マンガの超高圧縮符号化に関する検討

河村　圭[†]　渡辺　裕[††]　富永　英義[†,††]

† 早稲田大学理工学部　〒 169–8555 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
〒 168–0051 東京都新宿区西早稲田 1–3–10 29–7 号館
E-mail: †{kei,tominaga}@tom.comm.waseda.ac.jp, ††hiroshi@giti.waseda.ac.jp

あらまし　マンガは線画を中心に構成されているため，ベクトル化が有効な符号化手法である．しかし，単純なベクトル化は，網点の存在により逆に符号量の増加をまねく．また，網点を含んだままの解像度変換はモアレの発生原因となる．そこで，本稿では，マンガをベクトル表現する際に網点部分を分離して多階調近似し，残りの線分と共に符号化する手法について検討する．

キーワード　マンガ，網点，ベクトル化，画像符号化

Study on Super-Heigh-Compression Coding for Comics

Kei KAWAMURA[†], Hiroshi WATANABE[††], and Hideyoshi TOMINAGA[†,††]

† Dept. of Elec., Info. and Comm. Eng., Waseda Univ., 3–4–1 Okubo, Shinjuku–ku, Tokyo 169–8555 Japan
†† Graduate School of GITS, Waseda Univ., 29–7 Bldg., 1–3–10 Nishi–Waseda, Shinjuku–ku, Tokyo 169–0051 Japan
E-mail: †{kei,tominaga}@tom.comm.waseda.ac.jp, ††hiroshi@giti.waseda.ac.jp

Abstract　Vectorization is an effective technique for comic image coding, since comics are mainly consist of line drawings. However, an existence of halftone-dots causes an increase of coding bitrate if simple vectorization is used. In addition, moires occur when a resolution of the image with halftone-dots is changed. In this paper, we propose a new technique to achieve highly efficient comic image coding. First, we separate the area of halftone-dots and line drawings from an image. Then, a continuous tone approximation is applied to the area of halftone-dots. Next, the conventional vectorization is applied to line drawings. Finally, these two components are mixed together.

Key words　Comics, Halftone-dots, Vector Convert, Image coding

1. はじめに

近年のコンピュータの発達により，ディスプレイ上で文書や画像を閲覧することが可能となり，商用のマンガ配信サービスが数多く開始されている．また，電子ペーパーの開発も盛んに行われており，電子ブックリーダーの未来形として新しい媒体が生まれつつある．

マンガを効率良く配信するためには，画像のデータ圧縮が必須である．現状では JPEG を用いた圧縮方式が主流である．しかし，本来 2 値画像であるマンガを多値画像として扱っていること，また，JPEG は自然画像を対象としていることから，線画中心のマンガを JPEG で符号化することは効率が悪いといえる．

我々は以前より，コンテンツオリエンテッド符号化という概念を提唱している．これは，コンテンツの特性に合わせた符号化を行う，という考え方に基づいている [1]．マンガやアニメーションをはじめとする線画中心の画像には，ベクトル表現を用いた符号化が適していると考えられる．

本稿では，マンガをベクトル表現に変換することを目的とし，マンガの特性に特化した符号化方式について検討する．

2. マンガの特性

本稿で扱う “マンガ” について，符号化に影響を与える特性について述べる．

2.1 原　　稿

商用のマンガは B4 用紙を用いてその中に A4 大の原稿を描いている．雑誌では A4 サイズを B5 サイズに縮小して印刷している．また，単行本では A4 サイズを B6 サイズに縮小して印刷している．

2.2 オフセット印刷

マンガはオフセット印刷によって印刷されている．紙にインキが着くか，着かないかによって原稿を再現する．また，網点を用いることで階調を再現する．これに対して，グラビア印刷では，インキの厚さによって階調を再現できる [2]．

2.3 網　　点

網点は，網目のように張り巡らされた小さな点によって階調を再現するオフセット印刷の手法である．ひとつひとつの点に濃淡をつけることはできないので，なんらかの方法で階調を 2 値画像に変換しなければならない．

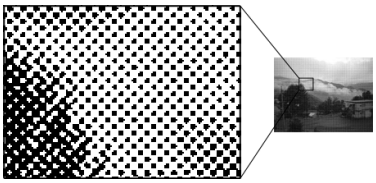

図 1　網点画像の拡大図

印刷業者が使用するイメージセッタや写真印刷機などの高価な印刷機は，点の大きさを変えることができる．そこで，網状になった点に大小の違いを持たせて階調を再現する．図 1 に網点画像の拡大図を示す．

