4	転送画像のデータ量	6
2.5	画像圧縮転送の必要性	7
2.6	対象画像の特徴	7
<b>3</b>	<b>既存の圧縮アルゴリズムとその問題点</b>	<b>9</b>
3.1	既存の候補アルゴリズム	9
3.2	GIF コーデック	10
3.2.1	特徴	10
3.2.2	問題点	11
<b>4</b>	<b>ビームスポット列画像向きアルゴリズムの提案</b>	<b>13</b>
4.1	提案アルゴリズム	13
4.1.1	アルゴリズム概要	13
4.1.2	符号化の仕様	15
<b>5</b>	<b>高速デコーダの実装</b>	<b>18</b>
5.1	処理方法	18
5.2	高速化	21
<b>6</b>	<b>評価</b>	<b>23</b>
6.1	評価方法	23
6.2	評価結果	23
6.3	考察	24
<b>7</b>	<b>あとがき</b>	<b>26</b>
	謝辞	27

## 目 次

1.1	モノクロレーザープリンタの構造 . . . . .	2
2.2	対象画像 . . . . .	8
4.3	符号化イメージ . . . . .	15
4.4	符号仕様 . . . . .	17
5.5	簡易フローチャート . . . . .	20

## 表 目 次

3.1	各圧縮コーデックの圧縮率評価 . . . . .	10
6.2	圧縮率および伸張速度表 . . . . .	24

# 1 まえがき

## 1.1 背景

レーザープリンタは企業用としてだけでなく家庭用プリンタとしても普及してきている。そのレーザープリンタの構造は図 1.1 の様になっており、印刷精度は図 1.1 中央の感光ドラムに対して照射されるレーザーの状態が決まるため、レーザーが正確に照射されているかを確認する必要がある。しかし、現状プリンタ 1 台のレーザーの照射範囲全域を一度に調査することができていないため、この確認作業にカラープリンタと違い、内部の構造が簡単なモノクロプリンタですら数十分単位の時間をかけており、精度を落とした部分的な調査ですら 10 分程度の時間を要してしまっている。検査精度向上による問題の早期発見、原因究明、対策効果確認は生産台数が増加している現在において非常に重要なものとなっている。

