

デザイン情報のデータベース化に関する研究

—画像データベースの試作—

デザイン・工芸部 藤田純一, 山田淳人

A Study of Database for Graphics

Jun'ichi FUJITA and Atsuhito YAMADA

新商品開発などでのデザイン作業段階では、発想の支援（アイデア）を得るために、様々な分野の豊富な情報が必要であり、中でも画像情報は重要である。そこで今回、画像比較などが高速なデータベースを研究・試作した。

検索機能や比較検討機能そして表示速度などを高いレベルに保つため、独自のデータベース構造を開発し約4000件の画像を登録してテスト運用を行ったが、機能、安定性とも実用に耐えうることを確認した。

1. 緒言

現在、デザイン事務所ではデザイン案のプレゼンテーションにCG（コンピュータ・グラフィックス）などの映像を用いなければコンペティションなどに参加し難い状況になってきており、また印刷業界ではDTP（電子印刷）の普及により、デジタル画像データの量が急速に増加している。またマルチメディア関連のコンテンツ制作現場では、デジタル化された画像音声は必須のものであり、デザイン関連の様々な業種で、デジタル素材の蓄積量は今後膨大なものになっていくことが予想される。

デジタル素材の利点として、一度デジタル化した後では再利用・修正・変更などが容易に行える点があげられる。

しかしながら再利用しやすいように整理・保存してあるかといえば、県内の大手企業でも未だそこまで手が回っていないのが現状である。

しかし、時間と人手をかけてデジタル化した素材であるので、データベース化しておけば、必要な時に必要な素材をすぐに取り出すことができ、デザイン作業時のアイデア抽出段階での有効利用や、作業時間の短縮も可能である。

データベース化を必要としているのは、画像を初めとして動画や音声など多岐に渡るが、画像は最もデザイン関連企業で使用頻度が高く、かつデザイン作業の基本であり、また画像データベースを研究・構築すれば後はその基本を生かすことにより他のデータベースも構築できることから、画像のデータベースを研究・構築することとした。

2. データベースの利用法の想定

一言で画像データベースを作るといっても、様々な利用形態が想定され、画像数や利用人数そして使用設備その他様々な違いがあり、そのすべての利用形態で満足できるデータベースの設計は非常に難しい。

そこで今回はデータベースを研究・構築するにあたって、利用方法・形態をある程度限定して考えた。

緒言でも述べたとおり、デジタル画像が膨大な数になっているのはデザイン関連の企業では共通の課題であるが、それはとりまおさず、当センターデザイン・工芸部の問題でもある。

そこで当部が研究・指導業務において、有効に使用できる画像データベースを研究・構築することで、それは課題を抱えた企業へそのまま適用できると考え、今回は当部が必要としているデータベースの利用形態に限定した。

当部でデータベース化したい素材を大別すると、以下の4つに分けられる。

- ①市販の著作権フリーの画像（デジタルデータ）
- ②当部で制作したCG画像（デジタルデータ）
- ③当部または外部で撮影した写真（アナログデータ）
- ④書籍・雑誌などの写真（アナログデータ）

しかし、上記のすべてをデータベースとして登録できる訳ではなく、例えば④に関しては画像の著作権・版権が存在するので、簡単には登録することはできない。

技術指導などで、雑誌などに載っている商品写真を基に説明やアイデア出しを行う際、このようなデータベースがあれば非常に便利ではあるが、今回はデータベースを研究し試作するのが本題であるので、この問題に関してはあえて触れず、著作権をクリアできる素材だけを対象とした。

なお、ここで画像と呼んでいるものは、写真データと、線画データの両方を含むものである。

画像数としては、①が約9000件、②が約300件、③が約3000件であるので、総数としては約1万件を越える程度の件数であるが、画像数は今後も当然増加することを考えて、約10万件程度の能力を持ったデータベースを構築することとした。

3. ソフトウェアの選定

約10万件のデータベースとなると、単純に計算してもサムネール（検索用の縮小画像）だけでも1GB程度の容量になり、非常に大規模なデータベースとなる。これだけの大容量のデータベースをハンドリング良く操作・検索できるようにするためには、市販のいわゆる画像カタログ用ソフトでは不可能で、純粋のデータベース用ソフトが必要となってくる。またデータベースソフトも大別すると2種類ある。カタログ形式とカード形式である。カタログ形式とカード形式の画面を写真1と2に示す。

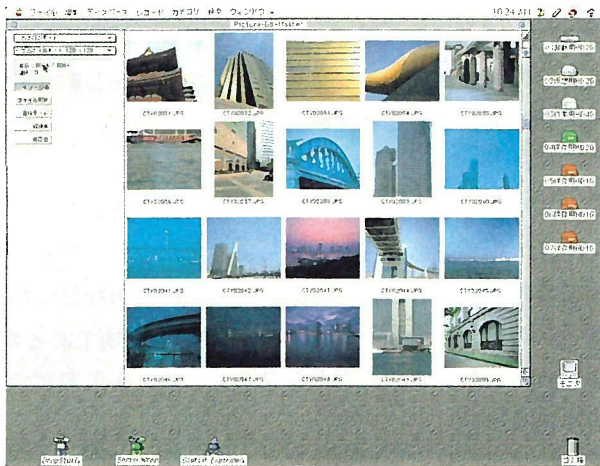


写真1 カタログ形式のデータベース

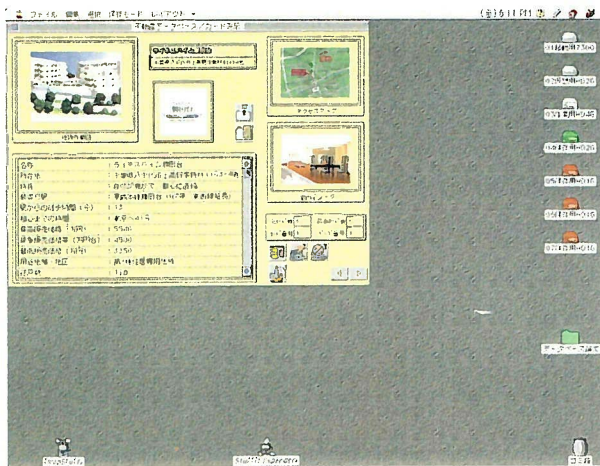


写真2 カード形式のデータベース

カタログ形式の長所は、複数の画像サムネールを一覧できることである。画面サイズの関係で、一度に見ることができる枚数は限られているが、検索時に非常に使いやすい形式である。ただしその画像に登録してあるキーワードやメモなどを見るには、追加処理を行う必要がある。

それに対しカード形式の長所は、1枚の画像に対して設定してあるキーワードやメモなどの情報が一覧して見ることができるもので、例えばデザイナーリストなどをデータベース化する際には非常に使いやすい形式である。

今回は、画像の検索を行うのが主目的であるので、カタログ形式のデータベースを採用した。

市販のデータベースソフトで、カタログ形式を採用しているのは約10種類弱あるが、画像数が10万件程度でも検索速度や表示速度があまり落ちない点、アップルスクリプト（アップルコンピュータ(株)が提供する自動化処理用プログラム言語）に対応しており自動処理が行える点、ネットワークに対応しておりサーバクライアント処理が可能点、キーワードがドラッグコピーできる点、動画や音声も扱え将来にも対応できる点、そしてメモ欄の使いやすさなどからCumulusを選定した。

