

## 関数化画像を用いたデジタルコンテンツ

4E-6-2

森 浩一\*, 和田 耕一\*\*, 寅市和男\*\*

\*筑波大学 工学研究科

\*\*筑波大学 電子・情報工学系

kom@is.tsukuba.ac.jp

## 1. はじめに

Internet や Web の発展によりデジタルコンテンツへの要求は高まってきている。より魅力的なデジタルコンテンツを作成するには画像が不可欠である。ここで、デジタルコンテンツの出力デバイスについて考えると、携帯端末、テレビ、PC、プリンタなど多様な解像度の出力デバイスが用いられる可能性がある。通常用いられるピクセル画像では一つの画像ファイルで効率よく多様な解像度の出力に対応することは非常に困難である。なぜなら、ピクセルの密度を変えるのは容易ではなく、画像生成時に精度が決まってしまうより高い精度の出力に対応できないからである。さらに、高精度な画像はファイルサイズが増大するために、読み込み精度とファイルサイズのトレードオフが生じてしまう。

これに対し、画像中の色領域の集合で画像を表現する方法（輪郭表現画像）では、領域の輪郭を線分やスプライン関数で表現しているため、大きさを自在に変えることが可能である。これにより、高品質な拡大表示と出力デバイスの解像度を十分に活用した出力が行える。さらに、ファイルサイズは画像の解像度よりも、複雑さに大きく影響されるといった特長を持つ。近年では Web のデジタルコンテンツ表現方法として輪郭表現画像が注目されており、Macromedia の Flash[1]は既に多くの Web ブラウザで表示可能であり、W3C では XML ベースの Scalable Vector Graphics

(SVG)の標準化が進められており Working Draft が公開されている[2]。

画像を作成する場合、何もない状態から作るよりも、既に存在する紙媒体の文書や図を利用できれば、画像の作成が用意となる。そのためには、紙媒体の文書や図を高精細にデジタル化できる必要がある。輪郭表現画像は、高精細なデジタルコンテンツに適した画像表現方式だが、スキャナなどからの入力ピクセル画像なので変換が必要となる。既存のラスター画像から輪郭表現画像に変換する手法では輪郭線を直線やベジェ、スプライン曲線のみで近似表現しているものが多い[3][4][5]。しかし、それらの近似関数では表現しづらい形状があり、そのような輪郭に対しては近似に要するデータが増加するか近似の精度が低下してしまう。例えば、曲線を直線で近似するような場合である。そのため、既存の手法は条件の良い簡単な画像に対してのみ適用されており、実用的な文書画像には適していなかった。

筆者らは、2値画像中の輪郭を直線、円弧、2次 B スプライン曲線を組み合わせて近似表現する手法を提案した[6][7]。ここではこのような画像表現方式を関数化画像と呼ぶ。3つの近似関数を組み合わせることで、直線やスプライン近似のみで表現する場合に比べ、より少ないデータ量で高品位な画像が得られる。[6]で提案されている近似関数を決定する手法では、最適化手法である Dynamic Programming を用いることによって適切な関数近似が達成されており、実用的な文書画像に対しても十分に適用が可能である。

本論文では、紙文書から取りこんだ文書画像を関数化画像によってデジタル化し、様々な場面に適用して関数化画像のデジタルコンテ

<sup>†</sup>Digital Contents using image represented by function approximation  
Koichi Mori, Koichi Wada, Kazuo Toraiichi  
University of Tsukuba

