

携帯端末向け画像解像度変換システム

吉川 裕章[†] 高梨 崇広[†] 内田 理^{††}
中里 純二^{†††} 中西 祥八郎^{††}

ネットワークを介して携帯端末に画像の配信を行う際、異なる複数の端末において最適な画像の表示を行うためには、管理者があらかじめ機種ごとに適した画像を用意しておかなければならない。そこで、携帯電話からのアクセスのたびにアクセス解析を行い、端末の解像度情報を取得し、自動的に最適なサイズおよびフォーマットの画像を生成して表示させるシステムを作成した。これによりシステムが使用するサーバー上のディスク領域を減少させ、また管理者の負担を減らすことが可能となる。さらに、ユーザーからの画像の投稿を組み合わせることにより、ユーザー間における最適な画像の交換が実現できた。

Resolution Conversion System for Cellular Phone

Hiroaki Yoshikawa[†] Takahiro Takanashi[†] Osamu Uchida^{††}
Junji Nakazato^{†††} Shohachiro Nakanishi^{††}

In case images are distributed to users of cellular phone via the Internet, administrators have to prepare the optimal images for every model of cellular phone, because there are various kinds of resolution for cellular phones. In this paper, we propose a system which generates the image with optimal size for each model automatically by getting the information about the user's model of cellular phone at every access. Using this system, storage space of the server and loads of administrators can be saved. Moreover, adding the image uploading system for cellular phone users, it is possible to exchange optimal images between users.

1. はじめに

近年、携帯電話やPDAといった携帯端末からのインターネット利用が増えており、特に携帯電話においては今後も利用が増え続けると思われる。携帯電話でインターネットを利

用したときの問題点の一つに画像の表示の問題がある。画像サイズが大きいものを表示させようとすると表示できないことも多く、さらに携帯電話では利用できる画像フォーマットも限られている[1,2,3,4]。

さらに携帯電話の機種によって画像表示領域は異なっており、従来は機種ごとに適切なサイズの画像を用意しておかなければならなかった。しかし、あらかじめ機種ごとに画像を用意するのではなく、携帯電話からのアクセスの度に最適な画像が自動で生成されることが望ましい。

そこで我々は、各携帯電話の持つ機種情報によって画像表示領域サイズを判定し、自動的に

[†]東海大学工学部電気工学科
〒259-1292 神奈川県平塚市北金目 1117
Faculty of Engineering, Tokai University
1117 Kitakaname, Hiratsuka, Kanagawa, 259-1292
Japan
^{††}東海大学電子情報学部情報科学科
School of Information Technology and Electronics,
Tokai University
^{†††}東海大学大学院工学研究科
Graduate School of Electronics, Tokai University

画像のリサイズおよびフォーマット変換を行うシステムを作成した。

2. 携帯電話の解像度・対応フォーマット

図 1 から分かるように、各携帯電話の持つディスプレイの解像度は多様であり、統一されていない。また、対応する画像フォーマットも機種によってそれぞれ異なっている。そこで、携帯電話に最適な画像の配信を行うために、CGI によるアクセス解析を行い、携帯端末の持つ環境変数から解像度の判定を行うことにした。