一方，レーザープリンタでは 1 つ 1 つの点の大きさを変えることはできない．そこで，一定の大きさの点を，位置や並び方（粗密）を変化させて階調を再現するディザ法（誤差拡散法）を用いる [3]．

2.4 線画とスクリーントーン

マンガは主に黒色の線画によって構成されている．また，スクリーントーンを使って階調を表現する．

スクリーントーンとは，プラスチックシートなどの透明な薄いフィルムに網点などを印刷し，接着剤を塗布した地紋シートである [4]．網点濃度が均一なスクリーントーンや，グラデーションのスクリーントーンが市販されている．

貼り付けたスクリーントーンの表面をナイフで削り，質感を出す技法がある．

2.5 網点の密度（網点面積率）

網点がすき間なくある状態を 100%（ベタ），網点が全くない状態を 0%，網点とすき間の量が半分ずつの状態を 50%という．網点面積率を指定する場合，以前は 10%刻み程度であった．DTP（Desktop Publishing）が導入されてからは，0〜100%まで 1%単位で指定できる．しかし，印刷機や紙，インキなどの調節が難しく，印刷で再現するのは困難である．細かくても 5%単位にとどめるが一般的である [4]．

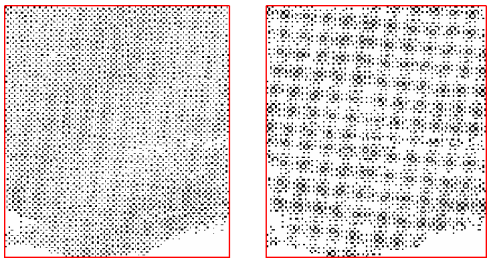

図 2　モアレ（左：300dpi，右：200dpi）

2.6　スクリーン線数

網点の粗さを決めるもので，1 インチの中に網点が何列並んでいるかをいう．スクリーン線数の数値が大きいほど網点は細かくなり，なめらかな階調表現が可能になるが，刷版，印刷の工程での管理が難しくなる．印刷可能なスクリーン線数は，使用する紙にも左右される．新聞などの粗い紙は 80〜100 線，カタログなどの 1 色部分は 133〜150 線，4 色部分は 175 線が一般的である．

2.7　モ　ア　レ

網点を縮小処理したとき，本来の模様とは異なる大きな周期構造をもつ模様が観察される現象のことである．印刷された写真をスキャナなどで読み取る場合にも発生し，画質を低下させる．

同じ網点の画像を，解像度を変えてから拡大した画像を図 2 に示す．全く異なる模様に変わることがわかる．

2.8　白と黒のランダム性

G4 圧縮方式をはじめとする従来の 2 値画像符号化方式は，ディザ処理や網点化された 2 値画像には向いていない．これは白と黒の画素が不連続であるのでランレングス法が適さず，また，白と黒の画素の出現確率がランダムであるのでハフマン法を適用しにくいためである．JBIG2 においては，画素パターンを順次辞書に登録し，再利用する手法によりこの問題を克服している．

3.　マンガの符号化に関する従来手法

現在，マンガ配信サービスの基礎となっている技術はラスター形式の画像符号化方式である JPEG と，各端末にあわせて画像を縮小する解像度変換である．

また，SVG（Scalable Vector Graphics）形式や Flash 形式などのベクトル形式や，それらを扱うツールなど，ベクトル表現の環境が普及してきた．

3.1　ラスター形式

ラスター形式の画像符号化方式を用いた現状の手法と，その問題点を整理する．

3.1.1　JPEG による配信の場合

本来，オフセット印刷は 2 値画像である．しかし，ディスプレイの表示解像度は PC の場合で 72dpi 程度なので，多値画像として表示する方が視覚的には高品質である．

原稿が残されていないマンガも存在することから，現状では製本されたサイズ，すなわち B6 サイズのマンガをスキャンしていると考えられる．2 値画像で扱う場合，図は一般的に 300〜400dpi で十分といわれている．また，多値画像で扱う場合は，150〜200dpi である（4.1 JPEG の解像度に対する性能評価参照）．商用サービスに耐えうる品質，すなわちブロックノイズやモスキートノイズなどが発生しないように画像を圧縮した場合，1 ページ 200KB 程度となる．この値は，典型的な電子コミックのファイルサイズとページ数の関係から算出される．単行本の場合 200 ページ程度なので，1 冊で 40MB になる．また，週刊誌の場合 500 ページ程度なので，1 冊で 100MB になる．

