

筑波大学大学院博士課程

システム情報工学研究科特定課題研究報告書

海底コアCTスキャンイメージ可視化のための
クラウドサービスの開発
—分析専用閲覧機能を有するタブレット端末
向けクライアントソフトウェア—

佐々木 慎

修士（工学）

（コンピュータサイエンス専攻）

指導教員 高橋 伸

2013年3月

概要

筑波大学大学院「高度 IT 人材育成のための実践的ソフトウェア開発専修プログラム」における特定課題研究として、独立行政法人海洋研究開発機構（JAMSTEC）高知コア研究所との共同研究のもと、「海底コア CT スキャンイメージ可視化のためのクラウドサービスの開発」をテーマにプロジェクトを実施した。JAMSTEC が保有する地球深部掘削船「ちきゅう」によって、地球環境変動、断層と地震および地殻内生物圏の解明を目的として海洋掘削航海が行われている。「ちきゅう」は海底を掘削し柱状の地層サンプルを柱状の掘削コア試料として採取しており、その試料を用いた研究が精力的に行われている。船上には X 線 CT スキャナが搭載されており、掘削コア試料をスキャンデータとして保存することが可能である。しかし掘削コア試料のスキャンデータをスムーズに閲覧するためには、データ量が大きく、さらに閲覧するには高いスペックの端末が必要である、コア試料を閲覧するために適したソフトウェアが存在しないといった問題がある。本プロジェクトでは上記で述べた課題を解決するためのサービスの開発を行った。本サービスでは掘削コア試料のスキャンデータの画像処理を行うサーバ、サーバで描画された画像を閲覧するためのクライアントアプリケーション *Virtual Core Viewer* を開発した。筆者は分析専用閲覧機能を持つタブレット端末向けクライアントアプリケーションの設計、開発を担当した。本サービスにより、ユーザはデータ容量や端末のスペック等を考慮せずに掘削コア試料を利用することが可能となり、掘削コア試料のスキャンデータの利用が促進され、地球科学関連の研究の発展や理解が促されると期待できる。

目次

第1章	はじめに	1
第2章	本研究の背景	3
2.1	予備知識	3
2.1.1	海洋掘削プロジェクトと掘削コア試料	3
2.1.2	DICOM	3
2.1.3	クラウドコンピューティング	5
2.2	従来の課題と本研究での解決策	5
第3章	海底コア CT スキャンイメージ可視化のためのクラウドサービス	8
3.1	システム概要	8
3.2	利用シーン	8
3.3	提供する機能	9
3.4	システム構成	10
3.4.1	クライアントアプリケーション	10
3.4.2	ゲートウェイ	10
3.4.3	レンダリングエンジン	12
3.5	筆者の担当範囲	13
第4章	分析専用閲覧機能を有するタブレット端末向けクライアントソフトウェア	14
4.1	担当した範囲の概要	14
4.2	機能設計	14
4.3	画面・インタフェース設計	16
4.3.1	START 画面	18
4.3.2	CORESELECT 画面	19
4.3.3	WHOLE 画面	20
4.3.4	COLOR 画面	21
4.3.5	CUT 画面	25
4.3.6	INFO 画面	26
4.4	内部設計	26
4.4.1	クライアントアプリケーションとサーバ間のインタフェース	27
4.4.2	画像描画リクエストの処理	27

4.4.3	コア検索リクエストの処理	29
4.4.4	色付けを保存するためのデータモデル設計	30
4.5	まとめ	30
第 5 章	まとめと今後の発展	32
	謝辞	34
	参考文献	35
付 録 A	プロジェクトの進行計画	36
付 録 B	Android アプリケーション設計書	38
B.1	プロジェクトディレクトリ構成	39
B.2	クラス一覧	42
B.3	メソッド定義	44
B.4	シーケンス図	68

目 次

2.1	掘削されたコア試料の流れ	4
2.2	DICOM フォーマット閲覧の一連の流れ	5
2.3	システム導入前, 導入後の利用イメージ	7
3.1	システム概要	9
3.2	システム構成	11
4.1	タブレットアプリケーションの画面遷移	17
4.2	通信中ダイアログ	18
4.3	START 画面	18
4.4	ネットワークの接続確認ダイアログ	19
4.5	CORESELECT 画面	20
4.6	WHOLE 画面	21
4.7	COLOR 画面	22
4.8	ヒートマップ関数	23
4.9	色付け変更前, 変更後の COLOR 画面	24
4.10	透過度変更前, 変更後のコア画像	25
4.11	CUT 画面	26
4.12	INFO 画面	27
4.13	画像処理リクエストの処理の流れ	28
4.14	色付けを保存するためのデータモデル	31
5.1	医用画像への適用例	33
A.1	開発スケジュール	37
A.2	Web アプリケーション版 Virtual Core Viewer	37
B.1	アプリケーションのディレクトリ構成	39
B.2	src のディレクトリ構成	40
B.3	layout のディレクトリ構成	41

表 目 次

4.1	クエリのパラメータ	29
4.2	レスポンスのパラメータ	29
4.3	コア検索に用いるパラメータ	30
A.1	スコープ	36
B.1	作成したクラス一覧 (1/2)	42
B.2	作成したクラス一覧 (2/2)	43

第1章 はじめに

筑波大学大学院「高度 IT 人材育成のための実践的ソフトウェア開発専修プログラム」における特定課題研究として、独立行政法人海洋研究開発機構（JAMSTEC）高知コア研究所との共同研究のもと、「海底コア CT スキャンイメージ可視化のためのクラウドサービスの開発」をテーマにプロジェクトを実施した。本報告書はこのプロジェクトの実施内容についてまとめたものである。

JAMSTEC は、日本と米国が主導する統合国際深海掘削計画（Integrated Ocean Drilling Program : IODP）[1] と呼ばれる海洋科学掘削計画の一機関であり、その掘削計画の一主力船である地球深部探査船「ちきゅう」を保有している。「ちきゅう」に搭載される大深度掘削システムにより従来では困難とされてきた断層帯や水深数千メートルの深海底から程度のよい掘削コア試料を採掘することが可能であり、それらの試料を用いて研究が行われている。「ちきゅう」船上ラボには X 線 CT スキャナが搭載されており、コア試料を船上で CT スキャンし、デジタルデータとして保存している。スキャンデータを用いることで掘削コア試料の内部構造を破壊することなく調査・分析を行うことが可能である。掘削コア試料のスキャンデータは高知コア研究所の Virtual Core Library を介してアクセス可能であり、自由にデータをダウンロードすることができる。

しかし掘削コア試料のスキャンデータを閲覧するためにはいくつかの課題がある。その課題は、掘削コア試料 1 本あたりのデータサイズが 300MB～400MB であり、それが数千本存在すること、スキャンデータの 3 次元化には高いスペックの端末が必要となること、スキャンデータは医用画像のフォーマットで保存されていること、閲覧には専用ソフトウェアが必要となること、である。

本プロジェクトでは上記で述べた課題を解決するためのサービスの開発を行う。本サービスでは掘削コア試料のスキャンデータの画像処理を行うサーバ、サーバで描画された画像を閲覧するためのクライアントアプリケーション *Virtual Core Viewer* を開発する。サーバ側でスキャンデータの保存と演算を実施し、描画に関わる一連の処理を行うことでユーザ側はデータの容量や端末のスペック等を考慮せずに掘削コア試料のスキャンデータを利用することが可能である。またクライアントアプリケーションとして掘削コア試料専用の閲覧インタフェースを開発し、コア解析のための新しい手段を提供する。

筆者は掘削コア試料専用のインタフェースを持つ Android タブレットアプリケーションの開発を担当した。Android タブレットアプリケーションでは、ドラッグやピンチなどタッチ操作を活かしたインタフェース、色付けやコアの検索機能など、従来の閲覧ソフトウェアには存在しなかった掘削コア試料を閲覧するための専用機能を提供する。

本プロジェクトの成果として、Google Play に Virtual Core Viewer という Android タブレット用アプリを公開した。またタブレット以外の PC 端末等では、高知コア研究所の Virtual Core Library から WEB アプリケーションを利用することができる。本プロジェクトで開発したサービスを研究者や教育関係者に利用してもらうことで、地球科学の研究のさらなる発展が期待できる。

本プロジェクトのメンバー，担当を以下に示す。

- 佐々木慎（アプリケーション担当）
- 岡本昂也（ミドルウェア担当）
- 坂本侑一郎（レンダリングサーバ担当）
- 山際伸一准教授（プロジェクト担当教員）
- 和田耕一教授（プロジェクト担当教員）
- 久光敏夫技術主任（JAMSTEC 高知コア研究所 共同研究者）

本プロジェクトは，開発を 3 段階に分けて開発を行っており，その開発スケジュールを付録 A に記載する。開発の区切りでは，久光技術主任や研究者の方にサービスを使用してもらい，そのフィードバックから追加や修正する機能を決め開発を行った。

本報告書は全 5 章で構成される。第 2 章では，本システム開発におけるプロジェクトの背景と開発するシステムの意義について述べる。第 3 章では，本プロジェクトで開発するシステムの構成について述べる。第 4 章では，筆者が担当した分析専用閲覧機能を有するタブレット端末向けクライアントソフトウェアについて述べる。第 5 章では，本プロジェクトのまとめと今後の発展について述べる。

第2章 本研究の背景

本章では、本プロジェクトに関する予備知識と本プロジェクトの意義について述べる。

2.1 予備知識

2.1.1 海洋掘削プロジェクトと掘削コア試料

東日本大震災のような巨大地震の成因や、地球の歴史を探る手段の一つとして海底掘削コア試料を用いた様々な分析・研究が精力的に行われている。ここ数十年をとっても、プレートテクトニクス の証明や、恐竜絶滅と隕石衝突など現在では常識ともいえる地球科学の知見が明らかにされてきた。(独)海洋研究開発機構(JAMSTEC)が所有する地球深部掘削船「ちきゅう」は、人類史上初めてマントルや巨大地震発生域への大深度掘削を可能にする世界初のライザー式科学掘削船であり、統合国際深海掘削計画(IODP)の一主力船として科学研究航海に携わっている[2]。「ちきゅう」はX線CTスキャナが船上に搭載された唯一の船であり、コア試料の判割やサンプリングによる物性や構造が変化する前に、その内部情報をデジタルデータとして保存することができる。掘削されたコア試料の保管までの様子を図2.1に示す。(1)まずIODPで掘削されたコア試料は1.5mごとに保存される。1.5mで区切られた試料を1セクションという。その後1セクションごとにX線CTスキャンが実施される。(2)実物の掘削コア試料は高知コア研究所にて冷蔵保存され、(3)掘削コア試料のスキャンデータはJAMSTEC横浜研究所で管理され、高知コアセンターのVirtual Core Library[3]を通じて簡単にアクセス可能である。ユーザはVirtual Core Libraryを介しスキャンデータをダウンロードし、専用ソフトウェアを用いることで可視化されたデータを閲覧することができる。

2.1.2 DICOM

DICOM(Digital Imaging and COmmunication in Medicine)は米国放射線学会(ACR)と北米電子機器工業会(NEMA)が開発した医用画像のフォーマットとそれらを扱う医用画像機器間の通信プロトコルである[4]。現在でもX線CTやMRIなどで撮影した医用画像の基本フォーマットとして用いられている。1スライスのDICOMファイルは、1枚の断面画像を格納しており、複数のDICOMファイルを繋ぎ合わせ再描画させることにより、3次元データとして利用することが可能である。DICOMフォーマットのファイルの閲覧にはOsiriX等のDICOMビューワソフトウェアが必要となる[5]。

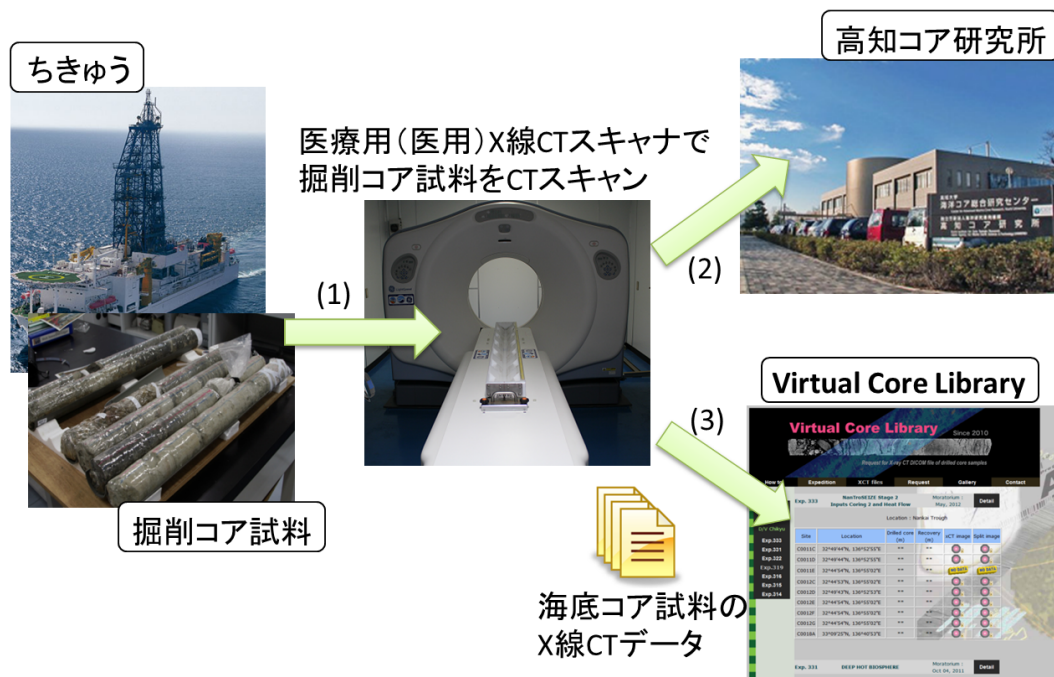


図 2.1: 掘削されたコア試料の流れ

DICOM フォーマットは JPEG や bmp フォーマットと違い、画像情報とタグ情報が含まれている。タグ情報には、画像 1 ピクセルあたりの実際の大きさやスライス厚、スライス方向などといった撮影情報に関するデータや、ID 番号、撮影部位、患者名などの情報がある。スライス一枚につきメタデータが含まれているため、画像のみの形式と比べファイルサイズが大きくなりがちである。また、患者名などの個人情報に関するデータが含まれるため、取扱いを慎重に行わなければならない。

