

TIFF タグ情報を利用した画像データベース構築

井出健治*, スフー・バトル**, 菊地 章***

デジタルカメラの普及に伴いデジタル画像は非常に身近なものになった。我々の周りにはデジタル画像が多数存在する。これらは学校での授業・特別活動など教育活動の中でも頻繁に利用される。一方、学校単位でこれらデジタル画像の分類・整理・データベース化は余り進んでいない。現在インターネット上のサーバに画像を保存するサービスは多くあるがネットワーク速度、著作権上の問題等で教育の現場で活用するにはまだ難しい。そこで本論文では学校教育活動で必要とされる機能を考慮した校内イントラネット上で動作する画像データベースの構築について述べる。データベース構築に際してはTIFF タグを利用し、画像に付加情報を与える事でデータベース管理を容易にする手法を採用した。さらにデータベースを情報機器変遷の教材提示用として授業で利用するための具体的なインタフェースの実装を行った。

[キーワード: 画像データベース, TIFF タグ, 画像透かし, LAMP 環境]

1. はじめに

現在の中高等学校における情報教育はコンピュータの利用に関わる教育・学習内容が中心であるが、その背景となる情報機器の変遷に関わる教育・学習内容も、科学と技術の融合の歴史や先人の知恵や工夫を知り考察する上において有意義なことである [1]。

過去に人類が作り出してきた計算補助具や機械式計算機は構造がシンプルであり、動作原理を理解することが容易である。また、使われる素材や手法こそ異なるものの、基本的な原理は現代の機器に継承されているものも存在する。こうした歴史上の機器は実務や研究に際してはもはや役に立つものではないが計算機を学ぼうとする者にとっては効果的な教材となりうる。そして、それは学習者の科学的な興味や認識を深め、知的好奇心を刺激するにちがいない。

これら歴史上の計算機器は貴重なものであり、教室においては写真による提示が欠かせない。デジタルカメラの普及と高画質化に伴い、我々が扱うデジタル画像データはその数やデータサイズにおいて増加の一途である。一方、それら画像データの分類整理はまだ十分になされていないのが現状である。現在の種々の画像フォーマットにおいては、例えばEXIFタグなど画像に情報を付加するための規格は存在するが、それらは画像自体の情報（画素数、カメラ名、ISO感度、焦点距離、露出、日時、位置情報など）であり、その画像を説明する文章などまとまった量の情報を付加する方法は見出せない。EXIF タグにはユーザコメント

タグが存在するが、サイズが限られており、教育利用に用いるには制約が多過ぎる。

そこで TIFF 画像の未使用タグを利用してユーザが自由に付加情報を付け加えることができるプログラムを制作した。さらに、それらをネットワーク上のコンピュータからブラウザを使用して操作することができるためのシステムを構築した。また、マルチユーザでの情報の共有を想定し、複数情報をユーザごとの権限に応じて画像に付加・閲覧できるようにした。

これらの情報を加えることで、画像の価値は飛躍的に高まり、情報の散逸を防ぎ、教育現場での教材の共有化を可能にすることができる。このとき、インタフェースは学校教育の授業における画像提示を想定し制作した。

2. 授業におけるプレゼンテーション

授業における教師のプレゼンテーションは、逐次型と対話型に分類できる。逐次型は事前に予定されたシナリオに従って構成される。多少の順序の入れ替わりはあるものの、基本的に説明は直線的に構成される。事前に緻密に構成することができ、講義や講演会形式で効率的に情報を伝達し理解を促すのに有効である。一方、対話型は目標が存在するが構成は必要に応じて話題が動的に分岐・統合していく形態である。教育現場での授業あるいはフォーラム・シンポジウム形式での会議において、説明側と理解側が相互に情報を交換

* 鳴門教育大学 大学院 (修士課程) 教科・領域教育専攻 生活・健康系コース (技術・工業・情報)

** (株)アルマス

*** 鳴門教育大学 大学院 自然・生活系教育部

しながらその場で必要に応じて構成を変化させていくことができる。また、様々なレベルや特質を持つ理解側に対して柔軟に対応することができる。実際の教育現場では、同学校の同学年の授業でもクラス間で生徒集団の特質は大きく異なる場合があり、硬直化した授業運営は生徒の理解度や興味・感心との乖離を生むことが少なくないことから後者の形態を採ることが多い。

