

# ブラウザ環境における 河川流域境界データ表示システムの試作

島根大学 総合理工学部

数理・情報システム学科 計算機科学講座 卒業論文

田中研究室

S023041 島村和義

2006年2月10日

# 目次

## 第1章 序論

## 第2章 標準地域メッシュ

- 2.1 概要
- 2.2 第1次地域区画
- 2.3 第2次地域区画

## 第3章 数値地図情報（流域境界）データ

- 3.1 概要
- 3.2 フォーマット
- 3.3 データの加工

## 第4章 JMC マップデータ（海岸線）データ

- 4.1 概要
- 4.2 フォーマット
- 4.3 データの加工

## 第5章 数値地図 50m メッシュ（標高）データ

- 5.1 概要
- 5.2 フォーマット
- 5.3 データの加工

## 第6章 Java サブレット

- 6.1 Java サブレットとは
- 6.2 なぜ Java サブレットなのか

## 第7章 X3D

- 7.1 X3D とは
- 7.2 なぜ X3D なのか

## 第8章 ブラウザ環境でのグラフィック表示

## 第9章 終論

## 第1章 序論

いままでの情報システムでは、行政境界ごとに区別されているのが主であった。それらは行政内での、または行政ごとの検索・表示などには向いていても、河川の汚染や洪水といった流域・地域単位での災害では対応するのに不十分である。

そこで、河川流域境界データを主な単位とし利用できるシステムの基礎的な仕組みを作ることが本研究のテーマである。

その基礎的な仕組みとして本研究では河川流域境界データ表示システムを試作した。

河川流域境界データ表示システムとは、河川流域境界データ（数値地図情報 KS\_273：国土地理院作成）をグラフィック化するものである。河川流域境界データは 2 次元の座標値データなので、まず構造化を行い、足りない部分を海岸線データ（JMC マップ：日本地図センター作成）で補う。そして標高データ（数値地図 50m メッシュ：国土地理院作成）を付け加え、3 次元のポリゴンデータとしてグラフィック化することにした。

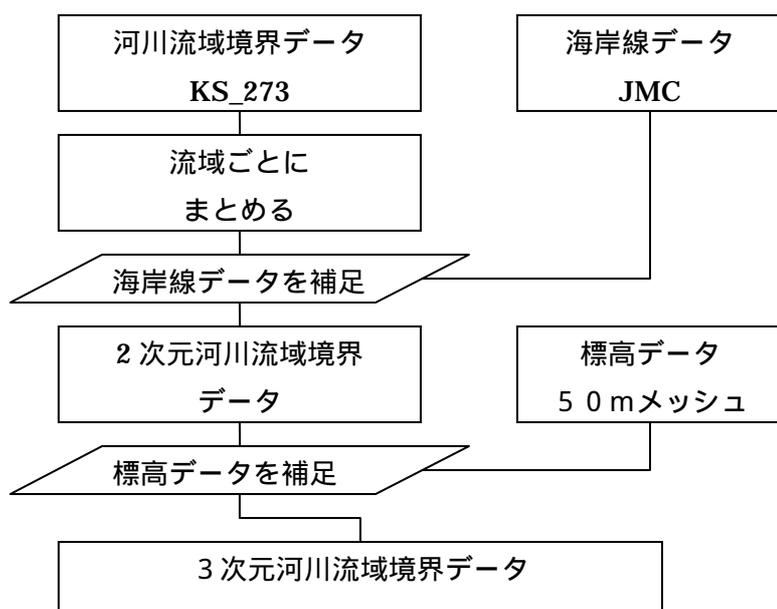


図 1 - 1：構造化の流れ

グラフィック化にあたって、サーバを通して Java サブレット（サーバサイドで動く Java プログラム）を使い、必要に応じた X3D（オブジェクト記述言語）データを作り、ブラウザに返すことにした。これにより汎用性が高く、一般の多くの人が利用できるようになる。

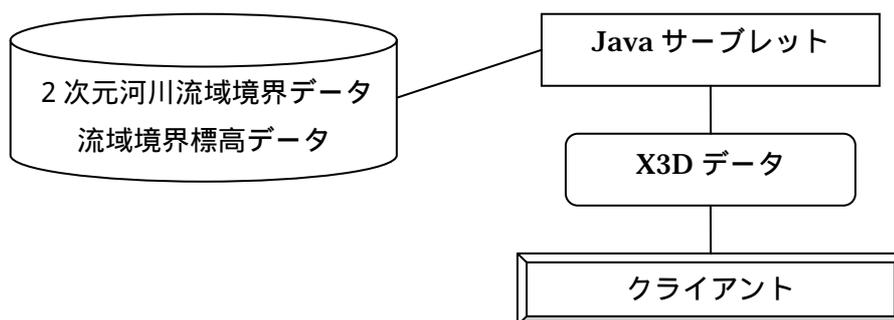


図 1 - 2 : ブラウザ表示の流れ

## 第 2 章 標準地域メッシュ

### 2.1 概要

全国的な規模で数値地図情報を整備する場合、“標準地域メッシュ”方式が広く採用されている。本研究に用いたデータも全てその形式に基づいたデータである。一定間隔で正方形に土地を分割 1 次 > 2 次 > 3 次メッシュ 階層的に一定単位で全国を分割する。

本研究では第 2 次地域区画 ( 2 次メッシュ ) のデータを利用している。

### 2.2 第 1 次地域区画

全国の地域を 1 度毎の経線と 3 分の 2 度 ( 40° ) 毎の緯線によって縦横に分断して第 1 次地域区画 ( 国土地理院薄幸の縮尺 20 万分の 1 地勢図の通常の区画に相当する範囲 ) が作られている。

第 1 次地域区画の地域メッシュ・コードは、区画南端の経度を 1.5 倍した 2 桁の数字と、西端経度から、100 を引いた 2 桁の数字とを 1. 緯度 2. 緯度の順に組み合わせた 4 桁の数字として定義されている。