本研究で扱う掘削コア試料のスキャンデータはすべて X 線 CT スキャナによって保存された DICOM フォーマットのファイルである。DICOM フォーマットの掘削コア試料の保存から閲覧するまでの一連の流れを図 2.2 に示す。CT スキャナによってスキャンされた掘削コア試料が DICOM フォーマットのファイルとして保存される。保存された DICOM フォーマットのファイル群から 3 次元画像を生成し、その 3 次元画像に対して色付けを行うことにより閲覧をすることが可能となる。

X 線 CT (Computed Tomography) とは、X 線コンピュータ断層撮影法の略称であり、様々な角度から X 線を照射することで、被写体の内部断面画像を得る技術である。X 線 CT スキャナによって保存された DICOM ファイルは画像情報を CT 値として保存している。CT 値とは X 線を吸収する程度を数値化したものであり、元素の密度と元素番号に比例関係にある係数である。物質の種類や構成元素、密度等によって異なる。CT 値は-1000~4000 の範囲を持つ値として定義され、水を基準値の 0 とし、空気を-1000 と定められている。

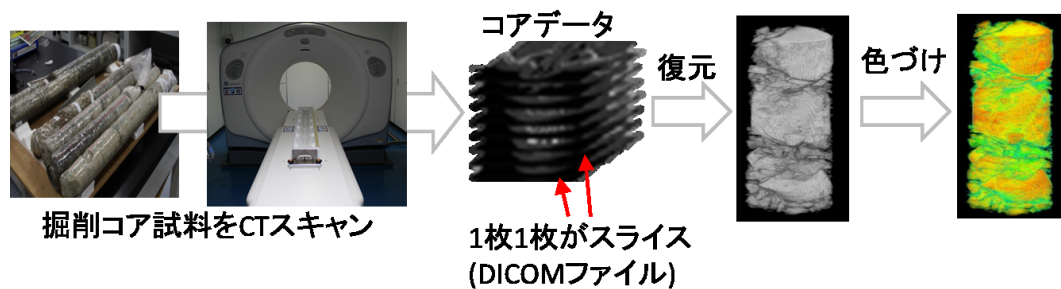


図 2.2: DICOM フォーマット閲覧の一連の流れ

2.1.3 クラウドコンピューティング

従来は、ユーザ自身がコンピュータのハードウェア、ソフトウェア、データなど必要なリソースを自分自身で管理しなければならなかった。これに対して、クラウドコンピューティング [8] というコンピュータの利用形態が注目されている。クラウドコンピューティングとは、インターネット上に浮かぶ巨大なコンピュータ群を必要に応じて利用できるコンピュータの利用形態のことである。

クラウドコンピューティングは、アプリケーションの機能をサービスとして提供する SaaS (Software as a Service)、アプリケーションを実行・開発するためのプラットフォームをサービスとして提供する PaaS (Platform as a Service)、サーバマシンやネットワークインフラをサービスとして提供する IaaS (Infrastructure as a Service) に分類される。クラウドコンピューティングのサービスの例として、仮想的なコンピュータ環境を提供する Amazon の Amazon Elastic Compute Cloud(Amazon EC2)[6] や Office の機能をブラウザから使用できる Microsoft の Office Web App[7] などがある。

クラウドコンピューティングでは、ユーザは必要最低限のリソースのみを用意すればよく、ユーザの端末スペックを考慮しないでネットワークでつながれた先のサービスを受けることができる。

2.2 従来の課題と本研究での解決策

前節で掘削コア試料の利用について述べたが、スキャンデータ化された掘削コア試料を用いた分析を行うには、以下のような課題がある。

- 1) データサイズが大きい
- 2) 端末に高い処理スペックが必要となる
- 3) 専用ソフトウェアが必要である

1) の課題は、掘削コア試料のスキャンデータのデータサイズの大きさである。掘削コア試料 1 セクションごとにスキャンが実施されるため、1 つの DICOM 画像郡には 1.5m 分の試料スキャン情報が格納される。公開されている掘削コア試料のスキャンデータの総数は約 4000 セクションあり、その 1 つのセクションあたり 300MB~400MB の容量がある。そのため複数の掘削コア試料のスキャンデータを閲覧するにはユーザ側には十分な保存容量が求められることになる。

この課題を解決するため本プロジェクトではリモートにあるサーバにスキャンデータを保存し、サーバを介して画像データを利用できるシステムを構築する。これによりユーザ自身がデータを保存する作業・準備は不要となる。

2) の課題は、掘削コア試料のスキャンデータを閲覧するために必要な端末のスペックの高さである。掘削コア試料のスキャンデータは、DICOM フォーマットで保存されており、閲覧するためには 2 次元スライス画像から 3 次元データを構築し描画する必要がある。この描画には高速なプロセッサと大容量のメモリが必要となる。

この課題を解決するため、リモートにあるサーバで描画処理を行い、その描画結果のみをユーザに返すアプローチを考案した。ユーザはサーバで描画された画像を閲覧できるスペックを持った端末があれば閲覧可能であり、高い処理スペックを持った端末を必要としない。そのため、従来は困難であった携帯端末での DICOM フォーマットのファイルの閲覧も可能となる。

3) の課題は、掘削コア試料を閲覧するための専用ソフトウェアが存在しないことである。掘削コア試料のスキャンデータは医用画像の標準フォーマットである DICOM で保存されているため、DICOM ビューワソフトウェア (OsiriX 等) を用いなければ画像の再構築はできない。しかし一般的な DICOM ビューワは医用に設計されており、骨や筋肉、靱帯などの生体情報にマップされた CT 値の閲覧は容易である一方で、掘削コア試料のような地質試料を描画する際にはユーザが適宜 CT 値に対して適切な色付けを行い、可視化しなければならない。

この問題については、掘削コア試料のスキャンデータ専用の閲覧インタフェースを提供する。

上記で述べた課題、解決策を踏まえ、本プロジェクトでは、DICOM フォーマットのファイルの 3 次元画像処理を行うサーバを構築し、ユーザには軽量の画像のみを返すサービスを構築する。加えて掘削コア試料のスキャンデータを閲覧するため専用のインタフェースを持ったクライアントアプリケーションを開発する。

これまでもネットワークに連結されたストレージを利用して DICOM 形式のファイルを提供する研究がある [9]。これらは DICOM フォーマットのファイルを提供するだけのサービスであり、DICOM ファイルを直接閲覧できる環境 (ソフトウェアや表示用機材) がないと利用できない。

また、リモートのレンダリングサーバを利用したボリュームデータの閲覧に関する研究がある [10]。この研究ではリモートのサーバで視覚化した 3 次元モデルを、MPEG ビデオストリーミングを通してモバイル端末に表示している。しかしこのシステムはインタラクションの問題が、ネットワークの帯域やレイテンシ等のネットワークの品質に左右されるという欠点があった。

本プロジェクトでは上記のようなユーザ側に残された負担や不便性を解決した新しいネットワークサービスを考案した。開発するシステムの導入前，導入後の利用イメージを図 2.3 に示す。従来は，ユーザのローカル端末に Virtual Core Library を通じて掘削コア試料のスキャンデータをダウンロードし，DICOM フォーマットが閲覧できる専用ソフトウェアを用いて閲覧していた。本サービス導入後は，リモートに保存されている掘削コア試料のスキャンデータをサーバ側で描画することで，その描画結果をインターネットを通して閲覧することが可能である。そのため，ユーザはローカルの端末にファイルをダウンロードすること，さらに描画演算すらも不要となるため，端末性能に依存せずに DICOM フォーマットのファイルを閲覧することが可能となる。

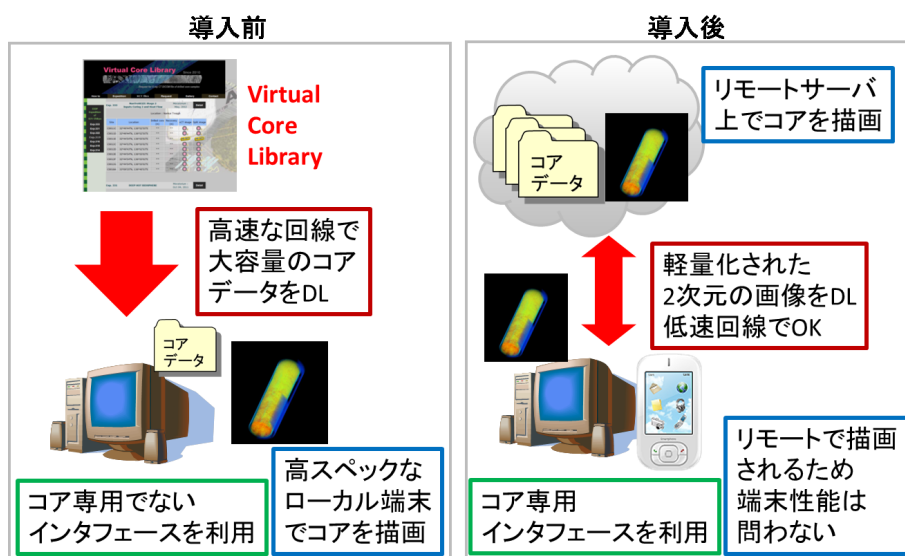


図 2.3: システム導入前，導入後の利用イメージ

第3章 海底コアCTスキャンイメージ可視化のためのクラウドサービス

本章では、本サービスの概要、システムの全体の構成について記述する。

3.1 システム概要

前節で述べた課題と解決策と地質学者からの要求を踏まえ以下のようにシステム要件を定義した。

- 端末性能に依存せず、掘削コア試料の閲覧が可能であること
- タブレット端末を用いて直感的に操作できること
- コアに対してインタラクティブな操作ができること
- 掘削コア試料の分析に適したインタフェースであること

上述した要件、前節で述べた課題、解決策をふまえ、タブレット端末等のモバイル機器上にて掘削コア試料のスキャンデータを閲覧することができるクライアントアプリケーション、及び掘削コア試料のスキャンデータの画像処理を行うサーバからなるクラウドサービスを開発する。本サービスのシステム概要を図 3.1 に示す。(1) まず PC やタブレット等のクライアントから、閲覧する掘削コア試料を選択する、表示する掘削コア試料を拡大するなどの操作を行い、その操作に応じた画像を描画するためのリクエストをサーバに送る。(2) サーバでは、クライアントによって選択された掘削コア試料の 3 次元モデルを DICOM ファイルから構築する。(3) その後、クライアントからの描画リクエストに応じて構築した 3 次元モデルを変換し、(4) 2 次元画像を生成する処理を行う。(5) 生成された 2 次元画像をクライアントに返し、クライアントで閲覧を行う。またクライアントアプリケーションには掘削コア試料を分析するためのインタフェースを有するアプリケーションを開発する。

3.2 利用シーン

本サービスの利用シーンの一つとして、「ちきゅう」船上での利用が考えられる。航海時は衛星回線による通信が主である。そのような貧弱な回線では 300MB～400MB の掘削コア試

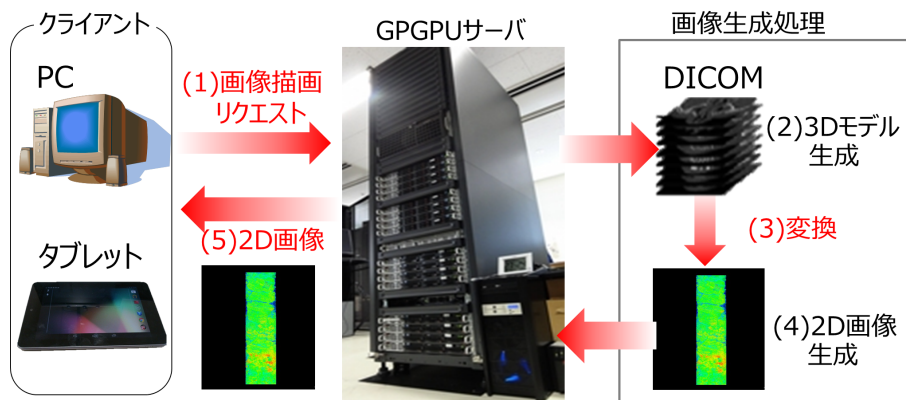


図 3.1: システム概要

料のデータをダウンロードするには非常に手間がかかる。しかし本サービスでは数十 KB の画像データのやり取りを行うため、貧弱な通信回線でも閲覧可能であり、インド洋等の海上を航海中に、過去に掘削した掘削コア試料を閲覧することが可能である。同様に陸上の研究施設でも、約 4000 本の掘削コア試料の中から自由に閲覧することが可能である。

また地球科学者の研究補助以外にも、学校の先生や広告、博物館での展示などで本システムを使用することも可能である。

3.3 提供する機能

掘削コア試料を閲覧するために本サービスで実現する機能について記述する。

1, 掘削コア試料の分析機能

掘削コア試料の拡大・縮小、回転、移動の機能は、3次元物体を閲覧する際の基本的な機能であり、物体中の全体像を表示させたり、任意の場所を詳細に見たい場合などに用いられる。地質学者が掘削コア試料を用いて分析を行う際は掘削コア試料の断面を観察することが多い。基本的には、柱状の掘削コア試料に対して縦方向に切断を行いその切断面を観察するため縦方向の切断を機能として提供する。

