

# クライアント・サーバ型 Web 対応 画像解析システムの開発

島根大学大学院 総合理工学研究科  
数理・情報システム学専攻 修士論文  
計算機科学講座  
田中研究室  
S009326  
森本 誠人

平成 14 年 2 月 20 日

# 目次

第 1 章 序論	3
第 2 章 使用データと画像解析の概要	4
2.1 衛星画像解析の原理	4
2.1.1 判別分析 (ベイズ判別法)	4
2.1.2 赤池の情報量基準 (AIC)	4
2.2 使用データの概要	5
2.2.1 Landsat TM データ	5
2.2.2 細密数値情報	6
2.3 衛星画像解析のための前処理	6
2.3.1 精密幾何補正	6
2.3.2 ミクセルの除去	7
2.3.3 細密数値情報のカテゴリ統合	8
2.3.4 教師データの取得	9
第 3 章 サーバ側処理とクライアント側処理の比較	10
3.1 サーバ側処理とクライアント側処理	10
3.2 Java による処理形態	11
3.2.1 Java サブレット	11
3.2.2 Java アプレット	11
3.3 画像生成、表示による比較	12
3.4 比較結果	14
3.5 システム開発の方針	16
第 4 章 実現する機能と要求分析	17
4.1 実現する機能	17
4.1.1 画素情報取得	17
4.1.2 教師データ選択	17
4.1.3 統計量算出	18
4.1.4 AIC と判別分析	18
4.2 要求分析	18
4.2.1 要求項目の洗い出し	18
4.2.2 ユースケース・モデルの作成	19

第 5 章 画像解析システムの実装	23
5.1 実装する環境	23
5.2 実現する機能の処理の詳細	24
5.3 解析結果の表示例	26
5.4 解析実行時間測定	30
5.4.1 測定対象	30
5.4.2 測定結果とその考察	30
第 6 章 終論	33

# 第1章 序論

## 研究の背景

今日、インターネットを利用したクライアント・サーバ型の定型システムが普及している。その理由として、

1. ネットワークの広帯域化 (ブロードバンド化) による、通信速度の高速化
2. サーバ側主体のデータ処理による、ソフトウェア等のコストダウン
3. Web ブラウザを利用することによる、利用者側の設定や機能操作の複雑さの軽減

が挙げられる。特に 1 は、画像等のマルチメディアデータ通信の促進に繋がっている。

しかし、巨大なマルチメディアデータの代表である画像データを扱う場合、文字・数値データに比べてさまざまな技術的問題が多い。特に、クライアントからの要求に対して目的の画像をリアルタイムに生成し提供する場合、画像の処理や通信がより複雑になり、時間がかかりやすい。

このような問題点を持つ中、クライアント・サーバ型環境でどのように操作性の高い画像情報システムを構築していくかを考えるのは非常に重要である。そのためにまず必要であると思われるのは、サーバ側、クライアント側で何の処理をさせるのかを明確にする、つまり効率的な分散型の開発が必須と思われる。

なおこのような状況によるものと思われるが、複雑な画像処理、画像生成をリアルタイムに行いクライアントに提供するような画像情報システムの事例はあまり広く知られていない。

## 本研究の概要

本研究では効率のよい処理形態、開発を意識し、Java のもつサーブレット、アプレットの機能を適宜組み合わせながら、クライアント・サーバ型の画像情報システムの開発を行った。開発したのは、人工衛星データの判別分析を行う画像解析システムである。利用者がブラウザから Web サーバにアクセスすることで選択した衛星データに対して判別分析処理を行い、解析結果の画像を生成してブラウザ上に返される。判別分析の結果得られるのは、地表面の物理的な状態を示す土地被覆図や地表の社会的な利用状態を示す土地利用図である。つまり利用者側において、ブラウザ上で土地被覆図や土地利用図を対話的に作成、表示することを可能とした。

## 第2章 使用データと画像解析の概要

本章では、今回開発した衛星画像解析システムの中で使用するデータ、解析項目として取り上げる判別分析(ベイズ判別法)、赤池の情報量基準(AIC)について、またこの解析のために必要となる前処理についての概要を説明する。

なお、それぞれの具体的詳細は文中の各引用文献を参照せよ。

### 2.1 衛星画像解析の原理

#### 2.1.1 判別分析(ベイズ判別法)

$p$  個の観測値からなるベクトルを  $\boldsymbol{x} = (x_1, \dots, x_p)'$  とし、 $\boldsymbol{x}$  が多変量正規分布に従うカテゴリ  $t$  ( $t = 1, 2, \dots, m$ ) に属する事後確率を  $P_t$  とする。各カテゴリで事前確率は等しいと仮定すると、事後確率を以下の式で表すことができる。

$$P_t(\boldsymbol{x}) = \frac{\exp(-\frac{1}{2}D_t(\boldsymbol{x}))}{\sum_{i=1}^m \exp(-\frac{1}{2}D_i(\boldsymbol{x}))} \quad (2.1)$$

$D_t(\boldsymbol{x})$  は、 $\boldsymbol{x}$  からカテゴリ  $t$  までの一般化平方距離と呼ばれている。式(2.1)の中から  $D_t(\boldsymbol{x})$  の部分が判別に必要な部分、つまり判別関数となり得る。以下に一般化平方距離に基づく2種類の判別関数を示す。

LDF(Linear Discriminant Function):線形判別法

$$D_t = (\boldsymbol{x} - \hat{\boldsymbol{\mu}}_t)' \hat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{x} - \hat{\boldsymbol{\mu}}_t) \quad (2.2)$$

$\hat{\boldsymbol{\mu}}_t$  : カテゴリ  $t$  に関する変数の平均ベクトル

$\hat{\boldsymbol{\Sigma}}$  : カテゴリに共通の分散共分散行列

QDF(Quadratic Discriminant Function):2次判別法

$$D_t = (\boldsymbol{x} - \hat{\boldsymbol{\mu}}_t)' \hat{\boldsymbol{\Sigma}}_t^{-1} (\boldsymbol{x} - \hat{\boldsymbol{\mu}}_t) + \log |\hat{\boldsymbol{\Sigma}}_t| \quad (2.3)$$

$\hat{\boldsymbol{\Sigma}}_t$  : カテゴリ  $t$  の分散共分散行列

LDF、QDFともに  $D_t$  を最小とするカテゴリ  $t$  に  $\boldsymbol{x}$  を分類する。

なおより詳細な原理は、引用文献[1]を参照せよ。

#### 2.1.2 赤池の情報量基準(AIC)

判別分析を行う上で、変数の数が多いとそれだけ変数の組み合わせ方はかなりの数になり、すべての変数の組み合わせの数の判別分析を行うのは、処理に時間がかかり効率が良くない。

また、高次元の変数組の方が低次元の変数組よりも判別の分類精度が優れているのではないかと  
 思われがちであるが必ずしもそうではない。限られた教師データ(分類基準データ)での使用変数  
 の増加は分類精度の低下をもたらす。これはヒューズの現象と呼ばれている。

そこで、処理効率と分類効率がともに優れた変数組を見つけるために、単純な計算でそれぞれ  
 の変数組の判別の効率の良さを評価できる指標として赤池の情報量基準 (Akaike's Information  
 Criterion : AIC) が考案されている。AIC は一般に以下のように定義されている。

$$AIC = -2 \log(\text{最大尤度}) + 2(\text{自由パラメータ数}) \quad (2.4)$$

式 (2.4) の右辺の第 1 項は分類効率に関係しており、最大尤度にはクラス間の分離の程度が当ては  
 まり値が大きいほど良い。第 2 項は処理効率に関係しており、自由パラメータ数には判別に用いる  
 変数組の次元が対応し値が小さいほど良い。AIC 基準はクラス間の分離の程度と使用変数の増加  
 に対するペナルティの 2 つを考慮している。AIC の値は低い方が良いので、クラス間の分離の  
 効果が高く、且つ低次元の変数組が AIC を最小化し、判別分析の効率においても最良であると考  
 えられる。

なお、より詳細な原理、またこの AIC を線形判別法、2 次判別法それぞれに適用させた線形 AIC、  
 2 次 AIC の詳細式については、引用文献 [1], [2], [3] を参照せよ。

## 2.2 使用データの概要

解析に使用するデータとして用いる、Landsat TM データと細密数値情報についての概要を示す。

### 2.2.1 Landsat TM データ

Landsat は、陸域の環境や資源などの観測を主目的とした地球観測衛星である。1972 年にアメリカ  
 NASA が 1 号機を打ち上げて以来、これまでに 7 号機を数える。観測センサは、MSS(Multispectral  
 Scanner System) と TM(Thematic Mapper) が使用されている。

今回は、Landsat5 号の TM センサから得られた日本の関東地域のデータから切り出した、1500pix-  
 els(横幅) × 1200(縦幅)lines の範囲のデータを使用する。観測バンド数 7band、空中分解能 30m で  
 ある。なお、Landsat についての詳細は引用文献 [4] を参照せよ。

表 2.1: 使用データの観測状況

シーン中心の観測日時	1989 年 12 月 20 日 0 時 39 分 08 秒 9 (GMT)
緯度 (シーン中心)	北緯 36 度 05 分 42 秒
経度 (シーン中心)	東経 140 度 07 分 16 秒
太陽仰角	24 度
太陽方位角	150 度