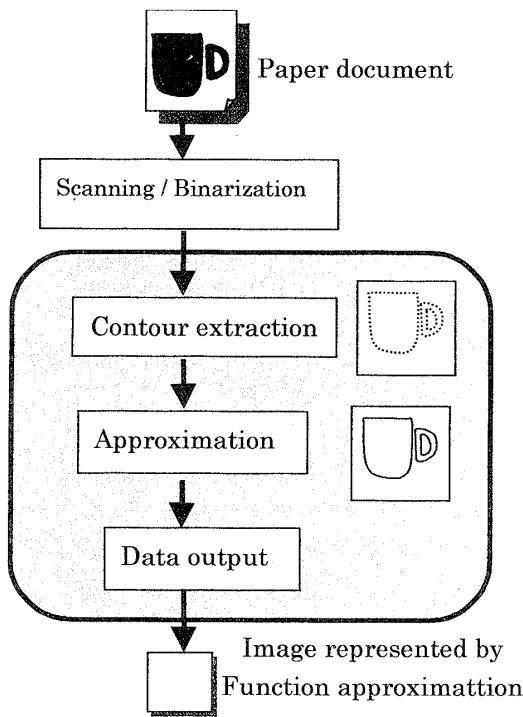


図1 関数化画像生成処理の流れ

ツとしての有用性を検証することを目的とする。

## 2. 関数化画像の生成手法

手法の詳細は[6]に述べられているので、ここでは概要のみを述べる。はじめに、図1に関数化画像生成処理の全体の流れを示す。

対象とする画像である文書画像では明確に輪郭が存在しており2値化してそのエッジを追跡すれば容易に閉輪郭線が得られる。関数化画像では、直線・円弧・2次Bスプライン曲線を使い分けて輪郭を近似表現している。その処理は、輪郭線を区分し、それぞれの区間をどの関数で近似するかを決定する処理であるとみなせる。輪郭の一部分をいくつかの関数で近似した例を図2に示す。この例からもわかる通り、近似関数が適当でないと再生成品質の低下や近似に要するデータの増加を招くことになるので、区間分割と近似関数を決定する処理が重要である。ここでは、