そこで高精度かつ高速でレーザーの照射状態の検査を行う全領域測定器具の開発が望まれている。

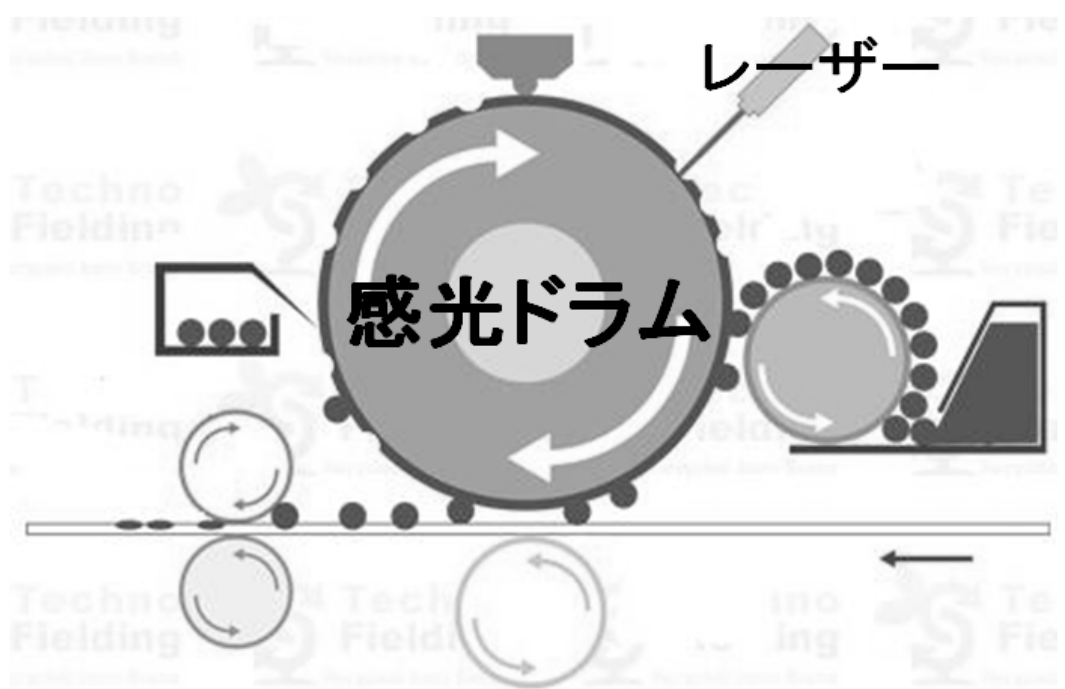


図 1.1: モノクロレーザープリンタの構造



## 1.2 研究目的

センサーから取り込んだ画像をそのまま転送するのでは、4:2:0 ハイビジョンの12倍もの情報の転送量になってしまうため、USB3.0 やカメラリンクを使用するとしても画像圧縮は不可欠である。またその画像圧縮のリアルタイムのデコードを、1台のパソコンでソフトウェアを用いて処理する必要がある。

そこで本研究では全領域測定治具において最適な圧縮アルゴリズムの確立とそれに対応する高速なソフトウェアデコーダーの開発を行い、速度と圧縮率を他の画像圧縮フォーマットと比較する事で妥当性の検証を行う。

## 2 対象画像の概要と圧縮転送の必要性

### 2.1 レーザープリンタのレーザービーム

カラーレーザープリンタと違い比較的内部の構造が簡単なモノクロレーザープリンタの動作について簡単に説明を行う。

モノクロレーザープリンタの内部の構造は図 1.1 に示した様になっている。

まず内部のレーザーから照射されたレーザー光が中央の感光体ドラムに当たる事により帯電していたその箇所が露光し、その箇所にインクが付きそのインクが紙に転写され、それを定着させる事により印刷を行っている。

従って、レーザーのずれは印刷した際の仕上がりに大きく影響してしまう。

たとえばレーザーが縦にずれたとすると、直線が連続的でなくなってしまうため、この様にレーザーの調整は印刷において非常に重要な作業であるといえる。

## 2.2 製造時のレーザービームの微調整

以上のような理由よりレーザービームの微調整が必要であるといえる。しかし、現在も微調整自体は行われているが一つのプリンタのレーザー全域を一度に確認する事はできていない。そのため微調整は非常に重要な作業であるにも関わらず効率的には行えていない。また、レーザー全域を一度に確認する技術はすでに存在しており、レーザープリンタを製造している企業は様々な手法を用いて検査を行っていると考えられるものの秘匿性が高く流用は難しい現状にある。

このため、現在よりも高速にレーザー全域を確認する事が微調整作業の高速化および精度向上のために非常に望まれている。

## 2.3 全領域測定治具とその構成

現在レーザーの照射位置を確認するために用いているイメージセンサーは単体ではレーザー域全体をキャプチャできない。製造時のレーザービームの微調整の効率を上げるには全領域のキャプチャが可能な全領域測定治具が必要である。全領域キャプチャ実現には複数台のセンサーを用いる。その個々のセンサーはレーザービームの照射位置をグレースケール

画像としてキャプチャするものである。

別途、開発中の全領域キャプチャ機構においてはプリンタ1台につき9つのイメージセンサーを用いている。キャプチャした画像は1台につきセンサー3つずつ接続されているFPGAボードで処理され、通信路を通りPCに送られて、検査される。省スペース化、コスト削減を図るために、プリンタ1台分としてFPGAボード3枚からの出力を1台のPCに接続している。

通信路にはカメラメーカーとボードメーカーが、コネクタやピンアサインや使用するケーブル等を標準化した産業用カメラの出力規格として高い実績があるカメラリンクインターフェースを用いる。