3. 画像データベースの必要性

コンピュータのハードウェアおよびデジタルカメラの進歩は著しく、解像度の高い画像を大量かつ迅速に処理することが可能になった。それに伴い、これら大量の画像を分類・整理し必要に応じて表示することが要求されるようになった。

授業での画像の提示を考えたとき、逐次型の場合は予め画像を選択して順序を整えておくことで準備が終わるが、対話型の場合は可能性のある膨大な画像の分類、インデックスの作成、検索用データベースへの登録などの煩雑な手続きを要する。

ローカルのハードウェアでこれら进行处理するアプリケーションには、多種多様なものが存在している。またインターネット上で画像データを管理し共有するためのシステムも次々に現れており、非常に高機能なものもある。ローカルのハードウェアにシステムを構築した場合、ハードウェアの移動・設置の煩雑さ、OS間のシステム移植の難しさ、データの共有や共同作業の難しさがある。また、インターネット上の既存のデータ共有サービスは便利だが、容量の制限、経済的な負担、低速インターネット回線がボトルネックとなることによる転送の遅さ、公開サーバにデータが置かれることによる著作権問題、そしてなにより既存のシステムに対する各学校や組織の実態に応じた柔軟なカスタマイズの難しさがあり、組織的に使用するにはまだまだハードルは高いと思われる。

そこで、これらの問題を解決する方法の一つとして、イントラネット内サーバを使用し、シンプルで自由度の高い画像データベースシステムの構築を試みた。イントラネット内では比較的高速なネットワークが使用でき、アクセス制限もしやすくデータが流出する危険性も減少させることができる。

4. 画像データベースの試作

4.1 基本設計

従来の Web サーバ上の文書や画像の管理は、HTML 文書と画像データをそれぞれのディレクトリに分類保存するものであった。しかしこの手法では HTML 文

書内のタグに記述したファイル名とそれによってリンクされる実際の画像ファイルの間に不整合が起りやすく、画像がリンク切れとなり、これらが大規模に発生すると修正は大変な手間を要した(図1)。

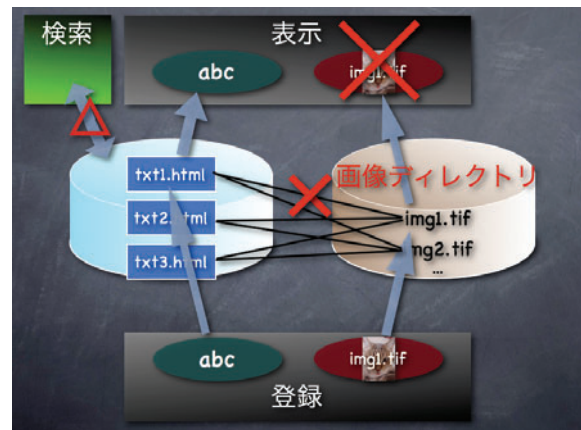


図1 従来のファイル配置

今回構築するシステムでは、各画像に関連するテキスト情報はすべて画像に埋め込んだ後サーバに保存し、データベースと適当な時期に同期させる。テキストデータと画像は一体不可分のためリンク切れや内容の不一致などの状態は原理的に起こらず、また万一データベース上のデータをすべて失ったとしてもテキストデータおよび画像との関連が失われることはない。また、画像に埋め込まれたデータを利用することにより新たにデータベースを自動的に再構築することも可能である。すなわち、画像自体がデータベースのバックアップになっていると考えることもできる(図2)。

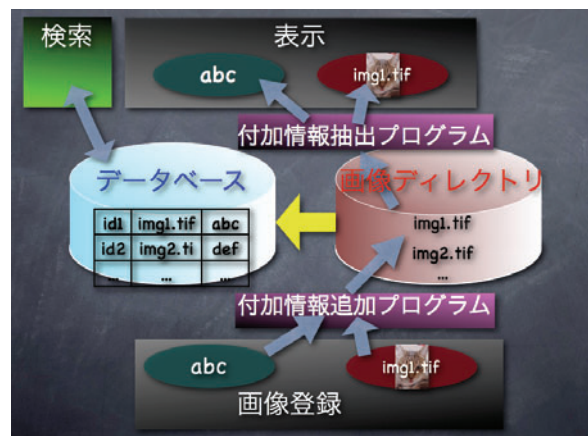


図2 データ付加手法を用いた方法

4.2 画像透かし技術を利用した付加情報の追加

従来の基本的な画像透かし手法を適用する場合、画像にちらつきが多く見られ、画質の劣化が深刻であった。そこで画像に現れるちらつきを減少させるため、人間の視覚特性を考慮した視認識回避手法、および画像全体の画質を向上させるための最小2乗法を導入し

