

携帯電話を利用したコミックの閲覧システムとその評価

山田雅之¹ 鈴木茂樹² ラフマツト・ブディアルト³ 遠藤守¹ 宮崎慎也¹

¹ 中京大学情報科学部 ² 中京大学大学院情報科学研究科 ³ School of Computer Sciences, USM

概要 本論文ではコミックを携帯電話端末上で閲覧できるシステムについて述べる。表示画面のピクセル数が十分でない携帯電話端末でコミックを提供するためには、コミックをコマ毎に分割して表示したり、台詞のテキストを抜き出して拡大表示したりする工夫が必要である。そのためには、コミックからのコマフレームの抽出、整列、ならびにフレーム画像内の台詞の抽出、整列等の処理が必要であり、本文では特にそれらの処理を実行するためのアルゴリズムを提案する。また、実際にコミックのデータを Web サイトからダウンロードし、携帯電話上でコミックを再構成するプログラムを実装し、携帯電話の画面に実際に表示される画像のクオリティやコンテンツとしてのコストパフォーマンスについて検証する。

キーワード：コミック，携帯電話，画像処理，レイアウト解析，データ表現

A System for Reading Comics on Cellular Phones and its Evaluation

Masashi Yamada¹ Shigeki Suzuki² Rahmat Budiarto³ Mamoru Endoh¹ Shinya Miyazaki¹

¹ School of Computer and Cognitive Sciences, Chukyo University

² Graduate School of Computer and Cognitive Sciences, Chukyo University

³ School of Computer Sciences, USM

Abstract : This paper presents a system for reading comics on cellular phones. It is difficult to display high resolution images in a low resolution cellular phone environment. It is necessary for comic images to be divided into frames and the contents such as speech text to be displayed at a comfortable reading size. We have developed a scheme how to decompose comic images into constituent elements frames, speech text and drawings. We implemented a system on the internet for a cellular phone company in our country, that provides downloadable comic data and a program for reading.

Keywords: Comic, Cellular phone, Image processing, Layout analysis, Data representation

E-mail: ¹{myamada,endoh,miyazaki}@sccs.chukyo-u.ac.jp ²ssuzu@katch.ne.jp ³rahmat@cs.usm.my

1. まえがき

近年、携帯電話は高機能化、高性能化、インターネットとの親和性の向上により、テキスト、音声、音楽、静止画、動画など Web 上の様々なメディアが利用可能になった。中でも携帯電話を画像表示端末とする活用方法が盛んに模索されている。コミックもその一つであり、コミック雑誌や単行本を持ち歩く手間がなく、携帯電話のみでいつでもどこでもコミックを閲覧できる利便性のもつ可能性は大きい。しかしながら、このようなアプリケーションを実現する上では電話機の携帯性や消費電力を考慮する必要があるため、表示可能な画像サイズに関する制約が問題となりやすい。これはコミックを

携帯電話で閲覧しようとする場合にも大きな制約となる。

コミックは通常 A4 サイズや B5 サイズ、文庫本サイズの紙に高解像度印刷されている。これを携帯電話の画面で閲覧するためには、コミックをコマ毎に分割して表示したり、台詞のテキストを抜き出して拡大表示したりする工夫が必要となる。そのためには、コミックからのコマフレームの抽出、整列、ならびにフレーム画像内の台詞の抽出、整列等の処理が必要となる。本文ではそれらの処理を自動もしくは半自動的に実行するためのアルゴリズムを提案する。

また、実際にコミックのデータを Web サイトからダウンロードし、携帯電話上でコミックを

再構成するプログラムを実装し、携帯電話の画面に実際に表示される画像のクオリティやコンテンツとしてのコストパフォーマンスについて検証する。

1.1. 既存の電子コミックサービス

電子コミックを配信し、端末側で閲覧できるサービスは既にいくつか存在する。ここでは代表的なサービスを紹介する。10daysbook[1]はPC上で閲覧できる電子コミックを配信しており、現在 4000 以上の作品を扱っている。利用者はページ単位のコミック画像を閲覧できる。Bitway-books[2]は PDA 端末上で閲覧できる電子コミックを配信しており、現在 80 程度の作品を扱っている。閲覧できる画像はページ単位ではなく、PDA 画面にあうよう切り出した画像である。Mangalian[3]は 2003 年 6 月で閉鎖されたが、それまで携帯電話向けのサービスを行っていた。配信されるコミックは携帯電話用に作られた 4 コマ漫画であった。