4. データベースの基本構造

画像データベースとは、ユーザーが必要とする画像を、キーワードなどにより検索・表示するのが目的である。

しかし画像数が膨大になればなるだけ、1つのキーワードでヒットする画像数も増加する。すると当然のことながら、複数の画像を比較するという作業が発生する。

画像を比較する時に、ある程度の大きさの画像で見ないと細部が良く判らず、比較検討作業がスムーズに行われない。しかし現状のデータベースソフトは、比較検討作業は、通常サムネールで行い、画像が気に入ったらオリジナルの高解像度の画像を読み出す構造になっている。しかしサムネールは画像が小さいため、細部まで比較できず、どうしてもオリジナル画像を読み込んで比較することになる。

ただしオリジナルの高解像度の画像は、ファイルサイズが大きいので、通常画像を読み込むだけでも数十秒程度掛かってしまい、スムーズな比較検討とは言い難い。まして5枚10枚といった複数の画像の比較を行うとなると、かなりの時間が必要となってしまう。

また今回のデータベースは、著作権フリーのCD-ROM素材がかなりの量を占めており、そのオリジナルを読み込みもした場合は、CD-ROMディスクの抜き差しが発生し、より以上に時間が必要となってしまう。

そこで今回、データベースの構造に通常とは異なる考え方を導入した。つまり、サムネールとオリジナル画像の中間に、比較検討用の画像を新規に作成し、それを用いて比較検討作業を高速化しようというものである。

この方法と、従来の方法の比較を図1に示す。

なお、比較検討用画像のサイズに関しては画像が大きい方が良いのは当然のことであるが、実際問題として読み込み速度やディスク容量、そしてディスプレイの表示面積や表示方法の事を考えるとあまり大きくても困るため、適当と思われるサイズを模索した。その結果、マルチメディア素材として使用されている画像の標準サイズや、放送局で使用されている静止画送出装置などは、いずれも横640ドット×縦480ドットであり、画面上でも十分比較が可能なことからこの値を標準値とした。またこの値であれば将来

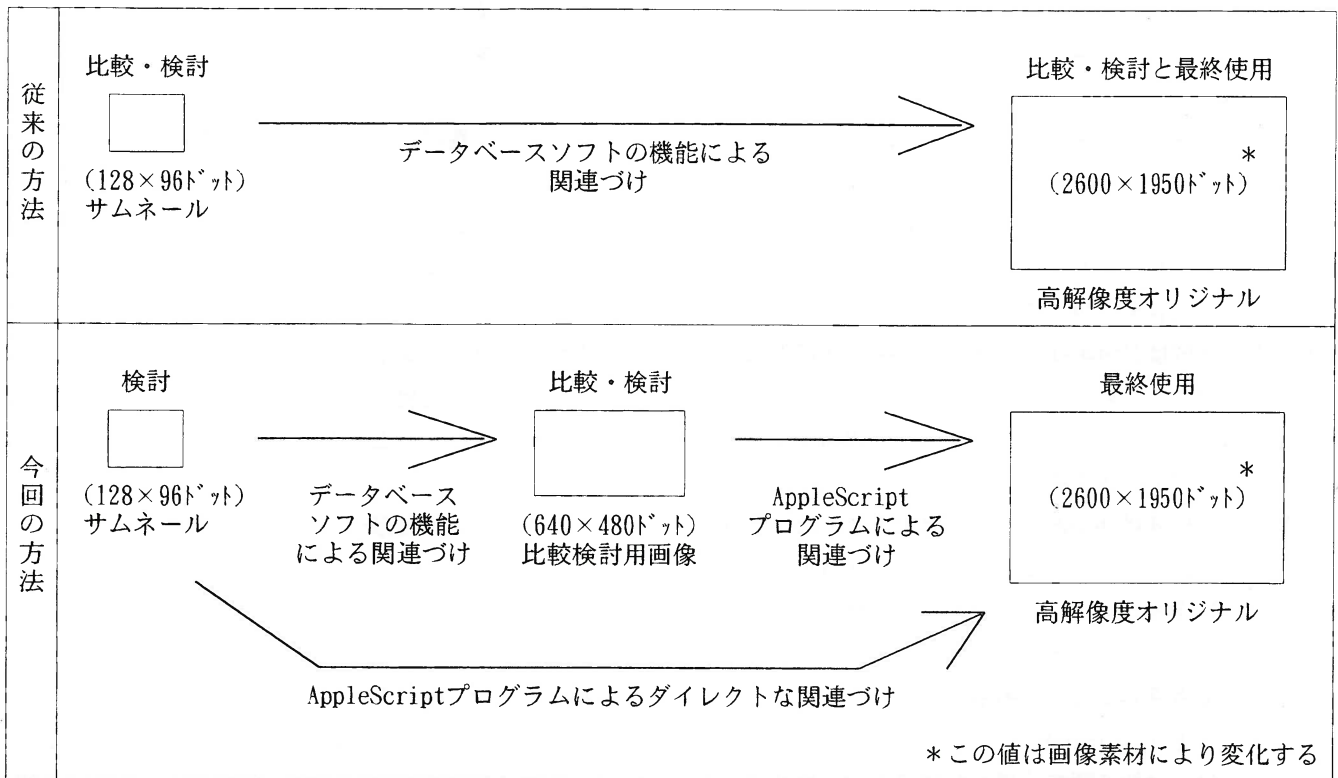


図1 データベース構造の比較

前述のマルチメディア制作の時や、放送局での高速静止画
送出装置への発展も考えられる。

この方法の長所としては、複数の画像の比較検討作業が
非常に高速にできる点である。従来の方法では、画像の比
較検討のために1枚のオリジナル画像を開こうとすると、
CD-ROMディスクを探すのに15秒、ディスクの挿入・認識に
15秒、ファイルを開くのに30秒とすれば約1分かかること
になる。

しかしながら今回考案した方法では、ファイルへのアク
セスは常時接続されているハードディスクだけで済むため、
上記の工程が2～3秒程度であり、画像の比較検討におい
て絶大なハンドリングを実現できる。

欠点としては、比較検討用の画像を新規に作成するた
めの、保存場所と作業時間が必要になることである。

前述の画像サイズ640×480ドットに何らかの圧縮を
かけて1つのファイルが平均80KBと仮定すると、10万件の比較
検討用画像では8GBのディスク容量が必要となる。しかし
現在ハードディスクは非常に低価格になってきており、ま
た必ずしも最初で8GBのディスクが必要ではなく、デー
タベースの登録が増加するに従って、その都度2GB程度
のディスクを増設して行っても良いので、実用的には問題
にならない。

また作業時間に関してはアップルスクリプトなどの自動
化処理を行うことにより、比較検討用画像の作成は時間
外の夜間に処理することが可能で、これも実用上問題とな
らない。

5. 使用した機材および環境の紹介

何らかのソフトを使用して研究を行った場合、処理時間
やハンドリングに関しては、使用した機材の影響がかなり
発生する。この報告書で書かれている処理時間やハンドリ
ングに関する記述は、以下の機材を用いて評価しているこ
とをあらかじめお断りしておく。