2, 掘削コア試料の選択機能

掘削コア試料を一意に選択する機能である。掘削コア試料は、航海を行ったプロジェクト名、掘削した位置、掘削したドリルの種類等から、掘削コア試料の試料名が決められている。例として、南海トラフへの航海プロジェクトにおいて、北緯 32 度、東経 136 度の場所で掘削されたコア試料は、333-C0011C という番号が掘削コア試料の試料名に含まれている。地質学者は上記に述べたような項目の識別番号を基にして数千本ある掘削コア試料の中から、掘削コア試料を一意に選択することができる。選択機能では、

各項目の識別番号を順番に絞り込んでいくことで掘削コア試料を選択できる機能を提供する。

3. CT 値に対する色付け機能

DICOM フォーマットで保存された掘削コア試料のデータには、掘削コア試料を入れるための筒や水など、掘削コア試料そのものの以外のデータも含まれている。また掘削コア試料ごとに含まれている CT 値の値も違っているため、表示する CT 値の範囲を指定する機能が必要となる。この機能を提供することで、掘削コア試料を分析するために必要な部分のみを表示することが可能となる。また CT 値に対する色付けや一部分の CT 値のみを透過する機能を提供することで、分析したい部分のみを強調することができる。

掘削コア試料に含まれる CT 値の範囲や幅を憶測で設定するのは困難であるため、その掘削コア試料の 3 次元データに含まれる CT 値の分布を表すヒストグラム図を表示する機能を提供する。

3.4 システム構成

本システムの構成を図 3.2 に示す。本システムは、ユーザが操作を行うクライアントアプリケーション、レンダリングエンジンへの処理の割り振りと掘削コア試料の検索を行うゲートウェイ、画像処理を行うレンダリングエンジンにわかれる。

3.4.1 クライアントアプリケーション

クライアントアプリケーションはユーザが掘削コア試料を閲覧するためのインタフェースである。Web ブラウザと Android アプリケーションの 2 種類を提供し、PC とタブレット端末両方から利用できるようにする。クライアントアプリケーションでは前節で述べた機能を利用できるインタフェースを提供し、そのユーザの操作に応じてクエリを作成し、サーバ側にリクエストを送る。レスポンスとして帰ってきたデータを解析し、掘削コア試料の画像、CT 値のヒストグラム図の表示を行う。

3.4.2 ゲートウェイ

ゲートウェイはクライアントアプリケーションからのクエリを受信し、レンダリングエンジンへの処理の割り振りや掘削コア試料の検索を行う役割を持つ。ゲートウェイはクライアントアプリケーションとのやり取りを行うためのレンダラ API と検索 API の 2 種類の API と、描画処理のためのレンダラモジュール、掘削コア試料検索のための検索モジュールの 2 種類のモジュールを持つ。

レンダラ API は掘削コア試料の描画のための API である。クライアントアプリケーションからのクエリを解析し、ロードバランサを通してレンダリングエンジンに命令を送る。クライアントアプリケーションにはレスポンスデータとして JSON 形式のデータを返す。

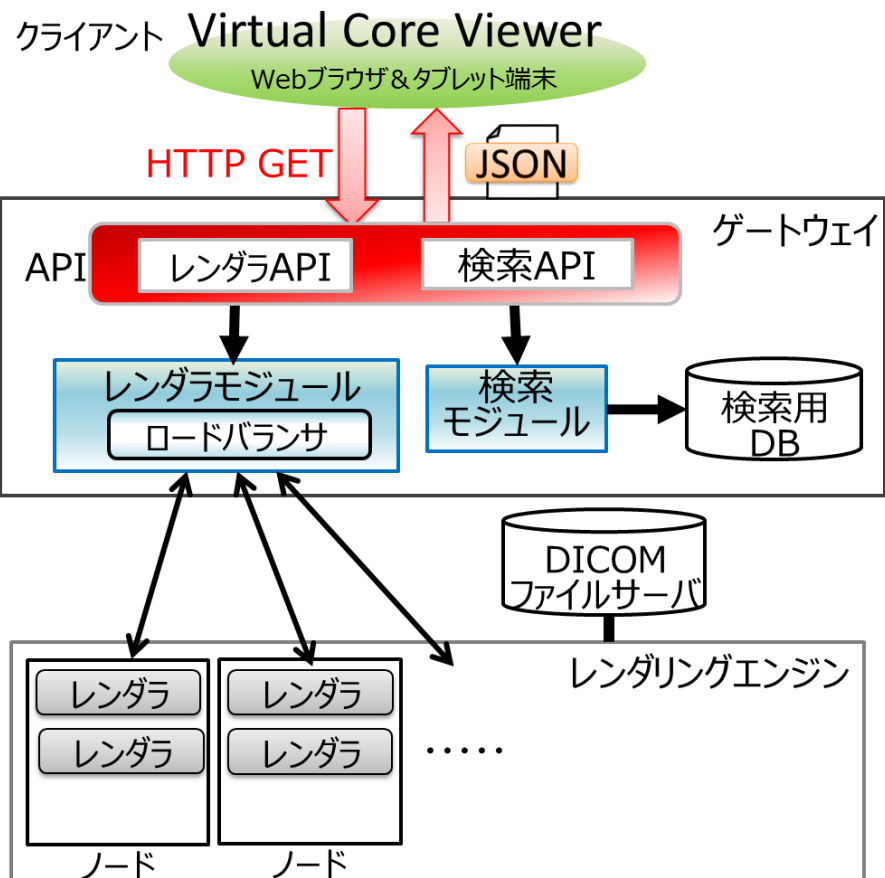


図 3.2: システム構成

検索 API は掘削コア試料の検索に用いられる API である。クライアントアプリケーションとは航海番号や掘削サイト番号等の掘削コア試料の検索に用いられるキーのやり取りを行う。

レンダラモジュールには、ロードバランサが含まれ、レンダリングエンジンのレンダラへの処理の割り振りを行う。レンダラの立ち上げ、停止を行う役割も持つ。

検索モジュールは検索 API で指定された、航海番号や掘削サイト番号等のキーより検索用 DB から掘削コア試料のスキャンデータを抽出する役割を持つ。クライアントアプリケーションからのリクエストに応じて、検索用 DB からデータを抽出する。また検索 DB への登録、編集を行うも役割を持つ。

ゲートウェイは CPU(Core i7 930 2.8GHZ)、メモリ 12GB で構成されている。

3.4.3 レンダリングエンジン

レンダリングエンジンは、レンダラによって 3 次元データから 2 次元画像を生成する処理を行う。レンダラ API からの命令に応じて 2 次元画像を生成するとともに描画する掘削コア試料のスケールや CT 値のヒストグラムの計算も行う。レンダラ API との通信にはソケット通信方式を用い、1 つの掘削コア試料につき 1 つのレンダラが生成される。画像生成に用いる掘削コア試料のスキャンデータはすべて DICOM ファイルサーバに格納されており、適宜、DICOM ファイルサーバからコアデータを読み込むことで画像を生成する。

クラスタは CPU(Intel Xeon E5645 2.40GHz (6 cores)) × 2、メモリ 12GB で構成されたノードが計 16 ノードで構成されている。ゲートウェイ、クラスタ間は Infiniband QDR で接続されており、ストレージは NFS で同期している。

本システムにおける、掘削コア試料のスキャンデータを閲覧する一連の流れを以下に示す。掘削コア試料のスキャンデータ閲覧のためにまずは、どの掘削コア試料を閲覧するかを指定する必要がある。そのためにクライアントアプリケーションから検索 API を利用し、掘削コア試料を一意に定めるためのリクエストを送信する。検索 API がリクエストを受け取ると、検索モジュールによって検索用 DB から情報を取得し、クライアントアプリケーションにレスポンスを返す。クライアントアプリケーションではレスポンスのデータを受けて、閲覧したい掘削コア試料のリクエストをレンダラ API に対して送信する。

レンダラ API がリクエストを受け取ると、レンダラモジュールを介してレンダリングエンジンに画像描画の処理を依頼する。画像描画処理の依頼は、レンダラモジュールのロードバランサにより、負荷の少ないノードへ依頼される。

レンダリングエンジンで画像を生成した後は、その画像情報をクライアントアプリケーションに返し、クライアントアプリケーションで 2 次元画像を表示する。以後、クライアントアプリケーションでの操作に応じて、レンダラ API を介したリクエストを送信し、レンダリングエンジンで操作情報を反映した画像が生成される。

3.5 筆者の担当範囲

筆者はクライアントのタブレット端末向けクライアントアプリケーションの設計，実装を担当した．タブレット端末向けクライアントアプリケーションでは，ドラッグやピンチなどタッチ操作を活かしたインタフェース，色付けや掘削コア試料の検索機能など，従来の閲覧ソフトウェアには存在しなかった掘削コア試料を閲覧するため専用のインタフェースを提供する．次章では，クライアントアプリケーションで提供する機能，そのインタフェース，アプリケーションの画面設計，アプリケーションの処理内容について記載する．

第4章 分析専用閲覧機能を有するタブレット端末向けクライアントソフトウェア

本章では、本プロジェクトにおいて、筆者が担当した分析専用閲覧機能を有するタブレット端末向けクライアントソフトウェアの設計・実装について説明する。

4.1 担当した範囲の概要

分析専用閲覧機能を有するタブレット端末向けクライアントソフトウェアは、3.3節で述べた掘削コア試料を閲覧するために必要となる機能を実現するインタフェースを有するアプリケーションである。

本アプリケーションは、操作に応じてクエリを作成し画像処理を行うサーバーへリクエストを行い、サーバーからのレスポンスによってコア画像を表示する役割を持つ。掘削コア試料を閲覧するための機能としてコアの絞り込み検索や縦方向の切断など、従来の閲覧インタフェースには存在しない機能を提供する。タブレット端末では、ドラッグやピンチなどタッチ操作が基本となることから、タッチ操作に適したインタフェースや画面の設計を行った。

本システムではタブレット端末の Android OS[11]を対象とした。Android OSはGoogleが作成したオープンソースのOSであり、iOSやWindows Phoneなど他のタブレット用OSと比べて、アプリケーションの配布が行いやすい、開発の柔軟性が高いといった利点がある。

4.2 機能設計

掘削コア試料のスキャンデータ閲覧方法について以下に記述する。

まず、閲覧する掘削コア試料を選択する必要がある。掘削コア試料は、掘削が行われた航海プロジェクト、掘削された位置などを示す識別番号により名前が付けられている。それらの識別番号を基に閲覧したい掘削コア試料を探し出す。

掘削コア試料のスキャンデータには掘削コア試料以外にも水やコアを保存するための筒など不要なものも含まれているため、不要のものを取り除き閲覧する。掘削コア試料は断層のずれや地層構造の変化を分析するために使用されるため、まず掘削コア試料の全体像を見てその掘削コア試料がどのような構造になっているのかを確認する。その後詳しく閲覧したい箇所をズームして分析を行う。断層等の分析を行う際は、柱状の掘削コア試料を縦方向に切断し、その断面を分析することが多い。

掘削コア試料のスキャンデータは上記で述べたように利用されている。そのため掘削コア試料を閲覧するために必要な機能を以下のように定義した。

拡大・縮小

表示するコア画像の拡大・縮小を行う機能である。掘削コア試料全体の表示、ある一部分を詳細に見たい場合などに必要となる。

回転

表示するコア画像の回転を行う機能である。掘削コア試料は円柱状の形状であるため、様々な角度から閲覧を行うために必要な機能である。

移動

表示するコア画像の移動を行う機能である。拡大などを行った際に表示できる範囲が制限されるため、表示位置を変更するために必要となる。

コアの絞り込み検索

閲覧するコアを一意に決めるための機能である。掘削コア試料は、航海番号、掘削サイト番号、ホール番号、コア番号、ビットタイプ、セクション番号の計6つの識別番号の組み合わせで掘削コア試料の試料名が決められている。現在掘削コア試料のスキャンデータの本数は4000本以上と膨大な数であり、掘削コア試料を一意に決めることが難しい。そのため、6つの識別番号を順に選択していくことにより、掘削コア試料のスキャンデータの候補を絞り込んでいくことで一意に定める。

コアの色付け

コア試料の密度組成の違いに対して色付けを行う機能である。コア画像はCT値の画像データが保存されており、サーバ側のレンダラでCT値と色の対応付けを行っている。そのCT値と色の対応付けをユーザが指定する機能を提供する。色付けの機能には、CT値の範囲を選択する、CT値と色の対応付けを変更するという2つが含まれる。CT値の範囲を指定する機能では、色付けを行うCT値の範囲を自由に指定する機能である。CT値の範囲は-1000～4000と定められているが、0以下は気体や液体のため、選択できるCT値の範囲を0-4000と制限し、その範囲の中で色付けを選択する。CT値と色の対応付けを変更する機能では、各CT値に対してどの色を対応づけるかを設定できる機能である。

CT値と色の対応表示

CT値に対してどの色がマッピングされているかを表示する機能である。

縦方向の切断

コアの縦方向の切断を行う機能である。地質学者は、掘削コア試料の断面を見ることで分析を行うことが多く、基本的には柱状の掘削コア試料に対して縦方向で切断した断面を分析に用いる。そのため本システムでは、縦方向の切断を機能として提供する。

透過度の変更

色付けの透過度を変更する機能である。透過度の変更を行うことで表面だけでなく、内部の3次元的な構成を閲覧することが可能となる。また分析したい部分以外の透過度を100%にすることで分析したい部分のみを閲覧することも可能となる。

