

衛星データ探索フレームワーク GGS II の提案と実装

長谷部雅彦[†], 山田大地[†], 中野雄介[†]
和久田龍[†], 菌部知大[†], 手塚宏史[†], 稲葉真理[†]

本論文では、現在開発中の地球規模の探索フレームワーク GEO Grid Spider II (GGS II) について述べる。GGS II によって、Web インタフェースを介し、衛星データを使った全世界探索アプリケーションを簡単に開発することができる。GGS II は、(1)データの取得と変換、(2)登録されたコンポーネントプロセスの組み合わせによる探索アプリケーションの構築、(3)キャッシュメカニズム (DPM cache) などの機能を持っている。実際に GGS II 上で複数のアプリケーションを作成した結果、簡単に世界探索を行うことができた。

GEO Grid Spider II: a global search framework using satellite data

MASAHIKO HASEBE[†] DAICHI YAMADA[†] YUSUKE NAKANO[†]
RYO WAKUTA[†] TOMOHIRO SONOBE[†]
HIROSHI TEZUKA[†] MARY INABA[†]

In this paper, we introduce our global search framework, GEO Grid Spider II (GGSII). Using GGSII, we can easily implement worldwide search applications using satellite data via web interface. GGSII has features (i) downloading and converting satellite data, (ii) constructing search applications by combining registered components process, (iii) a cache mechanism called DPM cache. We implemented several applications on GGSII and show their results.

1. はじめに

我々は、Webブラウザを介して衛星による地球観測データを用いた地球規模の探索を行うためのフレームワーク GEO Grid Spider II (GGS II) を開発している。GGS II はユーザが手軽かつ短時間で効率的に探索を行えることを目的としている。そのためにGGS II は、地形データの取得、Webインタフェースによるアプリケーションの作成・実行、ユーザによるプログラムの登録という機能を提供する。GGS II で探索を行うためアプリケーションは、複数のコンポーネントプロセスで構成されており、ユーザはあらかじめGGS II に登録してあるコンポーネントを利用してよいし、独自に開発したコンポーネントを登録して利用することもできる。

近年、地球観測衛星による衛星探索データの取得が容易になっている。無料で公開されているものも多くあり、GGS II はこれらのデータを用いることで手軽に探索を行えることを目指した。衛星探索データには、標高、可視光、温度等の種類があり、これらはデータ形式も異なっており、保存されているサイトも様々である。GGS II はユーザが必要としているデータを、あらかじめ登録された各衛星データのデータベースから

取得し統一的に扱うことを可能にすることで、ユーザは、データを対象の場所や目的にあわせて収集する、ユーザの環境に適したデータに変換するといった煩雑な手間を省くことができる。

ユーザは Web ブラウザで GGS II にアクセスすることで、手軽に探索を行うことができる。グラフィカルユーザインタフェースを介し、可視化されたコンポーネントプロセスを組み合わせることで目的にあったアプリケーションを作成することができる。これに加え指定された条件を入力し、アプリケーションを実行することによって、簡単に探索を行うことができる。

GGS II が用意したコンポーネントプロセスを組み合わせるだけでは、探索の方法が限定されてしまう。そこで GGS II はユーザが作成したプログラムを登録できる機能も提供し、ユーザは独自に用意したプログラムを登録することで任意の処理をアプリケーションに組み込むことができる。またこの登録されたプログラムは、GGS II のコンポーネントプロセスとして、他のユーザも再利用することができる。

ウェブアプリケーションでは短時間でユーザに結果を返すことが望ましい。しかし世界中を探索するには膨大な時間がかかる。探索処理は前処理が共通なことも多いことと、パラメータを変えて試行錯誤が行われるという特徴に着目し、GGS II ではコンポーネントプロセスの中間結果をキャッシュすることで、処理時間

[†] 東京大学大学院理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo

の短縮を行っている。これによって同一のプロセスを行う際には高速な処理が可能となる。

本研究では以上のような特徴を備えた GGS II を提案し、設計・実装を行った。また、その上に、サンプルアプリケーションとして富士山と類似する山の探索アプリケーション、直線検出アプリケーション、顔検出アプリケーション、円検出アプリケーションを実装し、実際の探索を行うことで動作確認をした。