配信された画像は，端末のディスプレイサイズに合わせて縮小処理（解像度変換）がなされる．PC のディスプレイの場合，1/2〜1/3 程度に縮小処理されて表示される．

3.1.2　従来手法の問題点

本来，オフセット印刷によって印刷されるマンガは 2 値画像である．同じ解像度の場合，2 値画像は多値画像に比べて格段に符号量が少ない．また，ハードコピー化，FAX 電送，蓄積などとの相性も良い．

ディスプレイの表示解像度はスキャナに比べて低い．このためスキャナをはじめとするコンピュータ入力機器から得られる画像を，紙面と同程度のサイズでディスプレイに表示するためには，解像度変換（縮小処理）が不可欠となる．

しかし，通常の 2 値画像から 2 値画像への縮

小処理を行ったのでは，線分の欠落や，線幅の相対関係が保存されないなどの問題が生じる．また，網点の縮小処理はモアレの発生原因となる．そこで，多値画像としてマンガを取り込み，符号化することで，線分の欠落や線幅の相対関係が保存されないこと，モアレの発生は低減できるが，配信時のファイルサイズが大きくなることは避けられない [5].

3.2 ベクター形式

高解像度の 2 値画像をベクトル表現に変換し，表示する際に解像度変換を行うことは有効な手法として考えられる [6].

フォントデータをベクトル化する研究がされ，実用化されている．また，ビットマップを PC のメモリ上に全て展開することが可能となり，ベクター形式を処理する環境が普及してきた．

3.2.1 AutoTrace

本ソフトウェアは，ラスター形式の画像をベクター形式の画像に変換するものである．GNU GPL の元で配布されている．ソースコードが自由に入手できるソフトウェアの中で最も変換精度が高い [7].

既存の画像入出力ライブラリとリンクすることで，数多くのフォーマットに対応している．デフォルト出力形式は EPS 形式である．ただし，Flash 形式への対応は十分でない．

ドローツール Tgif のプラグインとして開発された歴史から，内部で用いる曲線の関数としては，ベジエ曲線が用いられている．

入力パラメータの 1 つとして，近似誤差（Error-Threshold）e がある．これにより，近似精度を変えることが出来る．近似誤差 e と出力されるファイルのサイズの関係を，図 3 に示す．近似精度を上げる，すなわち近似誤差 e を小さくするとデータ量が増える．デフォルト近似誤差 e は 2.00 である．この値は，画素数に依存し，解像度という概念はない．

3.2.2 Flash

Flash は Macromedia 社が開発した，音声やベクトル表現のアニメーションを組み合わせて Web コンテンツを作成するソフト，または作成されたコンテンツである．インターネット上での配布を考慮して作られたため，冗長性は少ない．

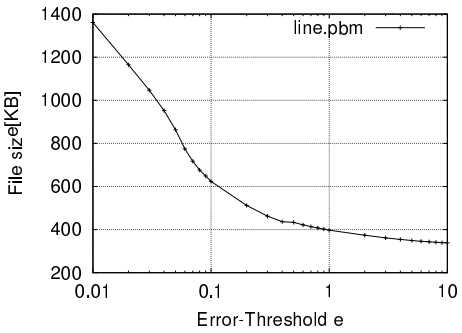

図 3　近似誤差 e と EPS 形式ファイルサイズの関係

Version 6 からは ZIP 圧縮がかけられるようになった．

Flash 形式のファイルを閲覧するためには，Web ブラウザに専用のプラグイン “Flash Player” をインストールしておく必要があるが，標準でインストールされているため普及率は高い．ファイルフォーマットは公開されている [8]. 内部で用いる曲線の関数は，B スプライン曲線である．

本稿では，AutoTrace により出力された EPS 形式を Illustrator [9] により Flash 形式に変換する．ただし，EPS 形式内のベジエ曲線を Flash 形式内の B スプライン曲線で近似する際の近似精度は 10 段階中 7 とする．

3.2.3 曲線の関数について

図形を関数化する研究が数多くなされている [10], [11]. 曲線の近似に用いられる関数曲線にはいくつか種類がある．現在，商用アプリケーションにおいて特に利用されているのが（3 次）ベジエ曲線と（2 次）B スプライン曲線である．

NURBS（非一様有理 B スプライン Non-Uniform Rational B-Spline）というベジエ曲線や B スプライン曲線を包括する曲線がある．精度を要する大規模な 3 次元データを扱う CAD の分野では標準的であるが，一般的な CG ではあまり普及していない．