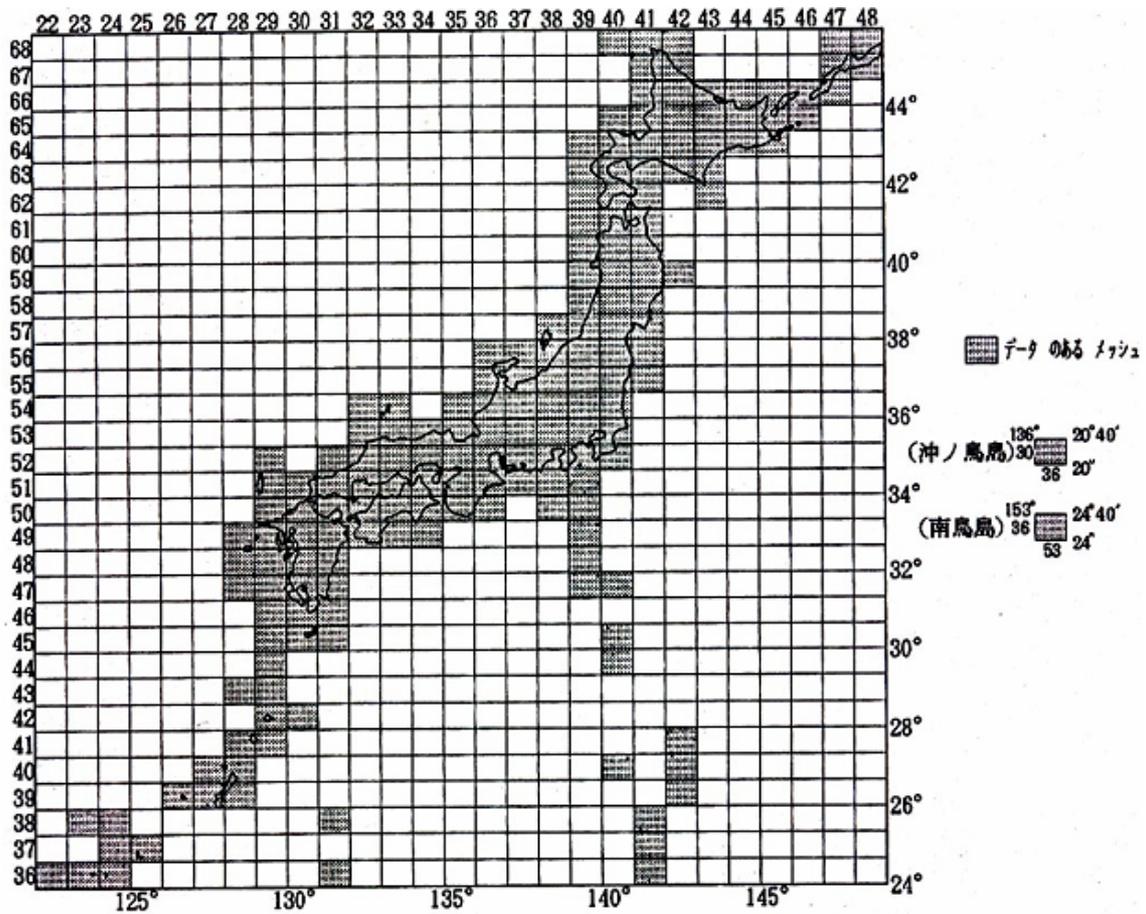


図 2 - 2 - 1 : 1 次メッシュの全国図

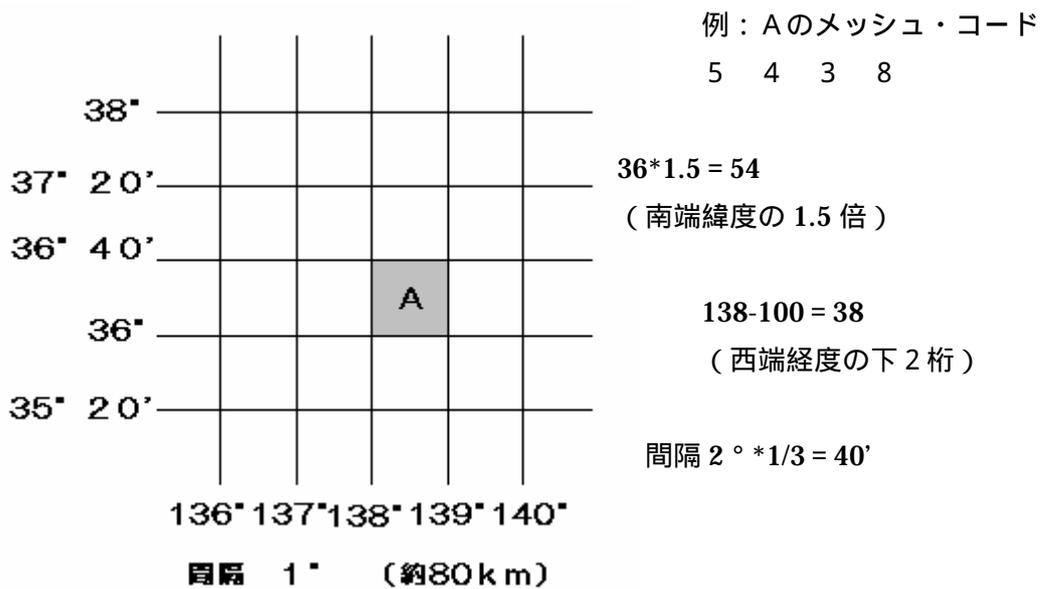


図 2 - 2 - 2 : 1 次メッシュの緯度・経度

## 2.3 第2次地域区画

第1次地域区画の縦横をそれぞれ8等分して第2次地域区画（国土地理院薄幸の縮尺2万5千分の1地形図の通常の区画に相当する範囲）が作られている。

第2次地域区画の地域メッシュ・コードは、第1次地域区画を8等分した区画に、経線方向については南から、緯線方向については西から、それぞれ0から7までの数値を付け、これを1．経線方向 2．緯線方向の順に組み合わせた2桁の数字として定義されている

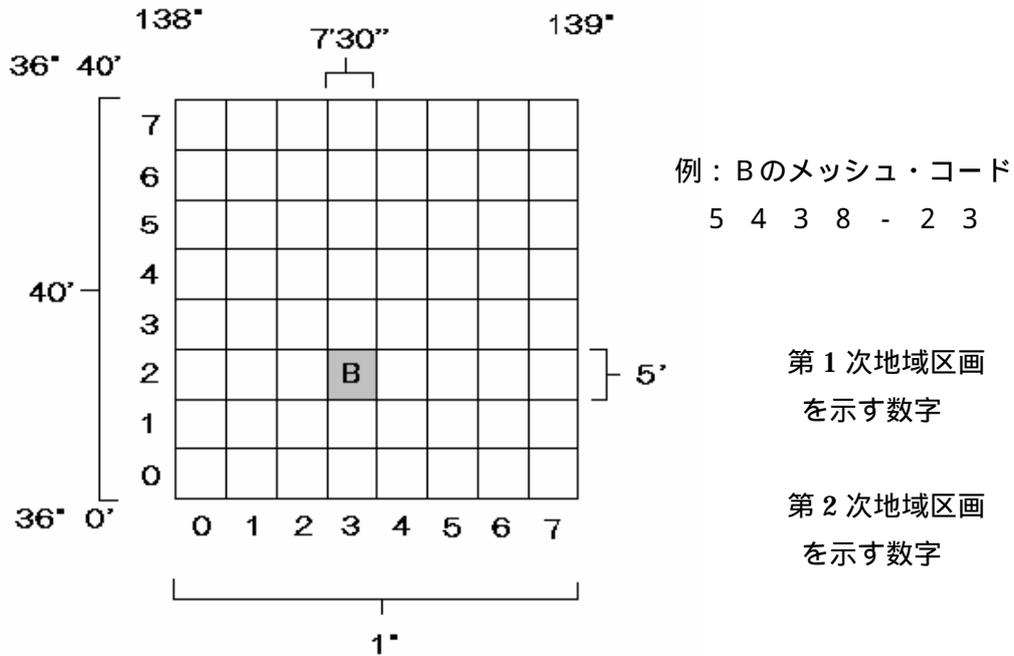


図2-3-1：2次メッシュの緯度・経度

## 第3章 数値地図情報（流域境界）データ

### 3.1 概要

数値地図情報（流域境界）は本研究の基盤となるデータである。  
数値地図情報として以下のファイルが CD-R メディアに入っている。

KS-200-1（土地利用面積）

KS-202-1（1 / 10 細分区画土地利用データ）

KS-271（河川単位流域台帳）

KS-273（流域界，非集水界線位置）

KS-602（3次メッシュ流域・非集水域面）

KS-617（1 / 10 細分方眼流域・非集水域）

今回使用したデータは KS-273（流域界，非集水界線位置）である。  
これには雨が降った際、ある特定の河川に集中すると考えられる地域を  
分合流点で分けた単位流域毎の座標値が入っている。

KS-273（流域界，非集水界線位置）は、2万5千分1相当のベクトル形式の地図データ  
（平面）である。データ項目は、図葉番号、流域界、非集水界線区分[左側（水系域コード、  
単位流域コード）右側（水系域コード、単位流域コード）]、始終点タグ、非集水界線[空欄、  
始終点タグ]、2次メッシュ正規化位置である。

データは、全国が1つのファイルにまとめられている。本研究では島根県の宍道湖に流  
れ込む斐伊川水系の単位流域データを抜き出し、並び替え、ポリゴンとして利用できるよ  
うにした。

## 3.2 フォーマット

- ファイル形式は IBM 固定長でレコード長は 40 バイト
- レコード件数は 1690929 件である。
- 各レコードは、復帰・改行コードで区切られている。
- 文字コードは、シフト JIS コードを使用している。
- レコード 1 行に座標値 1 点ずつ入っている。
- 座標値は 2 次メッシュごとに正規化されており、左下が(0,0)、右上が(100000,100000)となっている。  
なお、x座標は右(東)方向、y座標は上(北)方向となる。

