

幾何形状画像検索システムの構築

S01A001

江野本 隆行

田中研究室

平成 14 年 2 月 24 日

目次

第 1 章 序論	2
第 2 章 LANDSAT	3
2.1 LANDSAT データ	3
2.1.1 概要	3
2.1.2 仕様	3
2.2 LANDSAT データからのイメージデータ抽出	4
第 3 章 検出の方法と実装	5
3.1 マッチング	5
3.1.1 テンプレートマッチング	5
3.2 テンプレートと実画像の比較方法	5
3.2.1 相関法	6
3.2.2 残差逐次検定法	6
3.2.3 自動閾値設定	7
3.3 実装	7
3.3.1 使用するテンプレート	7
3.3.2 検出	7
第 4 章 web ブラウザを用いたシステムの構築	9
4.1 java サブレットによる実装	9
4.1.1 サブレットとは	9
4.1.2 サブレットと CGI の違い	10
4.1.3 サブレットによる実装	11
4.2 検索結果の画像表示	11
第 5 章 終論	16

第1章 序論

航空衛星画像から幾何形状画像探索を、人的に行うのは非常に手間がかかり、膨大なデータ量の中から検索するのであれば、なおさらである。この検索作業を自動化する事により、検索にかかる時間を大幅に短縮できる可能性があるのではないだろうか。

本研究では、いくつかの検索方法を用い、陸上の衛星画像データの中から、幾何形状構造物の検索を行い、さらに検索システム的高速化、JAVA サーブレットを用いた Web システムの構築を試みる。

第2章 LANDSAT

2.1 LANDSAT データ

2.1.1 概要

今回検索に用いる画像データは、地球観測衛星 LANDSAT - 7 ETM + である。LANDSAT は Band 1 ~ 8 の 8 バンド構成であるが、そのうち RGB を表す Band1 ~ 3 のデータを用いることにする。

2.1.2 仕様

ETM+:

Enhanced Thematic Mapper, Plus エンハンスドセマティックマッパープラス (Landsat-7)

ラベル

LANDSAT-7 ETM+	2000/11/24
D 107- 35(0)	HEOC
OR HDF	
N36.04/E140.13	
NASDA-EOC R0000771 - 001 01/01	

- 仕様

衛星名 : LANDSAT - 7

センサ名 : ETM+

観測日 : 2000/11/024

中心緯度 : N36.038

中心経度 : E140.121

ピクセル数 : 6858

ライン数 : 6316

地上ステーション : HEOC

ロジカルフォーマット : HDF

データサイズ (MB) : 495.8

メディア : CD-ROM

8 バンド構成

分解能 30m

[2] より

- CD-ROM の構成

CD-ROM の構成は図 2.1 の様になる。

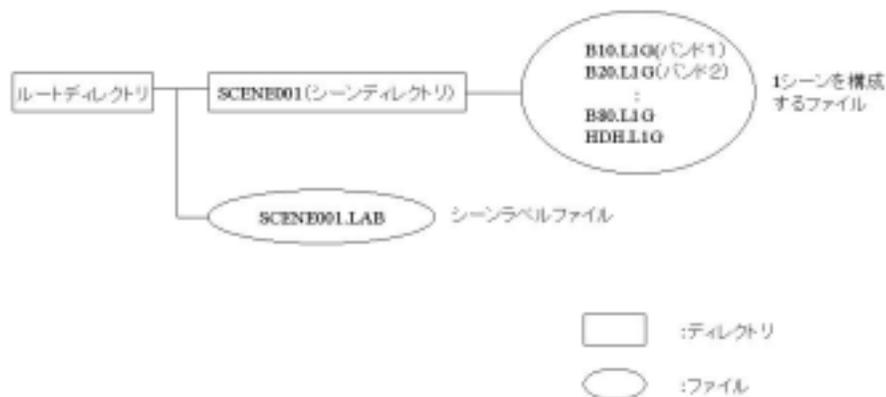


図2.1:CD-ROMのディレクトリ構成図

2.2 LANDSAT データからのイメージデータ抽出

今回使用した LANDSAT データは、一つのファイルが一つのバンドに対応しており、一つのファイルには、ピクセル数 6858、ライン数 6316 のイメージデータが格納されており、このデータの中から検索対象となる 1000×1000 ピクセルのイメージデータを取り出す。今回使用するバンドは 1 ~ 3 で、それぞれ R (赤)、G (緑)、B (青) に対応しており、この作業を 3 バンド分行う。次に、この 3 つのバンドから、データの並びが {rgbrgbrgb...rgb} となるように、一つの RGB ファイルへ格納する。この RGB ファイルを用いて検索を行う。