## 2.2.2 細密数値情報

本調査は、宅地関係情報の体系的整備の一環として、首都圏の主要部を対象に、宅地等の利用の現況および変化の状況を、高精度の土地利用に関する数値情報として整備し、その活性を図ることにより、今後の宅地関連政策の総合的展開に必要な基礎資料を得ようとするものである。調査方法は、空中写真や各種資料を用いて計測用図を作成し、これらを計測の上データファイルを作成する手順で行われる。計測用図は、カラー空中写真等に基づき 15 種類の土地利用区分を図化して作成している。計測は、スキャナーを用いて行い、10 メートルメッシュの詳細な単位で数値化されている。

今回は昭和 64 年（平成元年）調査の土地利用区分データを使用した。なお、細密数値情報についての詳細は引用文献 [5] を参照せよ。

## 2.3 衛星画像解析のための前処理

ここでは、衛星画像解析を行う前に必要となる前処理についての概要を述べる。前処理を行うことで、観測直後の生のデータから解析可能なデータを作成する。

### 2.3.1 精密幾何補正

人工衛星データは生のデータのままで、応用分野の定量的解析に耐えることが困難である。よって、データ解析を行うために生のデータを補正することが必要となる。

ここでは、画像座標（Landsat TM）と地上参照座標（細密数値情報）との対応関係を求めることで、人工衛星データを地図データに重ね合わせる精密幾何補正を行う。精密幾何補正は一般に以下のステップで行う。

なお、精密幾何補正の詳細と、今回用いたデータで実際に補正を行った過程で得られた情報については、引用文献 [1] を参照せよ。

#### 地上基準点 (GCP) の取得

GCP (Ground Control Point) とは、画像上の画素に対応する地上の座標点のことである。Landsat TM 画像と細密数値地図を比べて、見た目が同じ地点の座標を記録する。次の座標変換式の決定の際 GCP が必要となる。

#### 座標変換式とその係数の決定

先で取得した GCP と回帰モデルを用いて、画像座標点と地図参照座標点との対応関係を近似の関数（回帰式）として表現する。また、その係数を求める。回帰モデルとして、線形回帰モデル、多変量線形回帰モデルが利用される。

#### 人工衛星データ画素の再配列

座標変換式（回帰式）が決定すると、歪んだ画像座標に、対応する地図参照座標（補正座標）を照合することが可能になる。しかし、補正後と補正前の座標点が一致する可能性は大変低い。

そこで、少しずれて存在する隣接した補正前の座標点の輝度値から内挿して補正画像を作成する必要がある。

内挿には、以下の2つの手法がよく知られている。

- 共一次内挿法 (bi-linear 法)
- 3次畳み込み法 (cubic convolution 法)

### 2.3.2 ミクセルの除去

今回扱う Landsat TM と細密数値情報とは分解能 (画素サイズ) が異なる。Landsat TM は 30m であるのに対して細密数値情報は 10m であり、これは面積にして 9 倍の差がある。つまり Landsat TM の 1 画素に細密数値情報の 9 画素が対応していることになる。この対応関係に従って、Landsat TM に  $1 \times 1$ 、細密数値情報に  $3 \times 3$  のウィンドウを設定し、これを同時に進めていけばすべての画素の対応関係をとることができる。

この時に、Landsat TM の 1 画素に対する細密数値情報の 9 画素がすべて同一のクラスであれば 1 画素に 1 クラスの対応関係がとれ、分類処理には問題はない。このような画素をピュアピクセルと呼ぶ。しかし、さまざまな土地利用が混み合っている地域などでは、細密数値情報の  $3 \times 3$  ウィンドウに複数のクラスが存在することがある。このような、1 画素に複数のクラスがある画素をミクセル (混合画素) と呼ぶ。このミクセルの存在は、分類処理において扱いにくい問題である。

本研究ではピュアピクセルのみを分類対象として扱うことにした。ミクセルの部分も判別分析は行うが、分類精度等はピュアピクセルのみを対象としている。そのために、1500pixels(横)  $\times$  1200lines(縦) の細密数値情報から、 $3 \times 3$  ウィンドウを動かしていくことで、ミクセルを除去した 500pixels  $\times$  400lines の細密数値情報のデータを作成した。

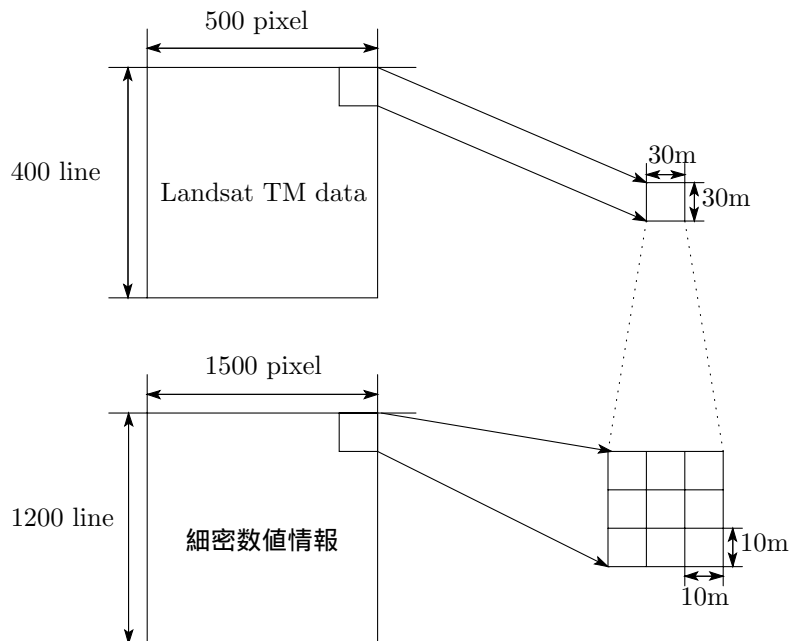


図 2.1: Landsat TM と細密数値情報の解像度の違い



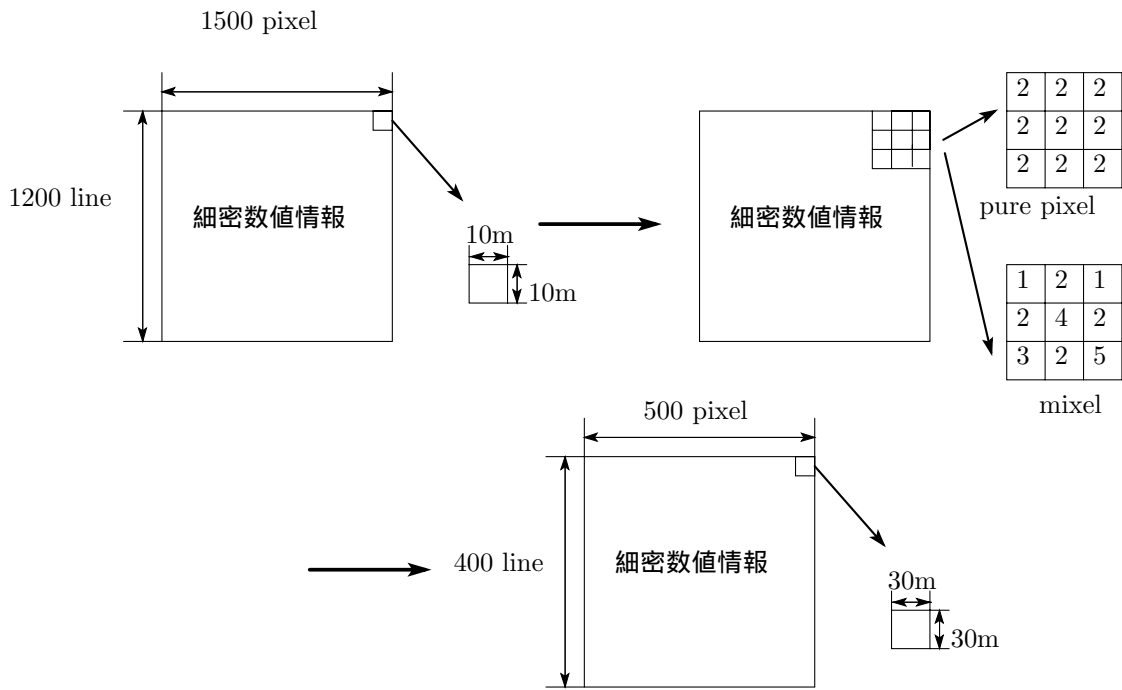


図 2.2: ミクセルの除去

### 2.3.3 細密数値情報のカテゴリ統合

細密数値情報は土地利用を扱っているのに対し、人工衛星データは土地被覆を扱っている。人工衛星データから土地被覆図の作成が判別分析を行う目的であるが、そのためにある程度細密数値情報と人工衛星データとの相関を高めておいた方が良い。

そこで今回、細密数値情報の土地利用 15 区分を土地被覆区分に変換して、結果 6 区分に統合した。統合に際しては、15 カテゴリの平均値や分布の状況をもとに、最終的な決定を主観的に下した。統合後の 6 カテゴリは、植生、田、造成中地、市街地、宅地、水域とした。

表 2.2: 統合した 6 カテゴリとピュアピクセル、ミクセル数

カテゴリ	ピュアピクセル数
[1] 植生	17470
[2] 田	20803
[3] 造成中地	4993
[4] 市街地	11229
[5] 宅地	10838
[6] 水域	1944
ピュアピクセル総数	67277
ミクセル数	132723