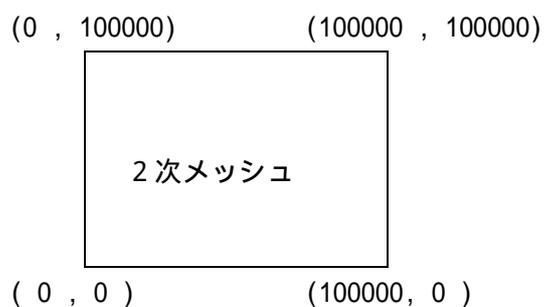


図 3 - 2 - 1 : 流域界 , 非集水界線位置データの座標値

データは以下のような内容になっている。

\* 流域界・非集水域界区分が“ 0 ”のとき

項目		桁数	累積桁数	データ形式	
2次メッシュ・コード		6	6	9(6)	
計測会社番号		1	7	9	
流域界・非集水域界区分		1	8	9	
流域界線	左側	水系域コード	5	13	9(5)
		単位流域コード	4	17	9(4)
流域界線	右側	水系域コード	5	22	9(5)
		単位流域コード	4	26	9(4)
終 始点タグ		1	27	9	
2次正規化		6	33	9V9(5)	
位置		6	39	9V9(5)	
スペース		1	40	X	

表 3 - 2 - 2 : 流域界・非集水域界データのフォーマット 1

例 : 533217508707401668707401672005053006153

備考 : 流域界・非集水域区分の“ 0 ”は流域界線

水系域コード : 数値地図ユーザズガイド

p 284 ~ 286 (水系域コード一覧) 参照

単位流域コード : 河川単位流域台帳ファイル (KS-271) 参照

始終点タグ : 1 - 図郭上の始終点

2 - ノード

3 - 図郭上のノード

0 - その他の中間点

\* 流域界・非集水域界区分が“ 1 ”のとき

項目		桁数	累積桁数	データ形式
2次メッシュ・コード		6	6	9(6)
計測会社番号		1	7	9
流域界・非集水域界区分		1	8	9
非集水域界線	空欄	18	26	X(18)
	始終点タグ	1	27	9
2次正規化		6	33	9V9(5)
位置		6	39	9V9(5)
スペース		1	40	X

表 3 - 2 - 3 : 流域界・非集水域界データのフォーマット 2

例 : 543321500000000008757100111100000023417

備考 : 流域界・非集水域区分の“ 1 ”は非集水域界線

- 始終点タグ : 1 - 図郭上の始点  
 2 - 図郭上の終点  
 3 - 閉ストリングの始点  
 4 - 閉ストリングの終点  
 0 - その他の中間点

ここでの2次メッシュコードとは、合計で6個の数字からなっているが、最初の4桁が、20万の1地勢図の第1次地域区画メッシュコードであり、続く2桁が、2万5千分の1地勢図の第2次地域区画メッシュコードである。今回使用したメッシュファイルのコード番号は、日本地図センターの“標準地域メッシュコード一覧図”の通りである。

### 3.3 データの加工

KS-273(流域界,非集水界線位置)のレコード順は2次メッシュで昇順となっているが、ノードごとに部分部分で連続的に入っていて、単位流域ごとにまとまっていない。このノードを繋げ、単位流域ごとの1枚のポリゴンに直し、利用可能なデータにした。

まず利用するにあたって、本研究では斐伊川水系の単位流域データを使用するので、斐伊川水系の単位流域データ(集水域)を抜き出した。それにあたって斐伊川水系の水系コードと島根県(一部)の1次メッシュコードを利用した。

例： 5332 17      50      87074      0166      87074      01672005053006153  
(メッシュコード) (右水系域コード)      (左水系域コード)

斐伊川水系の水系域コード : 87074

島根の1次メッシュコード: 高梁 5233、大社 5332、松江 5333、浜田 5232

そうして抜き出したデータを利用してポリゴンデータを作る方法を下に示す。

1. 始終点ノードを利用し、パーツ(部分的に繋がっている座標値)を判別する。  
始終点ノードが1もしくは2、で挟まれている部分がパーツである。

例：

	始終点ノード	
52324750870740250870750223	2	062510086853
52324750870740250870750223	0	062782087959
52324750870740250870750223	0	063801088691
52324750870740250870750223	0	066246090059
52324750870740250870750223	0	067981089674
52324750870740250870750223	0	072236091405
52324750870740250870750223	2	074905091965
//ここでデータが切れている		
52324750870740249870740250	1	067151100000
52324750870740249870740250	0	068263098568
52324750870740249870740250	0	069743098167
52324750870740249870740250	0	071948096758
52324750870740249870740250	0	074153095576
52324750870740249870740250	2	076736093641

2. メッシュコードを利用して、座標値を 2 次メッシュごとの相対座標から水系に合わせた絶対値座標になおし、配列に格納していく。

例： 5332 17 50 8707401668707401672005053006153  
(1 次 2 次 メッシュコード)

3. 2 に加えて、判別したパーツごとの左右の単位流域コードをチェックし、配列に入れていく。

例： 5332175087074 0166 87074 0167 2005053006153  
(右単位流域コード) (左単位流域コード)

3. 単位流域コードごとにパーツを配列から抜き出し、パーツの両端の座標値を調べ、その座標値が一定値以上に近い点の場合につなげる。

例：

. . .  
689006 683284  
690203 681967  
690018 681288  
//この上下を繋げる  
690421 681158  
694513 678941  
. . .

この 1 ~ 3 の作業を行い、流域界データをポリゴンデータに変換した。  
書き出したデータは以下のようなフォーマットになっている。

(X 座標値 Tub Y 座標値) 改行

. . .

(X 座標値 Tub Y 座標値) 改行

空行 (この空行でポリゴンデータを分けている)

(X 座標値 Tub Y 座標値) 改行

. . .

(以降 繰り返し)

## 第4章 JMC マップデータ (海岸線データ)

### 4.1 概要

JMC マップは、20 万分 1 相当のベクトル形式の地図データ (平面)。データ項目は、行政界・海岸線、道路、鉄道、河川・湖沼、市区町村名等の記号・注記。

データは、1 次メッシュ (約 80km 四方 : 20 万分 1 地勢図) 単位に 1 つのファイルにまとめられている。本研究では海岸線・湖沼の水涯線データを利用し、海岸線・湖沼の水涯線を含めた流域界のポリゴンデータとして利用できるように加工した。

### 4.2 フォーマット

- ・ 1 次メッシュ単位にファイルになっている。
- ・ ファイル内のデータは、2 次メッシュ番号順に並んでいる。
- ・ 2 次メッシュ内のデータは、メッシュ・ヘッダー・レコードに続き、必要に応じてノーレコード、ラインレコード、エリアレコード、ポイントレコード、注記テキストレコードが続きます。なお、ラインレコードには座標値レコード、エリアレコードにはのエリア構成レコードが続く。
- ・ 2 次メッシュ内の各座標値は、左下を(0,0)、右上を(10000,10000)とする正規化座標で記録されています。なお、x 座標は右 (東) 方向、y 座標は上 (北) 方向となる。
- ・ 各レコードは、復帰・改行コードで区切られている。
- ・ 文字コードは、シフト JIS コードを使用している。

データファイルは 1 次メッシュ単位になっており、以下のような構造になっている。

・データファイル（1次メッシュ単位）

メッシュヘッダーレコード（2次メッシュ単位）
レイヤーヘッダーレコード
ノードレコード
ラインレコード
座標値レコード
エリアレコード
エリア構成ライン番号レコード
レイヤーヘッダーレコード（道路）
ラインレコード
座標地レコード
レイヤーヘッダーレコード（鉄道）
ラインレコード
座標地レコード
レイヤーヘッダーレコード（河川・湖沼）
ラインレコード
座標地レコード
レイヤーヘッダーレコード（記号・注記）
ポイントコード
座標地レコード
注記テキストレコード
レイヤーヘッダーレコード（鉄道）
ラインレコード
座標地レコード
メッシュヘッダーレコード（2次メッシュ単位）
・
・