## 2.4 転送画像のデータ量

センサーでキャプチャする画像は $1944 \times 1092$ の256階調のグレースケールで、接続されているセンサーは30fpsの速度で画像をキャプチャし出力する。

FPGA1台にはセンサーが3つ接続されており、PCにはFPGAが6台接続されているので、ビットレートは $1944 \times 1092 \times 8 \times 540$ で、およそ9170Mbpsである。

これは非圧縮の 4:2:0 ハイビジョン放送の 12 倍もの転送データ量である。

## 2.5 画像圧縮転送の必要性

前述したように本研究で扱うビットレートはおよそ 9170Mbps と非常に大きく、このビットレートは今回使用するカメラリンクでのビットレートの 680Mbps よりも非常に大きく、PC 間転送の速度制約として、現在高速な通信路として有用とされている規格の USB3.0 でさえ理論値は 5Gbps となっておりこのままではリアルタイムの転送要求に応えられない。

これらをリアルタイムで処理するためには画像の圧縮が必要不可欠であり、圧縮、伸張の処理時間を考慮しなくても最低でも 1/13.5 のサイズに画像データを圧縮する必要がある。

そこで本全領域測定治具では、FPGA 内でセンサーから出力された画像のエンコードを行い、通信路を経由し PC でデータを受け取り、そのデータに対してデコードを行う。

## 2.6 対象画像の特徴

今回扱う対象画像は図 2.2 のようなもので、画素値に変化が余り見られないところがほとんどであり、重要なビームスポット付近でのみ画素が大きく変化しているという特徴がある。