我々の研究で対象とする紙ベースの通常のコミックは膨大なコンテンツの蓄積があるが、このようなコミックを対象とした携帯電話向けサービスはまだない。

1.2. 他の関連研究

コミックを題材とした研究は、まだまだ多くはないが、以下のようなものが代表的なものとして挙げられる。

山田らはコミック画像から人物を抽出する方法を提案している[4]。しかしながら、一般に人物の自動抽出アルゴリズムに汎用性を持たせるのは難しいと考えられる。低解像度でのコミック閲覧を考えた場合には、人物などの構成要素よりもむしろコマごとに分割表示のためのフレーム抽出や拡大表示のための台詞抽出の自動化が重要な課題となる。

Kurlander らはコミックチャットというシステムを提案している[5]。このシステムは予め用意した人物や吹き出しのパターンを使って、チャットデータからコミックを自動生成するも

のであり、このために必要となるのは画像合成の機能のみである。

一方、我々の研究では、携帯電話の搭載メモリ量や通信コストを考慮した効率の良い画像圧縮および展開方法の実現が主目的となる[6][7]。画像の圧縮方法については GIF, JPEGをはじめとして、すでに多くの手法が提案されている。また、近年では統計的手法を画像圧縮に利用した方法も提案されている[8]。本研究で提案するデータ形式では、コミック画像が一般に白黒の 2 値画像で、主にペンや筆で描かれたストロークが基本単位となっていることを利用している。

2. システム概要

ここでは、本システムで仮定するコミックの構造およびシステム全体の概要を述べる。

2.1. コミックの構造および表現方法

コミックはページから成り、各ページはフレームから成る。各フレームは絵部と台詞部からなる。この構造を次のように表現する。

コミックはページの集合 P から成る。ページ $p \in P$ はフレームの集合 F から成り、 F は順序をもち、 F の要素 $f_i \in F$ は一列化できる：

$$f_1 < f_2 < \dots < f_n,$$

ここで記号 " $<$ " は順序関係を表し、 $A < B$ は A が B より前に読まれるべきフレームであることを表す。各フレームは台詞部をもち、台詞部は台

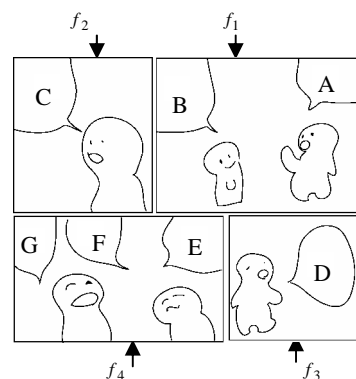


図 1: ページの例

詞の集合 S で表す. 同様に S は順序をもち, S の要素 $s_i \in S$ は一列化できる:

$$s_1 < s_2 < \dots < s_m,$$

フレームは台詞部の他に絵部をもつ. よってフレームは絵部 d と台詞部の組 (d, S) で表現できる.

例 1: 図 1 にコミックのページの例を示す. このページは 4 つのフレームをもつ. コミックが右開きの場合には, これらのフレームの順序は $f_1 < f_2 < f_3 < f_4$ となる. フレーム f_4 の台詞部は 3 つの台詞の集合 $\{E, F, G\}$ である. これら台詞の順序は $E < F < G$ である. 絵部はフレームから台詞を取り除いたものでフレーム f_4 の絵部は



である. これを d_4 で表すとフレーム f_4

は絵部と台詞部の組 $(d_4, \{E, F, G\})$ で表現できる.

2.2. システムの概要

図 2 にコミックの画像から携帯電話用データを生成する手順とこのデータを Web サイトからダウンロードし携帯電話上でコミックを再構成するシステムの概念を示す.

まず, コミックの各ページをスキャナーで読み込み, 十分な解像度をもつ 2 値画像を得る. 各ページはいくつかのフレームから構成されており, この画像をフレームに分割して時間順に整列する. 次に, 各フレームは絵部と台詞部に分離する. 絵部は圧縮したデータに変換し, 台詞部は OCR でテキストデータに変換する. こ

うして得られるフレームのデータと, 端末側でフレームを再構成するプログラムを Web サイトで提供する. ユーザは携帯電話からこれらをダウンロードし, コミックを閲覧する. 一度データを取得すれば無接続状態で何回でも閲覧できる.

3. フレームの整列

システムは携帯電話用データを自動生成するため, ページをフレームに分割し, これを整列する. 新聞や事務文書のレイアウトを解析する方法は既にいくつか提案されているが, これらは構成要素が長方形であると仮定している. 一方, コミックのフレームは長方形のみでなく任意の凸多角形をとり得るため通常のレイアウト解析は適用できない. そこで, ここでは

- i) フレームは凸多角形である
- ii) フレームは輪郭線をもつ
- iii) コミックは右開き本形式である

を仮定する. i), ii) はここで扱えるコミックを限定する条件である. iii) は本文での説明を容易にするためのものである. 本章ではこれらの仮定に基づいたフレーム整列方法を述べる.