ベジエ曲線は，2 つの通過点と 2 つの制御点によって 1 つの区間曲線を定義する，3 次関数の曲線である．PS（Post Script）や EPS（Encapsulated Post Script）の基本曲線となっている．

3 次関数を用いるので，少ない制御点で多くの表現が可能であるが，1 つ 1 つの区間曲線については計算コストがかかる．

B スプライン曲線は，2 つの通過点と 1 つの制御点によって 1 つの区間曲線を定義する，2 次関数の曲線である．TrueType フォントの基本曲線となっている．2 次関数を用いるので多くの通過点，制御点が必要となるが，1 つ 1 つの区間曲線についての計算コストは小さい．また，ディスプレイをはじめとする低解像度へのラスタライズに関する研究，実用化がなされている．

ベジエ曲線を B スプライン曲線で完全に置き換えることは出来ない．これは，関数の次数が異なるためである．

4. ラスター－ベクター変換による符号化

本節では，従来手法である JPEG と，提案手法の核となるベクトル表現について，ファイルサイズに関する予備実験を行う．なお，ベクトル表現としては，AutoTrace により出力される EPS 形式を Flash 形式に変換し，これを比較対象とする．

以下の評価に際して利用した原稿の概観を図 4 に示す．

4.1 JPEG の解像度に対する性能評価

広く流通している JPEG 圧縮ソフトウェア [12] においては，入力パラメータの 1 つに品質パラメータ Q がある．デフォルトでは品質パラメータ $Q = 75$ である．B6 サイズの原稿をグレイスケール画像として取り込み，JPEG による圧縮を行う．いくつかの品質パラメータ Q に対して，解像度とファイルサイズの関係を調べた．結果を図 5 に示す．

JPEG による圧縮は解像度が上がるにつれ，ファイルサイズも増加することがわかる．ファイルサイズは解像度の 2 乗，すなわち画素数に比例している．

また，典型的な電子コミックのファイルサイズ，すなわち 200KB に対応する解像度は，150〜200dpi であると考えられる．解像度 166dpi，品質パラメータ $Q = 75$ の画像をディスプレイに収まるように縮小して閲覧したところ，十分鑑賞が可能であった．

図 4　入 力 画 像

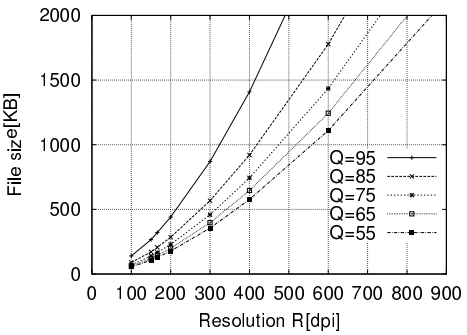

図 5　JPEG における解像度とファイルサイズの関係

4.2 ベクトル表現の解像度に対する性能評価

B6 サイズの原稿を 2 値画像として取り込み，ベクトル表現に変換する．いくつかの近似誤差 e に対して，解像度とファイルサイズの関係を調べた．結果を図 6 に示す．