本論文の構成は以下の通りである。2 章で GGS II の目的と使用方法を詳述する。3 章で GGS II の設計と実装について解説する。4 章では GGS II の評価について述べ、5 章では関連研究について触れる。最後にまとめを 6 章で、今後の課題を 7 章で述べる。

2. GEO Grid Spider II

この章では GGS II の目的と使用方法を述べる。

2.1 目的

近年では衛星探索データなどの計算機で処理可能なデータが容易に取得できるようになり、それらのデータを用いることで地球規模の探索が可能になった。しかし衛星探索データには、標高・可視光・温度等といった多数の種類があり、収集には画像データをダウンロードする、そのデータを環境に適したものに变换する、探索に必要な範囲内のデータを全て集める、といった手順を踏まなければならない。そこで GGS II は、このような公開されている衛星探索データに対し、上記の一連の作業を自動で行い、統一的に扱う手段をユーザに提供することによって、ユーザが手軽に探索を行えることを可能にする。

また地理情報に詳しくないユーザが探索を行う場合、簡単にアプリケーションを使用できることが重要である。既存のアプリケーションでは、機能が多く搭載されているが故に操作も習熟を要する。GGS II では Web ブラウザでアクセスでき、目的にあったアプリケーションを容易に構成することができるユーザインタフェースを提供する。ユーザはあらかじめ GGS II で用意されたコンポーネントプロセスを組み合わせることで目的の探索を実行できる。

プログラミングを行えるユーザであれば、独自に用意したプログラムを GGS II に登録することで、探索の手法を広げることができる。この登録されたプログラムは GGS II 上のコンポーネントプロセスとして共有されるので、他のユーザも再利用が可能である。

我々は、ユーザが手軽かつ短時間で効率的に探索を行うためにこのような機能を搭載した GGS II を提案する。

2.2 使用方法

ここでは Web ブラウザ上で動作するユーザインタフェースの使用例を説明する。またプログラムの登録について触れる。

2.2.1 アプリケーションの作成・実行

Web ブラウザで開いた画面上には座標指定エリアとアプリケーション作成エリアが用意されている。

(1) 座標指定エリア (図 1) では、地図から座標入力を用いることで探索範囲を設定する。地図を用いる場合、ピンをドラッグアンドドロップすることで範囲を囲み、枠内が指定範囲エリアとなる。またここでは探索に用いるデータの種類の選択することができる。

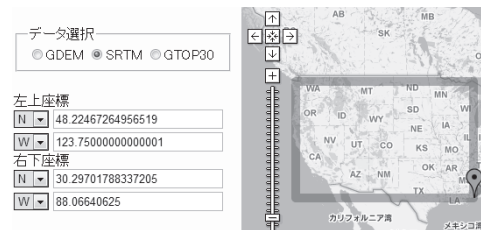


図 1 座標指定エリア

フレームワーク内では、ユーザが指定した探索範囲のデータを取得する。そのとき、探索範囲がデータファイル複数枚を跨いでいれば、そのデータ複数枚を繋ぎあわせた後ユーザの探索範囲にマッチする形でデータの切り出しを行う。また、データファイルが大きすぎるのであれば、適切な小領域に切り出しを行い、小領域毎に探索を行う。これらの処理はフレームワークが自動的に行っており、ユーザの視点では、指定した領域の一つの衛星データに対して作成したアプリケーションが実行されているように見える。

(2) アプリケーション作成エリアでは、目的にあったコンポーネントプロセスを選択し、組み合わせることでアプリケーションを作成する。ユーザは種類によって分けられたコンポーネントプロセス群の中から適切なものを選ぶ。コンポーネントプロセス同士の入出力のフォーマットが同じものであれば、接続することができる。接続されたコンポーネントプロセスは、一方のプロセスの出力がもう一方のプロセスの入力となっていて、連続した処理の流れを表している。コンポーネントプロセスにパラメータ入力が必要であれば、入出力用のボックスを接続してユーザはパラメータを入力する。これを繰り返すことでユーザは目的にあったアプリケーションを作成できる。操作は図 2 のような画面上で行う。