また，検査の際の要求として圧縮は完全な可逆圧縮である必要はないが，  
ビームスポット付近は可逆でなければならない．

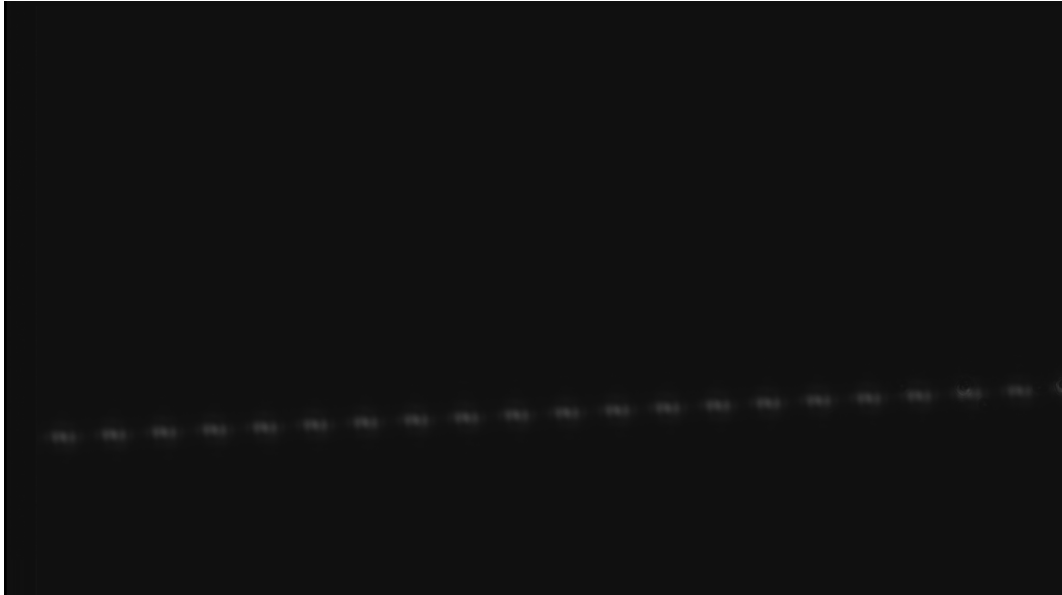


図 2.2: 対象画像

## 3 既存の圧縮アルゴリズムとその問題点

### 3.1 既存の候補アルゴリズム

本研究の対象画像を圧縮する際の選択肢として、現在様々な特徴を備えた圧縮コーデックが複数存在している。画像関係の圧縮コーデックでは JPEG や PNG などがよく知られている。また、今回は特定の傾向を備えた画像に対して圧縮を行う為、画像専用の圧縮コーデックだけでなく様々なファイルに対し圧縮を行える ZIP などの汎用的な圧縮コーデックも検討する必要がある。しかし、対象画像の特徴で前述したとおり、対象画像のビームスポット列付近はデータの劣化を認める事が出来ないため JPEG などの非可逆圧縮方式の圧縮は用いる事が出来ない。

それらを踏まえ可逆圧縮方式である GIF , PNG , ZIP , GZIP について圧縮率の評価を取った。

評価結果については表 3.1 にまとめた。

表 3.1 から分かるように、今回扱っている対象画像に対しては GIF が最も圧縮率が高い事が分かった。結果として他の圧縮コーデックとそれほど圧縮率に差はでなかったものの、本研究では非常に大きなフレームレートを処理しなければならないので最終的には大きな差になると考えられる。以上より GIF コーデックを本研究で使用する圧縮コーデックにとし

て採用するか考察する対象とする。

	サイズ (KB)	圧縮率 (%)
GIF	161	7.8
PNG	180	8.7
ZIP	184	8.9
GZIP	183	8.8

表 3.1: 各圧縮コーデックの圧縮率評価

## 3.2 GIF コーデック

### 3.2.1 特徴

GIF はアメリカのパソコン通信ネットワーク CompuServe で画像交換用に開発された画像ファイルフォーマットで、LZW と呼ばれる辞書式圧縮法が用いられている画像圧縮コーデックである。辞書式圧縮法とは出現したビット列を辞書に登録し、登録したページ番号を用いて符号化する方法である。

256 色以下の画像を扱う事ができる可逆圧縮形式のファイルフォーマットであるため、1 画素が 8bit256 色であり、画像の劣化が許されない今回の対象画像に対して有効であるといえる。また、GIF は色、明るさ共に



同じである画素が連続している画像に対し高い圧縮率が期待できるという特徴があり、本研究で対象としているセンサーを用いてキャプチャしたビームスポット列画像はビームスポット付近でのみ明るさの変化が目立つグレースケール画像であるため、処理速度を考慮しなければ既存の画像圧縮コーデックの中では最適といえる。

また、処理についても比較的初期に作られたコーデックであるためアルゴリズムが単純で JPEG などと比べて高速だと言われている。

### 3.2.2 問題点

本研究では 540fps という非常に大きなフレームレートで画像を処理する必要があるが、GIF の圧縮で使用されている LZW 圧縮法は可変長符号化であるため動的にテーブルを用意する必要があり、ハードウェアでの実装が難しく、また、並列処理を行う事ができなくなってしまうために高速化の妨げになり、要求フレームレートを満たす事ができなくなる可能性がある。また、他の画像圧縮コーデックよりは高速と言われているが、あくまでそれらと比べて高速であるだけであり、リアルタイムで複数の画像を圧縮するには速度が足りないと考えられる。

これらの問題点を踏まえ、開発期間低減および速度要求の実現のため

にハードウェアで実装しやすく、高速で動作させる事ができる可変符号長を用いない GIF コーデックに代わる可逆圧縮形式のアルゴリズムを確立する必要がある。

## 4 ビームスポット列画像向きアルゴリズムの提案

### 4.1 提案アルゴリズム

#### 4.1.1 アルゴリズム概要

以下では実際に実装したアルゴリズムの概要について説明する。

本圧縮アルゴリズムでは、ランレングス符号化と差分符号化を併用している。対象画像のビームスポット付近以外では同じ画素が連続的に出現しやすいという特徴を利用し、ビームスポット以外をランレングス符号化で大幅に圧縮を行い、対照的に画素の変化が激しいビームスポット付近では差分符号化を用いてランレングス符号化のみを使った場合に起こる符号量の増加を抑えている。