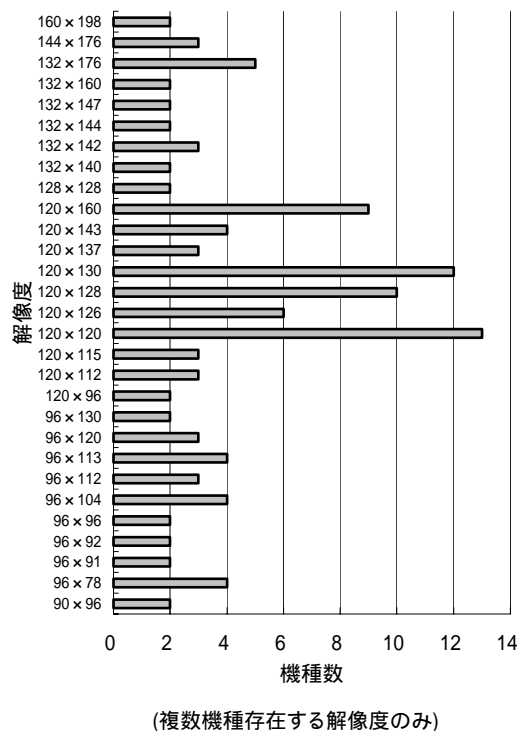


図 1 携帯電話解像度一覧

携帯電話の持つ代表的な環境変数およびその内容は以下の通りである[5]。
(表示例：au / C5001T)

HTTP_X_UP_DEVCAP_ISCOLOR

表示結果：1

端末がカラー表示可能機種であることを示す。“1”であればカラー表示可，“0”であればカラー表示不可となる。

HTTP_USER_AGENT

表示結果：KDDI-TS21 UP. Browser/6.0.2.276
(GUI) MMP/1.1

ブラウザ情報および端末情報を示す。“TS21”はデバイス ID を指し、機種名称 “C5001T” を意味する。デバイス ID から機種名称への変換には、別途プログラムへの入力作業が必要となる。

HTTP_X_UP_DEVCAP_SCREENCHARS

表示結果：20,8

一画面内に表示できる文字数の情報を示す。この場合は横 20 文字、縦 8 文字の表示が可能であることが分かる。

ENV(HTTP_X_UP_DEVCAP_SCREENPIXELS)

表示結果：144,135

ブラウザ表示時の解像度情報を示す。この場合は横 144 × 縦 135 ピクセルの表示領域を持つことが分かる。

HTTP_ACCEPT

表示結果：application/vnd.phoncom.mmc-xml,
image/gif, image/jpeg, ほか

端末が扱うことのできるデータのフォーマットを示す。この他にもテキストや HTML、着信メロディのデータ等も含まれる。

今回必要な情報は、ディスプレイの解像度（横幅および縦幅）、機種ごとの対応画像フォーマットである。そこで各キャリアの持つ環境変数について、携帯電話の種類別に調査を行った（表 1）。

表 1 携帯電話の持つ環境変数

キャリア	機種名	解像度情報
DoCoMo(i-mode)		×
au(Ez-web)	×	
J-PHONE (J-sky)		
TU-KA(Ez-web)	×	

表 1 から分かるように、DoCoMo 以外の携帯電話は環境変数として解像度情報を持つ

が、DoCoMo の端末のみ解像度情報を持たない。そこで、DoCoMo の端末に関してはあらかじめ機種ごとの解像度情報をチェックし、プログラム内にインプットしておき、アクセスの度に参照する処理が必要となる。そのため、新機種が発売される度にプログラムの更新が必要となる。

また、各キャリア(DoCoMo, au, J-PHONE, TU-KA) が対応している画像のフォーマットは以下の通りである。(2003年1月現在)

- DoCoMo : JPEG , GIF**
- au : JPEG , GIF , PNG**
- J-PHONE : JPEG , PNG**
- TU-KA : JPEG , PNG**

3. システム

今回、アクセス解析にはCGIプログラムの記述に優れている Perl(Perl5.0)を、画像の解像度変換には標準で多数の画像処理に関する Class が用意されている Java(J2SE1.4)を用いてプログラムの作成を行った。

3.1 変換・配信システム

システムの実現にあたり、端末の機種名および解像度を判定するための“phonemodel.pl”、最終的な表示画面となる HTML を出力するための“top.cgi”、画像の解像度変換を行う“Transform.java”を作成した。目標とするシステムの概要を図2に示す。

