

# アニメーション画像符号化の基礎検討

Basic Study on Animation Image Coding

宮澤 敏記\*1

亀山 渉\*2

渡辺 裕\*2

阪谷 徹\*3

富永 英義\*1\*2

Toshinori MIYAZAWA\*1

Wataru KAMEYAMA\*2

Hiroshi WATANABE\*2

Toru SAKATANI\*3

Hideyoshi TOMINAGA\*1\*2

\*1 早稲田大学 理工学部 電子・情報通信学科

\*2 早稲田大学国際情報通信研究センター

\*3 NTT サイバースペース研究所

\*1 Dept. of Elec. Info. and Comm. Eng., WASEDA Univ. \*2 Global Info. and Tele. Institute, WASEDA University

\*3 NTT Cyber Space Laboratories

## 1. はじめに

アニメーション画像を DCT を用いて符号化することによる問題点 (高周波成分除去による雑音) を解決するために、我々はアニメーション画像の特徴である線画、均等色領域、背景領域に着目し、アニメーション画像に特化した符号化方式の検討を行っている<sup>[1]</sup>。

本稿では、1. アンチエイリアスの影響による均等色領域抽出法の不備に関する解決手法、かつ 2. 直線及び長さの短い線の符号化法を提案することで従来のアニメーション画像符号化方式の改善を図った、また線画符号化に関して画質評価を行ったので報告する。

## 2. アンチエイリアスを考慮した均等色領域抽出法

### 2.1 問題点・提案アルゴリズム

入力画像において、アンチエイリアス処理のために線画の周辺画素に中間色が挿入されている。この中間色の影響で、均等色領域を正確に抽出できないという問題があった (図 1 参照)。

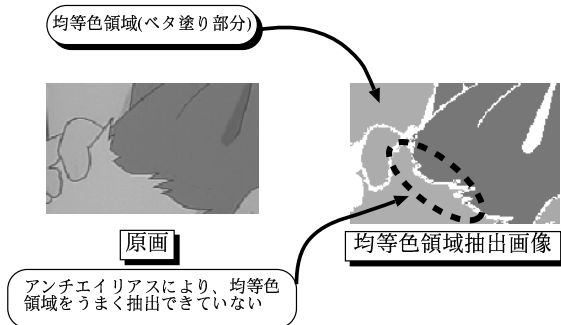


図 1: アンチエイリアスの影響

そのため、我々は均等色領域画像とは別に抽出した線画を元に、線画まで均等色領域を拡張する手法を考案した。考慮した点は、1. 原画において必ずしも線画は連続でない点、2. 線画の周囲に複数の均等色領域が存在し得る点である。前者に関しては、データ数の少ない線画を中心として均等色領域を拡張する必要があり、後者に関しては、領域拡張の対象となる画素の周囲を調べ、最も多く存在する画素の色を選択する必要がある。提案アルゴリズムの詳細は以下である。

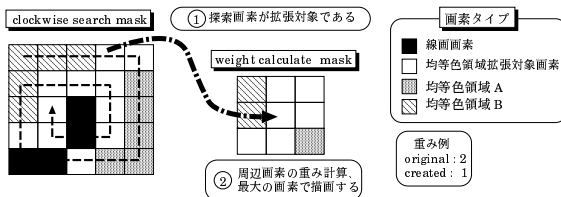


図 2: 均等色領域拡張法概念図

線画の一つの画素を中心とした時計回りに探索するためのマスクを用意する。探索する過程で、均等色領域拡張対象画素に到達した際、その画素を中心として重み計算を実行する。そして、最大の重みを持つ領域の色で対象画素を描画する。マスクで線画をトレースすることにより、アンチエイリアス部分を均等色領域として拡張して行く。