- もとの輪郭に忠実な関数近似
- 関数の区間幅をできるだけ長くする
- ある程度のノイズは無視する

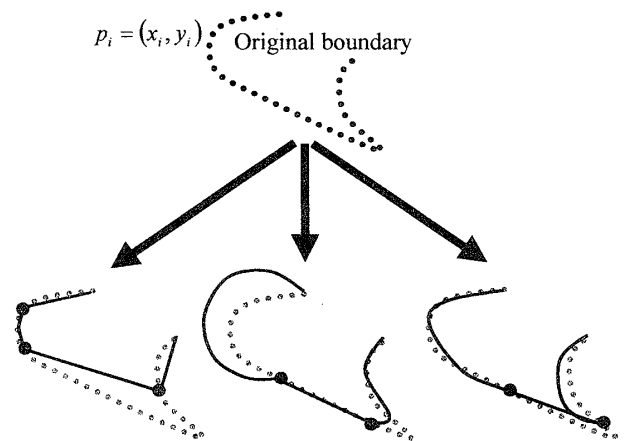


図2 輪郭の関数近似例

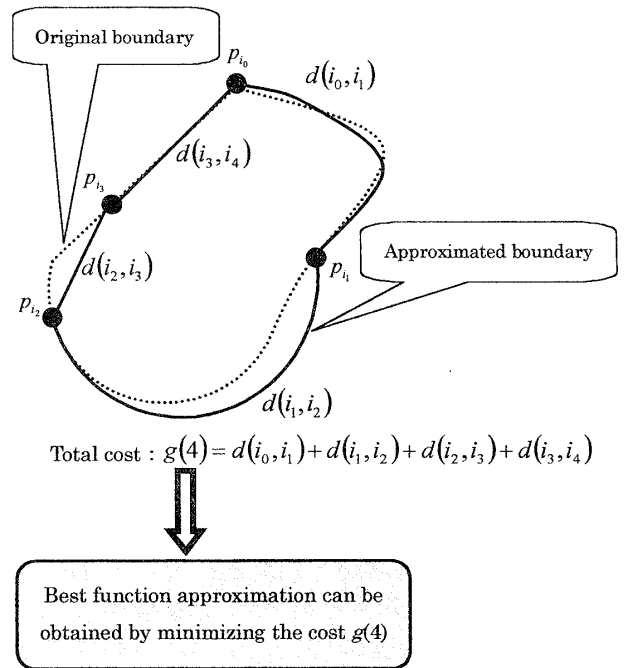


図3 近似コスト

といったことを最適な関数近似の目標として設定している。

[7]の手法では、輪郭点列の曲率によって近似関数を識別していた。理論的には、直線区間では曲率は0であり、円弧区間は曲率が一定となるので判別が可能である。しかし、離散的な点列から曲率を推定しているのもとも値が正確ではないことと、ノイズによって値が変化してしまうことなどから関数近似が不適切となることが多かった。また、曲率という局所的な特徴を用いているため全体

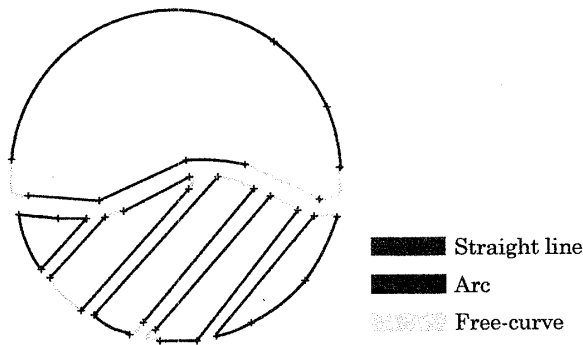


図4 関数近似手法の実行例

として適切な関数近似にならないという問題もあった。

そこで[6]では、輪郭の局所的な特徴だけでなく全体的な特徴を捉えて関数近似することを目指して、近似された輪郭コスト関数を算出し最適な関数近似を見つける、という最適化問題に帰着させた。そのために、区間毎の近似誤差を区間長で割った値をコスト関数として、それぞれの区間のコストの和を最小化することで適切な関数近似を行っている（図3）。このような最適化には **Dynamic Programming** が有効である。図4は[6]の手法を簡単な図形に適用した例であり、適切な関数近似がなされていることが分かる。

### 3. 関数化画像の再生・表示

#### 近似輪郭再構成手法

関数化画像のファイルには輪郭を再構成するために必要最低限のデータが格納されている。以下ではそれぞれの近似関数別の必要データと再構成手法を述べる。

#### 直線区間

直線区間は両端の座標が分かればよいが、始点は一つ前の区間の終点であるため実際には終点のみが保存される。再構成するには終点を輪郭に追加すればよい。

#### 円弧区間

円弧は始点、終点、中点が分かれば再構成できるが直線区間と同様の理由により、中点と終点のみが保存される。再構成するに

は、データから円弧の半径、開始角、終了角を算出し描画する。

#### 自由曲線区間

自由曲線区間は、輪郭の長さ方向を媒介変数とした  $x$  方向  $y$  方向二つの2次Bスプライン曲線で構成されている。データとしてはスプライン係数が必要となるが、スプライン係数は誤差がある範囲内でスプライン係数の数が最小となるように定められる[5]。再構成の際には、スプライン係数と2次Bスプラインとの畳み込み計算によって近似輪郭が得られる。

#### 3.1. 出力デバイスによる要求の違い

関数化画像を出力するには再構成された輪郭を描画すればよいのだが、多くの場合ポリゴンやベジェ曲線を描画する機能を利用することになる。このとき、出力デバイスの解像度に合わせた密度で輪郭点列を再構成すれば出力デバイスの解像度を十分に活用することができる。

#### プリンタ

プリンタの場合、多少出力に時間をかけてもプリンタの解像度を最大限生かした出力結果が要求される。したがって、デバイスの解像度に合わせて高精度なポリゴンを生成しそれを出力すればよいことになる。

#### ディスプレイ

解像度の低いディスプレイでは高い表示品質はあまり望めないためプレビュー的な表示が主目的となり、画像の詳細を見るためには拡大表示し、全体の概要を見るには画像を縮小表示するといった処理が頻繁に行われる。したがって、表示品質よりも高いインタラクティブ性が要求されるので、描画は高速に行われなくてはならない。