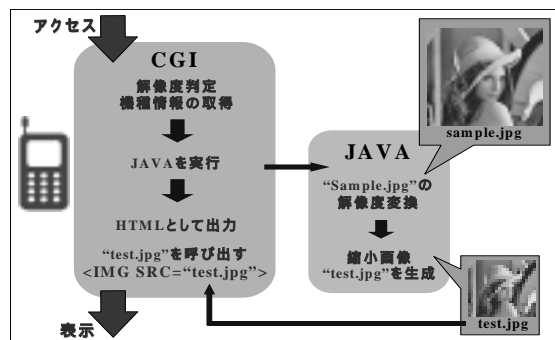


図2 システムの概要

以下にシステムの流れを示す。まず、サーバー側に任意の画像を数種類用意しておき、これに合わせてHTMLで記述したメニューを用意

する(図3)。

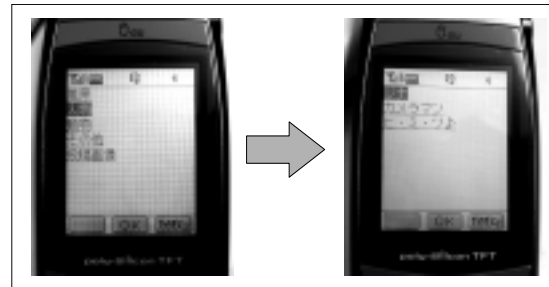


図3 HTMLメニュー画面

メニューから画像が選択されると“top.cgi”が動作し、ここから端末の機種名や解像度情報の判定を行う“phonemodel.pl”が参照される。ここで得た解像度情報によって“top.cgi”は“Transform.java”を起動し、画像の解像度変換を行う。ここで、端末のディスプレイに合わせて画像の解像度変換を行う際、ディスプレイの縦幅・横幅のどちらに合わせて解像度変換を行うかという問題がある。調査の結果、多くの携帯電話では縦方向への画像のスクロール表示は可能であるが、横方向へのスクロールはできないという特徴が見受けられた。これを受け、“Transform.java”では、ディスプレイの横幅に合わせて元画像の解像度変換を行うことにした。

端末に合わせて解像度変換され、生成された画像は一時的に保存され、最終的に“top.cgi”が出力するHTML内でタグによって表示される。

このようなシステムを用いて画像の閲覧を行う際、一度画像が表示された状態から続けて別の画像の表示を試みると、端末にはキャッシュとして最初の画像が残り、画像の更新が行われないという問題が起こる。この問題を防ぐには、HTML中のタグで指定するファイル名部分を画像ごとに変化させ、画像選択の度に新たにデータの読み込みを行う必要がある。そこで、指定した画像をバイナリモードで読み出すCGIプログラム“binmode.cgi”を作成し、これをタグで画像として指定する。更にGETメソッドを利用して、以下のように記述する。