#### 2.3.4 教師データの取得

判別分析において必要な母集団(カテゴリ)の統計量(平均、分散共分散)等を求めたりするために、教師として人間がすでに持っている処理地域の情報から各カテゴリの代表的な区域を選出する。この選出した区域のデータを教師データと言う。(分類基準データやトレーニングデータなどとも呼ばれる。本論文では教師データで統一する。)

教師データの設定は分類において重要な意味を持つ。教師データにそのカテゴリの代表性がないと分類精度が良くない結果になるかもしれない。自分の目で見ても慎重に選択する必要がある。

なお、今回開発する画像解析システムでは、効率的に教師データ選択を行えるような機能の実現を試みた。

## 第3章 サーバ側処理とクライアント側処理の比較

本章では、画像解析システムを開発する上で、処理をサーバ側で行う場合とクライアント側で行う場合の比較を行った。ただ比較するのではなく、比較結果を基に、今回開発の画像解析システムでどのように効率的に処理分散させるかを考察した。

なお、画像データは巨大なマルチメディアデータの代表例であり、ここでの比較検討は、音声や動画といった他のマルチメディアデータを扱う場合にも共通する。

### 3.1 サーバ側処理とクライアント側処理

今日のインターネットの世界では、Webサーバから単にHTMLや文書や画像ファイルをクライアント側に送るだけでなく、プログラムを実行してその結果をHTMLで送り返したり、Webブラウザ上で何らかの動作をさせるといった処理が頻繁に行われている。Webを利用したクライアント・サーバ型システムを構築する上で、このような処理をサーバ側で行うのか、クライアント側で行うのかを考える必要がある。

#### サーバ側処理

一般にサーバ側処理では、クライアント側からの要求に応じてプログラムを実行し、結果をクライアントに送信するという流れになる。処理はすべてサーバ側で行われるので、クライアント側はWebブラウザ上に結果を表示するのみで負荷はあまりかからない。ただし、同時に多数の要求がサーバに集中するとサーバ側の負荷増により処理効率が悪くなってしまう。

サーバ側処理で知られている技術としては、CGI(Common Gateway Interface)、Java サーブレット、JSP(JavaServer Pages)、ASP(Active Server Pages)などが挙げられる。

利用例としては、ホームページのアクセスカウンタ、掲示板、オンラインショッピングの申し込み画面などに多く使われている。

#### クライアント側処理

一方クライアント側処理では、サーバ側からクライアント側に処理を行うプログラムやスクリプトを送り、Webブラウザ上で実行する。サーバから送られるHTMLファイルの中にスクリプトを埋め込んだり、タグで実行プログラムを呼び出したりすることで処理が行われる。

クライアント側の資源を利用するのでサーバ側にかかる負荷が分散される。ただし、ダウンロードサイズが大きかったり、複雑な処理が多いと、実行に時間がかかってしまう。

クライアント側処理で知られている技術としては、Java アプレットや ActiveX コントロールなどが挙げられる。

利用例では、簡単なゲームや画像を動かすといった、Web ページに動的機能を与えるのによく使われている。

## 3.2 Java による処理形態

ここでは、今回のシステム構築で実装言語として使用する Java が持つ、サーバ側実行技術 (Java サブレット)、クライアント側実行技術 (Java アプレット) について説明する。

### 3.2.1 Java サブレット

Java サブレットは、Java 対応の Web サーバ上で動作する Java プログラムである。クライアントからの要求に応じて、サーバ上で Java のプログラムを実行し、実行結果を反映した HTML を生成して Web ブラウザに表示する。

Java サブレットは、同じくサーバ側処理で最もポピュラーな CGI(Common Gateway Interface) とよく比較される。CGI との大きな違いは、CGI のプログラムが Web サーバから独立しているのに対し、Java サブレットのプログラムは Web サーバの中に組み込まれるという点である。つまり、CGI のプログラムは必要になる度に呼び出され起動し直されるのに対し、Java サブレットのプログラムは一度起動すると Web サーバが停止されるまでそのままメモリに常駐して動きつづける。よって、アクセスごとに呼び出される CGI よりも CPU やメモリに負担がかからず、高速な処理が可能となる。例えば、アクセスが取引時間内に集中し、そのつどデータベースとのやりとりが発生するオンライン証券会社のサイトなどで威力を発揮する。

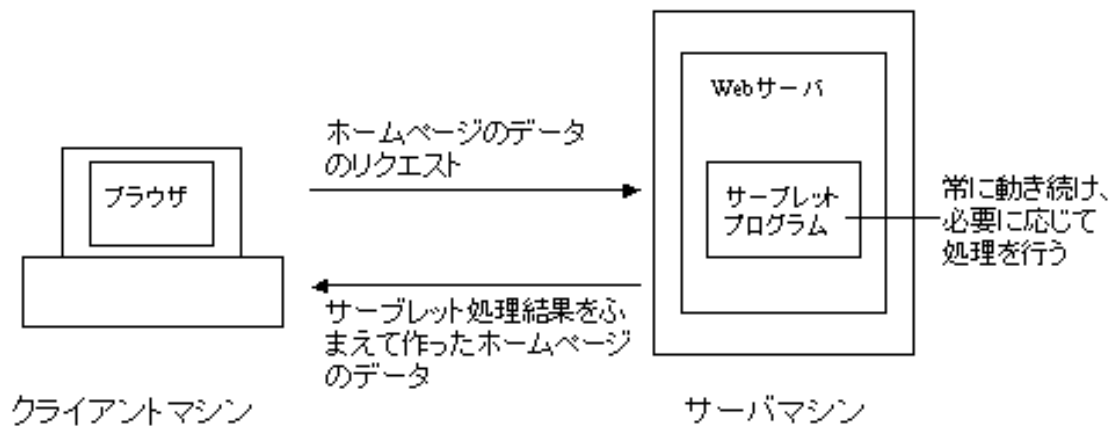


図 3.1: サブレット

### 3.2.2 Java アプレット

Java アプレットは Web サーバから Web ブラウザ側へ配信され、Web ブラウザによって実行される Java プログラムである。HTML ファイルの中でアプレットタグを記述して、アプレットを呼び出す。アプレットタグで指定されたアプレットは画像ファイルなどと同様に Web サーバからダ

ダウンロードされ、Web ブラウザに組み込んである JavaVM 上で実行される。

### アプレットのセキュリティ

Web サーバからダウンロードされて自動的にプログラムが実行されることは、Web ページの内容が豊かなものになる反面、セキュリティ的には大変脆弱となる。例えばコンピュータを故障させるような悪質なプログラムも実行されてしまう。Java アプレットでは、そもそも悪質なプログラムが作れないように、プログラムがクライアント側でできることに一定の制限を加えている。例えば以下のことが行えない。

- クライアントマシン上のデータの読み書き
- クライアントマシンにつながっている周辺機器の使用
- アプレットのコピー元 (サーバ) 以外のマシンとインターネット経由で通信

なお、悪意のないプログラムに対して上記のような制限を回避して柔軟性を持たせる方法として、ブラウザ側の Java プラグインのセキュリティ設定を変更したり、デジタル署名をプログラムに付けることが挙げられる。

## 3.3 画像生成、表示による比較

ここでは、今回開発する画像解析システムで、処理の中心をサーバ側におくかクライアント側におくかを比較、検討する。今回のシステム開発で用いる細密数値情報データの画像の生成処理とその表示を比較対象とし、その処理形態 4 パターンを決めてそれぞれを作成、実行、比較を行った。

- Java サブレット
- Java サブレット + 外部プログラム
- Java アプレット
- Java アプレット + 外部プログラム

### Java プログラムと外部プログラム

この比較では、画像生成のメインルーチンを Java 以外のプログラム (外部プログラム) で作成し実行させることも行った。これは、Java ですべて実現するのではなく、画像を扱う処理に適した言語を適時利用すれば、コーディングの負担がかからず、Java のプログラムが複雑になるのを防げるのではと考えたからである。

Java には、JavaVM の外部で動作するプログラムを扱うクラス、メソッドが用意されており、Java プログラム中から例えば Windows の実行形式ファイル (拡張子 exe) を呼び出して実行することが可能である。

今回、Java 外部プログラムとして呼び出すプログラムを Matlab で記述した。Matlab で作成したプログラム (拡張子 m) をコンパイルして実行形式ファイルに変換したものを Java プログラムの中で外部プログラムとして実行させた。

## アプレットの制限回避

前節で述べたように、アプレットには、セキュリティの問題から動作に制限がある。今回の比較において、

- Java アプレット
- Java アプレット + 外部プログラム

の2パターンの実行は、そのままではアプレットの制限に引っかかり実行不可能である。制限に引っかかる原因は、アプレットファイルとともにダウンロードする必要がある、細密数値情報データファイルと外部プログラムファイルのクライアント側ディスク上での読み書き実行の禁止である。そこで、ここで扱う一連のデータファイル、プログラムファイルが悪意を持たないという前提で、以下の手順でセキュリティの制限を緩める設定を行った。

### • クライアント側設定

1. クライアント側に Java プラグインをインストールする  
(Java プラグインは Java の最新機能 (API) をブラウザで使うためのソフトである)
2. Java プラグインの java.policy ファイルに以下の文を追加する (例)

```
grant codeBase "http://192.168.0.7:8080/classes" {
    permission java.io.FilePermission "<<ALL FILES>>","read, write, execute";
};
```