スケールの表示

閲覧しているコアの実際の縮尺を表示する機能である。実際の縮尺を表示することにより、実物の掘削コア試料と比較を行いやすくする。

CT 値のヒストグラム表示

閲覧している画像のCT 値のヒストグラムを表示する機能である。掘削コア試料に含まれるCT 値の範囲や幅を予想することは困難である。そのため、3次元データに含まれるCT 値の分布を表示し、色付けを行うCT 値を選択する際に目安となるようにする。

CT 値と色の対応を保存

閲覧しているCT 値と色、透過度の対応付けを保存する機能である。現在閲覧している色付けや透過度を保存し、次回以降もその組み合わせを利用することが可能である。頻繁に使用する色付けのパターンなどを、再設定することなく呼び出すことが可能となる。

解像度の変更

レンダリングエンジンで描画を行うコア画像の解像度を変更する機能である。1つのコアの3次元データは最大で1 GByte 程あり、これをそのまま描画するにはかなりの時間を要する。そのためレンダリングエンジンでは、描画処理の時間短縮のために、3次元データの圧縮を行っている。本システムでは、5段階の解像度を用意し選択できるように機能を提供する。画像が荒くてもよいので応答時間が短くしたい場合や、応答時間は気にしないが詳細に見たい場合など、ユーザが求める品質や応答時間に応じて解像度を選択できる。

4.3 画面・インタフェース設計

本アプリケーションで設計した画面はSTART 画面、CORESELECT 画面、WHOLE 画面、COLOR 画面、CUT 画面、INFO 画面の6画面である。START 画面はアプリケーション起動後に表示される画面であり、端末のネットワーク設定の確認等を行う。CORESELECT 画面は、最初に閲覧するコアを選ぶための画面である。WHOLE 画面は、掘削コア試料の画像を画面全体に表示する画面である。COLOR 画面は掘削コア試料の画像の色付け、透過度に関する設定を行う画面である。CUT 画面は、掘削コア試料の切断を行うための画面である。INFO 画面は閲覧する掘削コア試料を変更する、解像度の変更を行うための画面である。各画面の遷移を図4.1に示す。

START 画面、CORESELECT 画面を除く4画面はタブにより画面の切り替えを行う。タブレット端末では画面の表示領域が限られているため、すべての機能を一画面に配置するのは

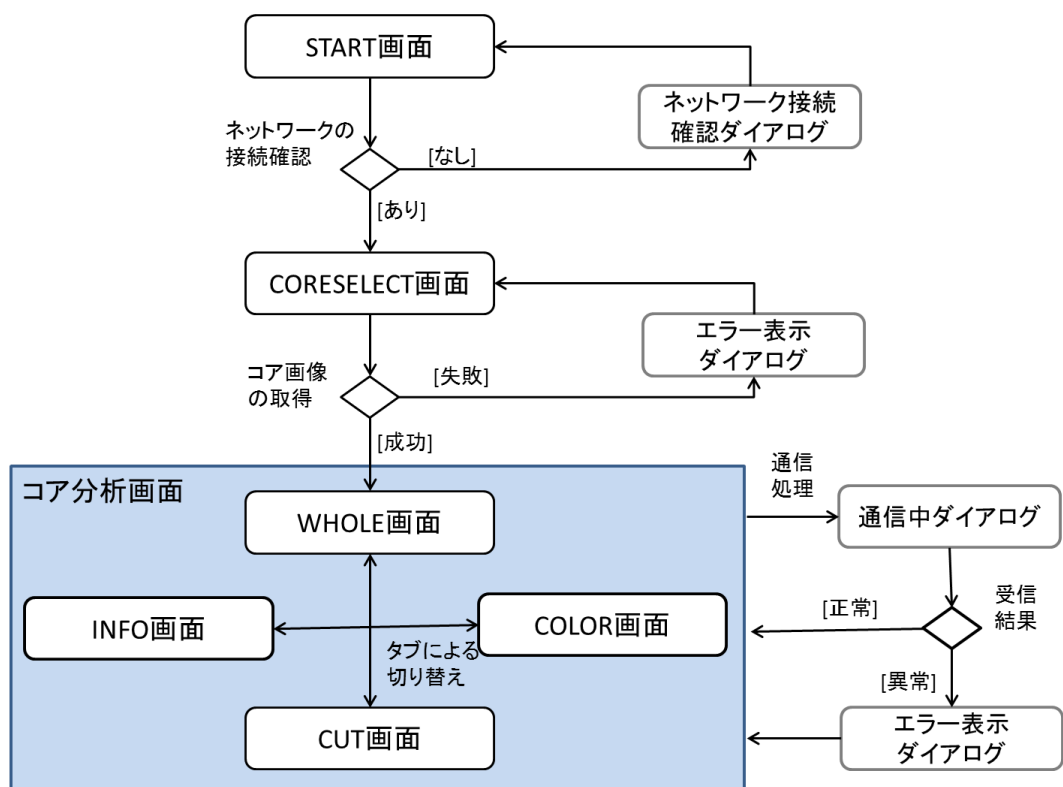


図 4.1: タブレットアプリケーションの画面遷移

好ましくない．分析を行う際に共通する機能をまとめ，それぞれの画面を切り替えやすくするようにタブを用いた画面の設計とした．

サーバ側の処理に一定の時間がかかるため，操作を行ってからレスポンスがあるまで，図 4.2 のようなダイアログを表示させ，通信の待機中であることを示す．ダイアログが表示されている間はユーザからの操作は受け付けない．以下にそれぞれの画面設計について詳しく記載する．

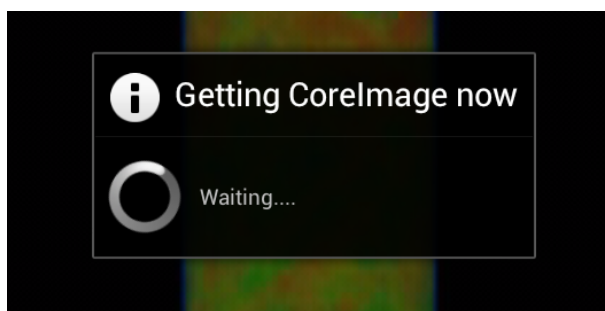


図 4.2: 通信中ダイアログ

4.3.1 START 画面

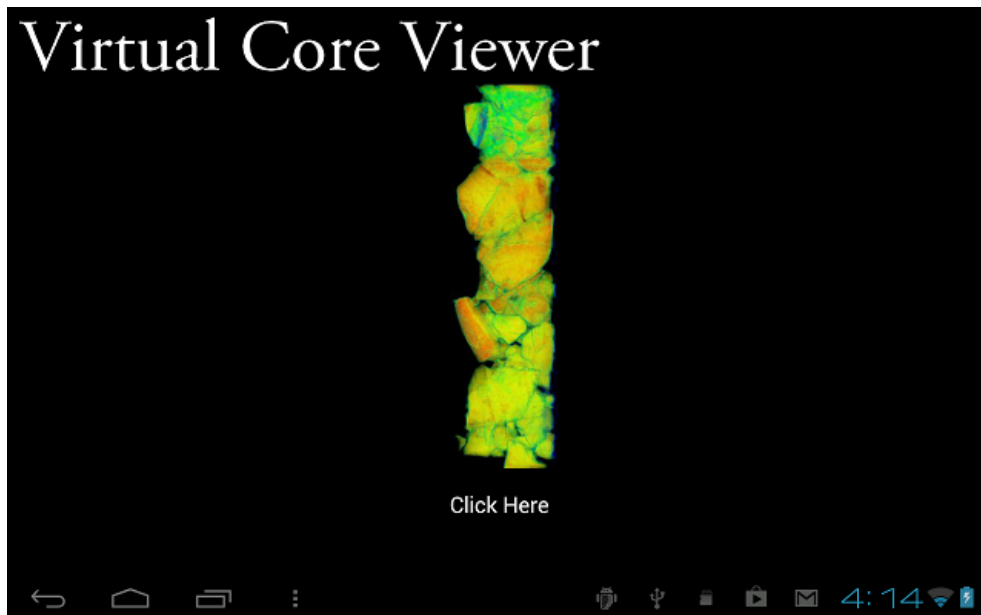


図 4.3: START 画面

図 4.3 はアプリを起動すると表示される画面である。本画面は、コア画像のスライドショー表示、ネットワーク接続の有無の確認の機能を持つ。図 4.3 の [Click Here] をタッチすることで最初に関覧するコアを選択するための CORESELECT 画面に遷移する。

コア画像のスライドショー表示では、コア画像を一定時間ごとに変化させて表示する。

ネットワーク接続の有無の確認では、Android 端末がネットワークに接続されているかどうかの確認を行う。本アプリケーションはネットワークに接続されていることが前提となっているため、本画面でネットワークに接続されているか否かを判断する。ネットワークに接続されていない状態で、図 4.3 の [Click Here] をタッチした場合は図 4.4 のようなダイアログを表示し、ネットワーク接続の設定の確認を促す。

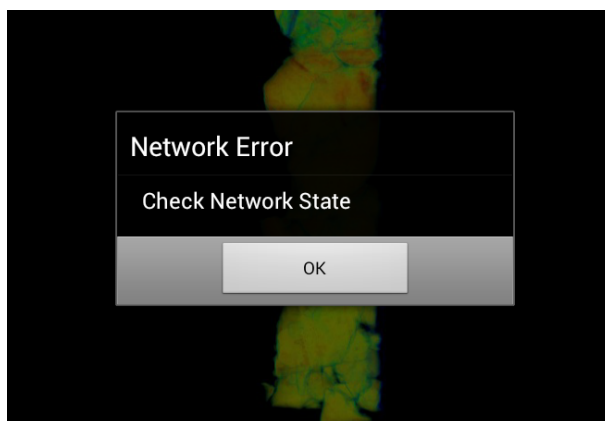


図 4.4: ネットワークの接続確認ダイアログ

4.3.2 CORESELECT 画面

図 4.5 は、最初に関覧するコアを選択する画面である。この画面では、コアの絞り込み検索の機能を提供する。

コアの絞り込み検索では、閲覧するコアを一意に決めるための機能を提供する。航海番号、掘削サイト番号等のキーを基にしてコアを一意に検索を行う。キーとなる項目は、ExpeditionNo(航海番号)、DisiteNo(掘削サイト番号)、HoleNo(ホール番号)、CoreNo(コア番号)、Bitttype(ビットタイプ)、SectionNo(セクション番号)であり、それぞれの項目を選択できるボタンを配置する。項目のボタンをクリックすると選択肢が表示されるのでその中から 1 つの項目を選んでいく。すべての項目を選んだ後に [SELECT] ボタンを押すことで、選択されたコアを閲覧するための WHOLE 画面に移行する。

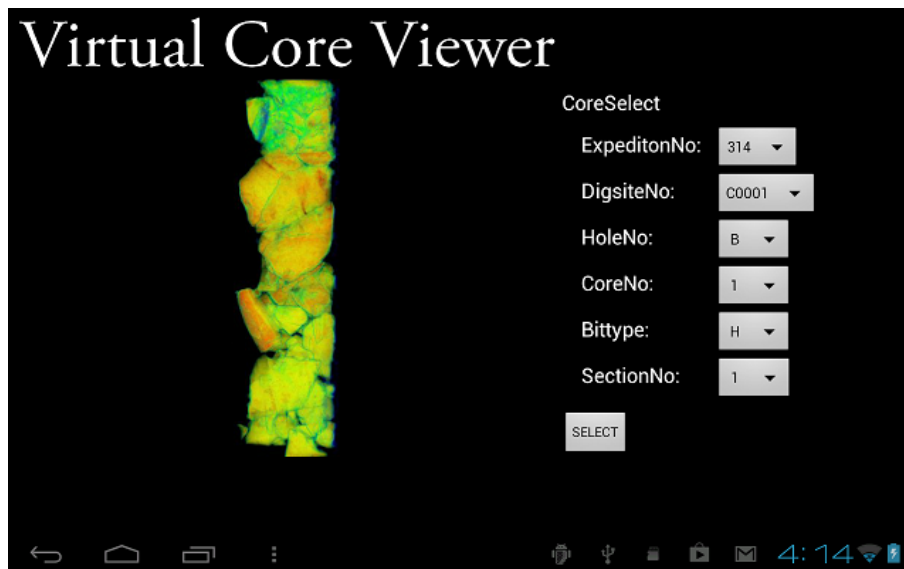


図 4.5: CORESELECT 画面

4.3.3 WHOLE 画面

図 4.6 はコア画像を画面全体に表示する画面である。この画面では、コアの拡大・縮小、コアの回転、コアの移動、スケールの表示の機能を提供する。

コアの回転

画面に対してドラッグ操作を行うことでコアを回転させる。上下方向のドラッグ操作が前後方向（手前，奥行）の回転，左右方向のドラッグ操作が左右方向の回転に対応している。ドラッグの開始点と終了点の長さに応じて回転角度を定めている。ドラッグ操作の回転，移動の切り替えはメニューによって切り替えることが可能である。ドラッグの終了時に画像取得のクエリを作成するので操作中はどの角度であるかが予想しづらい。そのためドラッグした量に応じて常時変化する簡易的な円柱を画面右上に表示し，操作の目安とする。

コアの拡大・縮小

ピンチイン動作で縮小，ピンチアウト動作で拡大を行う。ピンチ動作の大きさによって，拡大率，縮小率を定めている。拡大・縮小の最大値，最小値については制限を設け，0.1～3.0 倍に制限している。