フレームの整列は, 隣接する 2 フレーム間の整列ルール R1, R2, R3 および隣接する 3 フレーム間の順序に制約を与えるルール R4 の 4 つのルールに基づいて行われる.

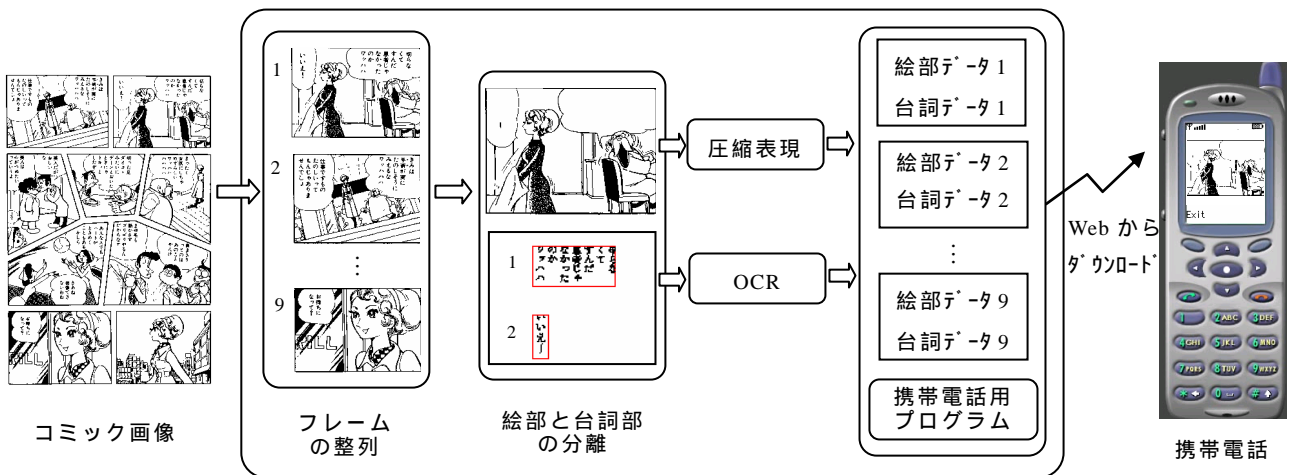


図 2: システムの概念

3.1. 2フレーム間の整列ルール

まず、隣接する2つのフレームの間に分割線を仮定し(図3), 分割線に向きを与える. この向きはベクトル (x, y) で表し, $y > 0$ または $(1, 0)$ を満たすように与える. この向きにむかって, 分割線の右手側を右フレーム, 左手側を左フレームと呼ぶ. ルール R1 は角度 α に基づき右フレームと左フレームの順序を決める. ここで α は x 軸と分割線のなす角度である.

$$(R1) \quad 90^\circ \leq \alpha < 180^\circ \Rightarrow rf < lf.$$

ここで rf と lf はそれぞれ右フレームと左フレームを表す.

例2: 図4(a) はルール R1 より $rf < lf$ となる.

ルール R2 と R3 は $0^\circ \leq \alpha < 90^\circ$ かつ隣接する2つのフレームの輪郭線が共通直線上にあるときの順序を決める. この共通直線を基底線と呼ぶ(図3). このとき, フレームの輪郭線には反時計回りの向きを仮定し, 基底線には同じ方向の向きを与える. ルール R2 と R3 は2つの角度 α と β を使い, 隣接する2つのフレームを整列する. ここで β は x 軸と基底線のなす角度で

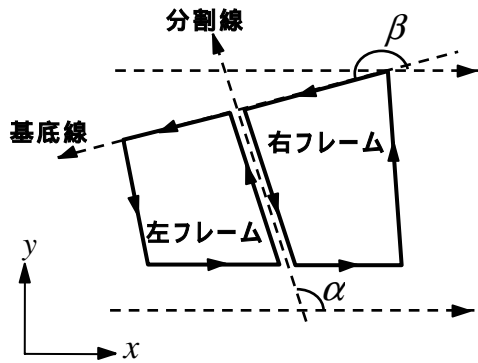


図3: 分割線と基底線

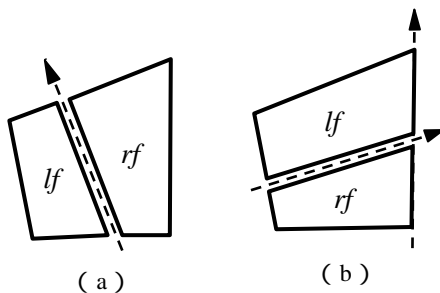


図4: 隣接する2フレームの整列例

ある. R2 と R3 は次のように定義する.

$$(R2) \quad 0^\circ \leq \alpha < 90^\circ \wedge \alpha \leq \beta \leq 135^\circ \Rightarrow lf < rf.$$

$$(R3) \quad 0^\circ \leq \alpha < 90^\circ \wedge 135^\circ < \beta < \alpha + 180^\circ \Rightarrow rf < lf.$$

例3: 図4(b) はルール R2 により $lf < rf$ となる.