上記のように記述すると、http://192.168.0.7:8080/classes 以下のすべてのファイルについて、読み書き実行のアクセスが許可される。

### • サーバ側設定

1. アプレットを呼び出す HTML ファイルで、通常<APPLET>タグを用いてアプレットを呼び出す部分を、Java プラグインでアプレットを実行させるように書き換える。例えば、以下のように変更する。

#### 変更前

```
<APPLET
  CODEBASE = "."
  CODE      = "TestApplet.class"
  NAME      = "Test"
  WIDTH     = 500
  HEIGHT    = 400
>
</APPLET>
```

#### 変更後

```
<OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"
  width="500" height="400"
  codebase="http://java.sun.com/products/plugin/1.3
```

```

/jinstall-13-win32.cab#Version=1,3,0,0">
<PARAM NAME="code" VALUE="TestApplet.class">
<PARAM NAME="codebase" VALUE=".">
<PARAM NAME="type" VALUE="application/x-java-applet;version=1.3">
</OBJECT>

```

上記は、Internet Explorer のブラウザ上で、Java プラグイン 1.3 を使ってアプレット TestApplet.class を呼び出し、実行する。

### 処理形態比較の実行環境

以下に、今回行った画像の生成処理とその表示についての処理形態比較の実行環境を示す。

表 3.1: 処理形態比較の実行環境

比較対象	細密数値情報データの画像化 (jpeg) と Web ブラウザでの表示
処理形態	<ul style="list-style-type: none"> <li>・ Java サブレット</li> <li>・ Java サブレット + 外部プログラム</li> <li>・ Java アプレット</li> <li>・ Java アプレット + 外部プログラム</li> </ul>
サーバスペック	コンピュータ名 : BOSS2001 OS : Windows2000 Server (Service Pack2) CPU : Pentium 866MHz メモリ : 256MB
クライアントスペック	コンピュータ名 : JIVE OS : Windows98 CPU : Pentium 300MHz メモリ : 128MB
Web サーバ	Tomcat(JBuilder4 付属)
ブラウザ	Internet Explorer Ver6.0
ネットワークアナライザ	ネットワークモニタ (Windows2000 付属)
サーバとクライアントの関係	同じ Lan 内
Lan 接続環境	MELCO 社製スイッチングハブ (100MB/sec)

ネットワークモニタは、サーバが LAN から受信するフレームに関する統計をキャプチャして表示する。ここから得られる処理時間などの情報をもとに、比較検討を行った。

## 3.4 比較結果

4 パターンそれぞれについて、5 回実行してネットワーク統計を取り、その中間値を 1 つ取り上げて比較対象とした。ネットワークモニタの結果から算出した結果を以下に示す。

表 3.2: 処理形態比較結果

			送信フレーム数	受信フレーム数	送信バイト数	受信バイト数	接続時間(秒)	他の時間(秒)	合計処理時間(秒)
1	Servlet	Client	59	114	3921	144979	0.95	-	0.95
		Server	114	59	144979	3921			
2	Servlet + exe	Client	70	109	4906	145181	3.95	-	3.95
		Server	109	70	145181	4906			
3	Applet	Client	126	157	8312	213344	10.47	3.0	13.47
		Server	157	126	213344	8312			
4	Applet + exe	Client	139	188	9396	258919	11.47	8.0	19.47
		Server	188	139	258919	9396			

	主なダウンロードファイル	セキュリティ制限回避	jpeg 生成
1	html jpeg(135KB)	-	サーバ側
2	html jpeg(135KB)	-	サーバ側
3	class(2.98KB) データ (195KB) html		クライアント側
4	class(2.65KB) データ (195KB) exe(42.5KB) html		クライアント側

接続時間は、クライアントとサーバ間でデータのやり取りを行った時間である。アプレットの場合ここでは、クライアントがサーバに対して画像表示の要求を出してから、必要なファイルがダウンロードされるまでが接続時間とした。それから画像生成表示までの処理はクライアントのみで行うので、他の時間とした。

サーブレットとアプレットを比較すると、アプレットはダウンロードに少し時間がかかり、その分ネットワークを行き来するフレーム数、バイト数も多かった。とくに細密数値情報データ (195KB) のダウンロードは、キャプチャしたフレームの詳細を調べてみると、5秒くらいかかっていた。この処理はアプレットプログラムの中で行っているのでアプレットの処理の遅さも理由として考えられる。

外部プログラムの使用については、サーバ側で2から3秒くらい、クライアント側で7から8秒くらいの実行時間であった。今回の比較においては Java のみの方が処理時間は速かった。元データを画像化 (jpeg 化) して表示する処理ならば、Java だけでもコーディングは複雑ではないし、処理時間も良いことが分かった。



### 3.5 システム開発の方針

データやプログラムのダウンロード時間やアプレットの処理があまり速くない、サーバ側での処理の方が明らかに速い、といったことから、限られたアクセス人数の場合、サーバ側で処理を行なわせた方が良く考えられる。サーバ側にかかる負荷についても、サーバレットを用いることで CGI に比べて負荷が軽減されると考えられる。

ただ、Web ブラウザ上にはサーバからの結果の画像が返ってくるのみでクライアントは操作ができない。例えば、画像の個々の画素にアクセスするようなことは、クライアントが Web ブラウザ上で何らかの操作ができないといけない。よってアプレットも適時用いて、アプレットとサーバレット間で通信することで Web ブラウザ上での操作を可能とさせる。

また、Java での外部プログラムの使用については、画像生成表示の比較では、Java のみで実行した方が処理時間が速いという結果であった。ただ、あまり複雑な処理はなかったので Java だけで対応できたが、今後画像解析システムを実現していく上で、数式や行列などの計算をプログラミングしていく必要がある。これを Java だけで対応していくのは厳しいと思われる。今回、外部プログラムは Matlab で作成したが、Matlab は行列計算やその他数値計算について、プログラムしやすくまた処理も速いというメリットがある。またコンパイラにかけることで実行形式ファイル (Windows の場合、exe ファイル) に変換できる。よって適時、Java の外部プログラムを利用していくのが効率的である。

これらのことをふまえて以下に、本画像解析システムにおいて、クライアント側、サーバ側それぞれで行う処理を大まかにまとめた。

表 3.3: クライアント側、サーバ側の大まかな処理

クライアント側	サーバ側	
	サーバレット	外部プログラム (Matlab)
・画像、解析結果の表示 ・サーバレットとの通信 ・入力フォーム	・アプレットとの通信 ・外部プログラムの呼び出し ・画像データの jpeg 化	・画像解析処理のメイン ・結果の画像データ生成

## 第4章 実現する機能と要求分析

本章では、今回開発した画像解析システムにおいて実現する機能についての説明と、本システムの要求分析を行う。

### 4.1 実現する機能

#### 4.1.1 画素情報取得

表示される画像の各画素には、画素値等の情報をもっている。よって、表示される画像の各画素(ピクセル)にアクセスして、その画素情報を取得できるようにする。今回表示する人工衛星画像(Landsat)、地図画像(細密数値情報)の各画素に対して、利用者が以下の情報を取得できるような機能を考える。

- 人工衛星データ：座標(X,Y)と全バンドの画素値(0~255)
- 地図データ：座標(X,Y)とそのカテゴリ名(植生、田、等)

#### 4.1.2 教師データ選択

各クラスごとの教師データを選択を、表示画像上で簡単に指定できるようにする。教師データの選択は判別分析の解析において非常に重要である。教師データとその統計量が判別に大きな影響をもたらすからである。衛星データと地図データとから、慎重に教師データを選択しないとけない。

一般に単体マシン上で教師データを生成するには以下のようなステップを踏む。

1. 画像表示ソフトを立ち上げて、衛星データと地図データを2つ並べて(比較しやすいように)表示する。
2. 衛星画像、地図画像を見比べて、教師データとして適する領域を慎重に探す。
3. 衛星データから、教師データとして選んだ領域を切り取るプログラムを作る。
4. プログラムを実行し、教師データを生成する。

2番目が最も重要であり、これはたとえシステム化を考えても必ず人間の目で慎重に見極めて選択しないとけない工程である。よって、この部分は人間の操作に任せて、それ以外の部分をクライアント・サーバ環境下でできるだけシステムチックに、また選択が容易で分かりやすく行えるようにしたい。よって、より簡単に教師データを選択、生成しやすく、表示できるシステムを考えないとけない。

### 4.1.3 統計量算出

選択した教師データについての統計量を算出できるようにする。ここでの統計量は基本的統計量である、平均と分散・共分散である。統計量を算出することで、選択した教師データが適しているかの判断資料となる。よって、教師データ選択からその統計量をすぐに算出、結果表示できるようなシステムを実現する。

### 4.1.4 AIC と判別分析

線形、2次それぞれについてのAIC算出と判別分析である。各概要は2章を参照せよ。AICについては考えられるすべてのバンドの組み合わせのAIC値を求めてかつ順位付けをする。判別分析は使用バンドの選択等、解析前に利用者がパラメータを指定することでさまざまな状況化での判別を対話的に行い、分類画像を得ることができるようなシステムを実現する。

## 4.2 要求分析

要求分析はシステム開発における最初でかつ最も重要な工程である。どんなシステム開発においても、ただ漠然と開発を進めていくよりも、どういったものを実現したいのか、利用者に対してどのようなサービスを提供すべきなのかを決めておくことで、システム開発の方向性がしっかり見えてくる。