表4-2-1：JMC マップデータの全体のフォーマット

ファイルフォーマットに関する説明は大量にあるが、本研究で使用した海岸線・湖沼の水涯線に関するデータフォーマットについてのみを記述する。

・メッシュ・ヘッダー・レコード

項目	開始	終了	形式	内容
レコードタイプ	1	2	A 2	“ M ” を記録
2次メッシュコード	3	8	1 6	2次メッシュコード
図名	9	2 8	N 1 0	当該2万5千分1地形図
レイヤー総数	2 9	3 1	1 3	当2次メッシュに含まれる レイヤー総数
・ 中略 ・	・ ・ ・	・ ・ ・	・ ・ ・	・ ・ ・
ライン総数	3 7	4 1	1 5	当2次メッシュに含まれる ライン総数
・ 中略 ・	・ ・ ・	・ ・ ・	・ ・ ・	・ ・ ・
空白	5 7	7 2	1 6 X	

表4-2-2：メッシュヘッダーレコード

・レイヤー・ヘッダー・レコード

項目	開始	終了	形式	内容
レコードタイプ	1	2	A 2	"H1":構造化が行われていない レイヤー "H2":構造化が行われている レイヤー
レイヤーコード	3	4	I 2	レイヤーの内容を表す 1 : 行政界・海岸線 2 : 道路 3 : 鉄道 5 : 河川・湖沼 7 : 記号・注記
ノード総数	5	9	I 5	当レイヤーに含まれるノード総数
ライン総数	1 0	1 4	I 5	当レイヤーに含まれるライン総数
エリア総数	1 5	1 9	I 5	当レイヤーに含まれるエリア総数
ポイント総数	2 0	2 4	I 5	当レイヤーに含まれる ポイント総数
レコード総数	2 5	2 9	I 5	当ヘッダーレコードを除いた 当レイヤー内のレコード総数
空白	3 0	3 0	1 X	
最初の作成年月	3 1	3 4	I 4	当レイヤーの最初の作成年月 西暦年の下 2 桁と月 2 桁
空白	3 5	3 5	1 X	
最終の更新年月	3 6	3 9	I 4	当レイヤーのデータ更新年月 西暦年の下 2 桁と月 2 桁
空白	4 0	7 2	3 3	X

表 4 - 2 - 3 : レイヤー・ヘッダー・レコード

・ラインレコード

項目	開始	終了	内容
レコードタイプ	1	2	“ L ” を記録
レイヤーコード	3	4	1 : 行政海岸線 2 : 道路 3 : 鉄道 4 : 河川湖沼
データ項目コード	5	6	表 4 - 2 - 4 ライン項目コード表参照
ライン一連番号	7	1 1	レイヤー内でラインが何番目に位置するかを示す一連番号
ライン種別コード	1 2	1 7	0 : 地上 1 : 地下・トンネル
始点ノード番号	1 8	2 2	
始点接続情報	2 3	2 3	0 : 図葉内ノード 1 : 隣接図葉に接続する 図郭線上のノード 2 : 隣接図葉に接続しない 図郭線上のノード
終点ノード番号	2 4	2 8	
終点接続情報	2 9	2 9	始点接続情報と同じコード
左側行政コード	3 0	3 4	ラインの向きに対しての 左側の行政コード
右側行政コード	3 5	3 9	ラインの向きに関しての 右側の行政コード
座標点の数	4 0	4 5	当ラインを構成する XY 座標点の数
空白	4 6	7 2	

表 4 - 2 - 3 : ラインレコード

・ライン項目コード

レイヤー	コード	データ項目
行政界・海岸線 1	5	海岸線
河川・湖沼 5	2	湖沼の水涯線

表 4 - 2 - 4 : ライン項目コード

上記のヘッダーレコードに続く数字が実際の海岸線・湖沼の水涯線情報 X , Y 座標点と思われる。

例：海岸線

```
L 1 5 3 0 50 503252499999 14 0 0
9175 8723 9173 8716 9189 8689 9208 8692 9270 8694 9289 8700 9302 8721
9301 8751 9271 8764 9255 8783 9229 8767 9210 8762 9192 8749 9175 8723
```

例：湖沼の水涯線

```
L 5 2 4 0 00 00 13 0 0
5488 0 5550 380 5600 480 5890 210 6620 250 6920 530 7230 450
8170 500 8720 920 8910 1410 9100 1480 9300 133010000 1215
```

上記のように次の“ L ”（又はM、等のヘッダーレコードの先頭の値が）が出るまでの間の数値が海岸線・湖沼の水涯線を形成する座標の集まりだと思われる。この座標値は下記のように 2 次メッシュの左下を原点とした相対座標である。

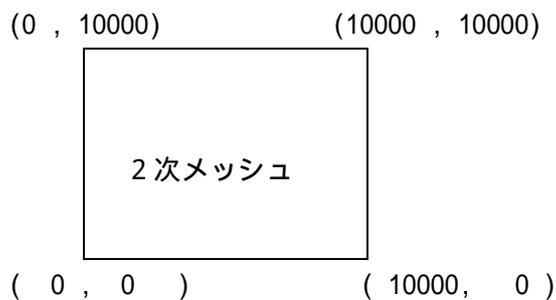


図 4 - 2 - 5 : 海岸線・湖沼の水涯線データの座標値

### 4.3 データの加工

目的のデータは

M (目的の2次メッシュ番号) で開始される2次メッシュファイル

H1	1	もしくは
H2	1	で始まる レイヤーヘッダーレコード
L	5	で開始するライン・座標値レコードのデータ
か		
H1	5	もしくは
H2	5	で始まる レイヤーヘッダーレコード
L	2	で開始するライン・座標値レコードのデータ

上記を用いて必要なデータを検索し、メッシュコードを用い流域界データと同じ絶対値座標に変換して、ファイルに書き出した。このとき海岸線・湖沼の水涯線データだけを取り出せばよい。

なお流域界ポリゴンデータとの結合はプログラムでは非常に困難であったので、今回は手作業で繋げていった。

書き出したデータのフォーマットは流域界ポリゴンデータと同じである。  
この補完されたポリゴンデータを2次元河川流域境界データとする。

## 第5章 数値地図 50m メッシュ (標高) データ

### 5.1 概要

国土地理院が刊行している 2 万 5 千分 1 地形図に描かれている等高線から求めた数値標高モデル (DEM: Digital Elevation Model) データ。2 次メッシュを経度方向および緯度方向に 200 等分して得られる各区画 (1/20 細分メッシュ、2 万 5 千分 1 地形図上で約 2 mm) の中心点の標高値が記録されている。標高点間隔は緯度 (南北) 方向で 1.5 秒、経度 (東西) 方向で 2.25 秒となり、実距離では約 50m となる。

