

平成 15 年度 修士論文

Content-oriented 符号化実現のための
アニメーション映像を対象とする符号化に関する検討

指導教授 渡辺 裕 教授

2004 年 3 月

早稲田大学大学院 国際情報通信研究科
国際情報通信学専攻

4202A150-3

中 神 央 二

Ouji NAKAGAMI

目次

第1章	序論	1
1.1	研究の背景	1
1.2	研究の目的	3
1.3	本論文の構成	4
第2章	アニメーション画像の特徴と符号化	6
2.1	はじめに	6
2.2	アニメーション画像の特徴	6
2.2.1	アニメーション画像の分類	6
2.3	アニメーション画像に対する周波数分解による符号化	7
2.3.1	DCTを用いた静止画像の符号化	7
2.3.2	アニメーション画像と符号化歪み	10
2.4	ベクトル記述言語による画像符号化方式	11
2.4.1	Macromedia Flash	12
2.4.2	SVG (Scalable Vector Graphics)	12
2.5	むすび	13
第3章	ベクトル記述言語変換のための関数近似手法	14
3.1	はじめに	14
3.2	パラメトリック曲線	14
3.2.1	エルミート曲線	14
3.2.2	ベジェ曲線	15
3.2.3	B-スプライン曲線	17
3.2.4	NURBS 曲線	18
3.3	数学的補間	19
3.3.1	ラグランジュ法	19
3.3.2	スプライン法	20
3.4	ベクトル変換アルゴリズム	22
3.4.1	Dynamic Programming による最適近似	23
3.5	SVG によるベクトル変換データの出力	24
3.6	むすび	29

第4章	ベクトル変換に基づくアニメーション静止画像符号化方式	30
4.1	はじめに	30
4.2	形状情報を用いたアニメーション画像の符号化	30
4.2.1	アニメーション画像からの形状情報抽出	30
4.2.2	抽出した形状情報のベクトル変換	32
4.3	従来方式の問題点	34
4.4	二層構造によるアニメーション画像の符号化の提案	34
4.4.1	画像平滑化手法による均等色領域抽出手法	35
4.4.2	結果と考察	41
4.4.3	考察	41
4.5	むすび	49
第5章	アニメーション映像内のセル画領域同定法	51
5.1	はじめに	51
5.2	アニメーション映像における動きの分類	52
5.3	差分画像を利用したオブジェクト抽出手法	53
5.3.1	動領域の抽出	53
5.3.2	大局的な動きへの対応	61
5.3.3	実験結果	69
5.4	抽出したセル画領域のマッチング	71
5.5	むすび	74
第6章	結論	82
6.1	総括	82
6.2	今後の課題	83
6.2.1	静止画における課題	83
6.2.2	動画における課題	84
	謝辞	85
	参考文献	86
	図一覧	92
	表一覧	95
	研究業績	96

第1章

序論

1.1 研究の背景

動画像・音声圧縮符号化方式として、ISO/IEC JTC1（国際標準化機構・国際電気諮問委員会）において規格化された MPEG-1, MPEG-2 は、蓄積メディアや、通信・放送の分野で広く用いられている [1][2]。また、近年策定された MPEG-4[3] や、その拡張である MPEG-4 Part 10, AVC (Advanced Video Coding) は、より低ビットレートでの動画像符号化方式として、期待がもたれている。

MPEG-1, 2, 4 では、DCT(Discrete Cosine Transform) を基盤にした符号化方式が用いられている。DCT は、時間領域にある画素の信号値を、周波数領域の係数に変換する方法のひとつである。DCT を用いると、画像信号を、その平均値（直流成分）と、明るさの変化が少ない成分（低周波成分）と、明るさの変化が激しい成分（高周波成分）に分離することが可能になる。

人物や風景など、自然画像の主な成分は、低周波成分である。そこで、DCT の後に量子化とよばれる作業が行われる。量子化を行うことで、視覚にとって目立ちにくい高周波成分を取り除くことができる。これにより、データ量を減らすことを可能にしている。

静止画像の符号化としては、JPEG や、JPEG 2000 が挙げられる [4][5]。JPEG は、1990 年に、ISO 10918-1(ITU-TS T.81) として規格化された。JPEG の符号化方式は、DCT を基盤にするという点において、MPEG と同じである。JPEG は、Intra フレームにおける MPEG の符号化と考えることができる。

JPEG 2000 は、2001 年 1 月に規格化された。DCT の代わりに、Wavelet 関数と呼ばれる基底関数を用いて、周波数変換が行われる。JPEG 2000 の特徴として、ピラミッドコーディングが挙げられる。MPEG や JPEG では、画像をいくつかの小ブロックに分け、その単位で DCT, 量子化を行う。JPEG 2000 では、このブロック分割の代わりに、フィルタにより周波数成分ごとに分割される。これにより、ブロック歪みと呼ばれる、ブロック間の符号化誤差による歪みが出現しない。

以上述べてきた符号化方式は、全て、画素情報の周波数領域への変換を基礎としている。この、変換符号化方式が、盛んに研究され、また普及している理由として、いかなる符号化対象画像に対しても、一定の性能を得られることが挙げられる。さらに、自然画像に対して

は、特に圧縮の効果が高い（見た目の性能劣化が少ない）ことも特徴である。

ところで、自然画像の他に、アニメーションやCG(Computer Graphics)といった、一見して自然画とは画像的性質が異なると思われるコンテンツも存在する。本論文では、これを便宜的に人工画像と呼ぶ。アニメーションは、日本では特に人気の高いコンテンツのひとつで、当然高品質な符号化の需要が存在すると考えられる。

これまで、人工画像に対しては、2つの対応が行われてきた。自然画像と同様に、周波数変換を基盤とした符号化方式を用いる方法がひとつである。伝送・蓄積メディアなど、既存のコンテンツの流通ネットワークで配布を行う際には、この方法がとられることが多い。

もうひとつは、アニメーション特有の画像の構造を利用して、専用の記述方式を用いて表現するという手法である。人工画像を構成する図形要素は、線や領域、面などの単位に分割可能なことが多い。そのため、各要素を数値で簡潔に表現するためのフォーマットが選択される。このようなフォーマットは、ベクトル(ベクタ)フォーマット、あるいはベクトル記述言語と呼ばれる。専用アプリケーション間でのデータ交換、また、インターネットなど、デコーダをソフトウェア的に配布可能な環境では、この方法がとられることが多い。

前者については、既存のインフラストラクチャを利用できるという利点がある反面、特に高圧縮時に、圧縮による符号化誤差が顕著になるというものが挙げられる。一般に人工画像では、信号レベルが極端に変化するエッジと呼ばれる部分が、画像中に多く存在する。周波数の観点からみると、これらは高周波成分に相当する。量子化によりこの部分の係数がカットされると、復号時にノイズとなる。この現象は、モスキートノイズと呼ばれる。DCTの代わりにWaveletを用いた場合でも、圧縮のため高周波成分を犠牲にする限り、同様の現象がおこる。そのため、人工画像の高圧縮のためには、周波数変換を基盤とする符号化方式は適していないといえる。

後者の専用の記述方式としては、これまで様々なベクトル記述言語が提案され、用いられている。近年、とりわけ広く用いられているものとして、Macromedia社のFlashや、W3C標準のSVG(Scalable Vector Graphics)が挙げられる。いずれも、任意の線や面、ポリゴンを組み合わせることで、アニメーションとして表現することが可能である。

この方式の特長として、ベクトル形式で記述するため、ラスタ形式に比べ、極めて少ない情報量で済む。通常、2次元画像で線や面を表現するためには、画像上の全ての点について、座標と色情報が必要である。しかし、ベクトル記述言語では、端点や通過点、制御点といった図形の特徴をあらわす点(以後、有意点と呼ぶ)のみでよい。再生時には、デコーダが有意点間を数式で補間することで、元の図形を再構成することが可能になる。コンテンツを作成する際に、決められたフォーマットに従ってデータを記述する際には、適した方式といえる。

近年、プロセッサ処理能力の著しい向上、また記憶メディアの大容量化に伴い、あらゆる分野でのコンテンツのデジタル化が進んでいる。そこには、自然画像以外のコンテンツも多く存在し、その画像的性質は、様々である。そのため、MPEGなどにより、すべてのコンテンツを統一的に符号化する方式は、利便性が高い反面、符号化効率の面で難点があるといえる。そこで考えられるのが、個々のコンテンツの性質を考慮して、各符号化対象に適した

符号化を用いる，コンテンツオリエンテッド (Content-oriented) な符号化方式である。

本論文では，この考えに基づき，符号化の対象として，セルアニメーション映像に着目した。日本で最も人気の高い人工画像のコンテンツのひとつであり，セル画による彩色という，形状符号化に適した性質を持つためである。符号化方式としては，ラスタ形式からベクトル記述言語への変換に着目した。ベクトル形式での形状符号化は，高圧縮が期待できるうえ，モスキートノイズなど変換符号化特有のノイズがないためである。

1.2 研究の目的

これまで，形状情報から特徴量を抽出し情報量の圧縮を図るという手法は，フォントの形状や地図情報を対象として行われていた [35]。対象の点列を正確に復元できるような有意点の座標を抽出する技術である。元のデータ量に比べて，大幅に圧縮することが可能になる。ただし，その対象は，直線や，滑らかな弧が主体の形状である。また，計算機の記憶容量や処理速度の問題から，2 値画像に限られてきた。

対象図形中の直線や曲線は，数学関数により近似されることで圧縮される。近似の手法として，もっとも基本的なものとして，一次関数によるものが挙げられる。すなわち，任意曲線の直線分割による近似である。曲線を，曲線関数で近似する方法も，研究されている。スプライン関数やベジェ関数を用いて，任意の曲線をその通過点や制御点で表現しようという手法である。有意点の抽出に，より多くの計算コストを要するが，滑らかな近似曲線を得られるという利点がある。

関数による近似手法を用いるもう一つの利点として，数式を用いて近似線の情報を復元するため，任意の解像度で出力を可能にすることが挙げられる。ラスタ画像では，出力の解像度を変更することは，画質の劣化を招く。画素間の情報がないので，周辺画素の情報から補うか，間引くことが必要なためである。対してベクトル画像では，近似曲線が別に定義されている。出力解像度は，その演算精度に応じて変更可能である。

しかし，対象がアニメーション画像の場合，フォント形状や地図情報の手法をそのまま用いることは難しい。ラスタ画像中の線や面から，有意点を抽出する際に問題が生じる。

第一の理由として，符号化対象の画像から，線要素及び輪郭形状の情報を抜き出すことの難しさが挙げられる。フォントや地図情報の場合，対象画像は白黒のため，有意点を抽出すべき線情報の分離が容易である。これに対し，アニメーション画像の場合，色情報を含んでおり，線や面が複雑に交わることもある。そのため，線や領域の抽出を正確に行うことが困難である。したがって，アニメーション画像の性質を考慮した形状抽出のアルゴリズムが必要になってくる。

次に，抽出した線分が，フォントや地図に比べ，複雑な曲線であることが多い。キャラクターなどを表現しているため，鋭角な曲率や，円弧，直線などが交じり合っているためである。複雑な曲線を関数により近似するためには，多くの制御点が必要となる。そのため，正確な近似のための制御点を探索する必要がある。

アニメーション映像中には，ベクトル記述言語では，表現不可能な情報も存在する。セル

画で描かれたキャラクターなど、単色で描かれている部分は、ベクトル記述言語による符号化の効果が大きい。しかし、水彩画で描かれた背景など、自然画に近い性質の領域においては、細かいテクスチャ情報が存在する。ベクトル記述言語は、形状の符号化には適しているが、テクスチャなどの符号化には適さない。そこで、背景領域が含まれる場合への対応を考える必要がある。

以上述べてきたことは、静止画に対する符号化の問題点である。動画への符号化に拡張するためには、さらに、アニメーション作品の特徴を考慮する必要がある。

これには、セルアニメーションが、有限なセル画と背景の組み合わせの撮影により構成されることに着目する。つまり、一枚のセル画を、異なるコマの撮影にも用いるため、フィルム中に冗長性が存在する。この冗長性を発見することで、大幅な符号化効率の向上を見込むことが可能になる。

MPEG では、フレーム間の符号化に、動き補償、動き予測という技術が用いられている。一般に、動画シーケンスは、時間的に相関が高い。つまり、連続するフレームはよく似た画像であることが多い。そこで、既に符号化された前フレームのブロックから、現フレームへのリンクを張ることによって、新たに係数を符号化する必要がなくなる。ブロックに含まれる係数と、リンクの情報では、符号化するのにかかるコストは、後者の方が圧倒的に少ないためである。これが動き補償であり、このリンクは動きベクトルとよばれる。そのため、どのブロックからの動きベクトルが、当該ブロックと一番相関が高いのかを調べる必要がある。これは動き予測とよばれる。

MPEG では、動き補償、予測は、画素の処理と同様、ブロック単位で行われる。しかし、本論文で提唱する符号化は、均等色の領域や線といった画像の構成要素単位での符号化である。そのため、この概念を拡張して、オブジェクトベースでの動き補償、予測を目標とする。

補償を行う単位は、セル画による領域である。全く同一の領域が別のフレームにも登場する可能性があるためである。これにより、既に符号化した領域情報を、そのまま利用することができる。一方で、予測を行うには、まず、どの部分がセル画による領域かを知る必要がある。フィルムには、セル画領域は背景などと合成されて撮影されるため、何らかの判定が要る。そして、判定した領域が、他のフレームに登場するか否かを探するためのマッチングも必要になる。

まとめると、本論文は、アニメーション映像の符号化に関する検討を行う。符号化には、形状の関数近似によるベクトル記述言語への変換を基盤とする。その際、画像中からの形状抽出方法、テクスチャの符号化方法について考察する。さらに、動画へ拡張するため、フレーム間符号化についての考察を行う。セル画領域の冗長性に着目し、領域ベースでの動き予測、補償のための検討を行う。

1.3 本論文の構成

第1章では、本研究の背景および目的について示す。

第2章では、DCT ベースの符号化方式が、エッジ成分を多く含むコンテンツに適さない

という問題点を挙げる．現在普及しているベクトル記述言語の紹介を行う．

第3章では，ベクトル変換のためのアルゴリズムについて述べる．近似のための数学的関数とその補間アルゴリズム，さらに符号化の基盤となる近似アルゴリズムについて説明する．また，符号化データの出力についても説明する．

第4章では，コンテンツオリエンテッド符号化方式の一つとして，セル画アニメーションの静止画像に特化した符号化方式について述べる．平滑化操作による，画像からの効率的な構成要素抽出法について説明する．また，画像をベクトル記述言語による層と，テキストチャ情報のみの層に分けることで，効率的に符号化を可能にする方式について検討する．

第4章では，提案する符号化方式を，静止画から動画へ拡張するための足がかりとして，アニメーション映像に含まれる，同一セル画の同定の手法について述べる．アニメーション映像からのオブジェクト抽出法，抽出オブジェクトの検索手法について述べる．

第5章では，本論文のまとめを行い，今後の課題について述べる．

第2章

アニメーション画像の特徴と符号化

2.1 はじめに

本章では、まずアニメーションの静止画像の特徴についてまとめる。次に、それを踏まえた上で、アニメーション画像に周波数分解による符号化を用いた場合に問題となる点を指摘する。そして、人工画像に適した符号化方式として、ベクトル記述言語に基づくものを紹介する。

2.2 アニメーション画像の特徴

人工的に作られたアニメーション画像は、自然画像に比べて、限定されている部分や特徴が多く見受けられる。

一般に、アニメーション映像では、セル画と呼ばれる透明なシートの上に、主人公などのキャラクターを描いて制作されている。背景部は、背景画として1枚の大きな絵を用意しておき、キャラクターの描かれているセル画は、背景画の上に重ねて撮影される。アニメーション映像製作の様子を図2-1に示す。

セル画の部分は、動きを表現するために大量に似た絵を作成する。そのため簡単に作成できるように、グラデーション等複雑な表現は使用せず、べた塗りで描かれていることが多い。逆に、背景画は制作枚数が少なく、立体感をだすため水彩画で描かれていることが多い。これにより、セル画に比べると自然画像に近い絵となっている。

近年、多くの制作スタジオが取り入れているデジタルアニメーションでは、着色、合成、撮影の部分をデジタル化し、コンピュータ上で行っている。しかしながら、大量の枚数を作成することに変わりはないため、キャラクター部は単色で着色されていることが多い。

2.2.1 アニメーション画像の分類

前節で述べた、アニメーション画像の特性に着目し、アニメーション画像を次のように分類する。

均等色領域 主にキャラクタなど、セル画で描かれた部分を指す。但し、後述の背景画像内

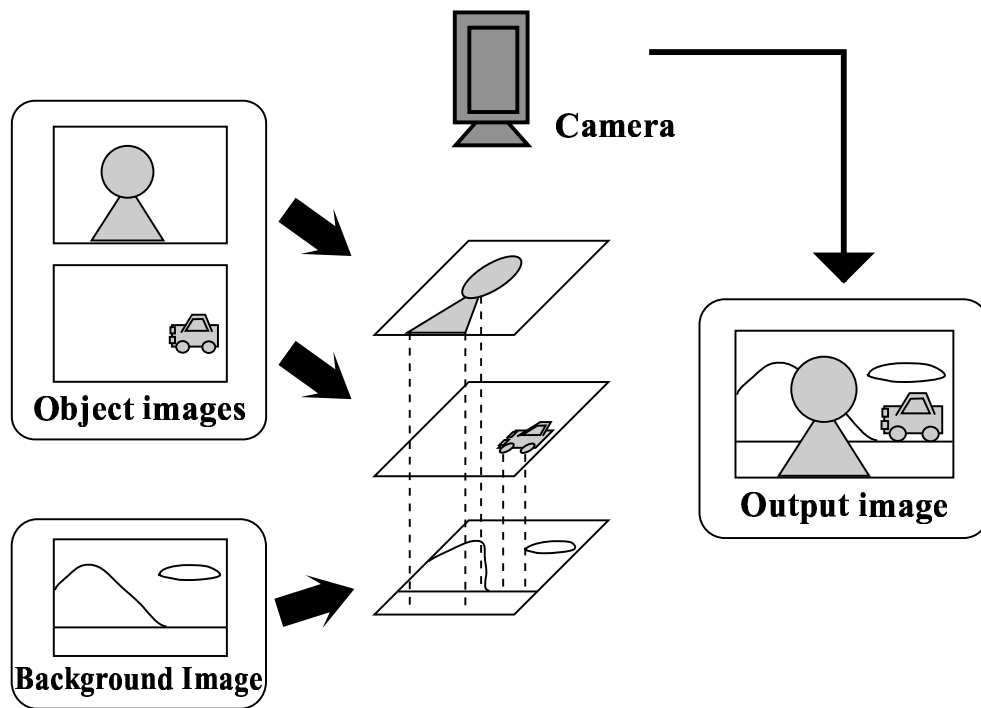


図 2-1: アニメーション画像の撮影

にも均等色の領域は含まれる。

背景領域 背景など，水彩画で描かれた，自然画に近い部分を指す。

線画 キャラクターの輪郭や，衣服のしわ，顔の表情などを描いている，主に黒の細線部分を指す。

この分類を，実際のアニメーション画像上に示したものを図 2-2 に示す。

2.3 アニメーション画像に対する周波数分解による符号化

2.3.1 DCT を用いた静止画像の符号化

DCT を用いた静止画像符号化方式の基本構成を図 2-3 に示す。符号化器において，入力画像はブロック単位で DCT，量子化，そして最後にエントロピー符号化が行われ，符号化データが出力される。復号器では，逆に圧縮された符号化データに対してエントロピー復号，逆量子化，そして IDCT が施され，復号画像が出力される。

品質の劣化が発生する原因は，非可逆操作である量子化の部分にある。まず，量子化の前段階である DCT 処理と，量子化の仕組みについて述べる。

DCT 処理 DCT 変換は，ブロック内の画素情報を，周波数領域へ変換するために用いられる。JPEG の場合，ブロックの大きさは，8 画素 × 8 ラインであり，ブロック単位に 2 次

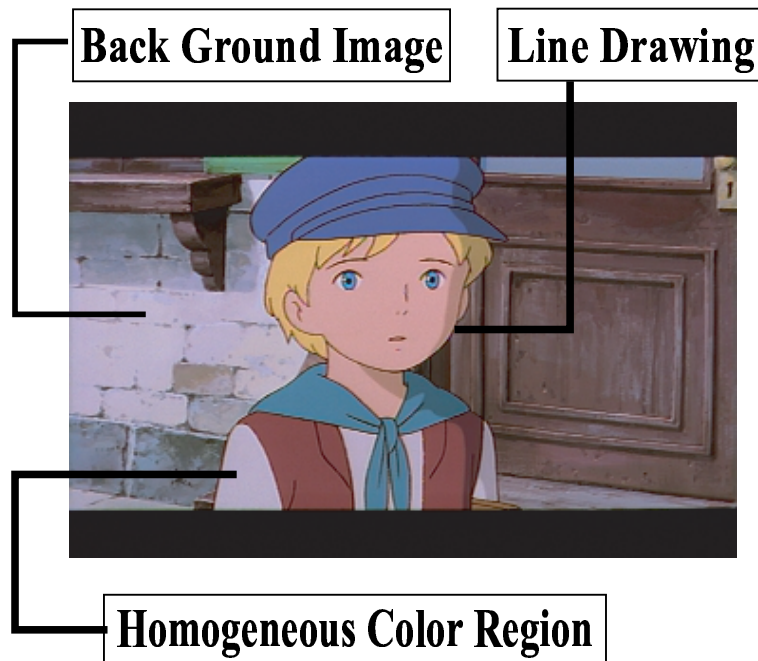


図 2-2: アニメーション画像の分類

元 DCT を施す．2次元 DCT と，その逆変換は，式 (2.1)，(2.2) により行われる．

$$\begin{aligned}
 F(u, v) &= \frac{2C(u, v)}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j) \cos\left(\frac{(2i+1)u\pi}{2N}\right) \cos\left(\frac{(2j+1)v\pi}{2N}\right) \\
 f(i, j) &= \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u, v) F(u, v) \cos\left(\frac{(2i+1)u\pi}{2N}\right) \cos\left(\frac{(2j+1)v\pi}{2N}\right)
 \end{aligned} \quad (2.1)$$

但し，

$$C(u, v) = \begin{cases} \frac{1}{2} & (u = v = 0) \\ \frac{1}{\sqrt{2}} & (u = 0, v \neq 0 \quad \text{or} \quad u \neq 0, v = 0) \\ 1 & (\text{otherwise}) \end{cases} \quad (2.2)$$

変換の結果として，ブロック内の画素と同じ数である 64 個の変換係数 $F(u, v)$ が得られる． $F(0, 0)$ は，直流成分係数と呼ばれ，残りは交流成分係数と呼ばれる．また， u, v の値が大きいほど，高周波成分を表している．

量子化 DCT は直交変換の一種であり，DCT 単独では情報の損失がないかわり，データの圧縮を行うことができない．そこで，DCT 変換により生じた変換係数の偏りを利用して圧縮を行う．これを量子化という．

具体的な量子化の方法は，DCT 係数を $F(u, v)$ ，量子化後の係数を $Z(i, j)$ とした場合，式 (2.3) で表される．

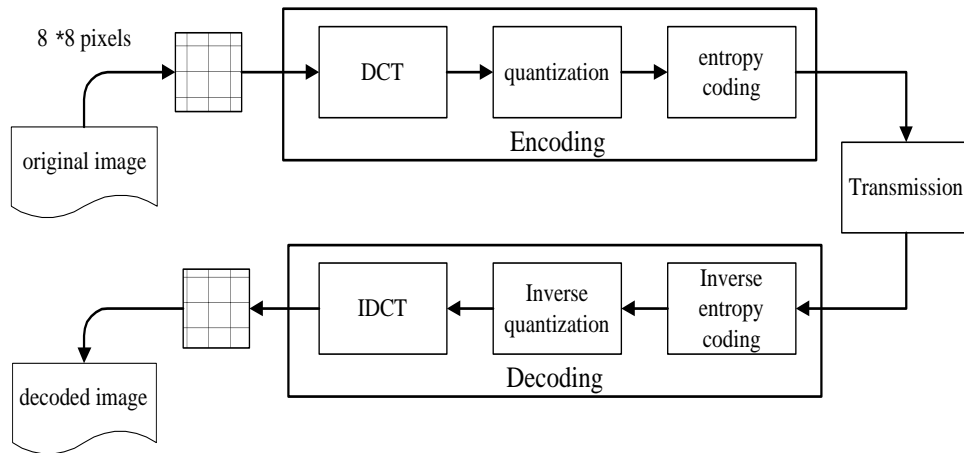


図 2-3: DCT を用いた静止画像符号化方式の基本構成

$$Z(u, v) = \text{round}\left(\frac{F(u, v)}{Q(u, v)}\right) \tag{2.3}$$

$\text{round}(x)$ は、 x の丸め操作を表す。また、 $Q(u, v)$ は、量子化配列（量子化テーブル）と呼ばれる 8×8 の行列の要素である。

JPEG の場合 [4]，表 2-1 に示される標準量子化テーブルが一般に用いられる。この量子化テーブルでは、少ない損失でも圧縮率が高くなるように、人間の視覚心理的な要因を考慮して決定されている。

表 2-1: 標準量子化テーブル

v/u	0	1	2	3	4	5	6	7
0	16	11	10	16	24	40	51	61
1	12	12	14	19	26	58	60	55
2	14	13	16	24	40	57	69	56
3	14	17	22	29	51	87	80	62
4	18	22	37	56	68	109	103	77
5	24	35	55	64	81	104	113	92
6	49	64	78	87	103	121	120	101
7	72	92	95	98	112	100	103	99

また、JPEG の場合、Quality パラメータにより、符号量と符号化後の品質を制御することができる。一般的に、式 2.4 を用いて、量子化テーブル $Q'(u, v)$ が新たに作成され、用いられる。

$$Q'(u, v) = \begin{cases} Q(u, v) \times \frac{(100 - \text{Quality})}{50} & (\text{Quality} \geq 50) \\ Q(u, v) \times \frac{50}{\text{Quality}} & (\text{Quality} < 50) \end{cases} \quad (2.4)$$

2.3.2 アニメーション画像と符号化歪み

自然画像の場合、滑らかな階調変化が多いため、量子化により発生する復号時の歪みは気にならない場合が多い。一方、アニメーション画像では、エッジを多く含む。エッジとは、輝度成分の急激な変化であり、均等色領域と線画の境界で発生する。そして、エッジの近辺で量子化によるノイズが発生する。

表 2-1 を見ると、右下の部分、すなわち高周波成分における量子化係数が大きくなる。高周波成分において人間の視覚特性からカットされる割合が多いことが分かり、これは、Quality パラメータが大きくなる程顕著である。

モスキートノイズ アニメーション画像を、DCT ベースの符号器にかけた場合、エッジを多く含んだブロックが存在することになる。これらブロックは、DCT 変換後、周波数成分が高域にまで広がる。その後、量子化を行うと、量子化係数が大きいため、DCT 係数の高周波部分ほとんどが 0 となり、圧縮される。この圧縮されたブロックを復号した場合、カットされた係数が時間領域においてノイズとなり発生する。この現象はモスキートノイズと呼ばれる。図 2-4 に、モスキートノイズ発生の様子を示す。

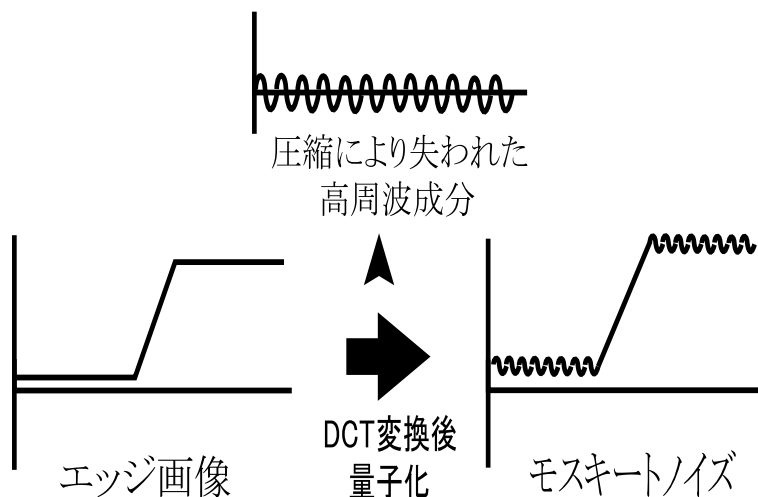


図 2-4: モスキートノイズ

図 2-5 に、実際のアニメーション画像において、モスキートノイズが発生した例を示す。

したがって、アニメーション画像は、特に高圧縮を考えた場合、DCT を基盤にする符号化方式に適さないと考えられる。そのため、本論文では、次節に述べるベクトル記述言語への変換を目的とする。



図 2-5: モスキートノイズによる品質の劣化

2.4 ベクトル記述言語による画像符号化方式

ベクトル記述言語は、座標値に基いたデータ系列をもとに、図形情報を格納する。線や領域といった画像中の構成要素が、その頂点や有意点を用いて記述される。復号の際には、表示の際には補間処理を行うことで、画像として表現する。これは、ベクタ-ラスタ変換と呼ばれる。

これまで、用途に応じてさまざまなベクトル記述言語が開発されている。主に、CAD (Computer Aided Design) や、3次元CG、地図データの表現、ドローソフトなどに用いられている。結局、ラスタとベクタの違いは画像情報としてのデータ構造の違いということである。

ベクトル記述言語の利点としては、画像サイズ(解像度)にとらわれない描画が可能である。頂点や有意点の間は、定義された直線や曲線を用いて数式により補間されるためである。ラスタ画像の場合、拡大や縮小を行う場合には、画素単位での補間法が必要となり、歪みの原因となる。

また、符号量を抑えることが可能である。復号に必要な情報は、画像中の構成要素の、通過点や曲率を表す点のみであり、全ての点ではない。そのため、符号量は画像の複雑さに比例するものの、一般に、元のデータ量に比べ、極めて小さくすることができる。

さらに、再編集が可能であることが挙げられる。ベクトル記述言語で記述された図形は、相対的な位置関係で表現されるため、変形や移動、拡大・縮小といった編集作業が容易である。

さて、本論文が符号化の対象とするのは、アニメーション映像である。2003年現在、ベクトル記述言語による、もっとも普及しているアニメーション符号化技術のひとつとして、Macromedia社のFlashが挙げられる。これに加えて、W3C(World Wide Web Consortium)勧告であり、XML(eXtensible Markup Language)による記述を行うSVG(Scalable Vector Graphics)が挙げられる。次章で述べる符号化は、出力としてSVGによる実装を目指す。そのため、両者についてその概要を説明する。

2.4.1 Macromedia Flash

急激なインターネットの普及に伴い、Web site を利用した企業広報、商品プレゼンテーションが盛んに行われている。Macromedia 社が開発した Flash は、オーサリングツールが優れていること、符号化データの圧縮効率が優れていることから急速に広まった Flash の主な機能として次の5点が挙げられる。

- グラフィック機能 Flash のグラフィック機能では、ベクターデータを採用している。ベクターデータのメリットとして、描画を図形単位で計算して書き出すため、対象のグラフィックを拡大、縮小した際も画像のばらつきが発生しないことである。

- アニメーション機能

Flash は2D アニメーション作成環境として優れた特性を持っている。その特徴としてはデータ構造がベクター、しかも PostScript ベースである。

- インターフェイス機能

Web での Flash の使用法としてインターフェイスとしての使用法が挙げられる。スクリプティングと変数により、更新の容易さも付け加わっている。

- パブリッシュ機能

ワンキーでブラウザの表示を確認できる、多数のファイルを一度に書き出せる、適切なスクリプトを含めた HTML を書き出すことができる。

- マルチユース機能

マルチユース機能とは Flash で作成したアニメーションやグラフィックをビデオ、プリントアウト、スクリーンセーバーなどに展開できる機能のことである。

Flash を用いて作成される動画データは、“軽量”、“ストリーミング対応”という特徴を持つ。これを用いて、インターネットや携帯端末上でのコンテンツ配信事業が開始されている。画像サイズは、データがベクター形式なので拡大・縮小しても画質劣化は少ない。また、インタラクティブ機能を持つため、ザッピング機能(マルチストーリー対応)が提供されている。

2.4.2 SVG (Scalable Vector Graphics)

Web の標準化団体である W3C は2003年1月14日に、SVG 1.1 を勧告として公開した。SVG のデータ形式は XML をベースとしており、内部データは全てテキストデータで記述されている。なお、SVG データを 算術符号化を用いた可逆圧縮アルゴリズムで符号化した、SVGZ という形式もある。SVG では直線と曲線といったベクトル情報で画像を表現するため、以下のような多くの特徴を持っている。

- ファイルサイズが小さい

GIF (Graphics Interchange Format) や JPEG のような画像フォーマットでは、画像を構成する全てのピクセル情報を持たなければならないが、SVG では少ないデータサイズでも、ドット数の巨大な画像データを表現できる。

- ディスプレイの解像度に依存しない

高解像度の CRT (Cathode Ray Tube) やプリンタ、携帯情報端末のいずれのデバイスでも、それぞれに最適なクオリティで表示できる。また、クオリティを落とすことなく、自由に画像を拡大・縮小できる。

- テキスト形式のため、編集が簡単

SVG では画像データの編集は、内部テキストを書き換えるだけで良い。またテキスト形式である、検索エンジンにキーワードを登録させたり、Perl や JavaScript などのプログラミング言語を使って、サーバやクライアント上で動的に画像データを生成するのも容易にできる。

- インタラクティブで高度な表現が可能

拡張性の高い XML を活かして、アニメーションや特殊効果、ユーザーの動作に反応するインタラクティブな動作など、様々な高度表現が可能。

SVG は業界に幅広く支持されており、各種ブラウザ向けプラグインや、専用の閲覧ツールが実装され、公開されている。また、携帯端末上で SVG の閲覧を可能にするための Mobile SVG Profiles の公開も行われている。Mobile SVG には、携帯電話向けの SVG Tiny と、ハンドヘルド PC など携帯端末向けの SVG Basic がある。前者は、BitFlash、CSIRO、KDDI、Nokia、ZOOMON らが、後者では、BitFlash、CSIRO、Intesis らが実装を行っている。

2.5 むすび

本章では、対象となるアニメーションの画像の特徴についてまとめた。アニメーションのような人工の画像では、自然画像と異なり、高い周波数成分が含まれるため、JPEG や MPEG に代表される変換符号化が適さない点を指摘した。そして、人工画像に適した符号化方式として、Flash、SVG というベクトル記述言語について紹介した。

第3章

ベクトル記述言語変換のための関数近似手法

3.1 はじめに

本章では、本符号化方式の要素技術である、ベクトル変換について述べる。ベクトル変換とは、ラスタデータである画素情報から、線や面といった構造を持つ情報、すなわちベクトルデータへ変換することである。

これには、画像から線や面といった情報を抽出することと、抽出した構造を、データ量削減のため任意の関数を用いて表現することが必要となる。前者は、必要な情報を選別するための、特徴量抽出に関する議論であり、次章の符号化に関連して述べる。本章では、後者の、点列補間や近似に関する数学的手法について述べる。

はじめに、数学的に曲線を表現するためのパラメトリック曲線について述べる。次に、ベクトル変換には、補間とよばれる技術が用いられるため、代表的な補間方法について述べる。そして、提案する符号化に用いられる、ベクトル変換手法について説明する。

3.2 パラメトリック曲線

ベクトル記述言語を用いて2次元図形を構成するには、パラメトリック曲線 (parametric curve) が用いられる。パラメトリックとは、パラメータ空間から対象空間への関数として定義されている事を意味する。2次元の場合、 $Q(t) = (x, y)$ の形で表現される。

代表的なパラメトリック曲線として、エルミート曲線、スプライン曲線、ベジェ曲線などが挙げられる。スプライン曲線には、与えられた制御点を全て通過する自然スプライン関数と、そうでないB-スプライン関数がある。ここでは、B-スプライン関数について説明する。

3.2.1 エルミート曲線

エルミート曲線は、補間関数としての性質を有するエルミート多項式 H_i^n より生成される曲線である。3次 ($n = 3$) のエルミート多項式 $H_i^3(u)$ は、次の性質を有する。性質を有する。

$$H_0^3(0) = 1, H_0^3(1) = 0, H_1^3(0) = 0, H_1^3(1) = 1,$$

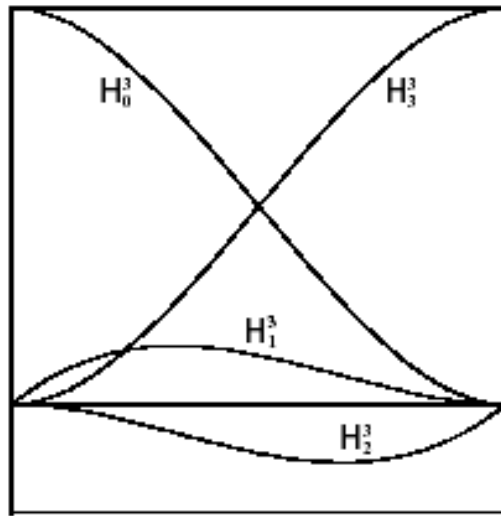


図 3-1: 3 次エルミート補間子

$$\begin{aligned} \frac{dH_0^3(t)}{dt} &= \frac{dH_3^3(t)}{dt} = 0 \text{ (for } t = 0, 1), \\ H_1^3(t) &= H_2^3(t) = 0 \text{ (for } t = 0, 1), \\ \frac{dH_1^3(0)}{dt} &= 1, \frac{dH_1^3(1)}{dt} = 0, \frac{dH_2^3(0)}{dt} = 0, \frac{dH_2^3(1)}{dt} = 1 \end{aligned}$$

ただし, $H_0^3(t) + H_3^3(t) = 1$ となる.