3.2. 3フレーム間の順序制約ルール

次に図5のようにT型隣接する3つのフレームの順序制約ルールを定義する. 図5においてフレームBとCは共通の基底線を持ち, フレームAはBとCの両方に隣接する. このT型隣接を $t(A, B, C)$

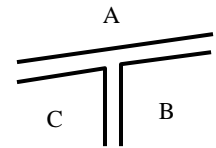


図5: T型隣接 $t(A, B, C)$

または $t(A, C, B)$

で表し, 順序制約ル

ール R4 を次のよう

に定義する.

$$(R4) \quad t(f_1, f_2, f_3) \Rightarrow (f_1 < f_2 \wedge f_1 < f_3) \vee (f_1 > f_2 \wedge f_1 > f_3).$$

以上のルール R1, R2, R3, R4 により得られる隣接フレーム間の順序を以降では単に隣接フレーム順序と呼ぶ.

例4: 図6に5つのフレームをもつページ例を示す. これらフレームについて隣接フレーム順序は次のようになる. ルール R1 より, $C < E, C < D, B < D, E < D$ が得られる. ルール R2 より, $A < C$ が得られる. ルール R3 より, $C < B$ が得られる. 3つのT型隣接があり, $t(A, C, B)$ についてはルール R4 より $(A < C \wedge A < B)$ または $(A > C \wedge A > B)$ が成り立つ. ここで R2 より $A < C$ となるのがすでに分かっているので $A < B$ となる.

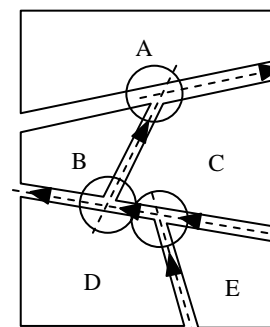


図6: T型隣接と分割線

3.3. フレーム整列のための探索

ルール R1 から R4 で得られた隣接フレーム順序に基づき、全てのフレームを整列する。ここでは各フレームをノードに見立てた深さ優先探索を用いる。この探索はあるフレームから隣接フレームを順次たどってフレーム列を生成し、隣接フレーム順序を全て充たす列が見つかった時点で探索を終了する。一般に隣接フレーム順序を充たすフレーム列は1つではないが、深さ優先探索により最初に見つかった列を解とする。適切な列が得られるよう、隣接フレームが複数あるときの探索順序をある優先順位に従い決定する。この探索の詳細は付録で述べることにし、ここでは探索の結果得られるフレーム列の例を示す。

例 5: 図 7 に示すページの例では、図 7(a)からは列 $B < A < E < D < C < G < F < I < H$ が、図 7(b)からは列 $F < B < C < D < E < H < I < A < G$ が探索結果として得られる。

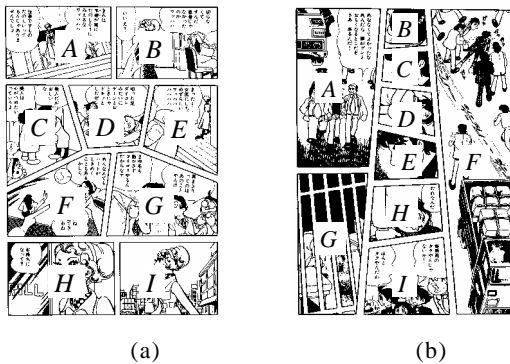


図 7: 探索のためのラベリング

4. 台詞の整列

1つのフレームの中に複数の台詞がある場合、これらには順序があり、台詞の位置関係に基づき整列できる。

フレームの画像から個々の台詞を見つけるには、まず、フレームをスキャンし、ある正方形領域を見つける。この領域は黒画素を含み、かつ、その黒画素がその領域以外の黒画素と連結しないような領域とし、領域の大きさは文字の大きさに合わせ予め指定しておく。見つけ出