第3章 検出の方法と実装

3.1 マッチング

画像のマッチングは、2枚の画像の照合、あるいは1枚の画像とそのモデルの間の照合を行うことである。照合をとるレベルにより、

- 画素データレベルのマッチング
- 画像から抽出した特徴のレベルにおけるマッチング

に分けられる。

2枚の画像が平行移動で重なる関係にある場合には、前者の方法を用いる事ができる。その主な方法には、相関法、残差逐次検定法(SSDA法)がある。

今回の研究における検索方法として、特徴レベルにおけるマッチングよりも、画像データレベルのマッチングの方が向いていると考える事ができる。そこで、具体的な検索方法である、相関法とSSDA法の2つの方法を検討して、検索を行うこととする。

3.1.1 テンプレートマッチング

画像データレベルのマッチングをする上で、まず検索したいテンプレートを用意する必要がある。テンプレートマッチングは、検索目標のテンプレートを作成し、そのテンプレートを実画像上で動かしていき、テンプレートとの類似度にとって検索を行うという幾何学検索手法の一つである。その類似度を図る指標を用いるのが、相関法、残差逐次検定法である。次に、具体的にテンプレートマッチングの方法を説明する。

3.2 テンプレートと実画像の比較方法

テンプレートマッチングの方法は、まず

検索したい画像のテンプレートを作成する

次に、作成したテンプレートを検索元である画像上で比較しながらスライドさせていき、テンプレートに最も類似した画像を検出する

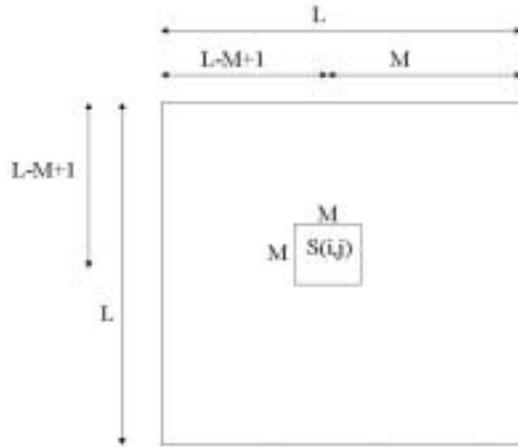
というものである。

検索は、まず検索元である画像の左上隅を基点として、そこからテンプレートのサイズ分取り出して、実際のテンプレートと比較を行う。次に、基点を一つ右に動かして、再び比較を行う。こう

して、右へずらしていき、一番右端にきたら一つ下の左端から再び比較を行っていく。そして、比較結果の良かった上位5つを記憶しておき、さらに良い候補が見つかった場合は、5つ目の候補の代わりに新たな上位5候補として組み込む。全ての検索を終了した時点で、結果のよかった上位5つを最終的な検出結果とする。

ここでテンプレートとの類似度を示す指標として、相関法と残差逐次検定法がある。

3.2.1 相関法



図のように、 $W(i, j)$ を大きさ $L \times L$ の画像とし $S(i, j)$ を大きさ $M \times M$ の画像とする。画像 W が画像 S のある部分に対応していることを知るために、

$$R(m, n) = \frac{\sum_{j=1}^M \sum_{i=1}^M S(i, j) * W(m+i, n+j)}{\sqrt{|\sum_{j=1}^M \sum_{i=1}^M S^2(i, j)|} * \sqrt{|\sum_{j=1}^M \sum_{i=1}^M W^2(m+i, n+j)|}} \quad (3.1)$$

を、 $1 \leq m \leq L - M + 1, 1 \leq n \leq L - M + 1$ の範囲で計算する。この $R(m, n)$ が最大値を取る (m, n) により画像 S と W の関係が得られる。この計算は直接計算せずにフーリエ変換を用いることもできる。相関法の問題点は計算コストが大きいことである。

3.2.2 残差逐次検定法

この方法は、Barnea and Silverman により提案され、SSDA (Sequential Similarity Detection Algorithm) と呼ばれている。誤差の累積である残差 e は、検索ウィンドウサイズを $L \times L$ とし、バンド数3とした場合

$$e = \sum_{m=1}^L \sum_{l=1}^L \sum_{b=1}^3 |A_n - B_n| \quad (n = 1, 2, \dots, L * L * 3) \quad (3.2)$$

を用いる。

ここで、 A_n を検索ウィンドウにおける画像データ、 B_n をテンプレートの画像データとする。

この累積誤差 e が最小になる上位五つを検索対象物とした。

3.2.3 自動閾値設定

残差逐次検定法 (SSDA 法) による検索とともに検索量に閾値を設定する事により高速化を図る。このとき、ウィンドウ内で最後まで誤差を加算した場合の残差に基づいて、閾値を自動的に更新していく。