3 次エルミート曲線 $E^3(t)$ は, 2 端点の座標値 P_0, P_1 と, その端点での傾斜 (1 次微分) ベクトル V_0, V_1 を与えることにより, 以下のように生成される.

$$E^3(t) = P_0 H_0^3(t) + V_0 H_1^3(t) + V_1 H_2^3(t) + P_1 H_3^3(t)$$

3.2.2 ベジエ曲線

n 次のバーンスタイン多項式 $B_t^n(t) (i = 0, 1, \dots, n)$ は,

$$B_t^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

で与えられる. このとき, 制御点 P_i と, その基底関数を用いて生成した曲線 $\alpha(t) = \sum_{i=0}^n P_i B_i^n(t)$ は, ベジエ曲線とよばれる. n 次のベジエ曲線 $p^n(u)$ の座標値は, deCasteljau のアルゴリズムを用いて, 再帰式により計算することも可能である.

$$p^r(t) = (1-t)p_i^{r-1}(t) + tp_{i+1}^{r-1}(t) \quad (p_i^0 = P_i, r = 1, 2, \dots, n, i = 0, 1, \dots, n-r)$$

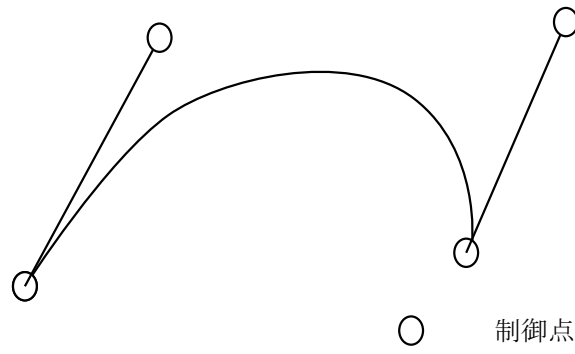


図 3-2: エルミート曲線

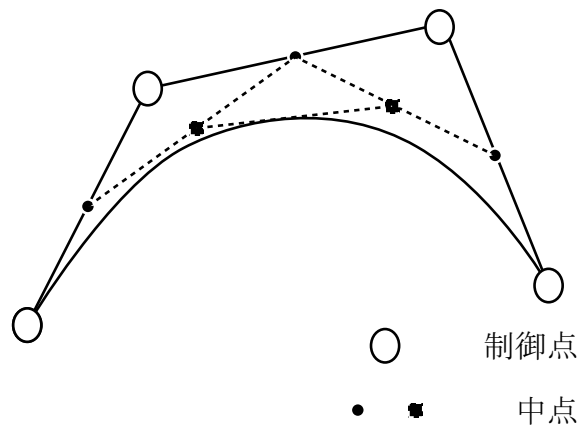


図 3-3: ベジエ曲線の例

n 次のベジェ曲線は、初期入力 of 点を線形に補間して線分を生成し、次にその線分を線形に補間して2次の曲線を生成し、... という手続きを繰り返して生成される。

図3-3にベジェ曲線を示す。エルミートと共通するところは、線分 AB を接線として発生し、点 B, C に引っ張られながら、線分 CD を接線として終わる。制御点間の midpoint を直線で結び、さらにその直線の midpoint を直線で結び、というように再帰的に行う。その結果、最終的に得られるのがベジェ曲線となる。四つの定義点によるポリゴンの形状から、描かれる曲線の形状を想像しやすいという特徴を持っている。

3.2.3 B-スプライン曲線

B-スプライン基底関数を $B_{i,K}(t)$ と (i は制御点の番号, K は階数 (次数+1))。B-スプライン基底関数は、切断べき関数の高階差分商、と定義される。すなわち、 $(K-1)$ 次の B-スプライン基底関数は、パラメータ t の関数として以下の数式で定義される。

$$B_{i,k}(t) = (-1)^k (t_{i+K} - t_i) \frac{(t - t_i)_+^{K-1}}{P_i(t_i)} + \frac{(t - t_{i+1})_+^{K-1}}{P_{i+1}(t_{i+1})} + \dots + \frac{(t - t_{i+K})_+^{K-1}}{P_{i+K}(t_{i+K})}$$

但し、

$$P_j(t_j) = (t_j - t_i)(t_j - t_{i+1}) \cdots (t_j - t_{j-1})(t_j - t_{j+1}) \cdots (t_j - t_{i+K})$$

これを計算すると、 $(t - t_i)_+^{K-1}$ のような切断べき関数がいっているのが容易ではない。

de Boor Cox の算法

一般に、スプライン基底関数は、以下に示すように漸化式で計算できる。この方法は de Boor Cox の算法と呼ばれている。

$K = 1$ (0 次) の場合は、

$$B_{i,1}(t) = \begin{cases} 1 & (t_i \leq t < t_{i+1}) \\ 0 & (t \leq t_i, t \geq t_{i+1}) \end{cases} \quad (3.1)$$

$K \geq 2$ の場合は、

$$B_{i,K}(t) = \frac{t - t_i}{t_{i+K-1} - t_i} B_{i,K-1}(t) + \frac{t_{i+K} - t}{t_{i+K} - t_{i+1}} B_{i+1,K-1}(t) \quad (3.2)$$

となり、 $B_{i,K}(t)$ は、次数の1つ低い $B_{i,K-1}(t)$ と $B_{i+1,K-1}(t)$ により計算できる。この式を用いれば、0 次の B-スプライン基底関数から順に、 $(K-1)$ 次までの基底関数を求めることが可能になる。

重要なことは、1 つの B-スプライン基底関数 $B_{i,K}$ を計算するには、 $(K+1)$ 個の節点が必要ということである。B-スプライン関数 $B_{i,K}$ は $t_i \leq t \leq t_{i+K}$ 以外では0となる関数 refb-basis に示すような関数である。

図3.2.3に、B-スプライン曲線を示す。B-スプライン曲線では、制御点を通過しない。定義点の移動に対して比較的緩慢に変化するので、滑らかな曲線を細かく制御したい場合に適する。

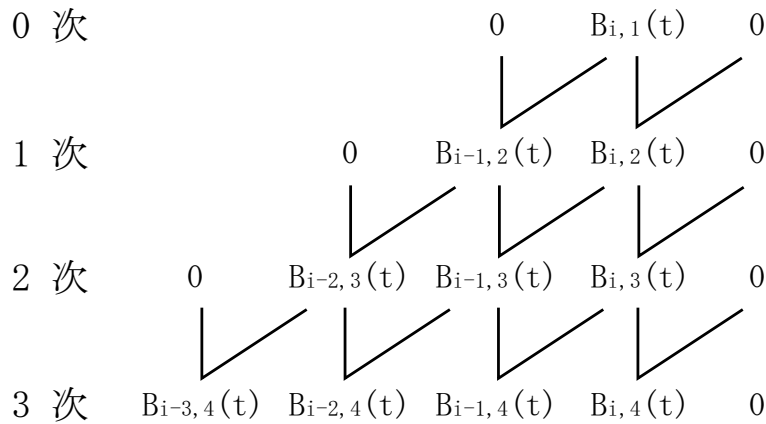


図 3-4: B-スプライン基底関数の計算ピラミッド

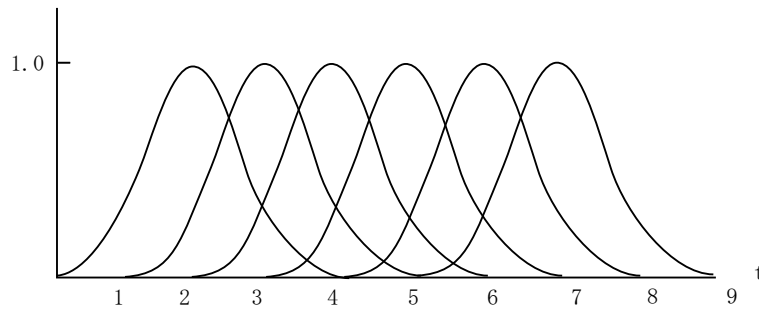


図 3-5: B-スプライン基底関数 (3次)

3.2.4 NURBS 曲線

NURBS というのは (Non-Uniform Rational B-Spline) の略で、日本語でいえば「非一様有理 B スプライン」となる。NURBS 曲線は、B-スプライン基底関数の線形結合で表現できる。そのため、NURBS 曲線を計算で求めるには B スプライン基底関数を計算する必要がある。

いま、 $Q_{0 \leq i \leq n-1}$ を制御点とすると、パラメータ t の関数として、

$$P(t) = \sum_{i=0}^{n-1} B_{i,K}(t) Q_i$$

となる。各制御点に対して B-スプライン基底関数を計算し、その値を係数として制御点を足し合わせる。

(K-1) 次の B スプライン基底関数の計算には、(K+1) 個の節点が必要である。 $B_{i,K}$ が節点 $t_i, t_{i+1}, \dots, t_{i+K}$ で構成されているとすると、合計 $(n + K)$ 個の節点 t_0, \dots, t_{n+K-1} があれば良いことになる。この様に 1 つの制御点に対する B スプライン基底関数を節点の組を

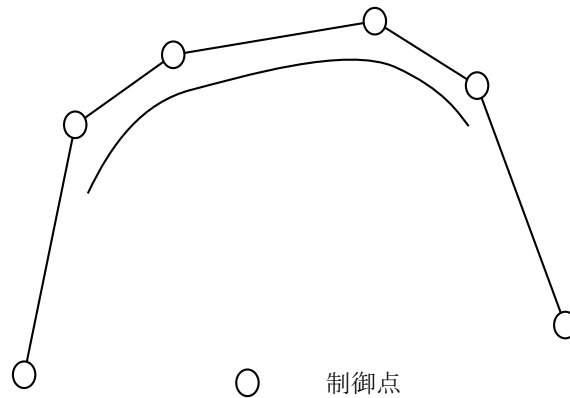


図 3-6: 3 次の B-スプライン曲線の例

1 つずつシフトして構成すると、節点でなめらかに接続することが保証される。

また、この式 (5) にはパラメータ t の有効範囲がある。曲線は t が $t_{K-1} \leq t \leq t_n$ の範囲にある時のみ存在する。例えば、 $K=4$ つまり 3 次の場合で考えてみる。制御点が 4 個、つまり節点の数が $4+4=8$ 個ある場合、図 3 でみると $t_i \leq t \leq t_{i+1}$ の区間の 4 つの B スプライン基底関数が有効となっている範囲のみで曲線が存在する。1 つの B スプライン基底関数は制御点 1 つに対応しているので、図 3 より、3 次の場合は制御点 4 つに対して 1 つの曲線区間が表されるということになる。図 4 に 3 次の 2 区間の曲線の例を示す。1 つめの区間は Q_0, Q_1, Q_2, Q_3 によって表され、2 つめの区間は Q_1, Q_2, Q_3, Q_4 によって表される。一般には $(K-1)$ 次の NURBS 曲線の場合 K 個の制御点で 1 つの曲線区間が表される。

3.3 数学的補間

与えられた点列間を、関数を用いて推定する手法を補間法と呼ぶ。パラメトリック曲線による代表的な補間法について説明する。

3.3.1 ラグランジュ法

ある関数 $f(x)$ について、 x_1, x_2, \dots, x_n の時の値が既知であり、それぞれ y_1, y_2, \dots, y_n とする。すなわち、

$$f(x_i) = y_i (i = 1, 2, \dots, n)$$

とする。このとき、

$$p(x_i) = y_i (i = 1, 2, \dots, n) \quad (3.3)$$

を満たす関数 $p(x)$ を, $f(x)$ の近似式と考える. そこで, 式 (3.3) を満たす $p(x)$ として,

$$p(x_i) = \sum_{i=1}^n y_i p_i(x) \quad (3.4)$$

とおく. このとき, 各 k

$$p(x_i) = \sum_{i=1}^n y_i p_i(x_k) \quad (3.5)$$

に対して,

$$p_i(x_k) = 0 (i \neq k) \quad (3.6)$$

$$= 1 (i = k) \quad (3.7)$$

と定義すれば, 式 (3.4) は, 式 (3.3) を満たす. そこで, 式 (3.6) を満たす $p_i(x)$ を考える. これは,

$$p_i(x) = \prod_{j=1, j \neq i}^n \frac{x - x_j}{x_i - x_j} = \frac{(x - x_1)(x - x_2) \cdots (x - x_n)}{(x_i - x_1)(x_i - x_2) \cdots (x_i - x_n)} \quad (3.8)$$

の形で与えられる. すると,

$$k \neq i : p_i(x_k) = \frac{(x - x_1)(x - x_2) \cdots (x - x_n)}{(x_i - x_1)(x_i - x_2) \cdots (x_i - x_n)} = 0$$

$$k = i : p_i(x_k) = \frac{(x - x_1)(x - x_2) \cdots (x - x_n)}{(x_i - x_1)(x_i - x_2) \cdots (x_i - x_n)} = 1$$

となり, 式 (3.6) を満たす. したがって, $p(x)$ は,

$$p(x) = \sum_{i=1}^n (y_i \prod_{j=1, j \neq i}^n \frac{x - x_j}{x_i - x_j}) \quad (3.9)$$

とあらわされる. 式 (3.9) は, ラグランジュの補間公式と呼ばれる.

3.3.2 スプライン法

与えられた全ての点を通る次数の高い式を, ラグランジュ法で求めるのは困難である. 次数が高いほど, 式が複雑になるためである. よって, 全ての点を用いるのではなく, 曲線を区間に分割することを考える. 区間ごとに式を求めることで, 全ての点を通る式を構成する. これが区分的多項式であり, その代表的なものにスプライン法がある. 以下に, スプライン法について説明する.

いま, 区間 $a \sim b$ において,

$$a = x_0, x_1, x_2, \cdots, x_{i-1}, x_i, x_{i+1} = b$$

となる点列があるとする。

ラグランジュによる方法は、区間 $a \sim b$ の全ての点を通る単一の補間多項式であった。ここで、全区間 $a \sim b$ を、

$$[x_0, x_1][x_1, x_2] \cdots [x_{i-1}, x_i][x_i, x_{i+1}] \cdots [x_{n-a}, x_n]$$

とする小区間として考える。そして、小区間毎に次数の低い補間多項式を考える。この際、各区間を補間し、かつ、その微分が接続点で滑らかになる様になるための関数を考える。この関数をスプライン関数という。

2 次スプライン補間

2 次のスプライン関数は、

$$S_i(x) = a_i + b_i x + c_i x^2$$

と表すことができる。

この 2 次のスプライン関数には、以下の 2 条件が必要である。

- $S_i(x_i) = y_i$ であること。ここで、 y_i は、ある x_i における関数値であり、この関数値は未知の関数 $f()$ から求められる値である。なお、 $S_i(x_i) = y_i$ の式は、未知の関数 $f()$ で与えられた点 $x_0, x_1, x_2, \dots, x_{n-a}$ において誤差が 0 であることを意味している。すなわち、未知の関数で与えられた点と、スプライン関数で与えられる点は一致すると考えることができる。
- $S'_i(x_i) = S'_{i+1}(x_i)$ であること。すなわち、1 次微分が各小区間のつなぎ目で連続であることが必要である。

3 次スプライン補間

スプライン補間には、一般に 3 次のスプライン関数が使われることが多い。3 次スプライン補間では、2 次スプライン補間に比べて条件が増える。

- $S_i(x_i) = y_i$ であること。これは 2 次の場合と同じである。
- 与えられた点 $x_i (i = 1, 2, \dots, n)$ では、

$$S'_i(x_i) = S'_{i+1}(x_i)$$

かつ、

$$S''_i(x_i) = S''_{i+1}(x_i)$$

を満たすこと。すなわち、1 次微分および 2 次微分において、各点での多項式の微分係数が一致することが必要となる。

- 両端の境界条件として，

$$f''(x_1) = f''(x_{n+1}) = 0$$

または，

$$f'(x_1) = y'_1 \text{ かつ } f'(x_{n+1}) = y'_{n+1}$$

が与えられること．

以上の3点が，3次における条件となる．

さて，区間 $[x_i, x_{i+1}]$ における3次の多項式を $S_i(x)$ とすると，

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \quad (3.10)$$

となる．スプライン補間を行うには，式(3.10)における各係数 a_i, b_i, c_i, d_i を求める．これを求めるには，以下の計算手順に従えばよい．

- $a_i = y_i$
- 連立方程式の準備をする．以下について，全区間内の与えられたデータを用いて求める．

$$h_i = x_i - x_{i-1}$$

$$\Delta y_i = \frac{y_i - y_{i-1}}{h_i}, \Delta y_{i+1} = \frac{y_{i+1} - y_i}{h_{i+1}}$$

$$3(\Delta y_i h_{i+1} + \Delta y_{i+1} h_i)$$

- 連立方程式を行列式に変換し，解く．これにはガウスの消去法等を用いる．これにより，全ての y'_i が求まる．
- $b_i = y'_i$
- $c_i = (3 \Delta y_{i+1} - 2y'_i - y'_{i+1})$
- $d_i = \frac{y'_i + y'_{i+1} - 2\Delta y_{i+1}}{h_{i+1}}$

3.4 ベクトル変換アルゴリズム

提案する符号化方式に用いるベクトル変換アルゴリズムは，3次のスプライン関数による補間を用いている．一般に，3次スプライン関数による近似は，元の全ての点を通るため，制御点の決定が容易となる．しかしこれには，局所的な制御が不可能となる問題がある．具体的には，曲率の高い部分における制御点の決定が難しくなる．曲率の低い，あるいは一様な区分曲線の近似であれば，時間的なサンプリングにより入力点列を削減し，その間を補間することで精度の高い近似が得られる．しかし，極めて鋭角な曲線や，直線と曲線が入り混じるような入力については，単純な補間法では，近似精度の劣化や，冗長な有意点の出現といった問題がおこる．そのため，アニメーション映像に用いられる，キャラクタの輪郭を表すような線分に対しては，効率のよいベクトル変換アルゴリズムが必要となる．

3.4.1 Dynamic Programming による最適近似

宮澤ら [15] は、補間処理に DP (Dynamic Programming) 処理を取り入れることで、任意の曲線に対する最適なスプライン補間アルゴリズムを提案した。DP は、取り得る全ての事象を評価関数に入力し、最適な解を再帰的に導出する手法である。

評価式定義 DP を輪郭線近似に適用するには、輪郭線上の全ての点を近似のための有意味点として扱い、その評価関数に、輪郭線近似の際の近似誤差算出式を用いる。近似誤差算出式としては、最小距離誤差を用いている。これは、原曲線上の点から最も近い位置にある近似曲線上の点までの距離の総和である。原曲線上の点の特徴（曲率値、鋭角点など）に基づいて重み付けが可能であり、高度な制御が可能のためである。算出式、算出法を式 (3-11) 及び図 3-7 に示す。

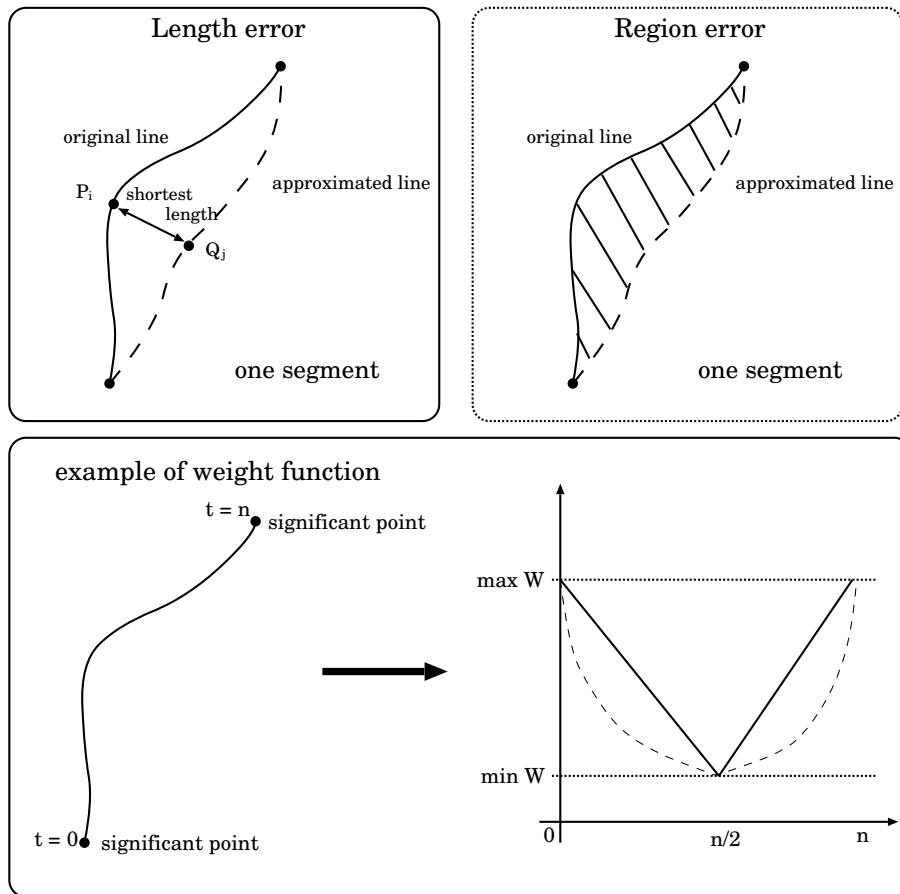


図 3-7: 誤差定義及び重み関数の例

$$E = \frac{1}{l} \sum_{i=1} \omega_i |P_i - Q_j|^2 \tag{3-11}$$

ここで、 E は近似誤差値、 l は輪郭線のセグメントにおける点数、 ω_i は原曲線上の点 i の重み関数であり、 P_i が同点の座標を示している。また、 Q_j は点 i から近似曲線上の最小距

離を与える点 j の座標を示している。近似誤差値はセグメントの距離で正規化されているため、誤差関数を用いるための誤差閾値は、あらゆる大きさの閉曲線に対し共通に用いることができる。

処理概要 DP 処理開始に当たって、第一に、曲率の大きい点から数点を有意点として選択する。

第二に、連続する有意点間をセグメント化する。なお、有意点間の距離が閾値以上である場合、セグメントを分割する処理を加える。これは、誤差関数の出力はセグメントの長さで正規化されることにより、セグメント長が大きい場合、細かな形状を正確に近似できなくなる現象が発生するためである。

第三に、セグメント間の誤差が閾値以上であるならば、セグメント間の全ての点を有意点候補とし、各々近似曲線を得る。ここで、閾値以下であるならば、そのセグメントにおける DP は終了する。

第四に、各々得た近似曲線について最小距離誤差が最小となる点を有意点として最終決定する。第三、第四の処理を誤差が閾値未満となるまで再帰的に行うことで、スプライン関数を用いた最適な近似が可能となる。

しかし、この段階では、曲率の高いセグメントに過剰な有意点が残るという問題がある。そこで、有意点抽出のために用いた DP を、抽出した有意点を削除する方向に用いる。これには、抽出した全有意点の中から1つずつ仮に削除し、各々近似誤差を順に算出する。

そして、その中で最も小さい誤差を与える有意点を実際に削除する。この処理を、削除目標の点数削減率に達するまで繰り返す。これにより、セグメント単位での処理ではなく、入力全体に対する近似誤差を測ることが可能になる。

そのため、輪郭線の中で最も冗長な有意点から順番に削除することが可能である。なお本アルゴリズムの繰り返し演算数は $\sum_{k=0}^{N_r-1} (N_s - k)$ と表される。ただし、 N_r は削除する有意点数、 N_s は本アルゴリズム実行前の有意点数である。

3.5 SVG によるベクトル変換データの出力

上記に述べたベクトル変換アルゴリズムの出力は、2章で紹介した SVG フォーマットを用いて記述される。

形状の記述には、Path エlement が用いられる。W3C のサイト * より、必要な部分を引用し、説明する。

Path Element Path は、輪郭を描いたり、線を描いたりすることに用いられる。また、クリッピングパスのために用いることもできる。

Path は、moveto(新しい通過点を設定する)、lineto(直線を描く)、curveto(曲線を描く)、arc(円弧を描く)、closepath(始点まで直線を引き、現在の PATH を閉じる) で定義される命

*<http://www.w3.org/TR/SVG11>

令を持つ。W3C から、関連するいくつかの Path のコマンドを紹介する。また、以下に、DOM インターフェースを引用する。

```

<!ENTITY % SVG.path.extra.content "" >
<!ENTITY % SVG.path.element "INCLUDE" >
<![%SVG.path.element;[
<!ENTITY % SVG.path.content
    "( ( %SVG.Description.class; )*, ( %SVG.Animation.class;
        %SVG.path.extra.content; )*)"
>
<!ELEMENT %SVG.path.qname; %SVG.path.content; >
<!-- end of SVG.path.element -->]]>
<!ENTITY % SVG.path.attlist "INCLUDE" >
<![%SVG.path.attlist;[
<!ATTLIST %SVG.path.qname;
    %SVG.Core.attrib;
    %SVG.Conditional.attrib;
    %SVG.Style.attrib;
    %SVG.Paint.attrib;
    %SVG.Color.attrib;
    %SVG.Opacity.attrib;
    %SVG.Graphics.attrib;
    %SVG.Marker.attrib;
    %SVG.Clip.attrib;
    %SVG.Mask.attrib;
    %SVG.Filter.attrib;
    %SVG.GraphicalEvents.attrib;
    %SVG.Cursor.attrib;
    %SVG.External.attrib;
    d %PathData.datatype; #REQUIRED
    pathLength %Number.datatype; #IMPLIED
    transform %TransformList.datatype; #IMPLIED
>

```

moveto コマンド “moveto” コマンド (M または m) は、現在点の更新をする。これは、ペンで新たな点から書こうとすることに相当する。パスのセグメントは、必ず “moveto” コマンドから始まる必要がある。

なお、以降に紹介するコマンドは、大文字によるコマンドは絶対座標、小文字によるコマンドは現在点からの相対座標を引数に持つ。

表 3-1 に, "moveto" コマンドの文法を示す.

Command	Name	Parameters	Description
M(absolute) or m(relative)	moveto	(x y)+	Start a new sub-path at the given (x,y) coordinate. M (uppercase) indicates that absolute coordinates will follow; m (lowercase) indicates that relative coordinates will follow. If a relative moveto (m) appears as the first element of the path, then it is treated as a pair of absolute coordinates. If a moveto is followed by multiple pairs of coordinates, the subsequent pairs are treated as implicit lineto commands.

表 3-1: moveto コマンド

closepath コマンド "closepath" コマンド (Z または z) を用いることで, パスを終端させることができる. これにより, パスの始点まで直線を引き, パスを閉じたものにする.

"closepath" コマンドが, "moveto" コマンドの直後にあった場合には, "moveto" コマンドによる移動先が, 次のパスの始点として扱われる. 他のコマンドの直後にあった場合は, 現在のパスと同じ点から開始するものとする. "closepath" コマンドでパスを終端した場合と, "lineto" コマンドでパスを閉じた場合の振る舞いは異なり, "lineto" コマンドの場合は, パスの始点と終点は結合したものとして扱われない. 表 3-2 に, "closepath" コマンドの文法を示す.

Command	Name	Parameters	Description
Z(absolute) or z(relative)	closepath	(none)	Close the current subpath by drawing a straight line from the current point to current subpath's initial point.

表 3-2: closepath コマンド

lineto コマンド "lineto" コマンドは現在点から, 指定した点までの直線を描く. 目標を 2 次元で指定する L (または l) と, 水平, 垂直の 1 次元で指定する H, V (または h,v) の 3 通りがある. 表 3-3 に, "lineto" コマンドの文法を示す.

quadratic Bezier コマンド 用いるのは, path 要素の T 命令である. これは, 略式のベジェ関数を描画する命令であるが, 第一制御点を現在の点と同一のものとすることが可能であるため, 上記アルゴリズムの出力を, 変換することなしに利用することが可能になる.

Command	Name	Parameters	Description
L(absolute) or l(relative)	lineto	(x y)+	Draw a line from the current point to the given (x,y) coordinate which becomes the new current point. L (uppercase) indicates that absolute coordinates will follow; l (lowercase) indicates that relative coordinates will follow. A number of coordinates pairs may be specified to draw a polyline. At the end of the command, the new current point is set to the final set of coordinates provided. bpath's initial point.
H(absolute) or h(relative)	horizontal lineto	x+	Draws a horizontal line from the current point (cpx, cpy) to (x, cpy). H (uppercase) indicates that absolute coordinates will follow; h (lowercase) indicates that relative coordinates will follow. Multiple x values can be provided (although usually this doesn't make sense). At the end of the command, the new current point becomes (x, cpy) for the final value of x.
V(absolute) or v(relative)	vertical lineto	y+	Draws a vertical line from the current point (cpx, cpy) to (cpx, y). V (uppercase) indicates that absolute coordinates will follow; v (lowercase) indicates that relative coordinates will follow. Multiple y values can be provided (although usually this doesn't make sense). At the end of the command, the new current point becomes (cpx, y) for the final value of y.

表 3-3: lineto コマンド

Command	Name	Parameters	Description
Q(absolute) or q(relative)	quadratic Bezier curve	(x1 y1 x y)+	Draws a quadratic Bezier curve from the current point to (x,y) using (x1,y1) as the control point. Q (uppercase) indicates that absolute coordinates will follow; q (lowercase) indicates that relative coordinates will follow. Multiple sets of coordinates may be specified to draw a polybezier. At the end of the command, the new current point becomes the final (x,y) coordinate pair used in the polybezier.
T(absolute) or t(relative)	Shorthand/smooth quadratic Bezier curve	(x y)+	Draws a quadratic Bezier curve from the current point to (x,y). The control point is assumed to be the reflection of the control point on the previous command relative to the current point. (If there is no previous command or if the previous command was not a Q, q, T or t, assume the control point is coincident with the current point.) T (uppercase) indicates that absolute coordinates will follow; t (lowercase) indicates that relative coordinates will follow. At the end of the command, the new current point becomes the final (x,y) coordinate pair used in the polybezier.