た画像平滑化手法、さらにTIFF ファイル構造を利用し秘匿性を維持しながらまったく画質が劣化しない非表示領域利用画像透かし手法[2]を開発してきた(図3)。これらの手法を用いることにより、秘匿性と画質を両立させながら画像データと説明文書をついにまとめることが可能となった。しかし、この手法においては、暗号化・復号化プログラム内で画像データを一旦全てメモリに読み込み処理するため、サイズの大きな画像を処理するにはメモリを大量に使用し、32bit プロセッサを使用したコンピュータで使用可能なメモリ上限を越えてしまうという問題が起こった。この問題に対してはメモリ容量の大きい64bitプロセッサを使用するコンピュータの導入によって解決することができた。一方で複雑な計算による計算機への負担および処理する画像サイズが暗号化により増大することによる転送時間の増大により、全体の処理速度の遅延が目立つようになってきた。重要で改変が許されない著作権のある情報を使用する場合はこの手法が有用であるが、データベース用の付加データ追加の用途としては品質が過度であり、暗号強度とのトレードオフでリアルタイム性に欠ける点に問題があり、より軽快なソフトウェアを開発する必要があった。

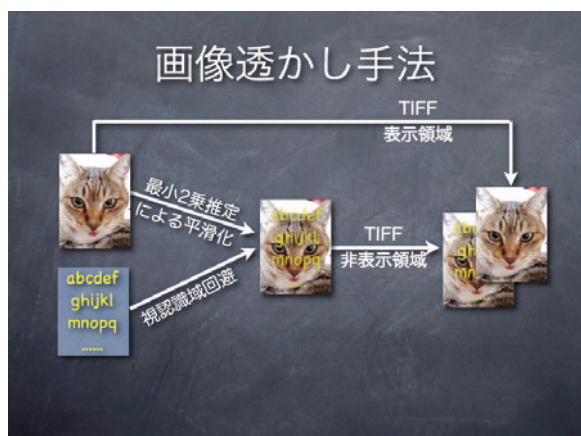


図3 画像透かし技術を利用した付加情報の追加

4.3 TIFFタグを利用した付加情報の追加

TIFF の仕様書[3]には、プライベートフィールドと値について、図4のように記されている。この仕様書より、32768 以上のタグナンバーは組織内においては種々のアプリケーションとの衝突を避けることができるものと考え、これらを複数使用して、複数のデータを一つの TIFF ファイル内に収める方法を採用することとした。収めるデータの暗号化の方法は一般的に公開されている手法から選び、暗号化・復号化のサーバへの負担を軽くした。複数の未使用タグ利用による複数データの付加と暗号化により、一つの画像ファイルを組織内の複数のメンバーが使用し、メンバー間で共

有するデータと、共有しない独自の秘匿データをその一つの画像ファイルに共存させることができ、共同作業の可能性を広げることができる(図5)。

…組織がその組織においてのみ意味をもつ情報を TIFF ファイルの中に保存することを望むかもしれない。タグナンバー 32768 以上は、時にプライベートタグと呼ばれるが、この目的のために予約されている。

(中略)

あなた自身でタグナンバーを選んではいけません。そうすることは将来深刻な互換性の問題を起す可能性があります。しかしあなたの TIFF ファイルがあなたのプライベートな環境から流出する可能性が無い、またはほとんど無いのであれば、リユーズ範囲である 65000-65535 の中の TIFF タグの使用を考慮していただきたい。あなたはこの範囲のタグ番号を使用するときには Adobe 社に連絡する必要はない。