### 4. デジタルコンテンツとしての機能

ここでは始めに関数化画像の利点をふまえ、関数化画像がどのような場面で有効かということについて考える。

## ひとつのファイルで様々な解像度に対応できる

ラスタ画像では、解像度を高くするとファイルサイズが増大するため、出力デバイスを想定して画像を作る必要がある。これに対して関数化画像では、作成時に出力デバイスをあまり想定しなくて済む。この特長は、ディスプレイ表示と印刷の両方が想定されるような場面、例えば、文書ファイリングやデジタルブックなどにおいて有効であると思われる。

### 高速、高画質な拡大縮小が可能

この特長はディスプレイ表示する場合に有効となる。ディスプレイの解像度は低いため、全体を表示すると詳細はどうしても省略されてしまう。しかし、特に図や地図などでは詳細を見る場面も多い。そうすると、画像の拡大や縮小が頻繁に発生することになる。ピクセル画像では画質を保った拡大は困難であるが、関数化画像では高速に行える。また、どの大きさが見易いかは人によって異なる。そのためにも、様々な拡大率で高い画質が得られることが必要となる。

これらを踏まえて、ここでは次のような場面への適用を考えることにする。

- ・ インタラクティブなコンテンツ
- ・ デジタルブック
- ・ ファイリングシステム

#### 4.1. インタラクション機能

関数化画像の特長を活用するには、拡大・縮小、画面のスクロールが自在にできなければ

ならない。

見ることだけが目的のコンテンツであれば、拡大・縮小・スクロールだけでもよいが、特に Web 上のコンテンツでは、さらなるインタラクション機能が要求される。基本的には、イベント（マウスボタンのクリック、マウスカーソルが領域へ入る、など）によってアクション（色が変わる、オブジェクトの大きさが変わる、リンクをたどる、音を出す、など）を起こすという枠組みで処理される。

### 5. コンテンツとしての利用例

関数化画像の有効性を確かめるために、PC 上に C++ を用いて関数化画像生成部を実装し、Java を用いて関数化画像表示部を実装した。Java を用いることによって、様々な実行環境に対応でき、Web 上のコンテンツとして動作させることも容易に行える。Java は実行速度の面でネイティブアプリケーションに比べて不利ではあるが、関数化画像の描画は高速に行えるのでアプリケーションとしての応答性に問題はない。

関数化画像表示アプリケーションは、ドラッグアンドドロップでスクロール、左右のマウスボタンで拡大縮小、といった機能を備えている。また 4.1 で述べたようなイベントに対する処理機能も備えている。インタラクション情報の付加は、現在のところ専用のエディタなどは用意していないので、プリミティブな指定が必要となる。また、以下に示す関数化