### 2.2 結果

本アルゴリズムの結果画像は図 5 のようになる。本アルゴリズムを用いることで、より線画に近づいた均等色領域を抽出することが可能となった。

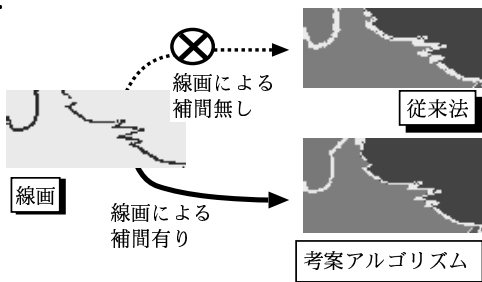


図 3: 線画情報に基づく均等色領域拡張法結果

## 3. 直線・スプライン曲線を組み合わせた線画・輪郭線符号化法

### 3.1 従来手法の問題点・提案

我々は、アニメーション画像における線画情報を 1. 線画、2. 輪郭線に分類し、符号化すべく検討を行っている。本稿で提案するのは、参考文献<sup>[1]</sup>で提案された手法を発展させ、線画を均等色領域の輪郭 (閉曲線) に存在する線として捉えるのではなく、線画データそのものを基とした符号化手法である。ここで、符号化対象となる線画データに着目すれば (図 4 左図)、スプライン曲線のみで線画を補間する際の問題点として次の 2 点が挙げられる。

- 目・髪など画像として重要な意味を持つ要素が、短い線で表現されており、スプライン曲線で補間できない点
- 局所的に、線画の中で直線要素が存在するため、スプライン曲線のみで線画を補間することは、データ品質・量の観点から不利である点

以上の問題を解決するために、次のようなアルゴリズムを提案する。まず、線画上を分岐点が存在するまで追跡し、線画を複数の線要素に分割する。次に分割した各線要素に対し、直線要素が存在するか調べる。そして直線要素が存在する場合、先に定義した線要素をさらに分割する。最終的に、作成された線要素群は、線長・線種 (直線・曲線) によって分類し、各線要素に適した線符号化方式を採用する。

### 3.2 結果・評価

実験では、線長が 20 画素以下の線を直線近似 (8 画素間隔でデータ点を抽出) した。入力画像 (線画) と補間画像の比較を図 4 に示し、またデータ量と誤差の関係を図 5 に示す。誤差算出方法は、線要素毎に、元の線と生成した線のなす面積を算出し、その総和を誤差として定義した。図 4 に示した出力画像は、図 5 から、同じデータ量で最も誤差の少ない画像であることが分かる。



図 4: 線画画像と直線・スプライン曲線補間画像の比較

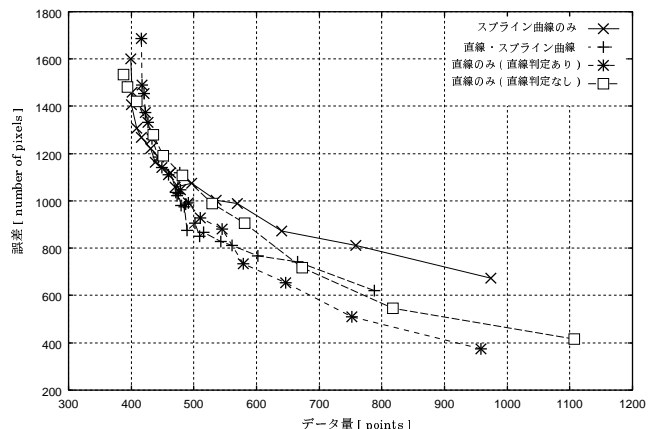


図 5: 直線・スプライン曲線補間におけるデータ量と誤差の関係

## 4. 今後の課題

現在、スプライン補間のための有意味な点は、曲率の大きな点を選択しているが、必ずしもすべての曲線に有意味な点が存在する訳でないため、等間隔に通過点を挿入している。この通過点により、スプライン曲線にうねりが生じてしまうことが判明している。今後は、本稿で提示した誤差算出法を元に等間隔に有意味な点を抽出するのではなく、誤差を最小とするような点を選択するアルゴリズムを考案する予定である。

### 参考文献

- [1] 古角, 渡辺, 小林: "アニメーション映像におけるスプライン / DCT ハイブリッド符号化の一検討", 情報学研報, AV24-3(3,1999)