図4 TIFFにおける利用可能タグ

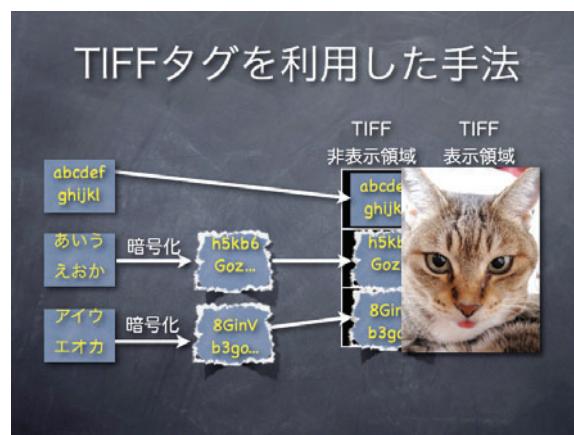


図5 TIFFタグを利用した付加情報の追加

4.4 TIFF画像ファイルの構造

TIFF 画像は図6に示すように、イメージファイルヘッダ (Image File Header, IFH)、イメージファイルディレクトリ (Image File Directory, IFD)、データ本体の 3 要素から構成される。

第 1 階層として、先頭に IFH があり、第 2 階層として複数の IFD があり、第 3 階層以降に各種テーブルとデータがある。IFH はファイルヘッダであり、TIFF ファイルを識別するためのヘッダ部分で、ファイルの先頭に一つだけ存在する。2 バイトのバイトオーダー識別子、2 バイトのバージョン番号、4 バイトの最初の IFD ポインタを含んでいる。IFD は画像の各種パラメータや、画像関連データ、カラーマップ、イメージデータへのポインタを格納した 12 バイトのデータエントリの集まりである。