このほかに、始めから全テンプレートを適用せずに部分テンプレートを適用し、その結果に基づいて全テンプレートを用いる 2 段階テンプレートマッチング法や、最も解像度の高い画像上でマッチングをせず、あらかじめ粗い画像を用意して探索効率を改善する方法などがあるが、今回の検索では、閾値を自動で設定していく方法をとる。3.3.2 項でその方法を具体的に説明する。

3.3 実装

3.3.1 使用するテンプレート

まず、検索で用いるテンプレートを用意する。テンプレートは縦横のサイズが同じ正方形の JPG イメージデータとする。ここでは、 20×20 ピクセルのサイズとする。(実際のデータは RGB の 3 バンドあるため、 $20 \times 20 \times 3 = 1200$ のデータ量となる。)

3.3.2 検出

テンプレートとして作成した JPG イメージファイルを、ImageMagick-win2k の convert コマンドにより RGB ファイルへ変換する。ImageMagick とは、画像の表示や対話的な画像操作を行なうアプリケーションで、それに附属する画像ファイル変換ツール convert によりファイル形式を変換することができる。この convert コマンドにより変換を行う。

この変換により、 20×20 テンプレートイメージファイルは、データの並びが { rgrbrgrbrgb...rgb } の RGB ファイルとなる。このテンプレート RGB ファイルと、検索対象画像 RGB ファイルを SSDA 法を用いて比較していく。

検索はまず原画像の左上隅を基点として開始する。この場合、テンプレートが 20×20 ピクセルであるため、その重なった部分である 20×20 ピクセルの画像を原画像から取り出し、SSDA 法により、誤差の累積である残差 e_1 を求める。次に、基点を一つ右に動かして再び SSDA 法により残差 e_2 を求める。このとき、以前求めた e_1 と比較し e_1 より大きければその時点で検索を終了し、さらに基点を右へ動かし、再び残差 e_3 を求める。もし、残差 e_1 より小さければその画像を、検索

対象画像とみなし、その残差 e を新たな閾値として設定し、検索を行っていく。こうして、より小さな残差の値を取る画像を検索していく。最終的に、最も残差の小さかった上位 5 つの検索対象画像を求める。こうしても検出された画像データは RGB ファイルとして作成され、Imagemagick の `convert` コマンドにより BMP イメージファイルへ変換される。

第4章 webブラウザを用いたシステムの構築

4.1 javaサーブレットによる実装

4.1.1 サーブレットとは

サーブレット (Servlet) は、J2EE(Java 2 Platform,Enterprise Edition) という、サン・マイクロシステムズが規定したサーバーサイドシステム構築の使用に含まれている機能の一つである。

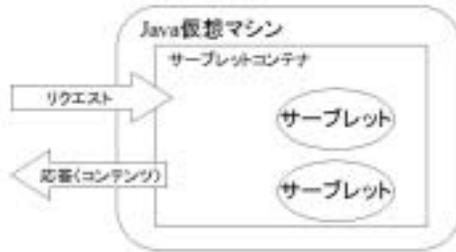
Java を使って web サーバーからクライアントに対して HTML などのコンテンツを返す役割を果たす。従来の CGI(Common Gateway Interface) とは異なり、java の持つさまざまな機能を活用できるサーブレットは Java の特性を生かした web サーバーの拡張が可能になり、非常に洗練された web システムの構築を行うことができる。

サーブレットは、web サーバー上で実行されるモジュール形式の Java プログラムで、サーブレットを動作させるには、サーブレットコンテナと呼ばれる実行環境が必要である。サーブレットコンテナは、サーブレットを作成してメモリにロードし、クライアントからのリクエストを受け付ける状態にする。また、サーブレットが作成したコンテンツをクライアントに提供したり、ロードされたサーブレットをメモリから破棄するといった制御を行う。

サーブレットコンテナは、一つの Java プログラムと考えることができ、サーブレットコンテナは、Java 仮想マシン (Java Virtual Machine : JVM) によって実行されます。

そして、サーブレットはサーブレットコンテナ内のオブジェクトとして扱われる。サーブレットコンテナは複数のサーブレットを管理するとともに、サーブレットの制御を行ってクライアントにサービスを提供する。サーブレットをプログラミングするということは、サーブレットコンテナが制御するクラス (オブジェクト) を作成することを意味します。