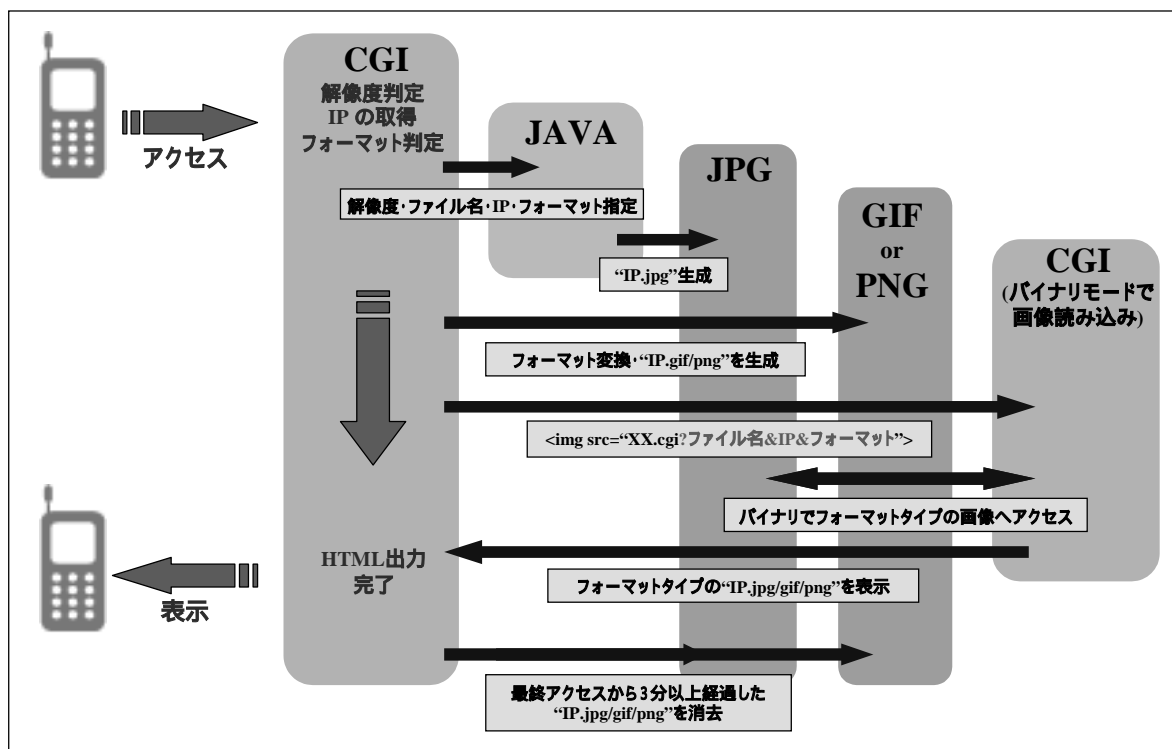


図4 システムの詳細

これにより、“?ファイル名”部が画像選択毎に変化するため、キャッシュ画像が表示されることはない。

また、一時的に生成される画像ファイル名が全ユーザー間で“test.jpg”固定であると、同時に複数のアクセスがあった場合に不都合が起こると想定される。しかし全てのアクセスの度に異なるファイル名の画像を生成・保存すると、アクセスが集中した場合にファイル数が膨大になる危険性も考えられるため、一時的に生成されるファイルの数は1ユーザーあたり最小限に抑えた方が良く考えられる。これらを踏まえ、生成される画像に関してまとめると、以下のことが言える。

- 1) 1ユーザーあたり生成されるファイルは1つが良い
- 2) ファイル名にユーザー固有の文字列が入っていることが望ましい
- 3) 一時的に生成された画像は早めに消去された方が良い

そこで、生成された画像ファイル名にIPアドレスを利用することにし、IPアドレスが“210.136.161.103”の場合、一時画像のファイル名を“210-136-161-103.jpg”とする。

更にファイル数の増加を避けるため、最終アクセスから3分以上経過した画像ファイルの自動消去を行うことにした。これにより、キャッシュおよび、同時に複数のアクセスがあった場合の問題を解決することができた。作成したシステムの詳細を図4に示す。

3.2 画像投稿システム

“3.1 変換・配信システム”において、システム管理者からユーザーへの一方向的な画像配信を実現することができた。ここでユーザーからの画像データの投稿が可能となれば、画像配信システムと組み合わせ、結果としてユーザー同士の相互的な画像の交換(図5)が可能となると考えた。

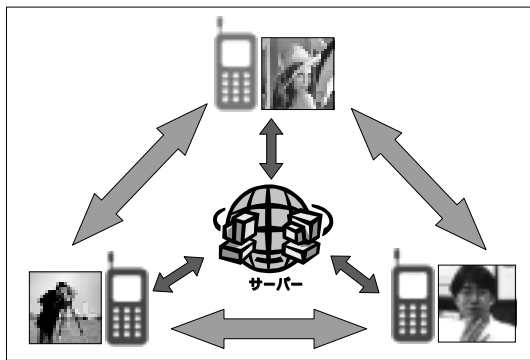


図5 端末間での相互間な画像の交換

PC等での一般的なデータ投稿の手段として、FTPでの転送、CGIによるデータのアップロードなどが挙げられる。しかし今回は携帯電話での利用を主として考えているため、アプリケーションの利用や、CGIを用いた内蔵HDDの参照といった手法は適していない。