### 5.2 フォーマット

- ・ファイルフォーマット 2 次メッシュ単位に 1 つのファイルになっている。

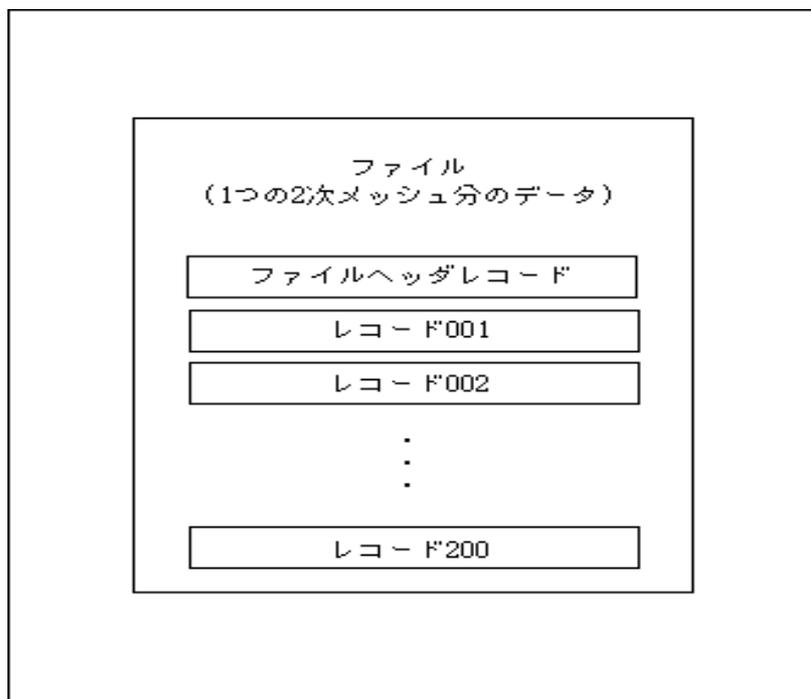


図 5 - 2 - 1 ファイルフォーマット

・ファイルヘッダーレコード

項目	開始	終了	仕様(*1)	内容
2次メッシュコード	1	6	16	標準2次メッシュコード
原図測量年紀	12	15	14	西暦年号
原図修正年紀	16	19	14	西暦年号
数値化年紀	20	23	14	西暦年号
東西方向の点数	24	26	13	東西方向のデータ点数
南北方向の点数	27	29	13	南北方向のデータ点数
2次メッシュ左下の緯度	30	36	17	度3桁、分2桁、秒2桁
2次メッシュ左下の経度	37	43	17	度3桁、分2桁、秒2桁
2次メッシュ右上の緯度	44	50	17	度3桁、分2桁、秒2桁
2次メッシュ右上の経度	51	57	17	度3桁、分2桁、秒2桁
図葉面数	58	58	11	当該メッシュにかかる図葉の面数(*2)
1番目の図名	59	78	N10	
位置フラグ	79	79	11	1番目の図葉の位置を示すフラグ(*3)
2番目の図名	80	99	N10	
位置フラグ	100	100	11	2番目の図葉の位置を示すフラグ
3番目の図名	101	120	N10	
位置フラグ	121	121	11	3番目の図葉の位置を示すフラグ
4番目の図名	122	141	N10	

原図の縮尺	7	11	15	縮尺の分母の数値を記録
-------	---	----	----	-------------

位置フラグ	142	142	11	4番目の図葉の位置を示すフラグ
記録レコード数	143	145	13	記録されているレコードの数
コメント	146	225	N40	利用の際に参考となる情報を記録
レコード1のフラグ	226	226	11	(*4)
.	.	.	.	レコード2～レコード199のフラグ
レコード200のフラグ	425	425	11	
余白	426	1009	584X	
復帰・改行	1010	1011		(*5)

図 5-2-2 ファイルフォーマット

・2次メッシュ内のレコード構成

メッシュコード	レコード番号	標高値
??????	001	1 2 3 4 … 199 200
??????	002	1 2 3 4 … 199 200
??????	003	1 2 3 4 … 199 200
.	.	.
.	.	.
.	.	.
??????	199	1 2 3 4 … 199 200
??????	200	1 2 3 4 … 199 200

図 5-2-3 2次メッシュ内のレコード構成

(各標高値は、メッシュの中心の標高値となる)

レコードは、北端から南端への順序で並んでいる。

各レコードには、2次メッシュコード、レコード番号、200個の標高値が記録されている。但し、レコードに含まれる全ての区画が海の場合にはレコード数は変化する事がある。

ファイルヘッダレコードの記録レコード数欄に記録された個数となる。各レコードは、復帰・改行コードで区切られている。文字コードは、シフト JIS を使用している。

項 目	開始	終了	仕様	内 容
メッシュコード	1	6	16	標準 2 次メッシュコード
レコード番号	7	9	13	北 南の順
標高値 1	10	14	15	西 東の順。
標高値 2	15	19	15	
標高値 3	20	24	15	
.	.	.	.	
.	.	.	.	
.	.	.	.	
標高値 200	1005	1009	15	
復帰・改行	1010	1011		CR・LFコード

図 5 - 2 - 4 レコードフォーマット

"開始"と"終了"は、レコードの各項目の開始と終了の位置。先頭からのバイト数で示している。つまり 1bit 目から 6bit 目までの 6 桁が、そのデータの 2 次メッシュコード。標高値は、0.1m 単位で表現されている。(100.0m は 01000 と表現されている) ただし 0.1m の位はすべて 0 となっている。海部には、-9999 (-999.9m) が入っている。

### 5.3 データの加工

50mメッシュ標高データに含まれているのは、200\*200の標高値のデータである。ファイルヘッダーレコード部分については、データの欠損など特別な理由がない場合は、本研究に特に必要はない。データファイルの編集の際に不必要なデータを全て除いている。実際必要なデータはデータ自体のメッシュ番号 標高データとその位置である。

今回、1次メッシュ4つ分((200\*8\*2)\*(200\*8\*2))のデータを抜き出し 3200\*3200のデータにまとめ、利用した。なお海部は"-9999"でなく"0"と置き換えた。

2次元流域境界データに標高データを補完するにあたって、流域境界の座標値にもっとも近い点の標高値を3次元流域境界データの標高値として補完した。

その方法を以下に示す。

1. 標高データは  $3200 * 3200$  の点データで、2次元流域境界データは XY 座標がそれぞれ  $0 \sim 1600000$  の値となっているので、標高データの点と点の間隔は 500 である。  
なので、2次元流域境界データの X 座標 Y 座標を抜き出し、それぞれを 500 で割り、余りが 250 以上かどうかを考慮し、どの点が一番近いか計算する。

$Y/500 = A$  行目    if (  $Y\%500 > 250$  ) A+1 行目

$X/500 = B$  列目    if (  $X\%500 > 250$  ) B+1 列目

2. A 行目の B 列目のところのデータを読み取り書き出す。

## 第6章 Java サブレット

### 6.1 Java サブレットとは

Java サブレットとは、サーバサイドで動く Java プログラムである。  
同じ Java プログラムで Java アプレットがあるが、これはクライアントサイドで実行する Java プログラムである。

Java アプレットの場合は Java プログラム自体と必要なデータをクライアントにダウンロードする必要がある。しかし、サブレットの場合は、Java プログラム自体をサーバ上で実行し、その結果を(多くの場合)HTMLとしてクライアントに送信するだけである。そのため、アプレットと比較した場合、次のような長所がある。

- ・重たいプログラム・データをクライアントにダウンロードする必要がない。
- ・サーバ側にサブレットを実行する環境を整備すればよいので、クライアントに負荷がかからない(最低限 Web ブラウザが動作すればよい)。

## 6.2 なぜ Java サーブレットなのか

サーブレットと似た技術に CGI がある。

CGI に対するサーブレットの利点を次に示す。

- ・ 効率的

CGI の場合、HTTP リクエスト毎に新たなプロセスが生成される。

それに対し、サーブレットの場合、Java 仮想マシンが唯一のプロセスとして常に動作しており、個々のリクエストはスレッドによって処理される。そのため、プロセス起動のためのオーバヘッドが生じない。また、処理終了後もメモリ上に存在するため、複数のリクエストから簡単に利用できる。

- ・ 開発しやすさ

サーブレットは HTML フォームデータの構文解析、HTTP ヘッダの読み取りと設定、クッキーの処理、セッションの管理、といった多くの基礎的機能を豊富に備えている。また、信頼性も高く再利用性にも富む。

- ・ 強力

複数サーブレットでのデータベース接続の共有など、資源を共有する最適化を容易に実装することができる。また、サーブレットは複数のリクエストにまたがって情報を保持するため、セッション管理や、前の処理結果のキャッシュなどの技術も簡単に実現できる。

- ・ 可搬性

サーブレットは Java の標準 API を使用するため、一般的にプラットフォーム非依存である。

- ・ 安全

CGI は通常 OS のシェルから実行するため、特殊文字 ( ¥n など ) に対して細心の注意が必要である。また、配列に対するバッファオーバーフローの攻撃対象になりうる。これに対し、サーブレットはこれらの問題点がない。

本研究ではサーブレット実行環境（サーブレットコンテナ）にフリーソフトウェアである Jakarta-Tomcat 5.5 を使用している。Tomcat は単体でも Web サーバの機能を備えている。また、IIS や Apach にサーブレットコンテナ単体の機能をリンクさせることも可能である。