ランレングス符号化は、符号化対象が画素に対して同様の画素が何回連続で出現するかを記録し、その出現回数とその画素値を符号化する符号化手法である。

また、差分符号化は単純に対象画素の一つ前の画素との画素値の差分をとり、その差分を符号化する符号化手法である。一般的にDPCMなどではこのような符号化を行う際は前画素との単純な差分ではなく、この単純な差分と、二つ前の画素と一つ前の画素との差分との差分をとる事により予測精度を高め符号化を行うが、本研究では高速化と実装期間の短

縮に向けての処理低減と単純差分でもある程度の圧縮精度が得られたことから単純な差分で実装している。

また、差分発生時でも小さな差分値の出現頻度が大半を占めているため、差分符号化の差分部分の符号長を階層的な固定長符号化で実装し、差分値によって符号長を変更することで圧縮率の向上を図った。これによりランレングス符号化ほど大幅な削減はできないが確実な圧縮が可能となっている。

ランレングス符号化の使用条件は同様の成分を持った画素が連続的に続いた場合で、それ以外の場合は差分符号化を用いる。

各圧縮法の適応イメージを図 4.3 に示す。

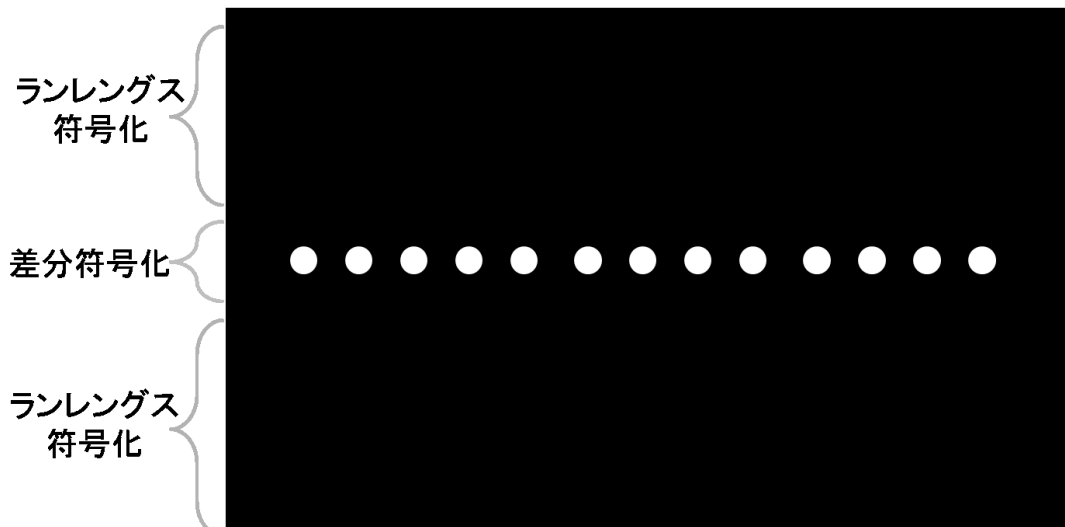


図 4.3: 符号化イメージ

#### 4.1.2 符号化の仕様

前述のアルゴリズムで符号化した際に発生する符号について説明する。本アルゴリズムではランレングス符号化と差分符号化の二つの符号化手法を併用しており、固定長でビット幅の階層化も行っているため、そのままではデコーダ側でどちらの符号化手法で符号化したものか、差分符号化ではどの階層で符号化したものかを判別する事が出来ない。そこで本アルゴリズムでは生成した符号の先頭に 3bit のマーカーを設置することによりデコーダ側で判別可能にしている。