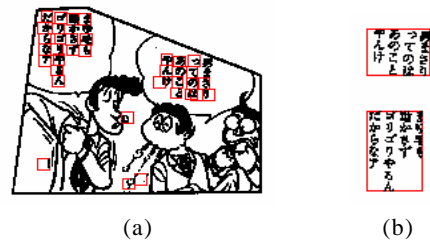


図 8: 台詞の領域抽出の例

した領域を距離に基づきグループ化し、各グループを囲む長方形領域を求め、それを1つの台詞領域とする。図 8(a)のフレームからは図 8(b)のように2つの台詞領域が求まる。台詞の整列にはこの台詞領域の右上端の位置を使う。

台詞を整列するには、まず、全ての台詞について、その台詞と他の台詞との順序を求める。2つの台詞の台詞領域を s_a , s_b とし、 $P(s_a)$ と $P(s_b)$ をそれぞれの右上端の位置とする。また、 $dist(p)$ をある位置 p とフレームを囲む最小長方形の右上端との距離とする。2つの台詞の順序を次のルール R5 により定める。

(R5) もし $\vec{V} = P(s_b) - P(s_a)$ が $\vec{v}_1 = (1,1)$ と $\vec{v}_2 = (-1,1)$ のなす 90 度角の間にあるなら、 $s_b < s_a$, そうでなく、もし $dist(P(s_b)) < dist(P(s_a))$ なら $s_b < s_a$, そうでもなければ $s_a < s_b$.

全ての2台詞間の順序を求めた後は、フレームの整列の場合と同様の方法により整列する。

5. 携帯電話用データ

本章では、フレームの絵部と台詞部について、携帯電話用のデータの表現方法とその表現への変換方法を述べる。

5.1. 絵部のデータ

まず、絵部のデータ表現について述べる。このデータを電話回線を使って端末側にダウンロードすることからデータサイズは小さいことが望ましい。また、携帯電話上でこのデータから絵部を再構築することから、データ構造は単純で高速に処理できることが望まれる。近年、GIF 形式に対応した携帯電話が普及しており、GIF 形式の画像を高速にレンダリングできるが、

データサイズの点で問題がある。これらの問題を解決するため、ここではコミックの特徴を考慮したデータ表現を提案する。

コミックの絵部は白地の紙にペンや筆で描かれるため白領域が多い。一方、コミックは高解像度印刷されペンや筆のストロークが滑らかに表現されている。これを携帯電話など低解像度の画面環境で再現するには白黒の2値では不十分であり、濃淡値によるアンチエイリアジング表示が望ましい。これらの理由から以下に述べる方法で絵部を表現し携帯電話用データを生成する。まず、絵部の画像を携帯電話の画面サイズにあわせ、縦、横どちらか大きい方が128画素になるよう相対縮小する。この過程ではバイキュービック法を用いて2値画像を濃淡画像に変換し、各画素の濃度値を1バイト表現する。次にこの濃淡画像を2ⁿ画素単位でブロック化し、白画素のみのブロックとそれ以外のブロック(濃淡ブロックと呼ぶ)に分ける。濃淡ブロックに含まれる画素の濃度値は1バイト表現から白(w)、グレー1(g1)、グレー2(g2)、...、グレーn(gn)、黒(b)のように数段階のインデックス表現に変換する。このようにして得られる、ブロックデータと濃淡ブロックの濃度データを絵部の携帯電話用データとする。

例 6: 絵部の大きさを横12×縦3画素とし、横4画素を1ブロックとした場合のデータ表現例を図9に示す。ブロックデータでは4画素全てが白であるブロックを1、濃淡ブロックを0で表す。濃淡ブロック内の各画素の濃度値は2ビットインデックスw,g1,g2,bで表す。

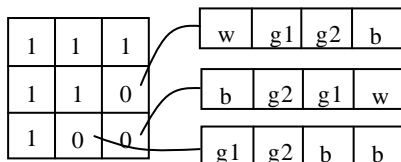


図9: ブロックデータ(左)と濃淡ブロックの濃度データ(右)

5.2. 台詞部のデータ

台詞部は次のように変換し、テキストファイ

ルで表現する。まず、4章で述べた方法でフレームの台詞領域を抽出する。そのあと台詞領域をOCRにかけ、台詞のテキスト(文字コード)を生成する。次にフレームごとに1つのテキストファイルを用意し、各行に台詞の内容を記録する。さらに、台詞がフレームのどこにあったかわかるように、各行の先頭には台詞領域の中心位置を記録する(図10参照)。