表 3-4: quadratic Bezier コマンド

最後に、SVG 形式での出力の例を挙げる。

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
  "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="352" height="240" viewBox="0 0 352 240"
  xmlns="http://www.w3.org/2000/svg" version="1.1">
<path d=" M87 50 T95 50 M95 50 T101 52 M101 52 T102 52 M102 52 T105 51 M105 51 T115
(中略)
  " fill="none" stroke="black" stroke-width="1"/>\\
</svg>
```

なお、このままでは冗長な表現を含むため、SVG に可逆圧縮を用いた SVGZ 形式を用いる必要がある。

3.6 むすび

本章では、ラスタ-ベクトル変換について述べた。ベクトル変換技術の基礎となる、パラメトリック曲線およびそれを用いた補間方法について説明した。また、提案する符号化に用いられる、ベクトル変換手法について説明した。

そして、符号化データを出力するための SVG フォーマットの詳細と、その出力例について示した。

第4章

ベクトル変換に基づくアニメーション静止画像 符号化方式

4.1 はじめに

本章では、アニメーション静止画像に適した符号化方式として、ベクトル変換に基づく符号化方式を提案する。これまでの先行文献とその問題点、提案手法、実験結果、課題について述べる。

4.2 形状情報を用いたアニメーション画像の符号化

形状情報に注目したアニメーション画像の符号化として、先行研究 [12][13][14] がある。これらでは、はじめにアニメーション画像より、線画像と均等色領域が抽出される。抽出された線要素に対しては、関数近似が行われ、有意点が抽出される。以後、これをベクトル変換と呼ぶ。次に、均等色領域に対しては、輪郭形状に対してベクトル変換を行い、別にその領域を代表する色情報が保存される。これにより、任意の領域について、輪郭情報と色情報のみで復元することが可能になる。さらに、線要素と均等色領域線として抽出されなかった部分については背景領域として扱われる。すなわち、背景領域 = 元画像 - (線画 + 均等色領域) として扱われる。背景領域については、DCT を用いて符号化される。背景領域には、DCT に適さないエッジとなる部分が含まれないためである。

以上をまとめると、先行研究では、図 4-1 に示す方式に従って符号化が行われる。

4.2.1 アニメーション画像からの形状情報抽出

形状をベクトル変換するために、アニメーション画像から、線画や均等色領域を抽出するためのアルゴリズムが必要となる。

均等色領域の抽出 アニメーション画像から、均等色領域の抽出をするには、領域成長法アルゴリズム [27] を用いる。これは、均等色領域が、輝度、色差信号ともにほぼ等しい点の集合で表現されているためである。

領域成長法アルゴリズムでは、次の手順に従って、均等色領域の検出を行う。

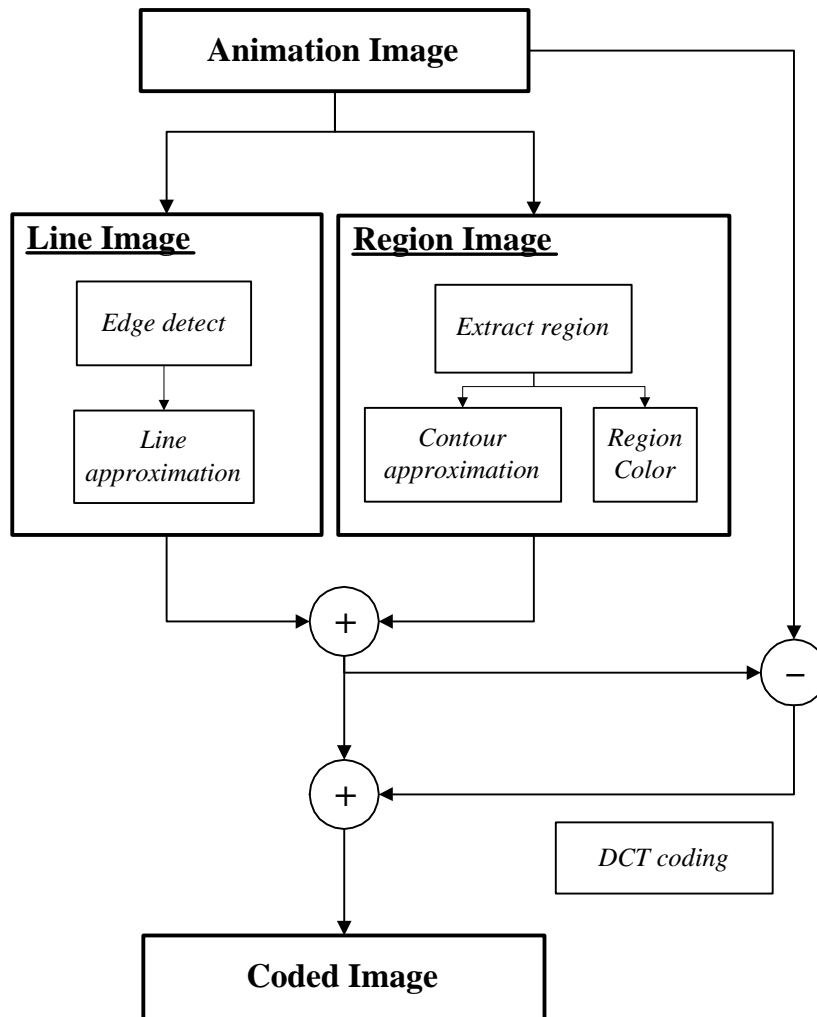


図 4-1: 先行研究におけるアニメーション画像符号化

1. 種となる一点から始まり，その近傍の点を閾値処理で判断していき，ほぼ同じ色情報をもつならば統合する．
2. 統合した際，色情報は領域内の全画素の平均値に更新する．
3. 領域に隣接する画素の近傍についても，1と同様の閾値判断をし，領域を広げていく．
4. 1～3を繰り返す．
5. 領域の成長が止まれば，まだ領域に含まれていない点を種とする．

以上により，画像中からほぼ等しい色を持つ点の集合が，領域として検出される．説明図を，図 4-2 に示す．

線画の抽出 原画像に微分オペレータを施し，領域の境界線及び線画を抽出する．微分オペレータは，原画像内のエッジ部分を検出するが，アニメーション画像内で，エッジとして検

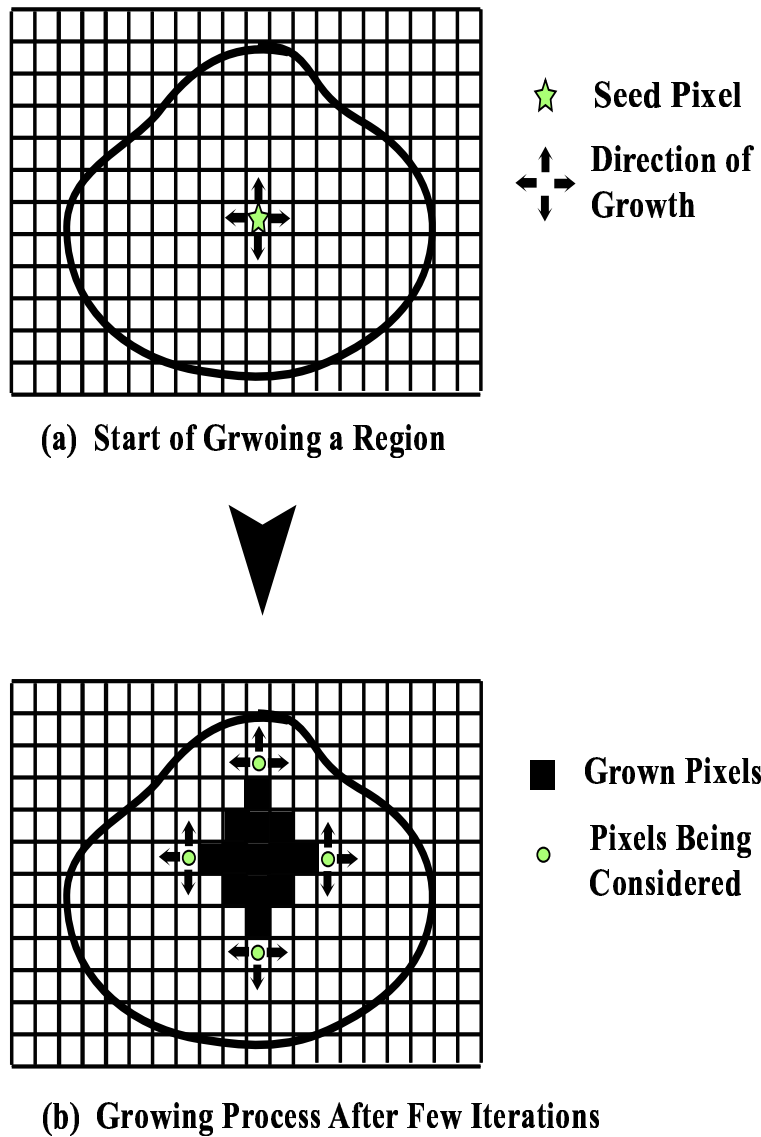


図 4-2: 領域成長法

出されるのは、主に線画部分と境界部分である。一般に、線画は数画素の幅を持って描かれる。すると、線画の両端でエッジが検出され、かつ、その両端において微分値の符号が逆となる。従って、ある範囲内において、微分値の符号が逆なエッジが検出されたならば、その間の画素を線画として抽出する。単一のエッジしか検出されなければ境界線として抽出する。

ラプラシアンオペレータを用いて、エッジを検出したものを図 4-3 に示し、そこから線画を抽出したものを図 4-4 に示す。

4.2.2 抽出した形状情報のベクトル変換

アニメーション画像より抽出された線画と均等色領域の輪郭線は、直線とスプライン関数を用いて近似される。関数近似を行うことで、線分より有意点が抽出される。



図 4-3: エッジの検出



図 4-4: 線画の検出

文献 [12] においては、均等色領域の輪郭線をスプライン関数を用いて近似を行うことが検討されている。輪郭線上のある点から始め、数画素の範囲でスライディングさせながら、画素間の差をベクトル化する。輪郭線を一周させ、それらのベクトルが大きく変化する点を曲率点として抽出する。曲率点間において、適当な間隔で通過点を保存する。曲率点と通過点を有意点とし、有意点間をスプライン関数で補間することで元の曲線の近似曲線を得ることができる。第3章における本稿の実験では、抽出した均等色領域の輪郭を近似するために、アルゴリズムを用いた。

文献 [14] では、スプライン関数と直線を両方用いた近似方式についての検討がなされている。線画を補間する際、短い線や直線要素に対してスプライン曲線のみで補間する場合の効率の悪さを指摘し、近似対象の線分について直線・曲線の区別を行うことが検討されている。具体的な近似アルゴリズムとして、Ramer 法 [21] を改良した、Modified Ramer 法が提案されており、第3章における本稿の実験では、抽出した線画像の近似法に、その成果を用いている。

4.3 従来方式の問題点

従来手法において、アニメーション画像に直接領域成長法を用いることで、均等色領域として抽出するとしたキャラクタ部分以外でも、背景領域から均等色領域を誤抽出することがある。誤検出の例を、図4-5に示す。誤抽出される主な原因は、背景領域のごく一部だけが均等色で描かれているからである。図4-5では、背景画像における扉や石垣の一部が良く似た色の集合で描かれている。このような部分では、領域成長法のアルゴリズムを用いる以上、必ず検出が行われる。



図 4-5: 背景領域の誤検出例

背景領域から誤抽出することの問題点として、

1. 誤抽出された領域は一般に複雑な形状をしていること
2. DCTで符号化するべき領域の定義が曖昧で、方式が統一されていないこと

が挙げられる。

誤抽出された領域は、ここで検出を目的としているキャラクタの輪郭線と比べて、意味を持った形状でないうえに、複雑である。すると、スプライン曲線等を用いて近似することが難しい。

後者に関しては、先のような誤抽出の問題が起こる以上、背景領域に対してDCTによる符号化を用いるという手法は、特に複雑な背景領域を多く含む画像において、その使い分けが正確に行われない場合が多い。

従って従来手法では、特に背景領域を多く含むアニメーション画像に対して符号化を行うことが難しい。

4.4 二層構造によるアニメーション画像の符号化の提案

本節では、アニメーション画像にべた塗りの領域が多く存在することに着目し、画像全体を、均等色領域の集合で表現することを提案する。これにより、従来手法における背景領域の誤抽出という問題を解決する。

アニメーション画像における背景領域は、自然画に近い性質をもつため、領域成長法では均等色領域として抽出することが不可能である。提案手法では、前置処理として平滑化を行うことで、背景領域におけるグラデーションやテクスチャ情報など、均等色領域として抽出できない情報を消去する。この後に領域成長法を用いることで、べた塗りの部分ではない背景領域からも、近似された均等色領域を抽出することが可能になる。すなわち、アニメーション画像中で、べた塗りで描かれている部分からも、背景領域として描かれている部分からも、均等色領域を抽出することが可能になる。

本方式の利点は、画像全体を統一した方式で扱えることである。前章では、従来手法において DCT で扱う背景領域と関数近似を用いて符号化する領域の区別が曖昧であったことを指摘したが、提案手法においては、対象のアニメーション画像全体から線情報と均等色領域を抽出し、有意点のみで符号化する。

提案方式での符号化アルゴリズムを図 4-6 に示す。アニメーション画像を、線画と均等色領域のみで表現する基本レイヤと、平滑化により失われた情報を補うための付加的なレイヤの、2層で符号化するものとする。

第1層で符号化される情報は、画像中の形状の有意点情報と、各均等色領域内の色情報のみなので、符号化される情報量は少ない。

しかし、平滑化操作を用いることにより、背景領域を多く含むような画像では符号化品質が損なわれる。そこで、4章で述べる方法により差分情報を加えることで、均等色近似による誤差を補う。

先行研究として Region based coding という自然画像に対する符号化が行われている [16][18]。自然画像を均等色領域の集合として表現した場合、等しい、或いは近い色情報を持つ点が集合していることは少ない。また、抽出された一つ一つの領域は小さく、複雑な形状である。そのため、輪郭部の符号化に関しては、チェーンコードが用いられている。チェーンコードは、形状を正確に符号化するという点で有効な手法である。しかし、輪郭を構成する全ての点を符号化するため、関数により近似した場合に比べ、データ量が増える。

アニメーション画像の場合、べた塗りで描かれた部分が多く含まれるため、抽出された個々の均等色領域は、自然画像のときに比べ大きいものが多い。輪郭形状を表す直線や曲線を関数で近似することにより、さらに効率良く符号化を行うことができる。

4.4.1 画像平滑化手法による均等色領域抽出手法

平滑化手法と、均等色領域への分割手法には様々な方法が存在するが、本稿では、Connected Filter と領域成長法を用いた。

Connected Operator による平滑化フィルタ

画像の平滑化操作のために、従来のローパスフィルタに代表される線形フィルタを用いると、エッジの部分まで平滑化される（図 4-7）アニメーション画像に多く含まれるエッジが平滑化されると、品質が著しく損なわれることになる。そのため、エッジ情報は保存された

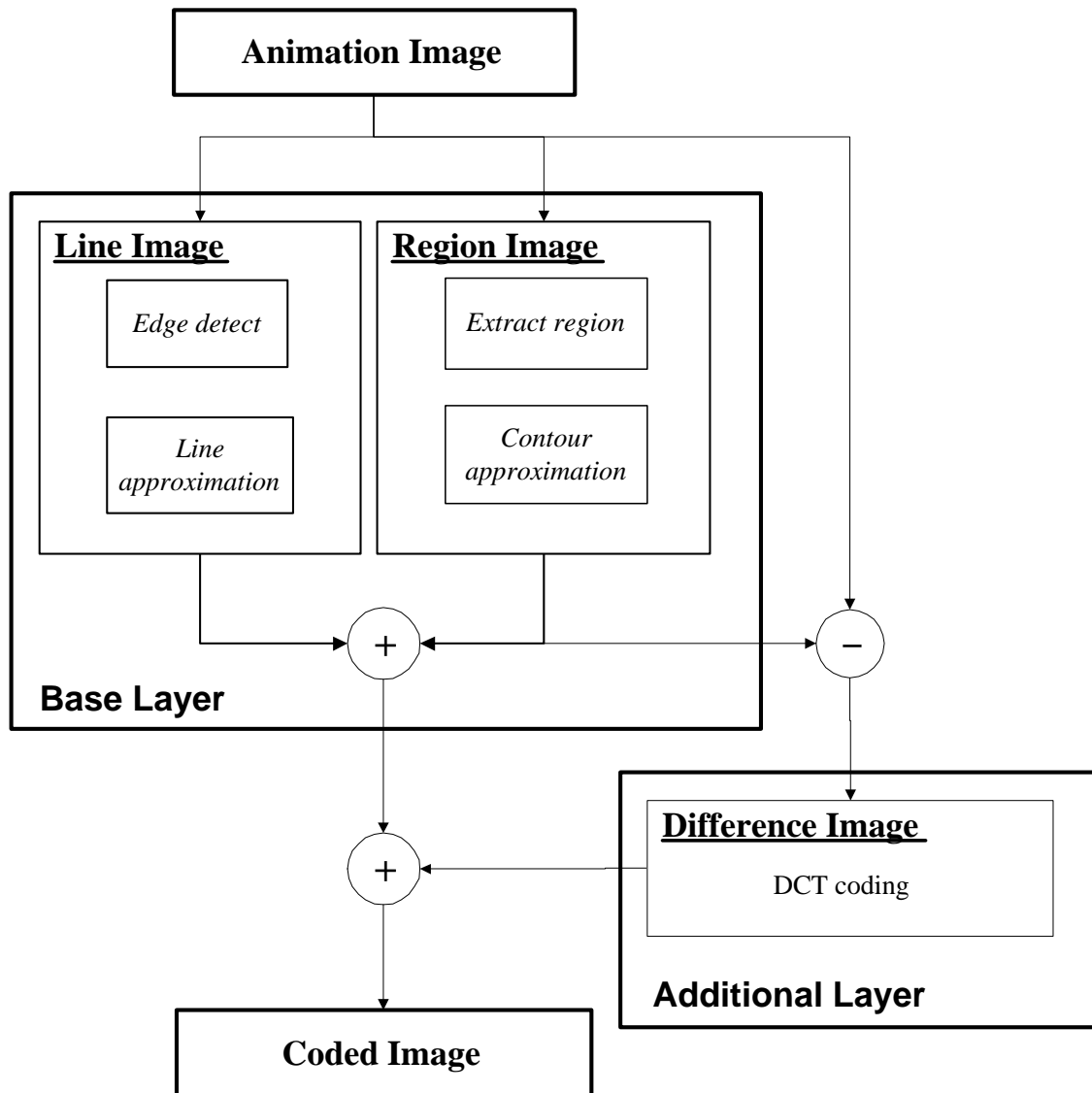


図 4-6: 提案方式による符号化

まま、領域内のみ平滑化されるような操作が必要となる。

以上の要件を満たす平滑化フィルタの一つとして、Connected Filter が挙げられる。

Connected Filter による平滑化操作における、基本概念から説明する。

図 4-8 における元画像 A は、グレースケール画像とする。 A に含まれるいくつかの領域 A_n は、等しい輝度値の集合した領域であり、Flat-zone と呼ぶ。Flat-zone は後述する方法により、輝度値の連結性を考慮して作成される。次に、 A_4 は A_2 の上に位置する領域なので、 A_2 と A_4 を統合した画像は、 A_2 と A_4 が別々に存在する画像より、“より粗い”と定義することができる。そして、画像全体を “より粗い”ものになるように変換する。

但し、先の定義によると、 A_3 は A_1 に統合されることになるので、消去される。 A_4 はサイズが小さいので消えるのが妥当だが、 A_3 が消えるのはあまり妥当ではない。そこで、領域を

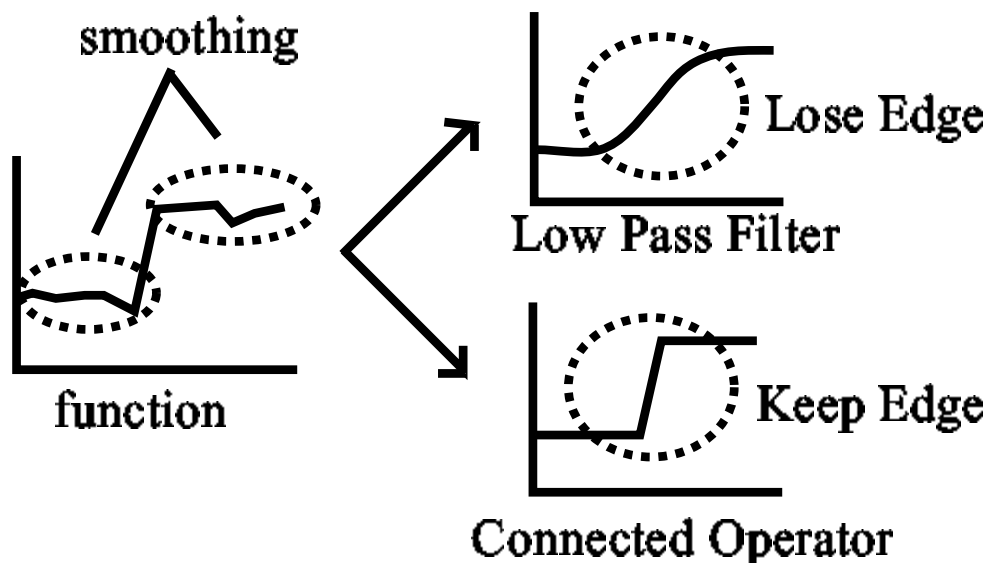


図 4-7: フィルタによる平滑化

消すための判断基準を設け、その判断に適合するものだけを、統合して消していく。この判断基準として様々なものが挙げられるが、本稿では、モルフォロジ処理を応用した Connected Open and Close を用いた。

Flat-zone それぞれの画素について近傍を調べ、同じ輝度値を持つ画素を結合した領域のことを Flat-zone とよぶ。仮に同じ輝度値の画素が 4, ないし 8 近傍にない場合は、その点は、一点でも Flat-zone とする。図 4-9, 図 4-10 に、各 Flat-zone を色分けして表示させたものを示す。

Connected Operator Flat-zone を一般の数式に適用するための演算子を、Connected Operator とよぶ。これは、連結性の判断に用いられる。

Connected Operator Ψ の定義

- 隣り合う画素 p, q が同じ値を持つならば、 $f(p) = f(q)$ と定義する。
- その時、 $\Psi(f(p)) = \Psi(f(q))$ とする。

Connected Filter Connected Operator を用いたフィルタを、Connected Filter とよぶ。Connected Filter は Flat-zone の統合処理により平滑化を行うため、元画像におけるエッジが完全に保たれるという特徴がある。平滑化の判断基準として、Flat-zone 同士の輝度差や大きさといったものが挙げられるが、ここで用いるのは、あまりパラメータに依存しない、モルフォロジ処理を用いたものである。

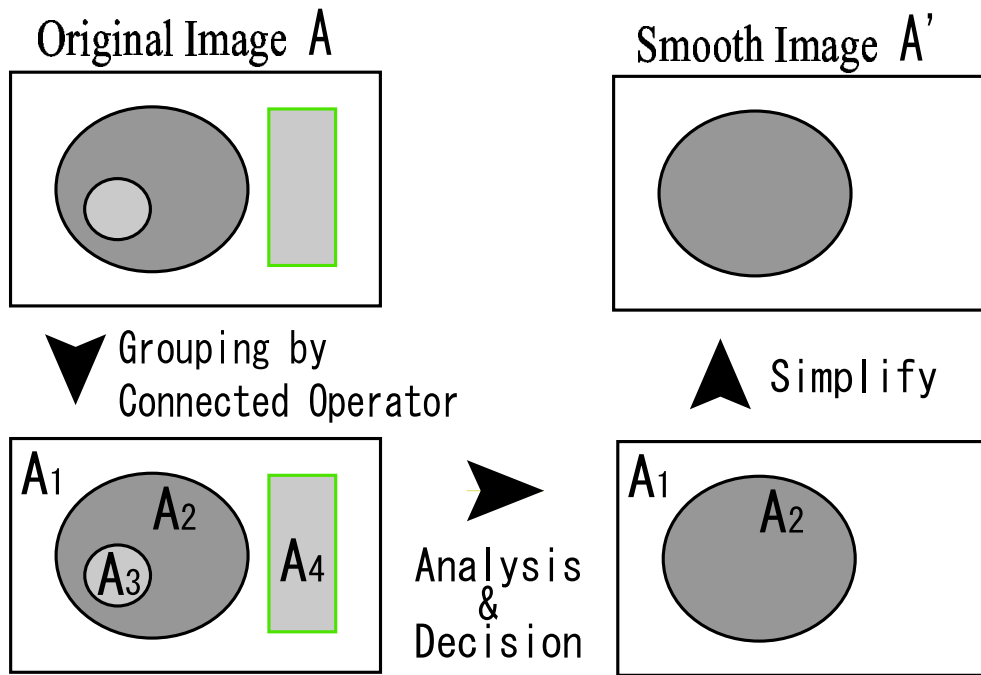


図 4-8: Connected Operator による領域の結合

モルフォロジ処理

モルフォロジ演算 [25] は、画像の特徴抽出やノイズ除去など様々な画像処理に用いられる基本的処理である。

モルフォロジの基本演算 モルフォロジ処理についての、基本的な演算を述べる。

2つの集合 F, B の Minkowski 和を $F \oplus B$ とする。これは集合の平行移動と和によって式 4.1 のように定義される。

$$F \oplus B = f + b : f \in F, b \in B \tag{4.1}$$

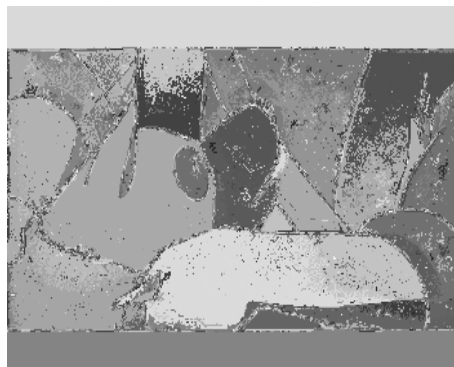


図 4-9: Flat-zone の例 (背景領域無しの場合)

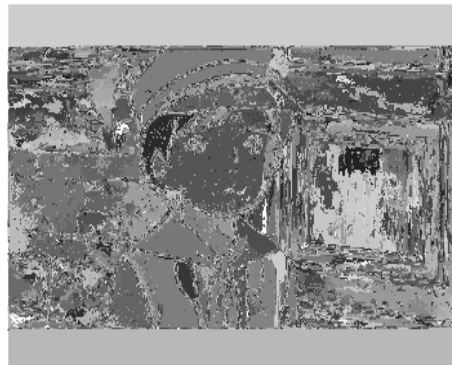


図 4-10: Flat-zone の例 (背景領域を含む場合)

2つの集合 F, B の Minkowski 差を $F \ominus B$ とする .

$$F \ominus B = f - b : f \in F, b \in B \quad (4.2)$$

モルフォロジの基本処理 モルフォロジにおける最も基本的な処理として、次の4つが挙げられる .

dilation 2次元のグレースケール画像 $f(X)$ の場合 [24]、構造要素 B とすると、次のように定義ができる . 但し、 $f(X)$ は点 X における輝度値とする .

$$(f \oplus b)(X) = \max f(X + Z) + b(Z) : Z \in B \quad (4.3)$$

erosion dilation の時と同様に、2次元グレースケール画像に対しては、

$$(f \ominus b)(X) = \min f(X + Z) - b(Z) : Z \in B \quad (4.4)$$

opening erosion の処理の後にミンコフスキー和を求める . この処理により画像中に穴があく .

$$f \circ B = (f \ominus b) \oplus b \quad (4.5)$$

closing opening とは逆に、dilation をしてからミンコフスキー差を求める . これにより穴を塞ぐ等の処理が行われる .

$$f \bullet b = (f \oplus b) \ominus b \quad (4.6)$$

Connected Open and Close

モルフォロジ処理において、Open and Close 処理は、孤立点の除去に用いられるが、それだけでは輝度値の連結性を考慮していないため、グレースケール画像に対する平滑化に適さない .

ここで、Flat-zone に着目し、孤立する Flat-zone を消去することを考える。孤立する Flat-zone は、ある領域内におけるノイズといえる。そのため、これを消去した場合、領域内において平滑化がなされることになる。

モルフォロジ処理を用いて以上の操作を行うためには、消去の対象が孤立した Flat-zone であるかどうかを判断する構成要素が必要になる。そこで、式 (4.7)、(4.8) のように 4 ないし 8 近傍の連結性を持つものを用いる。中心点から 4 近傍の結合性を調べる場合、式 (4.7) が判断基準になる。 r は半径であり、 $r=1$ の場合十字型になる。中心点から 8 近傍の場合、式 (4.7) が判断基準になり、その形状は正方形になる。

$$B_4(r) = (x_1, x_2) \in Z^2 : |x_1| + |x_2| \leq r \quad (4.7)$$

$$B_8(r) = (x_1, x_2) \in Z^2 : \max(|x_1|, |x_2|) \leq r \quad (4.8)$$

以上の構成要素を用いて、Open and Close 処理を行う。この処理を、Connected Open and Close と呼ぶ。

平滑化の処理結果を図 4-11 に示す。図中、白線部における輝度値の変化をグラフにした。横軸は X 座標、縦軸が輝度値である。エッジを保ったまま平滑化が行われていることがわかる。

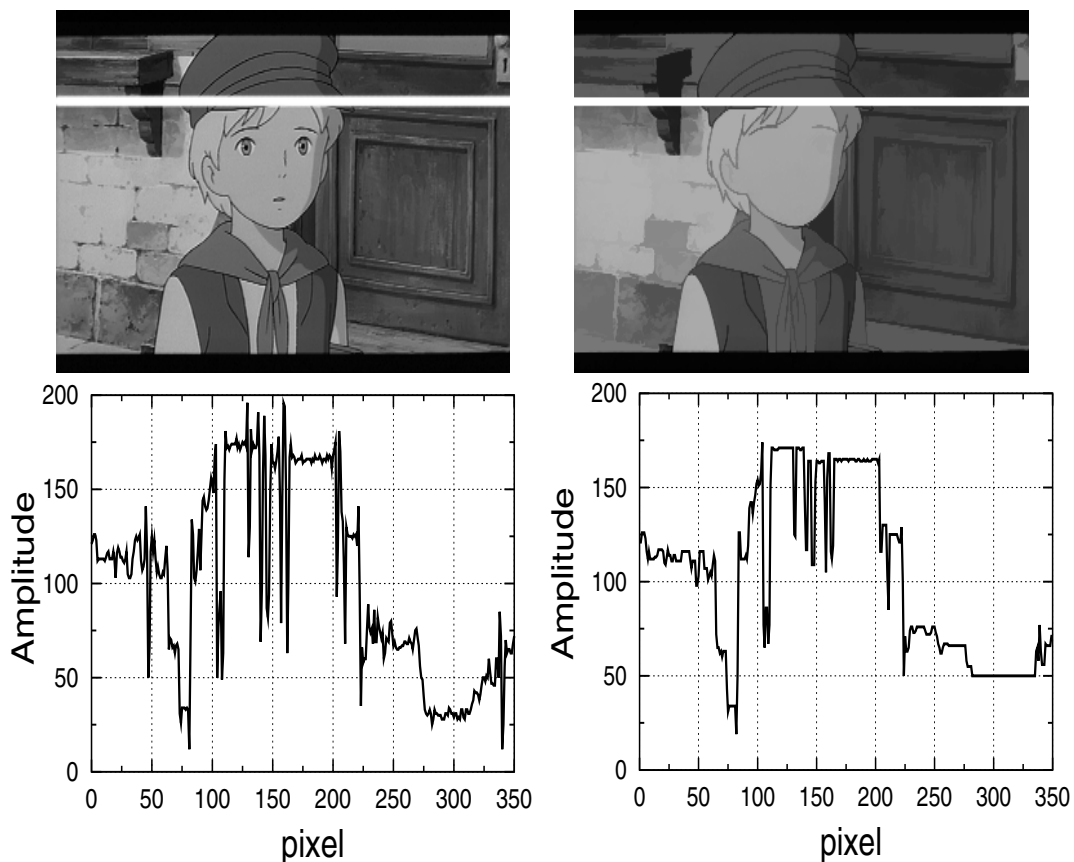


図 4-11: Connected Operator によるフィルタの適用

この結果に対し、領域成長法を用いることで、背景部も、ある程度の大きさの領域として抽出が可能になり、画像全体を均等色領域の集合として表現することができる。

4.4.2 結果と考察

実験概要

今回実験に用いた画像は、2種類のコンテンツから、1つは複雑な背景領域を含まないもの (dog1,anpan1)、もう1つは比較的複雑な背景領域を含むもの (dog1,anpan2) の、計4画像を選んだ。提案手法では、背景領域が簡単な場合には平滑化の影響が薄いことと、背景領域が複雑な場合でも均等色領域の抽出が可能であることを示すためである。

表 4-1: 実験条件

テスト画像	dog1,dog2,anpanman1,anpanman2
画像	4:2:0 format
	輝度信号 : 352[pixel] × 240[line]
	色差信号 : 352[pixel] × 240[line]
色空間	YCbCr

各画像について、元画像と、平滑化結果、平滑化結果に領域抽出法を用いた結果を示す。さらに、提案手法における第1層の符号化画像生成結果を示すために、2章で述べた手法による線画抽出結果と、抽出した線画に細線化処理を施した後に関数近似を行った結果についても示した。最後に、抽出した均等色領域の輪郭を近似した画像に線画を合成した結果を示した。

4.4.3 考察

Connected Filter による領域内のみ、エッジが失われない平滑化を行ったので、dog1 や anpanman1 といったほぼ均等色領域のみで描かれている画像においては、ほとんど影響を受けない (図 4-13, 図 4-25)。一方、dog2 や anpanman2 における複雑な背景領域においては、平滑化の結果が顕著に見られる (図 4-19 図 4-31)。

均等色領域の抽出に関しては、dog2 などで一部小さな領域において抽出できない部分が残っている。しかし、概ね他の画像に対しては、線画部分を除いた画像全体からの領域抽出に成功している。

線画と均等色領域の合成による基本レイヤに関しては、線画の抽出、近似精度が問題として挙げられる。

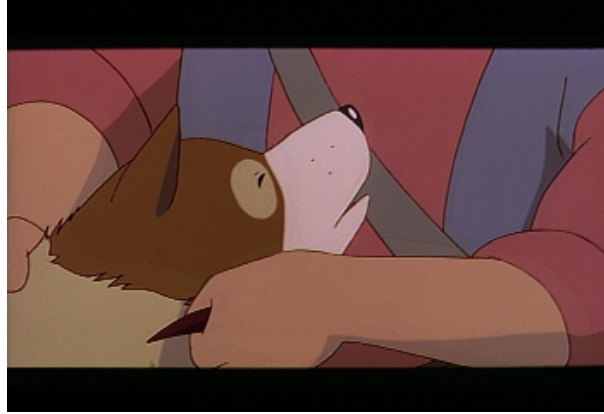


図 4-12: 元画像 (dog1)

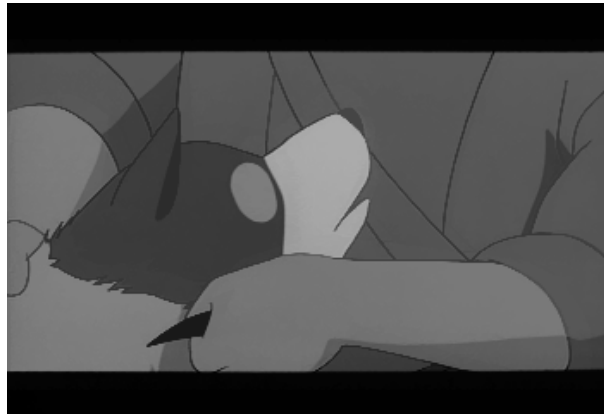


図 4-13: 輝度値への平滑化处理結果 (dog1)

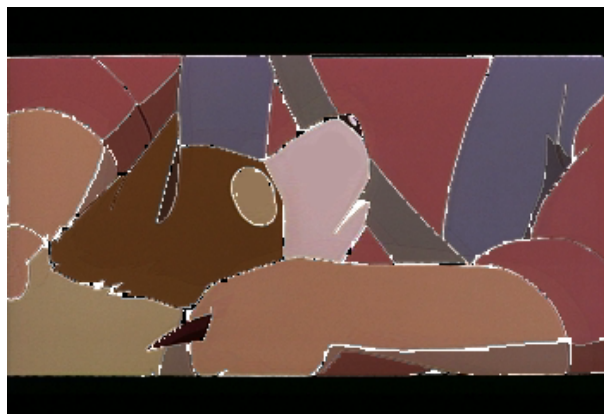


図 4-14: 均等色領域抽出結果 (dog1)

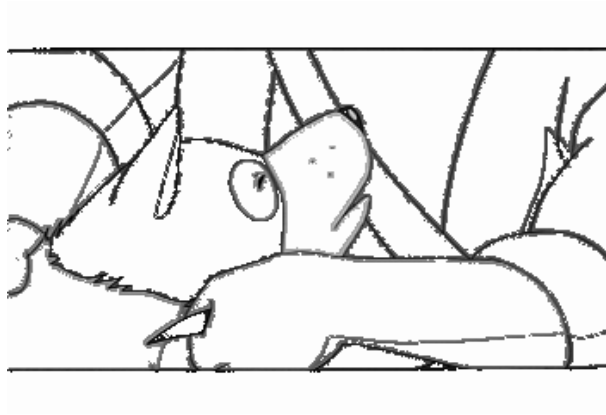


図 4-15: 線画の抽出 (dog1)

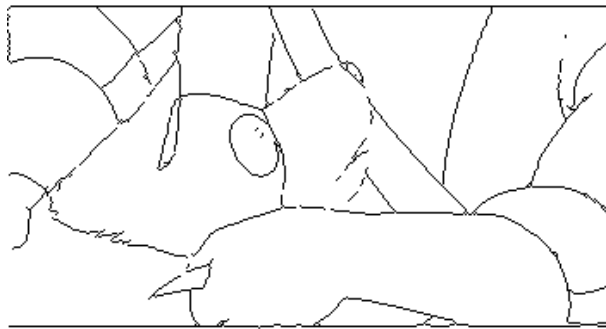


図 4-16: 細線化処理を施したもの (dog1)



図 4-17: 線画，均等色領域合成結果 (dog1)

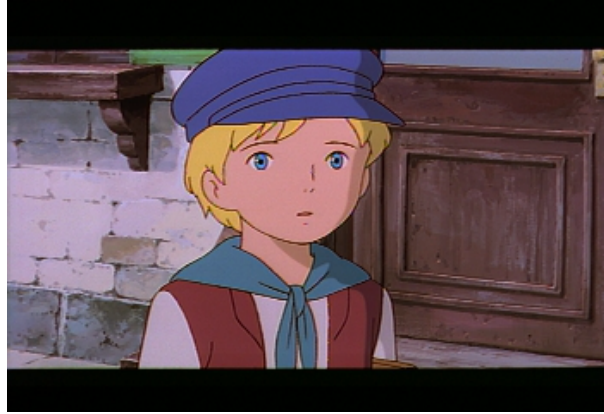


図 4-18: 元画像 (dog2)



図 4-19: 輝度値への平滑化処理結果 (dog2)



図 4-20: 均等色領域抽出結果 (dog2)



図 4-21: 線画の抽出 (dog2)



図 4-22: 細線化処理を施したもの (dog2)



図 4-23: 線画, 均等色領域合成結果 (dog2)



図 4-24: 元画像 (anpanman1)



図 4-25: 輝度値への平滑化処理結果 (anpanman1)



図 4-26: 均等色領域抽出結果 (anpanman1)



図 4-27: 線画の抽出 (anpanman1)

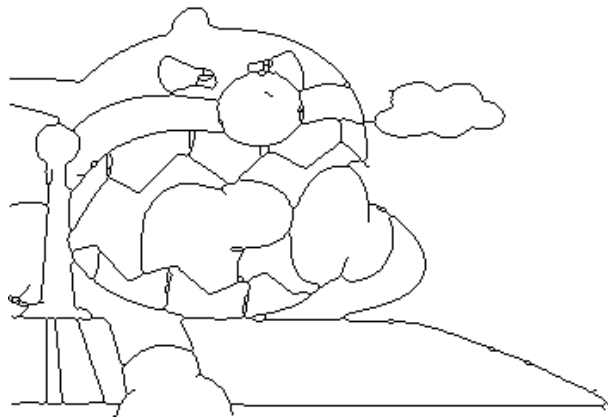


図 4-28: 細線化処理を施したもの (anpanman1)



図 4-29: 線画，均等色領域合成結果 (anpanman1)



図 4-30: 元画像 (anpanman2)



図 4-31: 輝度値への平滑化処理結果 (anpanman2)



図 4-32: 均等色領域抽出結果 (anpanman2)

4.5 むすび

本章では、ベクトル変換に基づくアニメーション静止画像の符号化方式を提案した。元画像を、ベクトルデータによる基本層と、テクスチャ情報の DCT 符号化による付加層の二層構造により符号化することを提案した。そのために、符号化の対象に平滑化操作を用いることで、画像全体からの線、および均等色領域の抽出が可能になることを示した。さらに、その際に失われたテクスチャ情報を補うため、差分情報をブロック単位で DCT 符号化することで、ユーザーの必要に応じた符号化品質を提供できることを示した。

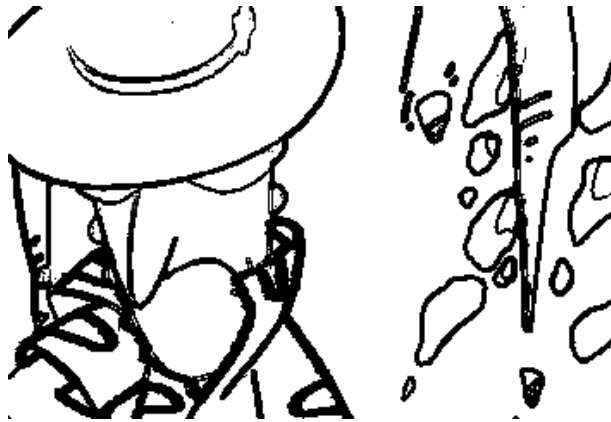


図 4-33: 線画の抽出 (anpanman2)



図 4-34: 細線化処理を施したもの (anpanman2)



図 4-35: 線画，均等色領域合成結果 (anpanman2)

第5章

アニメーション映像内のセル画領域同定法

5.1 はじめに

本章では，アニメーション映像より，セル画により表現されたオブジェクト領域を抽出し，さらに同一のセル画による領域を同定する手法について検討する．

一般に，セルアニメーションでは，彩色行程の煩雑さから，同一のセル画が何度も使用されることが多い．テレビ放映されるアニメーション番組において，実際に作成されるセル画の枚数は，すべて異なるコマとして計算した場合の10分の1程度と言われている．すなわち，同一のセル画を異なる配置で撮影することで，少ないセル画で複雑な動きを表現している．ここに着目して，映像中のどの部分が同一セル画によるものかを判定することができれば，符号化の際の情報量を減らすことが可能になる．

そこで，前章で述べた符号化方式を，動画像へと適用するための第一歩として，入力をセルアニメーションシーケンス，出力を同一セル画による領域とその対応位置関係とするシステムを提案する．これは，MPEG等では，動き予測の技術に相当する．MPEGではブロック単位で輝度・色差信号がもっとも近い部分を探すため，必ずしも動きベクトルと画面中のオブジェクトの移動は一致しない．対して，本提案システムでは任意領域の対応関係を探索し，移動先の画像中の座標，フレーム数を探索することで，オブジェクトの正確な移動の検出を目的とする．

従来，アニメーション映像に対する研究として，セル画の彩色を容易にするため，隣接フレーム間におけるキャラクタのパーツの対応付けが行われてきた [41], [42]．これらの問題点として，対応付けの際に色情報を使えないこと，2領域が変形する場合にも対応付けをする必要があることから，実際に得られる対応関係に精度が得られないということが挙げられる．そのため，形状情報のみに依存するマッチングアルゴリズムは，今回のシステムを実現する上であまり適当ではないといえる．

本研究で要求されるのは，輝度，色差信号および形状が同一の領域を，定められたシーケンス中から，任意の位置において検出を可能とする技術である．同一領域の検出に関する技術として，これまで多くの検討がなされてきた [51] [58] [60] [62] [20] [65] [66] [67]．これらは，大きく2種類に分けられる．ひとつは，テンプレートマッチングに代表される画素レベルでのマッチングを基盤におく手法である．もうひとつは，ハフ変換やフーリエ記述子な

ど、検索対象の領域の図形的特徴量を抽出することで、特徴点同士のマッチングを行うものである。

一般に、後者は検索対象へのノイズに耐性があり、輪郭形状の変化を伴わない拡大・縮小などの変形に強いという長所がある。このため、セル画に対して行われる特殊効果や、カメラからセル画の距離をずらすことで行われる擬似的な遠近法に適應できるといえる。しかし、主として形状のみに着目するため、セル画領域内部の変化に対応できないこと、輝度、色差信号ともに対応関係をとることが困難であることの2点が問題となる。一方、前者では、単純なテンプレートマッチングに改良を施すことで、拡大・縮小のみならずアフィン変換全般に対応可能なマッチング法の研究も行われている [67]。そのため、本研究では、後者のテンプレートマッチングを用いてオブジェクトの対応関係を探索するものとする。

ところで、領域の検索で重要となるのは、検索技術もさることながら、検索元となる領域を切り出す技術である。画像検索システム等では、ユーザからの入力を元として、アーカイブなりから検索が行われる。対して提案システムでは、検索部分への入力セル画で描かれた領域全てである。つまり、どのセル画領域が、シーケンス中に複数回登場しているかは不明のため、全てのセル画領域が検索元の候補となる。そのため、検索元の領域を切り出すために、アニメのシーケンスより、セル画で描かれたと思われる領域を抽出する技術が必要になる。

以上をまとめると、提案システムで必要となるのは、

1. 動きを検出し、それが画像中のどの領域であるかを抽出する技術
2. その領域が、別のフレームでどのように使われているかを検索する技術

の2つである。

ここでは前提として、動きを検出する際には、予めショットの単位へ分割されたシーケンスを対象とする。これは、シーン分割を行う自然画の手法が、精度良くアニメーション映像に適用できるという前提 [72] に基づいている。

5.2 アニメーション映像における動きの分類

セル画の部分は、オブジェクトの動きを表現するために、1コマずつ僅かに変化させて、フィルムに撮影される。この際、同一、あるいは数枚のセル画の繰り返しによりキャラクターの移動等を表現することが多い。これには、様々な手法が用いられるが、最も基本的なものとして、次の3つが挙げられる。

スライド キャラクターの平行移動のこと。隣合うフレームにおいて、セル画を上下左右にわずかにずらして撮影することで表現される。この時、拡大・縮小、回転を加えることもあり得る。

リピート 限られた複数枚のセル画を繰り返し用いることで、動きを表現する。瞬きのシーンなど。

ローリング セル画のスライド等で一定の範囲を揺れるように反復移動させる方法．馬車の車輪など．

符号化への応用を考えた際，情報量の縮体のために最も効果が高いと考えられるのは，スライドとリピートの動きである．そこで，これらの動きに特化した検出，判定アルゴリズムが必要となる．なお，ローリングに関しては，スライドとリピートの組み合わせと考えることができる．

5.3 差分画像を利用したオブジェクト抽出手法

先に述べたアニメ特有の表現手法から，あるセル画によるオブジェクトが，シーケンス内の別のフレームにおいても登場する可能性は高い．そこで，検索のためのテンプレートとして，オブジェクトを抽出する必要がある．これには，オブジェクトの外形を精度良く抽出できる手法が必要になる．

まず，提案アルゴリズムのフローチャートを，図 5-1 に示す．前提として，動きを検出する際には，予めショットの単位へ分割されたシーケンスを対象とする．これは，シーン分割を行う自然画の手法が，精度良くアニメーション映像に適用できるという [72] の文献に基づいている．

5.3.1 動領域の抽出

はじめに，フレーム間において，変化している領域を検出する．背景が固定されている場合，アニメーション映像において変化している部分は，必ずセル画領域である．ただし，映像表現上の技法として，背景をずらして撮影したり，カメラとの距離を変えて撮影することで，背景領域においても変化する場合はある．ここではそれを大局的な動きと呼び，それが存在する場合は後に考察する．

アニメーション映像における動領域を検出するには，フレーム間の差分画像を利用する．自然動画像を対象としたオブジェクト抽出には，動きベクトルやオプティカルフローを利用する手法 [56] が提案されている．しかし，アニメーション映像の場合，自然動画像と異なり，平坦な色遣いの領域が多く含まれる．そのため，正確なオブジェクト抽出に用いるのは難しい．また，予め背景画像を作成する手法も提案されているが [71] ，画面中に占めるオブジェクトの割合が小さく，かつ大きな移動がある場合にしか適さない．これではアニメの場合，先に挙げたスライド以外の動きの検出に用いることが難しい．

一方で，アニメーション映像の特徴として，同一の背景領域を連続するフレームで用いることが挙げられる．そのため，大局的な動きが存在しない場合，差分画像を生成することで完全に背景を消去することが可能である．

隣接フレーム間で，差分を絶対値として扱い画像を生成した場合，変化した部分の領域が検出される（図 5-2 参照）．この領域の輪郭線は，前後フレームにおける各領域の輪郭線を併合せたものとなる．ここでは，この輪郭線を粗い輪郭線と呼ぶ．各フレームにおけるセ

ル画領域の、正確な輪郭線を抽出するには、粗い輪郭線を用いる。その際、さらに元画像のエッジ情報と、輪ゴムのアルゴリズム [69] を用いる。

輪ゴムのアルゴリズム

輪ゴムのアルゴリズムは、視覚的な凹凸情報を抽出してオブジェクト形状を段階的に表現する手法である。境界線構造の解析方式として、オブジェクトの回転・サイズ・微少ノイズに高い耐性を有する。これには、まず概念的にオブジェクトに輪ゴムをかける。そして、輪ゴムとオブジェクトとの接触/非接触関係を、画素の反転処理により抽出する。この反転処理の操作を反復することで、オブジェクトの全境界線特徴の抽出を行うものである。

以下、輪ゴムのアルゴリズム (Iterated Rubberband Fitting Analysis) の概略を述べる。

p を境界画素とし、 S を対象オブジェクトの境界画素の集合とする。 l を1画素の1辺分の長さをもつ境界線分で単位境界線分 (Unit Boundary Line) とし、 B を境界線の集合とする。このとき、図5-3に示すように、オブジェクトを構成する各画素 p が有する単位境界線分の個数を $N(p)$ とすると、 $N(p) = i (0 \leq i \leq 4)$ で、 i は、0~4の整数となり、Type 0~4の4つに分類される。図5-4に、境界線情報の定義を示す。まず、オブジェクトの特徴抽出処理過程において、 n 本目に適用される輪ゴム閉曲線 $W(n)$ のうち、オブジェクトと接触している境界線分を $T(n)$ (Touched Line)、非接触の線分を空中線 $F(n)$ (Floating Line) とする。このとき、 $W(n), T(n), F(n)$ の関係は以下のように定義される。

$$F(n) \cap T(n) = 0 \quad (5.1)$$

$$F(n) \cup T(n) = W(n) \quad (5.2)$$

また、オブジェクトの境界線 B のうち、輪ゴム $W(n)$ の接線 $T(n)$ と接する境界線分を $W(n)$ に対する有意境界線分 $C(n)$ (Critical Boundary Line)、接線と接触しない境界線分を冗長境界線分 $R(n)$ (Redundant Boundary Line) とする。1本目の輪ゴム $W(1)$ による境界線は $C(1)$ 、または $R(1)$ のいずれかに分類されるため、以下の定義が得られる。

$$C(1) \cap R(1) = 0 \quad (5.3)$$

$$C(1) \cup R(1) = B \quad (5.4)$$

$n(2 \leq n)$ 本目の輪ゴム $W(n)$ に対しても同様に以下の定義が成り立つ。

$$C(1) \cap \cdots \cap C(n) \cap R(n) = 0 \quad (5.5)$$

$$C(1) \cup \cdots \cup C(n) \cup R(n) = B \quad (5.6)$$

IRFAでの境界線の特徴抽出の手順について、箇条書きで述べる。

1. まず、原図形に対し、画素の4連結性を満足しない境界構成画素として、境界線抽出処理を施す。

2. 前ステップで求めた境界構成画素に対して、輪ゴムかけの際の角点となる点 P_n を求める。
3. 前ステップの結果に対して、1本目の輪ゴムかけ操作を行う。前ステップの残存する境界線情報のうち、輪ゴムとの接線となった境界線分を有意境界線分として抽出する。これにより、残存境界線分が得られる。
4. 前ステップで、1本目の輪ゴムかけ操作を行った結果に対して画素の反転処理を行い、前ステップの結果との差分をもとめる。
5. 前ステップの結果を連結領域ごとに分割する。
6. 各連結領域に対して、2本目、3本目の輪ゴムかけ操作 (Step 2 ~ 4) を行う。
7. さらに分割された場合、再帰的に輪ゴムかけ操作 (Step 2 ~ 4) を行う。

セル画領域の真の輪郭線抽出

輪ゴムのアルゴリズムを用いて、フレーム前後で変化しているセル画領域の抽出を行う。これには、輪ゴムの始点を差分画像より生成した粗い輪郭線情報とし、その対象を、各フレームにおけるエッジ画像とする。エッジ画像に含まれるのは、セル画によるオブジェクト領域の輪郭線だけではない。一方、先に生成した粗い輪郭線は、セル画によるオブジェクト領域の輪郭線を含むが、フレーム間の変化の AND をとっているため、正確な輪郭線とはいえない。そこで、輪ゴムのアルゴリズムを用いることで両者を結びつけ、セル画領域の真の輪郭線を抽出するのである。そして最後に、検出した輪郭線を用いて、元画像から後段のマッチングに用いるテンプレート画像を切り出す。

以上をまとめると、輪郭線抽出のアルゴリズムは次のようになる。

1. 図 5-5 に示すように、対象とするシーケンス内の連続フレーム F_i , F_{i+1} において、差分画像 D_i を生成する。
2. D_i のエッジ画像より、粗い輪郭線 C_i を生成する。これには、任意の閉輪郭線の追跡を行う。
3. F_i , F_{i+1} より、エッジ画像 E_i , E_{i+1} を生成する。
4. $T_i = IRFA(F_i, C_i), T_{i+1} = IRFA(F_{i+1}, C_i)$ ここで、 T_i, T_{i+1} は、 $i, i+1$ 番目のフレームにおけるセル画領域の真の輪郭線、 $IRFA(A, B)$ は、 B を初期輪郭線として、 A に輪ゴムのアルゴリズムを適用することを示す。
5. 図 5-5 に示すように、 F_i , F_{i+1} より、 T_i, T_{i+1} を用いてセル画領域を切り出す。

ただし、元の領域の一部のみが変形する場合には、輪ゴムは内側からかける必要がある (図 5-7 参照)。すなわち、 A が、 $A'(\supset A)$ または、 $A'(\subset A)$ になる場合である。このとき、

差分の輪郭線から，外側にたどることにより，前者では後フレームの A' ，後者では前フレームの A の領域を検出することができる．

本方式の利点として，処理量が少ないこと，さらに，後のテンプレートマッチングへの入力として，ピクセルベースでの領域の抽出が可能になることが挙げられる．

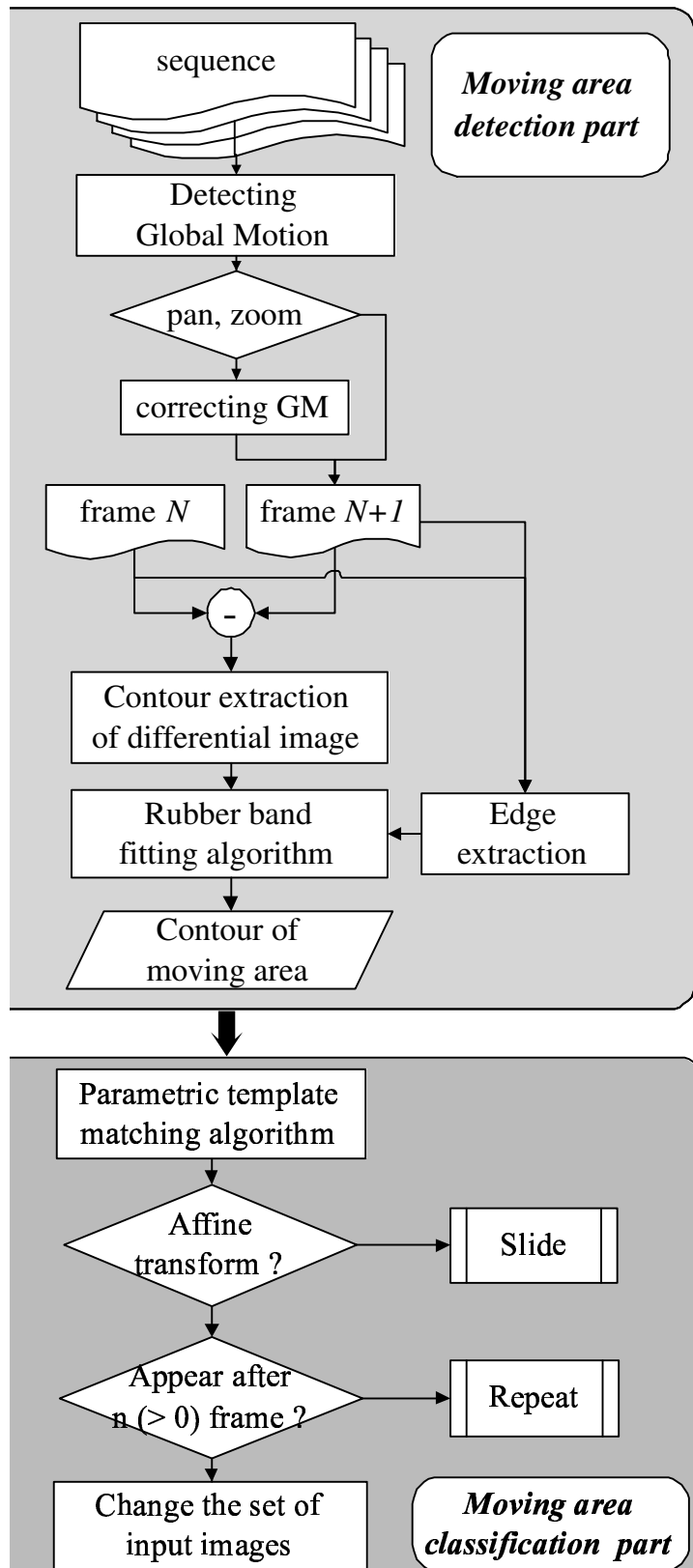


図 5-1: 動領域の検出・追跡アルゴリズム

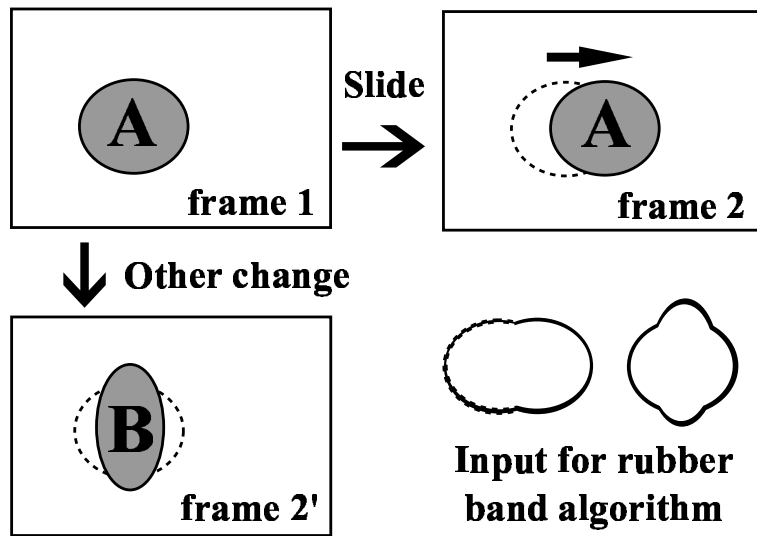


図 5-2: 動領域の粗い輪郭線検出

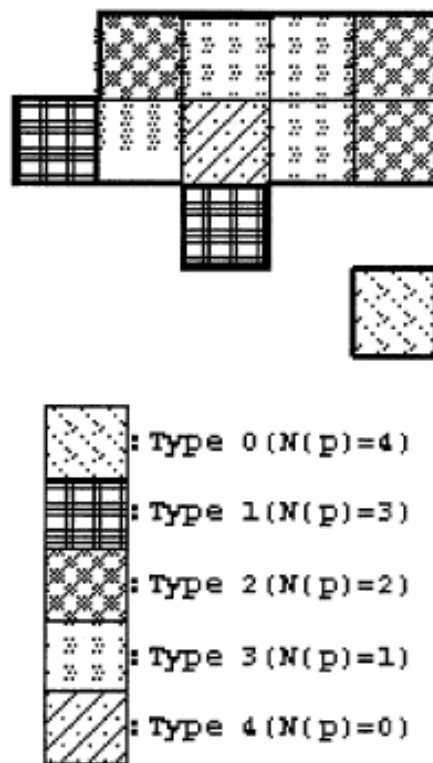


図 5-3: 境界画素の境界線長による分類

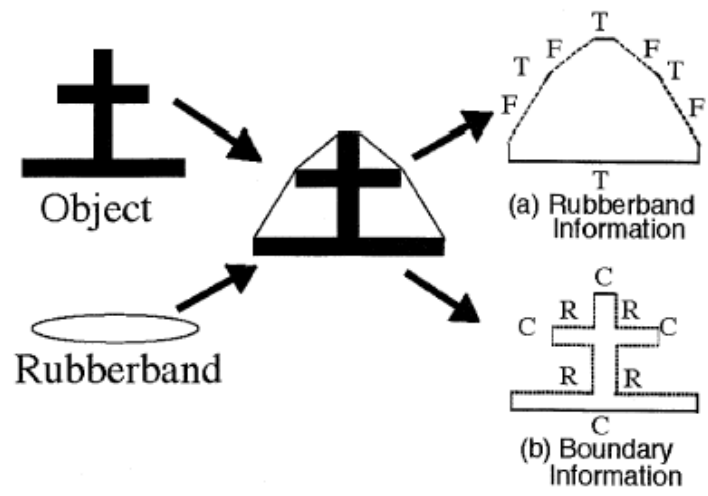


図 5-4: IRFA における境界線情報の定義

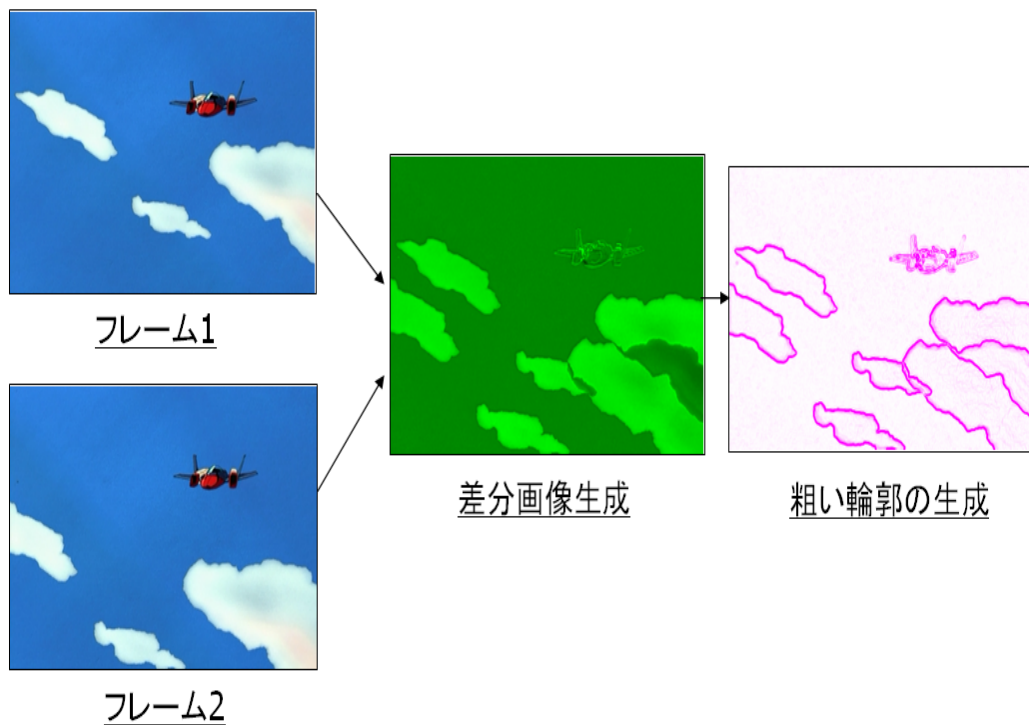


図 5-5: 粗い輪郭線の生成

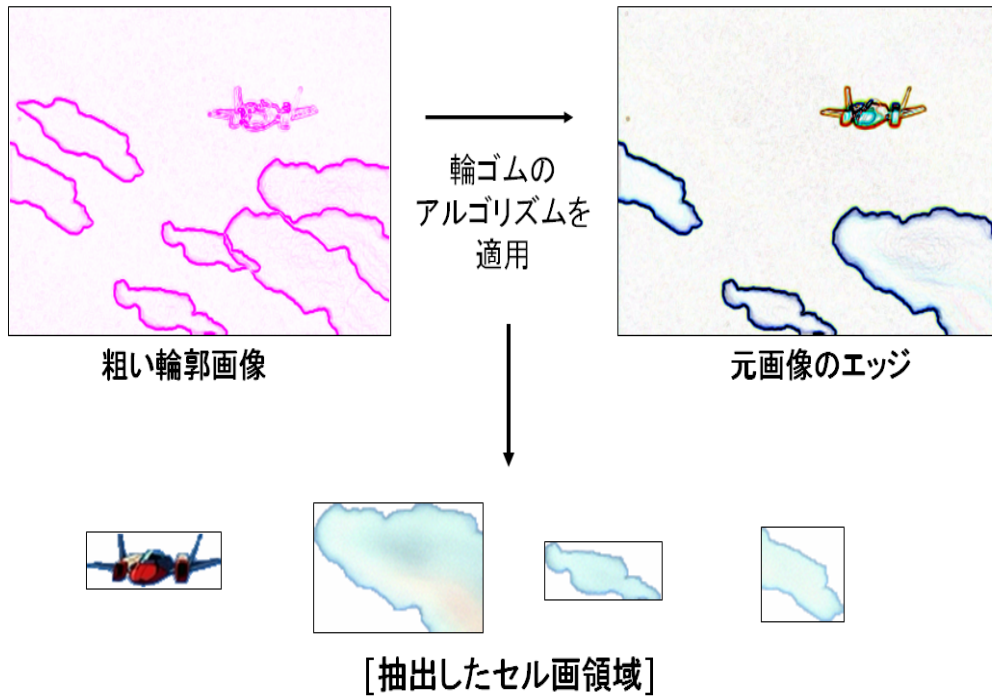


図 5-6: セル画領域の真の輪郭線の抽出



図 5-7: 内側からの輪ゴム処理

セル画領域の抽出例

図5-8 から、図5-15 に、スライドのシーケンスからオブジェクトを抽出した様子を示す。ショット内の連続フレームから差分画像とエッジ画像を生成し、その結果得られた粗い輪郭を利用して、セル画領域を抽出する。

このように、セル画の領域だけが移動するシーケンスの場合、提案手法により精度良く領域の抽出が可能になることがわかる。

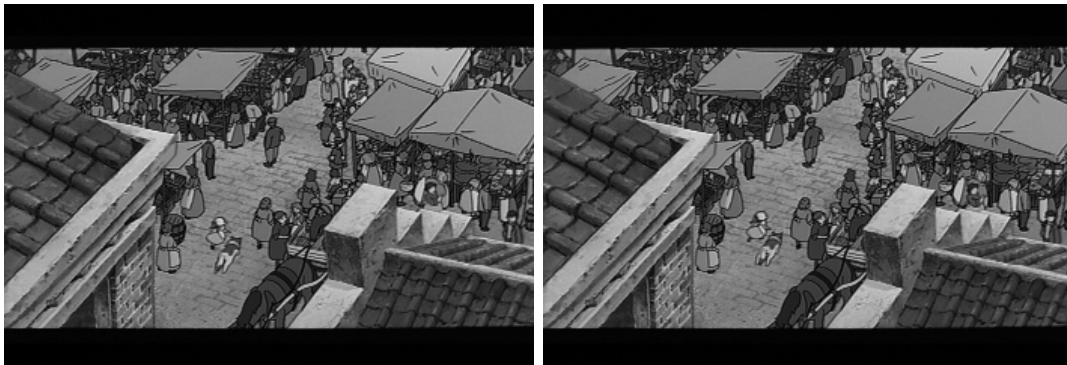


図 5-8: 連続フレーム ('slide')

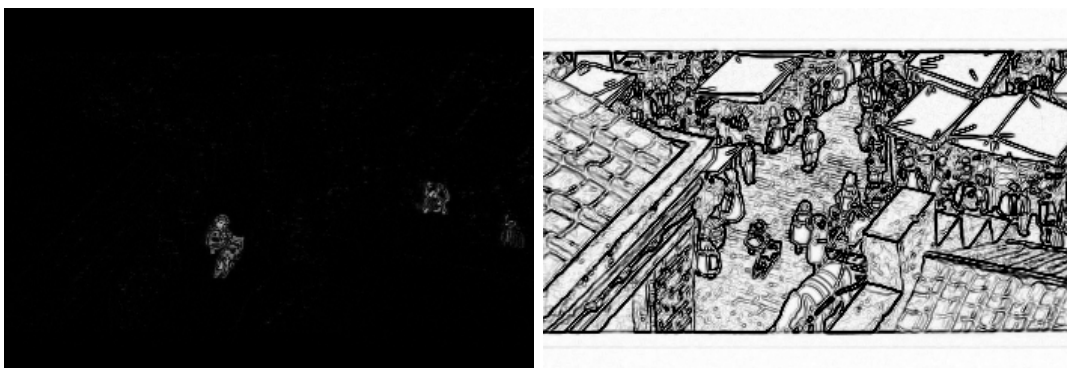


図 5-9: 差分画像とエッジ画像 ('slide')

5.3.2 大局的な動きへの対応

カメラでパンやズームを行うと、画面全体が移動ないし拡大・縮小する。ここでは、これを大局的な動きとよぶ。アニメーション映像においても、擬似的に作られたパンやズームのシーンが登場する。セル画をフィルムに撮影する際、カメラは固定であるため、その背景領域を水平にずらしたり、またはカメラとの距離をかえることで表現される。

前項で述べた動領域の抽出手法では、ショット中にこの大局的な動きが存在する場合、領域の抽出に失敗するという問題がある。なぜならば、単純な差分画像の生成では、背景領域

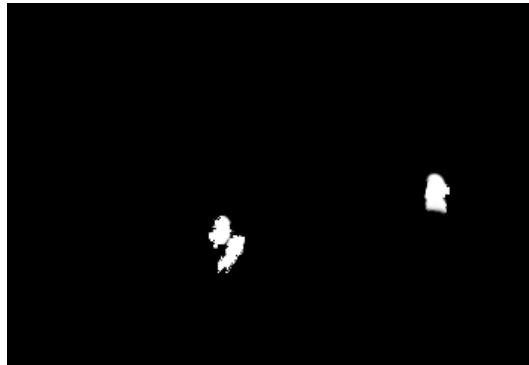


図 5-10: 粗い輪郭の抽出 ('slide1')



図 5-11: セル画領域の抽出 ('slide1')

を消去することができない。先に述べたセル画領域の輪郭線抽出アルゴリズムの、1 番目の段階である。当然、この後の段階の処理にも支障を来す。そこで、ショットにおける大局的な動きを判別し、さらにそれを補正することが必要となる。

図 5-16, 5-17 に大局的な動きがある場合の元画像と、それにより粗い輪郭の生成に失敗する例を示す。

ショットにおける大局的な動きを判別するために、最も原始的な方法は、全探索による手法である。すなわち、隣接フレーム間において、前フレームを上下左右にずらして後フレームとの差分画像を生成する。各点における輝度・色差信号の絶対値差の、重なり部分における総和が最も小さくなったときの "ずれ" が、フレーム間における大局的な動きに相当する、というものである。

ただし、これにはいくつか問題がある。重なり部分の面積は一定ではないため、絶対値差の総和は単位面積当たりで比較することになるが、これだけでは正確に大局的な動きを検出することができない。例えば、背景ではなく複数のセル画領域が移動しているだけの場合でも、誤検出する可能性がある。また、大局的な動きが、パンなど平行な移動に限定するならばよいが、ズームなど拡大・縮小を伴う移動を考慮すると、演算量が莫大になる。

そこで、自然画を対象にこれまで多くの検討がなされているオプティカルフローを用いた



図 5-12: 連続フレーム ('slide2')

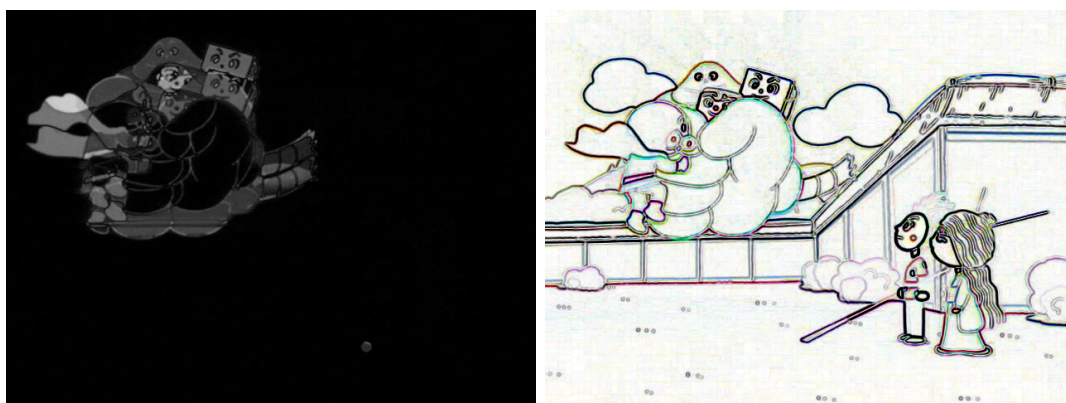


図 5-13: 差分画像とエッジ画像 ('slide2')

手法に着目した [68][70][54] . これらを用いると, ショットにおける大局的な動きを判別することが可能になる. また, パンやズームの識別も可能となる. 一方で, オプティカルフローのみでは, ピクセルレベルで大局的な動きを決定することが難しいという問題もある. そこで, オプティカルフローによりショットにおける大局的な動きを判別し, 完全に背景領域を消去するための, 正確な移動量については全探索により求めることとした.

図 5-18 に, 後に述べる提案手法により大局的な動きの補正を行った場合の例を示す.

ショットにおけるカメラモーションの決定

はじめに, ショットにおけるカメラモーションを決定する手法について述べる. これにはオプティカルフローを用いることを述べた. だが, オプティカルフローを人工的な画像に用いた場合, 平坦な色遣いの部分で, 対応点が正確にとれないという問題がある. そこで, 木村 [54], 西原 [55] らの手法を参考に, フローの信頼性を高める処理を行った.

木村ら [54] の手法では, 勾配法を基に, フローの分布に関する仮定を置いている. 一つは画像の小領域では, オプティカルフローは一定であること, もう一つは, 画像全面に渡り滑



図 5-14: 粗い輪郭の抽出 ('slide2')

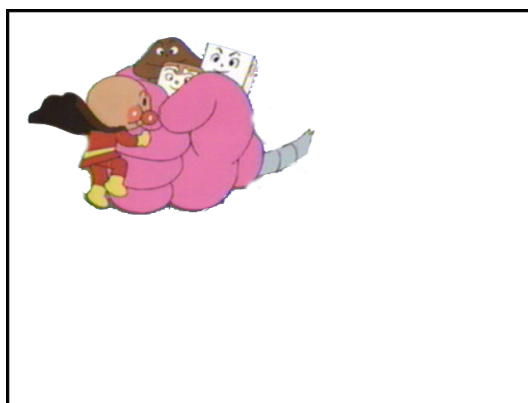


図 5-15: セル画領域の抽出 ('slide2')

らかに変化するということである。この仮定を基に、抽出された各フローの信頼度関数を評価することで、サブピクセルレベルの小さな移動の検出精度に優れる結果が得られる。

以下、オプティカルフローの導出法について説明する。オプティカルフローは、連続フレームにおける、ブロックベースの時間微分で求められる。