コアの移動

画面に対してドラッグ操作を行うことでコアの表示位置を変更させる。上下方向のドラッグ操作によりコアの表示位置を変化させる。

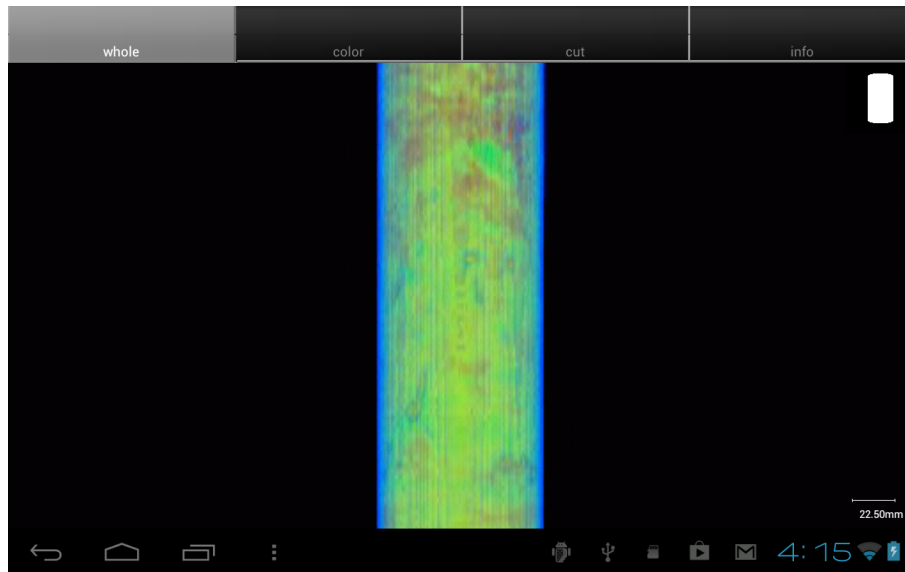


図 4.6: WHOLE 画面

ドラッグ操作がコアの移動，コアの回転どちらに割り当てるかはメニューから切り替えることができる．デフォルトではコアの回転にドラッグ操作が割り当てられている．

スケールの表示

表示されている画像の実際の縮尺を表示する．画面右下に一定の長さ線分と，その線分の実際の表示縮尺を表示する．

COLOR 画面，CUT 画面，INFO 画面の左側に表示されるコア画像は WHOLE 画面に表示されるものと同じ画像であり，上記で述べた操作を行うことが可能である．色付けを変更させる，縦切断を行う等の操作を行った場合すべての画像が変化する．

4.3.4 COLOR 画面

図 4.7 はコアの色付け，透過度を変更する画面である．この画面では，CT 値と色の対応表示，透過，CT 値のヒストグラム表示，CT 値と色の対応を保存の機能を提供する．

コアを描画するための主な機能として，CT 値と色を対応づけと色の透過度の変更がある．図 4.7 の (2) は CT 値と色の対応付けを行う部分であり，図 4.7 の (6) は，透過度を選択する部分である．色付けを行う部分と透過度を設定する部分を同じ部分で行うと操作が行いにくくなるため，それぞれの操作を分離した箇所で行えるように設計している．

CT 値と色の対応表示

図 4.7 の (2) は CT 値がどの色に対応づけられているのかを表している．図 4.7 の (2) は，横軸に CT 値，縦軸に色値を取った図となっている．横軸の CT 値は CT 値選択に

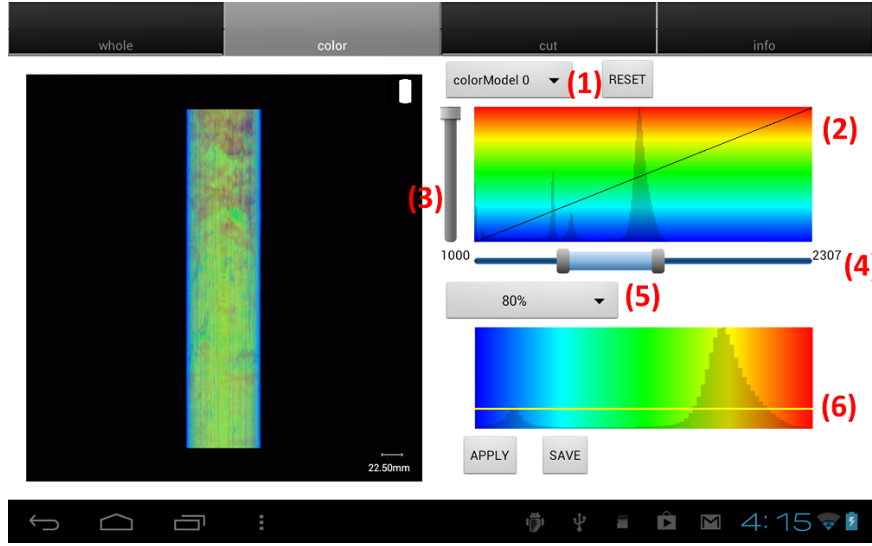


図 4.7: COLOR 画面

よって選択されている範囲の CT 値となっている．実際に色付けを行う CT 値とその CT 値がどの色にマッピングされているかは図 4.7 の (6) に示されている．色付けには RGB のヒートマップを用いている．RGB 値は CT 値 $v(v_{min} \leq v < v_{max})$ を用いて，式 4.1，式 4.2，式 4.3 で表され，図 4.8 で示される．ヒートマップの関数は値が低い場合は青色に，値が高い場合は赤色にマッピングする．

$$R(v) = \begin{cases} 0 & (v_{min} \leq v < \frac{v_{min}+v_{max}}{2}) \\ 0.25v - 32 & (\frac{v_{min}+v_{max}}{2} \leq v < \frac{3(v_{min}+v_{max})}{4}) \\ 15 & (\frac{3(v_{min}+v_{max})}{4} \leq v < v_{max}) \end{cases} \quad (4.1)$$

$$G(v) = \begin{cases} 0.25v & (v_{min} \leq v < \frac{v_{min}+v_{max}}{4}) \\ 15 & (\frac{v_{min}+v_{max}}{4} \leq v < \frac{3(v_{min}+v_{max})}{4}) \\ 63 - 0.25v & (\frac{3(v_{min}+v_{max})}{4} \leq v < v_{max}) \end{cases} \quad (4.2)$$

$$B(v) = \begin{cases} 15 & (v_{min} \leq v < \frac{v_{min}+v_{max}}{4}) \\ 31 - 0.25v & (\frac{v_{min}+v_{max}}{4} \leq v < \frac{v_{min}+v_{max}}{2}) \\ 0 & (\frac{v_{min}+v_{max}}{2} \leq v < v_{max}) \end{cases} \quad (4.3)$$

ヒートマップ関数に応じて線形に色付けを行うだけでなく，3 次スプライン補間を用いての色付けも提供する．図 4.7 の (2) の領域内でタッチした点を 3 次スプライン曲線の補間点とし，色付けの曲線を変更する．点 $(X_i - 1, Y_i - 1)$ ， (X_i, Y_i) の間を結ぶ 3 次スプライン補間の式 4.4 に開始点，補間点，終了点を適用させることで曲線の描画を行う．また補間点の最大の個数は 8 つに制限している．図 4.7 の RESET ボタンを押すこ

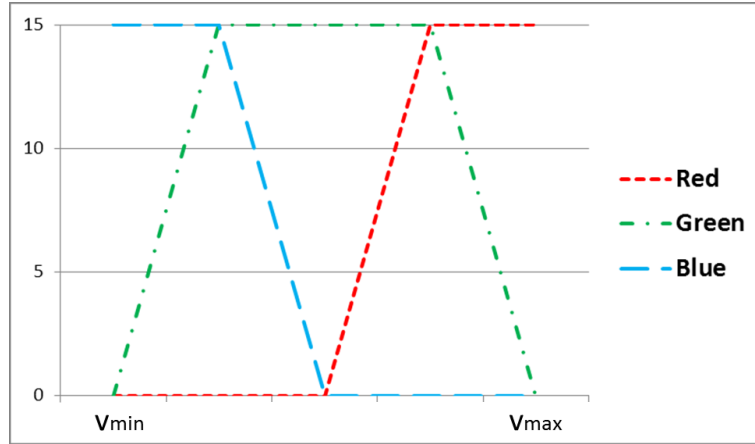


図 4.8: ヒートマップ関数

とで補間点を取り消すことができる。初期状態や補間点を一点も選択していない場合は CT 値に対して線形に線を描いている。

$$S_i(x) = \frac{f''(x_{i-1})}{6h_i}(x_i - x)^3 + \frac{f''(x_i)}{6h_i}(x - x_{i-1})^3 + \left(\frac{y_{i-1}}{h_i} - \frac{h_i * f'(x_{i-1})}{6}\right)(x_i - x) + \left(\frac{y_i}{h_i} - \frac{h_i * f''(x_i)}{6}\right)(x - x_{i-1})$$

ただし

$$h_i = x_i - x_{i-1}, f'(x_i) = \frac{y_i - y_{i-1}}{x_i - x_{i-1}}, f''(x_i) = \frac{f'(x_{i+1}) - f'(x_i)}{x_{i+1} - x_{i-1}}, \quad (4.4)$$

$$h(0) = 0, f''(x_0) = 0, f''(x_n) = 0$$

図 4.9 は、実際に色付けを変化させ適用した図である。図 4.9 の右図は CT 値に対して線形にヒートマップ関数を適用した場合であり、左図は 2 点の補間点によって色付けを変化させた場合である。このように濃淡の差を強調させた色付けを行うことが可能となる。

CT 値のヒストグラム表示

CT 値の分布を表すヒストグラムを表示する。ヒストグラムは横軸に CT 値、縦軸に 3 次元データの CT 値の度数を取った図となっている。図 4.7 の (1) には CT 値が 0~4000 の範囲の分布を表しているヒストグラムを重ねて表示し、色付けを行う CT 値を選択するための目安となる。図 4.7 の (6) には CT 値が色付けを行う範囲のみのヒストグラム図を重ねて表示し、色付けを行う範囲の CT 値がどのように分布しているかを確認するための目安となる。

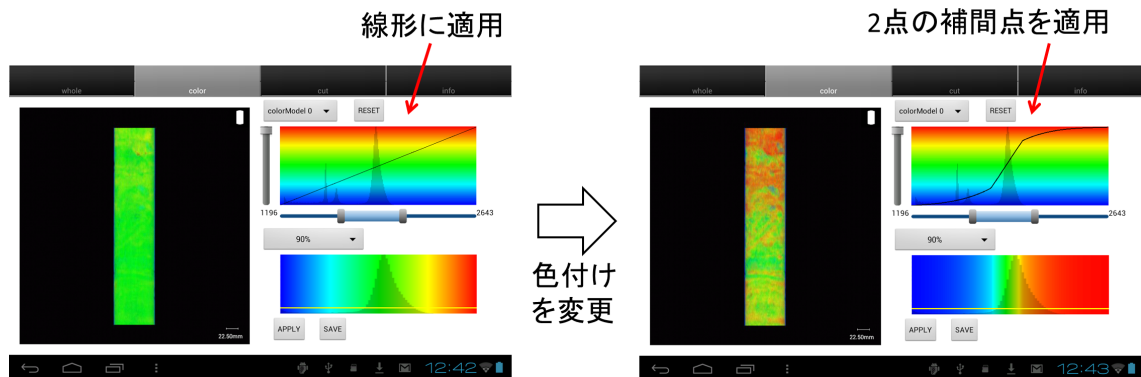


図 4.9: 色付け変更前, 変更後の COLOR 画面

表示するヒストグラム図の上限は CT 値の中の最大値となっているが, CT 値が低い部分の分布がわかりづらくなっている. そのため, 図 4.7 の (3) のスライダーによって, ヒストグラムの上限を変化させることによって, CT 値が少ない場所の分布を強調することができる.

色付け CT 値の範囲選択

図 4.7 の (4) のスライダーにより色付けを行う CT 値の最小値, 最大値を選択する. 左側のつまみは最小値を表しスライダー左側に数値を表示している. 右側のつまみは最大値を表しスライダー右側に数値を表示している. 二つのつまみの間をタッチしスライドさせることで, CT 値の幅を固定したまま色付け対象の CT 値を変更することができる.

透過度の変更

透過度の変更の機能には, 表示する CT 値すべてを一括して変更する方法とある範囲を個別に指定する方法の 2 種類を提供する.

一括設定

コアの透過度を図 4.7 の (5) の選択肢から選択することで, 色付けを範囲の透過度を一括して指定する. 透過度は 11 段階 (0%~100%) の中から選択することができ, 透過度を選択すると透過度が図 4.7 の (6) に反映される.

個別設定

コアの透過度を個別に指定する場合は図 4.7 の (6) を使用する. 図 4.7 の (6) は縦軸を透過度, 横軸を CT 値とし, 図 4.7 の (2) によって決められた CT 値と色の関係がマッピングされている. 図 4.7 の (6) の範囲をタッチすることにより一定区間ごとの透過度を設定する. 色付けを行う CT 値の範囲を 10 に区切り, 10 区ごとに個別に透過度を設定することができる. CT 値に対する透過度は黄色の線で示される.