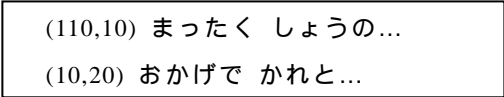


図10: 台詞部のデータファイル

6. システムの実装と評価

本研究ではシステムを試作し、評価実験を行った。本章ではシステムの実装方法と評価について述べる。

6.1. 携帯電話用プログラム

携帯電話用プログラムはJava言語を使って実現する。このプログラムは1) イベント処理、2) 絵部の表示処理、3) 台詞の表示処理を行う。

イベント処理ではユーザのボタン操作に従い、表示するフレームを切り替える。また、1つのフレームに複数の台詞がある場合はボタン操作に従い、その表示も切り替える。

絵部の表示処理では、現在のフレームの絵部を絵部データから生成し表示する。ボタン操作によるフレームの切り替えが起こるたびに、その都度、絵部データから絵部を生成表示するこ

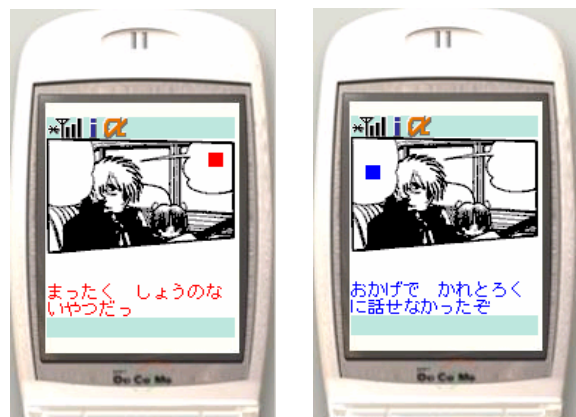


図11: 携帯電話用プログラムの実行例

とにより、携帯電話のメモリ使用を抑える。提案した絵部データ表現は比較的単純な構造なことから、生成表示が高速に行える。

台詞の表示処理では、台詞部のデータファイルを参照し、各行に記録された台詞を携帯電話画面の下段に表示する。この際、文字を明瞭にするため携帯電話のフォントを使う。また、各行の座標値を使って、台詞の位置にマークを表示する。ユーザはフレーム上の台詞位置をそのマークで知ることができる。

図 11 にはエミュレータを使ったときのプログラム実行例を示す。

6.2. Web によるアプリケーションの提供

Web を通してユーザに携帯電話用プログラムとコミックのデータを提供する。ここでは携帯電話の通信料金やメモリ量の制限を考慮し、次の方式を採用した。

コミックのデータはページ単位とし、1 ページの全フレーム（絵部データと台詞部データ）をまとめたものとする。携帯電話用プログラムとコミックデータは1つのアプリケーションにまとめられ、サーバ上に格納される。ユーザは携帯電話から所定の Web サイトをアクセスし、ページ毎にこのアプリケーションをダウンロードする。閲覧するときはダウンロードしたアプリケーションを端末上で起動することにより、コミックデータを展開表示する。



図 12: 携帯電話でコミックを閲覧する様子



12Kbyte



2Kbyte



15Kbyte



3Kbyte

図 13: 圧縮前の絵部(左)と再構築された絵部(右)

6.3. 評価

本研究では手塚治虫著、「ブラックジャック」、第1巻、第6話(全23ページ)[9]を使って、携帯電話用データの生成と携帯電話での閲覧実験を行った。図 12 には実機を使った実験の様子を示す。

まず、全23ページについて提案方法によるフレームの整列を試みた結果について述べる。このうち3ページはフレームの輪郭線がないなどの理由から自動でフレームに分割できないため対象外とした。残りの20ページについてフレームの整列を行い、正しい列が得られた。

次に、本論文で提案した絵部のデータ表現について評価する。図 13 に2つの絵部例を示す。左側は元画像を携帯電話画面にあわせ縮小して得られた濃淡画像であり、これらのデータサイズは上下それぞれ12Kバイトと15Kバイトである。右側は、左側の画像を提案したデータ表現に変換し、さらにそのデータから携帯電話用プログラムにより再構築した絵部である。このときの絵部データのサイズは上下それぞれ2Kバイトと3Kバイトである。絵部データを作る

際、濃淡画像の濃度値をインデクス値に変換する。ここではインデクス値は2ビット、すなわち4色 w,g1,g2,b で表現した。

ダウンロードするデータの大部分は絵部のデータが占める。表1には1話分、1アプリケーション(1ページ分データと携帯電話用プログラム)、および、1フレームのサイズを示す。現在、携帯電話で1度にダウンロードできるアプリケーションサイズは制限されており、本実験で使った携帯電話では30Kバイトまでとなっている。一方、我々の携帯電話用プログラムは3Kバイト、これにコミックのデータを合わせたアプリケーションのサイズは8Kバイトから19Kバイトの範囲内であり、23ページ全てについて制限サイズを下回った。