まず固定長でビット幅の階層化では 3bit,6bit,9bit の 3 種類のビット幅で階層化している。

3bit での符号化はバイナリで 0 7 の 8 値を表現する事が出来るが、本研究では 8 値の内 3 値を前述のマーカ-に用いていて、残りの 5 値を差分符号化に割り振っていて、差分が -2 2 のときは 3bit での差分符号化になる。同様に 6bit での符号化では 6bit で表現できる 64 値に -31 32 の数値を割り当て、差分値がこの範囲の場合 6bit での差分符号化している。そして、6bit で差分値が収まらない場合は 9bit を用いて差分符号化をしている。6bit,9bit の差分符号化ではそれぞれ符号の先頭に 3bit のユニークなマーカ-を設置している。

同様にランレングス符号化では、先頭に 3bit のユニークなマーカ-を設置後 9bit を用いて符号化対象画素の前の画素との差分を格納しており、その後 12bit を用いて符号化する画素が何個あるかを示すランレングス長を格納している。

以上の符号化の仕様をまとめたものを図 4.4 に示す。

3bit符号化	
000	差分
001	
010	
011	
100	
101	マーカ-
110	
111	

6bit符号化	
マーカ-(110)	差分(6bit)

9bit符号化	
マーカ-(111)	差分(9bit)

ランレングス符号化		
マーカ-(101)	差分(9bit)	ラン長(12bit)

図 4.4: 符号仕様

## 5 高速デコーダの実装

### 5.1 処理方法

実装したデコーダの処理について簡単に説明する。

実際の治具では符号化したデータは12bit単位で入力してPCに入ってくる。しかし、符号化したデータ自体の最小単位はマーカ-と差分符号化の3bitでの符号化であるため3bitである。そこで本デコーダでは、まず12bitの入力データを3bit単位で分割し、これを1フレーム分連続で行う。その後分割した3bitのデータを一つずつ読み込み、マーカ-かどうかの判定を行う。マーカ-でないなら差分符号化の3bitでの符号化なので既にデコードしている前画素に記録している差分を加え画素をデコードする。

次に3bitがマーカ-だった場合マーカ-の中身を見てどの種類の符号化が行われているのか判定する。6bit,9bitでの符号化のマーカ-だった場合それぞれ次の6bitまたは9bitに差分が格納されているので、それを読み込み3bitでの符号化のデコードと同様に前画素に記録している差分を加え画素をデコードする。