$$E_x u + E_y v + E_t = 0 \quad (5.7)$$

E は輝度値を表し、 E_x, E_y, E_t は、それぞれ座標 x, y および時間 t におけるその微分を表す。 u および v は、フローの x, y 成分を表す。この関係は、画像上の任意の点で成り立つため、画像を小ブロック S に区切ったとしても、同様に成り立つ。

そこで、式 5.7 を変形した、

$$E_t(x, y) = -u E_x(x, y) - v E_y(x, y) \quad (5.8)$$

が、ブロック S において成り立つ。

実際の画像シーケンスに適用する際には、誤差関数 E_r を考慮して、

$$E_t(x, y) = -u E_x(x, y) - v E_y(x, y) + E_r(x, y) \quad (5.9)$$

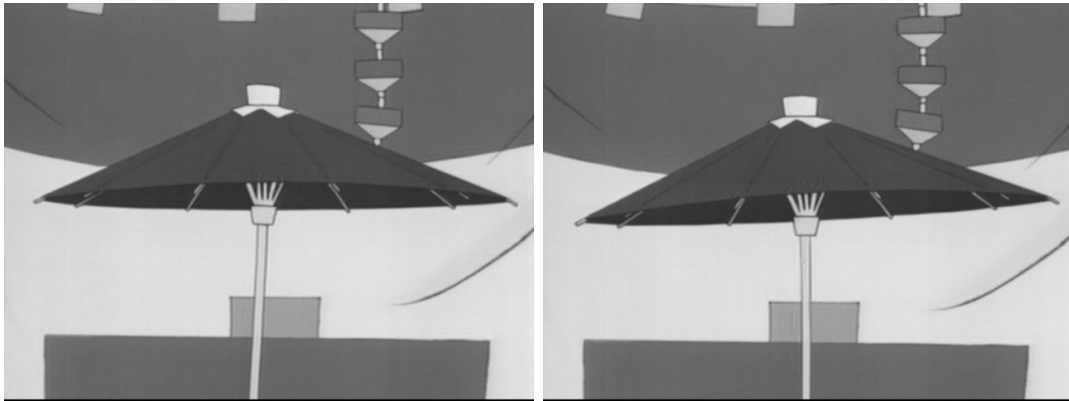


図 5-16: 連続フレーム ('pan3')

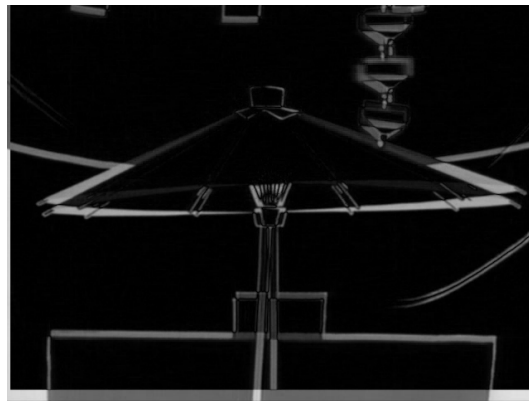


図 5-17: 大局的な動きにより粗い輪郭の生成に失敗する例 ('pan3')

とする必要がある。

E_r の絶対値は, E_x および E_y との内積が直交するとき, 最小になる。したがって,

$$(E_r \cdot E_x) = 0, (E_r \cdot E_y) = 0 \quad (5.10)$$

の条件を満たす。

なお, 関数 f_1 および f_2 の内積 ($f_1 \cdot f_2$) は, 次の通り計算される。

$$(f_1 \cdot f_2) = \int \int_S f_1(x, y) f_2(x, y) dx dy \quad (5.11)$$

これを用いて, 式 5.9, 5.10 より, 次式が導かれる。

$$\begin{pmatrix} -g \\ -f \end{pmatrix} = \begin{pmatrix} a & h \\ h & b \end{pmatrix} \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} \quad (5.12)$$

(u_0, v_0) は, 求めるべきフローを表す。また, 式 5.12 中の各記号は, 次の通りである。

$$a = (E_x \cdot E_y), b = (E_y \cdot E_y), c = (E_t \cdot E_t) \quad (5.13)$$

$$f = (E_y \cdot E_t), g = (E_t \cdot E_x), h = (E_x \cdot E_y), \quad (5.14)$$

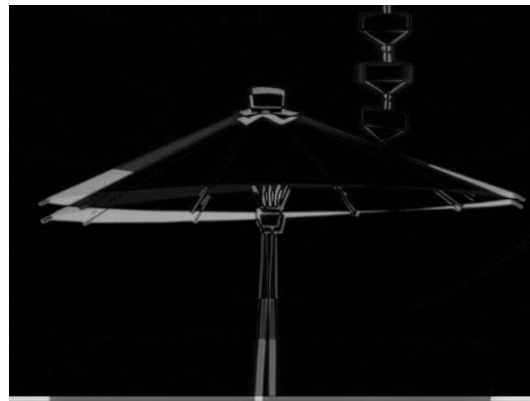


図 5-18: 大局的な動きを補正した場合の差分画像 ('pan3')

以上より，求めるべきオプティカルフローは，次のように計算される．

$$\begin{pmatrix} u_0 \\ v_0 \end{pmatrix} = \frac{1}{ah - h^2} \begin{pmatrix} b & -h \\ -h & a \end{pmatrix} \begin{pmatrix} -g \\ -f \end{pmatrix} \quad (5.15)$$

次に，各オプティカルフローに，信頼性の指標を導入する．時間微分関数 E_t は，フレーム間での画像の移動を検出する．これは，次式のように2つに分けられる．

$$E_t(x, y) = E_m(x, y) + E_{r0}(x, y) \quad (5.16)$$

$$E_m(x, y) = -(u_0 E_x(x, y) + v_0 E_y(x, y)) \quad (5.17)$$

E_m は，画像の移動による項であるといえる．一方で， E_{r0} はそうではない．

ここで， E_t のエネルギーは次式の通り計算される．

$$\begin{aligned} (E_t \cdot E_t) &= (E_m \cdot E_m) + (E_{r0} \cdot E_{r0}) \\ &= |E_m|^2 + |E_{r0}|^2 \end{aligned} \quad (5.18)$$

E_{r0} と， E_m は直交するため，それぞれの絶対値の2乗の和となる．そこで， E_r を最小にする E_{r0} は，マッチングの度合いを測る指標として用いることができる．これを次のように定義する．

$$r_e = |E_{r0}| = \min(|E_r|) \quad (5.19)$$

これは，式 5-13, 5-15 の記号を用いて，次のようになる．

$$r_e = \sqrt{gu_0 + fv_0 + c} \quad (5.20)$$

r_e が大きな値となる場合，それは信頼度の低いフローであるといえる．ここに閾値設定を置く事で，べた塗り部分や，同じパターンの繰り返しなどでフローが散る現象を抑えることが可能になる．

図 5-19 から 5-21 に，抽出したオプティカルフローの様子を示す．



図 5-19: オプティカルフローの抽出 ('pan2')

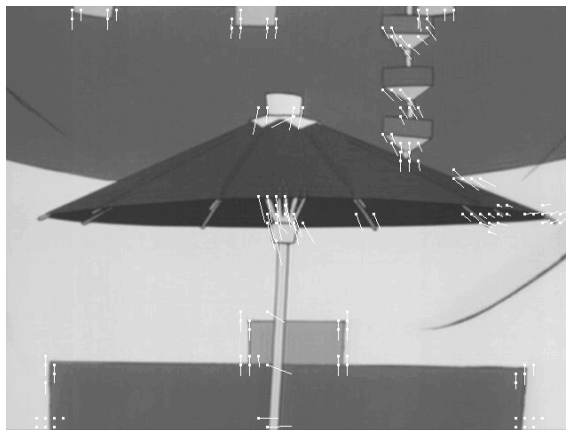


図 5-20: オプティカルフローの抽出 ('pan3')



図 5-21: オプティカルフローの抽出 ('zoom1')

次に、ショットから抽出されたフローは、橋ら [70] の手法によりクラスタリングを行う。これにより、ショット中のパン、ズームの判定を行う。全画面でオプティカルフローが近い向きを持つパン区間では、特定のクラスの頻度が高くなる。一方、ズームでは均等に各クラスに動きベクトルが分布する。検出の手順としては、次のようになる。

1. 8つのクラス(図5-22参照)の頻度の、全体に占める割合を出す。
2. ヒストグラムの中で最大の頻度のクラスを探す(c1とする)
3. その最大のクラスの両側のクラスの内に大きい方のクラスを選ぶ(c2とする)
4. c1とc2のクラスの割合を加える。
5. 閾値によりパン・ズームの分離を行なう。

ここで、最大頻度のクラスのみの割合を用いないのは、動き予測の誤差から2つのクラスに大多数が分布する場合があるからである。

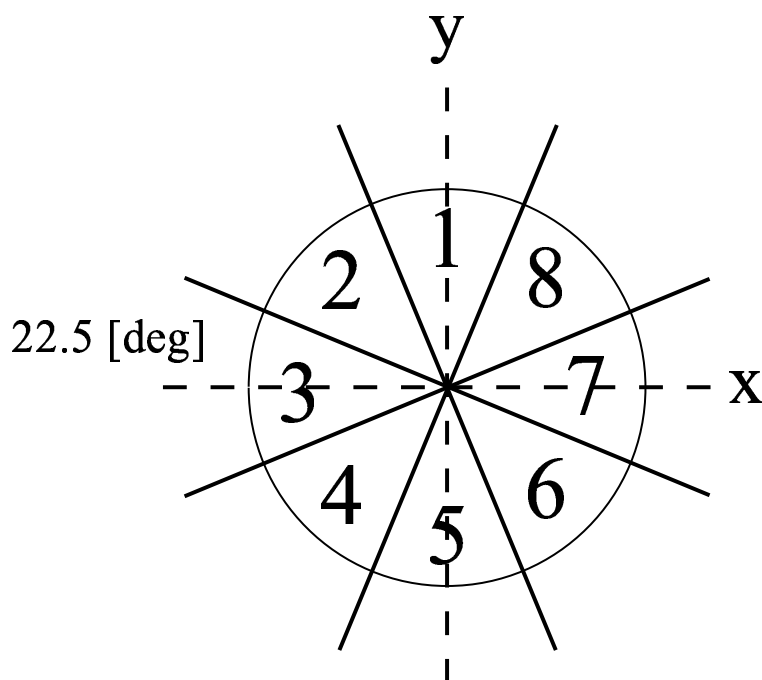


図 5-22: 8 方向へのクラスタリング

大局的な動きの補正

検出されたパン、ズームの情報をもとに、各フレーム間において背景部がどのような動きをしているかを探索する。これには、秦泉寺ら [68] の手法で提案される、差分画像における

オブジェクトの面積を最小にする動きが、大局的な動きを表現している、という条件を用いる（図5-23参照）。

パンのシーンであれば、最尤クラスタの方向に対して全探索を行い、差分画像において閾値以上の画素の数が最も少なくなる点を求める。

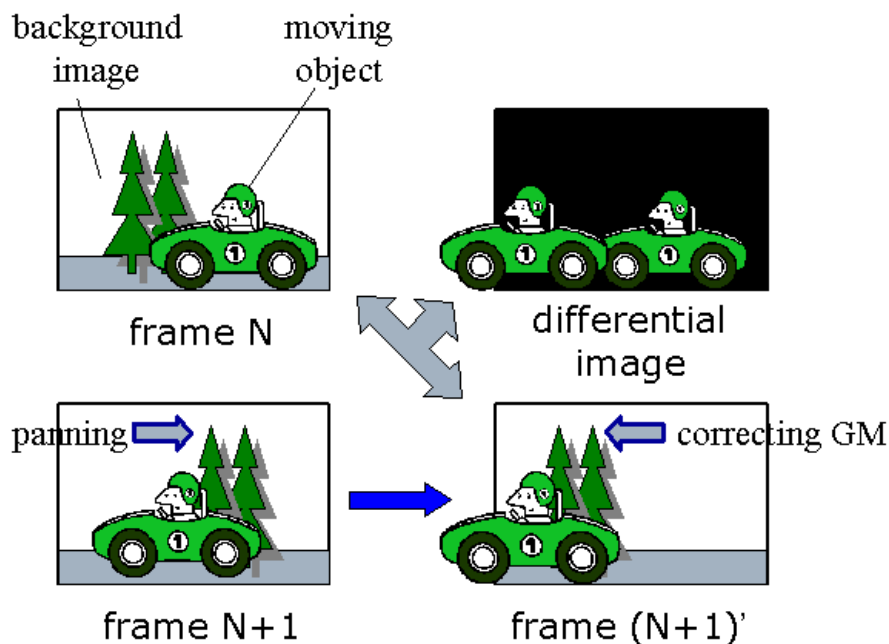


図 5-23: 大局的な動きの補正

この結果を用いて差分画像を生成することで、大局的な動きがある場合にも、無い場合と同様の手法でオブジェクトを抽出することが可能になる。

5.3.3 実験結果

大局的な動きの検出，補正に関して，実験を行った．いずれもパンやズームを含むショットに対して，オプティカルフローを用いることでその検出を行った．そして，パンを含むショットに関しては，全探索法により求めた補正值と，手動で求めた実測値を比較した．

実験条件を表5-1に示す．図5-24に，パン（“pan1”～“pan3”）およびズーム（“zoom1”）を含むショットにおける各フレーム間の全フローを同一平面上に展開したものと，それをクラスタリングした結果を示す．なお，“pan1”，“pan3”はオブジェクトの動きを含み，“pan2”および“zoom1”はオブジェクトに動きは無い．

フローの信頼度を高める前置処理の結果，平坦な色遣いを含む映像においても，パンやズームの検出を行うことが可能であることが確認された．特に，“pan3”ではオブジェクトの動きが比較的多いが，クラスタリングアルゴリズムと併せることで，正確な判定が可能になっている．

次に，パンのショットにおいて，全探索を行った結果を図5-28～図5-30に示す．各フレー

表 5-1: 実験条件

テスト画像	“pan1”, “zoom1”	“pan2”, “pan3”
画像解像度	352[pixel] × 240[line] 4:2:0	640[pixel] × 480[line] 4:2:0
色空間	YCbCr	YCbCr

ムにおける大局的な動きを時系列上にグラフにしたもので、横軸はフレーム番号、縦軸は画素数である。

単純な全探索法では、誤探索の問題がある。画面中に占める割合の大きいオブジェクトが移動する場合、そちらを大局的な動きと判定することがある。ショットのシーン判定を行わない場合、いくつかのフレームにおいて誤判定した。しかし、前段階で抽出されたパン方向へ探索範囲を絞ることにより、これを回避できた。すなわち、オブジェクトがカメラモーションと別の方向に動く場合は、その探索結果を除外することが可能になる。結果を見ると、検出結果はほぼ実測値と等しく、正確に抽出することが可能であるといえる。

オブジェクト抽出結果

提案手法によるオブジェクト抽出は、エッジ情報に基づいているため、輪ゴム法の入力として理想的な初期輪郭線が得られれば、正確なオブジェクトの抽出が可能になる。ただし、差分画像のエッジが、背景領域における輪郭線と交差している場合などは、初期輪郭線が得られず、オブジェクトの抽出そのものに失敗する。

表 5-2 にオブジェクトの抽出結果を示す。フレーム間でオブジェクトに移動等の変化がある場合のうち、差分画像から初期輪郭線が得られた場合を成功と判断し、その割合を示す。テスト画像は、先ほどのシーケンス (“pan1” および “pan3”) に加え、大局的な動きが無い場合のショット (“fix1” および “fix2”) について示す。これらの実験条件は、“pan1” と同じである。

表 5-2: オブジェクト抽出結果

テスト画像	オブジェクトに変化がある場合	初期輪郭線獲得に成功した場合	割合 %
pan1	7	5	71
pan3	20	14	70
fix1	8	8	100
fix2	18	17	94

大局的な動きがある場合は、1 ピクセルでもその補正がずれると、背景領域による偽の輪郭線が発生し、オブジェクトの輪郭線と交差するため輪郭線が得られないことがある。その

ため、全探索において、オブジェクトの面積最小となる点という条件に加え、さらに精度を上げるための拘束条件が必要になると考えられる。一方で、大局的な動きが無い場合には、オブジェクトの正確な抽出が可能となった。

5.4 抽出したセル画領域のマッチング

判定すべき領域を抽出した次の段階は、その領域がどのような変化をしているかを分類する。スライドであれば、次フレームにおいて、同一の領域が、位置変化や拡大・縮小、回転といった変化を伴って登場する。一方、リピートをしている部分であれば、 $n(> 1)$ フレーム後に、同一の領域が同じ位置に登場する。したがって、抽出した領域は、そのショット中の全フレームを対象として、アフィン変換の可能性を考慮しつつ、探索する必要がある。

これには、テンプレートマッチングの技術を用いる。アフィン変換に対応する必要があること、また大量のフレーム中を対象に検出する必要があることから、矩形のテンプレート領域を全体にずらして相関関係を求める SSDA 法では対応できないため、パラメトリックテンプレート法を用いる ([67])。これは、テンプレートの集合がなすパラメトリックなテンプレート空間を構成し、このテンプレート空間と探索画像とのマッチングを高速に行うアルゴリズムである。以下、アルゴリズムを説明する。

一般に、画像サイズ $l_x \times l_y$ である2つの画像 g, t の正規化相関値 ρ は、次のように計算される。

$$\rho = \frac{\sum_{(k,s) \in T} (\Delta t) \times (\Delta g)}{\sqrt{\sum_{(k,s) \in T} (\Delta t)^2} \sqrt{\sum_{(k,s) \in T} (\Delta g)^2}} \quad (5.21)$$

ここで、

$$\Delta t = t[k, s] - \bar{t}$$

$$\Delta g = g[k, s] - \bar{g}$$

$$T = ((k, s) | 1 \leq k \leq l_x, 1 \leq s \leq l_y)$$

であり、 $t[k, s], g[k, s]$ は、位置 (k, s) での画像の輝度値を表す。また、 \bar{g} は、画像 g に対する平均値であり、

$$\bar{g} = \frac{\sum_{(k,s) \in T} g[k, s]}{\sum_{(k,s) \in T}} \quad (5.22)$$

で定義される。画像 g の正規化関数画像 \hat{g} を、

$$\hat{g}[k, s] = \frac{\Delta g}{\sqrt{\sum_{(k,s) \in T} (\Delta g)^2}} \quad (5.23)$$

と表し、 \hat{g}, \hat{t} の相関演算 (\hat{g}, \hat{t}) を、

$$(\hat{g}, \hat{t}) = \sum_{(k,s) \in T} \hat{g}[k, s] \times \hat{t}[k, s] \quad (5.24)$$

とすると、正規化相関値 ρ は、次式で表される。

$$\rho = (\hat{g}, \hat{t}) \quad (5.25)$$

正規化された画像 \hat{t} は以下の条件を満たす。

$$\sum_{(k,s) \in T} \hat{t}[k, s] = 0 \quad (5.26)$$

$$\sum_{(k,s) \in T} \hat{t}[k, s] \times \hat{t}[k, s] = 1 \quad (5.27)$$

以上を用いて、2画像間の正規化相関をパラメトリックな空間と対象画像間の正規化相関に拡張する。 t_1, \dots, t_M を、 $l_x \times l_y$ サイズのテンプレート画像としたとき、正規化されたテンプレート $\hat{t}_1, \dots, \hat{t}_M$ から線形に張られるテンプレート空間を、 M 個のパラメータ $w_i (i = 1, \dots, M)$ を導入することにより、

$$\sum_{i=1}^M w_i \hat{t}_i (0 \leq w_i \leq 1, \sum w_i = 1) \quad (5.28)$$

として構成できる。この大きさを1に正規化したテンプレート、

$$t(w) = \frac{\sum_{i=1}^M w_i \hat{t}_i}{|\sum_{i=1}^M w_i \hat{t}_i|} \quad (5.29)$$

は、正規化条件 5.26 を満たす。すなわち、この空間内のテンプレートはパラメータ w_i を用いて $t(w)$ と表すことができ、画像 g とパラメトリックテンプレート $t(w)$ との正規化相関値は、

$$(\hat{g}, t(w)) \quad (5.30)$$

で計算することができる。

次に、 \hat{g} をある正規化画像としたとき、この \hat{g} と最大の相関を示す空間内の正規化テンプレート $t(w)$ を求める。これは、

$$(\hat{g}, t(w)) \quad \max \sum_{i=1}^M w_i = 1 \quad (5.31)$$

を満たす $w_i (i = 1, \dots, M)$ を求める最大化問題となる。これをラグランジュの未定定数法を用いて解くと、最大相関値を与えるパラメータ \vec{w} は、

$$\vec{w} = \frac{H^{-1} \vec{G}}{(\vec{n}, H^{-1} \vec{G})} \quad (5.32)$$

で与えられる。ここで、

$$\vec{w} = \begin{pmatrix} w_1 \\ \vdots \\ w_M \end{pmatrix} \quad (5.33)$$

$$H = \begin{pmatrix} (\hat{t}_1, \hat{t}_1) & \dots & (\hat{t}_1, \hat{t}_M) \\ \vdots & \ddots & \vdots \\ (\hat{t}_M, \hat{t}_1) & \dots & (\hat{t}_M, \hat{t}_M) \end{pmatrix} \quad (5.34)$$

$$\vec{G} = \begin{pmatrix} (\hat{g}, \hat{t}_1) \\ \vdots \\ (\hat{g}, \hat{t}_M) \end{pmatrix} \quad (5.35)$$

$$\vec{n} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \quad (5.36)$$

である。パラメトリックテンプレート $t(w)$ が表現する特徴量 v (位置ずれ量, 回転角, スケール値, あおり角等) はパラメータ \vec{w} と端点テンプレートが表現する特徴量 v_1, \dots, v_M を用いて,

$$v = \sum_{i=1}^M w_i v_i \quad (5.37)$$

と求まる。

以下に, パラメトリックテンプレートマッチング法をアルゴリズム形式でまとめる。

1. マッチングの基準の変動に対応して, テンプレート空間を張る端点テンプレート t_1, \dots, t_M を選択する。
2. 端点テンプレート間の相互相関行列 H を計算する。
3. テンプレートと同サイズのマッチング画像 g と, 端点テンプレートの相関値ベクトル \vec{G} を計算する。
4. 式 5.32 からパラメータ \vec{w} , 式 5.37 から特徴量 v を計算する。

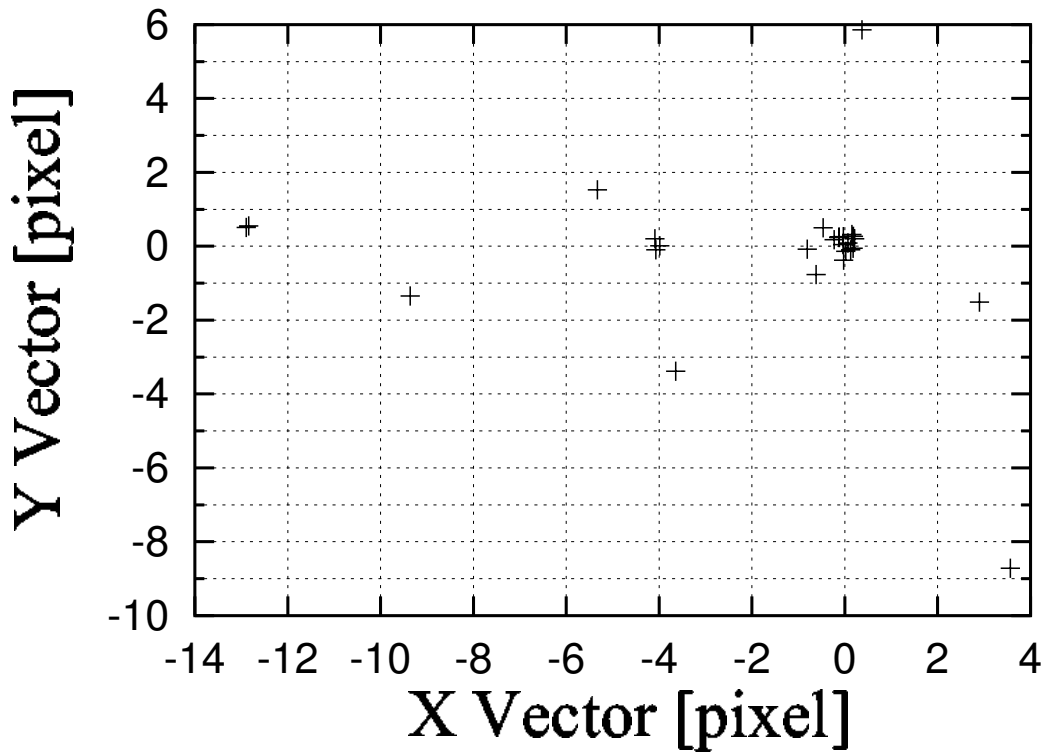
マッチング画像とパラメトリックテンプレートとの最大相関値及び特徴量は端点テンプレートとマッチング画像との相関値ベクトル \vec{G} と端点テンプレート間の相互相関行列 H を用いて計算することが可能である。

以上を用いて同一のセル画領域の検索を行う。スライドやリピートが検出された場合, そこは同一のセル画により表現された領域である。スライドならば, 前フレームからの位置変化など移動パラメータ, リピートならば元のフレームからのフレーム数を保存する。これにより, 前章で述べた静止画符号化方式を動画像に応用する際, 符号化する情報量を減らすことができる。

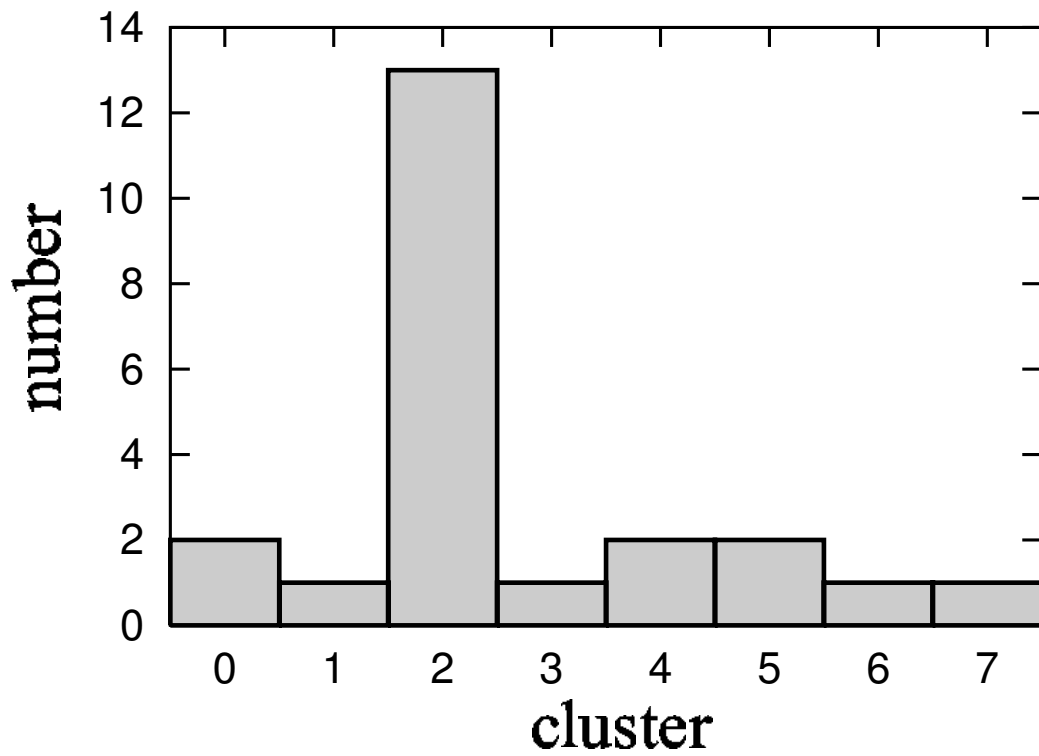
5.5 むすび

本章では，セルアニメーション映像を対象に，ショット中からセル画によるオブジェクトを抽出するための手法を提案した．実験により，カメラ固定の状態では，高い確率でオブジェクトの抽出に成功することを確認した．

また，ショット中に大局的な動きを含む場合への対応を考察した．実験の結果，カメラワークの判定は，オブジェクトの動きによらず成功することがわかった．大局的な動きを補正して差分画像を生成することで，オブジェクトの抽出率が向上した．しかし，補正に1画素でもずれが生じた場合，抽出に失敗することがあることも判明した．今後の課題としては，ズームや，パンとズームを同時に含むような場合でも適用できるかどうかを確認することが挙げられる．

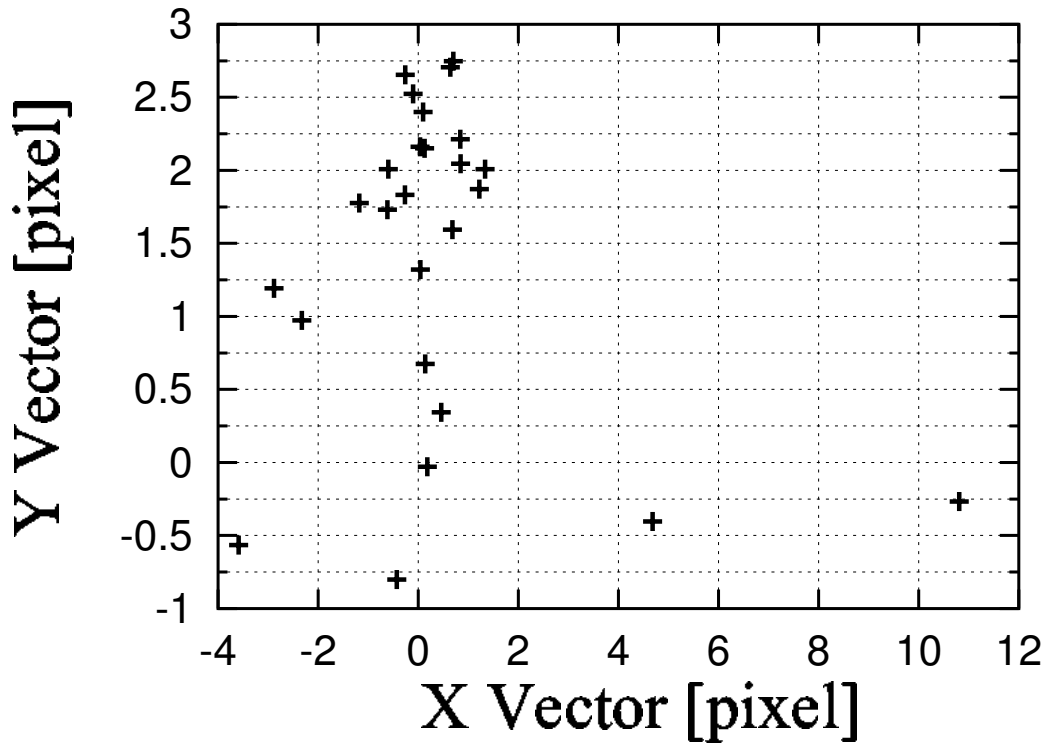


(a) “ショット中のオプティカルフロー”

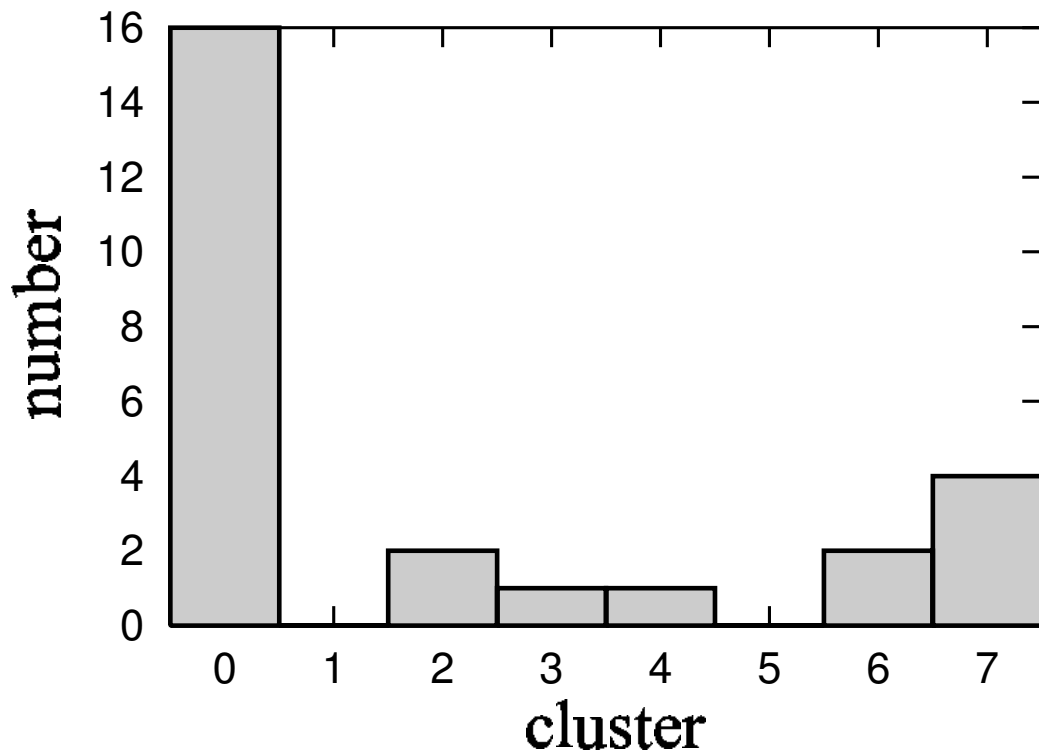


(b) クラスタリング結果

図 5-24: 大局的な動きがあるショットでのオプティカルフロー (“pan1”)

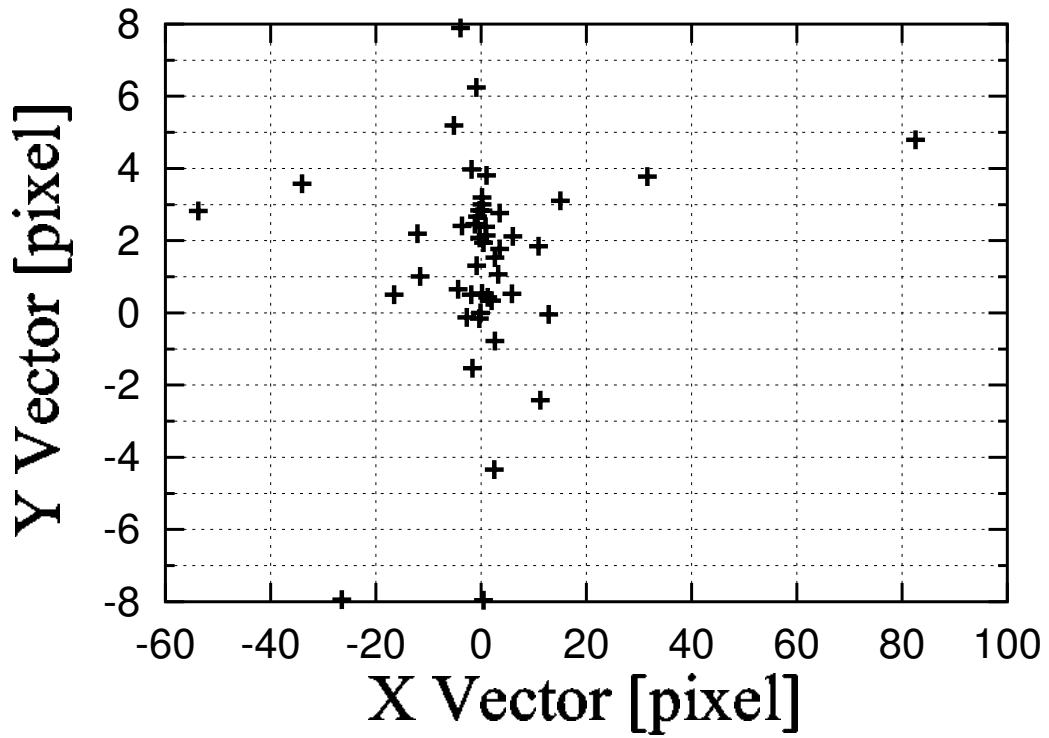


(a) “ショット中のオプティカルフロー”

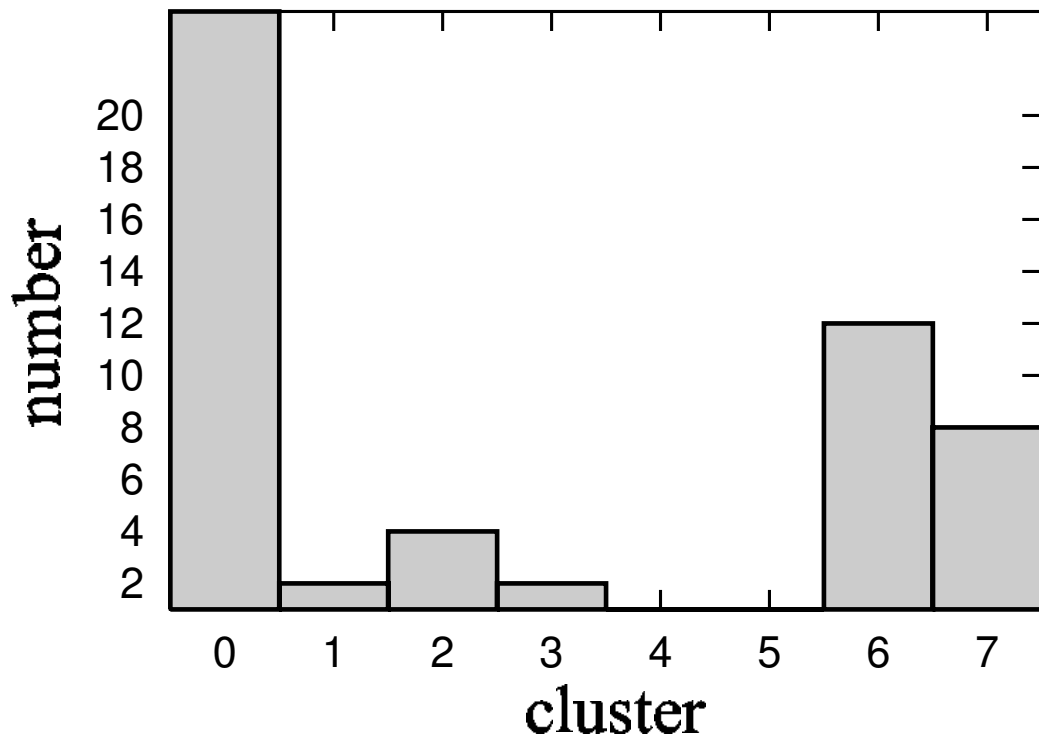


(b) クラスタリング結果

図 5-25: 大局的な動きがあるショットでのオプティカルフロー (“pan2”)

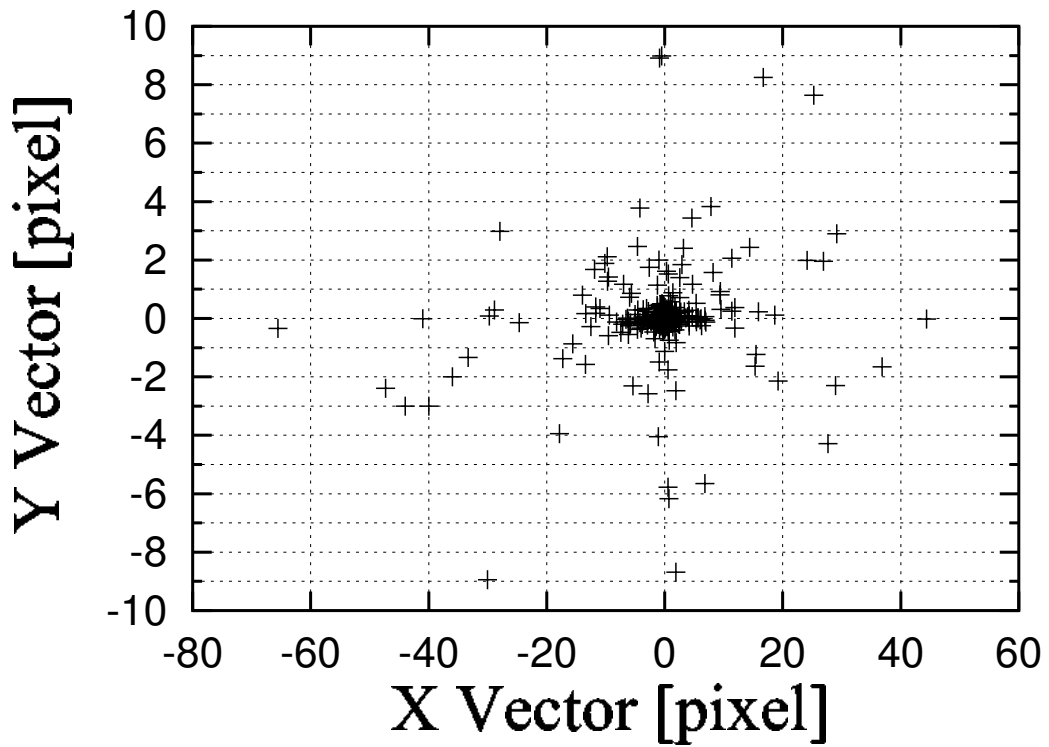


(a) “ショット中のオプティカルフロー”

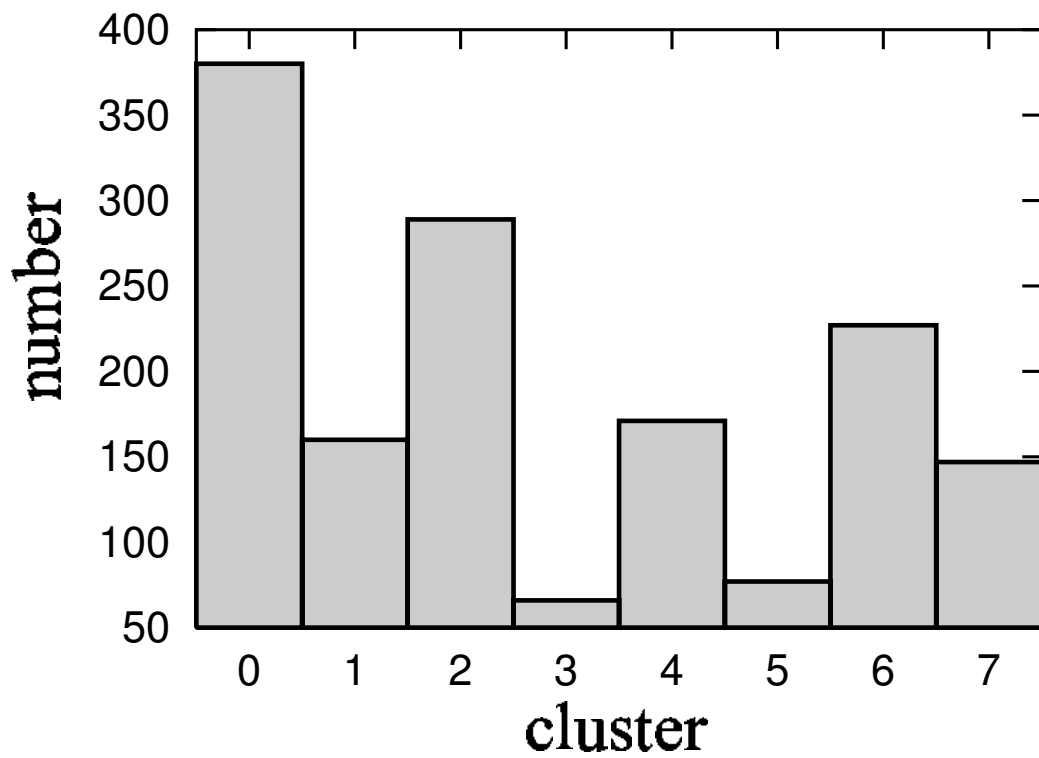


(b) クラスタリング結果

図 5-26: 大局的な動きがあるショットでのオプティカルフロー (“pan3”)

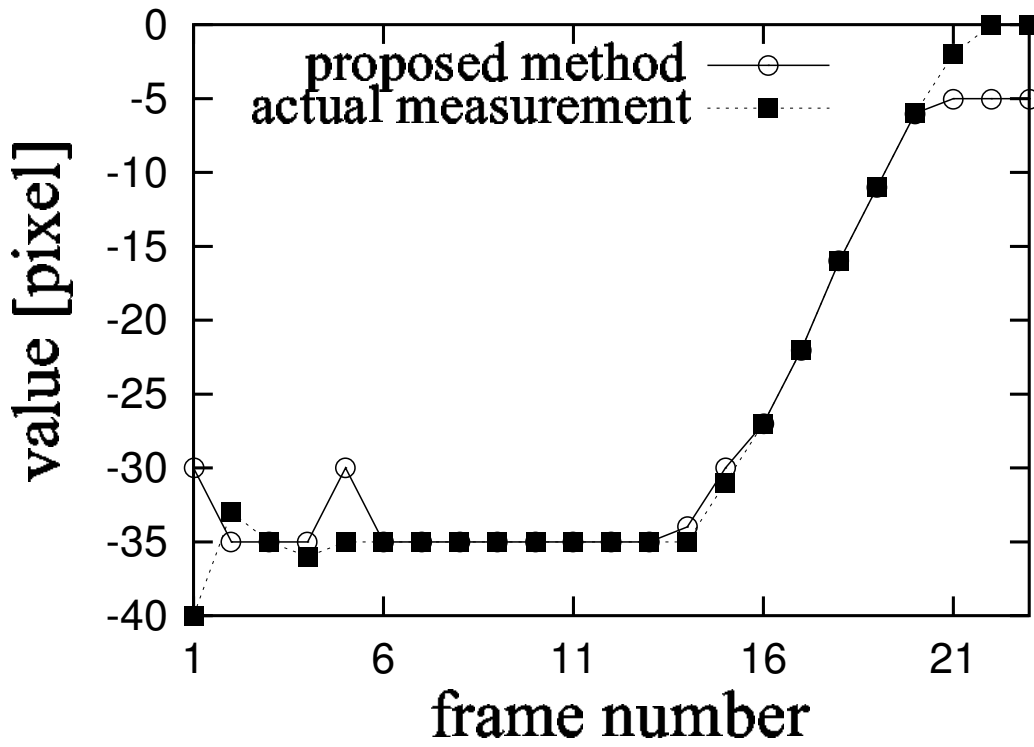


(a) “ショット中のオプティカルフロー”

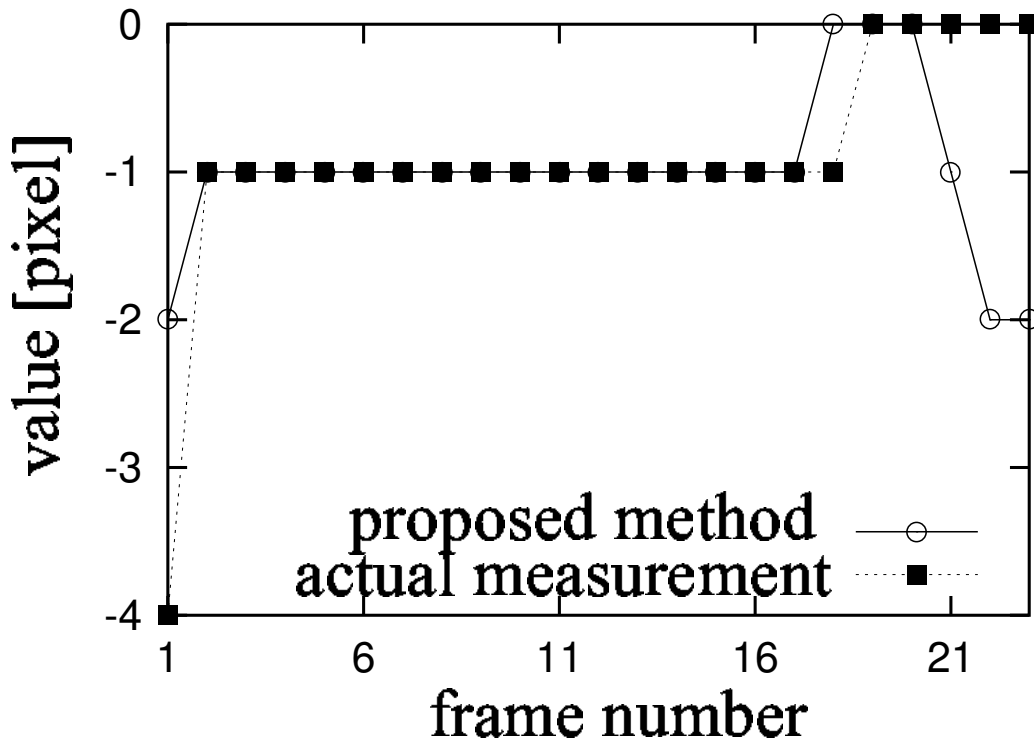


(b) クラスタリング結果

図 5-27: 大局的な動きがあるショットでのオプティカルフロー (“zoom1”)

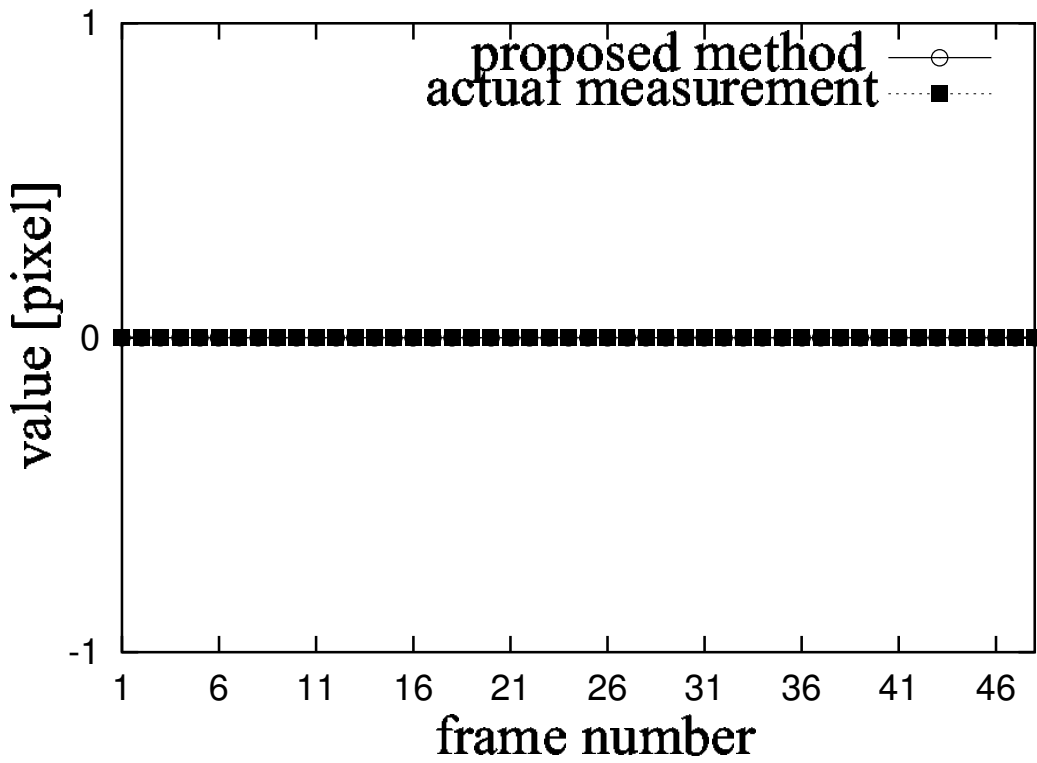


(a) X 成分

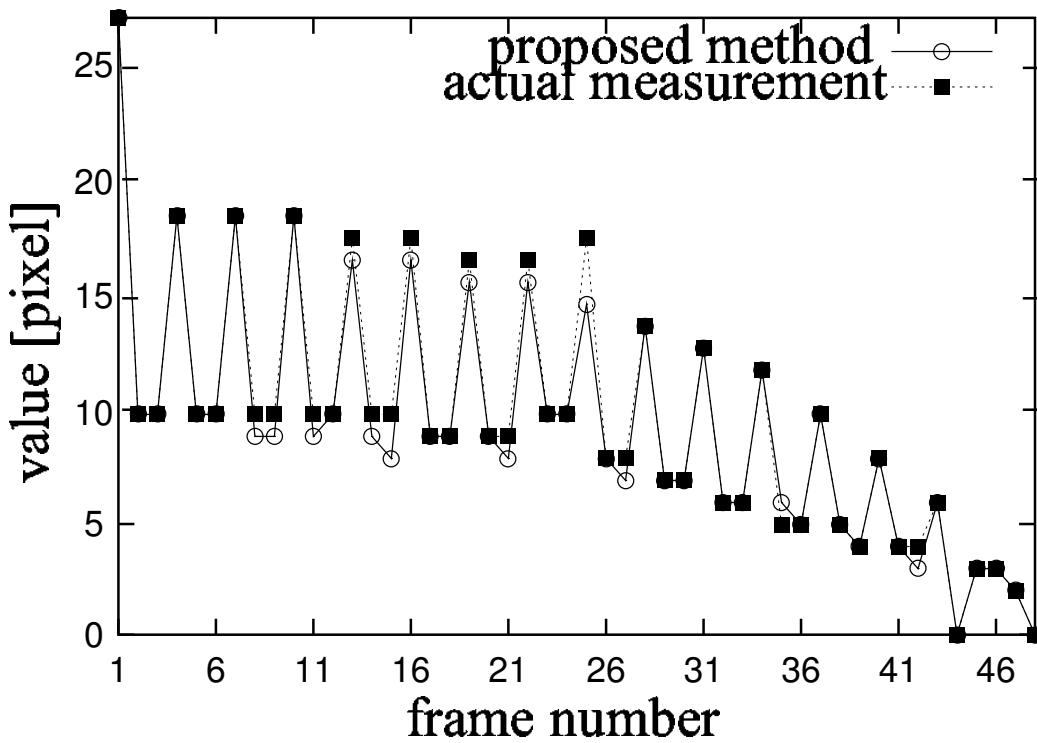


(b) Y 成分

図 5-28: “pan1” における検出結果

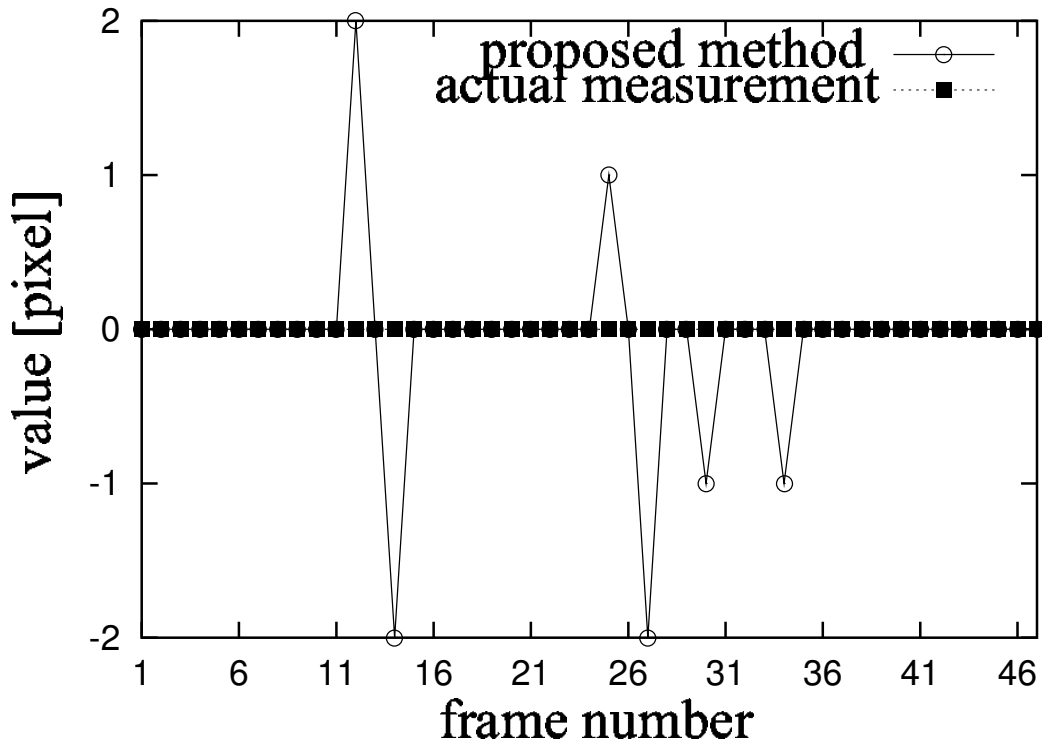


(a) X 成分

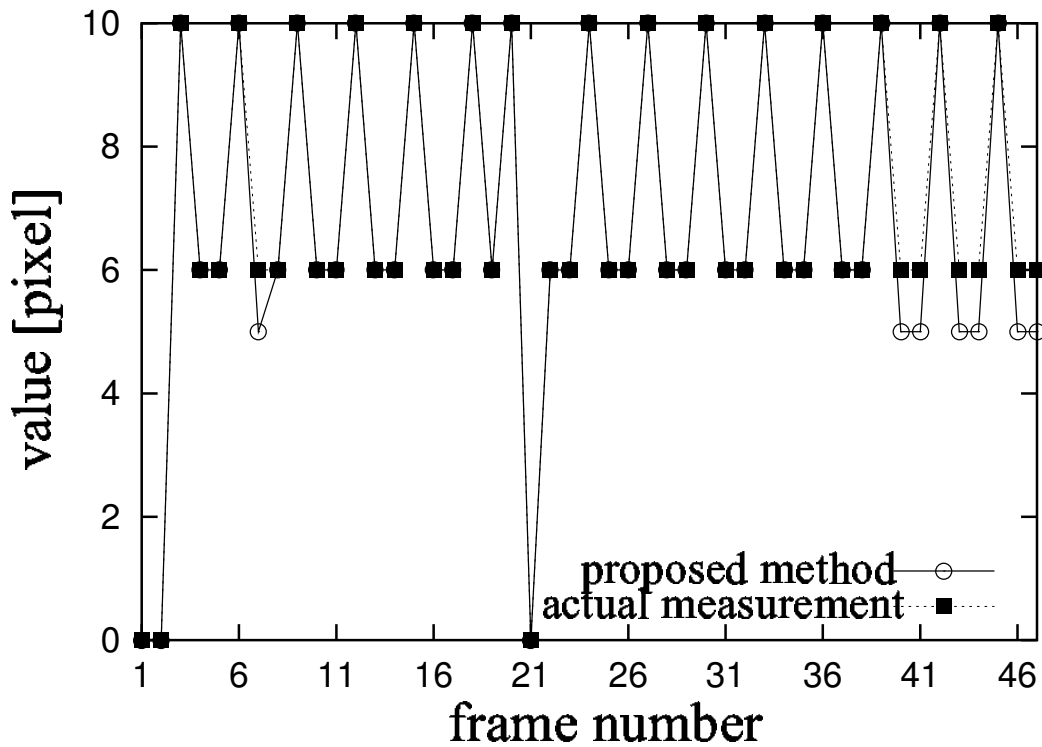


(b) Y 成分

図 5-29: “pan2” における検出結果



(a) X 成分



(b) Y 成分

図 5-30: “pan3” における検出結果

第6章

結論

本章では本論文を総括し、今後の検討課題を述べる。

6.1 総括

第1章は序論であり、研究の背景及び研究の目的を述べた。研究の背景・目的として次が挙げられる。動画像・音声符号化方式の国際標準として MPEG-1, MPEG-2, MPEG-4 が策定された。一方で、端末の高機能化が進み、デコーダのソフトウェア処理が実現された場合、新たな符号化方式のコンテンツを復号可能となる可能性がある。そこで、コンテンツオリエンテッド方式の一例として、セル画アニメーション符号化方式を開発し、コンテンツオリエンテッド符号化方式の有効性を示す必要性を述べた。

第2章では、現在多くの符号化の基盤となっている DCT を用いた符号化方式の問題点を挙げ、アニメーションコンテンツに対する符号化ノイズについて述べた。すなわち、自然画像を DCT 変換すると、係数パワーが低周波領域に集中するため、高周波成分の情報を粗く削っても画質劣化は目立ちにくい。一方、高周波成分を多く含むコンテンツに対し DCT 変換すると、係数パワーが低周波領域から高周波領域まで一様に分散してしまう。これにより、高周波成分の情報を粗く削ると画像内のエッジ周辺にモスキートノイズと呼ばれる雑音が顕著に現れるというものであった。一方で、既存のベクトル記述言語である、Macromedia 社の Flash, および W3C 標準の SVG について述べた。本論文で提唱する符号化方式は、SVG を用いて記述するため、ベクトル記述言語の特徴、符号化に用いる際の機能について述べた。

第3章では、本符号化方式の基礎となる、ベクトル変換のためのアルゴリズムについて述べた。はじめに、数学的に曲線を表現するためのパラメトリック関数について述べた。また、これによる点列間の補間アルゴリズムについて説明した。そして、アニメーション画像に適応する場合の補間アルゴリズムの問題点を指摘し、先行研究による効率のよい線近似手法を紹介した。最後に、符号化データ出力のための SVG フォーマットについて、W3C の規格から一部を引用し、近似データの出力例を示した。

第4章では、静止画に対するセル画アニメーションの符号化方式を提案した。アニメーション画像の特性に着目し、画像に含まれる線要素及び輪郭形状の情報を符号化する方式に

関する検討を行った。これには、符号化画像を二層で構成することを提案する。まず、元画像から抽出した線画と均等色領域の輪郭線を、スプライン関数により近似することで、有意点情報を抽出する。従来手法における背景領域からの均等色領域誤抽出の問題を解決するため、平滑化操作を用いることで、複雑な背景を含む画像にも対応した。基本層では、それら有意点の情報と、領域についての色情報のみのベクタ画像である。また、平滑化により失われた情報を補うため、付加層において、抽出された均等色領域における元画像との差分情報を、DCTにより符号化することを提案した。

第4章では、前章で述べた符号化方式を、動画へ適用するためのアルゴリズムを述べた。はじめに、アニメーション映像において、キャラクタの動作など、セル画により表現される動きについて分類した。次に、これら動作を検出するためのアルゴリズムとして、フレーム間の差分画像に基づいた方法を提案した。これにより、セル画領域の検出が可能になった。映像中に大局的な動きがある場合についても、その検出、補正アルゴリズムを提案した。実験の結果、大局的な動きを補正して差分画像を生成することで、精度良くオブジェクトを抽出できることを確認した。そして、検出したセル画領域を、他のフレームから探索する技術についても述べた。

第5章は結論であり、本論文の総括及び今後の課題について述べた。

6.2 今後の課題

静止画 (Intra frame) に関わる課題と、動画 (Inter frame) に関わる課題を述べる。

6.2.1 静止画における課題

近年、オブジェクトレベルでの階層的符号化の研究が行われている [17]。これは、画像中の領域の重なり関係を判断し、階層に分割して符号化を行うものである。たとえば、顔の領域では、そこに包含される目や口といった領域を、顔の下の階層に分類する。さらに、目を白目の部分と黒目の部分に、という具合である。これにより、要求されるビットレートに応じて、情報の少ない階層を切り落とすことで、スケーラブルな伝送を可能にするものである。また、顔の中の一部分が変化した場合には、その階層のみ差分情報を符号化することで、顔全体を改めて符号化する必要がなくなり、符号量の削減にも役立つ。

本論文で述べた静止画符号化では、DCT で符号化するテクスチャ情報を付加的な層として扱った。しかし、線と面の包含関係をより詳細に分類することは、重要な検討課題である。これにより、基本層として扱ったベクトル情報も、さらに細かい層に分割できる可能性がある。さらに、符号化した各領域や線分にリンクを張ることが可能になる。リンク情報を作成することで、単に領域や線分の集まりだった符号化情報がオブジェクトとしての構造を持つことになる。たとえば、顔の輪郭線に対する目の領域といった包含関係が、対応するベクタ情報にリンク構造を与え、顔を構成するオブジェクトとしての意味を持たせる。よって、符号化情報により高いレベルでの抽象度を与えることが可能になる。オブジェクト単位での検

索や、重要度の低いオブジェクトの切り捨てによる圧縮，再編集する際のオブジェクトレベルでの変更など，高次の取り扱いへの可能性もある．

6.2.2 動画における課題

本論文では，セル画領域単位での動き予測・補償の提案を行った．さらに高圧縮を実現するために，動き情報自体のベクトル化が挙げられる．ある領域が，異なるフレームに位置を変えて登場した場合，その位置は，時間をパラメータとして，座標上に曲線を描く．MPEG等では，動き補償を行うブロックでは，予測元のブロックからの時間と座標のみを符号化する．つまり，一本のベクトルとして表現される．これに対し，同一の領域が常に，あるいは一定の間隔で登場するという前提がある場合，たとえばその重心の座標は，時空間上の曲線として扱うことができる．したがって，この曲線を関数で近似することにより，線分の近似と同様に高圧縮を行うことが可能になる．

入力信号に対する考慮も必要になる．一般に，セルアニメはフィルムに秒 24 コマの速度で撮影される．一方，日本のテレビでの伝送は，NTSC(National TV Standards Committee)とよばれる方式で行われ，1秒で 29.97 フレームでインターレース表示する．そのため，秒あたりのフレーム数を調整するため，俗にテレシネ変換と呼ばれる処理が行われる．インターレースのため，テレビの 1 フレームは，2 つのフィールドから構成される．そこで，フィルムの 2 コマをテレビの 5 フィールドに置き換えることによって，フリッカーを防ぐ．具体的には，フィルムの 1 コマ目をテレビの 2 フィールド分に変換し，2 コマ目をテレビの 3 フィールド分に変換するというわけです．NTSC で伝送されたコンテンツの符号化を考える際には，この影響を考慮する必要がある．

謝辞

本研究を進めるにあたり、日頃から惜しみなく御指導して頂きました早稲田大学国際情報通信研究センター 渡辺 裕 教授に深く感謝致します。

貴重な時間を割いて研究の方向性を御指導頂きました同 富永 英義 教授に心から感謝致します。

また、研究の方向性をはじめ、研究の細部に至るまで、数々の有意義な御意見、御助言を賜りました同 亀山 渉 教授に深く感謝致します。

本研究のきっかけを与えて下さり、研究の進め方から文章の書き方まで丁寧に御指導下さった平成 13 年度 富永研究室卒 の 宮澤 敏記 氏にはこの場を借りて心から深く感謝致します。

貴重な御意見、様々な御提案を頂いた AVS ゼミ の皆様に御礼申し上げます。特に、方向性の指針となる考えを多数与えてくれた、清水 直人 君、金田 瑞規 君に 感謝します。

富永研究室の皆様に、御礼申し上げます。特に、斬新な発想で、停滞した進捗に多々風を吹き込んでくれた田邊 集 君に 感謝します。

最後に、私をここまで育てて下さった家族に深く感謝します。

2004 年 3 月 15 日

関連図書

- [1] ISO/IEC 11172-2, International Standard: “Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbits/s,” 1993
- [2] ISO/IEC 13818-2, International Standard: “Information technology- Generic coding of moving pictures and associated audio information:Video,” 1996
- [3] ISO/IEC 14496-2, International Standard: “Information technology-Coding of audio-visual objects,” 2000
- [4] ISO/IEC 10918-1, International Standard: “Digital Compression Coding of Continuous Tone Still,” 1994
- [5] ISO/IEC IS 10918-1, International Standard: “JPEG 2000 image coding system,” 2000
- [6] Maria Kuklisova, “Construction of Non-Uniform Basis Functions For Spline Curves And Surfaces,” In Proc. WSCG2000 Conf., pp. 56-61, 2000
- [7] Fabian W. Meier, Guido M. Schuster and Aggelos K. Katsaggelos, “A mathematical model for shape coding with B-splines,” Signal Processing: Image Communication, Vol. 15, pp. 685-701, 2000
- [8] Ujjaval Y. Desai, Marcelo M. Mizuki, Ichiro Masaki and Berthold, “Edge and Mean Based Image Compression,” M.I.T. Labo. A. I. Memo, No. 1584, Nov. 1996
- [9] Chuang Gu and Murat Kunt, “Very Low Bit-rate Video Coding Using Multi-criterion Segmentation,” In Proc. First IEEE International Conference on Image Processing, ICIP’94, pp. 418-422, vol. II, Nov. 1994
- [10] J. Zaletelj, R. Pecci, F. Spann, A. Hanjalic and R. L. Lagendijk, “Rate Distortion Optimal Contour Compression Using Cubic B-Splines,” In Proc. Signal processing IX, Eusipco-98, Ninth European Signal Processing Conference, pp. 8-11 Sep. 1998
- [11] Wu-Chih Hu and Hsin-Teng Sheu, “Quadratic B-spline for Curve Fitting,” In Proc. Natl. Sci. Council. Roc(A), Vol. 24, No. 5, pp. 373-381, 2000
- [12] 古角 康一, 渡辺 裕, 小林 直樹, “スプライン/DCTハイブリッド符号化方式を用いたアニメーション映像における領域抽出の一検討,” 信学技法, pp . 23-30 , Nov. 1999

- [13] 古角 康一, 渡辺 裕, 小林 直樹, “アニメーション映像における領域抽出の一検討,” 画像符号化シンポジウム (PCSJ99), P-3. 15, Sep. 1999
- [14] 宮澤 敏記, 亀山 涉, 渡辺 裕, 阪谷 徹, 富永 英義, “アニメーション画像符号化の基礎検討,” 通信学会総合大会, pp.66, D-11-66, Mar. 2001
- [15] 宮澤 敏記, “コンテンツオリエンテッド符号化方式とデコーダダウンローダブルアーキテクチャの研究,” 2001 年度 早稲田大学大学院理工学研究科 修士論文, Mar. 2002
- [16] P. Salembier and J. Serra, “Flat Zones Filtering, Connected Operators, and Filters by Reconstruction,” IEEE Transactions on image processing, Vol. 4, No. 8, pp.1153-1160, Aug. 1995
- [17] P. Salembier, F. Marques, M. Pardas, R. Morros, I. Corset, S. Jeannin, L. Bouchard, F. Meyer, and B. Marcotegui, “Segmentation-based video coding system allowing the manipulation of objects,” IEEE Trans. on Circuits and Systems for Video Technology, Vol. 7, No. 1, pp. 60-74, Feb. 1997
- [18] C. A. Christopoulos, W. Philips, A. N. Skodras, J. Cornelis, “Segmented image coding: Techniques and experimental results,” Signal Processing:Image Communication, pp.63-80, Nov. 1997
- [19] 森 浩一, 和田 耕一, 寅市 和男, “関数化図形表現を用いた紙文書のデジタル化,” 情報処理学会研究報告 Vol.99, No.57, pp.17-23, Jul. 1999
- [20] 斉藤 剛, 穂坂 衛, “拡張した 2 次有理 Bezie 曲線の性質とその曲線近似法への応用,” 情報処理学会論文誌, Vol.31, No.01, pp.33-41, Jan. 1990
- [21] U.Ramer, “An iterative procedure for the polygonal approximation of planar curves,” Computer Graphics and Image Processing Vol.1, pp.244-256, 1972
- [22] S.Carlsson, “Sketch Based Coding of Grey Level Images,” *Signal Processing* Vol.15,pp.57-83, 1988
- [23] S.Carlsson, “Image reconstruction from coded edge data,” *Proc. of Premier Colloque Image*, Biarritz, France, pp.57-83, May 1984
- [24] Robert M. Haralick, Stanley R. Sternberg, and Xinhua Zhuang, “Image Analysis Using Mathematical Morphology,” IEEE Trans. Pattern Anal. Machine Intelli., PAMI-9, 4, pp. 532-549, July 1987.
- [25] J. Serra, “Image Analysis and Mathematical Morphology,” Academic Press, London, 1982

- [26] Z. Wang, "Fast algorithms for the discrete W transform and for the discrete Fourier transform", IEEE Trans. Acoust., Speech, and Signal Process., vol. ASSP-32, pp. 803-816, Aug. 1984
- [27] R. Adams, L. Bischof, "Seeded Region Growing," IEEE Trans. on Pattern Analysis and Machine Intelligence, 16(6), pp 641-647, Jun. 1994
- [28] 石村 信道, 橋本 猛, 辻本 修一, 有本 卓, "修正動的計画法による線図形のスプライン近似," 信学論誌, Vol. J68-D, No. 2, pp. 169-176, Feb. 1985
- [29] 石村 信道, 橋本 猛, 有本 卓, 前田 文郎, "平面線図形のスプライン関数近似について," 情処全大, IT83-5, pp. 31-38, 1983
- [30] 山内 康晋, 森下 明, 梅木 直子, 土井 美和子, "落書き風アニメーションシステム," 信学技法, HIP98-32, pp. 1-7, Nov. 1998
- [31] 武田 哲也, 岡田 至弘, "等色線情報に基づくカラー画像記述," 情処全大, 1C-3, Vol. 2, pp. 109-111, 1993
- [32] 山岸 健太郎, 坂上 勝彦, 山本 和彦, 松原 仁, "輪郭図形からの注目候補点の検出方法," 信学論誌 D-II, Vol. J72-D-II, No. 5, pp. 750-759, May 1989
- [33] Ariel Shamir and Ari Rappoport, "Quality Enhancements of Digital Outline Fonts," Comput. & Graphics, Vol. 21, No. 6, pp. 713-725, 1997
- [34] 中嶋 正之, 安居院 猛, 吉井 孝司, 今井 継彦, "地図・図面入力の技術動向," 信学技法, Vol. IE88-37, pp. 1- 7, 1988
- [35] 中嶋 正之, 安居院 猛, 飯塚 久登, "市街地地図に対するパラレル・ベクトル・トレーサを用いたグラフ構造解析," 信学論誌, Vol. J67-D, No. 12, Dec. 1984
- [36] 中嶋 正之, 安居院 猛, 北村 宏行, "2 値画像に対する自動滑縁化彩色処理," 信学論誌, Vol. J70-D, No. 11, Nov. 1987
- [37] 中嶋 正臣, 野中 俊一郎, 三部 靖夫, 中村 太一, "領域分割符号化におけるフレーム構造に関する研究," 信学論誌, Vol. J80-D-II, No. 2, pp. 459-467, Feb. 1997
- [38] 金子 満, 中嶋 正之, "セルアニメタッチ画像生成の3次元CG画像の2次元化アルゴリズム," テレビジョン学会誌, Vol. 49, No. 10, pp. 1288-1295, 1995
- [39] 安居院 猛, 目崎 祐史, 長尾 智晴, 中嶋 正之, "2重の正方形移動法による機械図面からの図面要素の抽出," 信学論誌, Vol. J73-D-II, No. 9, Sep. 1990
- [40] 安居院 猛, 古川 知幸, "計算機による等高線の抽出," 信学技報, PRL79-69, pp.41-47, 1979

- [41] 岩田 耕一, 安居院 猛, “セルアニメーション画像の間の閉領域同定方法に関する研究,” 信学技法, PRL80-9, pp.9-16, Sep. 1980
- [42] 塩原守人, 後藤敏行, 吉田真澄, “3次元構造に基づくアニメーション画像のフレーム間対応,” 信学論誌, Vol. J73-D-II, No. 10, pp. 1687-1695, Oct. 1990
- [43] 亀山 涉, 富永 英義, “ドキュメントアーキテクチャ要素の自動識別に関する研究 -道路地図からの道路構造の抽出-,” 信学技報 IE86-116, pp. 49-56, Mar. 1987
- [44] Jobst Loffler, “Object-based Image Coding for Cooperative 3D Visualization,” In. Proc WSCG2000, pp. 197-203, Feb. 2000