次に、絵部データから再構築される絵部の表示について述べる。本研究では絵部を2値画像として表現する方法も試みたが、表示の際、エリアジングが目立ち、劣化が激しかった。一方、本論文で述べたデータ表現を用いた場合、再構築される絵部(図13右参照)はエリアジングが目立たず、比較的劣化が少ない。なお、絵部の表現へ変換する際のパラメータの値と画質との関係を明らかにする必要があるが、これは今後の課題としたい。

次にコミック提示方法の評価について述べる。我々の提示方法では、フレームや台詞を読む順に表示する。このため、フレームや台詞を進めるボタン操作で簡単に閲覧でき、ストーリーも十分理解できることを第三者による閲覧実験で確認した。しかしながら本提示方法では、元のページに対して、a)フレームがどのように配置されていたか、b)フレームがどのくらいの大きさかという情報は閲覧者に提示しておらず、これまでのコミックの形態と異なるため、違和感がある。これについては a), b)の情報も何らかの形で提示することを検討したい。また、b)に関連して、現在全てのフレームを携帯電話の画面サイズに縮小しているため、もともと大き

いフレームほど、縮小率が大きくなり、見づらくなる場合がある。これについては、大きいフレームの場合は細分割するなどの方法を検討する必要がある。一方、台詞については、抽出し、携帯電話用フォントで表示しており、読みやすさを維持している。

最後に PDA コミックサービスで広く使われている閲覧システム RoidTime2[10]と我々のシステムとの比較について述べる(表2参照)。まず、画像の切り出し過程は RoidTime2 の場合、手動で行っているが、我々はフレームの抽出・整列の自動化を試みた。コミックデータの取得方法については、PDA は各種ストレージを利用できるが、携帯電話はまだストレージに対応するものが少ないため、本研究では電話回線を利用することを仮定した。台詞の表示については、RoidTime2 は抽出せず画像の一部として表示する。一方、我々のシステムは抽出し携帯電話フォントで表示する。1フレームのサイズについては、濃淡画像の場合、RoidTime2 では約50Kバイト(解像度240×320)であり、我々のデータ表現では約2Kバイト(解像度128×128)となった。

表1：データサイズ

	サイズ(Kbyte)
1話	311
1アプリケーション	8~19
1フレーム	2

表2：PDA コミックとの比較

	RoidTime2	提案システム
画像抽出・整列	手動	自動化
データ取得	ストレージ	電話回線
台詞表示	画像のまま	抽出し、携帯電話フォントで表示
フレームサイズ	50Kbyte (240×320)	2Kbyte (128×128)

7. むすび

本論文では、コミックを携帯電話で閲覧する

ため、コミックを構成要素に分解してデータ化し、携帯電話上でコミックを再構成する方法を提案した。まず、コミックの画像をフレームに分割し、それらを整列してから携帯電話用のデータ表現へ変換する方法を述べた。また、このデータから携帯電話上でコミックを再構成するプログラムについて述べた。提案した方法を実装するシステムを作成し実験を行った結果、データのサイズは許容範囲であった。また、携帯電話でコミックの内容を理解でき、絵部のクオリティも維持できることを確認した。

今後はより多くの種類の作品を使って評価実験を行い、コミックの表示方法等の改善を図りたい。その他の課題として、大きいフレームを見やすくするためのフレームの細分割方法やPDAなど他の小型端末による閲覧方法などがあげられる。

謝辞

第19回 NICOGRAPH 論文コンテストならびに本論文の論文審査において貴重なコメントを下さいました査読者の方々に感謝いたします。また、本研究の遂行にあたり、作品の利用、論文への転載を快諾くださいました株式会社手塚プロダクションのご厚意に深謝いたします。本研究の一部は文部科学省私立大学ハイテク・リサーチ・センター補助金による。

文献

- [1] 株式会社イーブックイニシアティブジャパン, 10daysbook, <http://www.10daysbook.com>.
- [2] 凸版印刷, bitway-books, <http://books.bitway.ne.jp>.
- [3] MBEAT.COM, Inc, mangalian, <http://www.mangalian.com>.
- [4] T. Yamada and T. Watanabe, "Identification of Person Objects in Four-Scenes Comics of Japanese Newspaper", Proc. of International Workshop on Graphics Recognition, pp.152-159, 1999.
- [5] D. Kurlander, T. Skelly and D. Salesin, "Comic Chat", Computer Graphics Proceedings, SIGGRAPH, pp.225-236, 1996.
- [6] M. Yamada, K. Amano and S. Miyazaki, "An Image Transformation Method for Reading Comics on Cellular Phones", Proc. of 6th IASTED International Conference, Internet and Multimedia Systems and Applications, pp.230-235, 2002.