## 第 6 章 X 3 D

### 6.1 X3D とは

X3D は eXtensible 3D の略であり、VRML97 の後継として設計された 3D グラフィックスのファイル形式の規格である。

VRML97 はその名の通り 1997 年に制定され、WWW で 3 次元のコンテンツを表現する手段として登場したが、当時のコンピュータのハードウェア性能の不足による画質の悪さや、当時のネットワーク性能の悪さによるソフトウェア（プラグイン）のダウンロード時間の長さにより、さほど普及しなかったと言える。

しかし現在、ADSL や光ケーブルなどのブロードバンドが普及し、X3D /VRML のプラグインはそれほど大きなソフトウェアと思われなくなった。またデータの大きさもさほど気にする必要もなくなり、物体を高精細にモデル化できるようになった。最近のハードウェアではちょっとした 3D グラフィックスの描画には充分すぎるほどの性能を持っている。

X3D と VRML97 の主な違いは、次の 6 点である。

- ・ XML 統合 : 3 次元コンテンツを Web サービスや分散ネットワークと統合可能。  
また、異種プラットフォーム間やアプリケーション間でのファイル / データ転送が可能。
- ・ 部品化 : 核となる軽いブラウザを構成できるので、配布が容易。
- ・ 拡張性 : 部品を追加して機能を拡張できるので、ニーズに応じたアプリケーションやサービスを提供可能。
- ・ プロフィール : 特定の応用分野ごとに決められた仕様のサブセット（標準拡張セット）を提供。

- ・ 進化性 : 既存の VRML97 コンテンツを X3D に変換可能。
- ・ 同報 / 組込対応 : 携帯電話からスーパーコンピュータまでが利用可能。

X3D は 3 次元ファイルフォーマットの国際標準であり、その仕様は完全にオープンになっている。また、X3D は豊富な機能をもっているため、工学・科学・医学分野での可視化、CAD ( 建築、土木、機械など)、訓練、シミュレーション、マルチメディア、娯楽、教育など、幅広い応用が考えられる。

X3D では 3 次元データは XML 形式で記述される。これは X3D の最大の特徴であると言える。これにより全てのアプリケーション間で実時間通信が可能である。その文法は以下のようなになる。

例: 円筒形の定義

```
<Shape>
<Appearance>
<Material diffuseColor="0.5 0.5 0" />
</Appearance>
<Cylinder radius="2" />
</Shape>
```

## 6.2 なぜ X3D なのか

X3D のほかに Web3D でよく利用されている Java 3D がある。  
Java3D に対する X3D の利点を次に示す。

- ・ 拡張性・部品化  
X3D は extensible というその名の通り、容易に拡張ができるように設計されている。逆にこれによりコアの部分は小さくでき、必要に応じて閲覧ソフトを軽量に実装することができる。また、そのおかげで今後携帯電話などモバイル機器での利用が広まる可能性がある。
- ・ XML 形式  
グラフィックデータを XML 形式にすることでデータの再利用性が高まる。  
また XML は Java などプログラム言語で操作するときに、操作用のライブラリがあり、XML 形式で書かれている X3D もそれを利用できる。

## 第7章 ブラウザ環境でのグラフィック表示

3次元河川流域境界データを Java サブレットと X3D を用いてグラフィック化する。  
まずサーバマシンのローカルに2次元河川流域環境データを置く。  
次にクライアントから要求が来たら、Java サブレットを用いてそのデータを読み取り、  
X3D の “ IndexedLineSet ” を使用し線のデータを書き出す。

例 : <Shape>

```
<IndexedLineSet coordIndex='0 1 2 3 4 5'> //点ごとの ID を割り振る
<Coordinate point='
364761 7210 550000 // X座標 Y座標 Z座標
364864 7130 550191
. . .
364761 7210 550000
'/>
</IndexedLineSet>
</Shape>
```

このようなポリゴンデータを繰り返すと、以下のような流域境界が表示される。



図 7 - 1 : 斐伊川水系流域境界 (上から見た場合)

地形（数値標高データ 3200 \* 3200）をブラウザに表示するにあたって、マシンの性能により 3200 \* 3200（約 55MB）から 640 \* 640（約 1 MB）までデータを落とした。

X3D の “ ElevationGrid ” を使用し、表示させると以下のような画像が得られる。

例：<Shape>

```
<ElevationGrid      creaseAngle='0.5'   xDimension='640'   xSpacing='2500'  
                    zDimension='640' zSpacing='2500'  
height='0 0 . . . / 標高地データ 640 * 640 /  
'/>  
</Shape>
```