- [45] I. Grinias, G. Tziritas, “A semi-automatic seeded region growing algorithm for video object localization and tracking,” Signal Processing: Image Communication, Vol. 16, pp. 977-986, 2001
- [46] Ronald Alferes and Yuan-Fang Wang, “Geometric and Illumination Invariants for Object Recognition,” IEEE Trans. on Pattern Analysis And Machine Intelligence, Vol. 21, No. 6, June 1999
- [47] Jean Louchet, Michael Boccara, David Crochemore and Xavier Provot, “Building new tools for Synthetic Image Animation by using Evolutionary Techniques,” In Proc. Evolution Artificielle, Sep. 1995
- [48] 市田 浩三, 清野 武, “平面曲線のセグメンテーション,” 信学論誌, Vol. J75-D-II, No. 11, pp. 689-696. Nov. 1975
- [49] 金子 透, 奥平 雅士, “線図形のセグメントチェーン符号化,” 信学論誌, Vol. J65-D, No. 11, pp. 1390-1397. Nov. 1982
- [50] 西川 克彦, 直井 聡, “輪郭表現による高品質文字パターン生成方式,” 信学論誌, Vol. J89-D-II, No. 12, pp. 2023-2031. Dec. 1989
- [51] 福永 邦雄, 浅野 剛, 村田 秀人, 泉 正夫, “エッジ画像グラフを用いた画像の対応,” 信学論誌, Vol. J73-D-II, No. 9, pp. 1512-1519. Sep. 1990
- [52] 栄藤 稔, 伴野 明, 岸野 文郎, “ステレオ輪郭像を入力とした物体の一般化円筒複合体による記述,” 信学論誌, Vol. J73-D-II, No. 9, pp. 1402-1412. Sep. 1990
- [53] 阿部 亨, 阿曾 弘具, 木村 正行, “3次元表面形状による人間の顔の自動識別 -B スプライン曲面の制御点を利用して-,” 信学論誌, Vol. J73-D-II, No. 9, pp. 1477-1484. Sep. 1990
- [54] 木村 和広, 太田 直哉, 金谷 健一, “精密なノイズモデルによるオプティカルフローの検出,” 情処研報, 96-CV-99-6, pp. 37-42, Mar. 1996

- [55] 西原 功, 吉田俊之, 酒井善則, “信頼度に基づくブロック統合を用いた動画像の高精度動きベクトル推定とその応用,” 映像情報メディア学会誌, Vol. 53, No. 1, pp. 148-156, Jan. 1999
- [56] P. Anandan, M. Irani, R. Kumar, and J. Bergen, “Video as an image data source: efficient representations and applications,” In Proc. *IEEE Int'l Conf. Image Processing (ICIP)*, Oct. 1995
- [57] Mohamed Ben, Hadj Rhouma and Hichem Frigui, “Self-Organization of Pulse-Coupled Oscillators with Application to Clustering,” *IEEE Trans. Pattern Analysis And Machine Intelligence*, Vol. 23, No. 2, Feb. 2001
- [58] 木村 彰男, 渡辺 孝志, “アフィン変換に不変な任意図形検出法として拡張された一般化ハフ変換,” 信学論誌, Vol. J-84-D-II, No. 5, pp. 789-798, May 2001
- [59] 牛丸 主税, 長谷山 美紀, 北島 秀夫, “マルチエージェントを用いた複数フレーム線分画像からの同一物体の抽出,” 情処AVM研究会, Vol. 27-12, pp. 67-72, Dec. 1999
- [60] Ronald Alferéz and Yuan-Fang Wang, “Image Database Indexing using a Combination of Invariant Shape and Color Descriptions,” The Second International Conference On Information Fusion, Sunnyvale, CA, July, 1999
- [61] Stefano Berretti, Alverto Del Bimbo and Pietro Pala, “Retrieval by Shape Similarity with Perceptual Distance and Effective Indexing,” *IEEE Trans. On Multimedia*, Vol. 2, No. 4, pp. 225-239, Dec. 2000
- [62] Hae-Kwang Kim and Jong-Deuk Kim, “Region-based shape descriptor invariant to rotation,” *Signal Processing: Image Communication*, Vol. 16, pp. 87-93, 2000
- [63] 本谷 秀堅, 出口 光一郎, “画像のスケールスペース解析にもとづく線図形の抽出,” 情処研報 コンピュータビジョンとイメージメディア, Vol. 105, No. 2, pp. 9-16, May 1997
- [64] 斉藤 文彦, “遺伝的アルゴリズムを用いた画素選択テンプレートによる画像マッチング,” 信学論誌, Vol. J-84-D-II, No. 3, pp. 488-499, Mar. 2001
- [65] Frederuc Juric and Michel Dhome, “Hyperplane Approximation for Template Matching,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 24, No. 7, pp. 996-1000, 2002
- [66] 上田 哲正, 佐藤 真一, 濱田 喬, “凸包を用いた階層的マッチング手法,” 信学論誌, Vol. J-80-D-II, No. 5, pp. 1026-1036, May. 1997

- [67] 田中 弘一, 佐野 睦夫, 大原 秀一, 奥平 雅士, “パラメトリックテンプレート法とその高精度ロバストマッチングへの応用,” 信学論誌, Vol. J-83-D-II, No. 4, pp. 1119-1130, Apr. 2000
- [68] 秦泉寺 久美, 渡辺 裕, 岡田 重樹, 小林 直樹, “MPEG-4 スプライト符号化を用いた超低レート動画像圧縮,” 信学論誌, Vol. J84-D-II, No. 5, pp. 758-768, May 2001
- [69] 小館 亮之, 朱 青, 金 云泰, 浦野 義頼, 富永 英義, “輪ゴムかけの概念に基づく境界線の構造記述と未定義文字処理システムへの応用,” 信学論誌 (D-II), Vol. J84-D-II, No. 1, pp. 1-11, Jan. 2001
- [70] K. Dobashi, A. Kodate, H. Tominaga, “Camera working parameter extraction for constructing video considering camera shake,” In Proc. *IEEE Int’l Conf. on Image Processing (ICIP)*, Oct. 2001
- [71] 境田 慎一, 苗村 昌秀, 金次 保明, “背景差分法と時空間 watershed による領域成長法を併用した動画像オブジェクトの抽出,” 信学論誌 (D-II), Vol. J84-D-II, No. 12, pp. 2541-2555, Dec. 2001
- [72] Bae. Y. J., 高橋 裕樹, 中嶋 正之, “セルアニメーション制作補助のためのキャラクタ動作データベース構築 第1報: Shot 変化の自動検出,” 信学全大, D-12-80, Mar. 2002
- [73] 角 文雄, 中嶋 正之, “2D アニメーション用標準動画データベースシステム,” 信学全大, D-12-81, Mar. 2002

目 次

2-1	アニメーション画像の撮影	7
2-2	アニメーション画像の分類	8
2-3	DCT を用いた静止画像符号化方式の基本構成	9
2-4	モスキートノイズ	10
2-5	モスキートノイズによる品質の劣化	11
3-1	3 次のエルミート補間子	15
3-2	エルミート曲線	16
3-3	ベジェ曲線の例	16
3-4	B-スプライン基底関数の計算ピラミッド	18
3-5	B-スプライン基底関数 (3 次)	18
3-6	3 次の B-スプライン曲線の例	19
3-7	誤差定義及び重み関数の例	23
4-1	先行研究におけるアニメーション画像符号化	31
4-2	領域成長法	32
4-3	エッジの検出	33
4-4	線画の検出	33
4-5	背景領域の誤検出例	34
4-6	提案方式による符号化	36
4-7	フィルタによる平滑化	37
4-8	Connected Operator による領域の結合	38
4-9	Flat-zone の例 (背景領域無しの場合)	38
4-10	Flat-zone の例 (背景領域を含む場合)	39
4-11	Connected Operator によるフィルタの適用	40
4-12	元画像 (dog1)	42
4-13	輝度値への平滑化処理結果 (dog1)	42
4-14	均等色領域抽出結果 (dog1)	42
4-15	線画の抽出 (dog1)	43
4-16	細線化処理を施したもの (dog1)	43
4-17	線画, 均等色領域合成結果 (dog1)	43
4-18	元画像 (dog2)	44

4-19 輝度値への平滑化処理結果 (dog2)	44
4-20 均等色領域抽出結果 (dog2)	44
4-21 線画の抽出 (dog2)	45
4-22 細線化処理を施したもの (dog2)	45
4-23 線画, 均等色領域合成結果 (dog2)	45
4-24 元画像 (anpanman1)	46
4-25 輝度値への平滑化処理結果 (anpanman1)	46
4-26 均等色領域抽出結果 (anpanman1)	46
4-27 線画の抽出 (anpanman1)	47
4-28 細線化処理を施したもの (anpanman1)	47
4-29 線画, 均等色領域合成結果 (anpanman1)	47
4-30 元画像 (anpanman2)	48
4-31 輝度値への平滑化処理結果 (anpanman2)	48
4-32 均等色領域抽出結果 (anpanman2)	48
4-33 線画の抽出 (anpanman2)	50
4-34 細線化処理を施したもの (anpanman2)	50
4-35 線画, 均等色領域合成結果 (anpanman2)	50
5-1 動領域の検出・追跡アルゴリズム	57
5-2 動領域の粗い輪郭線検出	58
5-3 境界画素の境界線長による分類	58
5-4 IRFA における境界線情報の定義	59
5-5 粗い輪郭線の生成	59
5-6 セル画領域の真の輪郭線の抽出	60
5-7 内側からの輪ゴム処理	60
5-8 連続フレーム ('slide1')	61
5-9 差分画像とエッジ画像 ('slide1')	61
5-10 粗い輪郭の抽出 ('slide1')	62
5-11 セル画領域の抽出 ('slide1')	62
5-12 連続フレーム ('slide2')	63
5-13 差分画像とエッジ画像 ('slide2')	63
5-14 粗い輪郭の抽出 ('slide2')	64
5-15 セル画領域の抽出 ('slide2')	64
5-16 連続フレーム ('pan3')	65
5-17 大局的な動きにより粗い輪郭の生成に失敗する例 ('pan3')	65
5-18 大局的な動きを補正した場合の差分画像 ('pan3')	66
5-19 オプティカルフローの抽出 ('pan2')	67
5-20 オプティカルフローの抽出 ('pan3')	67

5-21 オプティカルフローの抽出 ('zoom1')	67
5-22 8方向へのクラスタリング	68
5-23 大局的な動きの補正	69
5-24 大局的な動きがあるショットでのオプティカルフロー ("pan1")	75
5-25 大局的な動きがあるショットでのオプティカルフロー ("pan2")	76
5-26 大局的な動きがあるショットでのオプティカルフロー ("pan3")	77
5-27 大局的な動きがあるショットでのオプティカルフロー ("zoom1")	78
5-28 "pan1" における検出結果	79
5-29 "pan2" における検出結果	80
5-30 "pan3" における検出結果	81

表 目 次

2-1 標準量子化テーブル	9
3-1 moveto コマンド	26
3-2 closepath コマンド	26
3-3 lineto コマンド	27
3-4 quadratic Bezier コマンド	28
4-1 実験条件	41
5-1 実験条件	70
5-2 オブジェクト抽出結果	70

研究業績

	題 名	発表年月	発表掲載誌	連名者
海外				
(1)	A Study on two-layer coding for animation images	2002 年 8 月	IEEE International Conference on Multimedia Expo (ICME) 2002	O. Nakagami T. Miyazawa H. Watanabe H. Tominaga
(2)	A study on content-oriented coding scheme and decoder downloadable system	2002 年 7 月	World Multiconference on Systemics, Cybernetics and Informatics (SCI) 2002	O. Nakagami N. Shimizu T. Miyazawa W. Kameyama H. Watanabe H. Tominaga
国内				
(3)	差分画像を利用したアニメーション映像からのオブジェクト抽出	2003 年 3 月	情報処理学会 AVM 研究会	中神 央二 渡辺 裕 富永 英義
(4)	アニメーション映像における動き情報検出の検討	2002 年 11 月	画像符号化シンポジウム PCSJ2002	中神 央二 渡辺 裕 富永 英義
(5)	二層構造によるアニメーション画像符号化の検討	2002 年 6 月	画像電子学会 第 30 回年次大会	中神 央二 宮澤 敏記 渡辺 裕 富永 英義
(6)	平滑化操作によるアニメーション画像符号化の検討	2002 年 3 月	電子情報通信学会 全国大会, D-11-54	中神 央二 宮澤 敏記 渡辺 裕 富永 英義
(7)	平滑化操作によるアニメーション画像符号化の検討	2002 年 3 月	情報処理学会 AVM 研究会 2002AVM36-11	中神 央二 宮澤 敏記 渡辺 裕 富永 英義

A STUDY ON TWO-LAYER CODING FOR ANIMATION IMAGES

Ouji NAKAGAMI[†], Toshinori MIYAZAWA[‡], Hiroshi WATANABE[†], Hideyoshi TOMINAGA^{†,‡}

[†] Graduate School of GITS
Waseda University
1-3-10, Nishiwaseda, Shinjuku-ku, Tokyo, Japan

[‡] Dept. of Elec. Info. and Comm. Eng.
Waseda University
3-4-1, Okubo, Shinjuku-ku, Tokyo, Japan

ABSTRACT

A coding scheme specifically designed for animation images is proposed. Taking characteristics of animation images into account, lines and homogeneous color regions are extracted from animation images. Lines and contours of the homogeneous color regions are approximated by straight-line and spline functions. We found that smoothing operations are effective to extract homogeneous regions from background images. Connected Filter is used for smoothing operation while keeping edges of the original image correctly. Precise approximation of the contours is also required. For this purpose, we propose to use Dynamic Programming with a feedback algorithm. As a result, an animation image can be represented by two layers; significant points as a base layer, and DCT blocks for an additional layer. For compensation scheme to the loss caused by smoothing operation, DCT is applied to the difference between the original images and lines and homogeneous color regions images. High-quality images can be obtained by adding this differential data to the approximated images.

1. INTRODUCTION

Discrete Cosine Transform (DCT) based coding, such as MPEG and JPEG, has been widely used for image compression. However, DCT-based approach may not be suitable for every kind of images. Some sort of images are often encoded efficiently by taking characteristics of the image content into account (Content-oriented Coding Scheme). As an example of this, we have been studying animation image coding.

Natural images mostly have smooth fluctuations in luminance, thus powers of DCT coefficient concentrate at low frequencies. On the other hand, animation images have many impulsive changes in luminance especially around line drawings, and for this, powers of DCT coefficient disperse from low to high frequencies. This effect makes coding efficiency worse in a DCT-based image coding, especially by causing many mosquito noises near edges.

Therefore, we have been proposing a coding scheme specifically designed for animation images [1]. Animation images are constructed from three components, such as (1) line drawings, (2) homogeneous color regions and (3) background regions (Fig.1). Line drawings, which an animation cartoonist creates, are usually black and serve as contour lines of homogeneous color regions. Homogeneous color regions can be defined as an area where pixel values of luminance and chrominance are almost the same. These regions can be seen on a cartoon character's face for example.

Background regions are similar to natural images since they usually have many colors and details without obvious edges.

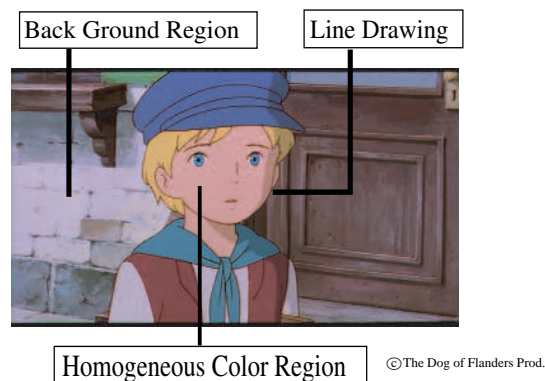


Figure 1: Three components of an animation image

In the method that we propose in this paper, first, line drawings and homogeneous color regions are extracted from an animation images. Then, all lines and contours of homogeneous color regions are approximated by some significant points. An advantage of this coding scheme is that it achieves high compression ratio by allowing some distortion of lines and edges. Computational complexity required for decoding process can be decreased, since only a few points are enough to represent lines. Furthermore, the size of the decoded images can be changed in any resolution by a number same as that of significant points. However, an encoder is estimated as 100 times more complex compared to that of DCT-based approach while a decoder has almost the same order.

2. ANIMATION IMAGE CODING SCHEME

Animation images are coded in two layers; base layer and additional layer. Fig.2 illustrates the flowchart of an animation image coding. Most animation images include many homogeneous color regions. Therefore, we represent animation images with lines and homogeneous color regions as a base layer image, which is constructed from only the significant points extracted from line image and region image. First, line drawings are extracted from an original animation image by using combinations of spatial filters. Next, homogeneous color regions in the original animation image are identified, and then the representative color and the contour lines are obtained. Detected lines, line drawings and

contour lines of homogeneous color regions, are traced to find significant points along the lines. These points are used to approximate the original lines by spline curves.

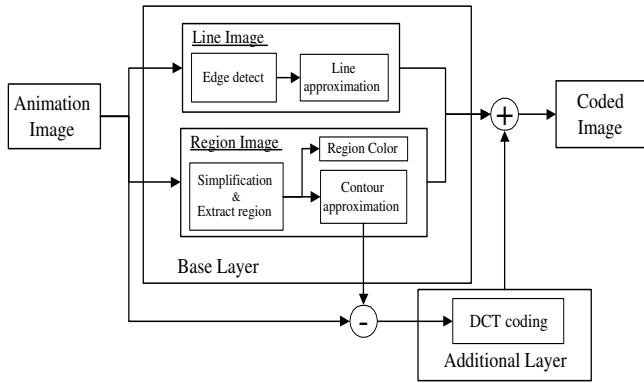


Figure 2: Flowchart of animation image coding

The extraction of a base layer has two issues to be solved. Firstly, in the animation image coding [1], background regions, which were defined as neither back ground regions nor line drawing in an animation images, were coded by DCT since they are close to natural images. A homogeneous color region is detected by region-growing algorithm [2]. However, it fails in some cases because of the mixture of the homogeneous and the background regions. Thus, it is necessary to improve such extraction process. We propose to introduce a joint method of smoothing operator and region-growing algorithm for this purpose. A flowchart of the proposed animation image coding is shown in Fig.2.

Secondly, line approximation algorithm fitting for various shapes to the contours is needed. Ramer algorithm [5] has been used in the conventional approach. It is suitable for straight-line approximation. Thus, significant points determined by the Ramer algorithm are not always proper for spline curve function. Therefore, Dynamic Programming with feedback algorithm is developed.

As an additional layer, a compensation method to the result of smoothing and approximation operation is proposed, and it enables to acquire a high quality coded image. DCT is used for the difference between the original images and the approximated images because luminance signals of the difference contain smooth changes. In addition, each region does not contain edges that are not suitable for DCT-based coding. High-quality images can be obtained by adding this differential data to approximated images that are created by homogeneous color regions and line drawings.

2.1. Region Extraction with Smoothing Operator

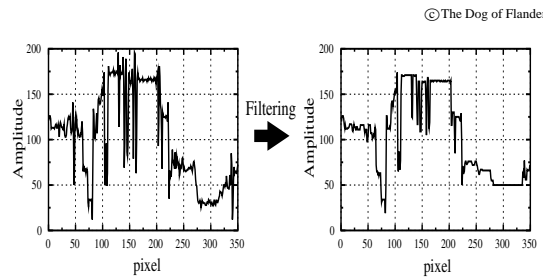
We use smoothing operator and region-growing algorithm jointly to extract homogeneous color regions from background regions. Smoothing operations by traditional linear filter loses edges in original images, that they are not suitable for animation images with many edges. Connected Filter [3], which is used in this method, is one of smoothing filters for the simplification of images. Since this filter keeps high frequency components unlike linear low-pass filters, the edges in the filtered image are kept. Fig.3 shows the

result of the smoothing by Connected Filter. Then, region-growing algorithm extracts homogeneous color regions more correctly than the conventional approach.

Fig.4 and Fig.5 show the process of the base layer coding. Fig.4(a) (“dog1”), which does not include background regions, indicate that the edges in the image are kept in spite of smoothing operation by Connected Filter. Fig.5(a) (“dog2”) indicates that the homogeneous color regions are extracted by region-growing algorithm from simplified background regions.



(a) the filtered image



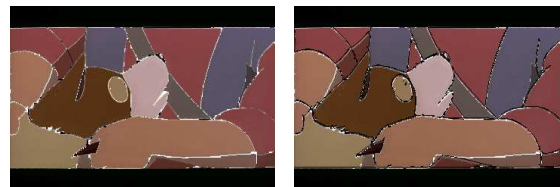
(b) the horizontal scan of the filtered image

Figure 3: The result of Connected Filter



(a) original image

(b) smoothing result



(c) extraction of homogeneous region

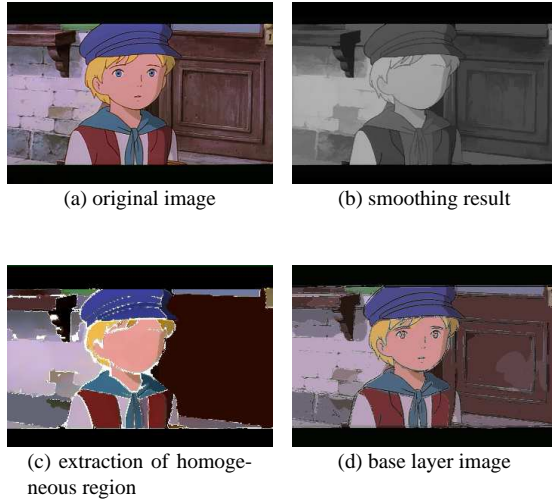
(d) base layer image

©The Dog of Flanders Prod.

Figure 4: Extraction of the base layer for the image with no background regions (“dog01”)

2.2. Dynamic Programming with feedback algorithm

Dynamic Programming examines all points on the curve since they may be chosen as significant points. It deter-



© The Dog of Flanders Prod.

Figure 5: Extraction of the base layer for the image including complexity background regions (“dog02”)

mines a set of the significant points that minimize the total error between original and approximated line. Although this method requires high computational complexity, the optimized approximation can be obtained.

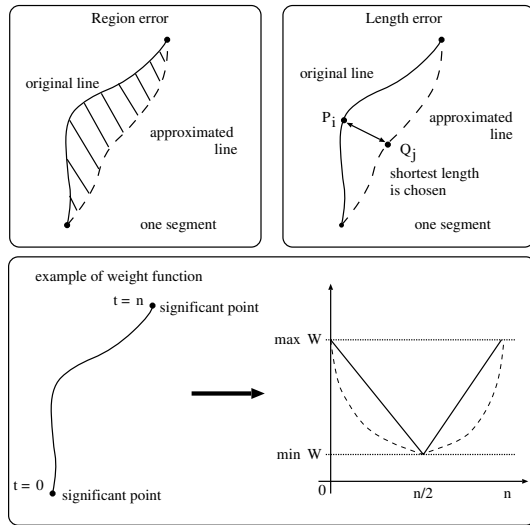


Figure 6: Estimation of the approximation error

There are two ways of estimating the total error in the conventional study[5], [6]; by region error algorithm and by minimizing the length error algorithm (Fig.6). The former estimates an error area between an original line and an approximation line. The latter does a summation of a length from points on an original line to the nearest points on an approximation line. The equation in the latter algorithm enables each approximation points to have a weight based on the characteristic of original lines (Eqn.(1)).

$$E = \frac{1}{l} \sum_{i=1}^l w_i |P_i - Q_j|^2 \quad (1)$$

where, E is an approximation error, l is the number of contour points, and w_i is the functional weight about point i on an original line. P_i are the coordinates of point i , and Q_j are the coordinates of point j , which is the nearest from point i . The set of significant points is calculated by the following steps.

1. Choose several significant points at high curvature points on original line.
2. Divide an original line into segments between significant points. The value of E is estimated by normalization of the length. Therefore, if the segment length is too long, DP algorithm cannot acquire a precise approximation result.
3. Check the value of E for each segment. If the error is larger than a threshold T_e , try other set of significant points in the segment.
4. Determine the points which minimize E of each segment as the significant points.

There are various kinds of contours in animation images. Since DP algorithm requires the threshold T_e to determine the best set of significant points, we must set suitable T_e for each contours. If the threshold is too small, many determined significant points concentrate at high curvature points on an original line. On the contrary, a large T_e increases the approximation error.

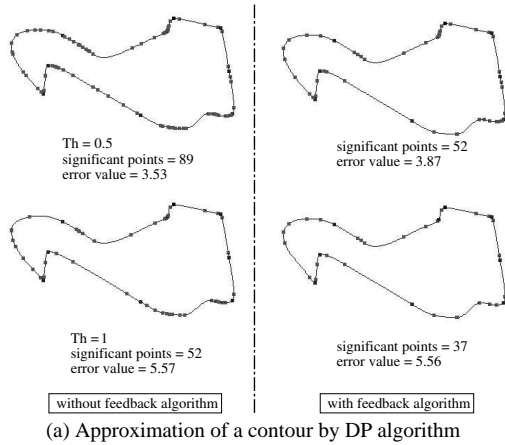
Therefore, we propose feedback algorithm to eliminate the useless significant points. First, a sufficient small T_e is set, and DP acquires enough significant points. Next, DP is applied again to eliminate significant points that do not reduce approximation error. Fig.7 shows the result of DP with and without feedback algorithm. Fig.7(a) illustrates the relation between the number of significant points and approximation error. DP with feedback algorithm enables more precise approximation in spite of the small number of significant points than that of conventional method (Fig.7(b)).

2.3. DCT Coding of Differential Image

We propose a compensational method to the result of smoothing and approximation operation as an additional layer. First, the difference between each approximated homogeneous color region and the original image is calculated. It is divided into 8×8 pixel blocks and coded by DCT. Some DCT blocks include edges when they locate on the contours of homogeneous color regions. To avoid mosquito noise caused by DCT coding, dummy data is padded to pixels not existing on homogeneous color regions (Fig.8).

Connected Filter is effective for not giving influence to the homogeneous color regions. The differential value is calculated at each point in the DCT block using Eqn.(2). D is the sum of $D(i, j)$ of the block. A block on the background regions has a bigger D than that homogeneous color regions because of the smoothing operation. Therefore, it is effective to prioritize the processing order of block according to the D value.