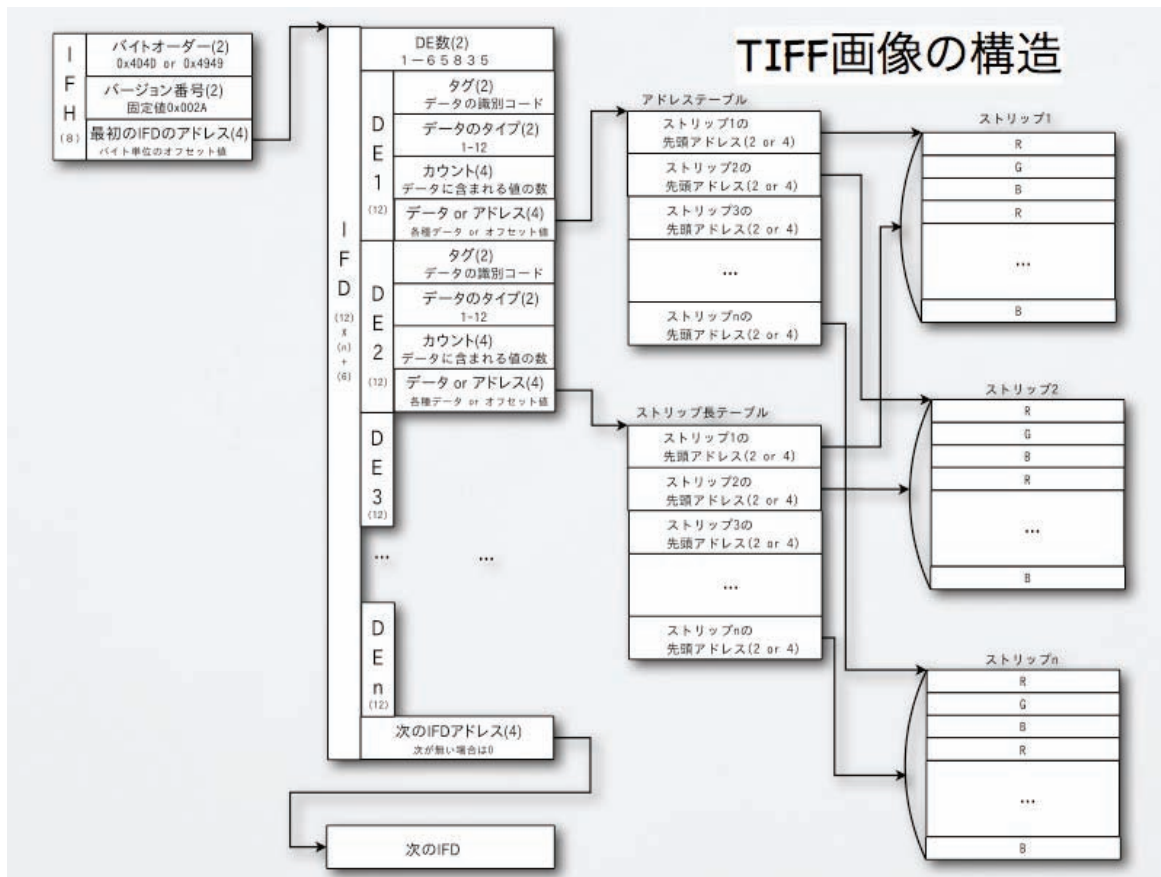


図6 TIFF画像の構造

4.5 TIFFタグ編集プログラム

画像透かしプログラムは、我々の開発した TIFF 画像の表示領域と非表示領域を組み合わせさせた画像透かし手法のプログラムを用い、また今回新たに TIFF ファイルの任意のタグの情報を追加、抽出、削除するプログラムを作成した。これらはサーバ環境においてコンパイル、得られた実行ファイルを PHP の system 関数を用いて制御する形で使用した。

図7はこれらのプログラムソースのパッケージに含めた Readme 文書である。図8はコンパイル後の addntg コマンドの簡易ヘルプ表示と実行の様子、図9はコンパイル後の tifchk コマンドによる TIFF タグの解析結果の様子で、左側は処理前、右側は処理後である。

TIFF フォーマット画像ファイルに新しくタグを追加し、様々なデータを画像に埋め込み、管理するためのプログラムです。

新しく追加するタグは各々連続しない独立した IFD に記述され、数珠つなぎにファイルの最後に追加していく仕様になっています。従ってこのプログラムで追加されたタグは独立した IFD と 1:1 に対応しています。

オプションを指定しない限りタグナンバーは 32768

(0x8000) を使用します。-t オプションを使用する際は、32768 以上のプライベート使用を認められているタグを使用してください。

各プログラムの説明

addntg : 新しくタグを追加し、指定した任意のデータを TIFF に埋め込むプログラム

extntg : 指定したタグのデータを TIFF から抽出し、指定したファイルに保存するプログラム

delntg : 指定したタグとデータを TIFF から取り除くプログラム

tifchk : TIFF ファイルのタグ情報を表示するプログラム

taglst : TIFF ファイルに含まれるすべてのタグを改行区切りで出力するプログラム

各プログラムの引数やオプションについては、コンパイル後 -h オプションで表示されます。

コンパイルおよび動作確認環境 (/proc/cpuinfo)
 Intel (R) Pentium (R) 4 CPU 3.00GHz Memory 2065984 kB Linux version 2.6.24-19-generic
 (buildd@palmer) (gcc version 4.2.3 (Ubuntu4.2.3-2ubuntu7))

```
Intel(R) Xeon(R) CPU X5355 @ 2.66GHz Memory
12334072 kB Linux version 2.6.23.15-137.fc8
(mockbuild@hammer2.fedora.redhat.com)
(gcc version 4.1.2 20070925 (Red Hat 4.1.2-33))
```

図7 Readmeの内容

```
$ ./addntg -h
Usage: addntg -i Infile -e Embedfile -o Outfile [-t tagnum]
-i:Input tiff file name
-o:Output tiff file name
-e:Embed file name
-t:New tiff tag number(decimal)
If there isn't '-t' option, we use tag number 32768.:-)

$ ./addntg -i ./apple1.tif -e ./test.txt -o embeded.tif -t65001
Input Tiff Size : 24575032 (Byte)
Input Char Size : 37 (Byte)
Output Tiff Size : 24575087 (Byte)
```

図8 addntgの簡易ヘルプと実行

```
$ ./tifchk ./apple1.tif
Tiff File Size : 24575032 (Byte)
Byte Order : 4949 (Hex)
Version : 42
First IFD Adr : 176b208 (Hex)

IFD[0] --> DE Count: 17
DE Tag Type Count Size Data
0 100 3(SHORT) 1 2 d00
1 101 3(SHORT) 1 2 920
2 102 3(SHORT) 3 6 176b2da
3 103 3(SHORT) 1 2 1
....(略)....
15 128 3(SHORT) 1 2 3
16 131 2(ASCII) 58 58 176fbfe
Next IFD Adr: 0x0000

ImageWidth (0x100): 3504 pixel
ImageLength (0x101): 2336 pixel
BitsPerSample (0x102):RG8888
Compression (0x103): None
Photometric (0x106): RGB
StripOffsets (0x111):176b2ee (H)
StripCount (0x111): 2336
RowsPerStrip (0x110): 1
StripByteCounts(0x117):176d76e (H)

$ ./tifchk ./embeded.tif
Tiff File Size : 24575087 (Byte)
Byte Order : 4949 (Hex)
Version : 42
First IFD Adr : 176b208 (Hex)

IFD[0] --> DE Count: 17
DE Tag Type Count Size Data
0 100 3(SHORT) 1 2 d00
1 101 3(SHORT) 1 2 920
2 102 3(SHORT) 3 6 176b2da
3 103 3(SHORT) 1 2 1
....(略)....
15 128 3(SHORT) 1 2 3
16 131 2(ASCII) 58 58 176fbfe
Next IFD Adr: 0x176fc38

IFD[1] --> DE Count: 1
DE Tag Type Count Size Data
0 fde 2(ASCII) 37 37 176fc4a
Next IFD Adr: 0x0000

ImageWidth (0x100): 3504 pixel
ImageLength (0x101): 2336 pixel
BitsPerSample (0x102):RG8888
Compression (0x103): None
Photometric (0x106): RGB
StripOffsets (0x111):176b2ee (H)
StripCount (0x111): 2336
RowsPerStrip (0x110): 1
StripByteCounts(0x117):176d76e (H)
```