コンピュータ本体として、アップルコンピュータ(株)製
PowerMac-8500/150(内蔵ハードディスク2GB、メインメ
モリ192MB、キャッシュ256KB、漢字Talk7.5.5)、外
部ハードディスクとして、ヤノ電器(株)製のNJ9100HD、ビ
デオカードはIMS社製TwinTurbo128M4で1152×840ドットの
フルカラー表示である。

また、Cumulusへのメモリ割り当ては25MBで設定した。
画像登録が増加するにしたがって、メモリ割り当て不足に
よるハンドリング低下やエラーなどの不具合を危惧し、管
理者モードでメモリ状態を常時監視していたが、4000件程
度登録したこの原稿執筆時でもメモリ不足による不具合は
発生しなかった。

またOSに関する事では、機能拡張書類やコントロールパ
ネル書類、そしてフォントなどは、必要最小限を除いてす
べて削除した。また、ネットワーク接続やファイル共有な
どに関しても、必要な時以外はすべて物理的・論理的な接
続を解除し、単独のパソコンとして最速で動作するよう
な環境を設定した。

6. データベースの設計

データベースを構成する基本の柱としてはいろいろな項目が考えられるが、今回は次の内容に関して検討を行った。

- ①比較検討用画像の保存方法（ファイル規格・圧縮方法）
- ②比較検討用画像の作成方法（自動化処理）
- ③検索用のキーワードの使用方法・基準
- ④注釈などで使用するメモ欄の使用方法・基準

以下、順に内容を説明する。

6.1 比較検討用画像のファイル規格・圧縮方法

画像を保存するための規格（ファイルフォーマット）は現在多岐に渡る。その中から今回使用する規格を決定するに当たって、以下の3点を重視した。

- ①規格自体に汎用性があること。理由としては、特殊なソフトが無いとファイルが開けないのは問題であるし、また今回機材はマッキントッシュを使用しているが、ネットワーク経由でWindowsマシンへデータを転送することも考慮したためである。
- ②ファイルサイズが小さいこと。Photoshopフォーマットなどは、画像に関する様々な情報も含んだ構造になっており、変更作業などでは有益ではあるが、ファイルサイズが非常に大きくなるので除外した。
- ③更に汎用的な圧縮が可能なこと。この画像は比較検討用であるので、前述の画像解像度で、圧縮による画質劣化が目視で確認できないのであれば、ディスク容量節約のため少しでもファイル容量が小さくする必要はある。

以上のことを基に検討を重ね、結果的にはJPEG規格（Joint Photographic Experts Group）を採用した。

JPEG規格と一言で言っても、非常に多くのソフトがサポートしており、圧縮の変数なども多岐に渡っており、それぞれに画質とファイル容量の微妙な差がある。

そこで今回8種類のテスト用画像を用意し、それぞれのJPEG圧縮での、目視による画質劣化度や圧縮率などを検討した。テスト画像を右ページに示す。（写真3～10）

それぞれの画像での評価基準を説明する。

写真3は、精細な風景写真である。花の輪郭、花畑の色、空と雲の混じり会う部分の輪郭、の再現性、そして家の回りに発生する輪郭ノイズの程度を評価した。

写真4も、精細な風景写真である。チューリップの花びらの肌感と色、木の樹皮の肌感、そして細かい葉っぱの輪郭再現性を評価した。

写真5は、色数の単純な例である。青空のグラデーションや帽子の色の再現性、肌の質感を評価した。またこの画像は高圧縮が可能ならず、圧縮率に関しても留意した。

写真6は、細かい木目模様である。色や輪郭の再現性を評価した。

写真7は、イラストレーターで描いた絵をビットマップに変換した画像である。輪郭やグラデーションの再現性を評価した。

写真8は、女性の髪の毛のアップである。髪の毛1本1本の輪郭やグラデーションの再現性、艶の表現を評価した。

写真9は、当部で制作したCG画像である。絨毯や壁紙の模様やグラデーションの再現性、そして室内照明の空気感を評価した。

写真10は、フラッシュ撮影の例であるが、フラッシュの光により焼き物の白肌のグラデーションがクリップしかかっている画像である。このグラデーションが潰れてしまわないかを評価した。

テストしたJPEGの種類は以下の通りである。

Photoshop-JPEG

Painter-JPEG

DeBabelizer-JPEG

GraphicConvertor-JPEG

また、上記のJPEG規格は、それぞれ圧縮変数の設定が可能で、最初は設定値を粗く取って画質変化と圧縮率の関係を把握した後、画質劣化/圧縮率の値が最小限になる設定値を探した。

目視による画質評価で、最も結果が良かったのがPhotoshop3.0によるJPEG圧縮（変数＝高画質）であった。画質劣化は無いわけではないが、等倍表示では目視ではまず判らず、なおかつ圧縮率も平均15%程度であった。今回評価したJPEGの種類と結果を表1に示す。

表1 JPEG圧縮のテスト結果

使用ソフト	圧縮変数	劣化度平均	圧縮率平均
Photoshop	最高画質	9 0 %	2 5 %
	高画質	8 0 %	1 5 %
	標準	6 0 %	9 %
Panter4	最高画質	9 0 %	2 6 %
	高画質	7 0 %	1 3 %
	標準	5 0 %	8 %
DeBabelizer	1 0 0 %	9 0 %	3 7 %
	7 5 %	6 0 %	9 %
	5 0 %	4 0 %	6 %
Graphic Convertor	1 0 0 %	9 0 %	3 2 %
	7 5 %	8 0 %	2 2 %
	5 0 %	6 0 %	1 5 %

テストを行った感想として、Photoshop以外のソフトでは、圧縮率が高くなった時の画質劣化が、同ソフトに比べ激しかったことと、画像によっては圧縮時にバグが発生し、不要な線などが入ったものがあった。