ここでは要求分析として、まず要求項目を洗いだしてどういったシステムにしていくのかをまとめる。そしてUML(Unified Modeling Language)のユースケース図を利用して、今回開発するシステムが利用者に提供するサービス、機能は何なのかを明らかにしていく。

### 4.2.1 要求項目の洗い出し

ここでは、どういったシステムにしていくのかを要求項目としていくつか挙げていく。

- 以下の解析を行なえる
  - － 線形判別分析
  - － 2次判別分析
  - － 線形 AIC
  - － 2次 AIC
- 上記解析の付加的情報を取得、操作できる
  - － 画素情報取得
  - － 教師データ選択
  - － 統計量算出
- 解析結果を、画像と数値データで提供する
- Web ブラウザ上で対話的に利用できる

- Lan 環境内に構築する
- 高速かつ効率よく処理が行なえる形態にする

#### 4.2.2 ユースケース・モデルの作成

ユースケースとは、簡単に言うとシステムがシステム利用者に対して提供するサービスのことである。ここではまずユースケース図を使ってシステムを外部から利用する人とユースケースとの関係を簡単な図で表す。そして、1つ1つのユースケースについての内容をユースケースドキュメントとして表すことで、システムの機能を定義していく。

ユースケース図

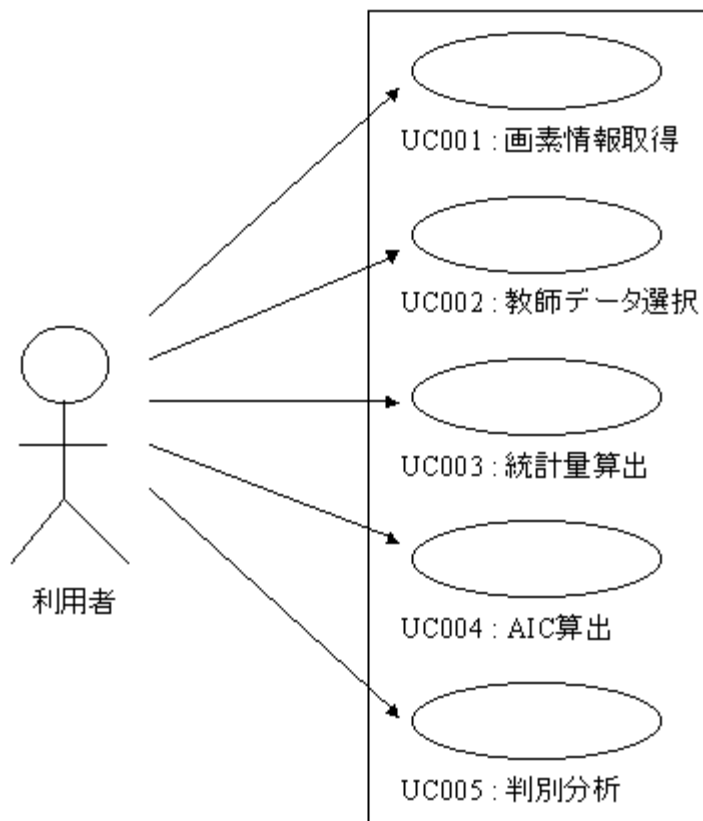


図 4.1: ユースケース図

## ユースケースドキュメント

上図で示したユースケースそれぞれについてのユースケースドキュメントを以下に示す。

### ユースケース名：画素情報取得 (UC001)

目的：表示されている衛星データ画像、地図データ画像の各画素の情報を取得する

概要：表示されている衛星データ画像、地図データ画像の各画素にアクセスし、その画素に関する情報（座標、全バンドの画素値、カテゴリ名）を取得する

アクター：利用者

事前条件：解析元となる衛星データ、地図データの画像が Web ブラウザに表示されていること

事後条件：アクセスした画素について、その座標 (X,Y)、全バンドの画素値、カテゴリ名といった情報が Web ブラウザ上に表示されていること

基本フロー：

- 1 利用者は情報を取得したい画素にマウスでクリックする
- 2 システムはクリックされた画素の座標を取得する
- 3 システムは取得した座標に対応する画素の情報を取得する
- 4 システムは Web ブラウザ上に、画素情報を表示する

備考：

- ・取得、表示する画素情報は
  - 衛星データ：座標と全バンドの画素値
  - 地図データ：座標とカテゴリ名である
- ・クリックした座標を中心とした 10 × 10 の四角枠を画像内に表示させる

### ユースケース名：教師データ選択 (UC002)

目的：表示されている衛星データ画像から教師データを選択する

概要：表示されている衛星データ画像上で、教師データとして選びたい範囲を指定し表示する

アクター：利用者

事前条件：解析元となる衛星データ、地図データの画像が Web ブラウザに表示されていること

事後条件：衛星画像上に、教師データとして指定した範囲が四角形で描写されていること

基本フロー：

- 1 利用者はカテゴリボタンを押し、選択する教師データのカテゴリを指定する
- 2 利用者は画像上に、教師データとして選択したい範囲をマウスを使って四角形で描写する

備考：

- ・実際には範囲指定するのみで生成はされていない。統計量算出や判別分析、AIC 算出時に教師データは生成され、それぞれの解析で用いられる。
- ・四角形で指定した部分のデータから選んだカテゴリに対応するデータのみを抜き出して教師データとする

ユースケース名：統計量算出 (UC003)

目的：選択したカテゴリの教師データの平均、分散・共分散を算出する

概要：選択したカテゴリの教師データの平均、分散・共分散を算出し表示する

アクター：利用者

事前条件：解析元となる衛星データ、地図データの画像が Web ブラウザに表示されており、  
画像上に教師データ選択範囲を指定していること

事後条件：平均、あるいは分散・共分散の値が Web ブラウザ上に表示されていること

基本フロー：

- 1 利用者はカテゴリボタンを押し、統計量を算出したいカテゴリを指定する
- 2 利用者は平均ボタンまたは分散・共分散ボタンを押し
- 3 システムは指定した範囲から、選んだカテゴリの教師データを生成する
- 4 システムは教師データの平均、あるいは分散・共分散を算出する
- 5 システムは Web ブラウザ上に、算出した統計量を表示する

ユースケース名：AIC 算出 (UC004)

目的：すべてのバンドの組み合わせについて線形 AIC あるいは 2 次 AIC を計算する

概要：すべてのバンドの組み合わせ（例えばバンド数 6 の場合 63 の組み合わせが考えられる）  
について線形 AIC あるいは 2 次 AIC を計算する。結果は順位付けした一覧表で表示させる

アクター：利用者

事前条件：解析元となる衛星データ、地図データの画像が Web ブラウザに表示されており、  
画像上に全カテゴリの教師データ選択範囲を指定していること

事後条件：すべての AIC を計算し、順位付け一覧表を生成して Web ブラウザに表示されていること

基本フロー：

- 1 利用者は線形 AIC あるいは 2 次 AIC ボタンを押し、処理を開始させる
- 2 システムはすべてのバンドの組み合わせについて線形 AIC あるいは 2 次 AIC を計算する
- 3 システムは計算した AIC の値を順位付けする
- 4 システムは順位付けした計算結果一覧表を生成する
- 5 システムは Web ブラウザ上に結果一覧表を表示する

ユースケース名：判別分析 (UC005)

目的：線形判別法あるいは 2 次判別法を用いて衛星データのカテゴリ判別を行う。

概要：線形判別法あるいは 2 次判別法を用いて、衛星データのカテゴリ判別を行う。

判別に使用するバンドの指定、ミクセルの扱いといったパラメータを設定して処理を行う。

結果は判別画像と判別効率表を Web ブラウザ上に表示させる。

事前条件：解析元となる衛星データ、地図データの画像が Web ブラウザに表示されており、  
画像上に全カテゴリの教師データ選択範囲を指定していること

事後条件：判別画像と判別効率表を生成し、Web ブラウザ上に表示されていること

基本フロー：

- 1 利用者は判別に使用するバンドの選択を行う

- 2 利用者はミクセルも判別を行うかの選択を行う
- 3 利用者は線形判別あるいは2次判別ボタンを押し、処理を開始させる
- 4 システムは判別処理を行う
- 5 システムは判別画像と判別効率表を生成する
- 6 システムは判別画像と判別効率表を Web ブラウザに表示する