図9 tifchkによるタグの解析

4.6 暗号化の手法

テキストの暗号化については、当初 GnuPG の導入を検討したが PHP による制御が難しいためlibmcryptを利用した PHP 拡張モジュールを利用することにした。libmcrypt では、DES、3DES、Blowfish、RIJNDAEL (ラインダール: AES 暗号とも呼ばれる) などのブロック暗号をサポートしており、利用可能な暗号モードは CBC、OFB、CFB、ECB である。

以上の選択肢から、現段階で

1. 解読の可能性が低い。
2. 使用メモリが少ない。
3. ライセンスの問題がない。

という条件を満たすことから、今回はラインダール暗号を CBC モードで使用することとした[4]-[6]。

ユーザ認証は PHP による HTTP 認証のフックによる。これは Apache モジュールとして実行した時のみ

有効で、CGI 版では利用できない。Apache モジュール上の PHP スクリプトにおいては、header() 関数を使用して“Authentication Required”メッセージをクライアントブラウザに送ることが可能であり、クライアントブラウザではユーザ名とパスワードの入力要求ウィンドウがポップアップ表示される。

一度ユーザがユーザ名とパスワードを入力すると、次回以降 PHP スクリプトを含むその URL は定義済みの変数 PHP_AUTH_USER, PHP_AUTH_PW, PHP_AUTH_TYPE にそれぞれユーザ名、パスワード、認証型が代入された状態で呼ばれ、定義済みの変数は配列 \$_SERVER および \$_HTTP_SERVER_VARS でアクセス可能である。Basic 認証および Digest 認証(PHP5.1.0以降)の両者がサポートされているが、今回は Basic認証を使用した。

認証 PHP スクリプト内では変数に格納されたユーザ名 (PHP_AUTH_USER) とパスワード (PHP_AUTH_PW) をデータベースに登録してあるユーザ名とパスワードと比較し、両方が一致した場合にそのユーザの使用するユーザタグナンバーおよび画像データベースのディレクトリレベル(各画像の公開度に応じた権限レベル)を変数に格納する。

なおパスワードはユーザが独自に保存するテキストの暗号化のパスワードとしても使用する。ユーザ情報を格納したテーブル“user”のフィールドは図10のとおりである。

```
mysql> show fields from user;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default| Extra |
+-----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| user | char(64) | YES | | NULL | |
| password | char(64) | YES | | NULL | |
| user tag | int(8) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
```

図10 userテーブル

4.7 全文検索

全文検索 (Full text search) とは、コンピュータにおいて複数の文書 (ファイル) から特定の文字列を検索することである。「ファイル名検索」や「ファイル内文字列検索」と異なり、「複数文書にまたがって文書に含まれる全文を検索する」という意味で使用される。

全文検索は grep 型と索引 (インデックス) 型に分類できる。今回しているデータベース MySQL は、バージョン 3.23.23 以降、索引 (インデックス) 型の全