図 4.10 は透過度を変更させた図である．図 4.10 の左図は透過度が 0% の場合であり，右図は青の部分の透過度を 90% に，その他の部分を透過度を 50% に設定した場合のコア画像である．

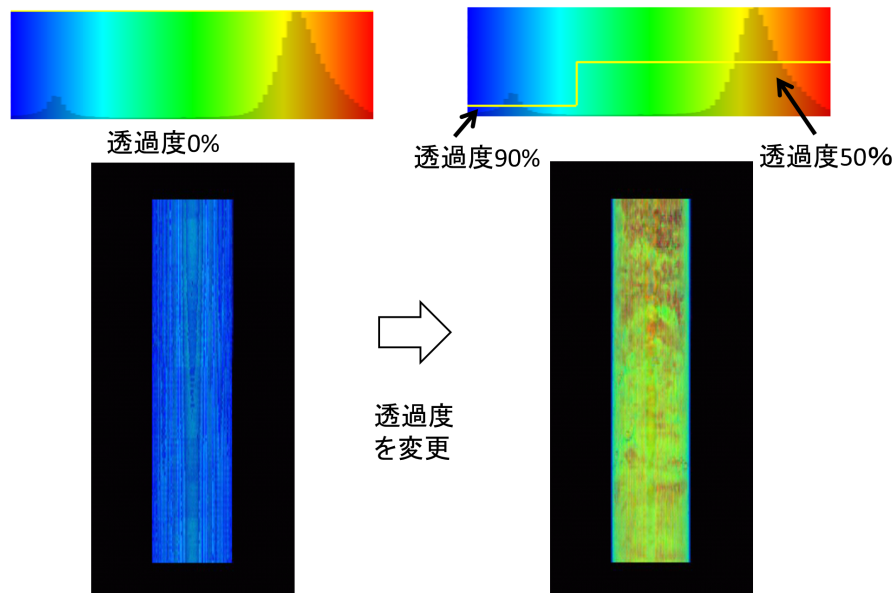


図 4.10: 透過度変更前，変更後のコア画像

CT 値と色の対応を保存

色付けを行う範囲，色付けの方法，透過度の設定を保存する．図 4.7 の [SAVE] ボタンを押すことにより，現在の色付け，透過の設定を保存する．色付け，透過の対応付けを表す名前を入力し，保存することが可能となる．保存してある対応付けは，図 4.7 の (1) から呼び出すことが可能である．

4.3.5 CUT 画面

図 4.11 はコアの切断を行う画面である．CUT 画面で提供する機能を以下に示す．この画面ではコアの切断の機能を提供する．

コアの切断

図 4.11 の (1) は切断箇所を表す画像であり，中央の白線に沿って切断することを示している．WHOLE 画面で行えるタッチの操作には対応していない．図 4.11 の (1) のコア画像に対する操作は図 4.11 の (2) スライドバーでのみ行える．スライドバーではコアの左右方向の回転角度を指定する．角度の指定後，[CUT] ボタンを押すと図 4.11 の 1 の画面中央の線より左側を切断したコア画像を表示する．現在は切断の回数を一回のみとしている．[UNDO] ボタンを押すことにより，縦切断の取り消しを行う．

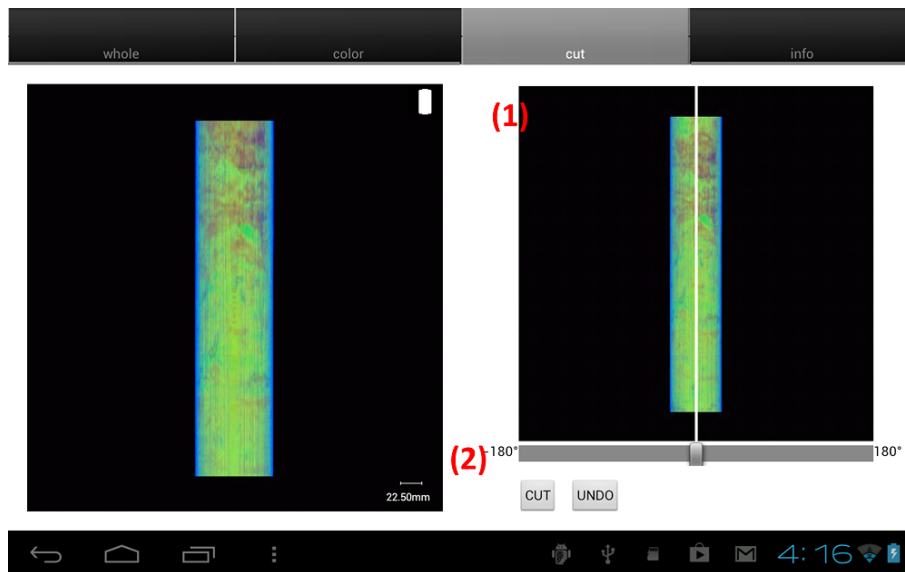


図 4.11: CUT 画面

4.3.6 INFO 画面

図 4.12 は表示コアの変更，解像度の変更を行う画面である．この画面ではコアの絞り込み検索，解像度の変更の機能を提供する．

コアの絞り込み検索

CORESELECT 画面で述べた検索機能と同様の機能を提供する．

解像度の変更

図 4.12 の (1) の選択肢から表示するコアの解像度を変更する．解像度は Highest, Higher, middle, lower, lowest の 5 段階から選択することができる．[SELECT] ボタンを押すことで，絞り込み検索機能で選択されている掘削コア試料を選択されている解像度でリクエストする．

4.4 内部設計

本アプリケーションの内部処理について記述する．開発した Android アプリケーションの詳細な設計については付録 B に記載する．本節では，クライアントアプリケーションとサーバ間のインタフェース，モジュール構成，画像描画リクエストの処理，コア検索リクエストの処理，色付けを保存するための DB 設計について記述する．

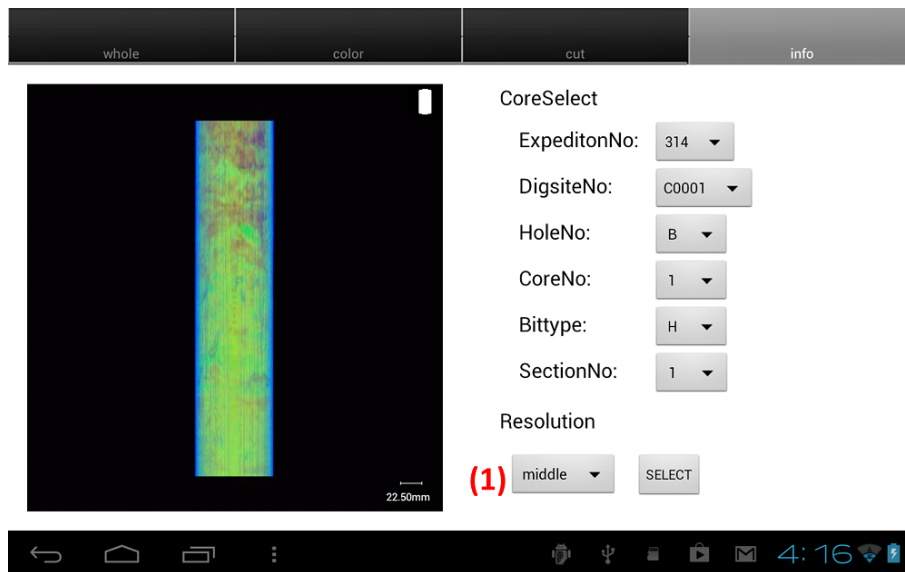


図 4.12: INFO 画面

4.4.1 クライアントアプリケーションとサーバ間のインタフェース

サーバ側では掘削コア試料の検索のための検索 API，画像描画処理のためのレンダラ API の二つの API が用意されており，それらの API を利用してサーバとクライアント間でデータのやり取りを行う．サーバが提供する 2 種類の API はどちらも，リクエストには HTTP GET を用い，レスポンスには JSON 形式のデータが返送される．

リクエストに用いられる HTTP GET は，パラメータとその値の組の羅列をリクエストのデータとして記述する．”`http://本サービスの URL?パラメータ=値&パラメータ=値&..`” のようにクエリを作成し，サーバに対してリクエストを行う．

レスポンスに用いられる JSON(JavaScript Object Notation)[12] は軽量なデータ記述形式であり，数字や文字列，配列などの様々なデータをキーと値の組み合わせを使ったデータ構造で記述できる．JSON は簡潔な記述でデータ構造を記述できるため，軽量なデータでメッセージをやり取りすることができる．

4.4.2 画像描画リクエストの処理

レンダラ API を利用した画像描画のリクエストを行う場合の一連の処理の流れを図 4.13 に示す．

レンダラ API でクエリとして用いるパラメータを表 4.1 に，レスポンスデータとして返ってくる JSON 形式のデータを表 4.2 に示す．ユーザの操作に応じて，表 4.1 に示すパラメータの値を基にクエリを作成する．

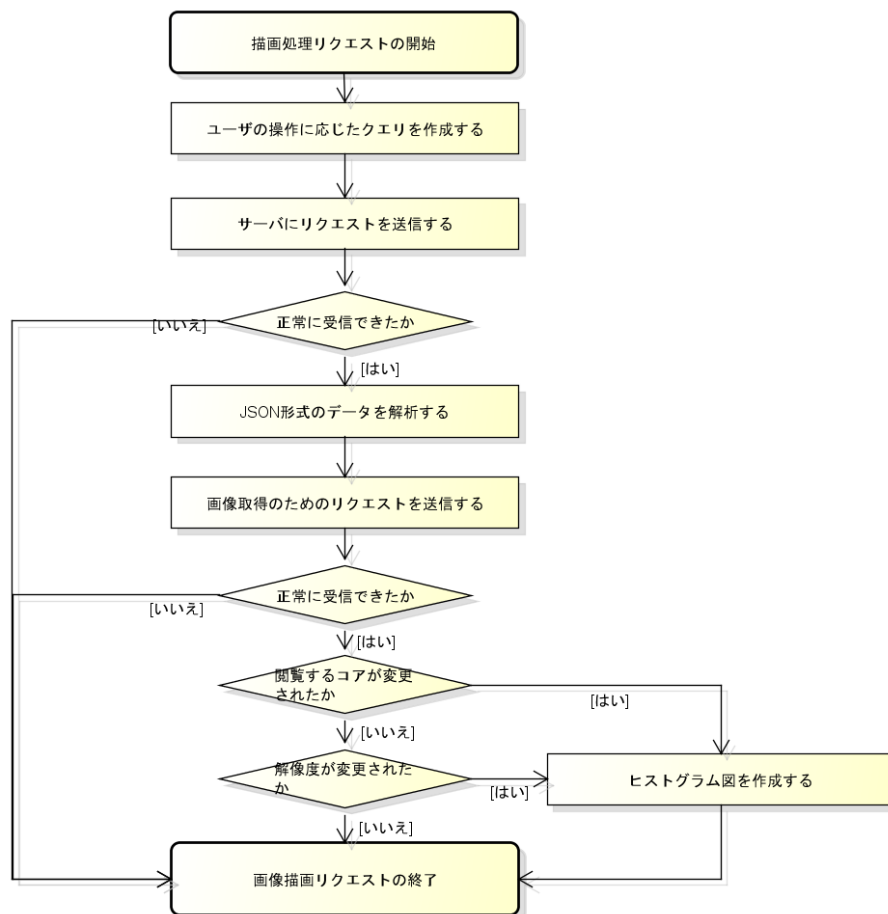


図 4.13: 画像処理リクエストの処理の流れ

表 4.1: クエリのパラメータ

変数名	型	概要
corenum	int	検索 API で返されたコア番号
zoom	float	拡大率
cut	int	縦切断の角度
resolution	int	解像度
rot_type	int	回転方法
(rotx,roty,rotz)	(int,int,int)	x,y,z 方向の回転角度
depth	int	コアの上下移動
color_type	int	色の付け方
cf	Array	CT 値と色の関係を表す組の配列
af	Array	CT 値と透過度の関係を表す組の配列
(eye_x,eye_y,eye_z)	(float,float,float)	現在の視点座標
(up_x,up_y,up_z)	(float,float,float)	現在の視点の上方向ベクトル
(movex,movey)	(int,int)	マウスの移動ベクトル
degree	int	回転角度

表 4.2: レスポンスのパラメータ

変数名	型	概要
img_url	String	生成された画像の URL
histgram	Array	CT 値のヒストグラムの配列
scale	float	1 ピクセルあたりの長さ
(eye_x,eye_y,eye_z)	(float,float,float)	現在の視点座標
(up_x,up_y,up_z)	(float,float,float)	現在の視点の上方向ベクトル

4.4.3 コア検索リクエストの処理

検索 API を利用したコアの検索に利用するクエリのパラメータと、レスポンスのパラメータを表 4.3 に示す。

表 4.3 に示されるパラメータを含んだクエリを検索 API にリクエストとして送信すると、そのクエリに対応するレスポンスのパラメータが返ってくる。レスポンスのパラメータは 1 つの値だけとは限らず、複数の値が含まれる。レスポンスのパラメータの中の一つをクエリのパラメータに追加してリクエストを繰り返すことで、閲覧するコアの識別番号である **coreid** を取得する。

表 4.3: コア検索に用いるパラメータ

クエリのパラメータ	レスポンスのパラメータ
なし	seano
seano	digsite
seano,digsite	hole
seano,digsite,hole	coreno
seano,digsite,hole,coreno	bittype
seano,digsite,hole,coreno,bittype	sectionno
seano,digsite,hole,coreno,bittype,sectionno	coreid

4.4.4 色付けを保存するためのデータモデル設計

色付けを保存するための DB 設計を示す。色と CT 値の対応付け、透過度と CT 値の対応付けをそれぞれデータベースに格納する。色と CT 値の対応は $\langle CT \text{ 値}, \text{色値} \rangle$ の組列によって表され、レンダラ API のパラメータの一つとなっている。 $\langle CT \text{ 値}, \text{色値} \rangle$ の組列の個数は、色付けを行う 3 次スプライン曲線の補間点の個数によって決定する。透過度と CT 値の対応は $\langle CT \text{ 値}, \text{透過度} \rangle$ の組列によって表され、レンダラ API のパラメータの一つとなっている。 $\langle CT \text{ 値}, \text{透過度} \rangle$ の組列の個数は透過度の設定の複雑度によって決定する。