## 第5章 画像解析システムの実装

本章では、画像解析システムの実装環境、システムにおける各機能の処理の詳細、Webブラウザでの実行表示例、実行時間測定について示す。

### 5.1 実装する環境

以下に、今回画像解析システムを実装した研究室の環境を示す。

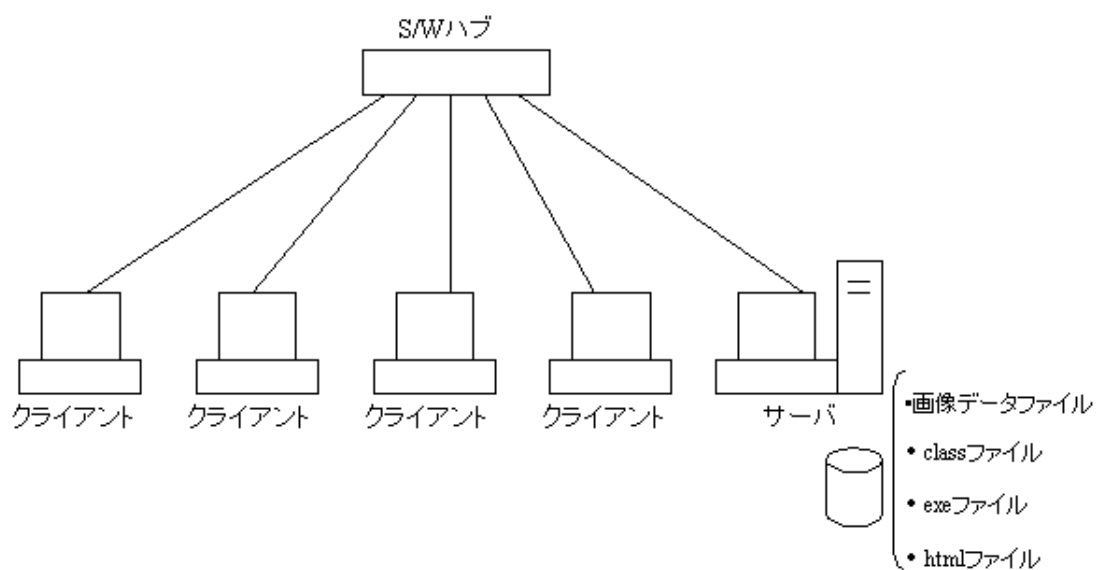


図 5.1: 実装する研究室の環境



表 5.1: 各マシン等の環境

サーバ	OS : Windows2000 Server (Service Pack2) CPU : Pentium 866MHz メモリ : 256MB Webサーバ : Tomcat(JBuilder4 付属)
クライアント	OS : Windows98 CPU : 300MHz ~ 450MHz メモリ : 128MB ~ 256MB Webブラウザ : Internet Explorer Ver5.0 ~ 6.0
サーバとクライアントの関係	同じ Lan 内
Lan 接続環境	MELCO 社製スイッチングハブ (100MB/sec)

## 5.2 実現する機能の処理の詳細

- 画素情報取得
  - アプレット (クライアント側)
    - \* 座標の取得
    - \* 座標をサブレットに送信
    - \* 画素情報をサブレットから受信
    - \* 画素情報をアプレット内に表示
  - サブレット (サーバ側)
    - \* アプレットから座標受信
    - \* 画像データから、受信した座標に対応した画素の情報を入手
    - \* 画素情報をアプレットに送信
- 教師データ選択
  - アプレット (クライアント側)
    - \* 指定したカテゴリに対する教師データ選択範囲を長方形で描写する
- 統計量算出
  - アプレット (クライアント側)
    - \* 教師データ選択範囲の四角形頂点座標、カテゴリ、統計量種類 (平均 or 分散・共分散) をサブレットに送信
    - \* 統計量 (平均 or 分散・共分散) の値をサブレットから受信
    - \* 統計量の値をアプレット内に表示
  - サブレット (サーバ側)

- \* アプレットから教師データ選択範囲の四角形頂点座標、カテゴリ、統計量種類 (平均 or 分散・共分散) を受信
  - \* 外部プログラムを呼び出して、教師データの生成とその統計量算出
  - \* 統計量 (平均 or 分散・共分散) の値をアプレットに送信
- AIC 算出 (線形、2 次)
    - アプレット (クライアント側)
      - \* 教師データ選択範囲の四角形頂点座標 (全カテゴリ)、AIC 種類 (線形 or 2 次) をサブレットに送信
      - \* 順位付け AIC 一覧表をブラウザ上に表示
    - サブレット (サーバ側)
      - \* アプレットから教師データ選択範囲の四角形頂点座標 (全カテゴリ)、AIC 種類 (線形 or 2 次) を受信
      - \* 外部プログラムを呼び出して、すべての AIC を算出し、順位付け AIC 一覧表を生成
      - \* 順位付け AIC 一覧表をブラウザに送信
- 判別分析 (線形、2 次)
    - アプレット (クライアント側)
      - \* 教師データ選択範囲の四角形頂点座標 (全カテゴリ)、使用バンド、分類画像の各カテゴリの色、判別種類 (線形 or 2 次) をサブレットに送信
      - \* jpeg 画像と判別効率表をブラウザ上に表示
    - サブレット (サーバ側)
      - \* アプレットから教師データ選択範囲の四角形頂点座標 (全カテゴリ)、使用バンド、分類画像の各カテゴリの色、判別種類 (線形 or 2 次) を受信
      - \* 外部プログラムを呼び出して、判別分析を行い、結果の分類画像データと判別効率表を生成
      - \* 分類画像データを画像化 (jpeg 化)
      - \* jpeg 画像と判別効率表をブラウザに送信

### 5.3 解析結果の表示例

解析対象となるデータを画像化 (jpeg 化) して表示するために、パラメータを指定してサーバ側に送る。

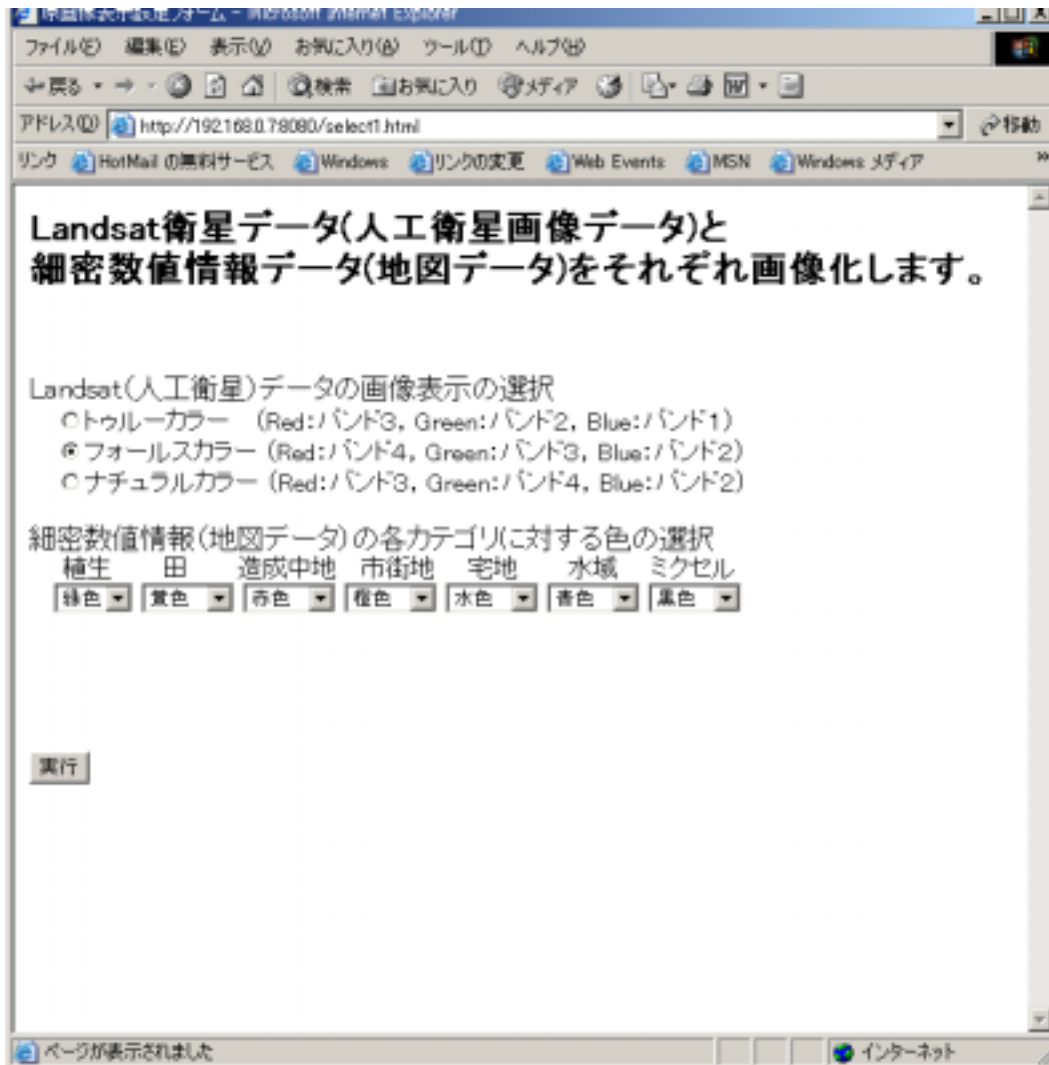


図 5.2: 原画像表示設定フォーム

jpeg で画像化した衛星データ (左)、地図データ (右) である。それぞれ画像上にマウスでクリックすることで、その座標に対応する画素情報 (画素値、カテゴリ) をサーバから取得し表示する。レスポンスはあまりかからない (1 秒以内)。

また、衛星画像上でカテゴリごとに教師データの選択範囲を指定できる。画像下のカテゴリボタンを押し、画像上で長方形を描写することで指定できる (何度でも描写やり直し可)。