ベクトル表現は解像度が上がるにつれ，ファイルサイズも増加することがわかる．300dpi 程度からは解像度に比例している．これは，近似誤差

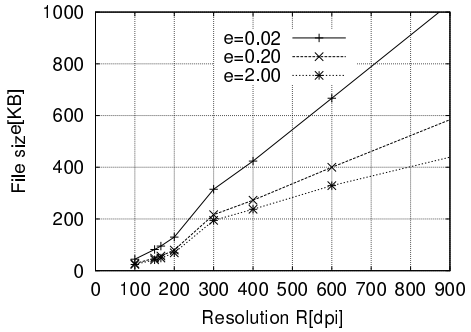

図 6　ベクトル表現における解像度とファイルサイズの関係

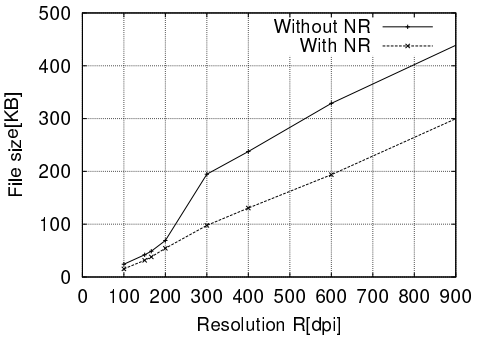

図 7　ノイズ除去の有無による解像度とファイルサイズの関係

e が画素数依存のため，解像度が上がるにつれ，相対的に近似誤差 e は小さくなるためであると考えられる．

300dpi 程度を境に，低解像度では急激にファイルサイズが小さくなっている．これは，網点や極細線などの微少な要素が消えるためである．

4.3　ノイズ除去の有無によるベクトル表現の性能評価

網点などの微少な要素が，ベクトル表現の符号化効率を著しく落としている可能性がある．そこで，2 値画像として取り込む際に孤立点除去などのノイズ除去（NR，Noise Reduction）を導入し，網点などの微少な要素を除去する．B6 サイズの原稿を 2 値画像として取り込み，ノイズ除去を施した後，ベクトル表現に変換する．ノイズ除去をした場合としない場合について，解像度とファイルサイズの関係を調べた．ただし，近似誤差 $e = 2.00$ とした．結果を図 7 に示す．

あらゆる解像度で，ファイルサイズが解像度に比例することが確認できる．また，ノイズ除去を施さなかった場合に比べ，300dpi 程度からは一定のファイルサイズが削減されていることがわかる．

2 値画像，300〜400dpi のとき，ノイズ除去ありでベクトル表現のファイルサイズは 100〜130KB である．

4.4　各手法の相互比較

JPEG，ベクトル表現でノイズ除去の有無の 3 通りについて，代表的な解像度，品質パラメータ

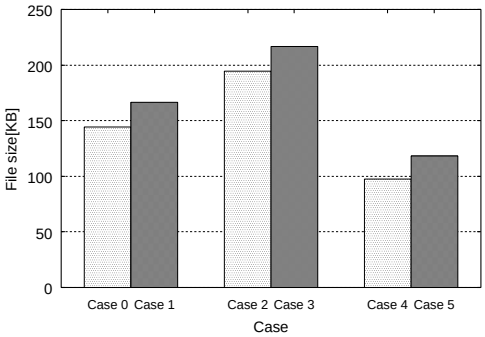

図 8　各手法の相互比較

Q，近似誤差 e を用いてファイルサイズの比較をする．ただし，JPEG の場合はグレースケール画像，ベクトル表現の場合は 2 値画像である．結果を図 8 に示す．

各ケースは以下の通りである．Case 1，2 の共通条件は，“JPEG，グレイスケール，166dpi” である．Case 1 は “品質パラメータ $Q = 65$”，Case 2 は “品質パラメータ $Q = 75$” である．

Case 3〜6 の共通条件は，“ベクトル表現，2 値画像，300dpi” である．Case 3 は “ノイズ除去なし，近似誤差 $e = 2.00$”，Case 4 は “ノイズ除去なし，近似誤差 $e = 0.20$”，Case 5 は “ノイズ除去あり，近似誤差 $e = 2.00$”，Case 6 は “ノイズ除去あり，近似誤差 $e = 0.20$” である．

JPEG のファイルサイズが 150KB に対して，ノイズ除去なしでベクトル表現のファイルサイズは 200KB である．網点を含む画像の場合，単

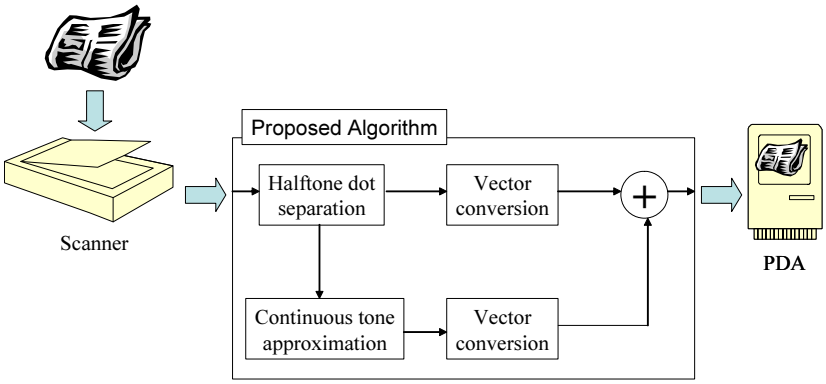

図 9　提案システム

純なベクトル表現への変換は有効とはいえない．しかし，ノイズ除去ありでベクトル表現は，約半分の 100KB となる．微少な要素を含まない画像に対して，ベクトル表現は有効である．