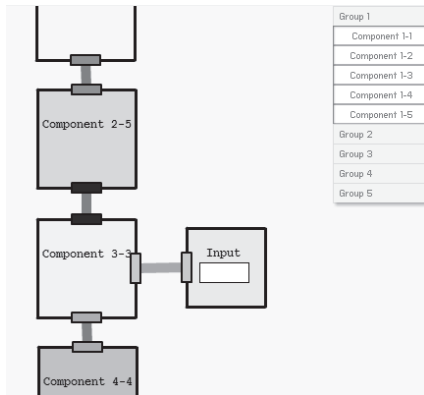


図 2 アプリケーション作成エリア

フレームワーク内で実行時に必要となるアプリケーションの情報は、アプリケーションを構成するコンポーネントプロセスの順序とそれぞれのコンポーネントが必要とする引数である。これを可視化しユーザにわかりやすい形でアプリケーションを構成してもらうのが、このアプリケーション作成エリアである。

(3) 実行ボタンを押すことで、サーバがアプリケーションの処理を行う。

(4) サーバの処理が終了するとウィンドウが現れ、処理結果が画像や座標情報として表示される。図 3 では処理結果を座標で受け取り、地図上に表示する例である。



図 3 結果表示の例

2.2.2 プログラムの登録

もし GGS II が提供しているコンポーネントプロセスの中に、ユーザが行いたい処理がみつからない場合は、ユーザは独自に作成したプログラムを登録することで、アプリケーションに組み込むことができる。プログラミング言語の制限はないが、入出力を指定されたフォーマットにする必要がある。フレームワーク管

理者が使用可能と判断すれば、そのプログラムは GGS II 上でコンポーネントプロセスと扱われる。

3. フレームワークの設計と実装

この章では、各項目の設計と実装について述べる。

3.1 衛星探索データの取得と整形

GSII では、ユーザが探索を行ないたい範囲と扱うデータの種類の指定することで、フレームワークは衛星データファイルを取得する。

衛星データファイルは、GSII が予め登録しておいた Web サイトからサーバ内へ収集される。その際に、データの種類の毎にファイルフォーマットが異なっていることがあるため、フォーマットを `GEOTiff[1]` に統一することでこれらの差異を吸収する。

また、衛星データファイルは一般に、全世界のデータを格子状に一定間隔で分割された状態で保管されているため、ユーザの指定した探索範囲が複数のファイルにまたがっていることがある。そのため、ユーザの要求にマッチした形になるように、ユーザの指定した探索範囲を満たすのに必要なファイルをリストアップしたあと、それらのファイルをつなぎあわせた後に必要な部分の切り出しを行う。これによって、ユーザが要求する探索範囲と一致した大きさの衛星データファイルを得ることができる。

これらのしくみによって、ユーザのデータファイル収集の手間を軽減し、衛星データファイルの形式などの詳細を知らなくても、衛星データに触れることができる。

3.2 ユーザインタフェース

ユーザが直接触れることになる Web ブラウザで表示されるユーザインタフェースは、範囲指定エリア・アプリケーション作成エリア・結果表示画面で構成されている。全体は JavaScript・HTML・CSS で実装を行う。範囲指定エリアでは、ユーザが指定した座標や用いるデータの種類の取得する。地図表示には Google Maps API[2]を用いている。アプリケーション作成エリアでは、FLASH を用いてユーザが作成したアプリケーションのコンポーネントプロセスの種類・接続関係・パラメータを取得する。実行ボタンが押されることで、以上のような入力がサーバへ送信され、処理が終了すると結果が返される。この結果は結果表示画面に画像や文字情報で表示される。

3.3 DPM cache

全世界を対象にして、衛星データ探索を行うには長い時間が必要である。また、探索条件や探索手法を少しずつ変えながら幾度も探索を行うことも考えられる

ため、一回の探索にかかる時間をなるべく短くしたい。そこで、プログラムの高速化手法であるメモ化の概念を拡張して、中間処理結果をキャッシュする Data-Process Memoization cache (DPM cache) [3]を提案し、GGSIの機能として搭載した。

メモ化とはプログラムの高速化手法の一つであり、同じ処理を行う際に記録しておいた以前の結果を用いて再計算を省くことができる手法である。探索処理では共通の前段階があることに着目し、GGSIに搭載されている DPM cache ではこのメモ化を利用している。探索を試行錯誤する際に、探索の中間結果が残っていて次回以降の実行時に再利用することができれば探索にかかるコストが軽減される。また、自分以外の誰かが行った別の探索の中間結果も自分の探索処理に利用することができるになればさらなる効率化を図ることができる。