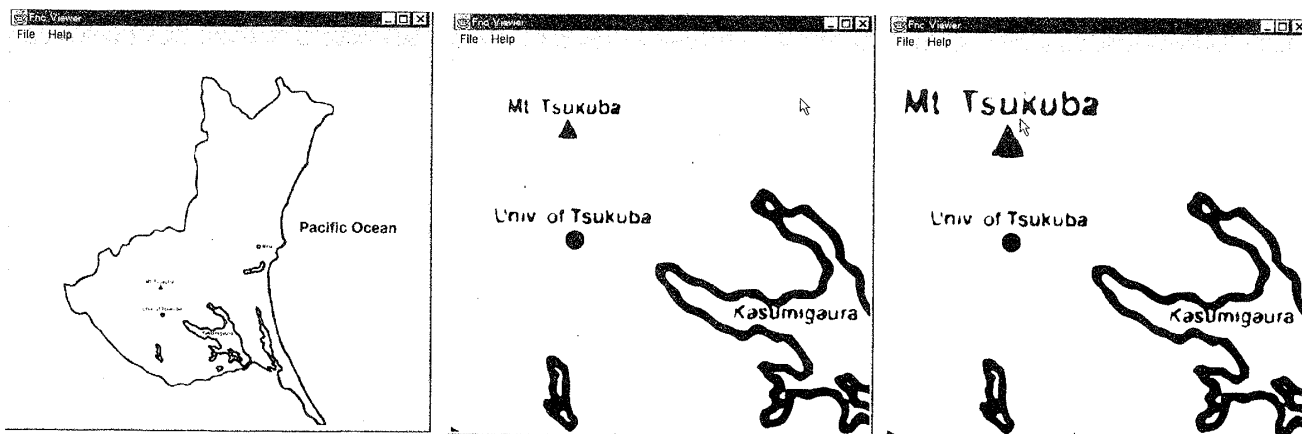


図5 インタラクティブなコンテンツの例

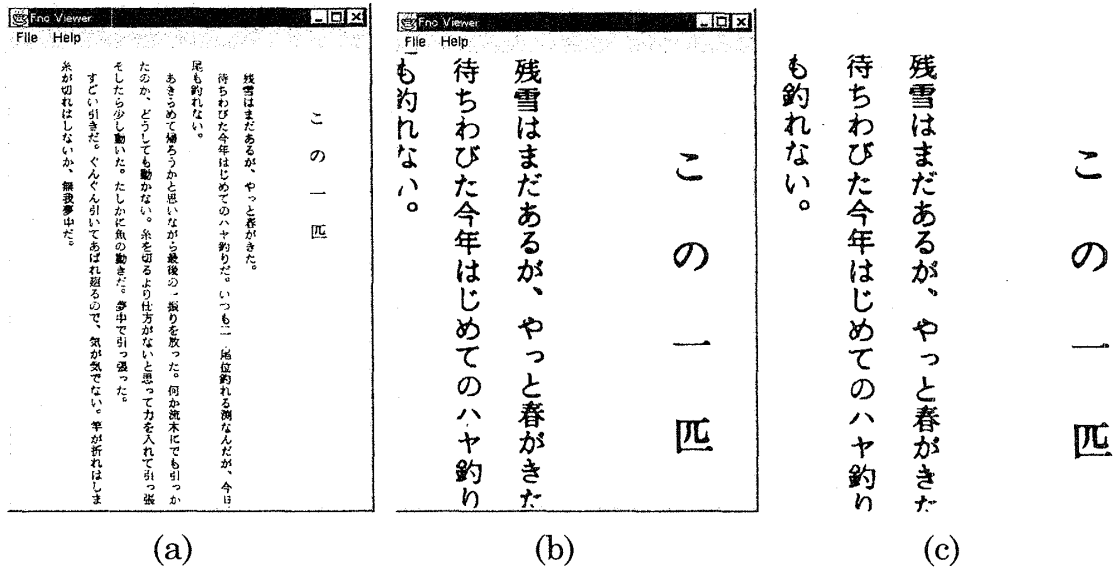


図6 本のデジタル化例

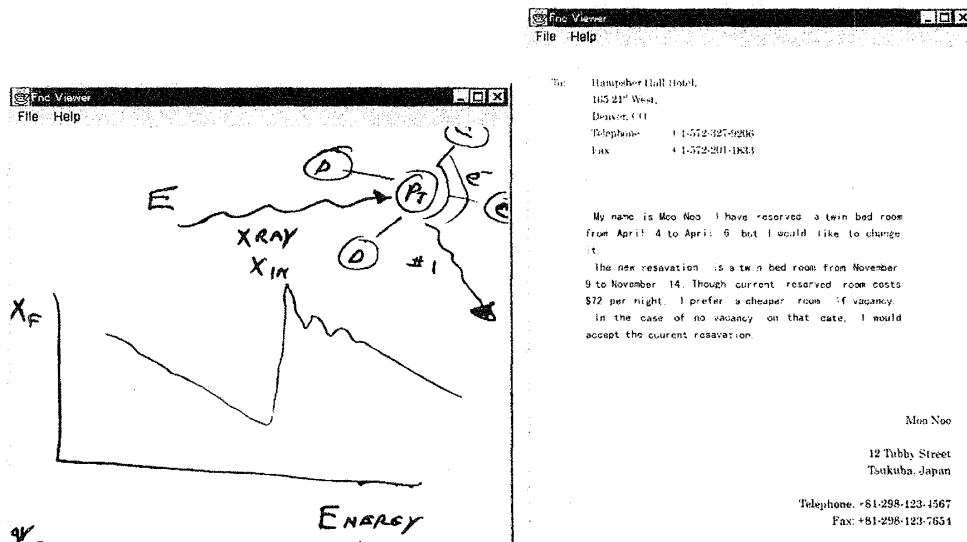


図7 メモ・文書のデジタル化例

画像は、全て紙媒体のものをスキャナで取り込んでデジタル化したものである。

図5は簡単なインタラクティブコンテンツの例である。ここでは地図を題材にしており、特定の領域にマウスポインタが入るとその部分だけ拡大表示される。ハイパーリンクがある部分をユーザーに知らせるときなどには有効である。地図のようなコンテンツでは拡大・縮小表示が必須なので、関数化画像によって紙の地図をデジタル化することは非常に有用であるといえる。

図6は、本の一部を関数化画像によってデジタル化した例である。(a)(b)は画面への出力で、(c)はプリンタへの出力結果である。この例から、関数化画像がひとつの画像ファイルで出力デバイスの解像度を十分に活用できていることが分かる。関数化画像生成は自動で処理されるので、紙媒体として存在する本などの高精度なデジタルブック化が容易に行える。

次は、文書ファイリングを模した例として、メモや文書をデジタル化した例を示す(図7)。メモや通常の文書などでは関数化画像のファ

イルサイズは非常に小さく、多量の文書を保存することに適している。図に示した画像では関数化画像ファイルの大きさはそれぞれ 26KB、68KB となったが、GIF 形式ではそれぞれ 54KB、97KB となった。

## 6. おわりに

本論文では、紙文書から取りこんだ文書画像を関数化図形によってデジタル化する手法について述べ、様々な場面での関数化画像のデジタルコンテンツとして有用性について述べてきた。実装は、ラスタ画像を関数化画像に変換する部分を C++ で、関数化画像の表示やユーザーインターフェースの部分を Java で行った。実際に紙文書をデジタル化し、簡単なインタラクション機能、拡大・縮小、印刷などの実験を行った。これらの実験から、関数化画像が、インタラクティブなコンテンツ、デジタルブック、ファイリングシステムなどに有効な画像表現形式であることが確認された。