画質判定は全てディスプレイ上でを行い、場合によっては画像を一部拡大して判定を行った。



写真3 テスト画像1



写真4 テスト画像2

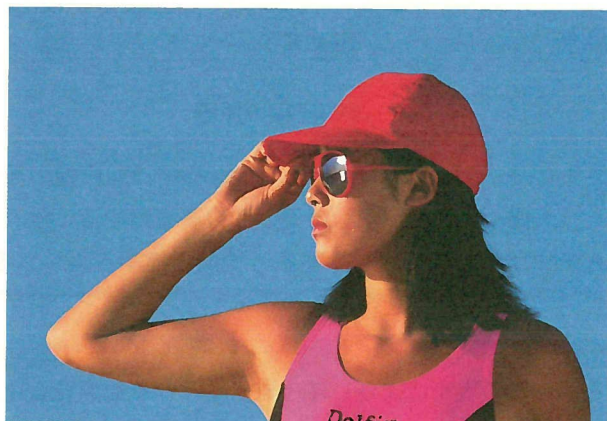


写真5 テスト画像3

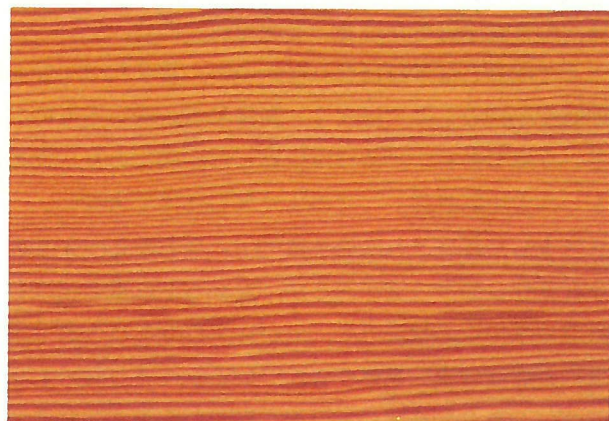


写真6 テスト画像4



写真7 テスト画像5

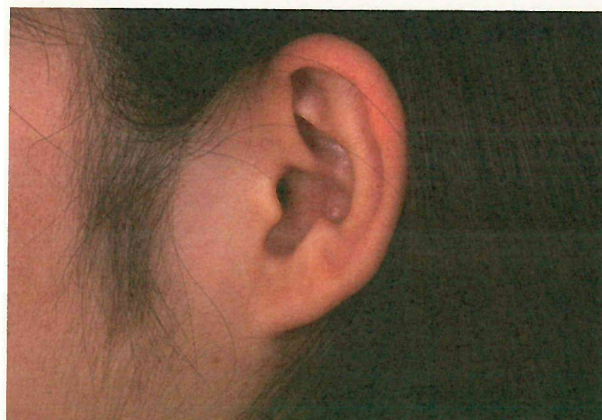


写真8 テスト画像6



写真9 テスト画像7

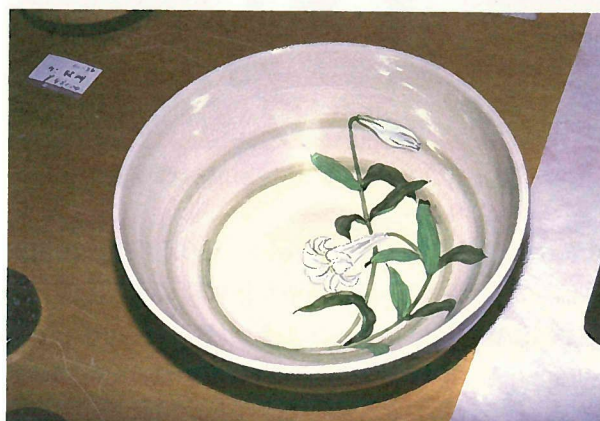


写真10 テスト画像8

写真11～13は判定に用いた拡大画像である。これは写真3の拡大画面で、主に輪郭ノイズを判定した。



写真11 拡大画像1 (オリジナル)



写真12 拡大画像2 (Photoshop高画質)



写真13 拡大画像3 (Painter標準画質)

また上記の結果は主観によるものであるため、客観的な裏付けをとるために、JPEG規格が圧縮時に用いるサブサンプリング周波数と量子化テーブルを、(株)写真化学の協力を得て抽出した。

サブサンプリング周波数とは、人間の視覚特性を考慮して輝度信号よりも色信号に高圧縮をかける比率のことで、

RGBの3原色信号を、輝度(Y)と色(UV)に変換する際の情報量の差である。

また量子化テーブルに関しては、JPEG圧縮の原理の根本に係わっているため、先に圧縮原理を図2に示す。

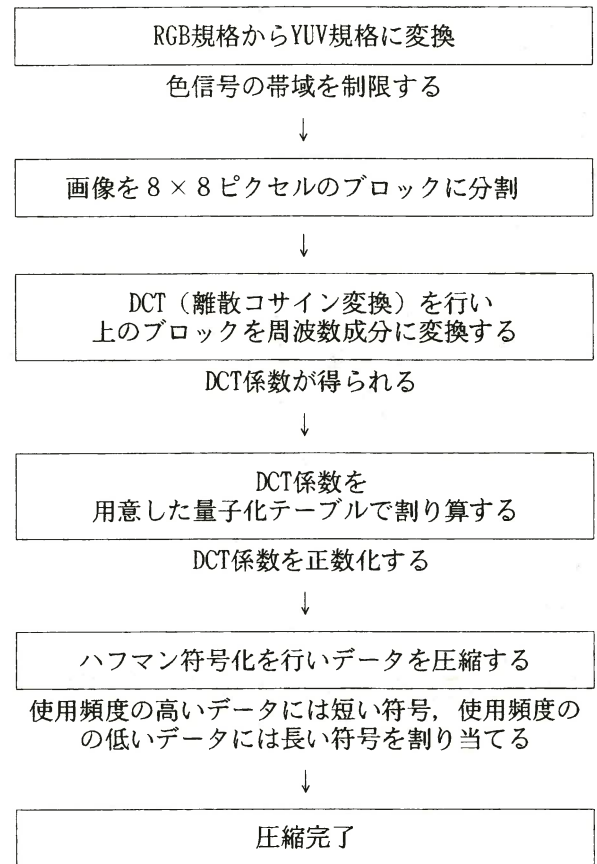


図2 JPEG圧縮の原理

この中で、DCT係数を正数化する箇所がJPEG圧縮の要であり、人間の視覚で判別しにくい情報を省略すること(量子化テーブルの値を大きくする)で高圧縮率を得られる。一般的には、細かい色の変化の領域が少ないほど、高圧縮率が得られる。¹⁾

しかしこの省略により、非可逆処理となり原画像と全く同じに復元することはできない。ただし省略する値を決定する量子化テーブルを適度に調整することで、画質の品質を保ったまま圧縮をかけることが可能である。

ここで注意しなければならないのが、量子化テーブルは圧縮率を決めるものではなく、画像品質を決めるものであるという点である。圧縮率は圧縮した結果であり、例えば「1/5の圧縮処理」などのようなことはできない。

テストした各ソフトの量子化テーブルを表2～5に示す。

8行8列の数字の表があるが、左から右に進むにつれて高周波の水平周波数成分を、上から下に進むにつれて高周波垂直の周波数成分を多く含んでいる。つまり表の右下方向の値が大きくなればなるほど、細かい画像が省略されることになる。²⁾

この設定値を検討する限り、目視での画質判定が正しいことが分かり、圧縮規格をPhotoshopのJPEG規格（変数＝高画質）に決定した。

しかし、このテストを終了した後Photoshop 4が発売されたため、比較を行った。結果を表6に示す。

表6 Photoshopのバージョンアップによる差

ソフト名	Photoshop 4		Photoshop 3
変数	6 高画質		高画質
その他	ベースライン標準	ベースライン最適	——
テスト1	140,205 バイト	136,493 バイト	136,140 バイト
テスト2	180,653 バイト	175,808 バイト	174,422 バイト
テスト3	53,423 バイト	51,025 バイト	50,282 バイト
テスト4	107,931 バイト	102,787 バイト	104,748 バイト
テスト5	41,611 バイト	36,495 バイト	33,393 バイト
テスト6	87,084 バイト	83,909 バイト	84,305 バイト
テスト7	57,489 バイト	54,657 バイト	54,384 バイト
テスト8	72,796 バイト	71,459 バイト	68,795 バイト

バージョン3と4では変数の設定方法が変更になっていたが、ごくわずかに容量が増加しただけで基本的な変化はなく、画質的にも目視では判別がつかないことから、以降はPhotoshop 4のJPEG規格（変数＝6高画質、ベースライン＝最適化）に変更した。