- [7] 山田雅之, 鈴木茂樹, Rafmat Budiarto, 遠藤守, 宮崎慎也, "携帯電話を利用したコミックの閲覧システム", 第19回 NICOGRAPH 論文コンテスト論文集, pp.93-98, 2003.
- [8] D. Chen, P. Seshadri, "An Algebraic Compression Framework for Query Results", Proc. of International Conference on Data Engineering, pp.177-188, 2000.
- [9] 手塚治虫, ブラックジャック, 1巻, 秋田書店, 1993.
- [10] 株式会社ロイド, Roid Time Net, <http://www.comictime.net>

付録

フレーム整列探索について

ページの全フレームを1列に順に並べるため、深さ優先探索を用いる。付図1にこの探索の手続き $FSS(U, L, f_c)$ を示す。この探索はあるフレームからスタートし、隣接するフレームを順次たどりフレーム列を作る。このとき、フレームの重複を防ぐため、既にフレーム列に含まれるようなフレームはたどらない。ここではフレーム列に含まれていないフレームを未探索フレームと呼び、この集合を U で表す。フレーム列が本論文で述べた隣接フレーム順序を充たすか否かは常にチェックされ、全てのフレームを含むフレーム列で隣接フレーム順序を充たすものが見つければ、その時点で探索を終了する。またフレーム列が隣接フレーム順序を充たさないと分かった時点で、他のフレーム列を探索する。

フレーム列を作るとき、あるフレーム f_c が複数の隣接フレームをもつなら、それら隣接フレームに優先順位を与え、その順にたどる。この優先順位は次のように定める。本論文ではフレームは輪郭線もち、凸多角形と仮定していることから、各隣接フレームはフレーム f_c の輪郭線(多角形)の各辺に対応する。隣接フレームに優先順位を設けるため、以下のような操作を行う。まず、 f_c の輪郭線に反時計回りの向きを与える。輪郭線の各辺の向きはこの輪郭線の向きに従うものとし、 x 軸となす角度が 225° 以上かつ 225° に一番近い辺を1番目とし、他の辺は1番目の辺から順に反時計回りに順位が与えられる。隣接フレームの優先順位は対応する辺の順位が与えられる。ここでは、フレーム f_c に隣接する未探索フレームの集合を N とし、フレーム $f \in N$ が N の中

で最も高い優先順位をもつとき f を $hn(N, f_c)$ で表す . }
 探索は , まず , あるフレーム f_c の次に隣接フレーム }
 が続くようなフレーム列を作るが , このようなフ }
 レーム列が隣接フレーム順序を充たさないと分かっ }
 たときは , 次のような探索を行う . 全ての未探索フ }
 レーム $f \in U$ について , f とページ右上端との最小 }
 直線距離を求め , この距離が未探索フレームの中で }
 最も小さいフレームを求める . ここではこのような }
 フレームを $md(U)$ で表す . 探索は f_c の次に $md(U)$ が }
 続くフレーム列を作り同様の処理を行う . }
 end

付図 1: フレーム整列探索の手続き

あるページ p のフレーム列を求めるときは予め ,
 どのフレームから探索するかを決めてから付図 1 の
 手続き呼び出す . 本研究では , 最右上のフレームを
 1 番目のフレームと考え , 常に手続きを
 $FSS(F - \{f\}, [f], f)$ のように呼び出し実験を行った .
 ここで F はページ p の全てのフレームの集合 ,
 $f = md(F)$ である .

付図 1 において , 関数 $satisfy(L)$ はフレーム列 L が
 全ての隣接フレーム順序を充たせば真を返す .
 $[L | f_n]$ はフレーム f_n を L に後方から追加して得ら
 れるフレーム列を表す .

procedure $FSS(U, L, f_c)$:

U : a given set of frames
 L : a list of frames (frame sequence).
 f_c : a given frame.

begin

```

if (  $U = \phi$  ) {
    print_solution(L);
    return true;
}
else{
     $N$  : a set of frames neighboring  $f_c$  .
     $N = N \cap U$  ;
    while(  $N \neq \phi$  ){
         $f_n = hn(N, f_c)$  ;
         $N = N - \{f_c\}$  ;
        if(  $satisfy([L | f_n])$  and  $FSS(U - \{f_n\}, [L | f_n], f_n)$  )
            return true;
    }
     $f_n = md(U)$  ;
    if (  $satisfy([L | f_n])$  and  $FSS(U - \{f_n\}, [L | f_n], f_n)$  )
        return true;
    else
        return false;
    
```