3.4 コンポーネントプロセスの管理と同一性の保証

GGSIはコンポーネントプロセスの参照透過性を利用して、実行した結果を中間ファイルとして保存し再利用している。そのため GGSIはコンポーネントプロセスとキャッシュを管理する上で、以下の条件を満たさなければならない。

1. コンポーネントプロセスの機能レベルの同一性の保証
2. 呼び出し引数の同一性の保証
3. プロセス・引数と出力（中間ファイル）の強い結び付け

GGSIでは、これらの条件を満たすために、コンポーネントプロセスのメモイメージと引数の組を元としたハッシュ値を生成し、それを利用することによってコンポーネントプロセスの管理と同一性の保証を行っている。

4. 評価

4.1 フレームワークの有効性

今回行った実験で動作させたアプリケーションは以下の4つである。

- ・富士山探索アプリケーションの移植
- ・直線検出アプリケーション
- ・顔検出アプリケーション
- ・円検出アプリケーション

富士山探索アプリケーションとして用いたのは、SPGF[4]という富士山に似た山を世界中から探し出すためのアプリケーションをフレームワーク上に移植したものである。直線検出・顔検出アプリケーションは、

それぞれの検出で OpenCV[5]ライブラリを用いた簡単なものである。それぞれのアプリケーションで行っていることは、GEOTiff 画像一枚に対して探索を行い、その結果を出力しているだけである。これらのアプリケーションをフレームワーク上で動作させることで、指定した範囲全ての探索を行うことができる。データ取得等の探索以外の処理はフレームワークで共通化されているので、実行時に探索に用いるアプリケーションを変更するだけで異なる探索が行える。これら以外のアプリケーションを追加する時は、GEOTiff 画像一枚に対し探索を行ってその結果を出力するようなプログラムになっていれば、すぐに追加し実行することが可能である。

今回の実験で使用したデータは、富士山探索アプリケーションでは SRTM[5]を、直線検出・顔検出・円検出アプリケーションでは LANDSAT[6]を用いた。SRTM はスペースシャトルによって収集された DEM (Digital Elevation Model: 数値標高モデル) データであり、海上や極地など一部の地域を除いた世界中のデータがある。LANDSAT は人工衛星によって撮影された可視画像であり、SRTM と同様にほぼ全世界をカバーしている。LANDSAT には複数のバンドがあるが、今回の実験ではバンド1を用いた。フレームワークでは衛星探索データを GEOTiff に変換することで、統一して扱い、探索に使用しているので、探索実行時に使用するデータを変更してもアプリケーションのプログラムの中身を変更する必要はない。

GGSIで SRTM のデータ一枚を取得するのに平均で約 5 秒かかる。SRTM のデータは全世界分では約 14,000 枚あるので、地球を全て探索する際にあらかじめ GGSIサーバに保存されているデータを用いれば、データ取得分の約 20 時間を短縮できることになる。

今回の計算に用いたサーバを表1に示す。このサーバを用いて、富士山探索では全世界を対象に、直線検出・顔検出では北アメリカを対象にして、GGSI上でアプリケーションを実行した。

表1 サーバ諸元

CPU	メモリ	HDD	OS
Xeon 2.40GHz×2	1GB	150GB	CentOS 5.5

4.2 富士山探索アプリケーション

4.2.1 アプリケーションの詳細

移植の実装はCで行った。このアプリケーションは、以下のような複数のプロセスから構成されている。

- 1: 探索範囲の衛星探索データファイルの取得（フレームワーク処理）
- 2: 山の領域を抽出し、山ごとの山頂の座標・領域の

面積等の情報処理

3：発見した山と富士山のシルエットを比較し、判定した類似度が指定された閾値を超えていれば富士山と似た山と判断し座標を出力

4.2.2 結果

このアプリケーションの出力は、発見した山の座標である。図4はGoogle Earth[6]上で出力結果の座標にピンを表示したものである。図5はその一部の中央アメリカのコスタリカを拡大したものであり、図5に写っている一番上のピンで発見した山の写真を、Panoramio[7]で取得したものが図6である。このように実際に富士山に似た山を発見することができた。

今回発見した山が実際に富士山に似ているかの検証は、Google EarthとPanoramioを用いて行ったが、図3のように結果座標からGoogle Maps上にフレームワークがピンを表示することも可能である。また例えば、出力座標から地名を取得し、その場所の画像を取得する機能をコンポーネントプロセスとして組み入れれば、フレームワーク上での検証も可能である。