また、今回の比較検討用画像は圧縮をかけることにより、平均で80KB前後の予定である。この画像を保存するために用意したハードディスクの容量は9GBで、単純計算すると約11万件の画像が保存できることになる。

しかし、OSが持つ現在のディスク管理システムHFS (Hierarchical File System) では、どんなに小さいファイルでも必ず必要となる容量（アロケーションブロック・サイズ）があり、これは表7のとおりハードディスクの容量に比例する。8GBのディスクを使用した場合、たとえ1バイトのデータでも、128KBの容量を占有してしまう。

表7 ディスク容量とアロケーションブロック・サイズ

ディスク容量	アロケーションブロック・サイズ
1GB	約16KB
2GB	約32KB
4GB	約64KB
8GB	約128KB

またMAC-OSの場合、ファイルは独自の構造を持っており、1個のファイルは実はデータフォークとリソースフォークと呼ばれる2つのファイルから構成されている。データフォークは通常のデータ部分であるが、リソースフォークには作成アプリケーション名やアイコン情報その他のGUIに関するデータが入っている。

ユーザーは通常このリソースフォークを見ることはできないためあまり認識されていないが、このことにより今回の場合、2つのアロケーションブロック・サイズを占有し、8GBのディスクで結局256KBにもなってしまふ。この状態では、前述の画像は約3万件しか登録できないことになり、ディスクの使用効率が非常に悪い。

以上のことを考慮し、今回使用する9GBのディスクはパーティションにより分割し、比較検討用画像を保存する領域には、できるだけ小容量のディスクを設定することとした。ただし、この方法ではディスクが操作が煩雑になるので、仮想RAIDなど色々な解決策を研究したが、結果としては9GBのディスクを、容量がそれぞれ異なる5つのパーティションに分割し、作業中のデータを4GB領域に、画像が不完全なため修正が必要で再保存した高解像度オリジナル画像を2GB領域に、残り1GB弱の3つの領域に比較検討用画像を保存することとした。この方法により現在登録済みの比較検討用画像を、4GBのディスクと1GB弱のディスクに登録した比較例を表8に示す。

表8 ファイルの占有容量と実際の容量との比較

ディスク容量	総画像数	ディスク占有容量	実際の容量
4GB	4067	902.2MB	447,180KB
0.905GB	4067	502.7MB	447,180KB

6.2 比較検討用画像の作成方法（自動化処理）

10万件の比較検討用画像を手作業で作成すると膨大な時間が必要となり現実的ではない。そこでなんらかの自動化処理を図る必要がある。

今なら、Photoshop 4のバッチ処理機能を使うことにより自動化は比較的容易にできるが、自動化に取り組み始めた時にはそのような機能はなく、解決方法を研究し、その結果以下の3つの方法が見つかった。

①PreFab Software社製のPreFab Playerをインストールすることで、Photoshopでのマウスやキーボードの操作などをAppleScriptで記述できるもの。ただし操作の記録・再生機能はない。

②DayStar Digital社製のPhotoMaticをインストールすることで、Photoshopの操作のほとんどを記録・再生することができる。AppleScriptで記述可能。

③Affinity Microsystem社製のTempoII plusを使用することで、Photoshopの操作を独自のマクロ言語で記録・再生することができる。AppleScriptからマクロを実行する。³⁾

この中から、操作の記録ができる点、動作の安定性、そしてフリーソフトである点などから、PhotoMaticを使用することとした。

PhotoMaticは、Photoshopに組み込む機能拡張である。インストールすると、メニューバーにメニューが新しくできるので、この機能を使用して操作の記録を行う。比較検討用画像を作成するための全ての工程（ファイルを選ぶ→ファイルを開く→画像解像度を640ドットに縮小→同名で別のフォルダにJPEG高画質で圧縮し保存）を記録し、後はこのプログラムを、AppleScriptで実行できるように若干の手直しをするだけである。

リスト1のプログラムは、著作権フリーのCD-ROM画像を自動処理した例である。上の部分がPhotoMaticがPhotoshopを制御する部分で、下半分がOSを制御する部分である。

```

on DPI_change_save()
  press modifier key caps lock
  select menu "画像解像度..." -----画像解像度の変更
  wait for window change
  wait for window change
  type string "72"
  type special key tab
  type string "640"
  type special key enter key
  wait for window change
  wait for window change
  wait for window change
  select menu "別名で保存..." -----JPEG圧縮と保存
  wait for window change
  wait for window change
  click x 8967 y 3627 dialog 0
  click x 571 y 1969 dialog 0
  type special key return
  click x 598 y 3005 dialog 0
  type special key return
  click x 598 y 3005 dialog 0
  type special key return
  select pop up menu name "JPEG" x 4266 y 9845 dialog 0
  type special key return
  wait for window change
  click x 1118 y 6514 document 0
  type special key return
  wait for window change
  wait for window change
  select menu "閉じる" -----作業画面を閉じる
  wait for window change
end DPI_change_save

on run
  set inFolder1 to (choose folder with prompt "処理するフォルダを選んでください。")
  set inFolder2 to "2Cumulus-DB:0001-SOZAI:6-四季-自然-SF:"
  tell application "Finder"
    set filelist to every document file of folder inFolder1
    repeat with aFile in (every file of inFolder1)
      set filename1 to name of aFile
      set inFilename to inFolder2 & filename1
      tell application "Adobe Photoshop™ 3.0J" to activate
      tell application "Adobe Photoshop™ 3.0J"
        PMGrabSemaphore
        PMOpenDocument alias inFilename
        do script DPI_change_save
        PMGiveSemaphore
      end tell
      tell application "Finder" to activate
      tell application "Finder"
        end tell
      end repeat
    end tell
  end run

```

リスト1 通常のプログラム例

ただし注意しなければならないことに、ビットマップ系の画像はこのプログラムの応用でほとんど処理できるが、それ以外の画像に関しては、Photoshop側のシステムの違いにより、画像ファイルが開けない場合がある。

例えばイラストレーターなどは、ファイルを開くときに線画データからビットマップ形式に変更するために、画像サイズや解像度の要求をしてくる。このプログラムでは、そのメッセージに割り込みをかけることができない。