5.　網点を考慮した符号化手法

4.4 で示したように，網点を除去してからベクトル表現にすることで，ファイルサイズをより小さく出来る．網点を単純にベクトル表現にすることは有効ではない．また，網点を含んだままでは，解像度変換の際にモアレが発生し，画質を低下させる．そこで，網点を分離することを検討する．さらに，ベクトル表現や網点の仕組みと性質を踏まえた上で，ラスター表現独特の網点をベクトル表現の枠組みで扱う手法を検討する．

5.1　提案システム

提案システムの概要を図 9 に示す．

既に印刷，出版されたマンガを，市販のスキャナで 2 値画像として取り込む．次に，ベクトル表現に変換し，SVG 形式や Flash 形式などのベクター形式で出力する．これを PDA（Personal Digital Assistant）や電子ペーパーなど，解像度やディスプレイサイズの異なる環境で閲覧する．

提案システムにより，現在 JPEG 形式で 200KB 程度に圧縮されるような原稿を，高解像度の 2 値画像としてスキャナから取り込み，64KB に圧縮することを目標とする．

提案システムを利用することで，500 ページ程度の雑誌は 32MB となる．将来，無線 LAN などにより数秒で端末に転送することが可能となる．

5.2　提案アルゴリズム

提案システムを実現するため，以下のアルゴリズムを提案する．網点をベクトル表現で扱いやすくするために，階調近似を行う．

（1） 網点を含む 2 値画像から，網点を抽出し，網点画像と線画像に分離する．

（2） 網点画像に対して階調近似を行い，階調近似画像を作る．

（3） 階調近似画像と線画像をそれぞれベクトル表現に変換し，階層化する．

5.3　網 点 分 離

提案アルゴリズム（1），網点の分離可能性について実験を行った．

分離手法として，網点が孤立点である性質を利用する．まず，画素の連結条件を 8 近傍としてラベリングをする．次に，しきい値以下の面積を持つ画素を網点であると判定する．ただし，しきい値は固定値で，手動で与える．網点と判定された画素は網点画像へ，残りの画素は線画像として分離する．

入力画像を図 4，線画像を図 10，網点画像を図 11 に示す．解像度は 300dpi，2 値画像である．

本実験により，網点は網点画像へ，網点以外は線画像へ分離されることが確認できた．

一方で，いくつかの課題が明らかとなった．

網点以外の短い細線が網点画像に分離される．本実験では，線は線画に分離されるべきである．しかし，8 近傍の連結条件下にいおいて細切れに

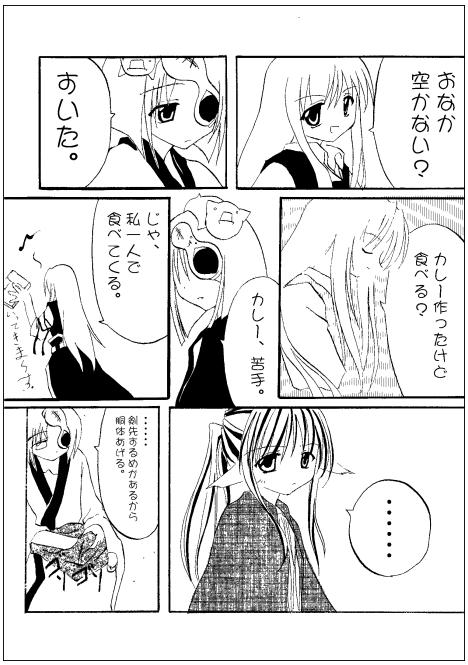

図 10　線　画　像

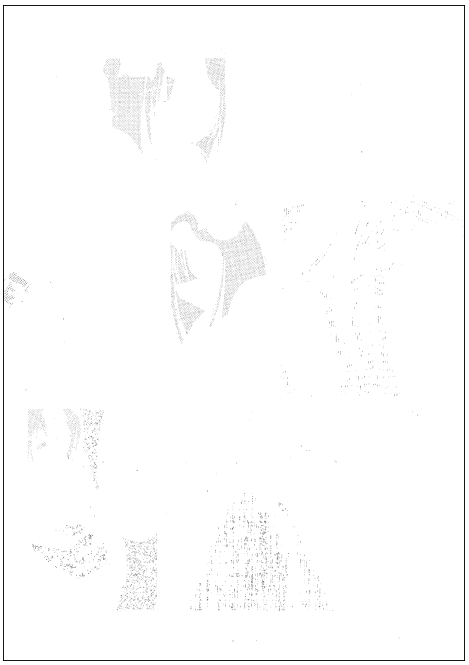

図 11　網 点 画 像

なるような線は，構成する画素数がしきい値以下のため網点と認識される．細切れの線が網点画像に分離されない手法を検討する必要がある．ただし，微少な要素はベクトル表現に向かないので，必要に応じて細切れになった線を接続するなど，別の手法を検討する必要もある．