色付けを保存するためのデータモデルとして図 4.14 に示すデータモデルを構築した。データモデルは、COLORMODEL テーブル、CTCOLOR テーブル、CTALPHA テーブルの 3 つのテーブルから構成される。

COLORMODEL テーブルは、色の対応付けとその対応付けを表す名前を格納したテーブルであり、主キーとして、MODELNUM を持ち、キーとして MODELNAME を持つ。

CTCOLOR テーブルは、色の対応付けと、その対応付けに含まれる CT 値と色の関係を格納したテーブルである。色外部キーとして COLORMODEL テーブルの主キーを、その他のキーとして CT 値を表す CTNUM と色値を表す COLORNUM が格納されている。

CTALPHA テーブルは、色の対応付けと、その対応付けに含まれる CT 値と透過度の関係を格納したテーブルである。外部キーとして COLORMODEL テーブルの主キーを、その他のキーとして CT 値を表す CTNUM と色値を表す ALPHANUM が格納されている。

色付けを行う際の補間点の数や、透過度の設定の仕方によって、CT と色の関係を表す組の数、CT と透過の関係を表す組の数が増える。そのため、COLORMODEL テーブルと CTCOLOR テーブル、CTALPHA テーブルに関係を作ることによって、カラーモデルを 1 つ選択した際に、CT 値と色の対応付け、CT 値と透過度の対応付けの保存・取得を行うことができる。

4.5 まとめ

掘削コア試料の分析に必要な機能を有したタブレット端末向けクライアントソフトウェアを開発した。クライアントソフトウェアでは、コアを閲覧するための機能を提供し、ユーザ

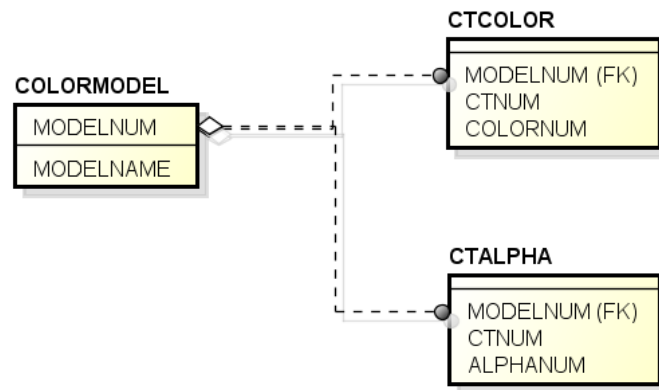


図 4.14: 色付けを保存するためのデータモデル

の操作によって，サーバ側に画像描画のリクエストを送信，描画結果のコア画像を表示する役割を持つ．

タブレット端末は画面サイズが制限されているため，分析を行うための画面を 4 つに分け，それぞれの画面をタブにより移動できる設計となっている．

従来の閲覧ソフトウェアには存在しない，CT 値に対する色付けを変更する，コアの縦切断を行うなど掘削コア試料の分析専用の機能を提供することで，専用ソフトウェアが存在しなかった問題を解決した．またドラッグやピンチなどのタッチ操作をインタフェースに活用し，掘削コア試料に対して直感的な操作を行うことが可能である．

第5章 まとめと今後の発展

本プロジェクトでは、掘削コア試料のスキャンデータを閲覧するためのクラウドサービスの開発を行った。

従来の掘削コア試料のスキャンデータを利用する際のデータサイズの大きさ、掘削コア専用のインタフェースが存在しないという課題を解決するためのシステムとして、3次元データから軽量の2次元画像を生成する画像描画サーバ、分析専用のインタフェースを持つクライアントアプリケーション **Virtual Core Viewer** を開発した。**Virtual Core Viewer** はタブレット端末向けアプリケーションと **Web** アプリケーションの2種類を開発した。

筆者は、分析専用閲覧機能を持つタブレット端末向けアプリケーション **Virtual Core Viewer** の設計および開発を担当した。タブレット端末向けアプリケーションの **Virtual Core Viewer** ではドラッグやピンチなどタッチ操作を活用したインタフェース、色付けやコア検索機能など従来の閲覧ソフトウェアには存在しない掘削コア試料を閲覧するための機能を有する。

本サービスによって、掘削コア試料のスキャンデータを利用する際に課題となっていた、データサイズの大きさや端末の性能を気にすることなく利用でき、かつ掘削コア試料の閲覧専用インタフェースでコアを分析することが可能となった。

本システムの今後の発展として、医用画像への適用、研究者間の情報共有があげられる。

本システムでは、医用の画像フォーマットである **DICOM** フォーマットの描画を対象としているため、実際の医用の画像にも適用可能である。医用画像を扱う際には個人情報問題となるが、氏名など個人情報が記録されている **DICOM** フォーマットのファイルを直接やり取りするのではなく、描画後の画像データのみのやり取りを行うので個人情報を伏せたまま閲覧することが可能である。本システムで描画した医用画像の例を図 5.1 に示す。現在は掘削コア試料専用の閲覧インタフェースであるので医用画像用のインタフェースも必要となる。

研究者間の情報共有の手助けとしては、アノテーション機能を付け **SNS** と連携させることが考えられる。アノテーション機能として、掘削コア試料の3次元データに対してメモ等のテキストやイラストを添付する機能を提供し、**Twitter** などの **SNS** と連携させることで研究者間での議論を行いやすくするなど、掘削コア試料のスキャンデータの利用価値を高めることが期待できる。

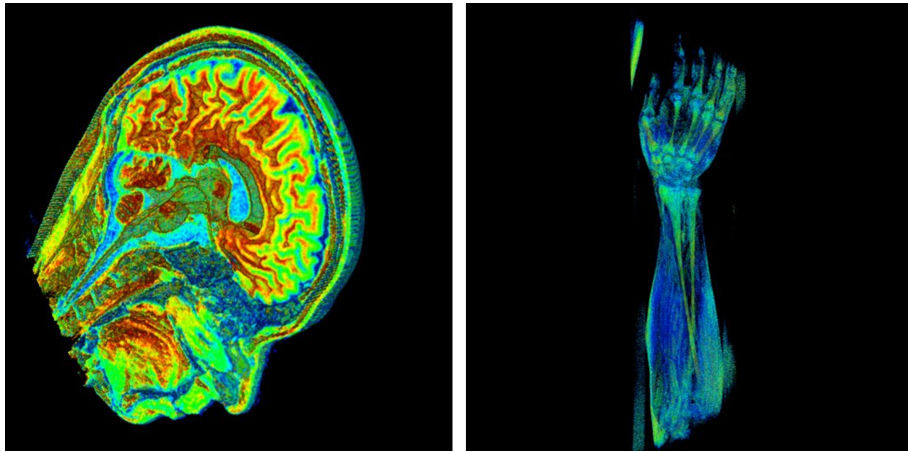


図 5.1: 医用画像への適用例

謝辞

本プロジェクトの担当教員である和田耕一教授，山際伸一准教授には，大変お世話になりました．日々のプロジェクトの指導を頂き，様々な場所での議論の機会を与えていただきました．心より感謝いたします．高知コア研究所の久光敏夫技術主任には，開発したシステムへのフィードバックや助言等様々な面でお世話になりました．深く感謝致します．

インタラクティブプログラミング研究室の，田中二郎教授，三末和男准教授，高橋伸准教授，志築文太郎准教授，**Simona Vasilache** 助教には大変お世話になりました．特に，指導教官である高橋伸准教授には，発表練習や報告書の執筆等で大変貴重な助言，ご指導をいただきました．心より感謝いたします．

高度 IT 人材育成のための実践的ソフトウェア開発専修プログラム担当の中沢研也教授，山戸昭三教授には，コース活動を通して，システム開発やプロジェクトの進め方等様々なことを教えていただきました．誠に有難うございました．

並列分散処理研究室の櫻井美知代さんには，出張の手配や書類等の作成など大変お世話になりました．

プロジェクトメンバーである岡本昂也氏，坂本侑一郎氏には，プロジェクトを進めるに当たり，様々な面でサポートを頂きました．深く感謝します．

最後に，自分を支えてくれた両親やすべての友人，大学生活でお世話になったすべての方々に感謝を申し上げます．

参考文献

- [1] IODP, <http://www.iodp.org/>
- [2] Drilling ship Chikyu returns deepest seabed core samples yet, Nature News Blog, <http://blogs.nature.com/news/2012/09/drilling-ship-chikyu-returns-deepest-seabed-core-samples-yet.html>
- [3] Virtual Core Library, <http://www.kochi-core.jp/VCL/>
- [4] DICOM Homepage, <http://medical.nema.org/>
- [5] A. Rosset, L. Spadola, and O. Ratib, Osirix: an open-source software for navigating in multidimensional dicom images, *Journal of Digital Imaging*, vol. 17, no. 3, pp. 205-216, 2004.
- [6] Amazon Elastic Compute Cloud(Amazon EC2), <http://aws.amazon.com/jp/ec2/>
- [7] Office Web Apps - Office.com, <http://office.microsoft.com/ja-jp/web-apps/>
- [8] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, Matei Zaharia, A View of Cloud Computing, *Communications of the ACM*, vol. 53 no. 4, pp. 50-58, 2010
- [9] M. Yakami, K. Ishizu, T. Kubo. T. Okada. and K. Togashi, "Development and Evaluation of a Low-Cost and High-Capacity DICOM Image Data Storage System for Research," *Journal of Digital Imaging*, Springer, vol. 24, no.2, pp. 190-195, April 2011.
- [10] F. Lamberti and A. Sanna. A streaming-based solution for remote visualization of 3D graphics on mobile devices. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):247.260, 2007.
- [11] Android, <http://www.android.com>,
- [12] JSON, <http://www.json.org>,

付 録 A プロジェクトの進行計画

本プロジェクトで提供する機能とその開発段階の対応を表 A.1 に示す。スコープ S0 が 1 段階目の開発内容、スコープ S1 が 2 段階目の開発内容、スコープ S2 が 3 段階目の開発内容を示している。1 段階の開発が終わるごとに共同研究者にサービスを提供し、フィードバックを得た。得られたフィードバックをもとに次段階の開発でシステムの改善を行った。最終段階であるスコープ S2 の開発が終了した段階で開発システムの一般公開を行った。

- スコープ S0:解析に必要な基本機能の実装
- スコープ S1:コア試料の検索、新機能の実装、UI の改善
- スコープ S2:基本機能の拡張、インタフェースの改善
(スコープ S2 の上記 2 項目は Android 端末向け Virtual Core Viewer でのみ実装)

表 A.1: スコープ

スコープ S0	スコープ S1	スコープ S2
拡大・縮小 コアの選択 色付け機能 CT 値と色の対応表示 縦方向の切断 透過	検索機能 スケールの表示 CT 値のヒストグラムの表示 UI の改善	CT 値と色の対応の保存 CT 値と色の対応の選択 UI の改善

各スコープにおける開発スケジュールの予定と実績を図 A.1 に示す。本プロジェクトでは、以下のような目標を立て、開発を進めた。

- 7 月末:第 1 回リリース スコープ S0 の機能を実装完了
- 10/4: 第 2 回リリース スコープ S1 の機能の実装完了、中間報告でデモを行う
- 12/5:最終リリース スコープ S2 の機能の実装完了、学会での成果報告

本プロジェクトは当初の予定通りに順調に開発を行い、12/1 には 3 段階の開発をすべて終え、Android 端末向け Virtual Core Viewer の公開を行った。Web アプリケーション版 Virtual Core Viewer を図 A.2 に示す。また 12/4-7 に行われた ICNC でポスター発表による成果発表を行った。