それを解決するために、色々なプログラムを試作してみたがなかなかうまくいかず、結果的にはOS上で前処理をすることで対応した。そのプログラムをリスト2に示す。

```

on DPIchange_Save()
  select menu "開く..." -----イラストレータ画像を開く
  wait for window change
  release modifier key caps lock
  click x 43 y 536 dialog 0
  press modifier key caps lock
  click x 9043 y 3065 dialog 0
  click x 2809 y 4253 dialog 0
  type special key return
  click x 2745 y 2222 dialog 0
  type special key return
  release modifier key caps lock
  click x 43 y 536 dialog 0
  click x 2532 y 3563 dialog 0
  press modifier key caps lock
  click x 2766 y 3525 dialog 0
  type special key return
  release modifier key caps lock
  click x 43 y 536 dialog 0
  click x 2532 y 2184 dialog 0
  press modifier key caps lock
  click x 2766 y 2261 dialog 0
  type special key return
  wait for window change
  type special key tab
  type string "640"
  release modifier key caps lock
  press modifier key caps lock
  select pop up menu name "RGB" x 4728 y 7514 document 0
  type special key enter key
  wait for window change
  select pop up menu name "画像の統合" x 10009 y 109 palette "レイヤー"
  select menu "別名で保存..." -----JPEG圧縮と保存
  wait for window change
  click x 8995 y 4171 dialog 0
  click x 598 y 5213 dialog 0
  type special key enter key
  click x 598 y 2701 dialog 0
  type special key enter key
  click x 652 y 5261 dialog 0
  type special key enter key
  click x 4918 y 8483 dialog 0
  type special key delete key
  type special key delete key
  type special key delete key
  type string ".JPG"
  type special key enter key
  wait for window change
  click x 2007 y 6422 document 0
  type special key enter key
  wait for window change
  select menu "閉じる" -----作業画面を閉じる
  wait for window change
end DPIchange_Save

on run
  set inFolder1 to (choose folder with prompt "NEXTフォルダを選んでください。")
  tell application "Finder"
    repeat with aFile in (every file of inFolder1)
      set filelist to every document file of folder inFolder1
      set filename1 to name of item 1 of filelist
      --display dialog filename1 & "を処理します。"
      move file filename1 of folder "NEXT" of disk "作業用HD1400"
        to folder "NOW" of disk "作業用HD1400" --入らない為2行に分けた
      tell application "Adobe Photoshop™ 3.0J" to activate
      tell application "Adobe Photoshop™ 3.0J"
        PMGrabSemaphore
        do script DPIchange_Save
        PMGiveSemaphore
      end tell
      tell application "Finder" to activate
      tell application "Finder"
        end tell
      move file filename1 of folder "NOW" of disk "作業用HD1400"
        to folder "FINISH" of disk "作業用HD1400" --入らない為2行に分けた
    end repeat
  end tell
end run

```

リスト2 イラストレータ用のプログラム例

このようなプログラムにより、ほぼ全ての規格の画像をPhotoshopで自動処理することが可能で、実際にこの画像データベースに登録してある約3000件の比較検討用画像を作成した。

しかしPhotoshopはバージョン4より、操作の記録・再生機能とバッチ処理ができるようになった。

この機能はフォルダ単位での一括処理ができ、また処理速度もPhotoMaticを使用した場合よりも高速である。同ソフトを導入してから約1000件の画像を処理したが、動作も非常に安定している。そのため現在はこの自動処理プログラムを用いており、ここで開発したプログラムはすでに使用していない。

6. 3 検索用キーワードの使用法・基準

画像を検索する際に、キーワードは必要不可欠のものである。Cumulusでは、バージョン3よりキーワードのことを、より広い捉え方でカテゴリと呼ぶようになった。

10万件の画像を検索するためのキーワードとなると、非常にその設定が難しい。人によりキーワードから想像する「イメージ」に違いが生じるからである。特に形容詞などの「トロピカル」や「明るい」のような単語は、画像を見る人によってイメージが異なり、的確な検索ができず、キーワードには不適切である。

そのため今回のデータベースでは、最も的確なイメージが単語として表現できる「名詞」で構成することとした。

例えば「トロピカル」を名詞検索する場合には、「海」、「砂浜」、「ヤシの木」などのような関連する名詞を想像し、リストから探すか入力すれば良い。

また「富士山」のような場合は非常に単純で、単語が登録されていれば、その名の通り「富士山」というキーワードを入力するだけで良い。

Cumulusでは、キーワードを階層構造にすることが可能で、「富士山」の上に「山」というキーワードを設定しておくことにより、関連画像の検索も容易である。

このように「誰が見ても確かにそうだ」と思われる名詞をキーワードにし、なおかつ1画像1キーワードではなく、1つの画像に対し複数のキーワードを付けることにより、検索効率を上げるとともに、関連画像へのアクセスも容易になることで「デザイナーの発想を支援できる」システムとした。

ただし、名詞をキーワードにすることによる欠点もあり、まずキーワードの数が膨大になることがあげられる。キーワード数が多いほどハンドリングへの影響が想定されるが、これは最近のハードウェアの進歩により、吸収できると考えた。元々Cumulus自体が検索速度が非常に速く、現在までに名詞キーワード数は約700単語を登録したが、検索時間は1～2秒程度である。

また「トロピカル」の例で説明したが、ユーザーが関連する名詞を想像する必要がある点である。自分がイメージしている画像を名詞に翻訳しなければならない。しかし前述の「形容詞」などと比較すると「名詞」を想像するのは容易であり、またキーワードリストから探し出しても良いのでこれも問題にはならないと考えた。

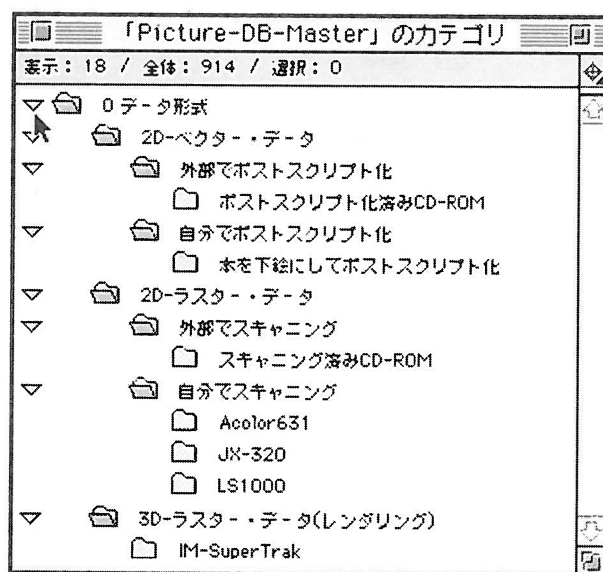
キーワードリストも常に先の700単語を表示しているわけではなく、場合によっては階層構造を限定して表示し、より大きな「くくり」の名詞を選択しても良い。通常は、この方法の方が検索し易いと思われる。

次の欠点は、名詞としての単語が付けられない画像がある点である。例えば名称不明の物が写った画像であったり、あるいは、イメージ写真のような「雰囲気だけ」の画像などの場合である。このような場合を想定して、本データベースでは、サブキーワードとして名詞以外の単語も別個に登録した。前者の場合は「名称不明」、後者は「イメージ」のようにである。