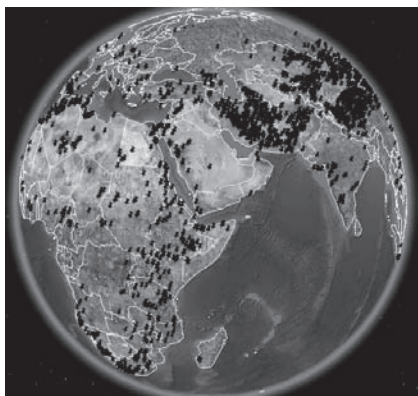


図4 富士山探索アプリケーションの結果

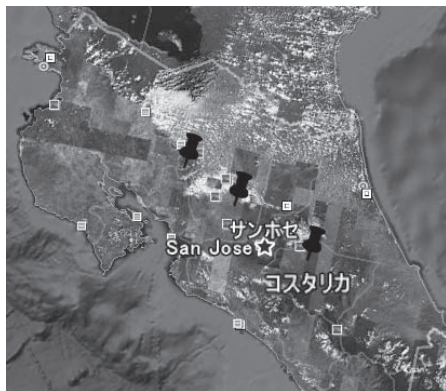


図5 結果のコスタリカ拡大図



図6 Volcan Arenal

4.3 直線検出アプリケーション

4.3.1 アプリケーションの詳細

直線検出アプリケーションでは、まず準備段階として検出に必要なフォーマットに変換するプロセスと、直線を検出し、入力画像に結果を出力するプロセスに分かれている。検出を行うプロセスではOpenCVを用いている。プログラムはC++で実装されたOpenCVのサンプルコードに対して、画像読み込み部分を変更した程度のものであり、開発期間も1日程度でコード行数も150行程度である。

4.3.2 結果

このアプリケーションでは、読み込んだ画像に対して検出した場所に、直線を描き込んでいる。図7は出力結果を拡大したものであり、図8はGoogle Earth上の同じ位置の写真である。図7上の白い直線が検出した結果で、川に沿って線が引かれているものもある。

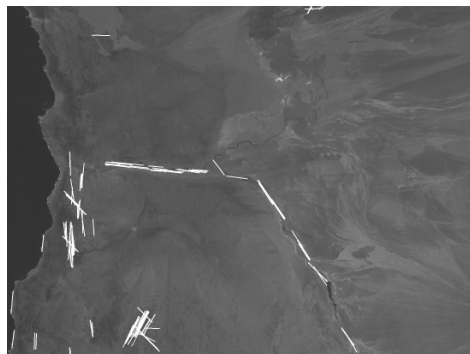


図7 直線検出の出力結果



図 8 Google Earth 上での画像

4.4 顔検出アプリケーション

4.4.1 アプリケーションの詳細

顔検出アプリケーションも直線検出アプリケーション同様、フォーマットを変換するプロセスと顔を検出するプロセスで構成されており、OpenCV を用いている。こちらも同じく簡単に開発できたものであり、開発期間は1日程度、コード行数は約200行である。

4.4.2 結果

このアプリケーションは、読み込んだ画像に対して顔と検出した箇所に丸を表示させている。図9と図10が出力結果の例である。

直線検出と顔検出のアプリケーションは、OpenCVの機能を利用して作成したものであるため、はっきりとわかる出力結果が得られないことも多かったが、ライブラリ等を利用し組み入れたプログラムを GGS II で簡単に動作させられることが確認できた。単に画像データ一枚に対し探索を行うようなプログラムの記述を加えフレームワーク上で実行するだけで、全世界の探索を行うことができる。

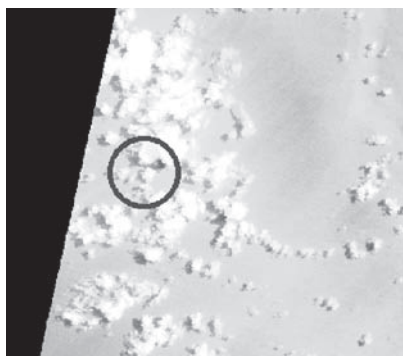


図 9 顔検出結果の例 1



図 10 顔検出結果の例 2

4.5 円検出アプリケーション

4.5.1 アプリケーションの詳細

円検出アプリケーションは Java で実装している。GGS II では、動作するアプリケーションの言語を制限していないので、他のアプリケーションと同様にフレームワークにのせることで探索を行うことができる。