そこで、どのユーザーでも簡単に利用でき、データ転送の手段としても一般的な「メールへの添付」を投稿手段として採用することにした。それに伴い、メールからBase64でエンコードされたデータ部分を抽出し、サーバー上の任意のディレクトリ内に保存する「upload.pl」を作成した。このプログラムは、ユーザー投稿専用のメールアドレスにメールが届く度に起動し、メール内から添付された画像データ、画像ファイル名、ファイル容量の抽出および保存作業を行う。

ユーザーから投稿された画像をメニューとして一覧表示し、他のユーザーからも選択可能とするために、ファイルリスト生成プログラム「read.cgi」を作成した。このプログラムは、投稿された画像が保存されているディレクトリ内部のファイル名を読み取り、リスト表示するものである。この中で、画像以外のファイルおよびbmpなど、システムが対応していないファイルや、ファイルサイズの大きい画像などは除去してリスト表示を行う。

リスト表示する際は携帯端末のディスプレイの大きさを考慮し、1ページあたり15件程度のリスト表示に抑え、それ以降は次ページに繰り越す処理を行うことにした。

これでユーザーからの投稿画像を、他のユーザーもリストから選択して表示することが可

能となった。

更に sendmail によるメールの返信を行い、画像を投稿したユーザーに対して最終的な画像表示URLの通知を行う処理を追加した(図6)。



図6 システムから返信されたURL

画像の投稿を行ったユーザーは、システムから返信されるメールに書かれたURLを保存し、これを他のユーザーへ転送することにより、簡単に他のユーザーの持つ携帯端末へ、最適なサイズの画像を配布することができる。

4. システムの検証

本システムを、Red Hat Linux release 6.2を搭載したサーバー上に実装し、検証実験を行った。今回の実験では、携帯端末として、端末1(J-N05：解像度160×180)、端末2(C5001T：解像度144×135)、端末3(N503i：解像度118×128)の3機種を用意し、カメラ付き携帯電話である端末1で撮影した画像を端末2,3で表示させることを試みた。まず、端末1で撮影した画像をメールに添付し、作成したシステムへの投稿を行う。次に、システムから返信されたURLを端末1から端末2,3に転送し、端末2,3からそのURLへアクセスした。その結果、端末2,3において、最適なサイズの画像が表示されており(図7)、システムが良好に動作していることが確認できた。その他の端末においても、一部機種を除き、良好に動作することを確認している。



端末 1



端末 2



端末 3

図 7 表示結果

5. まとめ

今回作成したシステムにより、異なるサイズのディスプレイを持つ多種の携帯端末に対し、適切なサイズに変換した画像を自動で配信することができた。また、画像をメールに添付させることによってユーザーからの画像のアップロードを実現し、ユーザー同士で、随時適切なサイズに変換された画像を交換することが可能となった。

今後の研究課題として、現段階では縦に長い画像の表示を行う際、ブラウザ上で表示可能なファイル容量の上限を超えてしまう可能性があり、それをどのようにして回避するかという問題が挙げられる。

また、このシステムを使用するためには、CGI(Perl5.0)および JAVA(JDK1.4)が許可されている環境が必要であるが、今以上に環境に左右されず、誰にでも手軽に設置できるシステムを構築することが目標である。

謝辞

本研究に対し有益な御助言をいただいた、東海大学電子情報学部情報メディア学科菊池浩明助教授に深く感謝いたします。

参考文献

- [1] NTT DoCoMo ホームページ
<http://www.nttdocomo.co.jp/>, (2003.1)
- [2] au ホームページ
<http://www.au.kddi.com/>, (2003.1)
- [3] J-PHONE ホームページ
<http://www.j-phone.com/>, (2003.1)
- [4] TU-KA ホームページ
<http://www.tu-ka.co.jp/>, (2003.1)
- [5]水野 貴明, “自分で出来る Web アクセス解析とデータ活用術”, ソーテック社, 2002.