$$D(i, j) = P_o(i, j) - P_h(i, j) \quad (2)$$



(a) Approximation of a contour by DP algorithm

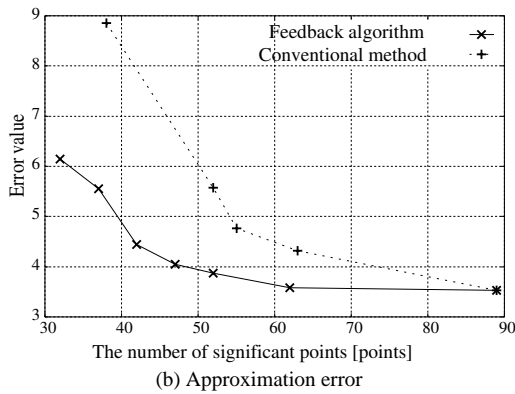


Figure 7: The result of DP with the feedback algorithm

$$D = \sum_{i=0}^7 \sum_{j=0}^7 D(i, j) \quad (3)$$

Fig.9 shows the effect of the additional layer. In the conventional method, the regions that can not be extracted as homogeneous color regions or line drawings were encoded by DCT. The homogeneous color regions detected in failure caused much error. On the contrary, the proposed method enables to control the number of DCT blocks to code. The result of Fig.9(b) shows that the additional layer is necessary for animation images including complicated background regions. When noisy image is input, noise component is once removed by Connected Filter, then it appears

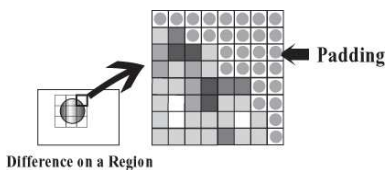


Figure 8: Padding dummy data

again in a differential image.

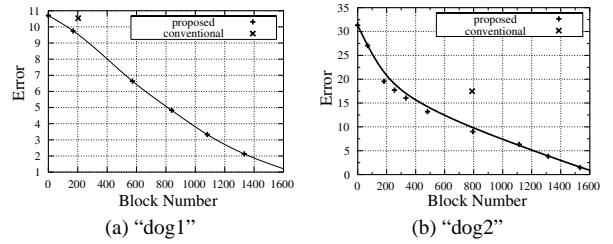


Figure 9: The effect of the additional layer

3. CONCLUSION

In this paper, we proposed the two-layer coding scheme designed for animation images. On the base layer, an animation image can be encoded at low bit rate since line drawings and contours of each homogeneous color region is expressed by significant points used for the approximation of the contour line. DP with feedback algorithm enables precise approximation for the contours by a controllable number of significant points. The lost data caused by smoothing operation is compensated by any number of DCT blocks on the additional layer. This enables smaller coding error than the conventional method. Currently, the proposed coding scheme is only designed for still images. Thus, this scheme can be used for the Intra frame coding of video coding.

4. REFERENCES

- [1] T. Miyazawa, W. Kameyama, H. Watanabe, T. Sakatani and H. Tominaga, "Animation Image Coding," in *Proc. World Multiconference on Systemics, Cybernetics and Informatics*, Vol.XIII, pp.404-409, July 2001.
- [2] R. Adams and L. Bischof, "Seeded Region Growing," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(6), pp 641-647, Jun. 1994.
- [3] P. Salembier and J. Serra, "Flat Zones Filtering, Connected Operators, and Filters by Reconstruction," *IEEE Transactions on image processing*, Vol.4, No.8, pp.1153-1160, Aug. 1995.
- [4] C. A. Christopoulos, W. Philips, A. N. Skodras and J. Cornelis, "Segmented image coding: Techniques and experimental results," *Signal Processing: Image Communication*, Vol.11, pp.63-80, Nov. 1997.
- [5] U. Ramer "An iterative procedure for the polygonal approximation of planar curves" *Computer Graphics and Image Processing*, Vol.1, pp.244-256, 1972.
- [6] F. Lu, E. E. Milios, "Optimal Spline Fitting to Planar Shape," *Signal Processing*, vol.37, pp.129-140, 1994.

A STUDY ON CONTENT-ORIENTED CODING SCHEME AND DECODER DOWNLOADABLE SYSTEM

Ouji NAKAGAMI[†], *Naoto SHIMIZU*[†], *Toshinori MIYAZAWA*[‡],
Wataru KAMEYAMA[†], Hiroshi WATANABE[†], Hideyoshi TOMINAGA^{†,‡}

[†]Waseda Univ.
Graduate School of GITS
1-3-10, Nishiwaseda, Shinjuku-ku, Tokyo, Japan

[‡]Waseda Univ.
Dept. of Elec. , Info. and Comm. Eng.
3-4-1, Okubo, Shinjuku-ku, Tokyo, Japan

1. INTRODUCTION

Content-oriented coding is the one that takes the characteristics of the image content into account. Natural and synthetic images have different characteristics. However, JPEG/ MPEG employs DCT regardless of the image content. Thus, their coding schemes are not optimized for synthetic images; such as animation images and computer graphics. As a promising application of the content-oriented coding, we have been studying animation image coding that is mainly based on line and region extraction.

Currently, a fixed coding algorithm has been used for all kind of images in the broadcasting application. An approach to switch coding algorithm for image contents has a chance to provide better coding efficiency as a whole than the conventional one. Thus, it is important to establish a certain mechanism distributing different decoders depending on the contents for real time applications.

In this paper, we propose a scheme to improve coding efficiency of animation images and mechanism to download necessary decoders in real time.

Keywords: Image coding, Animation image, Smoothing operation, Decoder downloadable, Multimedia system, Middleware

2. ANIMATION IMAGE CODING

2.1. Coding scheme

Animation images have many impulsive changes in the luminance signal around line drawings. At the same time, many homogeneous color regions also exist. When signals of an animation image are transformed to frequency domain using DCT,

the energy of DCT coefficients disperses from low to high frequency. In order to achieve high compression ratio, many coefficients are regarded as zero by quantization. Thus, undesired noises often appear around line drawings by DCT-based image coding.

Therefore, we have been proposing a coding scheme specifically designed for animation images. In the animation image coding method [1], line drawings and homogeneous color regions are extracted from an image. Then, all lines and contours of homogeneous color regions are approximated by some significant points. An advantage of this coding scheme is to achieve high compression ratio by allowing some distortion of lines and edges. Computational complexity required for decoding process can be decreased since few points are enough to represent lines. Furthermore, size of decoded images can be changed in any resolution by the same number of significant points.

However, the conventional method cannot encode animation images having complicate background regions efficiently. In addition, high approximation accuracy to the contours is required to reduce the encoding error. Thus, more robust coding scheme to overcome these problems is proposed in the following.

2.2. Vector-DCT Hybrid Coding

In the animation image coding [1], the rest of line drawings and homogeneous color regions was coded by DCT since background regions are close to natural images. A homogeneous color region is detected by Region-growing algorithm [2]. However, it fails in some cases because of the mixture of the homogeneous regions and the background. Thus, it is necessary to improve such extraction process.

We propose a smoothing operator and region-growing algorithm jointly for this purpose. A flowchart of the proposed animation image coding is shown in Fig.1. Animation images are coded in two layers; Base layer and additional layer. Base layer is constructed from the significant points extracted from line image and region image. Connected fil-

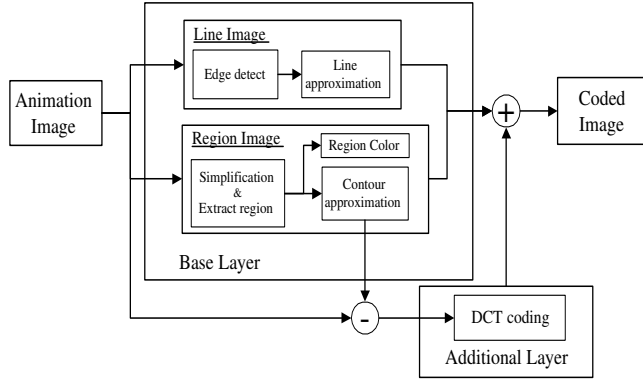


Figure 1: Flowchart of animation image coding

ter [3], which is used in this method, is one of smoothing filters for simplification of images. Since this filter does not remove high frequency components unlike linear low-pass filters, the sharpness of edges in the filtered image is kept. Then, region-growing algorithm can extract homogeneous color regions from whole part of the image.

At this stage, the animation image is represented by only lines and homogeneous color regions, and the lines and the contours of the regions are approximated by significant points.

A compensation method for the result of smoothing and approximation operation is needed as the additional layer to acquire a high quality coded image.

First, the difference between each approximated homogeneous color region and an original image is calculated. It is divided into 8×8 pixels blocks and coded by DCT. Some DCT blocks include edges when they locate on the contours of homogeneous color regions. For avoiding mosquito noise caused by DCT coding, dummy data is padded to the pixels. The sum of the difference value in the each blocks is calculated by Eq.(1)

$$D(i, j) = P_o(i, j) - P_h(i, j) \quad (1)$$

where, $P_o(i, j)$ and $P_h(i, j)$ is the value of the original image and extracted homogeneous color region. In Eq.(2), D_s is the sum of $D(i, j)$ of the

block.

$$D_s = \sum_{i=0}^7 \sum_{j=0}^7 D(i, j) \quad (2)$$

At the block on the background regions, D is bigger than the one on the homogeneous color regions because of the smoothing operation. Therefore, it is suitable to prioritize the block in order of D value in encoding.

Fig.2 shows the effect of the additional layer. In the conventional method, the regions that cannot be extracted as homogeneous color regions or line drawings were encoded by DCT. The homogeneous color regions detected in failure caused much error. On the contrary, the proposed method enables to control the number of coding DCT blocks. The result of Fig.2 shows that the necessity of the additional layer when animation images include complicated background regions.

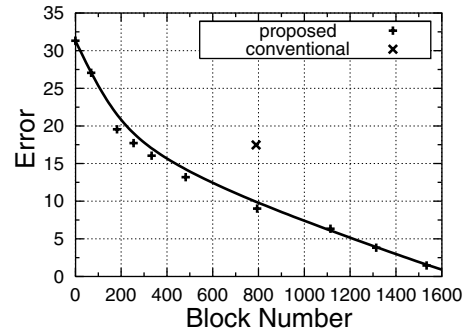


Figure 2: The effect of additional layer

2.3. Dynamic Programming with feedback algorithm

According to region-based image coding [3][4], all parts of an image should be regarded as the set of homogeneous color regions. This approach is more suitable for animation image coding. In animation images, homogeneous color region is usually larger than that of natural images. Therefore, its contours are efficiently vectorized by extracting the significant points.

Dynamic Programming (DP) algorithm is effective for fitting various shapes to contours of the regions [5]. Ramer algorithm [6] has been used in the conventional approach. It was developed for straight-line approximation. Thus, significant

points determined by the Ramer algorithm are not always proper for spline curve function.

Dynamic Programming can examine all points on the curve since they may be chosen as significant points. It can provide the set of significant points that minimize the total error between original and approximated line. Although this method requires high computational complexity, the optimized approximation can be obtained.

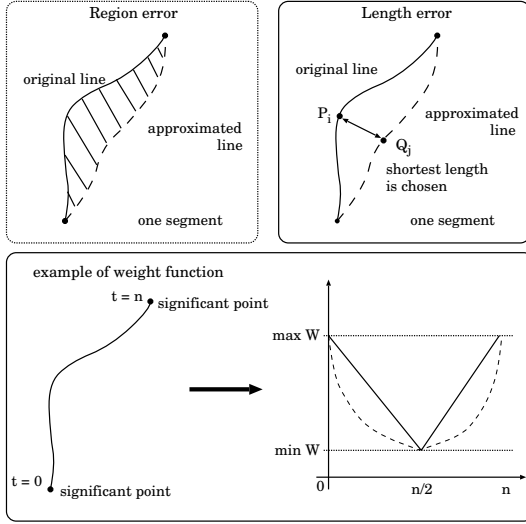


Figure 3: estimation of the approximation error

There are two ways of estimating the total error in the conventional study; region and length error algorithm. They are shown in Fig.3. The former estimates an error area between an original line and an approximation line, and the latter calculates a sum of a length from points on an original line to the nearest points on an approximation line. The latter can add weight by

$$E = \frac{1}{l} \sum_{i=1}^l w_i |P_i - Q_i|^2 \quad (3)$$

where, E is an approximation error, l is the number of contour points, and w_i is the functional weight about point i on an original line. P_i are the coordinates of point i , and Q_j are the coordinates of point j , which is the nearest from point i . The set of significant points is calculated by the following steps.

1. choose several points as significant ones at high curvature points on original line.
2. divide an original line into some segments between significant points. The value of E

is estimated by normalization of the length. Therefore, if the segment length is too longer, DP algorithm can not acquire a precise approximation result.

3. check the value of E for each segment. If larger the T_e of the error, try other set of significant points on the segment.
4. determine the points which minimize E of each segments as the significant points.

Since DP algorithm requires the error T_e to determine the best set of significant points, we should set suitable T_e for each contours. If T_e is too small, many determined significant points concentrate at high curvature points on an original line. On the other hand, the large T_e increases the approximation error. Therefore, we propose feedback algorithm to eliminate the useless significant points. First, sufficiently small T_e is set, and DP acquires enough significant points. Next, DP is applied to eliminate significant points that do not reduce approximation error. Fig.4 shows the result of DP algorithm with and without feedback algorithm. Fig.4(a) illustrates the relation between the number of significant points and approximation error. DP with feedback algorithm enables more precise approximation in spite of the small number of significant points than the conventional method (Fig.4(b)).

3. DECODER DOWNLOADABLE SYSTEM

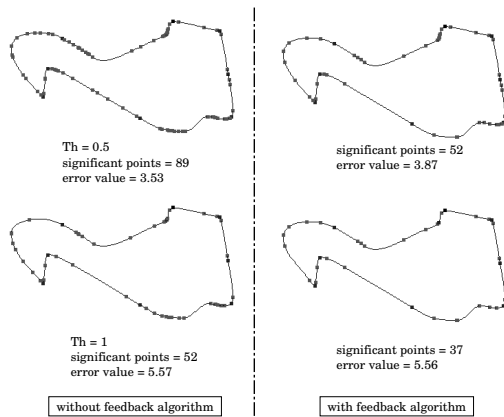
3.1. Requirements

Decoder downloadable system is a system architecture to distribute various decoders. It is necessary not only to develop image coding algorithms but also to prevail decoders implementing them.

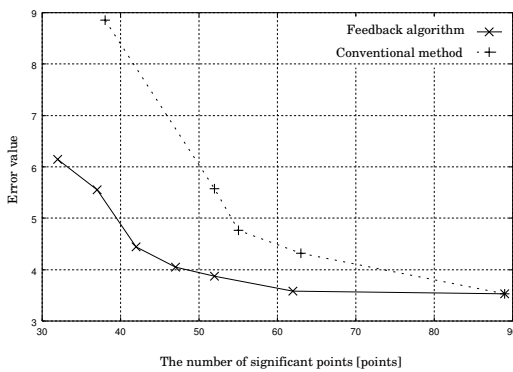
Multimedia systems on the Internet, such as Microsoft Media Player, RealPlayer and so on, provide functions to play back new types of media by installing appropriate decoder software, which is called "Plug-in," via network when required. However, Plug-in has problems as follows:

1. Time delay

The time delay happens when a new content starts and a client does not have the suitable decoder. This interrupts seamless playback of contents and becomes a serious problem especially in case of TV broadcasting.



(a) approximation of a contour by DP algorithm



(b) approximation error

Figure 4: the result of DP with feedback algorithm

2. Installation problem

Plug-in requires viewers of contents a little knowledge about image coding. TV broadcasting has been universal service, and Internet streaming will be the same situation, too. The system in the universal service should not expect clients having service installation knowledge. This point is very important for applying Content-oriented coding scheme, because it will require more knowledge about image coding.

3. Closed specification

The specification of Plug-in and the way to transfer Plug-in are closed in each multimedia software vendor. There is no compatibility among the software vendors. Developers of multimedia system are urged to create the

architecture to download the decoders.

In case, we will apply Content-oriented coding to not only Internet streaming but also TV broadcasting, there is no integrated system to distribute them for both TV broadcasting and Internet streaming. We implement Decoder downloadable system to achieve such a system and to provide developers of multimedia system the functions to download the decoders.

3.2. Proposed system architecture

In this section, we propose the system architecture which enables to download decoder dynamically and to play back without delay before a new content starts.

The proposed client's system architecture is shown in Fig.5.

On a server side, there are Content server, Decoder server and Information server. Content server stores actual bitstreams of the contents. Decoder server stores the binary codes of the decoders. Information server stores the information shown in Fig.6. Clients can seek where the suitable decoders for each content is stored. It is done by exchanging this information between clients and Information server, and clients can download decoders dynamically. We implement the client system by using Java. This is because the proposed system should work in the various environments.

We develop the part of media processing with JMF (Java Media Framework) [7] as the first step. A question is often raised about the processing performance of Java. Therefore, we designed the system in the such a way that the module for media processing depending on JMF can be detached from other parts entirely.

In this implementation, CORBA (Common Object Request Broker) [8] is used to define its interface between server and client being independent from the hardware specification, because of the independency of CORBA GIOP (General Inter-ORB Protocol) from any transport protocol.

The procedure in the system is executed in order the following:

- (1) The scheduler in the client system starts to parse the information shown in Fig.6. The system confirms whether the decoder exists in the client's side or not.

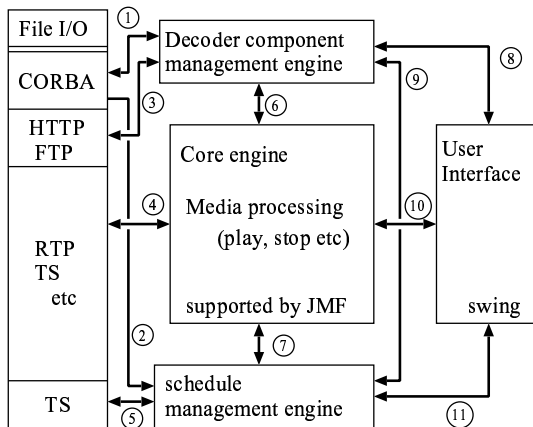


Figure 5: Decoder downloadable system architecture

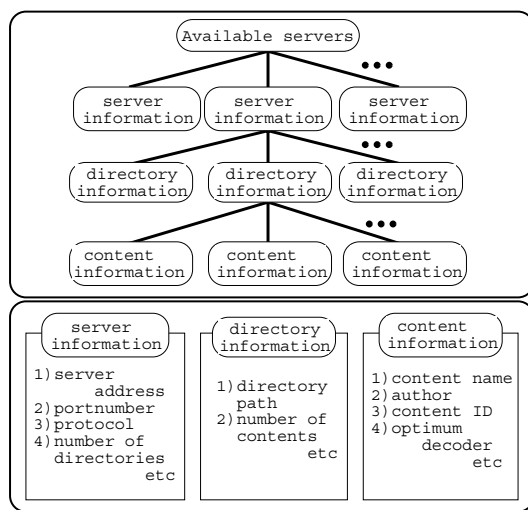


Figure 6: Content information

- (2) Sending the information about the content that the client does not have a suitable decoder to Information server.
- (3) Information server starts to seek where the suitable decoder is stored using the information sent in (2). It returns this seeking information to the client.
- (4) Decoder information has the IP address of Decoder server, the filename of the decoder and so forth. The scheduler of the client decides when to start downloading the decoder from Decoder server by that information.

The role of each part in the system is as follows:

- **Core engine**
This part provides the implementation of media play, the support of trick mode and so forth. The decoders are actually installed in this part.
 - **Decoder component management engine**
This part manages the decoders in the client's system, queries about what decoder is needed, download the decoders, deletes the decoders in the stock.
 - **Schedule management engine**
This part controls when to start media play, downloading the decoders, deleting the decoders. It indicates to each part that provides actual above functions.
 - **User interface**
This part is the interface for user input.
 - **File I/O**
This part selects each protocol in accordance with the purposes. 1) CORBA : sending and receiving Content Information, 2) HTTP or FTP : downloading the binary data of the decoders, 3) RTP (Real-time Transport Protocol) or TS (Mpeg-2 Transport Stream) : receiving media data
- Each interface in the system is described as follows:
- interface 1:** Receiving the information about the decoders, sending the information of contents.
 - interface 2:** Receiving the information about the contents.
 - interface 3:** Notification of decoder information, download binary data of decoders.
 - interface 4:** Download media data of contents.
 - interface 5:** Download EPG (Electronic Program Guide) data that is used by the client's scheduler.
 - interface 6:** Confirmation for existence of decoders, provision of decoder modules to Core Engine.
 - interface 7:** Media play, indication when decoder's memory load starts, providing the information about what state the memory load is.
 - interface 8:** Indication whether decoder is added or deleted to the system by the users.

interface 9: Providing the information about the contents, indication when downloading the decoder and deleting the decoder in the client system start, confirmation for existence of decoders.

interface 10: Acceptance of the input information by the users.

interface 11: Indicating the information of contents that can be played back, the choice of the contents by the users.