4.5.2 結果

図11は出力結果を富士山探索と同様に Google Earth 上に表示したものである。このアプリケーションは、直線検出や顔検出のように簡単に開発したものではないので、はっきりとした結果が表れている。図11の場所はブラジルであり、円形農場を検出できている。

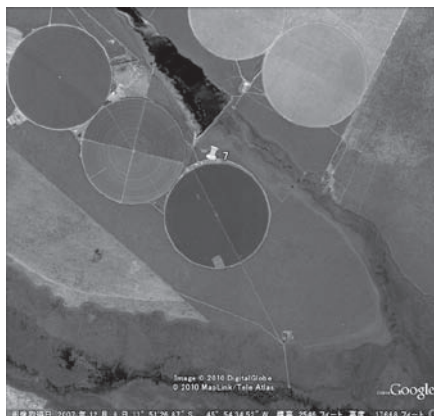


図 11 円検出結果

5. 関連研究

先行研究として、GEO Grid Spider (GGS) [10]がある。GGS は、GEO Grid (Global Earth Observation Grid) [11][12]を用いて世界を探索することを目的としている。

GEO Grid とは、グリッド技術を用いた、地球観測衛星データの大規模アーカイブ・高度処理を行うシステムで、ASTER センサーデータ[13]を提供することでユーザが手軽に地球データを扱えることを目指している。

GGIS 上で任意のアプリケーションを動かすことで、容易に世界を探索できる。しかし GGIS は ASTRE センサーデータに特化しているということと、利用するにはユーザ登録が必要であるといった制約があったため、我々は新たに多くの種類のデータを扱うことができ、手軽に利用できることを目的とした GGIS II の開発を行った。GGIS を用いたアプリケーションの一例として、[4]の研究で用いられているものがある。これは富士山と類似している山を探索するアプリケーションで、富士山と似た山を世界中から発見することができている。GGIS と比較して GGIS II は、あらかじめ Web サイトに登録することで様々な種類のデータを扱えることができ、すでにあるコンポーネントプロセスを再利用し組み合わせることができるという利点がある

既存の製品で地理情報を扱うシステムでは、ArcGIS[14]がある。ArcGIS は地理情報の表示・分析・検索・管理等の機能を提供している。しかしユーザがこのような多くの機能を使いこなすには、ソフトウェアの使用法と機能を把握しなければならず、手軽に扱いたいユーザにとってはそのことが障壁となることもある。GGIS II では、ユーザのデータ収集の手間を省き、ユーザが簡単にアプリケーションを構成できるようなアプローチを行っている。

6. まとめ

手軽かつ短時間で効率的に地球規模の探索を行うことを目的とする衛星データ探索フレームワーク GGIS II を提案し、設計・実装した。GGIS II ではユーザが手軽に扱えるユーザインタフェースを提供し、衛星探索データを自動で取得する機能を備え、またユーザ独自のプログラムも登録可能である。また短時間で効率的に探索を行うために DMP cache を搭載し、計算コストを省いている。

今回の実験では、DEM データの SRTM と可視光データの LANDSAT を用い、実際に富士山探索アプリケーション・直線検出アプリケーション・顔検出アプリケーション・円検出アプリケーションを GGIS II 上で実行することで動作確認をし、結果を検証した。その結果 GGIS II を用いることでユーザは、衛星画像データ種類を指定することができ、データの取得を意識しないで探索を行えることが確認できた。直線検出・顔検出アプリケーションは、単に OpenCV のサンプルコードに変更を加えた程度のものであったが、これらを実行し検証したことで、アプリケーション・コンポーネントプロセスの追加が容易であることと、そのアプリケーションを GGIS II 上で動作させることで簡単に世界

を対象にした探索を行えることが確認できた。

今後は、富士山探索アプリケーションのような、実際に研究に用いられるアプリケーションを利用しやすいフレームワークに発展させることも考えている。地理情報を扱っている研究者が利用しやすいフレームワークを調査し、GGIS II に反映させることを目指す。そして研究者に実際に利用してもらうことで、科学研究に貢献したい。

7. 今後の課題

GGIS II では、ユーザが意識せずに様々なデータを扱えることを目的の一つとしている。現在 GGIS II は人工衛星が取得したデータを扱っているが、温度等の他の種類のデータや衛星探索データに限らず他の地理情報を適用することを検討したい。