サーブレットコンテナとして JBuilder には Tomcat が搭載されており、非常に簡単にサーブレットを使った web システムの作成とデバックができるようになっている。

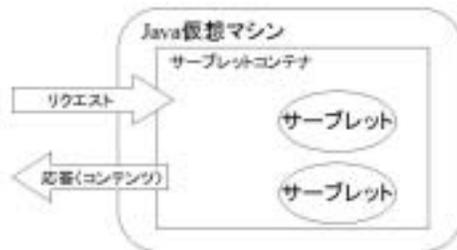


4.1.2 サーブレットと CGI の違い

また、CGI が一つのプロセスであるのに対して、サーブレットはサーブレットコンテナに管理された一つのクラスであり、サーブレットでは、サーブレットコンテナが CGI のプロセスに相当するものであるといえる。ただし、Java であるため、サーブレットコンテナは Java 仮想マシンの環境下で動作しているプロセスとなる。そして、そのプロセスが作成したオブジェクトがリクエストに対する HTML コンテンツなどを生成して、クライアントに返す処理を行う。

クライアントからのリクエストを受け取り、そのリクエストに応答できるサーブレットが存在しない場合、サーブレットコンテナはサーブレットをメモリにロードする。つまり、該当するクラスをインスタンス化して、オブジェクトを作成する処理が行われる。

これ以降、作成されたインスタンスは、サーブレットコンテナが責任を持って管理することになる。サーブレットはコンテンツの作成が終了しても消滅することはなく、作成されたインスタンスはサーブレットコンテナに管理されることになり、サーブレットコンテナが終了したり、任意の破棄要求が無い限りはメモリに常駐することになる



。

4.1.3 サブレットによる実装

今回の画像検索システムを Web 上で実現するため、この JAVA サブレットを用いる。テンプレートマッチングを行う上で必要となるのが、検索目標であるテンプレートである。今回の検索システムは、サーバー側で用意したテンプレートを、クライアントが選択して検索するというものではなく、Web システムを活かした、対話的なものにした。

今回の検索で使用するテンプレートは、クライアント側であるユーザーが作成したものを使用する。またその際のイメージファイル形式は JPG とし、テンプレートサイズは、検索対象のイメージサイズが 1000×1000 ピクセルであるため、そのテンプレートは当然の事ながら、それよりも小さくする必要があり、縦横同ピクセルのものとする。さらに、使用する衛星画像の分解能が変化したときに備え、アップロードしたテンプレートのサイズを 0.5 倍、1.0 倍、2.0 倍と変更できるようにした。