文検索をサポートしており、敢えて grep 型の仕組みを整える必要が無いため、索引(インデックス)型の全文検索システム[7]を加えることとした。MySQLにおける全文検索では、デリミタ文字(半角スペース等)で区切られた複数の単語から成るカラムに対して FULLTEXT インデックスを張っておき、それを専用の MATCH ... AGAINST 構文を用いた SELECT 句で検索することにより、マッチ率の高い順(近似値順)にソートされたレコードセットを取得することができる仕組みである。つまり、MySQL 単体で全文検索の仕組みが完結する。日本語(マルチバイト)文字の対応もバージョン 4.1.1 以降が対応している。ただし、日本語は単語間にスペースを入れる(分かち書き)の習慣が無いため、日本語の文章そのままでは全文検索のインデックスを作成することはできない。このことに対する対応としては次の 3 つの方法が考えられた。

[方法1] 文章を形態素解析にかけ、単語レベルまで分解し、それらをスペース区切りのテキストとして格納する。

[方法2] 文章を n 文字ごとに規則的に分割しておく(Ngram[8])、これらを BOOLEAN MODE で検索する。

[方法3] MySQL の中に全文検索エンジン[9]を組み込むパッチを利用する。

MySQL にパッチを充てることや再コンパイルを行うなど変更を可能な限り避け、新規環境へのシステムのインストール作業を簡素化したいという方針で、[方法1]に対応した。日本語構文解析は MeCab[10]の形態素解析システムを利用した。Ngram のシステムもさほど複雑ではないため実装は可能だと思われる。

5. サーバー環境

データベースシステムは、オペレーションシステムに Linux、Web サーバに Apache、データベースに MySQL、そしてそれらを制御するスクリプト言語に PHP を用いた図 11 の LAMP 環境を使用した。これらはすべて無償で公開されている。さらに、独自に開発した画像透かしプログラムおよび未使用タグ利用データ付加プログラムを組み合わせ、サーバ上の資産をネットワーク上のクライアントの Web ブラウザで操作・表示することとした。

ブラウザにおけるインターフェースは、大別して操作機能、表示機能、作業機能の 3 つに分類し、それぞれは、HTML タグ <a href> の target 属性を用いて、別ウィンドウ(タブブラウザにおいては別タブになる)に表示することとした。これは、複数のディスプレイ

への表示時に教師側の表示部と生徒側の表示部にそれぞれウィンドウを振り分けて、生徒側の表示部に不要な情報が表示されるのを防ぐためである。

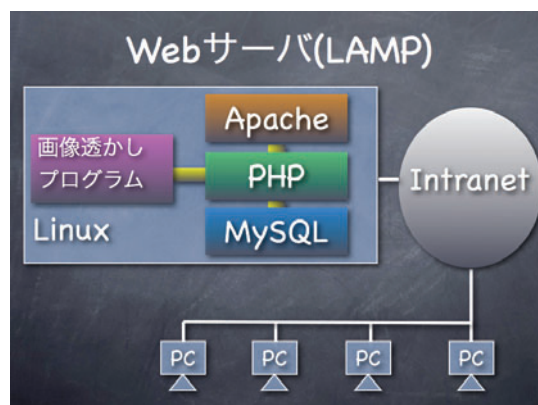


図 11 LAMP環境

6. スクリーンショット

以下に実際にブラウザでアクセスした様子を示す。最初のユーザ認証を図 12 に、キーワードを入力しての検索結果を図 13 に、選択画像の表示を図 14 に、画面をクリックして表示させたスライダーによる拡大・縮小を図 15 に、ユーザ毎の説明文章表示を図 16 に、ファイル登録フォームを図 17 に示す。