世界規模の探索では、多大な時間がかかる。処理時間の短縮のために、中間データをキャッシュする DMP cache に加え、サーバを分散化することも検討している。現在 GGIS II の処理は一台のサーバで行われているが、複数台のサーバを用意し探索を自動的に並列処理させることでさらなる処理時間の短縮を図る。衛星データ探索の場合、一地域の探索処理が別の地域の探索処理結果に影響を及ぼすことが少ないと予想できる。つまり探索処理を地域ごとに分割して並列に行うことが可能であり、複数のサーバによる同時実行によって処理時間の短縮が見込めると考えている。またデータの分散についてどのように分散するかが問題であるが、一般的に探索対象となる領域が連続して繋がった領域であることを利用して、例えば隣あった領域のデータファイルは別のサーバに配置するといった方法が考えられる。

Web アプリケーションでは、可能な限り短時間で処理結果を返し、ユーザにストレスを与えないことが重要になる。そこで応答性を高める方法として、アプリケーションの処理結果を Web インタフェース上に逐次表示させることも検討したい。ユーザは処理の中途結果を知ること、実行中の処理を止めるかどうかの判断ができ、処理中の結果から条件の変更や新たなアプリケーションの作成をすることで、効率的な探索を行うことができる。途中結果だけでなく処理開始前にも、計算時間の見積もりや中間データのキャッシュ状況を可視化したものを表示することで、ユーザに対し効率的な探索を行うための情報を与えることができる。

現在の GGIS II は処理結果を画像や座標等で Web ブラウザ上に表示することを想定しているが、ユーザが

独自にこのような結果を取得して加工したい場合を考え、結果を取得する機能を API として公開することも検討したい。他にも計算処理を GGS II ではなくユーザ自信が行えるようにするために、衛星探索データを取得する機能の API というのも考えられる。

謝辞 産業技術総合研究所 山本直孝氏、関口智嗣氏、サイボウズ 山本泰宇氏、ドリームポート 金子勇氏に感謝する。

参考文献

- [1] GeoTIFF, <http://trac.osgeo.org/geotiff/>
- [2] Google Maps API ファミリー - Google Code, <http://code.google.com/intl/ja/apis/maps/>
- [3] 山田大地, 中野雄介, 長谷部雅彦, 和久田龍, 菌部知大, 手塚宏史, 稲葉真理, "衛星データ探索フレームワークのための Data-Process Memoization cache の提案"
- [4] R. Wakuta, T. Sonobe, "SPGF Search Places by Geographical Features all around the world Search and verification system", SAINT2010, Seoul, KOREA
- [5] OpenCV.jp, <http://opencv.jp/>
- [6] huttle Radar Topography Mission, <http://www2.jpl.nasa.gov/srtm/>
- [7] Landsat.org Home Page, <http://landsat.org/>
- [8] Google Earth, <http://earth.google.com/intl/ja/>
- [9] Panoramio - 世界の写真, <http://www.panoramio.com/>
- [10] T. Sonobe, T. Tsutsumi, S. Gouda, S. Sekiguchi, N. Yamamoto, H. Yamamoto, H. Tezuka, I. Kaneko, K. Hiraki, M. Inaba, "GEO Grid Spider", 25th NORDUnet Conference, Copenhagen, Denmark, 2009
- [11] Y. Tanimura, N. Yamamoto, Y. Tanaka, K. Iwao, I. Kojima, R. Nakamura, S. Tsuchida, S. Sekiguchi, "Evaluation of Large-Scale Storage Systems for Satellite Data in GEO Grid", Proceedings of ISPRS 2008, Vol.37, Part B4, pp.1567-1573, 2008
- [12] S. Sekiguchi, Y. Tanaka, I. Kojima, N. Yamamoto, S. Yokoyama, Y. Tanimura, R. Nakamura, K. Iwao, S. Tsuchida, "Design Principles and IT Overview of the GEO Grid", IEEE Systems Journal, Vol.2, No.3, pp.374-389, 2008
- [13] ASTER SCIENCE PROJECT, <http://www.science.aster.ersdac.or.jp/index.html>
- [14] ESRI ジャパン株式会社 - GIS (地理情報システム) ソフトウェアリーダー, <http://www.esrij.com/index.html>