ランレングス符号化のマーカ-だった場合は次の9bitに渡り差分が格納されているので差分符号化と同様の処理を行い、まず1画素のみデコー



ドする．その後，次の 12bit に渡りランレンクス長が格納されているので

先ほどデコードした画素値をランレンクス長分だけ格納する．

以上のデータの分割以外の処理を 1 フレーム分の画素をデコードし終わ

るまで繰り返すことでデコードが完了する．

上記を簡単にまとめたフローチャートを図 5.5 に表す．

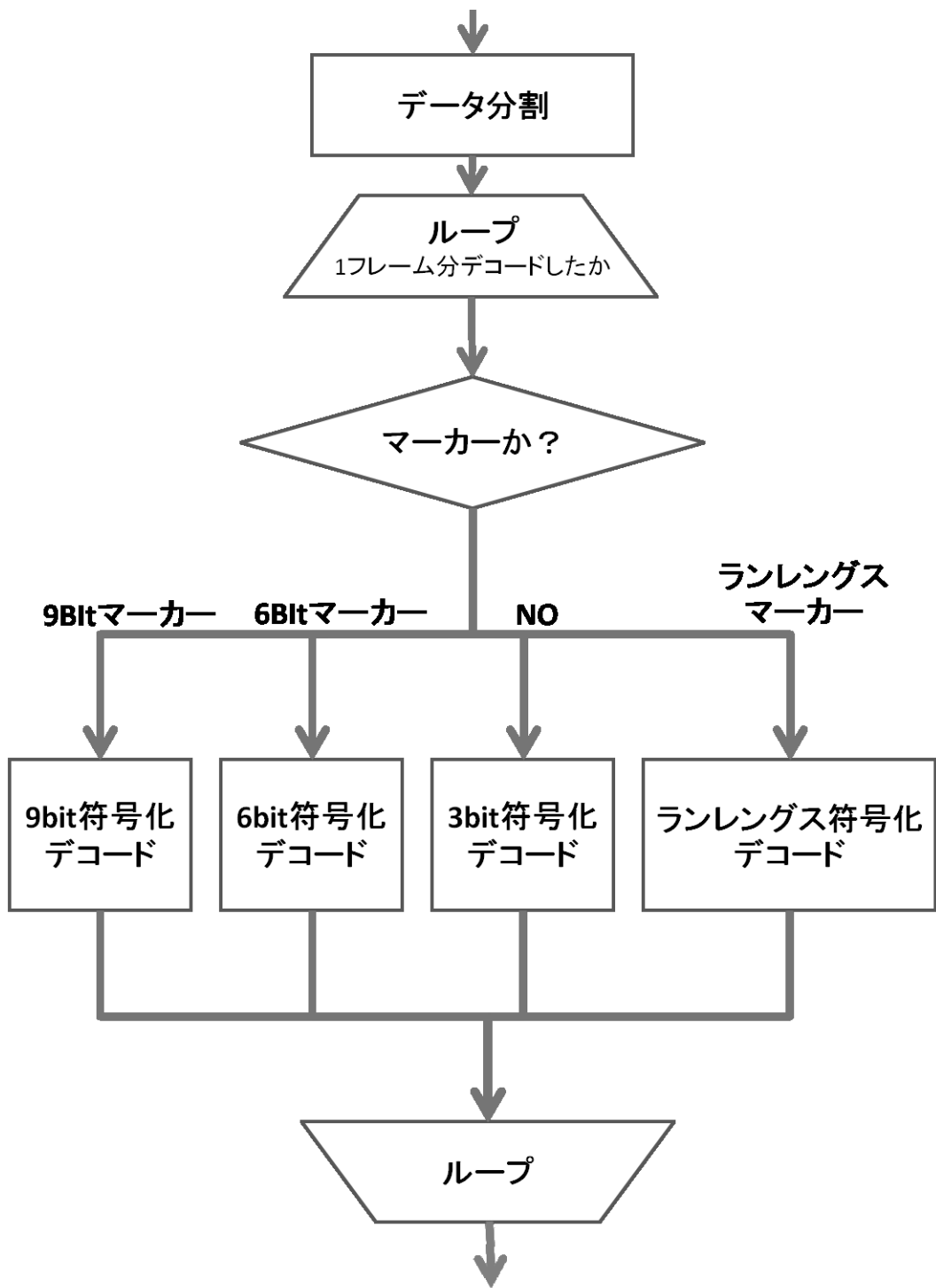


図 5.5: 簡易フローチャート

## 5.2 高速化

本研究で実装を目指している治具では非常にビットレートが大きいいため大幅な処理時間の低減が必要であるため高速化を行った。

重い処理を行う場合の高速化ではマルチスレッド、マルチタスクなどが有効だが実装期間の関係で共に行わず最適化による高速化を行った。また、最適化を行わずにマルチスレッドなどを行った際、後に最適化などを行い処理を変更してしまうと既存のマルチスレッド処理が出来なくなってしまう場合があるため、マルチスレッドなどを行う場合でも先に最適化は行うべきである。

実行した最適化手法については基本的なループ展開、可読性を損なわないレベルでのインライン展開、条件分岐による処理回数の偏りを用いた最適化である。

ループ展開は、ループ内の処理を直接記述する事により条件判定回数を減らし高速化する手法である。デメリットとしてコード量が増えサイズが大きくなってしまいが、今回組み込みなどと違いリソースは十分にあると言えるので問題はないと考えられる。

インライン展開は、呼び出し先の関数での処理を呼び出しもとに直接書く事により関数呼び出し時に発生するオーバーヘッドを削減する事が出

来る .