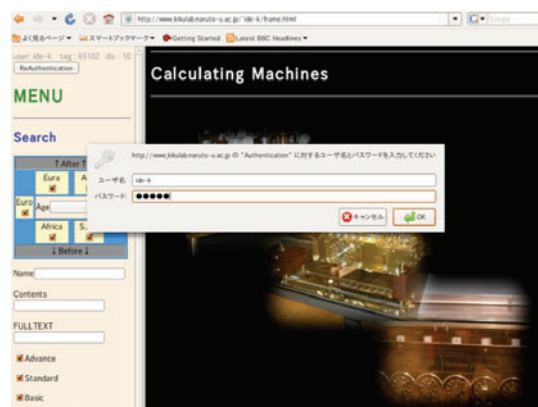


図 12 最初の認証画面

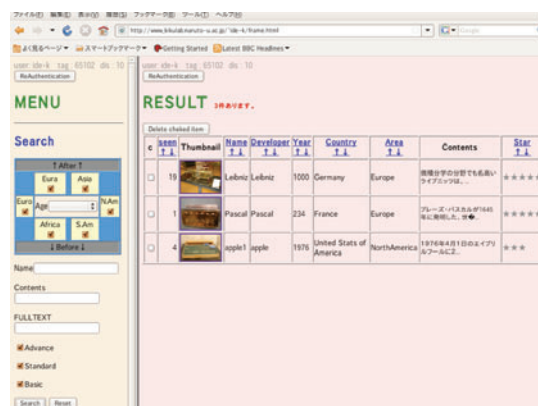


図 13 検索結果のリスト表示

7. まとめ

任意の説明文章を画像と一体化することができる新たな画像データベースを構築した。この中で扱った TIFF タグを利用した画像への情報付加の技術は、簡便な手法であり、デジタル画像に様々なデータを付加することでその価値を高めることができる手法である。デジタルカメラの一般化に伴いデジタル画像の取得は非常に簡単になったが、それらの活用方法についてはまだ模索が続いている。このタグ利用情報付加技術は、今後ますます増加していくことが予想されるデジタル画像の取扱いの活用方法を広げる一助となるのではないかと考える。

本研究においては情報機器変遷の分野に特定したシステムの構築を行ったが、教育活動における画像表示に必要とされる付加情報、そしてそれらの分類や検索方法を検討することにより、一般的な構造にすることが可能である。更に教育現場での実践を行い、システムのブラッシュアップを図っていきたい。

参考文献

- [1] 菊地章, 井上淳一: 情報教育の観点から見た情報機器の変遷. 日本産業技術教育学会, 第 43 巻, 第 1 号, pp. 53-60, 2001.
- [2] 菊地章, スーパーバトル, 櫻本尚子: TIFF 画像の表示領域と非表示領域を組み合わせた画像透かし手法. 日本産業技術教育学会誌, 第 49 巻, 第 4 号, pp. 279-288, 2008.
- [3] Adobe Developers Association: Tiff revision 6.0 final-june 3, 1992, <http://partners.adobe.com/public/developer/tiff/index.htm>.
- [4] Wikipedia 暗号: <http://ja.wikipedia.org/wiki>.
- [5] IT 辞典 rijndael: <http://dictionary.rbbtoday.com/Details/term2291.html>.
- [6] ブロック暗号化モード: <http://www.triplefalcon.com/Lexicon/Encryption-Block-Mode-1.htm>.
- [7] MySQL 全文検索: <http://dev.mysql.com/doc/refman/4.1/ja/fulltext-search.html>.
- [8] MySQL FULLTEXT+Ngram: <http://www.tatamilab.jp/rnd/archives/000390.html>.
- [9] Senna 組み込み型全文検索エンジン: <http://qwik.jp/senna/FrontPageJ.html>.
- [10] MeCab: Yet Another Part-of-Speech and Morphological Analyzer, <http://mecab.sourceforge.net/>.



図 14 選択画像の表示

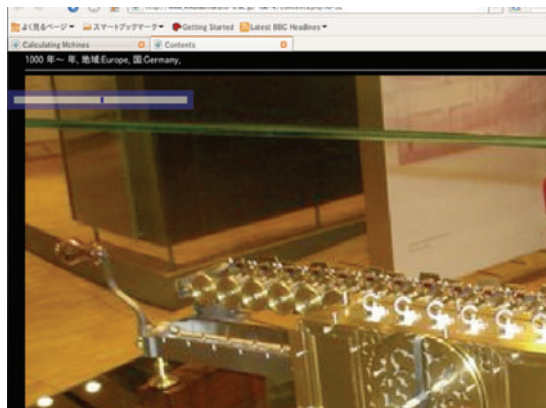


図 15 スライダー操作による画像拡大



図 16 ユーザ毎に暗号化された文章の表示

A screenshot of a file upload form titled 'Upload Image & Data'. The form contains several input fields for metadata: 'Image(TIFF only)', 'Object Name', 'Developer', 'Year_From', 'Year_To', 'Country', 'Area', 'Photo_Place', 'Photo_Date', 'Photo_er', 'Description(Text)', and 'ide-k Data(Text)'. There are also radio buttons for 'Star' (1, 2, 3, 4, 5) and 'Level' (Advance, Standard, Basic, All). A 'Disclosure' section has radio buttons for 'Public', 'Organization', and 'Administrator'. An 'Embed Method' dropdown is set to 'raw'. At the bottom, there are 'Upload' and 'Reset' buttons.

図 17 ファイル登録フォーム