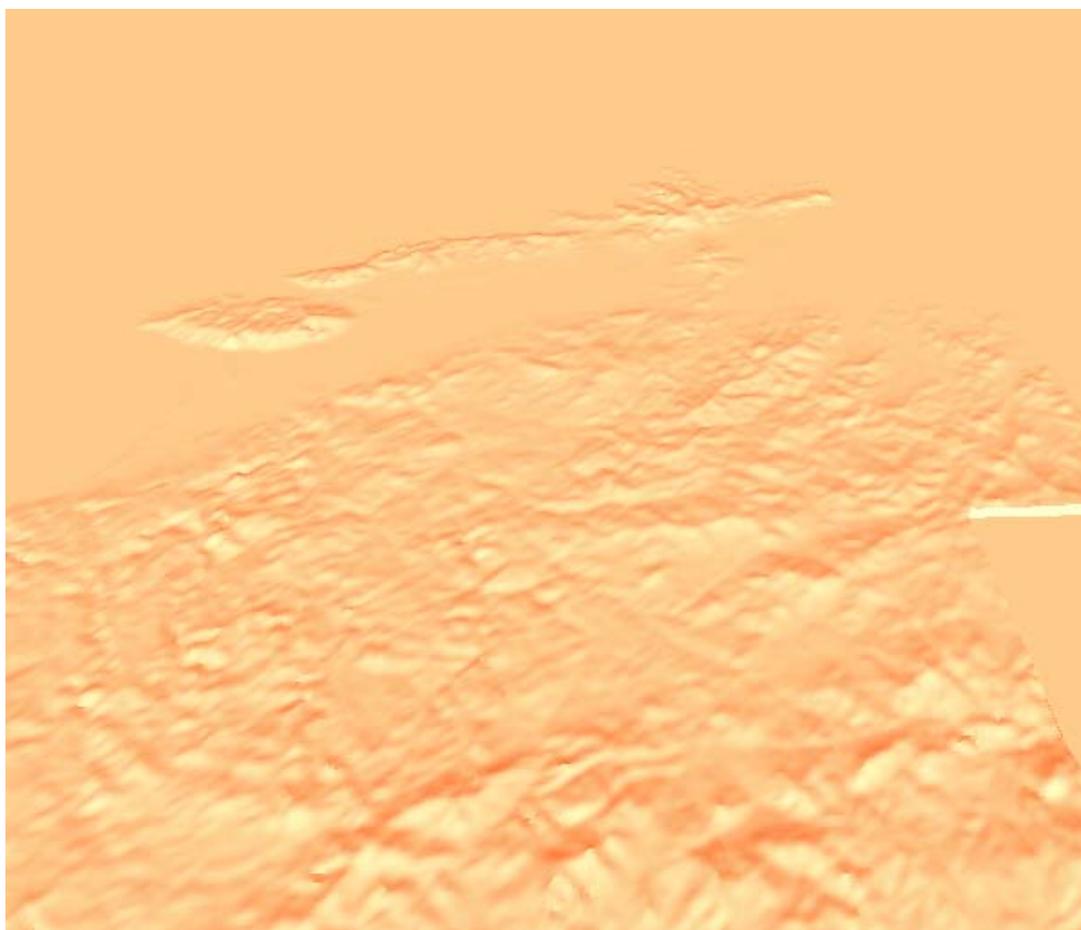


図 7 - 2 : 島根の地形の 3 D 表示

この地形データに 3 次元流域境界データを重ねて表示させた画像を以下に示す。

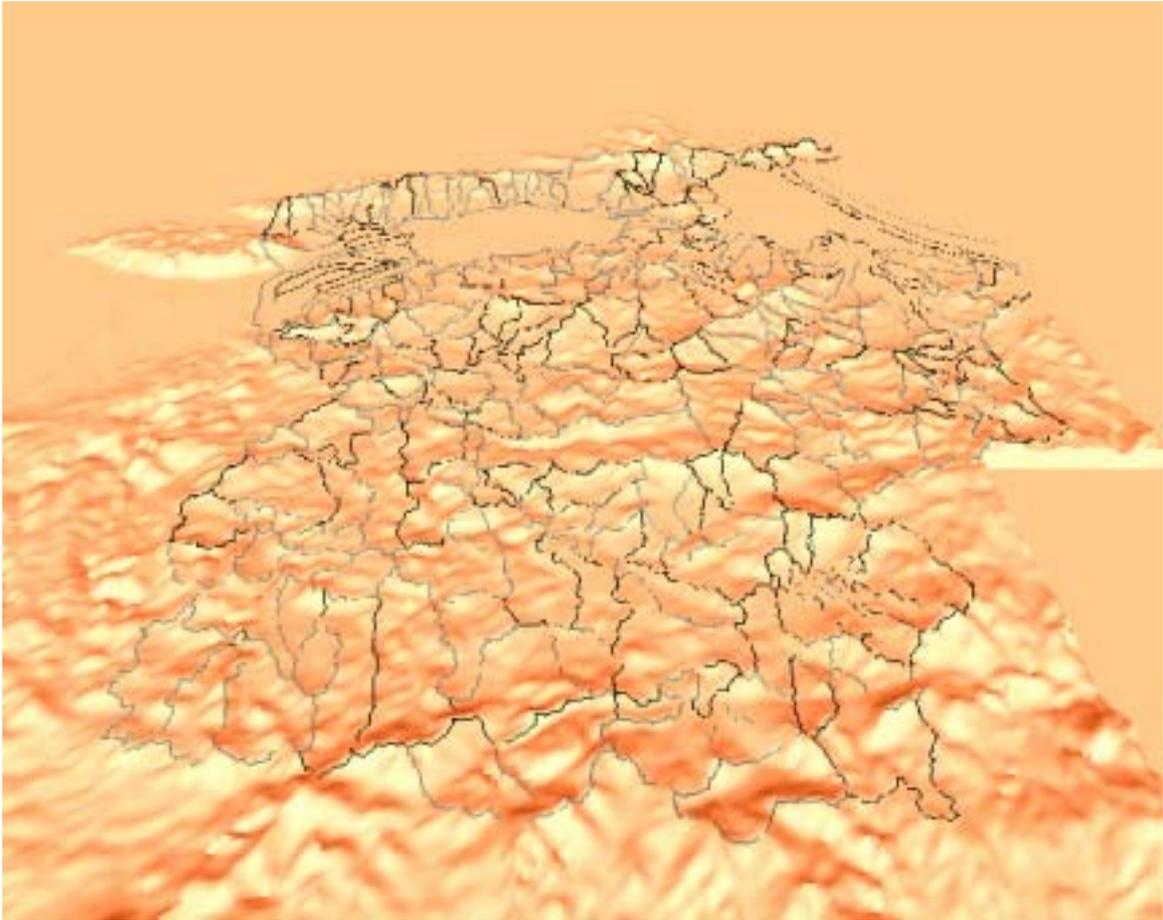


図 7 - 3 : 3 次元流域境界と地形の重ね合わせ 1

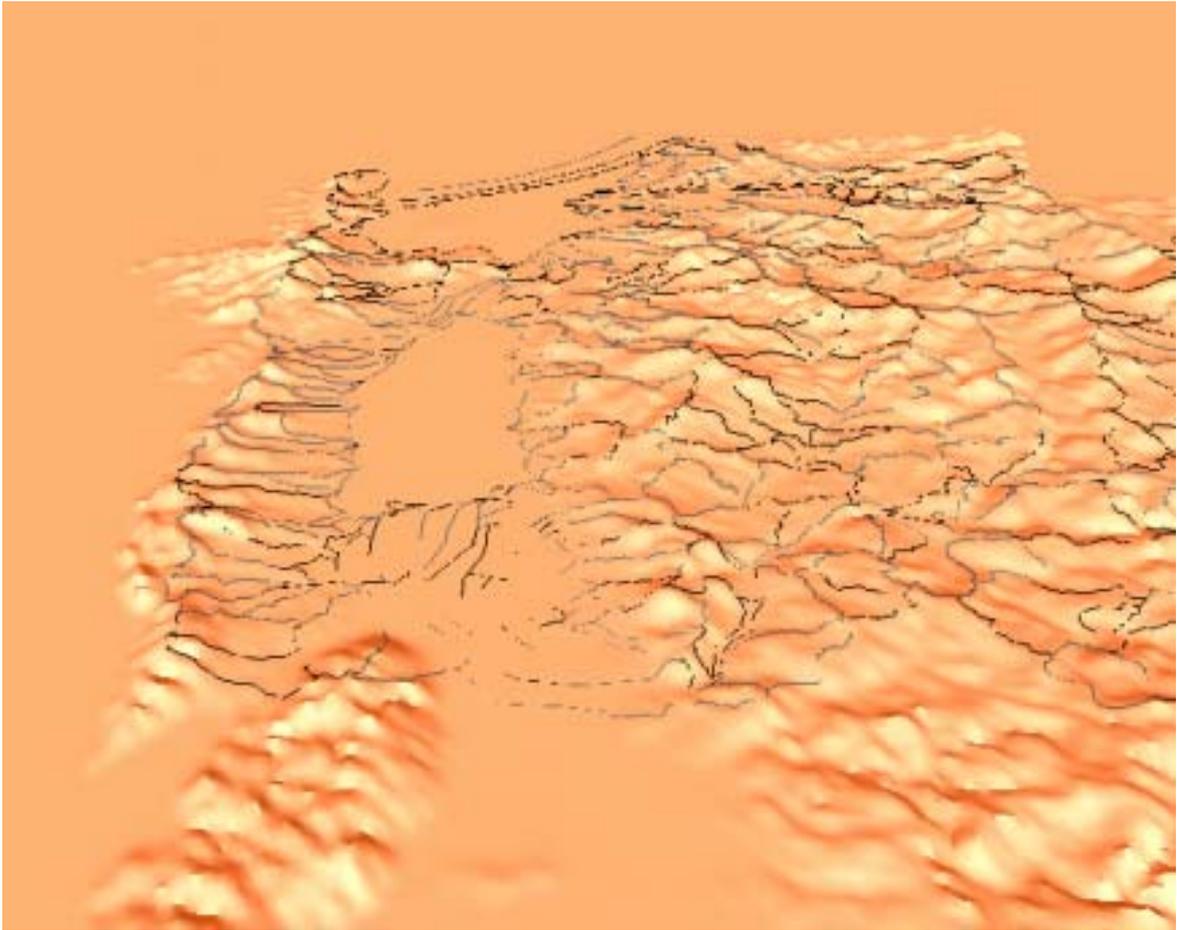


図 7 - 4 : 3 次元流域境界と地形の重ね合わせ 2

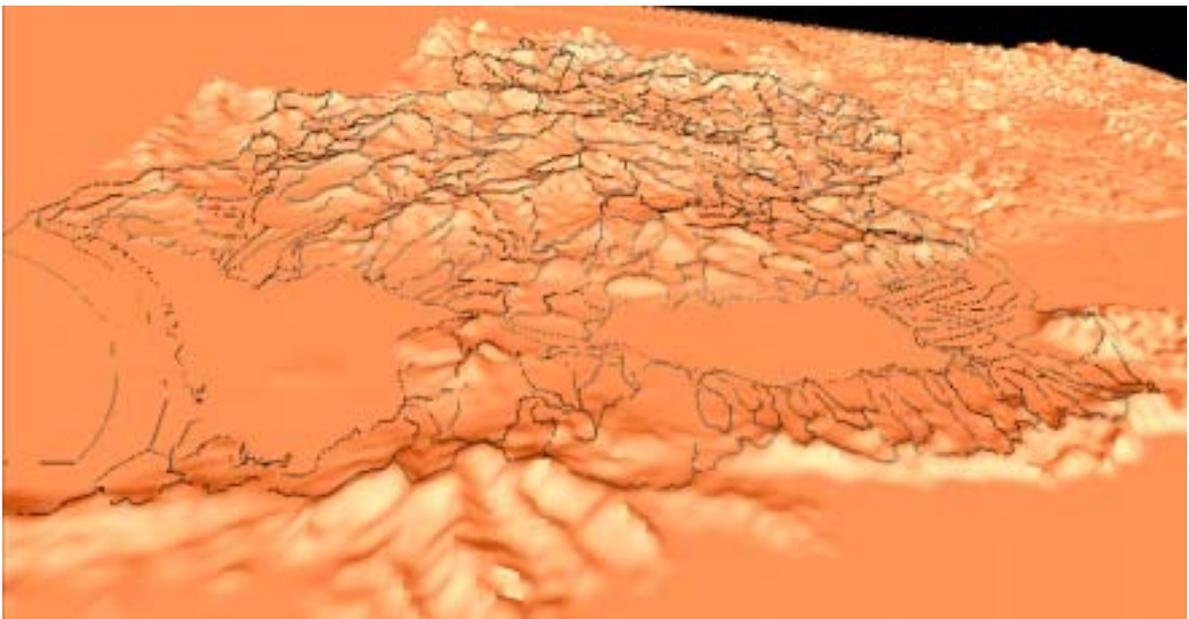


図 7 - 5 : 3 次元流域境界と地形の重ね合わせ 3

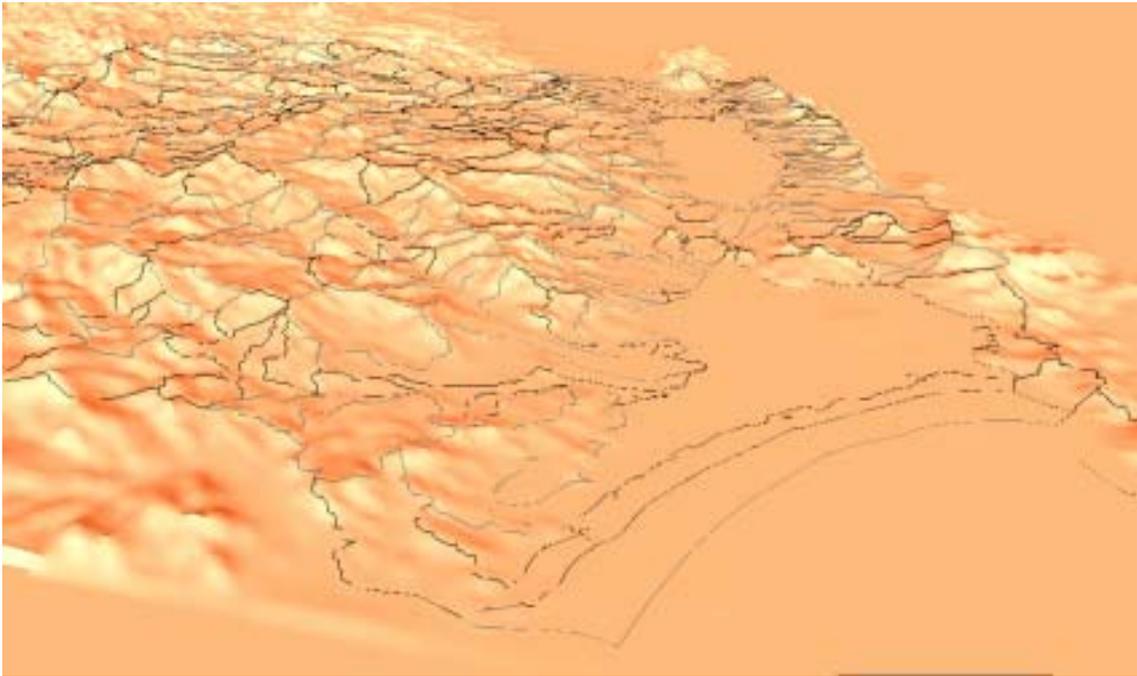


図 7 - 6 : 3 次元流域境界と地形の重ね合わせ 4

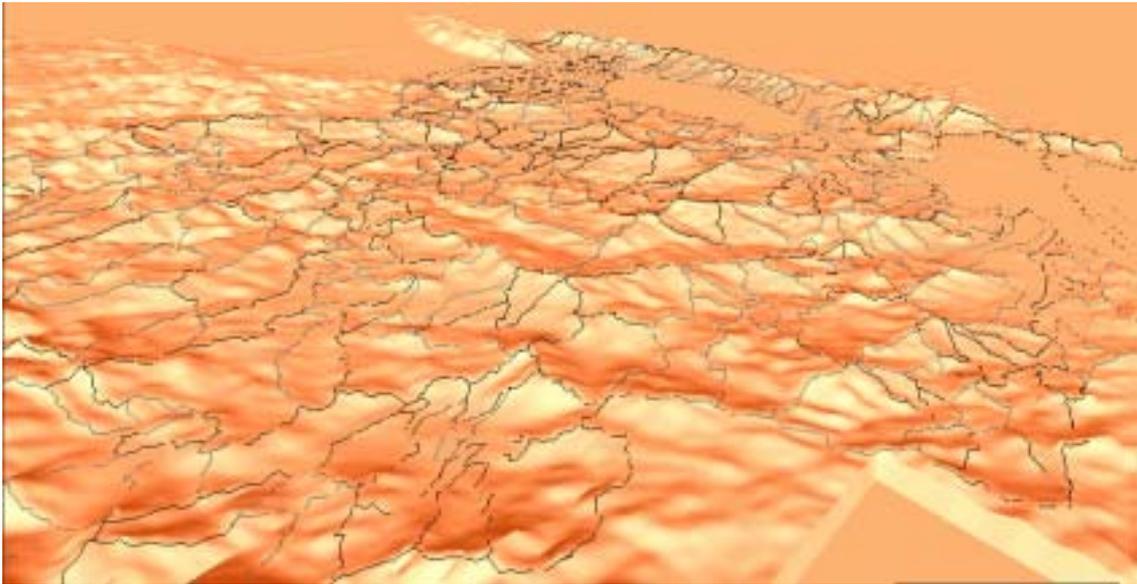


図 7 - 7 : 3 次元流域境界と地形の重ね合わせ 5

## 第8章 終論

終論として本研究の今後の課題や展望等を述べる

- ・グラフィック表示の部分化

本研究では流域全体の表示を行っているが、これではデータが大きくなり利用しにくい面がある。なので必要な部分だけ切り取り、データを返すことができるシステムにすることでより軽快に表示、動作させることが出来る。

- ・DBの利用

本研究ではサーバマシンのローカルに置いた2次元流域境界データを読み取りブラウザに返していたが、DBを利用するようにしたい。

そうすることで多くのデータに対応できるようになり、データの随時更新にも対応できるようになり、より機能が高まる。

- ・検索システム

人口、道路、鉄道、その他のデータを加え、流域境界を基準とした検索システムを構築し、様々な状況を設定して検索できるようにしたい。

例えば、

- ・斐伊川の流域左右5 kmにおける総人口
- ・台風時の避難・救助活動の可否
- ・河川の汚染の被害拡大予想

といったものが出来る。

## 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先生には、心より御礼申し上げます。また、田中研究室の伊達さんには、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，本研究で作成したプログラム及びデータ，並びに関連する発表資料等のすべての知的財産権を，本研究の指導教官である田中教授に譲渡致します。

## 引用文献

数値地図ユーザズガイド 監修 建設省国土地理院 (平成 10 年 1 月 1 日)

数値地図 5 0 mメッシュ (標高) 日本- 国土地理院 CD-ROM Index.htm

JMC マップ (日本)(財) 日本地図センター CD-ROM Readme.txt

島根大学総合理工学部 数理・情報システム学科 卒業論文

- ・国土空間データのグラフィック的構造化 金納陽晴 2000 年
- ・Java 3 D を用いた 3 次元表示空間情報システム 埜田千帆 2002 年