Web 上でのシステムの流れを以下に示す。

クライアントは検索したいテンプレートイメージファイルを作成し、サーバーにアップロードする

アップロード後、クライアント側でテンプレートのサイズを 0.5、1.0、2.0 倍の中から選択し、検索を実行する。

サーバーはアップロードされたイメージファイルを解析し、検索可能なファイル形式に変換し、テンプレートマッチングによる検索を行う。

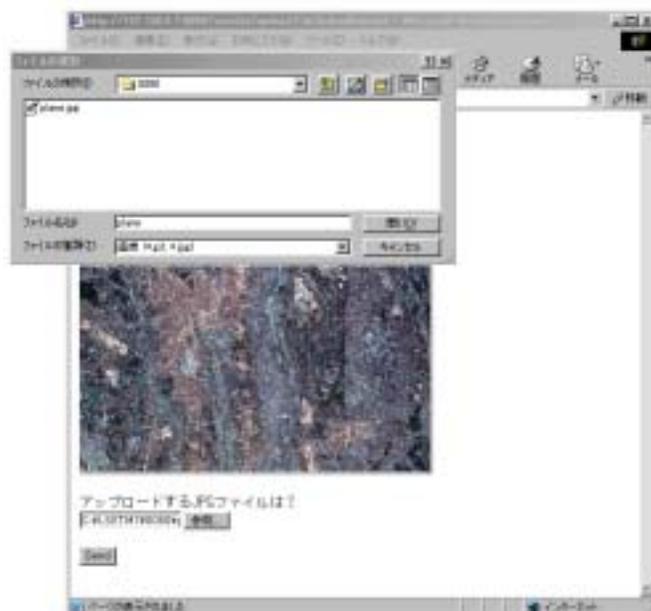
検索終了後、検索結果をクライアント側へ返す。

4.2 検索結果の画像表示

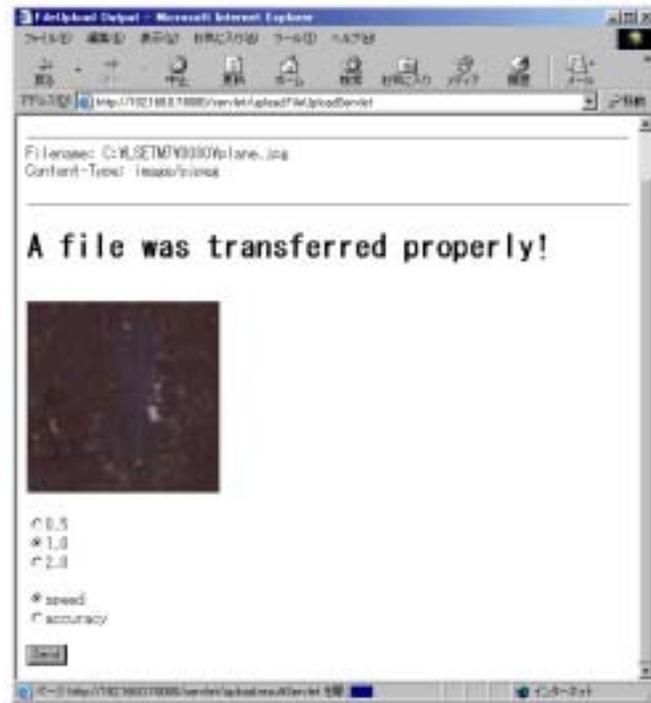
検索結果のイメージデータファイルは RGB ファイルであるが、Imagemagick の convert コマンドにより JPG ファイルに変換する。

作成した Web ページの流れ

1. テンプレートを送信。



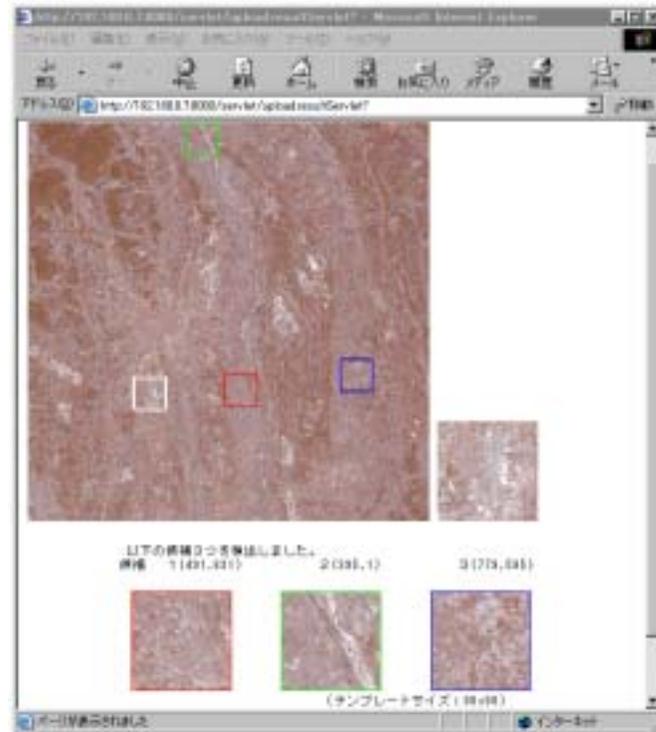
2. 送信結果。テンプレートサイズの変更、検索速度の選択。



3. 検索終了。



4. 画像表示。



第5章 終論

今回の検索システムでは、3バンドの衛星画像を用い、より、対話的な検索システムの構築を試みた。今後の課題としては、テンプレートの回転も含めた検索を行うことであり、さらに実用的なものにしていく事である。今後は、地図情報のないような地域において、携帯端末からの、現在地を中心とした建造物の検索における利用が考えられる。

参考文献・URL

- [1] 古川正寿 著
「Jbuilder サブレット開発」
ソフトバンク パブリッシング株式会社 pp.1~9
(2001)
- [2] 「RESTEC」
http://www.restec.or.jp/restec_1.html
- [3] 「Pattern Recognition/Media Understanding Society Page」
<http://yindy1.aist-nara.ac.jp/PRMU/contest/index.html>
- [4] 「Hello World」
<http://www.seto.nanzan-u.ac.jp/~amikio/NISE/member/saiki/software/servlet/top-servlet.html#FileUpload>

謝辞

本研究を進めるにあたり、最後まで丁寧なご指導、助言を下さった田中章司郎先生に深く感謝の意をここに表します。また、田中研究室の森本誠人さん、藤井宏行さん、森下旭君には、研究に関して数々の助言をいただきました。厚く御礼申し上げます。なお、本論文と本研究で作成したプログラムなどの全ての著作権は、田中章司郎先生に譲渡いたします。