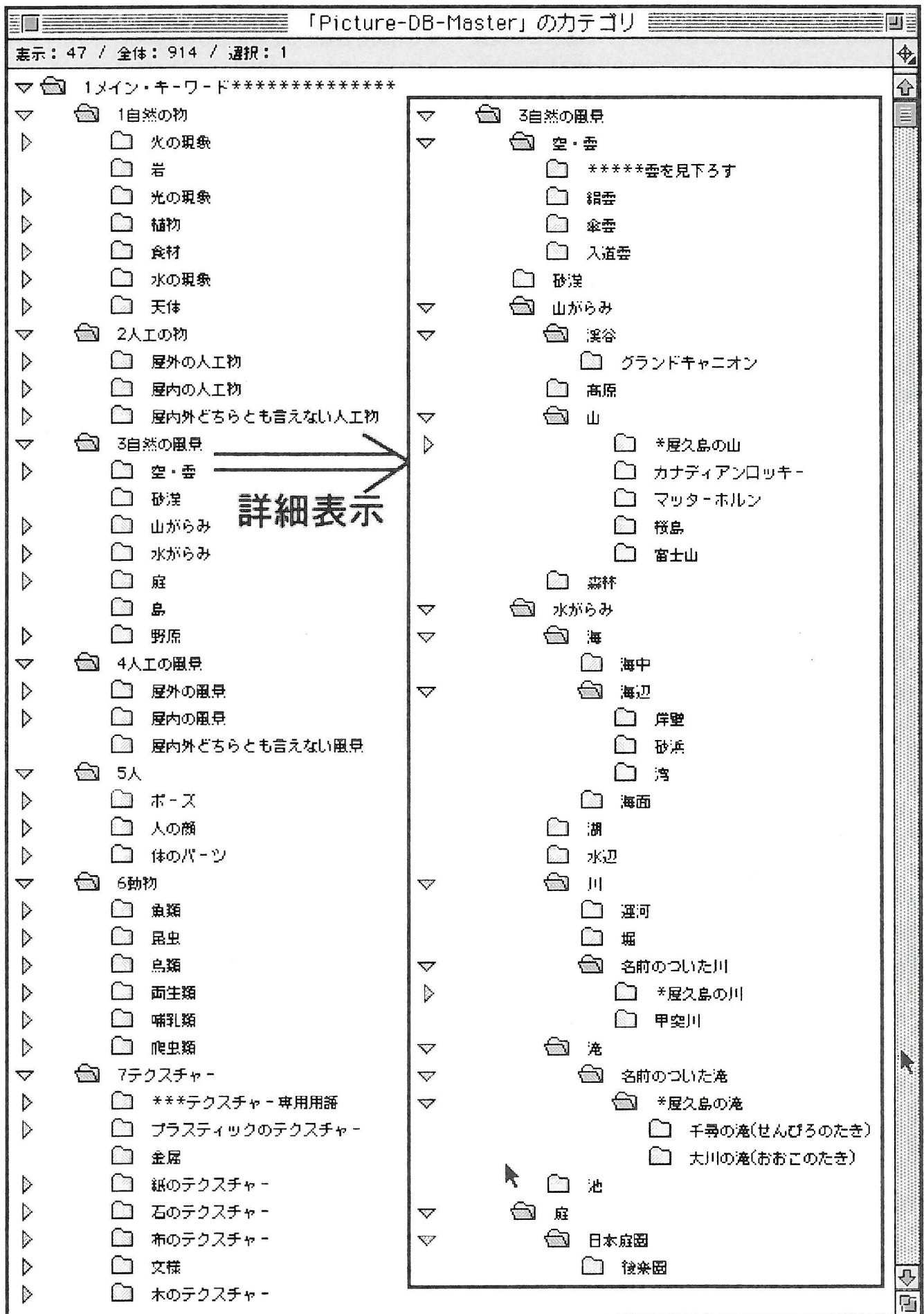
このサブキーワードは、名詞キーワードを更に補強するために使用するもので、名詞と混在しないようする必要があり、「物」の名称以外を設定した。

最後に、画像の形式による検索も想定し、画像の簡単な作成履歴をもキーワードとして登録することとした。例えば猫の写真が探したい場合、「猫」のキーワードで検索すると、結果は写真もあるがイラストも出てくる。その場合、「猫」と「ラスターデータ」という2つの単語で検索すれば、上記の検索は一発で可能である。このキーワードは「データ形式」という項目で、サブキーワードと同じく、名詞キーワードとは独立して別個に設けてある。

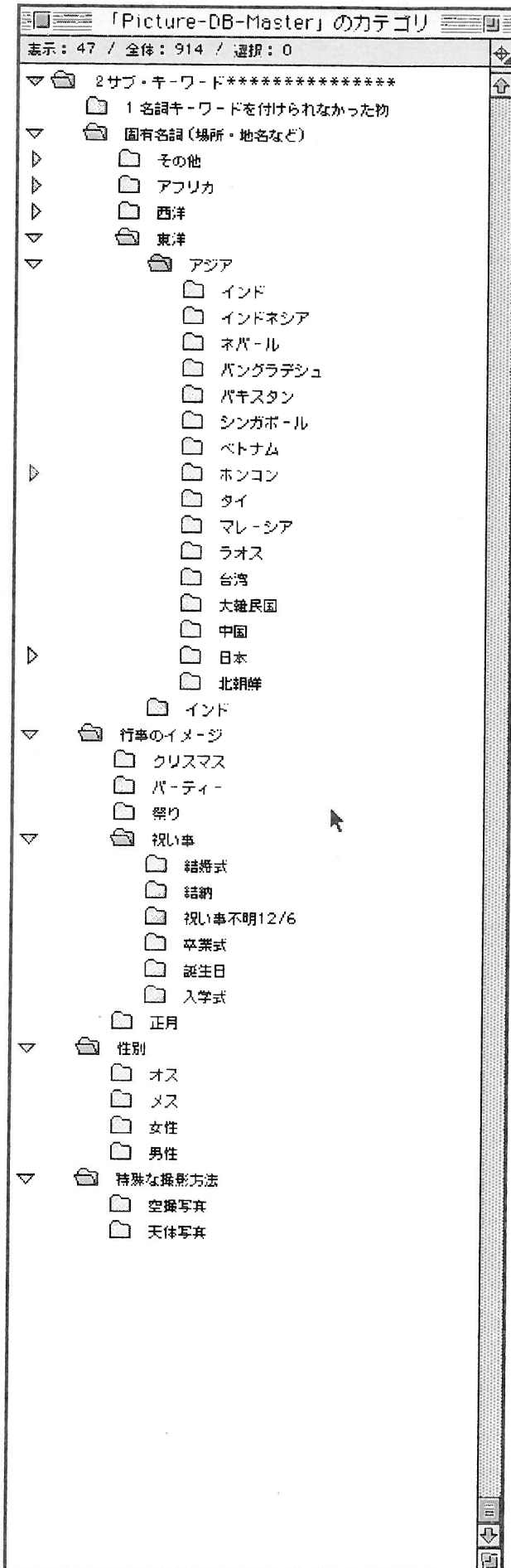
リスト3に「データ形式キーワード」を、リスト4に「メインキーワード」の一部を、リスト5には「サブキーワード」の一部を示す。



リスト3 データ形式キーワード



リスト4 メインキーワードの一部



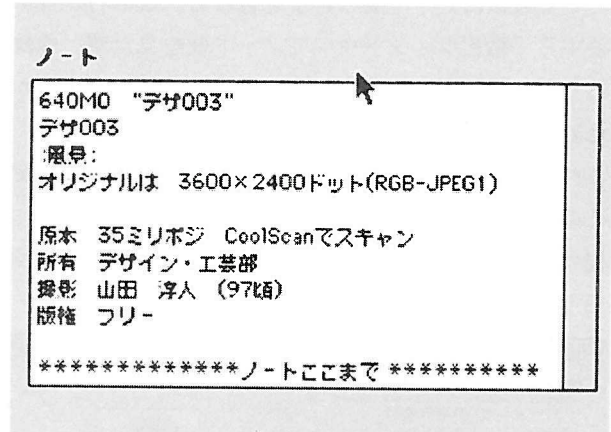
リスト5 サブキーワードの一部

6.4 メモ欄の使用法・基準

画像を検索する時にはキーワードで検索するが、検索した画像に関する固有の情報を知りたい場合がある。そのため、今回メモ欄（Cumulusではノート欄と呼んでいる）を有効利用することにした。