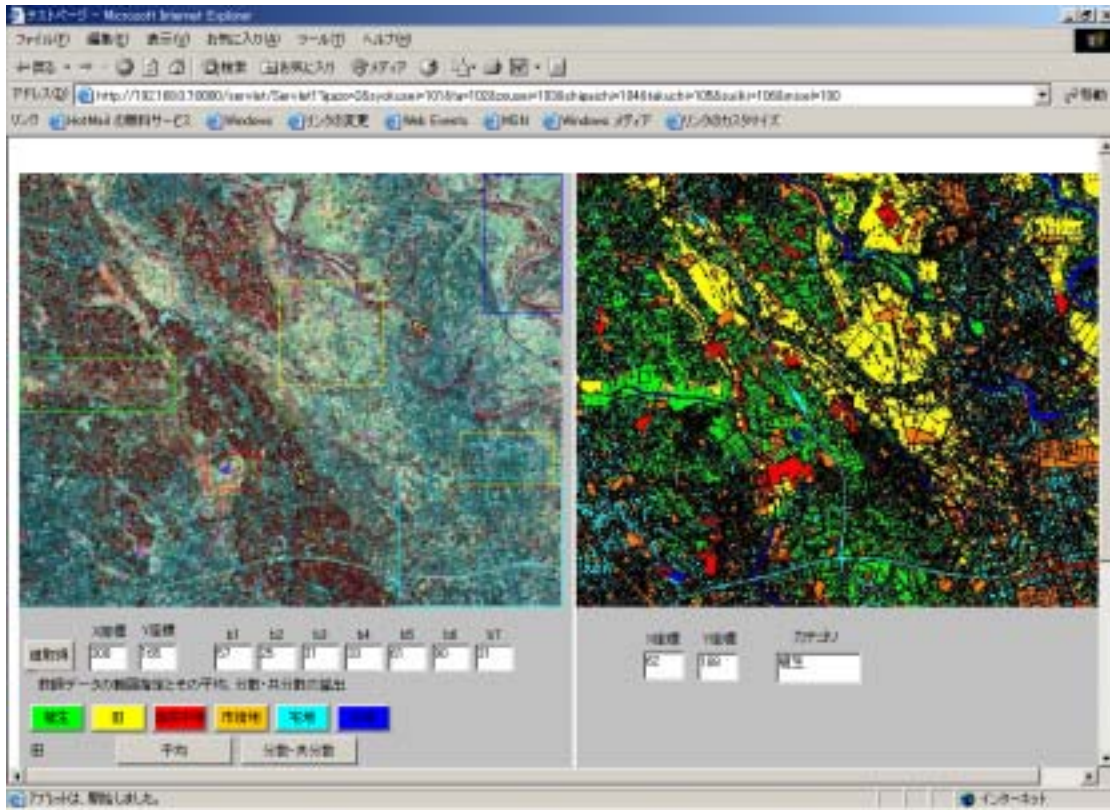


図 5.3: 原画像表示例

前ページで指定した教師データに対して、その統計量を求めている。画像上に選択した状態で、カテゴリボタンを押し、平均または分散・共分散ボタンを押すことで、指定したカテゴリの教師データの統計量をサーバ側で計算し、表示される(統計量の計算は Java の外部プログラム (Matlab で作成) 呼び出しで行われる)。

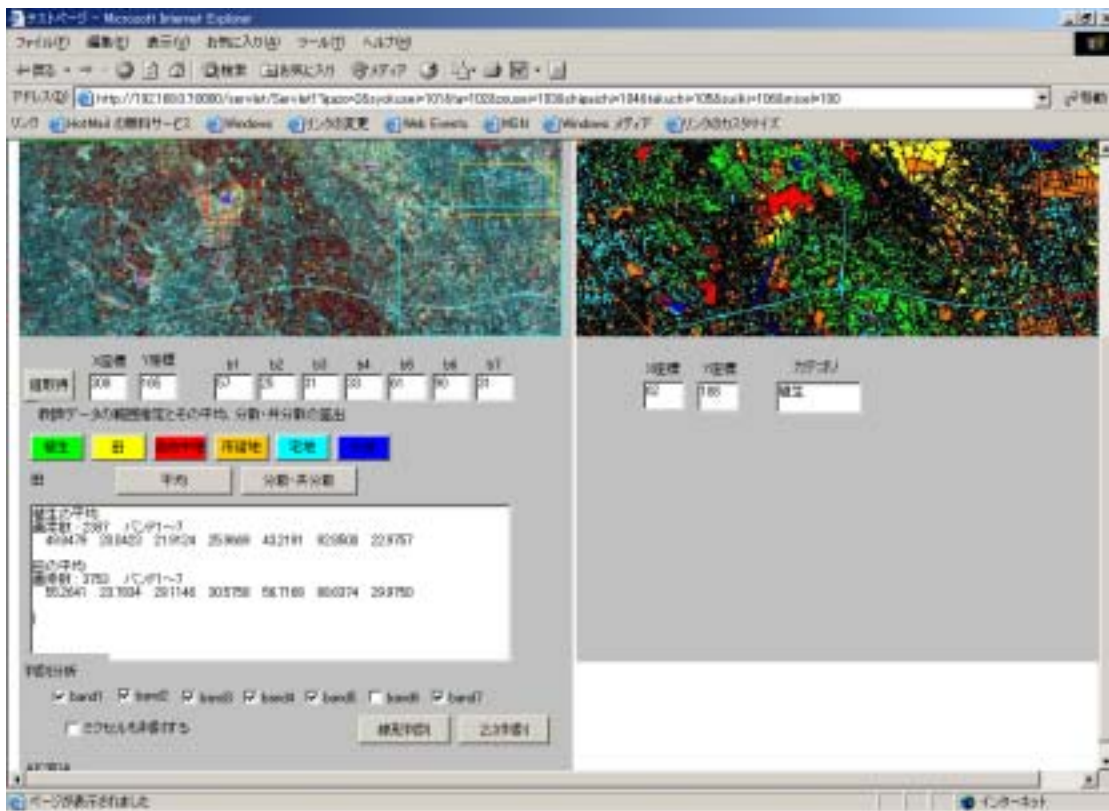


図 5.4: 統計量表示例

前ページで、全カテゴリについて教師データの範囲を指定し、判別に使用するバンドの選択、ミクセルを扱いを決めて、線形または2次判別ボタンを押すことでサーバ側で判別処理が行われ(この処理もJavaの外部プログラム(Matlabで作成)呼び出しで行われる)、結果の分類画像をブラウザに返す。右の数字は、地図データと比較した分類精度である。以下の画像は、バンド1~5,7を使用した時の線形判別の結果で、約20~23秒のレスポンスがかかった。

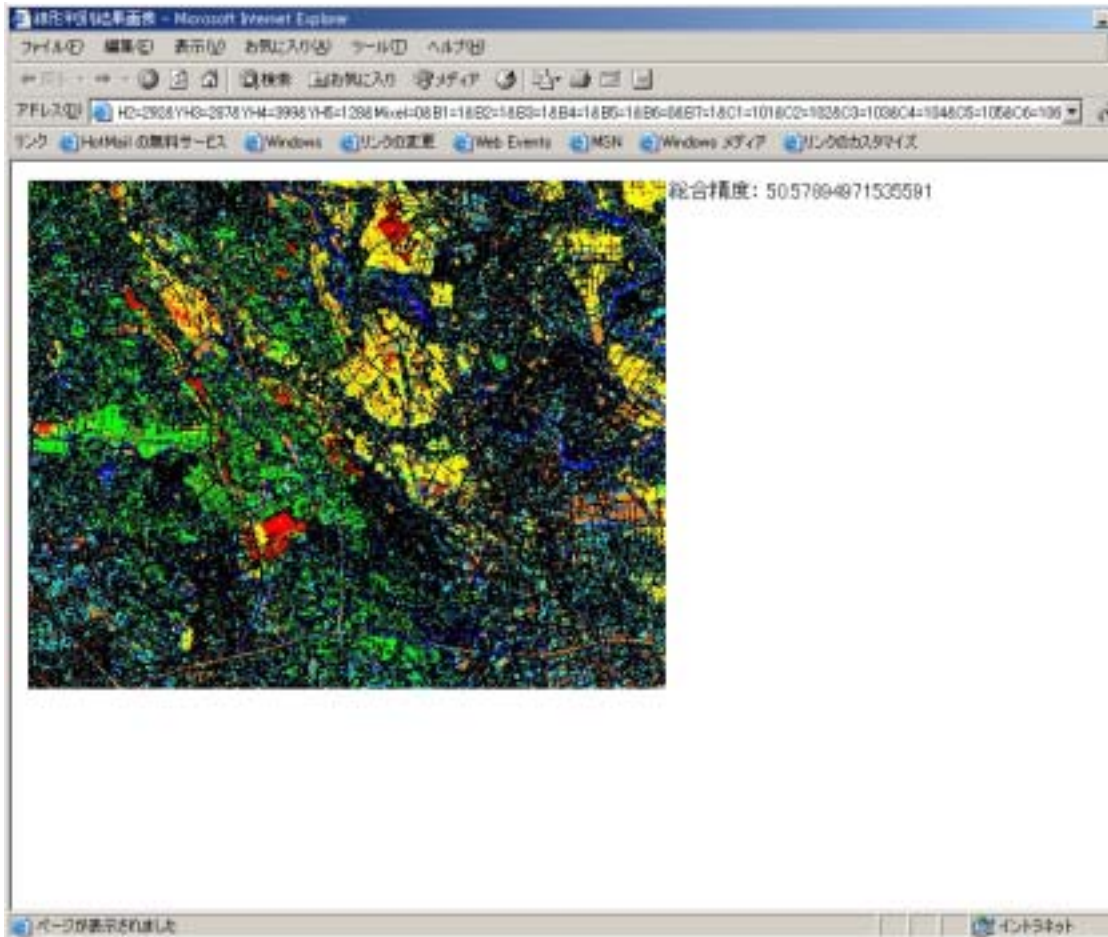


図 5.5: 線形判別画像例

## 5.4 解析実行時間測定

### 5.4.1 測定対象

特定の指定した教師データに対して、システム内で判別分析（線形、2次）、AIC（線形、2次）算出を実行した時の実行時間を示す。教師データの範囲は、図 5.3 とした。以下に今回選択した各カテゴリの教師データの総数を示す。