The ability of scheduling for downloading decoders in this system is evaluated in case of Pull Service [9]. The result shows the optimum number of downloading decoders that achieves the minimum time delay is given by the probability that the user switch the content, such as channel hopping. However, this probability may obey a specific distribution. We take an approach to leave place for implementing the function that this distribution is given by storing the history of changing the contents to the scheduler. Furthermore we will implement the remaining part of this system for Push Service, and evaluate it.

4. CONCLUSION

We proposed the animation image coding and the Decoder downloadable system which puts this coding scheme to practical use.

Animation image coding adopts two-layer coding scheme. On the base layer, an animation image can be encoded at low bit rate since line drawings and contours of each homogeneous color region is expressed by few significant points approximating the contour line. DP with feedback algorithm enables precise approximation for the contours by the controllable number of significant points. The lost data caused by smoothing operation is compensated by any number of DCT blocks on additional layer. This enables smaller coding error than the conventional method. Our proposed coding scheme is only designed for still images. Thus, this scheme can be used for the Intra frame coding of video coding.

Decoder downloadable system is composed of Content, Decoder and Information server. This system enables dynamic decoder downloading from Decoder server by negotiation between clients and Information server. Therefore, clients can decode the contents without the knowledge about image

coding scheme and can play back media contents seamlessly. This system, developed by Java and CORBA, is superior for the reason of portability. It will be applied to TV broadcasting as well as Internet streaming.

5. REFERENCES

- [1] T. Miyazawa, W. Kameyama, H. Watanabe, T. Sakatani and H. Tominaga, "Animation Image Coding," *World Multiconference on Systemics, Cybernetics and Informatics (SCI) 2001, Proc. Vol.XIII*, pp.404-409, July 2001.
- [2] R. Adams, L. Bischof, "Seeded Region Growing," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(6), pp 641-647, June 1994.
- [3] P. Salembier and J. Serra, "Flat Zones Filtering, Connected Operators, and Filters by Reconstruction," *IEEE Transactions on image processing*, Vol.4, No.8, pp.1153-1160, Aug. 1995.
- [4] C. A. Christopoulos, W. Philips, A. N. Skodras, J. Cornelis, "Segmented image coding: Techniques and experimental results," *Signal Processing:Image Communication*, pp.63-80, Nov. 1997.
- [5] F. Lu, E. E. Milios, "Optimal Spline Fitting to Planar Shape," *Signal Processing*, vol.37 pp.129-140, 1994.
- [6] U. Ramer, "An iterative procedure for the polygonal approximation of planar curves," *Computer Graphics and Image Processing Vol.1*, pp.244-256, 1972.
- [7] Sun Microsystems, "Java™ Media Framework API," <http://java.sun.com/products/java-media/jmf/>
- [8] Object Management Group, "CORBA/IIOP Specifications," <http://www.omg.org/technology/documents/>
- [9] T. Miyazawa, N. Shimizu, W. Kameyama, H. Watanabe, H. Tominaga, "A study on Architecture of Decoder Downloadable System for pull service," *PCSJ2001*, pp.27-28, Nov. 2001.

差分画像を利用した アニメーション映像からのオブジェクト抽出

中神 央二[†] 渡辺 裕[†] 富永 英義[†]

[†] 早稲田大学 大学院 国際情報通信研究科
〒 169-0051 東京都新宿区西早稲田 1-3-10

E-mail: †{ouji,tominaga}@tom.comm.waseda.ac.jp, †hiroshi@giti.waseda.ac.jp

あらまし 本稿では、アニメーション映像における動オブジェクト抽出の検討を行う。具体的には、セル画により表現される動き情報を利用する。アニメーション映像中の動きには、その制作手法から、スライド、リピート、ローリングなどの表現手法が用いられる。これら動作を検出するためのアルゴリズムとして、フレーム間の差分画像に基づいた方法を提案する。特に、映像中に大局的な動きがある場合について、その検出、補正アルゴリズムを結果とともに示す。実験の結果、大局的な動きを補正して差分画像を生成することで、精度良くオブジェクトを抽出できることを確認した。

キーワード オブジェクト抽出, 差分画像, アニメーション映像

Object extraction using differential images from animation images

Ouji NAKAGAMI[†], Hiroshi WATANABE[†], and Hideyoshi TOMINAGA[†]

[†] Graduate School of Global Information and Telecommunication Studies,
Waseda University
Nishiwaseda 1-3-10, Shinjuku-ku, Tokyo, 169-0051 Japan

E-mail: †{ouji,tominaga}@tom.comm.waseda.ac.jp, †hiroshi@giti.waseda.ac.jp

Abstract In this paper, we propose an object extraction method from cartoon films. This method utilizes motions represented by celluloid images. In animation, some sheets of celluloid represent frames of actions. They are called “slide,” “repeat,” “rolling,” and so on. Proposed algorithm detects such actions by using differential images between continuous frames. When background images are moving, motion compensation is necessary to extract objects. Therefore, algorithms which detects and corrects background motion are also proposed. The experimental result shows effectiveness of this method.

Key words object extraction, differential image, animation movie

1 はじめに

我々は、対象となるコンテンツに最適な符号化方式を適用するという、コンテンツオリエンテッド符号化の概念を提唱している [1], [2]。なかでも、アニメーション画像の性質に着目し、静止画像に対する符号化の検討を行ってきた。これは、画像の形状情報に着目し、領域単位で符号化を行うものである。今回は、この符号化方式を動画画像へ応用するために、アニメーション映像中に含まれる、セル画で描かれた領域の抽出手法について考察する。

ここで対象とするのは、いわゆるセルアニメーションと呼ばれるものである。これらは、セル画と呼ばれる透明なシートの上に、主人公などのキャラクターを描いて制作される。背景部は、背景画として1枚の大きな絵を用意しておき、キャラクターの描かれているセル画を、背景画の上に重ねて撮影する。

一般に、彩色行程の煩雑さから、同一のセル画が何度も使用されることが多い。TV 放映されるアニメーション番組において、実際に作成されるセル画の枚数は、すべて異なるコマとして計算した場合の10分の1程度と言われている。したがって、映像中のどの部分が同一セル画によるものかを判定することができれば、符号化の際の情報量を減らすことが可能になる。

本稿では、その前段階として、セル画により表現されたオブジェクトを抽出する手法について、提案アルゴリズムと実験結果を述べる。

2 アニメーション映像における動きの分類

セル画の部分は、オブジェクトの動きを表現するために、1コマずつ僅かに変化させて、フィルムに撮影される。この際、同一、あるいは数枚のセル画の繰り返しによりキャラクターの移動等を表現することが多い。これには、様々な手法が用いられるが、最も基本的なものとして、次の3つが挙げられる。

スライド キャラクターの平行移動のこと。隣合うフレームにおいて、セル画を上下左右にわずかにずらして撮影することで表現される。この時、拡大・縮小、回転を加えることもあり得る。

リピート 限られた複数枚のセル画を繰り返し用いることで、動きを表現する。瞬きのシーンなど。

ローリング セル画のスライド等で一定の範囲を揺れるように反復移動させる方法。馬車の車輪など。

符号化への応用を考えた際、情報量の縮体のために最も効果が高いと考えられるのは、スライドとリピートの動きである。そこで、これらの動きに特化した検出、判定アルゴリズムが必要となる。なお、ローリングに関しては、スライドとリピートの組み合わせと考えることができる。

3 差分画像を利用したオブジェクト抽出手法

先に述べたアニメ特有の表現手法から、同一のセル画によるオブジェクトが、シーケンス内の別のフレームにおいても登場する可能性は高い。そこで、検索のためのテンプレートとして、オブジェクトを抽出する必要があ

る。これには、オブジェクトの外形を精度良く抽出できる手法が必要になる。

まず、提案アルゴリズムのフローチャートを、図1に示す。前提として、動きを検出する際には、予めショットの単位へ分割されたシーケンスを対象とする。これは、シーン分割を行う自然画の手法が、精度良くアニメーション映像に適用できるという [13] の文献に基づいている。

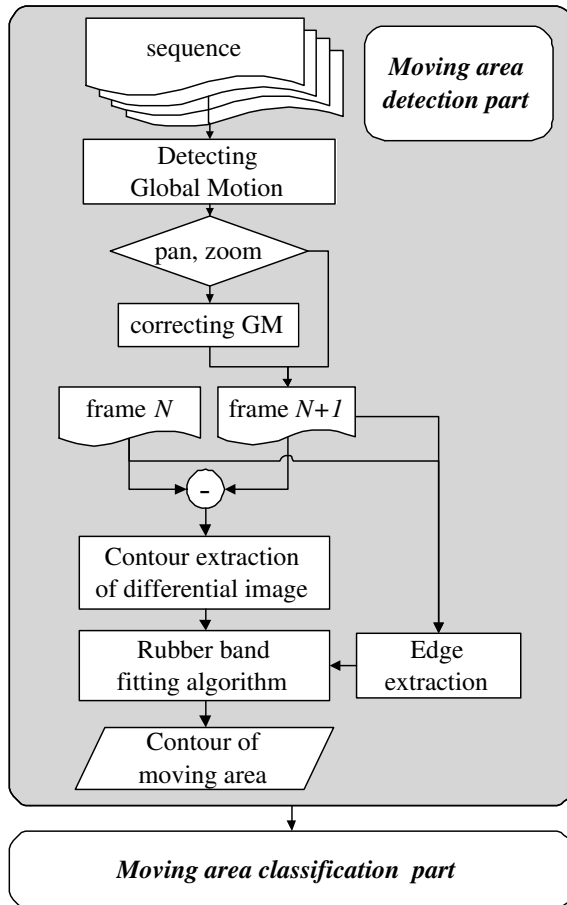


図1 動領域の検出・追跡アルゴリズム

3.1 動領域の抽出

アニメーション映像における動領域を検出するには、フレーム間の差分画像を利用する。

自然動画画像を対象としたオブジェクト抽出には、動きベクトルなどを利用する手法 [7] が提案されている。しかし、アニメーション映像の場合、自然画像と異なり、平坦な色遣いの領域が多く含まれる。そのため、正確なオブジェクト抽出に用いるのは難しい。また、予め背景画像を作成する手法も提案されているが [12]、画面中に占めるオブジェクトの割合が小さく、かつ大きな移動がある場合にしか適さない。これではアニメの場合、先に挙げたスライド以外の動きの検出に用いることが難しい。

一方で、アニメーション映像の特徴として、同一の背景領域を連続するフレームで用いることが多い。そのため、差分画像を生成することで完全に背景を消去することが可能である。隣接フレーム間で、差分を絶対値として扱い画像を生成した場合、変化した部分の領域が検出

される(図2参照)。その輪郭線は、前後フレームにおける各領域の輪郭線を併合せたものとなる。この粗い輪郭線情報を利用して、各フレームにおいて対応する領域の、真の輪郭線を抽出する。これには、元画像のエッジ情報と、輪ゴムのアルゴリズム [10] を用いる。

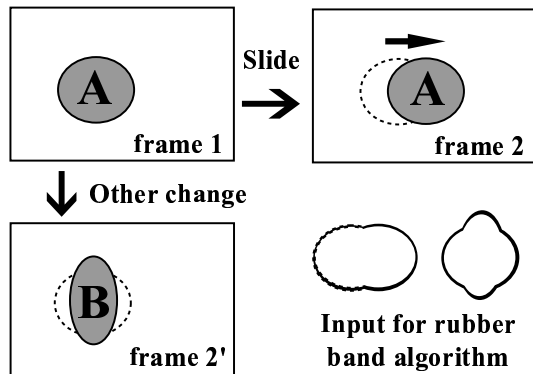


図2 動領域の粗い輪郭線検出

輪ゴムのアルゴリズムは、視覚的な凹凸情報を抽出してオブジェクト形状を段階的に表現する手法である。境界線構造の解析方式として、オブジェクトの回転・サイズ・微小ノイズに高い耐性を有する。

これには、まず概念的にオブジェクトに輪ゴムをかける。そして、輪ゴムとオブジェクトとの接触/非接触関係を、画素の反転処理により抽出する。この反転処理の操作を反復することで、オブジェクトの全境界線特徴の抽出を行うものである。

今回は、輪ゴムの始点を、差分画像より検出した粗い輪郭線情報とし、その対象を、元画像におけるエッジとする。アニメーション画像にはたくさんのエッジ成分が含まれるが、これによりそのフレームにおいて変化している領域の輪郭線のみを検出することができる。本方式の利点として、処理量が少ないこと、ピクセルベースでの抽出が可能になることが挙げられる。

ただし、元の領域の一部のみが変形する場合には、輪ゴムは内側からかける必要がある。すなわち、 A が、 $A' \supset A$ または、 $A' \subset A$ になる場合である。このとき、差分の輪郭線から、外側にたどることにより、前者では後フレームの A' 、後者では前フレームの A の領域を検出することができる。

最後に、検出した輪郭線を用いて、元画像から後段のマッチングに用いるテンプレート画像を切り出す。

3.2 大局的な動きへの対応

アニメーション映像においても、背景画像を動かした場合、自然画におけるカメラモーションのような大局的な動きが発生する。前項で述べた動領域の抽出手法では、ショット中にこの大局的な動きが存在する場合、領域の抽出に失敗するという問題がある。単純な差分画像の生成では、背景領域を消去することが不可能になるためである。

そこで、ショットにおける大局的な動きを判別し、次にそれを補正することが必要となる。前者にはオプティカルフローを基盤とする手法、後者には全探索による手法を用いた。オプティカルフローのみでは、ピクセルレ

ベルで大局的な動きを決定することが難しいこと、逆に全探索のみではショットにおけるパンやズームといった判定を行うことができず、正確な対応点探索を行うことができないからである。

はじめに、ショットにおけるカメラモーションを決定する手法について述べる。前述の通り、一般にオプティカルフローや動きベクトルを人工的な画像に用いた場合、平坦な色遣いの部分で、対応点が正確にとれないという問題がある。そこで、木村 [5]、西原 [6] らの手法を参考に、フローの信頼性を高める処理を行った。

木村ら [5] の手法では、勾配法を基に、フローの分布に関する仮定を置いている。一つは画像の小領域では、オプティカルフローは一定であること、もう一つは、画像全面に渡り滑らかに変化するということである。この仮定を基に、抽出された各フローの信頼度関数を評価することで、サブピクセルレベルの小さな移動の検出精度に優れる結果が得られるとしている。

本研究では、アニメーション映像に精度良く適用するために、西原ら [6] の手法をもとに、さらにエッジの法線方向にあたるフローに重みを付加した。これにより、アニメーション映像においても、大局的な動きを反映したフローを抽出することが可能になった。

ショットから抽出されたフローは、土橋ら [11] の手法によりクラスタリングを行うことで、パン、ズームの判定を行う。全画面でオプティカルフローが近い向きを持つパン区間では、特定のクラスの頻度が高くなる。一方、ズームでは均等に各クラスに動きベクトルが分布する。検出の手順としては、次のようになる。

1. 8つのクラス(図3参照)の頻度の、全体に占める割合を出す。
2. ヒストグラムの中で最大の頻度のクラスを探す ($c1$ とする)
3. その最大のクラスの両側のクラスの内に大きい方のクラスを選ぶ ($c2$ とする)
4. $c1$ と $c2$ のクラスの割合を加える。
5. 閾値によりパン・ズームの分離を行なう。

ここで、最大頻度のクラスのみの割合を用いないのは、動き予測の誤差から2つのクラスに大多数が分布する場合があるからである。

次に、検出されたパン、ズームの情報をもとに、各フレーム間において背景部がどのような動きをしているかを探索する。これには、秦泉寺ら [9] の手法で提案される、差分画像におけるオブジェクトの面積を最小にする動きが、大局的な動きを表現している、という条件を用いる(図4参照)。パンのシーンであれば、最尤クラスタの方向に対して全探索を行い、差分画像において閾値以上の画素の数が最も少なくなる点を求める。

この結果を用いて差分画像を生成することで、大局的な動きがある場合にも、前項で述べた、無い場合と同様の手法でオブジェクトを抽出することが可能になる。

3.2.1 実験結果

大局的な動きの検出、補正に関して、実験を行った。いずれもパンやズームを含むショットに対して、オプティ

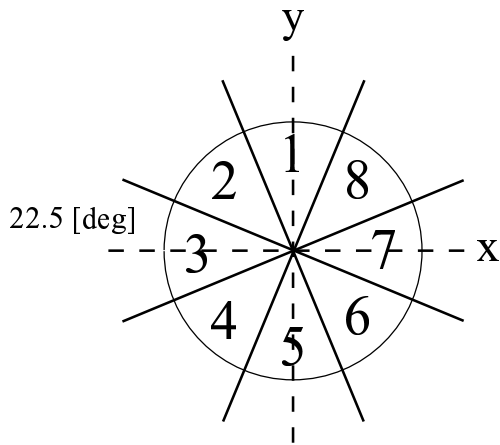


図 3 8 方向へのクラスタリング

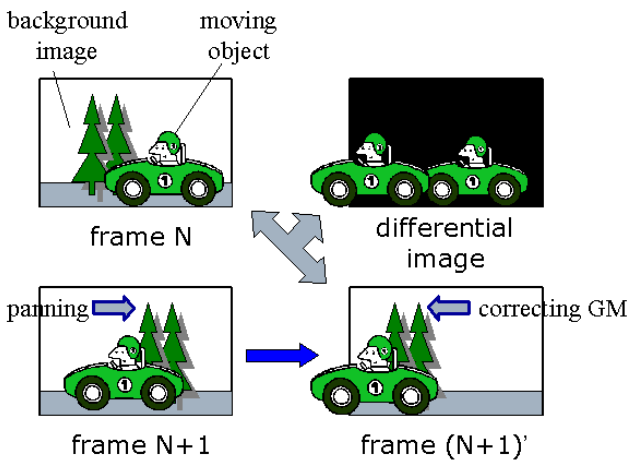


図 4 大局的な動きの補正

カルフローを用いることでその検出を行った。そして、パンを含むショットに関しては、全探索法により求めた補正值と、手で求めた実測値を比較した。

実験条件を表 1 に示す。図 5 に、パン (“pan1” ~ “pan3”) およびズーム (“zoom1”) を含むショットにおける各フレーム間の全フローを同一平面上に展開したものと、それをクラスタリングした結果を示す。なお、“pan1”, “pan3” はオブジェクトの動きを含み、“pan2” および “zoom1” はオブジェクトに動きは無い。

表 1 実験条件

テスト画像	“pan1”, “zoom1”	“pan2”, “pan3”
画像解像度	352[pixel] × 240[line]	640[pixel] × 480[line]
	4:2:0	4:2:0
色空間	YCbCr	YCbCr

フローの信頼度を高める前置処理の結果、平坦な色遣いを含む映像においても、パンやズームの検出を行うことが可能であることが確認された。特に、“pan3” ではオブジェクトの動きが比較的多いが、クラスタリングアルゴリズムと併せることで、正確な判定が可能になっている。

次に、パンのショットにおいて、全探索を行った結果を図 6 ~ 図 8 に示す。各フレームにおける大局的な動き

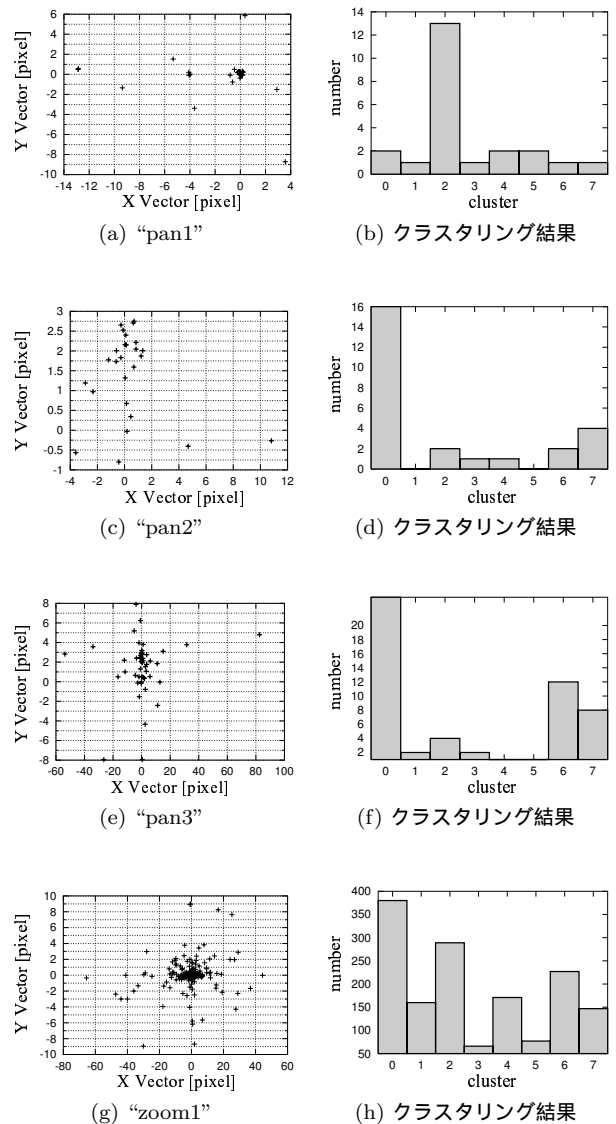


図 5 大局的な動きがあるショットでのオプティカルフローを時系列上にグラフにしたもので、横軸はフレーム番号、縦軸は画素数である。

単純な全探索法では、誤探索の問題がある。画面中に占める割合の大きいオブジェクトが移動する場合、そちらを大局的な動きと判定することがある。ショットのシーン判定を行わない場合、いくつかのフレームにおいて誤判定した。しかし、前段階で抽出されたパン方向へ探索範囲を絞ることにより、これを回避できた。すなわち、オブジェクトがカメラモーションと別の方向に動く場合は、その探索結果を除外することが可能になる。結果を見ると、検出結果はほぼ実測値と等しく、正確に抽出することが可能であるといえる。

3.3 オブジェクト抽出結果

提案手法によるオブジェクト抽出は、エッジ情報に基づいているため、輪郭線法の入力として理想的な初期輪郭線が得られれば、正確なオブジェクトの抽出が可能になる。ただし、差分画像のエッジが、背景領域における輪郭線と交差している場合などは、初期輪郭線が得られ

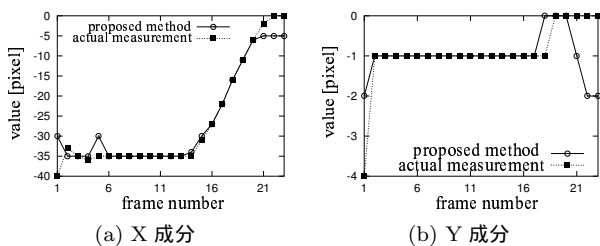


図 6 “pan1” における検出結果

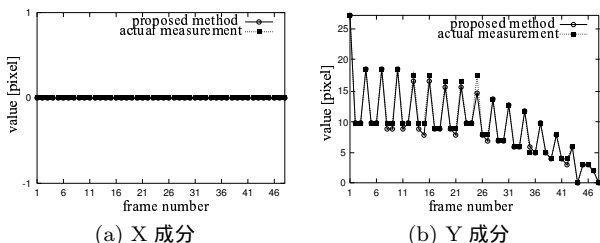


図 7 “pan2” における検出結果

ず、オブジェクトの抽出そのものに失敗する。

表 2 にオブジェクトの抽出結果を示す。フレーム間でオブジェクトに移動等の変化がある場合のうち、差分画像から初期輪郭線が得られた場合を成功と判断し、その割合を示す。テスト画像は、先ほどのシーケンス (“pan1” および “pan3”) に加え、大局的な動きが無い場合のショット (“fix1” および “fix2”) について示す。これらの実験条件は、“pan1” と同じである。

表 2 オブジェクト抽出結果

テスト画像	オブジェクトに変化がある場合	初期輪郭線獲得に成功した場合	割合 %
pan1	7	5	71
pan3	20	14	70
fix1	8	8	100
fix2	18	17	94

大局的な動きがある場合は、1 ピクセルでもその補正がずれると、背景領域による偽の輪郭線が発生し、オブジェクトの輪郭線と交差するため輪郭線が得られないことがある。そのため、全探索において、オブジェクトの面積最小となる点という条件に加え、さらに精度を上げるための拘束条件が必要になると考えられる。一方で、大局的な動きが無い場合には、オブジェクトの正確な抽出が可能となった。

4 まとめと今後の課題

セルアニメーション映像を対象に、ショット中からセル画によるオブジェクトを抽出するための手法を提案した。実験により、カメラ固定の状態では、高い確率でオブジェクトの抽出に成功することを確認した。

また、ショット中に大局的な動きを含む場合への対応を考察した。実験の結果、カメラワークの判定は、オブジェクトの動きによらず成功することがわかった。大局的な動きを補正して差分画像を生成することで、オブジェクトの抽出率が向上した。しかし、補正に 1 画素で

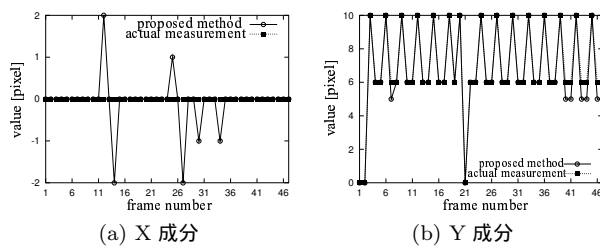


図 8 “pan3” における検出結果

もずれが生じた場合、抽出に失敗することがあることも判明した。

今後の課題としては、ズームや、パンとズームを同時に含むような場合でも適用できるかどうかを確認することが挙げられる。

参考文献

- [1] O. Nakagami, T. Miyazawa, H. Watanabe, and H. Tominaga, “Study on two-layer coding for animation images,” In Proc. *IEEE Int'l Conf. on Multimedia and Expo (ICME)*, Aug. 2002
- [2] N. Shimizu, T. Miyazawa, W. Kameyama, H. Watanabe, H. Tominaga, “A novel decoder downloadable system for content-oriented coding,” In Proc. *IEEE Globecom 2002*, Nov. 2002
- [3] 岩田, 安居院, “セルアニメーション画像の間の閉領域同定方法に関する研究,” 信学技法, PRL80-9, pp. 9-16, Sep. 1980
- [4] 塩原, 後藤, 吉田, “3次元構造に基づくアニメーション画像のフレーム間対応,” 信学論誌 (D-II), Vol. J73-D-II, No. 10, pp. 1687-1695, Oct. 1990
- [5] 木村, 太田, 金谷, “精密なノイズモデルによるオプティカルフローの検出,” 情処研報, 96-CV-99-6, pp. 37-42, Mar. 1996
- [6] 西原, 吉田, 酒井, “信頼度に基づくブロック統合を用いた動画像の高精度動きベクトル推定とその応用,” 映像情報メディア学会誌, Vol. 53, No. 1, pp. 148-156, Jan. 1999
- [7] P. Anandan, M. Irani, R. Kumar, and J. Bergen, “Video as an image data source: efficient representations and applications,” In Proc. *IEEE Int'l Conf. Image Processing (ICIP)*, Oct. 1995
- [8] 田中, 佐野, 大原, 奥平, “パラメトリックテンプレート法とその高精度ロバストマッチングへの応用,” 信学論誌 (D-II), Vol. J-83-D-II, No. 4, pp. 1119-1130, Apr. 2000
- [9] 秦泉寺, 渡辺, 岡田, 小林, “MPEG-4 スプライト符号化を用いた超低レート動画像圧縮,” 信学論誌 (D-II), Vol. J84-D-II, No. 5, pp. 758-768, May 2001
- [10] 小館, 朱, 金, 浦野, “輪ゴムかけの概念に基づく境界線の構造記述と未定義文字処理システムへの応用,” 信学論誌 (D-II), Vol. J84-D-II, No. 1, pp. 1-11, Jan. 2001
- [11] K. Dobashi, A. Kodate, H. Tominaga, “Camera working parameter extraction for constructing video considering camera shake,” In Proc. *IEEE Int'l Conf. on Image Processing (ICIP)*, Oct. 2001
- [12] 境田, 苗村, 金次, “背景差分法と時空間 watershed による領域成長法を併用した動画像オブジェクトの抽出,” 信学論誌 (D-II), Vol. J84-D-II, No. 12, pp. 2541-2555, Dec. 2001
- [13] Bae, Y. J., 高橋, 中嶋, “セルアニメーション制作補助のためのキャラクタ動作データベース構築 第 1 報: Shot 変化の自動検出,” 通信全大, D-12-80, Mar. 2002
- [14] 角, 中嶋, “2D アニメーション用標準動画像データベースシステム,” 通信全大, D-12-81, Mar. 2002

アニメーション映像における動き情報検出の検討

A study on moving area detection for animation images

中神 央二 *1

渡辺 裕 *1

富永 英義 *1, *2

Ouji NAKAGAMI*1

Hiroshi WATANABE*1

Hideyoshi TOMINAGA*1, *2

*1 早稲田大学 国際情報通信研究センター

*2 早稲田大学 理工学部 電子・情報通信学科

*1 Graduate School of GITS, WASEDA Univ. *2 Dept. of Elec. Info. and Comm. Eng., WASEDA Univ.

1 はじめに

我々は、対象となるコンテンツに最適な符号化方式を適用するという、コンテンツオリエンテッド符号化の概念を提唱し、研究を行ってきた [1], [2]. なかでも、アニメーション画像の性質に着目し、静止画像に対する符号化の検討を行ってきた. 今回は、この符号化方式を動画画像へ応用する際の第一歩として、アニメーション映像中に含まれる、同一セル画による領域の同定について考察する.

ここで対象とするのは、いわゆるセルアニメと呼ばれるものである. これは一般に、映像を作成する際、水彩画で描かれた背景の上に、セル画によるキャラクタを重ねて撮影される. この際、彩色行程の煩雑さから、同一のセル画を何度も使用することが多い. TV 放映されるアニメーション番組において、実際に作成されるセル画の枚数は、すべて異なるコマとして計算した場合の 10 分の 1 程度と言われている. したがって、映像中のどの部分が同一セル画によるものかを判定することができれば、符号化する際の情報量を減らすことが可能になる.

2 アニメーション映像における動きの分類

セル画の部分は、それが表現するオブジェクトの動きを表現するために、1 コマずつ僅かに変化させたものが用いられる. この変化の種類には、様々な手法が用いられるが、最も基本的なものとして、次の 3 つが挙げられる.

スライド キャラクタの平行移動のこと. 隣合うフレームにおいて、セル画を上下左右にわずかにずらして撮影することで表現される. この時、拡大・縮小、回転を加えることもあり得る.

リピート 限られた複数枚のセル画を繰り返し用いることで、動きを表現する. 瞬きのシーンなど.

ローリング セル画のスライド等で一定の範囲を揺れるように反復移動させる方法. 馬車の車輪など.

すなわち、アニメーション映像における動きとは、一枚、あるいは数枚のセル画で表現されることが多い. この動きを検出、判定することで、実際に用いられているセル画の数を数えることができる. なお、ローリングに関しては、スライドとリピートの組み合わせと考えることができる.

3 提案アルゴリズム

従来、アニメーション映像に対する研究としては、セル画の彩色を容易にするため、隣接フレーム間におけるキャラクタのパーツの対応付けが行われてきた [3], [4]. これらでは、対応付けの際に色情報を使えないこと、領域が変形する場合にも対応付けをする必要があることから、実際に得られる対応関係に、それほど精度が得られないという問題があった. すなわち、形状情報のみに依存するマッチングアルゴリズムは、今回のシステムを実現する上であまり適当ではないといえる. そこで考えられるのはテンプレートマッチングを基盤とした検出アル

ゴリズムであるが、はじめにテンプレートとなるべき領域を切り出すことが必要となる.

まとめると、提案システムで必要となるのは、1. 動きを検出し、それが画像中のどの領域であるかを抽出する技術、2. その領域が、別の画像でどのように使われているかを検索する技術、の 2 つである.

なお前提として、動きを検出する際には、予めショットの単位へ分割されたシーケンスを対象とする. これは、シーン分割を行う自然画の手法が、精度良くアニメーション映像に適用できるという前提 [15] に基づいている. また、今回は自然画におけるカメラモーションのような、背景画像を動かした場合の大局的な動きについて考慮していない. 背景を固定したショットにおいて、画像中の任意の数の領域が変化した場合に絞っている. 提案アルゴリズムのフローチャートを、図 1 に示す.

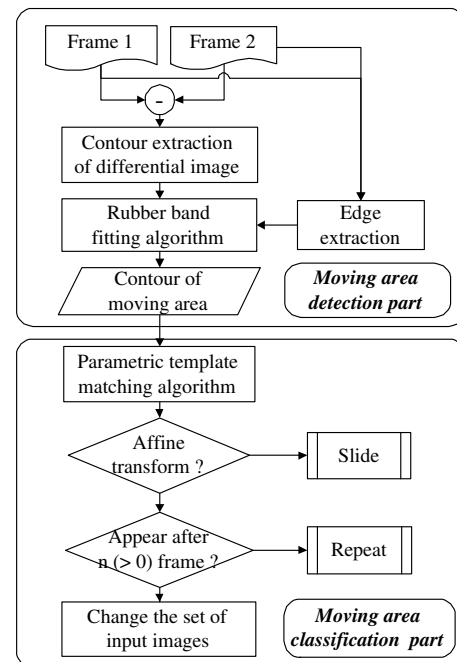


図 1 動領域の検出・追跡アルゴリズム

4 動領域の抽出

アニメーション映像における動領域を検出するには、フレーム間の差分画像を利用する. 自然動画像を対象としたオブジェクト抽出には、動きベクトルやオプティカルフローを用いるもの (!! 参考文献) も提案されているが、アニメーション映像の場合、平坦な色遣いの領域が多く含まれるため、誤検出が起きることが多い. また、予め背景画像を作成する手法も提案されているが (!! 参考文献), これはスライド以外の動きの検出に用いることができない.

差分を絶対値として扱った画像を生成した場合、変化した部分の領域が検出される (図 2 参照). 変化前の領域を A , 変化後を B とすれば、差分画像は $A \cup B$ の領域となる. そしてその輪郭線は、前後フレームにおける各

領域の輪郭線を併合させたものとなる。この粗い輪郭線情報を利用して、各フレームにおいて対応する領域の、真の輪郭線を抽出する。これには、元画像のエッジ情報と、輪ゴムのアルゴリズム（!! 参考文献）を用いる。

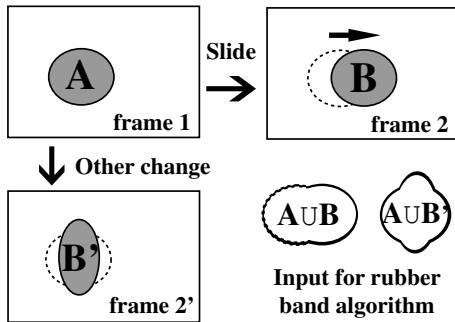


図2 動領域の粗い輪郭線抽出

輪ゴムのアルゴリズムは、視覚的な凹凸情報を抽出してオブジェクト形状を段階的に表現する手法である。まず概念的にオブジェクトに輪ゴムをかけ、輪ゴムとオブジェクトとの接触/非接触関係を、画素の反転処理により抽出する一連の操作を反復するという比較的単純な操作によって、オブジェクトの全境界線特徴の抽出を行うものである。

今回は、輪ゴムの始点を、差分画像より検出した粗い輪郭線情報とし、その対象を、元画像におけるエッジとする。アニメーション画像にはたくさんのエッジ成分が含まれるが、これによりそのフレームにおいて変化している領域の輪郭線のみを検出することができる。

最後に、検出した輪郭線を用いて、元画像から後段のマッチングに用いるテンプレート画像を切り出す。

5 動きの判定

判定すべき領域を抽出した次の段階は、その領域がどのような変化をしているかを分類する。スライドであれば、次フレームにおいて、同一の領域が、位置変化や拡大・縮小、回転といった変化を伴って登場する。一方、リピートをしている部分であれば、 $n(> 1)$ フレーム後に、同一の領域が同じ位置に登場する。したがって、抽出した領域は、そのショット中の全フレームを対象として、アフィン変換の可能性を考慮しつつ、探索する必要がある。

これには、テンプレートマッチングの技術を用いる。ただし、アフィン変換に対応する必要があること、また大量のフレーム中を対象に検出する必要があること、から、通常のテンプレートマッチング法では対応できないため、パラメトリックテンプレート法を用いる（文献!!）。これは、テンプレートの集合がなすパラメトリックなテンプレート空間を構成し、このテンプレート空間と探索画像とのマッチングを高速に行うアルゴリズムである。

スライドやリピートが検出された場合、

6 まとめと今後の課題

アニメーション映像において、キャラクターの動作など、セル画により表現される動きを検出する技術の検討を行った。部分的な動きについて、それを表現する領域の抽出、追跡を行うことで、動きの種類の判定と、冗長性の検出を行った。今回のアルゴリズムでは、自然画におけるカメラモーションのように、画面全体が動く場合に対応できないという問題点がある。大局的な動きへの対応は、今後の課題である。

参考文献

- [1] O. Nakagami, T. Miyazawa, H. Watanabe, and H. Tominaga, "Study on Two-layer Coding for Animation Images," In Proc. *IEEE Int'l Conf. on Multimedia and Expo (ICME)*, Aug. 2002
- [2] N. Shimizu, T. Miyazawa, W. Kameyama, H. Watanabe, H. Tominaga, "A novel decoder downloadable system for content-oriented coding," In Proc. *IEEE Globecom 2002*, Nov. 2002 to appear.
- [3] 岩田, 安居院, "セルアニメーション画像の間の閉領域同定方法に関する研究," 信学技法, PRL80-9, pp.9-16, Sep. 1980
- [4] 塩原, 後藤, 吉田, "3次元構造に基づくアニメーション画像のフレーム間対応," 信学論誌 (D-II), Vol. J73-D-II, No. 10, pp.1687-1695, Oct. 1990
- [5] 松山, 有田, 長尾, "線分の空間的配置関係に基づく線図形の構造マッチング," 情処論誌, Vol. 24, No. 6, pp. 735-744, Nov. 1983
- [6] P. Anandan, M. Irani, R. Kumar, and J. Bergen, "Video as an image data source: efficient representations and applications," In Proc. *IEEE Int'l Conf. Image Processing (ICIP)*, Oct. 1995
- [7] 孫, 大町, 加藤, 阿曾, "特徴量の要素の相関を考慮した高速・高精度な識別関数と文字認識への応用," 信学論誌 (D-II), Vol. J-81-D-II, No. 9, pp.2027-2034, Sep. 1998
- [8] 村瀬, V. V. Vinod, "局所色情報を用いた高速物体探索 -アクティブ探索法-, " 信学論誌 (D-II), Vol. J-81-D-II, No. 9, pp.2035-2042, Sep. 1998
- [9] Ronald A., and Yuan-Fang W., "Geometric and Illumination Invariants for Object Recognition," *IEEE Trans. on PAMI*, Vol. 21, No. 6, pp. 505-536, June 1999
- [10] George Nagy, "Twenty Years of Document Image Analysis in PAMI," *IEEE Trans. on PAMI*, Vol. 22, No. 1, pp. 38-62, Jan. 2000
- [11] 田中, 佐野, 大原, 奥平, "パラメトリックテンプレート法とその高精度ロバストマッチングへの応用," 信学論誌 (D-II), Vol. J-83-D-II, No. 4, pp.1119-1130, Apr. 2000
- [12] 斉藤 文彦, "遺伝的アルゴリズムを用いた画素選択テンプレートによる画像マッチング," 信学論誌 (D-II), Vol. J-84-D-II, No. 3, pp.488-499, Mar. 2001
- [13] 金谷, 金澤, "テンプレートマッチングによる対応探索の自動しきい値設定法," 情処研究報告, 2002-CVIM-132-4, pp. 23-30, Mar. 2002
- [14] Clark F. Olson, "Maximum-Likelihood Image Matching," *IEEE Trans. on PAMI*, Vol. 24, No. 6, pp. 853-857, June 2002
- [15] Bae. Y. J., 高橋, 中嶋, "セルアニメーション制作補助のためのキャラクター動作データベース構築 第1報: Shot 変化の自動検出," 通信全大, D-12-80, Mar. 2002
- [16] 角, 中嶋, "2D アニメーション用標準動画データベースシステム," 通信全大, D-12-81, Mar. 2002

二層構造によるアニメーション画像符号化の検討

A Study on Two-layer Coding for Animation Images

中神 央二*¹ 宮澤 敏記*²
Uuji NAKAGAMI*¹ Toshinori MIYAZAWA*²
† 早稲田大学大学院 国際情報通信研究科
† Graduate School of GITS, WASEDA Univ.

渡辺 裕*¹ 富永 英義*^{1,2}
Hiroshi WATANABE*¹ Hideyoshi TOMINAGA*¹
‡ 早稲田大学理工学部 電子・情報通信学科
‡ Dept. of Elec. Info. and Comm. Eng., WASEDA Univ.

1 はじめに