ノート欄も、記入する内容などの基準を決めておかなければならないのは当然で、特に今回は、データベースの構造に独自の設計をしているため、高解像度オリジナル画像の保存場所などもノート欄に記述されている必要がある。

リスト6にノート欄の使用例を示す。



リスト6 ノート欄の使用例

順を追って説明する。まず1行目で、高解像度のオリジナル画像が保存してある媒体（ハードディスク、CD-ROM、MOなど）の種別と、その媒体の呼び名（正式な名ではなくタイトル名である点に注意）を記述してある。

2行目は、その媒体の正式な名（ディスク名称）を記述してある。

なぜこの1行目と2行目で、呼び名と正式な名を分けたかということ、数あるCD-ROMディスクの中には、タイトル名とディスク名が異なる場合があるためである。

現在CD-ROMの抜き差しは手作業でおこなっているので、ユーザーが必要なのは「媒体の呼び名」であるが、将来CD-ROMオートチェンジャーなどを使用した自動抜き差しを考慮すると、この場合は「媒体の正式名称」が必要となるからである。また、複数枚あるシリーズもののディスクでタイトルはそれぞれ違うが、ディスク名は全く同じという場合もあった。そのため安全策として、このような方法を取った。

3行目は高解像度オリジナル画像の保存のパスである。画像を開くためAppleScriptが必要とする。

4行目は高解像度オリジナル画像の詳細データである。画像のサイズ、RGB/CMYK/グレースケールなどの色に関するデータ、そして圧縮規格などを記入してある。

5行目は予備行で、これ以降に関しては、画像の原本がある場合のみの記入になる。

6行目に、原本の種類を記入する。35ミリポジ／ネガや、サービスプリントなのか、外部機器で制作したCGなのか、などである。

7行目に、その原本の所有者を記入。

8行目は、写真の場合は撮影者、CGの場合は制作者を記入し、原本が作成された時期も併せて記入する。

10行目は、画像に関する著作権などの記述である。

著作権が全くない場合は、フリーとし、著作権や肖像権が存在した場合は、その内容を記入する。

10行目は予備行。

そして最後にノート欄の終了を告げる。この行がないと後述する「情報ウィンドウ」でノート欄を見た時、登録してあるキーワードやノートが完全に表示されない場合があるためである。

これまで述べたキーワードやノート、そして画像に関するその他の情報は、1つの比較検討用画像に対して1つの情報ウィンドウとして持っている。情報ウィンドウの例を写真14と15に示す。

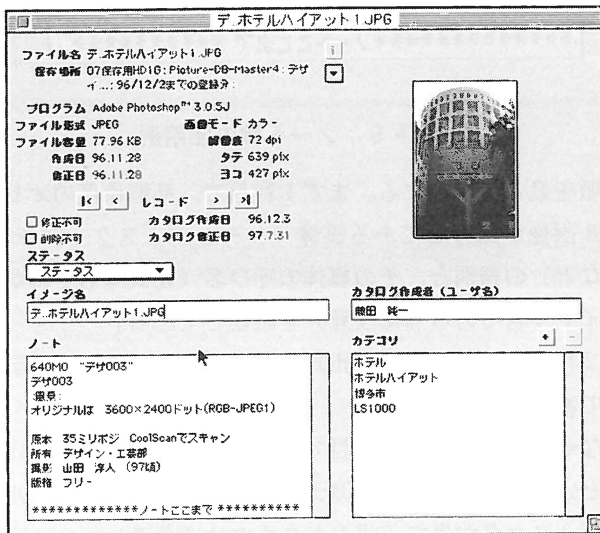


写真14 情報ウィンドウの例1

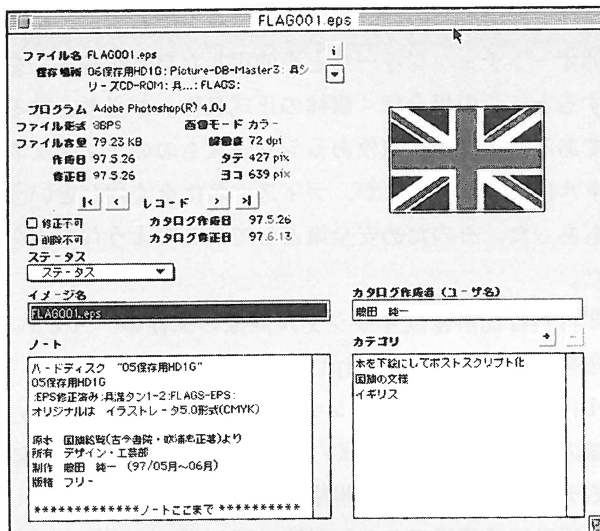


写真15 情報ウィンドウの例2

写真14は、当部が撮影・所有している画像であるが、最も一般的なスタイルである。比較検討用画像のファイル名や保存場所、データ規格やサイズなどの画像に関する情報、画像やカタログの作成・修正日に関すること、サルネーム、この画像を登録したユーザー名、そしてキーワードとノートが表示される。

写真15は、著作権フリーのイラストレーター画像であるが、これはデータに一部不良があり、原画に手直しをした後で、登録した場合の例である。

1人のユーザーだけしかこのデータベースを使用しないのであれば、この情報ウィンドウはあまり必要としないが、ネットワークで複数の端末がデータベースをアクセスに来る場合は、データベースの管理者はこの情報ウィンドウに注意を払う必要がある。

7. 結 言

以上、画像データベースの研究・試作を行ってきたが、現在、部内で限定して運用をしている。かなり複雑な構造設計で付帯情報も多かったが、検索・表示速度は実用に耐えうる性能に仕上がった。しかし実際に運用し始めるには、まだ以下の点が課題として残っている。

- (1) 自動化処理を実現するために、AppleScriptで動作するように設計したが、現在当部で所有しているCumulusはAppleScript対応版ではないため、実際の自動化処理のテストは行っていないこと。
- (2) 画像データベースとしては、登録されている画像の絶対数が不足しており、検索した時、該当画像が少ない場合があること。
- (3) 名詞キーワードが不足しており、画像を登録する際、該当する名詞がない場合、現状ではキーワードを考えながら登録作業をしており、画像登録作業に時間がかかること。

(1)に関しては、ソフトのバージョンアップなどで対処できるので、早急に行っていきたい。

(2)と(3)の問題点に関しては、現在も登録進行中であり、登録待ちの画像もあと4000件近くある。その中から、上の問題点が解決されるような画像から先に登録していくことにより、少しでも早く、実際運用を行えるようにする予定である。

参 考 文 献

- 1)松田芳久：Professional-DTP, 1, 50(1996)
- 2)遠藤俊明：インターフェース, 12, 160(1991)
- 3)木曾たかし：日経M A C, 8, 210-218, (1995)