表 5.2: 今回選んだ各カテゴリの教師データの総数

カテゴリ	教師データ数
[1] 植生	2387
[2] 田	3753
[3] 造成中地	644
[4] 市街地	1062
[5] 宅地	1691
[6] 水域	634
合計	10171

線形判別、2次判別はバンドの組み合わせから 6 パターンを選び実行を行った。判別分析の実行時間は、利用者がブラウザ（タブレット）上の処理開始のボタンを押してサーバに要求してから、サーバ側で教師データ生成、判別分析実行、結果の jpeg 画像生成等の一連の処理を行い、jpeg 画像をクライアント側に返すまでの時間とした。

線形 AIC、2次 AIC はバンド 6 を除くバンド 1~5,7 のすべての組み合わせ（63 組）の AIC を一度にすべて算出した。AIC 算出の実行時間は、利用者がブラウザ（タブレット）上の処理開始のボタンを押してサーバに要求してから、サーバ側で教師データ生成、AIC 算出、結果のテキスト生成等の一連の処理を行い、結果をクライアント側に返すまでの時間とした。

判別分析（6 パターン）、AIC（2 パターン）、それぞれのパターンについて 5 回実行してネットワーク統計を取り、その中間値を 1 つ取り上げて結果とした。なお、測定には Windows2000 付属のネットワークモニタを用いて、そこから得られた情報を基に実行時間を算出した。

### 5.4.2 測定結果とその考察

表 5.3: 線形判別処理時間結果例

使用バンド数 (使用バンド)		送信フレーム数	受信フレーム数	送信バイト数	受信バイト数	実行時間 (秒)	jpeg サイズ (バイト)
6バンド (1,2,3,4,5,7)	Client	72	113	5367	145851	21.27	138970
	Server	113	72	145851	5367		
5バンド (2,3,4,5,7)	Client	67	112	5067	145838	17.63	139001
	Server	112	67	145838	5067		
4バンド (2,3,5,7)	Client	67	114	5067	145825	16.00	138902
	Server	114	67	145825	5067		
3バンド (2,5,7)	Client	69	111	5187	145219	14.36	138446
	Server	111	69	145219	5187		
2バンド (1,3)	Client	63	108	4827	145883	13.27	139284
	Server	108	63	145883	4827		
1バンド (2)	Client	64	105	4887	140986	9.53	134549
	Server	105	64	140986	4887		

表 5.4: 2次判別処理時間結果例

使用バンド数 (使用バンド)		送信フレーム数	受信フレーム数	送信バイト数	受信バイト数	実行時間 (秒)	jpeg サイズ (バイト)
6バンド (1,2,3,4,5,7)	Client	75	116	5547	146695	19.09	139664
	Server	116	75	146695	5547		
5バンド (1,3,4,5,7)	Client	66	115	5007	146830	17.69	139841
	Server	115	66	146830	5007		
4バンド (2,4,5,7)	Client	68	114	5127	146986	16.31	140063
	Server	114	68	146986	5127		
3バンド (1,3,7)	Client	67	115	5067	146329	14.80	139352
	Server	115	67	146329	5067		
2バンド (2,4)	Client	63	115	4827	148369	13.67	141380
	Server	115	63	148369	4827		
1バンド (2)	Client	70	110	5247	149277	9.53	142558
	Server	110	70	149277	5247		



表 5.5: AIC 処理時間結果例

AIC の種類 (使用バンド)		送信フレ ーム数	受信フレ ーム数	送信バ イト数	受信バ イト数	実行 時間 (秒)	jpeg サイズ
線形 AIC (1,2,3,4,5,7)	Client	10	11	1618	3337	3.61	-
	Server	11	10	3337	1618		
2 次 AIC (1,2,3,4,5,7)	Client	10	11	1618	3337	3.73	-
	Server	11	10	3337	1618		

図 5.3, 表 5.2 による教師データ、200000 画素、バンド数 6(バンド 6 を除く 1~7 のバンド) の衛星データに対する、判別分析、AIC 算出の実行時間等の結果は以上ようになった。

判別分析、AIC 算出共に、線形と 2 次での実行時間はあまり変わらなかった。判別分析では約 9 秒から 22 秒の実行時間であり、クライアントに送信される jpeg 画像も 135 ~ 140KB 程度で通信時間もあまりかからないようであった。AIC では約 3 から 4 秒の実行時間であった。実行時間の多くが教師データ生成であり、実際の AIC 算出部分は短時間の処理で結果を得ることができた。また結果はテキストデータのみで少ないデータのやりとりですんだ。

判別分析、AIC 共に、Matlab の行列、ベクトル計算を可能な限り利用して多くのデータを一気に処理することができた。これにより、for 文中心のプログラムよりもはるかに高速に結果を得ることができた。ただし、一度に多くのデータを処理することでサーバ側のメモリ等資源をかなり消費するのではと考えられる。今回のサーバ環境において 500 × 400 × 6 サイズの衛星データ処理には耐えることができたが、数十、数百メガバイトクラスのデータサイズを扱う場合はその処理に耐えられない可能性が高くなる。また for 文中心のプログラムで実行させると相当多くの実行時間を必要とする。よって、まず AIC 算出によりある程度の判別予測をたてる必要があるだろう。

なお、同教師データにおける判別分析、AIC の詳細結果 (精度、AIC 順位等) については、引用文献 [1] を参照せよ。

## 第6章 終論

終論として、本研究の課題や今後の展望等を述べる。

### 画像の拡大・縮小機能

今回の画像解析システムに限らず、一般に画像を扱う上で必要と思われるが今回実現できなかった機能として拡大・縮小機能がある。Web ブラウザ上でいかに使用しやすいように画像の拡大・縮小を実現させるか、また拡大・縮小することにより、元画像との座標の対応関係をいかにとるか考える必要がある。

### 教師データの指定方法

判別分析で必要である教師データの指定を Web ブラウザ上で Java アプレットの機能を用いてグラフィックに選択できるようになった。今回は長方形で範囲を指定するようにしたが、さらに多角形で範囲指定できるようになれば、より詳細にかつペイント感覚で選択できるであろう。

### Web ブラウザからモバイル端末への展開

近年、携帯電話をはじめとしたモバイル端末の進化と普及が著しく、画像を代表とした、マルチメディアデータも扱うことができるようになってきた。

モバイル端末から要求することにより、現在地周辺の最新の衛星画像が入手可能になるだろうし、そこから今回のような土地被覆図や土地利用図が生成、入手できると思われる。このようにリアルタイムに画像を生成、送信されることで、例えば、

- 災害発生などの緊急時に、現在地周辺の被害状況を知るための高精度の地形図を短時間で作成、提供する
- 湖の汚染進行状況を現地調査するために、同地域の数年前の衛星画像と今日最新の衛星画像から両者の土地被覆図を作成、比較して汚染が進んでいる所をその場で特定し、すぐに調査に取りくめる

といったことが可能となるだろう。

## 謝辞

本研究にあたり、最後まで熱心な御指導をいただきました田中章司郎先生には、心より御礼申し上げます。

また、田中研究室生の藤井宏行君、江野本隆行君、森下旭君、荒木俊太郎君、富岡俊幸君、中村吉郎君、平田純一君、辰己圭介君、貫目洋一君、驒範之君、高木明君、森岡慶彦君には、本研究に関して数々の御協力と御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等のすべての著作権を田中章司郎教授に譲渡致します。

## 引用文献

- [1] 森本 誠人  
「赤池の情報量基準による人工衛星データ判別効率の予測」  
島根大学総合理工学部 卒業論文,2000
- [2] Y.Fujikoshi  
A criterion for variable selection in multiple discriminant analysis,  
Hiroshima Math. J. 13, 203/214 (1983)
- [3] Pynnönen, Seppo (1998)  
Testing for additional information in variables in multivariate normal  
classification with unequal covariance matrices.  
*Acta Wasaensia*, No23,pp.87-88
- [4] 宇宙開発事業団 地球観測センター編集  
「地球観測データ利用ハンドブック-ランドサット編・改訂版」  
(財) リモート・センシング技術センター,1990
- [5] 建設省国土地理院監修  
「細密数値情報(首都圏宅地利用動向調査)説明書」  
(財) 日本地図センター,1988
- [6] Cay S. Horstmann, Gary Cornell  
「コア Java2 Vol.1 基礎編」  
「コア Java2 Vol.2 応用編」  
(株) アスキー, 2000
- [7] 日本サン・マイクロシステムズ株式会社編者  
「Java プログラミング サブレット & JavaWebServer」  
(株) サイエンス社,1998
- [8] 「Mamezou」  
<http://www.mamezou.com>