本論文ではファイルフォーマットや表示アプリケーションなどは独自に開発したものをを用いているが、SVG 画像として扱っていくことも検討している。標準となるであろう SVG に対応することでより広範囲な応用が見込める。

現段階ではインタラクション情報の付加はプリミティブな方法で行っているが、今後は編集アプリケーションを開発し、より一体化したデジタルコンテンツ作成システムを構築していく予定である。

## 参考文献

1. Macromedia Flash,  
<http://www.macromedia.com/>
2. W3C Scalable Vector Graphics (SVG),  
<http://www.w3.org/Graphics/SVG/>
3. Paul L. Rosin : "Technique for Assessing Poltgonal Approximation", IEEE Trans. Pattern Analysis and Machine Intelligence vol. 19, No. 6, pp. 659-666, 1997
4. Lejun Shao, Hao Zhou : "Curve Fitting with Bezier Cubics" , Graphical Models and Image Processing Vol. 58 No.3, pp. 223-232, 1996
5. 寅市和男, 関田巖, 森亮一 : 「高品質文字フォントの自動圧縮」, 電子情報通信学会論文誌 D, Vol. J-70-D, No.6, pp. 1164-1172, 1987
6. 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, 1999
7. 堀内隆彦, 大瀧保広, 寅市和男, : 「マルチフォントの自動関数化における接合点の多段階抽出法」, 電気学会論文誌 C, Vol. 113 No.12 , pp. 1136-1143 , 1993