網点は印刷時の劣化により，複数個が連結してしまう “ブリッジ” が生じることがある．このような場合，視覚的には連結していることが認識されないが，構成する画素数としてはしきい値より大きいので，線画として分離される．

本実験では，しきい値を画像に合わせて手動で与えた．解像度やスクリーン線数などの条件が変わると，しきい値を変える必要がある．

黒の孤立点のみを網点として分離した．そのため，網点面積率が大きな網点，すなわち黒地に白い孤立点は分離されない．また，網点面積率が 50%前後の網点は，白と黒の画素がそれぞれいくつか連結して構成されているため，孤立点という性質を持たない．

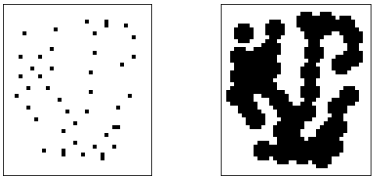

図 12　膨 張 処 理

5.4　網 点 近 似

提案アルゴリズム（2），分離した網点画像の多値画像への変換可能性について実験を行った．

まず，孤立点である網点を，図 12 に示すような膨張処理を行うことで大きな領域とする．1 つ 1 つの領域を網点領域と呼ぶ．ただし，膨張回数は手動で与える．次に，網点領域毎に含まれる網点の面積（画素数）を求め，網点領域に対する網点面積率を算出する．網点面積率を網点領域の階調とみなし，塗りつぶす．階調で近似した画像を，階調近似画像と呼ぶ．

網点領域画像を図 13，階調近似画像を図 14 に示す．また，線画像と階調近似画像を重ね合わせ

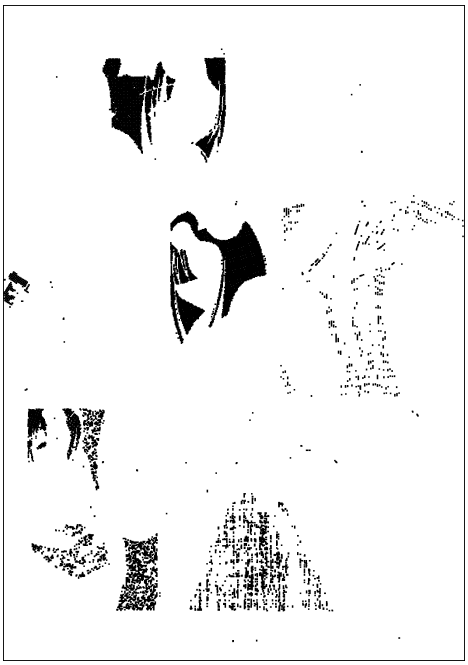

図 13　網点領域画像

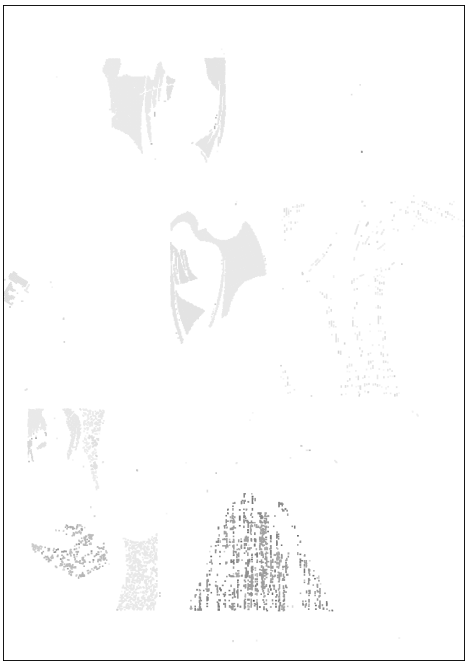

図 14　階調近似画像

た画像，再構成画像を図 15 に示す．

本実験により，網点画像を多値画像に変換できることが確認できた．

一方で，いくつかの課題が明らかとなった．

網点領域の定義が曖昧である．一般に，網点の境界を特定することは難しい．また，網点領域は必ずしも線画によって囲まれていない．

網点面積率を階調に対応づけるときに，線形であると見なした．しかし，オフセット印刷ではブリッジやかすれ，ドットゲインなどが生じる．さらに，原稿に対して印刷物は，一般に階調が狭くなる．網点面積率と階調の関係を一意に与えることは難しいので，柔軟に調整できるようにする必要がある．