条件分岐による処理回数の偏りを用いた最適化では , 実際にエンコーダーで対象画像をエンコードし、その際の各符号化方法の出現回数を調査し出現頻度がもっとも大きい符号のデコード処理ほど条件判定や処理が少なくなる様に処理や処理順番を変更した .

大まかには以上のような最適化処理を行うことにより高速化を図った .

## 6 評価

### 6.1 評価方法

実際に全領域測定治具を動作させる環境での評価は治具が未完成なためできないので、同一環境で GIF と本研究での作成物とでデコード速度および圧縮率を比較し評価をとる。

圧縮対象はレーザービームスポット列画像のサンプル画像で、非圧縮のグレースケールの画像フォーマットの pgm である。この画像を各フォーマットに変換後、linux の time コマンドを用いてそれぞれの生成物に対してデコードを行う際のユーザー CPU 時間を算出し、速度とする。この際の精度は 1m 秒である。

ただし、評価を行う際同様の検査を連続的に 100 回行い、その平均を評価結果としている。

### 6.2 評価結果

評価結果を表 6.2 に示す。

対象画像には 2074KB の実際のビームスポット列画像を使用している。

	ユーザー CPU 時間 (ms)	サイズ (KB)	圧縮率 (%)
GIF	58	161	7.8
本研究	1.5	61	2.9
要求	1.7	160	7.7

表 6.2: 圧縮率および伸張速度表

### 6.3 考察

圧縮率では GIF フォーマットが 7.8% で本研究のフォーマットでは 2.9% である。当初の圧縮率の最低要求が 1/13，つまり 7.7% であるため，GIF フォーマットではわずかではあるが満たせておらず，本研究のフォーマットでは目標値のおよそ 2.7 倍の圧縮率を記録している。また，圧縮率が高くなればなるほど処理に割ける時間に余裕が生まれるため，本研究で作成したフォーマットは対象画像に対して非常に有効であるといえる。

速度に関しては，GIF フォーマットの 58ms に対し 1.5ms と，こちらでも 38.7 倍と非常に高速に動作することが確認できた。速度要求が 1.7ms である事を考慮すると既存の GIF デコーダーでは 34.1 倍の高速化を行わなければならない，困難である。

一方本研究で実装したデコーダは 1.5ms であるため速度要求を十分に満たしているといえる。

よって本研究で開発を行った圧縮アルゴリズムおよびそのデコーダは全  
領域測定治具を実装するに当たり十分に有効であるといえる。

## 7 あとがき

本研究では、レーザープリンタのレーザービームの照射範囲全域を1度に検査し、検査精度向上および、検査時間短縮を目的としている全領域測定治具に最適な圧縮アルゴリズムを確立すると共に、そのアルゴリズムに準拠する高速で動作するソフトウェアデコーダーを開発した。

また、その圧縮アルゴリズムおよびソフトウェアデコーダを評価し、対象画像であるビームスポット列画像に対して、既存の圧縮コーデックよりも高圧縮で高速であることを示した。また、全領域測定値具の実装の際の速度要求、圧縮率要求も共に満たしていることを示した。

しかし、本研究のアルゴリズムでは開発期間が限られたためDPCMで一般的に用いられている予測誤差を用いた予測精度を高める処理を省いたので、予測誤差をと取り入れることによりさらなる圧縮率の向上が見込まれる。またマルチコアCPUを用いての高速化で非常に有効といえるマルチスレッド、マルチタスク処理も行っていないため、これらを実装する事により大幅な高速化が期待できる。

今後、そういった処理を組み込む事により一層の圧縮率向上と高速化を図る必要がある。



## 謝辞

本研究を遂行するにあたり，日頃から熱心にご指導，ご助言をしてくださった近藤利夫教授，佐々木敬泰助教に深謝致します．また，計算機アーキテクチャ研究室の諸氏には本研究の様々な局面にて多大なご協力を頂きました．ここに感謝の意を表します．