今日、Discrete Cosine Transform (DCT) を基盤においた符号化方式は、MPEG や JPEG をはじめとして、画像の圧縮に広く用いられている。しかしながら、DCT ベースの手法は、必ずしも全ての種類の画像に適しているとは限らない。ある種の画像は、その特性を考慮することで、より効率良く符号化することが可能である。これを、我々は Content oriented coding と呼び、その具体例として、アニメーション画像符号化の研究に取り組んでいる [1]。

2 二層による符号化

一般にアニメーション映像を作成する際には、水彩画で描かれた背景の上に、セル画によるキャラクタを重ねて撮影される。セル画の部分は、動きを表現するため枚数が多いという都合上、通常は線画とべた塗り部分、すなわち均等色領域として表現される。

アニメーション画像を、DCT で符号化する際に相性が悪いのは、主にセル画の部分であるといえる。線画やキャラクタの顔部分など均等色の領域の輪郭で、輝度・色差が激しく変化するため、多くの高周波成分が存在する。高い圧縮率を得るため、DCT 変換後の係数を量子化する段階で、これら高周波成分が失われると、復号時にモスキートノイズとして画質劣化を招く。

提案手法においてアニメーション画像は、基本層と付加層の 2 層で符号化する。図 1 に、符号化処理の流れ図を示す。

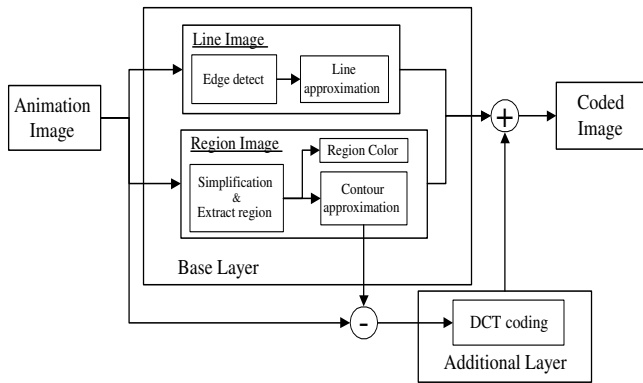


図 1 符号化の流れ

2.1 基本層の作成

基本層では、均等色領域と、線画の部分に着目して符号化する。まず、原画像から微分フィルタを用いて線画を抽出する。次に、同じく原画像から、領域成長法 [3] を用いて均等色領域を抽出し、各領域の輪郭線と代表色を抽出する。線画や均等色領域の輪郭として検出された線に対し、線近似の手法を用いて、有意点を生成する。復号時には、この有意点情報から元の点列を再構成できるため、元画像よりも符号化量を減らすことが可能になる。

ところで、アニメーション画像における背景領域は、自然画に近い性質をもつため、領域成長法のみでは全ての背景領域を均等色領域として抽出することが不可能である。そこで、前置処理として平滑化を用いることで、背景領域におけるグラデーションや、テクスチャ情報など均等色領域として抽出できない情報を消去する。この後に領域成長法を用いることで、べた塗りの部分ではない背景領域及び近似された均等色領域を抽出することが可能になる。これによりアニメーション画像中で、べた塗りで描かれている部分からも、背景領域

として描かれている部分から、均等色領域を抽出することが可能になる。

以上の要件を満たす平滑化フィルタの一つとして、Connected Filter を用いた [4]。このフィルタの利点は、元画像におけるエッジ成分を完全に残すことにある。このフィルタをアニメーション画像に対して用いることにより、アニメーションに多いエッジの情報を保ったまま、背景領域に含まれる、グラデーションや細かな情報を消去することができる。

この後、領域成長法を用いることで、背景部からも、ある程度の大きさの領域が抽出可能になり、画像全体を均等色領域の集合として表現できる。

2.1.1 実験結果

図 2 と図 3 に、基本層の作成結果を示す。元画像の解像度は 352×240 [pixel] で、色空間は YCbCr、サンプリング比 4:1:1 である。一般的なアニメーション画像から、背景を含まない画像 (図 2(a)) と含む画像 (図 3(a)) を選び、符号化を行った。



(a) 元画像

(b) 基本層の作成結果

©The Dog of Flanders Prod.

図 2 背景領域がない画像 (“dog1”)



(a) 元画像

(b) 基本層の作成結果

©The Dog of Flanders Prod.

図 3 複雑な背景領域を含む画像 (“dog2”)

抽出した領域についてのデータを表 1 に示す。図 2 では、背景領域を含まず単純なため、抽出される領域数は少ない。一方、図 3 では、平滑化の結果として背景領域から抽出される領域の一つ一つが小さいため、各領域の平均の輪郭点数は小さくなる。

これらの出力結果は SVG 形式とした。なお、ここで示すサイズに関しては、SVG を可逆圧縮した SVGZ 形式のものを示した。

輪郭線の近似に関しては、[2] のアルゴリズムを用いた。各領域について、元の領域を構成する点数の 1 割程度まで点数の削減が可能になっている。

2.2 付加層の作成

基本層で符号化される情報は、画像中の形状の有意点情報と、各均等色領域内の色情報のみなので、符号化される情報量は少ない。しかし、全体を均等色領域として近似することにより、背景領域を多く含むような画像では符号化品質が損

表 1 抽出した領域とデータサイズ

	“dog1”	“dog2”
領域数	27	41
平均輪郭点数	360	291
平均有意点数	43	31
圧縮率 [%]	11.9	10.6
サイズ [KB]	3.26	5.17

なわれる．そこで，さらに差分情報を加えることで，背景領域の均等色近似による誤差を補う．これを付加層とし，前述の基本層に追加することで，高品質な符号化画像を実現する．

元画像と，線画と抽出した全ての均等色領域を合成したものの差を，差分画像と定義する．差分画像には，主に平滑化により失われた，比較的高い周波数成分を持つ情報が含まれる．このような画像には，周波数成分に着目した符号化方式が適しているといえる．

また，差分情報は，元画像と線画・均等色領域による近似画像の差全体を1つの画像として扱うのではなく，各均等色領域について元画像との違いを求め，さらに，それらをブロック単位で扱う．これにより，符号化するブロックを選ぶことで，ユーザが目的に応じて符号化量と品質の制御を行うことも可能になる．次に，付加層の符号化手順を示す．

1. 差分画像を 8×8 画素のブロックに分割する．
2. 近似した均等色領域の一つ一つについて，上にかかるブロックを調べる．
3. 該当するブロックにおいて，均等色領域と元画像の差分 D を式 (1) 及び式 (2) で求める．
4. D の大きなブロックから優先的に符号化する．

$$D(i, j) = P_o(i, j) - P_h(i, j) \quad (1)$$

$$D = \sum_{i=0}^7 \sum_{j=0}^7 D(i, j) \quad (2)$$

ただし， $P_o(i, j)$ 及び $P_h(i, j)$ は，元画像と，対応する均等色領域における点 (i, j) の画素値である．

2.2.1 均等色領域の輪郭部を含む領域における処理

均等色領域の輪郭を含むブロックにおいても，差分情報を符号化する．抽出された領域が， 8×8 画素のブロックより小さい場合や，複雑な形状をしており差分画像に輪郭を含むブロックが多く含まれる場合に効果がある．

ただし，通常の DCT，量子化という手順により符号化を行った場合，差分画像においても輪郭を含むブロックではモスキートノイズが発生する．これは，ブロック中において，均等色領域ではない部分と元画像の差がエッジとなることによる．

そのため，パディング処理を行うことでこの問題を解決する．比較的差分画像の値に近い，偽の値を，画像のない部分に詰め込むことを行う．これにより，DCT 変換を行った周波数領域でのブロック情報を低周波成分に集めることが可能になる．図 4 に，パディング処理の様子を示す．

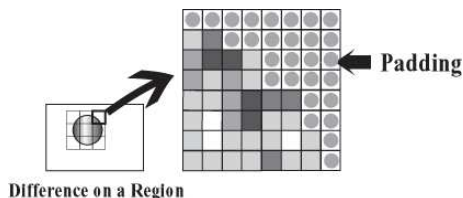


図 4 パディング処理

2.2.2 差分の大きいブロックの優先的符号化

各差分ブロックについては，全てを符号化するのではなく，含まれる情報の絶対値が大きいものから優先的に符号化する手順をとる．

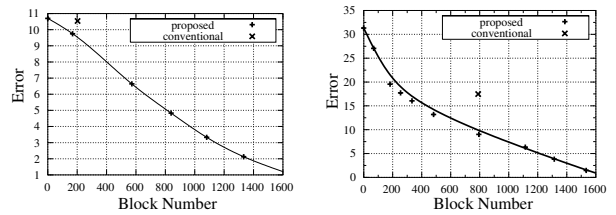
差分情報は，色と形状の正確な近似が行われている部分においては，その絶対値が小さいものと考えられる．すなわち，

キャラクタなど元画像においてべた塗りの部分を近似した領域については，均等色領域による近似の誤差が小さい．このような領域は，差分を DCT で符号化したとしてもその効果は薄い．一方，平滑化操作を行った背景領域では，元々均等色領域ではない訳であるから，この部分でのブロックにおいて差分情報が多く含まれる．

したがって，差分情報の絶対値が大きいブロックから優先的に符号化を行えば，より効率よく差分情報を補うことができる．

2.2.3 実験結果

図 5 に差分を符号化した場合の実験結果を示す．横軸は符号化ブロック数，縦軸は式 (3) により求めた符号化誤差である．



(a) “dog1”

(b) “dog2”

図 5 付加層の効果

$$Error = \sum_{i=0}^{x_size-1} \sum_{j=0}^{y_size-1} \frac{abs(P_o(i, j) - P_c(i, j))}{x_size \times y_size} \quad (3)$$

従来手法では，線画，均等色領域として抽出できなかった領域を DCT により符号化するブロックとするため，データは 1 点のみ得られる．誤検出した背景領域の近似誤差により，符号化誤差が大きくなる．

提案手法では，符号化差分ブロックが増加すると，誤差の減少の割合は緩やかになる．後半になる程，差分ブロックに含まれる情報が少なくなり，補正の効果が薄いからである．同じ符号化ブロック数において，従来手法より誤差を減らすことが可能になった．

また，背景領域が複雑である dog2 では，均等色領域の多い dog1 に比べ，差分ブロック補正が行われない場合の誤差が大きい．従って，このような画像では，付加レイヤの必要性が高いといえる．

3 まとめ

高圧縮を目的に画像全体を領域に分割し，各輪郭を関数近似により表現することを提案し，ベクタ情報のみで表現される基本層を作成した．さらに，近似により失われた差分情報を補足するための DCT 変換という方式を付加層と定義し，アニメーション画像を階層的に符号化するための検討を行った．

参考文献

- [1] 宮澤，亀山，渡辺，阪谷，富永，“アニメーション画像符号化の基礎検討”，通信学会総合大会，pp.66, D-11-66, Mar. 2001
- [2] 宮澤，中神，亀山，渡辺，富永，“Dynamic Programming によるセル画アニメーション内均等色領域輪郭線符号化の検討”，情処研報，AVM36-12, 2002.
- [3] R. Adams and L. Bischof, “Seeded Region Growing,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(6), pp 641-647, Jun. 1994.
- [4] P. Salembier and J. Serra, “Flat Zones Filtering, Connected Operators, and Filters by Reconstruction,” *IEEE Transactions on image processing*, Vol.4, No.8, pp.1153-1160, Aug. 1995.
- [5] C. A. Christopoulos, W. Philips, A. N. Skodras and J. Cornelis, “Segmented image coding: Techniques and experimental results,” *Signal Processing: Image Communication*, Vol.11, pp.63-80, Nov. 1997.

平滑化操作によるアニメーション画像符号化の検討

A Study On Animation Image Coding By Smoothing Operation

中神 央二^{*1}

宮澤 敏記^{*1}

渡辺 裕^{*2}

富永 英義^{*1*2}

Ouji NAKAGAMI^{*1}

Toshinori MIYAZAWA^{*1}

Hiroshi WATANABE^{*2}

Hideyoshi TOMINAGA^{*1*2}

† 早稲田大学理工学部 電子・情報通信学科
† Dept. of Elec. Info. and Comm. Eng., WASEDA Univ.

‡ 早稲田大学大学院 国際情報通信研究科
‡ Graduate School of GITS, WASEDA Univ.

1 はじめに

我々は、アニメーション画像の特性に着目し、周波数分解による符号化ではなく、形状情報による符号化方式の検討を行っている。本稿では、符号化に際しての画像の分類を見直し、1. 画像平滑化手法による画像全体からの均等色領域の抽出、次に2. 抽出された均等色領域における元画像との差分情報のDCTによる符号化を提案する。

2 画像平滑化手法による均等色領域抽出法の検討

2.1 従来手法の問題点

従来の方式 [1] では、アニメーション画像に、領域成長法を直接用いるため、べた塗りのキャラクタ部分以外でも、滑らかな色変化を含む背景領域を誤抽出するという問題があった。

その結果起こる問題点として、1. 誤抽出された領域は一般に複雑な形状をしており、直線とスプライン曲線による近似が難しく、データ量が増えること、2. DCTで符号化するべき領域の定義が曖昧で、方式が統一されていないことが挙げられる。

2.2 提案手法

これらの問題を解決するために、我々は [3][4] を参考に、画像全体を平滑化する Connected Operator による前置処理を用いることを考案した。(図1)

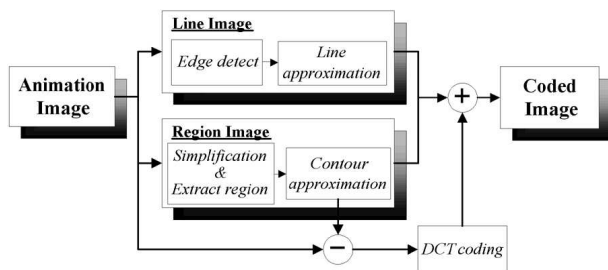


図1 提案方式のフローチャート

以下、その概要を述べる。

画像の輝度値を閾値で判定し、いくつかのレベルに分割する。それぞれの画素について近傍を調べ、レベルが一致するものを連結する。連結した画素の集まりを層と定義し、隣り合う層は必ず輝度値レベルの上下関係を持つ。一番端になる層から、隣り合う層との大きさ、輝度値の差を調べ、条件を満たさないものを消去する。これにより、画像のエッジは保存されたままで、領域内の細かい情報を消去することができる。べた塗り領域であるキャラクタ部は、この影響を受けない。

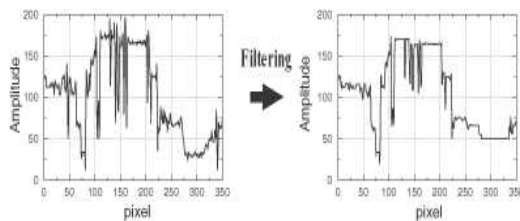


図2 平滑化の処理結果

この結果に対し、領域成長法を用いることで、背景部も、ある程度の大きさの領域として抽出が可能になり、画像全体を均等色領域の集合として表現できる(図3)。



©The Dog of Flanders Prod.

図3 Connected Operator によるフィルタの適用

3 差分情報のDCTによる符号化

3.1 提案手法

提案方式では、DCTで符号化する情報は、元画像と先に作成した均等色領域の差分とする。

まず、差分をDCTで符号化することの妥当性について述べる。領域内に着目すれば、1. 差分情報は大きなエッジを含まずモスキートノイズの原因となりにくいこと、2. 均等色領域として直流成分を別に保存しているためブロックノイズが現れにくいこと、が挙げられる。従って、差分はDCTで符号化してもノイズの原因になりにくいといえる。

また、各マクロブロックは、トータルの符号化量の制御を可能にするため、適応的に符号化するものとする。

3.2 結果

図4に、従来手法と提案手法での、符号化前後の差を示す。横軸は符号化マクロブロック数、縦軸は符号化前後の差である。

従来手法では、線画、均等色領域として抽出できなかった領域を全て符号化マクロブロックとするため、一点となる。誤検出した背景領域の近似誤差により、符号化誤差が大きい。

提案手法では、近似した領域上で、元画像との差分が大きいマクロブロックから優先的に符号化を行っている。符号化マクロブロック数が少ないうちは、主に背景部分での差分が補われることがわかる。

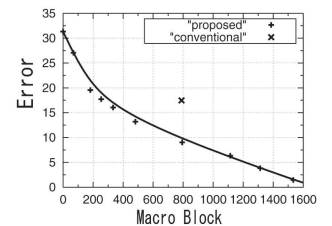


図4 符号化 MB 数と誤差の関係

4 まとめ

本稿では、高圧縮を目的に、画像全体を領域に分割し、各輪郭を関数近似により表現するための前置処理について述べた。また、色情報の単色近似により失われたテクスチャ情報を補足するために、差分情報にDCT変換を用いるという方式について考察した。

参考文献

- [1] 古角, 渡辺, 小林, "スプライン/DCTハイブリッド符号化方式を用いたアニメーション映像における領域抽出の一検討," 信学技法, pp. 23-30, November(1999)
- [2] 宮澤, 亀山, 渡辺, 阪谷, 小宮, 富永, "アニメーション符号化の基礎検討," 通信学会全国大会 (2001)
- [3] P.Salembier and J.serra, "Flat Zones Filtering, Connected Operators, and Filters by Reconstruction," IEEE Transactions on image processing, Vol.4, No.8, pp.1153-1160, August(1995)
- [4] C.A.Christopoulos, W.Philips, A.N.Skodras, J.Cornelis, "Segmented image coding: Techniques and experimental results," Signal Processing:Image Communication, pp.63-80, November(1997)

平滑化操作によるアニメーション画像符号化の検討

中神 央二[†] 宮澤 敏記[†] 渡辺 裕^{††} 富永 英義^{†,††}

[†] 早稲田大学 理工学部 電子情報通信学科
〒 1169-8555 東京都新宿区大久保 3-4-1
^{††} 早稲田大学 大学院 国際情報通信研究科
〒 169-0051 東京都新宿区西早稲田 1-3-10

E-mail: †{ouji,miyazawa,tominaga}@tom.comm.waseda.ac.jp, ††hiroshi@giti.waseda.ac.jp

あらまし 本研究では、アニメーション画像の特性に着目し、画像に含まれる線要素及び輪郭形状の情報を符号化する方式に関する検討を行う。本稿では、従来手法における背景領域からの均等色領域誤抽出の問題を解決するため、符号化に際しての形状情報及び背景画像の抽出手法を見直した。すなわち、背景部分に残された均等色領域を抽出することで、有意点のみでアニメーション画像を符号化することを目指した。これを実現するため、エッジを保存する平滑化操作を用いた。また、平滑化により失われた情報を補うため、抽出された均等色領域における元画像との差分情報を、DCTにより符号化することを提案した。

キーワード 画像符号化, アニメーション映像, 平滑化操作, 領域抽出, 差分画像

A Study On Animation Image Coding By Smoothing Operation

Ouji NAKAGAMI[†], Toshinori MIYAZAWA[†], Hiroshi WATANABE^{††}, and Hideyoshi TOMINAGA^{†,††}

[†] Department of Electronics, Information and Communication Engineering, Waseda University
Okubo 3-4-1, Shinjuku-ku, Tokyo, 169-8555 Japan

^{††} Graduate School of Global Information and Telecommunication Studies, Waseda University
Nishiwaseda 1-3-10, Shinjuku-ku, Tokyo, 169-0051 Japan

E-mail: †{ouji,miyazawa,tominaga}@tom.comm.waseda.ac.jp, ††hiroshi@giti.waseda.ac.jp

Abstract We proposed a coding scheme specially designed for animation images. Considering about characteristics of animation images, we try to extract lines and homogeneous color regions from animation images. There are two issues to improve this coding scheme. Firstly, animation images are not only represented by lines and homogeneous regions, but also background regions. Conventional approach uses DCT to encode background regions because it is usually smooth like natural images. We use smoothing operator and region-growing algorithm jointly for this purpose. The region-growing algorithm extracts homogeneous color regions correctly than the conventional approach. Secondly, compensating scheme for loss by smoothing operation is considered. We propose to use DCT for the difference between the original images and the approximated images because luminance signals of the difference contain smooth changes. High-quality images can be obtained by adding this differential data to approximated images that are created by homogeneous color regions and line drawings.

Key words Image coding, animation image, smoothing operator, region extraction, different image

1 はじめに

アニメーション映像は、毎日数多くの作品がTVで放送され、また、ビデオ化、DVD化されている。子供から大人まで楽しめるコンテンツも多く、幅広い世代で人気のあるコンテンツである。

アニメーション映像の特徴として、線画やべた塗りで描かれている部分が、多く含まれることが挙げられる。すなわち、線画の境界や均等色領域の輪郭部分で、エッジが多く存在するといえる。アニメーション映像に対し、高圧縮、低ビットレートの伝送を目的として、従来の周波数分解による符号化を行うと、特にエッジの部分で品質が損なわれるという問題点がある。

そこで、本研究では、アニメーション映像の特性に着目し、周波数分解による符号化ではなく、画像に含まれる線要素及び輪郭形状の情報を符号化する方式に関する検討を行っている。

本稿では、符号化に際しての従来の方式におけるアニメーション映像の分類を見直し、1. 画像平滑化手法による画像全体からの均等色領域の抽出、次に2. 抽出された均等色領域における元画像との差分情報のDCTによる符号化を提案する。

2 従来手法について

先行研究 [1][2] では、アニメ映像を線画、均等色領域、背景の3つの領域に分類している。図1に先行研究における、アニメーション映像符号化の流れ図を示す。

一般にアニメーション映像を作成する際には、水彩画で描かれた背景の上に、セル画によるキャラクタを重ねて撮影される。セル画の部分は、動きを表現するため枚数が多いという都合上、通常は線画とべた塗り部分で表現される。

アニメーション映像を、DCTで符号化する際に相性が悪いのは、主にセル画の部分であるといえる。線画や均等色領域の輪郭部分で、輝度・色差が激しく変化するため、多くの高周波成分が存在する。高い圧縮率を得るため、DCT変換後の係数を量子化した場合、エッジ周辺の高周波成分が失われ、復号時にモスキートノイズとして画質劣化を招く。

そこで、エッジ(線)はスプライン関数で近似し、べた塗り部分を検出した均等色領域は、その輪郭について同じく近似することを提案している。また、ここで言う背景とは、線やべた塗りとして検出できなかった部分を指す。

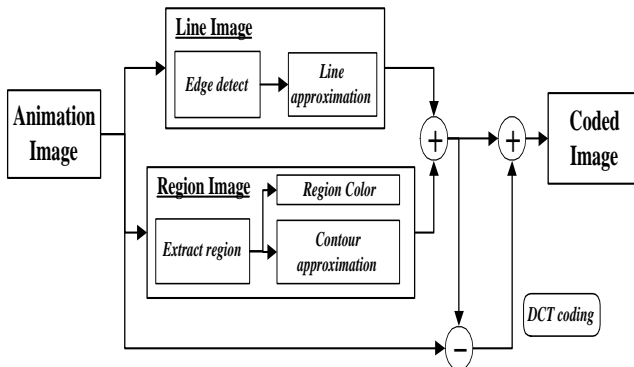


図1 従来方式による符号化

この手法の問題点として、均等色領域の検出に用いる領域成長法では、本当に純粋なべた塗り領域以外も検出してしまふことがある。

2.1 領域成長法

ここで用いられている領域成長法について、アルゴリズムを述べる(図2参照)。

1. 種となる一点から始まり、その近傍の点を閾値処理で判断していき、ほぼ同じ色情報をもつならば統合する。
2. 統合した際、色情報は領域内の全画素の平均値に更新する。
3. 領域に隣接する画素の近傍についても、1と同様の閾値判断をし、領域を広げていく。
4. 1~3を繰り返す。
5. 領域の成長が止まれば、まだ領域に含まれていない点を種とする。

以上により、画像中から均等色の領域が検出される。

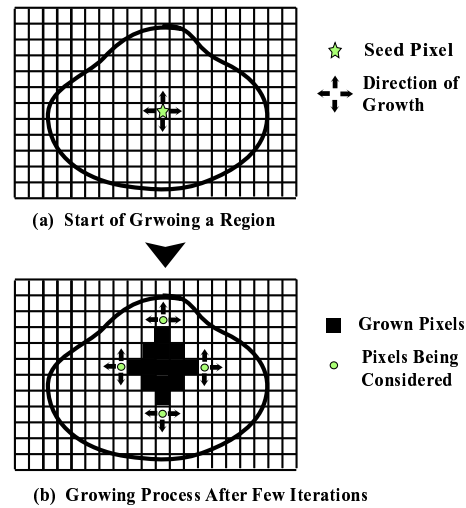


図2 領域成長法

2.2 従来方式の問題点

アニメーション映像に、直接領域成長法を用いることで、均等色領域として抽出すべきキャラクタ部分以外でも、一部を誤抽出してしまう。つまり、背景部分であっても比較的似た色ならば領域成長法で検出される。



©The Dog of Flanders Prod.

図3 背景領域の誤検出

その問題点として、

1. 誤抽出された領域は一般に複雑な形状をしており、直線とスプライン曲線による近似が難しく、データ量が増えること、

2. DCT で符号化するべき領域の定義が曖昧で、方式が統一されていないことが挙げられる。

3 提案方式

本稿では、アニメーション画像にべた塗りの領域が多く存在することに着目し、画像全体を、均等色領域の集合で表現することを提案する。これにより、従来手法における背景領域の誤抽出という問題を解決する。

アニメーション画像における背景領域は、自然画に近い性質をもつため、領域成長法のみでは全ての背景領域を均等色領域として抽出することが不可能である。提案手法では、前置処理として平滑化を用いることで、背景領域におけるグラデーションや、テクスチャ情報等均等色領域として抽出できない情報を消去する。この後に領域成長法を用いることで、べた塗りの部分ではない背景領域からも、近似された均等色領域を抽出することが可能になる。

すなわち、アニメーション画像中で、べた塗りで描かれている部分からも、背景領域として描かれている部分からも、均等色領域を抽出することが可能になる。

本方式の利点は、画像全体を統一した方式で扱えることである。従来手法においては DCT で扱うとする背景領域と関数近似を用いて符号化する領域の区別が曖昧であったことを指摘したが、本手法で抽出した領域と線画を併せたものに関数近似を行うことで、画像全体を、有意点のみで符号化できる。

提案方式での符号化アルゴリズムを図 4 に示す。アニメーション画像を、線画と均等色領域のみで表現する基本レイヤと、平滑化により失われた情報を補うための付加的なレイヤの、2 層で符号化するものとする。

第 1 層で符号化される情報は、画像中の形状の有意点情報と、各均等色領域内の色情報のみなので、符号化される情報量は少ない。しかし、全体を均等色領域として近似することにより、背景領域を多く含むような画像では符号化品質が損なわれる。そこで、さらに差分情報を加えることで、背景領域の均等色近似による誤差を補う。

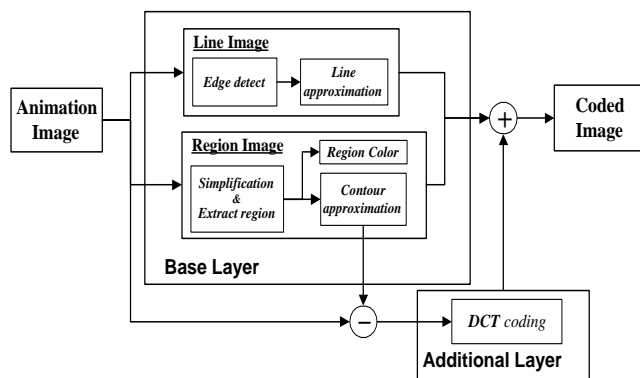


図 4 提案方式による符号化

4 画像平滑化手法による均等色領域抽出法の検討

画像の平滑化操作のために、従来のローパスフィルタに代表される線形フィルタを用いると、エッジの部分まで平滑化される（図 5 参照）アニメーション画像に多く

含まれるエッジが平滑化されると、品質が著しく損なわれることになる。そのため、エッジ情報は保存されたまま、領域内のみ平滑化されるような操作が必要となる。

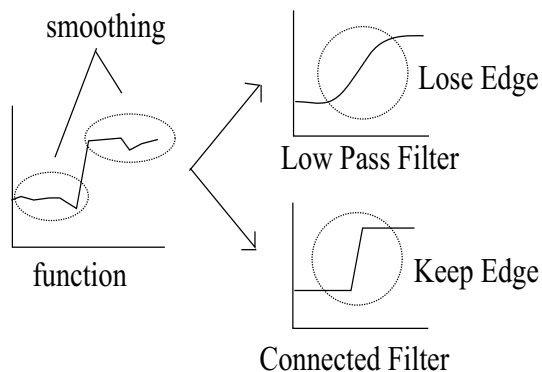


図 5 フィルタによる平滑化

4.1 Connected Filter を用いた平滑化

以上の要件を満たす平滑化フィルタの一つとして、Connected Filter を用いたものが挙げられる [4]。Connected Filter では、Flat-zone という概念を導入することで、元画像におけるエッジ成分を完全に残すことを可能とした。このフィルタをアニメーション画像に対して用いる利点は、べた塗り領域は平滑化することなく、主に背景領域に対して平滑化を行うことができることにある。

4.1.1 Flat-zone

それぞれの画素について近傍を調べ、同じ輝度値を持つ画素を結合した領域のことを Flat-zone とよぶ。仮に同じ輝度値の画素が 4, ないし 8 近傍にない場合は、その点は、一点でも Flat-zone とする。

図 6 に、各 Flat-zone を色分けして表示させたものを示す。

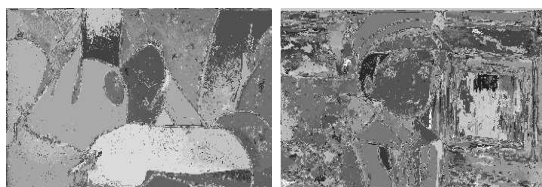


図 6 視覚化した Flat-zone

Flat-zone の統合処理は、隣り合う Flat-zone の大きさ、輝度値の差を判断することで行う。ここで用いたのは、判断基準に対する Flat-zone の半径と、繰り返し回数をパラメータとして、モルフォロジー処理を行うものである。

4.2 Connected open and close

モルフォロジー処理において、Open and Close 処理は、孤立点の除去に用いられるが、それだけでは輝度値の連結性を考慮していないため、グレースケール画像に対する平滑化に適さない。

そこで、構成要素に式 (1),(2) のように 4 ないし 8 近傍の連結性を持ったものを用いる。これにより孤立する Flat-zone の統合処理が可能になる。孤立する Flat-zone を消去すれば領域内のみを平滑化することができる。

$$B_4(r) = \{ (x_1, x_2) \in Z^2 : |x_1| + |x_2| \leq r \} \quad (1)$$

$$B_8(r) = (x_1, x_2) \in Z^2 : \max(|x_1|, |x_2|) \leq r \quad (2)$$

Connected Filter の処理結果を図 7 に示す．図中，白線部において，平滑化処理前と後の，輝度値の変化をグラフにしたものを図 8 に示した．横軸が X 座標，縦軸が輝度値である．エッジを保ったまま平滑化が行われていることがわかる．



図 7 Connected Operator によるフィルタの適用

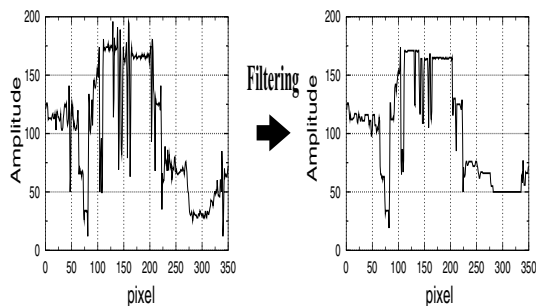


図 8 X 軸の断面

この結果に対し，領域成長法を用いることで，背景部も，ある程度の大きさの領域として抽出が可能になり，画像全体を均等色領域の集合として表現できる．

4.3 実験結果

2 種類のアニメーションコンテンツから，背景領域の簡単なものと複雑なもの計 4 画像に対して，提案手法による符号化実験を行った．実験条件を次に示す．実験結

表 1 実験条件

テスト画像	dog1,dog2,anpanman1,anpanman2
画像フォーマット	4:4:4 format
	輝度信号：352[pixel] × 240[line]
	色差信号：352[pixel] × 240[line]
色空間	YCbCr

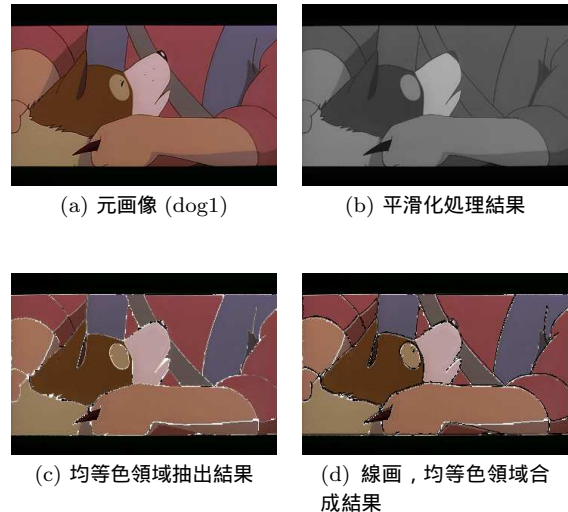
果を図 9 から図 12 に示す．

Connected Filter による領域内のみで，エッジが失われない平滑化を行ったので，dog1 や anpanman1 といったほぼ均等色領域のみで描かれている画像においては，ほとんど影響を受けない(図 9(b)，図 11(b))．一方，dog2 や anpanman2 における複雑な背景領域においては，平滑化の結果が顕著に見られる(図 10(a) 図 10(a))．

均等色領域の抽出に関しては，dog2 など一部小さな領域において抽出できない部分が残っている．しかし，概ね他の画像に対しては，線画部分を除いた画像全体からの領域抽出に成功している．

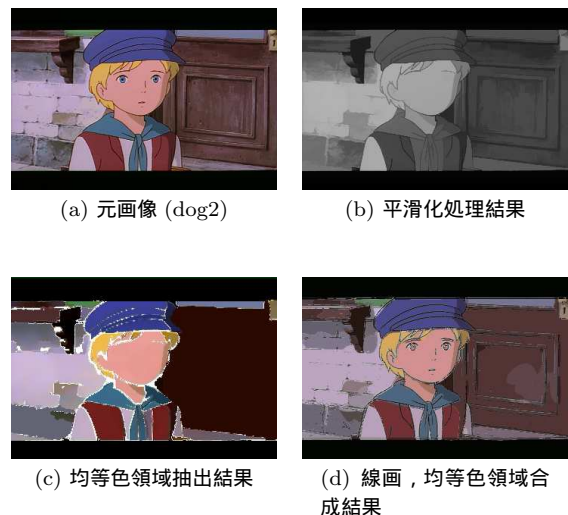
5 差分情報の DCT による符号化

前章で述べた平滑化手法を用いて，画像全体から均等色領域を抽出することにより，線画と均等色領域の集



©The Dog of Flanders Prod.

図 9 背景領域がない画像



©The Dog of Flanders Prod.

図 10 複雑な背景領域を含む画像

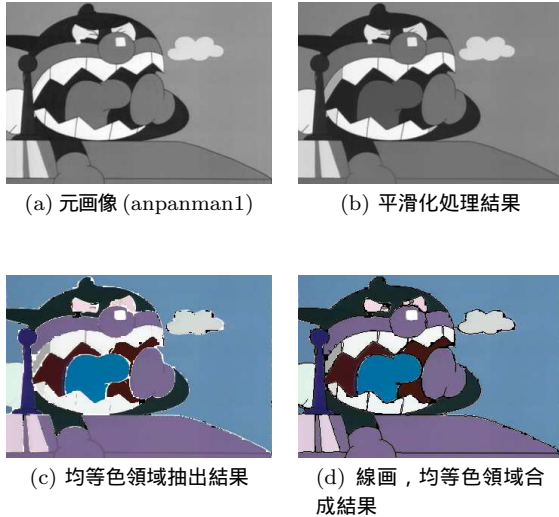
合によりアニメーション画像を符号化することが可能になった．しかし平滑化操作を用いることにより，符号化された画像は，特に背景領域に関して，領域内部のテクスチャ情報が失われているという問題点がある．

本章では，前章の失われたテクスチャの情報を補うことについて考察を行う．

5.1 差分情報の位置づけ

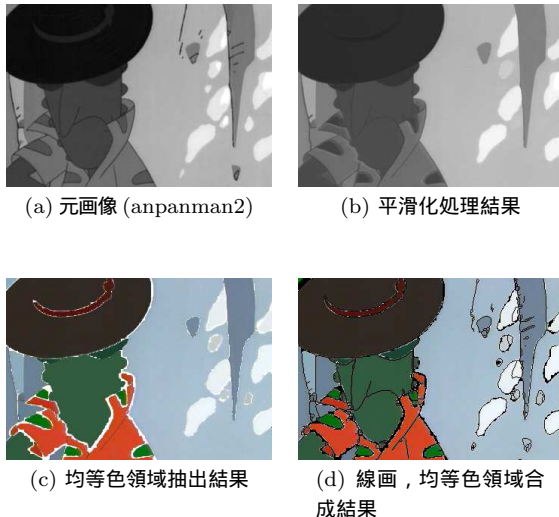
元画像と，線画や均等色領域を合成したものの差を，差分画像と定義する．差分画像には，主に平滑化により失われた，比較的高い周波数成分を持つ情報が含まれる．このような画像には，周波数成分に着目した符号化方式が適しているといえる．

この差分画像を，先に作成した線画や均等色領域による画像の上のレイヤー，すなわち付加的な情報としておくことで，符号化品質を高めることが可能になる．さらに，符号化画像の階層を分けることで，ユーザが目的に応じて符号化量と品質の制御を行うことも可能になる．



©やなせたかし/フレーベル館・TMS・NTV

図 11 単純な背景領域を含む画像



©やなせたかし/フレーベル館・TMS・NTV

図 12 複雑な背景領域を含む画像

5.2 DCT による符号化の妥当性

まず、差分を DCT で符号化することの妥当性について述べる。

領域内のみに着目すれば、

1. 均等色領域として直流成分を別に保存しているためブロックノイズが現れにくいこと
2. 差分情報は大きなエッジを含まず、モスキートノイズの原因となりにくいこと

が挙げられる。

ブロック歪 ブロック歪は、低周波係数（主に直流成分）の量子化誤差により生じる。差分情報の場合、均等色領域内の色情報として、直流成分は別に確保しているため、ブロック歪は発生しにくい。

モスキート歪 差分情報においても、ブロックが均等色領域の輪郭を含むような場合、エッジが存在することに

なるので、量子化によりモスキート歪が発生する可能性がある。

関連研究 [5] では、SIC (Segmented image coding) による符号化の一環として、画像をセグメントに分割する方法と、さらにそれぞれのセグメントの上に DCT で情報を付加する方法を提案している。但し、[5] の場合、輪郭にまたがる DCT ブロックでは一切符号化せず、また、輪郭以外の領域上の全てのブロックに対して差分を符号化している。

しかし、デコード時に差分を符号化したブロックと符号化していないブロックでは、品質の差がはっきりと現れるため問題がある。この問題を解決するため、提案手法では、パディング処理を行いモスキート歪を回避することで、輪郭を含むようなブロックにおいても符号化を行う。

5.3 差分情報の符号化

始めに、差分情報の符号化方法について述べる。差分情報は、元画像と、線画・均等色領域による近似画像の差全体を 1 つの画像として扱うのではなく、各均等色領域について元画像との違いを求め、さらに、それらをブロック単位で扱う。

1. 差分画像を 8×8 画素のブロックに分割する。
2. 近似した均等色領域の一つ一つについて、上にかかるブロックを調べる。
3. 該当するブロックにおいて、均等色領域と元画像の差分を式 (3) で求める。

$$Diff(i, j) = Pixel_{orig}(i, j) - Pixel_{homo}(i, j) \quad (3)$$

5.3.1 均等色領域の輪郭部を含む領域における処理

均等色領域の輪郭を含むブロックにおいても、差分情報を符号化する。抽出された領域が、 8×8 画素のブロックより小さい場合や、複雑な形状をしており差分画像に輪郭を含むブロックが多く含まれる場合に効果がある。

ただし通常の DCT、量子化という手順により符号化を行った場合、差分画像においても、輪郭を含むブロックではモスキートノイズが発生する。ブロック中において、均等色領域ではない部分と元画像の差がエッジとなるからである。

そのため、パディング処理を行うことでこの問題を解決する。パディングとは、情報を含まない部分でも差分に近いダミーの情報を詰め込むことである。

パディング処理 ブロック内で色信号に大きな差があると、DCT 変換による高周波成分が増加し、画像復元時にノイズとして現れる。

このため、比較的差分画像の値に近い、偽の値を、画像のない部分に詰め込むことを行う。これにより、DCT 変換を行った周波数領域でのブロック情報を低周波成分に集めることが可能になる。図 13 に、パディング処理の様子を示す。

5.3.2 差分の大きいブロックの優先的符号化

各差分ブロックについては、全てを符号化するのではなく、含まれる情報の絶対値が大きいものから優先的に符号化していくものとする。

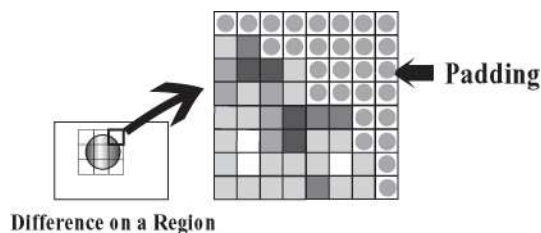


図 13 パディング処理

差分情報は、色と形状の正確な近似が行われている部分においては、その絶対値が小さいものと考えられる。すなわち、キャラクタなど元画像においてべた塗りの部分を近似した領域については、均等色領域による近似の誤差が小さい。このような領域は、差分を DCT で符号化したとしてもその効果は薄い。一方、平滑化操作を行った背景領域では、元々均等色領域ではない訳であるから、この部分でのブロックにおいて差分情報が多く含まれる。

従って、差分情報の絶対値が大きいブロックから優先的に符号化を行えば、より効率よく差分情報を補うことができる。

5.4 実験結果

図 14 に差分を符号化した場合の実験結果を示す。横軸は符号化ブロック数、縦軸は式 (4) により求めた符号化誤差である。

$$Error = \sum_{i=0}^{x_size-1} \sum_{j=0}^{y_size-1} \frac{abs(Pixel_{orig}(i,j) - Pixel_{homo+dct}(i,j))}{x_size \times y_size} \quad (4)$$

従来手法では、線画、均等色領域として抽出できなかった領域を DCT により符号化するブロックとするため、一点となる。誤検出した背景領域の近似誤差により、符号化誤差が大きい。

提案手法において、符号化差分ブロックが増加すると、誤差の減少の割合は緩やかになる。これは、元画像でのべた塗り領域上における差分情報を補っているため、差分情報自体が少なく、補正の効果が薄いからと考えられる。

また、背景領域が特に複雑である dog2 では、均等色領域の多い dog1, anpanman1 に比べ、差分ブロック補正が行われない場合の誤差が大きい。従って、このような画像では、付加レイヤの必要性が高いといえる。

6 まとめ

高圧縮を目的に画像全体を領域に分割し各輪郭を関数近似により表現するための前置処理について述べた。さらに、近似により失われた差分情報を補正するための DCT 変換という方式により、アニメーション画像を階層的に符号化するための検討を行った。

今後の課題として、線画に対する関数近似手法の精度向上により、有意点情報による符号化品質を高めることが必要と考えられる。

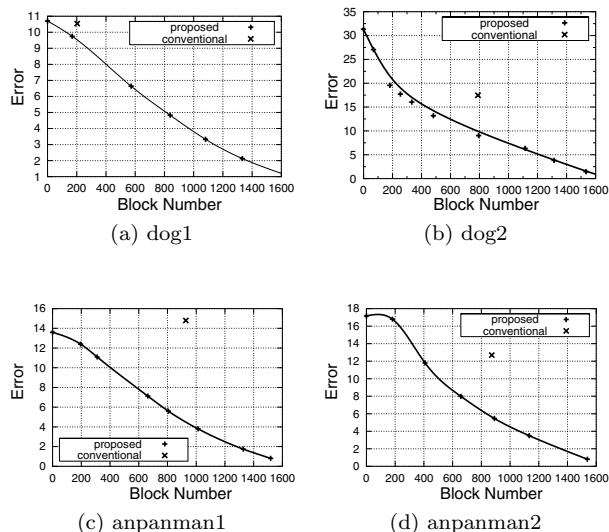


図 14 符号化した差分ブロック数と誤差

謝辞

本稿で実験に用いた画像の一部は、日本テレビ放送網株式会社のご好意によりお借りしました。ここに感謝の意を表します。

参考文献

- [1] 古角, 渡辺, 小林, "スプライン/DCT ハイブリッド符号化方式を用いたアニメーション映像における領域抽出の一検討," 信学技法, pp. 23-30, Nov.(1999)
- [2] 古角, 渡辺, 小林, "アニメーション映像における領域抽出の一検討," 画像符号化シンポジウム (PCSJ99), P-3.15, Sep.(1999)
- [3] 宮澤, 亀山, 渡辺, 阪谷, 富永, "アニメーション画像符号化の基礎検討," 通信学会総大会, pp.66, D-11-66, Mar.(2001)
- [4] P. Salembier and J. Serra, "Flat Zones Filtering, Connected Operators, and Filters by Reconstruction," IEEE Transactions on image processing, Vol.4, No.8, pp.1153-1160, Aug.(1995)
- [5] C. A. Christopoulos, W. Philips, A. N. Skodras, J. Cornelis, "Segmented image coding: Techniques and experimental results," Signal Processing:Image Communication, pp.63-80, Nov.(1997)
- [6] 森, 和田, 寅市, "関数化図形表現を用いた紙文書のデジタル化," 情報処理学会研究報告 Vol.99, No.57, pp.17-23, Jul. (1999)
- [7] 斉藤, 穂坂, "拡張した 2 次有理 Bezie 曲線の性質とその曲線近似法への応用," 情報処理学会論文誌, Vol.31, No.01, pp.33-41, Jan. (1990)
- [8] U.Ramer, "An iterative procedure for the polygonal approximation of planar curves," Computer Graphics and Image Processing Vol.1, pp.244-256 (1972)