本実験では，網点領域毎に均一の階調とした．しかし，グラデーション網点や作家が網点に削り作業を行った場合，均一の階調ではない．

5.5　ベクトル変換と階層化

提案アルゴリズム（3），階調近似画像と線画像をそれぞれベクトル表現に変換し，階層化し，Flash 形式で出力する実験を行った．ベクトル表現への変換は AutoTrace を用いた．近似誤差 e はデフォルトの 2.00 である．階層化，Flash 形式への変換は Illustrator を用い，曲線の近似精度は 10 段階中 7 とした．

AutoTrace の出力を階層化したベクトル表現画像を図 16 に示す．また，線画像を Flash 形式に変換した時のファイルサイズは 97KB，同様に階調近似画像は 53KB，両者を階層化した画像は 151KB となった．従来手法の JPEG 圧縮とファイルサイズを比較した場合，同程度となる．

本実験により，網点を含む画像をベクトル表現に効率的に変換できることが確認できた．

しかし，文字やキャラクターの髪型のように複雑な図形と，吹き出しやキャラクターの胴体のように単純な図形の両方を含むマンガに対して，AutoTrace では十分なベクトル表現の変換精度が得られないことが明らかとなった．

6. ま と め

本稿では，マンガの特性に合わせて符号化する手法について検討した．網点を分離し，階調近似

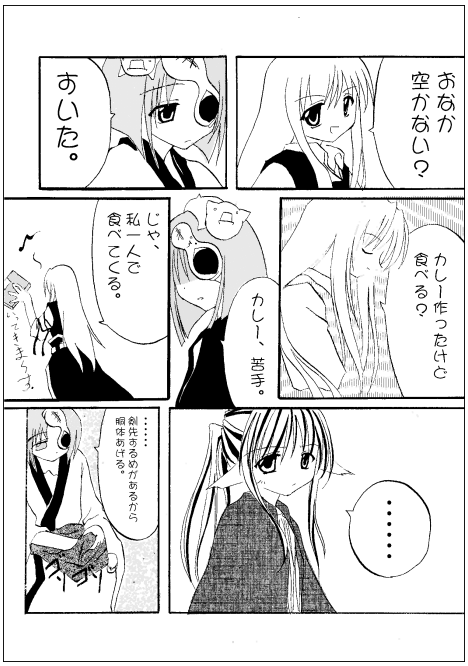

図 15　再構成画像

図 16　ベクトル表現画像

した後，線画と共にベクトル表現に変換する手法を提案した．そして，いくつかの実験により，提案手法が有効であることが確かめられた．

提案システムの目標値である 1 ページ 64KB の圧縮を達成するには，既存のベクトル変換ソフトである AutoTrace の改良が必要である．

今後は，実験を通して明らかとなった網点の分離や階調近似における課題に取り組む．また，本手法に適した評価基準の開発も課題である．

参 考 文 献

[1] O. Nakagami, T. Miyazawa, H. Watanabe, H. Tominaga, “A Study on two-layer coding for animation images,” IEEE Int. Conf. on Multimedia Expo (ICME) 2002, WedAmPO3: Compression II, Aug. 2002.

[2] 佐柳和男，“デジタルプリントと網点，” 印刷雑誌，Vol.66，1983.

[3] 阿部淑人，“網点の基本と最新動向，” 印刷雑誌，Vol.85，2002.

[4] 日本印刷学会，“増補版印刷辞典，” 印刷学会出版部，1991.

[5] 吉田雅之，渥美栄司，高橋利至，“ソフトコピー表示における 2 値画像の多値化縮小処理，” 三菱電機技法，1998 年 9 月号.

[6] “TrueType Reference Manual,” http://developer.apple.com/fonts/TTRefMan/

[7] “AutoTrace,” http://autotrace.sourceforge.net/

[8] “Macromedia Flash File Format (SWF) Specification,” http://www.macromedia.com/software/flash/open/licensing/fileformat/

[9] “Illustrator,” http://www.adobe.co.jp/

[10] R. Haruki, T. Horiuchi, “Scalable Image Coding by Spline Approximation for a Gray-scale Image,” Proc. 4th Int. Conf. Document Anal. and Recog., vol1, pp.407-411, 1997.

[11] 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抽出法，” 電学論 C，113 巻 12 号，1993.

[12] “Independent JPEG Group,” http://www.ijg.org/