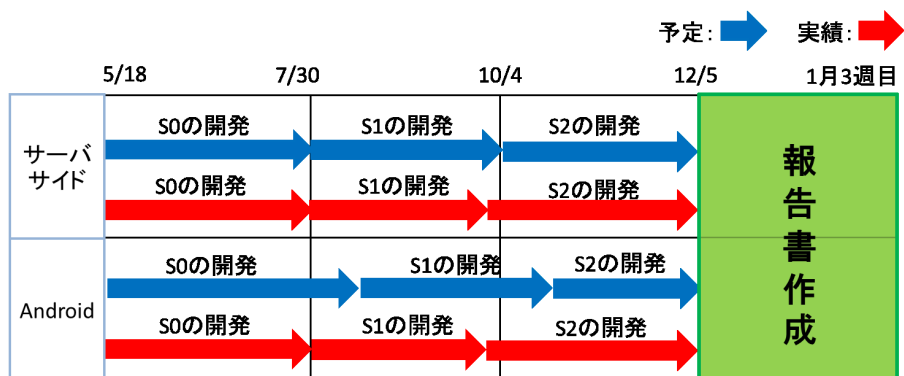


図 A.1: 開発スケジュール

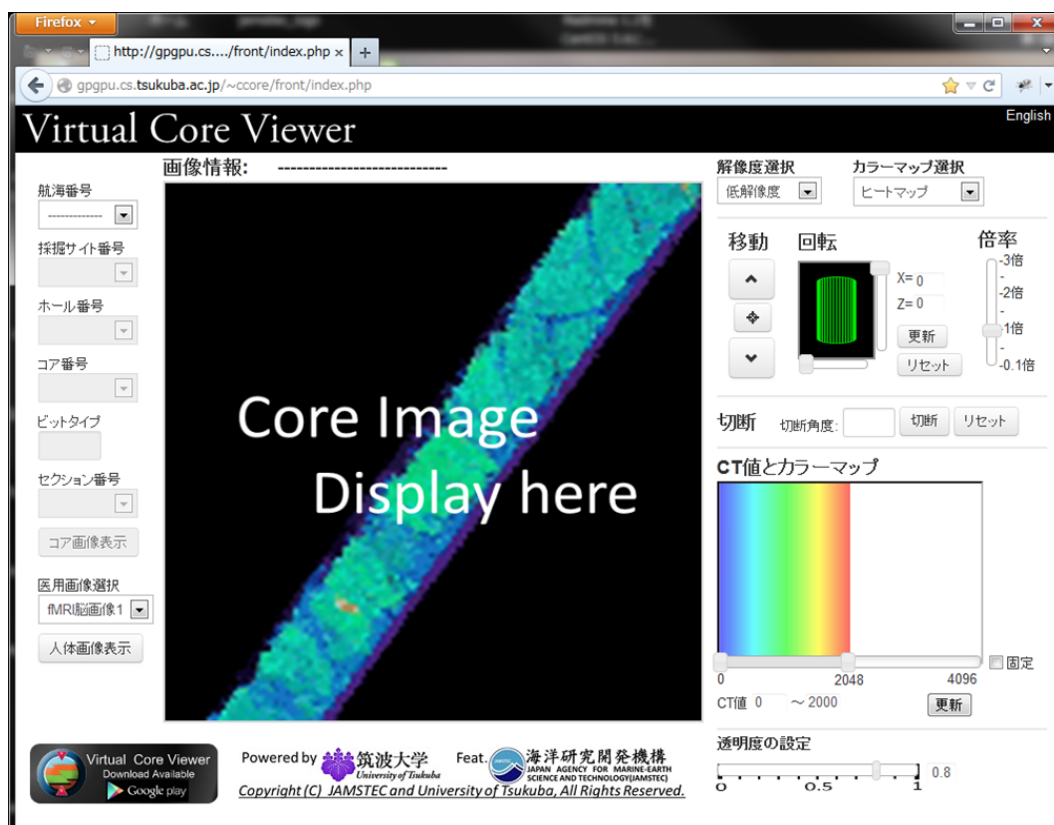


図 A.2: Web アプリケーション版 Virtual Core Viewer

付 録 B Android アプリケーション設計書

B.1 プロジェクトディレクトリ構成

アプリケーションのプロジェクトのディレクトリ構成を以下に示す。src ディレクトリ, layout ディレクトリはそれぞれ個別に記載する。layout フォルダは画面を構成する xml ファイルを保存するディレクトリである。layout-large ディレクトリ, layout-xlarge ディレクトリは画面サイズに合わせた xml を保存するディレクトリであり, xml ファイルの名前は同じであるが画面サイズに合わせた内容の xml を記載する必要がある。端末で表示する際には一番近い画面サイズの近いディレクトリから xml ファイルを自動的に読み込む。

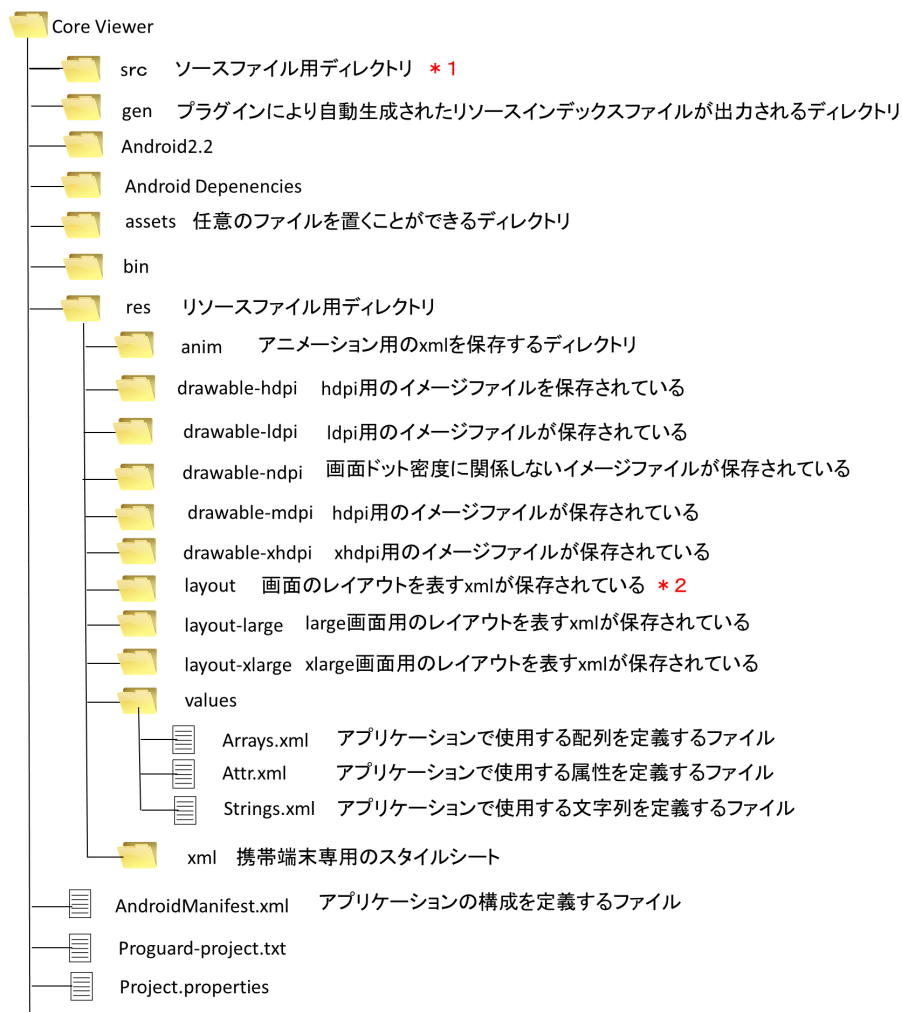


図 B.1: アプリケーションのディレクトリ構成



図 B.2: src のディレクトリ構成

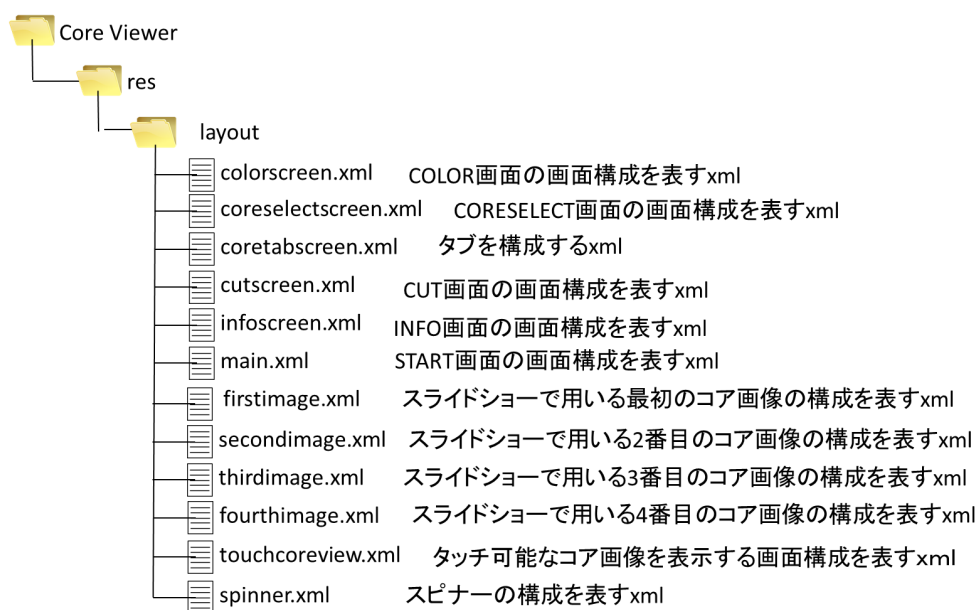


図 B.3: layout のディレクトリ構成

B.2 クラス一覧

表 B.1: 作成したクラス一覧 (1/2)

クラス名	概要
StartScreenActivity	Activity のサブクラスである。アプリケーションの起動時に最初に呼び出される。端末がネットワークにつながっているかどうかの確認を行う役割も持つ。
CoreSelectScreenActivity	Activity のサブクラスである。始めに閲覧するコアを絞り込み検索で検索する画面を表示する役割を持つ。
CoreTabActivity	TabActivity のサブクラスである。4つの画面をタブで切り替えられるようにするための役割を持つ。
WholeScreenActivity	Activity のサブクラスである。コア画像を画面全体に表示する役割を持つ。
ColorScreenActivity	Activity のサブクラスである。色付け、透過に関する画面を表示する役割を持つ。
CutScreenActivity	Activity のサブクラスである。切断に関する画面を表示する役割を持つ。
InfoScreenActivity	Activity のサブクラスである。コアの検索、解像度の変更を行う画面を表示する役割を持つ。
TouchCoreView	View のサブクラスである。掘削コア試料の画像の描画、スケールの表示をする役割を持つ。ドラッグやタッチに認識を行い、操作に応じたクエリを作成する。
ColorMapView	View のサブクラスである。CT 値と色の対応付けと、CT 値のヒストグラムの表示を行う役割を持つ。

表 B.2: 作成したクラス一覧 (2/2)

ToukaMapView	View のサブクラスである。CT 値と透過度の対応表示, CT 値のヒストグラムを表示する役割を持つ。
CutCoreView	View のサブクラスである。切断用の掘削コア試料の画像を表示する役割を持つ。
CoreGLView	GLSurfaceView のサブクラスである。OpenGL のレンダリングが可能なビューであり, タッチ操作に対応した円柱を表示させるクラスである。
CoreCylinder	CoreGLView に描画する円柱状オブジェクトの構造を定義してある。
CoreRenderer	Renderer のサブクラスである。GLSurfaceView の描画の役割を受け持つ。
CoreParams	掘削コア試料の描画のパラメータを処理, 保存するためのクラスである。
RenderHttpGet	レンダラ API を利用したサーバ間との通信を行うクラスである。
SearchHttpGetTask	AsyncTask のサブクラスである。検索 API を利用したサーバ間との送受信のやり取りを行うクラスである。バックグラウンドで動作し, ネットワークの取得中にはダイアログを表示させる。
HttpGetHandler	Handler のサブクラスである。SearchHttpGetTask のタスク完了時に呼ばれる UI のハンドラである。送受信の成功, 失敗のメソッドをオーバーライドして使用する。
SearchParams	検索 API を利用するためのデータの保存や処理を行う役割を持つ。
ColorModelDB	色の対応付け, 透過度の設定のデータベースへのアクセスを行うクラスである。削除や保存などデータベースを利用するためのメソッドを持つ。
ColorModelDBHelper	SQLiteOpenHelper のサブクラスである。データベースの作成を行うクラスである。

B.3 メソッド定義

各クラスに含まれるメソッドの説明を記載する．変数の値を取得，保存するのみの `get` メソッド，`set` メソッド，コンストラクタ，デストラクタ，は適宜省略してある．また `Activity` のライフサイクルである `onCreate` メソッド，`onResume` メソッド等も適宜省略してある．

StartScreenActivity クラス

private boolean isNetworkEnable(Context context)

引数 **Context context**

戻り値 なし

処理内容

端末がネットワークに接続されているかを判断する．

isTablet(Context context)

引数 **Context context**

戻り値 なし

処理内容

画面サイズがタブレットであるか判断する．スクリーンサイズは `context.getResources().getConfiguration().screenLayout` によって取得しする

CoreSelectScreenActivity クラス

private void setSpinner(int n)

引数 **int n** 検索で用いる項目の ID

戻り値 なし

処理内容

検索で用いる項目のスピナーに値をセットする．項目 ID によって指定されたスピナーにその項目の要素を追加する．

private void exec_get(int n)

引数 **int n** 検索で用いる項目 ID

戻り値 なし

処理内容

スピナーの値を取得するために通信処理を行う．項目 ID の値からクエリとなる URL を取得し，SearchHttpGetTask のインスタンスを作成し，バックグラウンドで処理を行わせる．SearchHttpGetTask のインスタンスの作成時には，UI ハンドラである SearchHttpGethandler の抽象メソッドを実装する．

CoreTabActivity クラス

protected void initTabs()

引数 なし

返り値 なし

処理内容

タブで表示する画面を構成する．タブで構成する情報には，タブの名称，そのタブの画面を構成する Activity を指定する．

ColorScreenActivity クラス

private void setColorModel2Screen(int modelNum)

引数 **int modelNum** カラーモデル番号

返り値 なし

処理内容

カラーモデル番号に対応した CT 値と色の対応付け，CT 値と透過度の対応付け，CT 値の最大値，最小値を取得し，画面に反映させる．

private void redrawScreen()

引数 なし

返り値 なし

処理内容

画面を再描画する．色付けを変更した場合，色付けを行う CT 値の範囲を変更した場合，透過度を変更した場合に呼び出される．

private SimpleCursorAdapter getModelList()

引数 なし

返り値 カラーモデル名のアダプター

処理内容

データベースからカラーモデル番号，カラーモデル名のリストを受け取る．その後リストから，カラーモデル名のみのアダプターに変換しアダプターを返す．アダプターへの変換はスピナーにセット可能な形式にするために行う．

private void saveColorModel(String modelName)

引数 **String modelName** カラーモデル名

返り値 なし

処理内容

引数のカラーモデル名でデータベースへ現在の CT 値と色の対応付け，CT 値と透過度の対応付けの登録を行う．

InfoScreenActivity クラス

private void setSpinner(int n)

引数 **int n** 検索で用いる項目の ID

戻り値 なし

処理内容

検索で用いる項目のスピナーに値をセットする．項目 ID によって指定されたスピナーにその項目の要素を追加する．

private void exec_get(int n)

引数 **int n** 検索で用いる項目 ID

戻り値 なし

処理内容

スピナーの値を取得するために通信処理を行う．項目 ID の値からクエリとなる URL を取得し，SearchHttpGetTask のインスタンスを作成し，バックグラウンドで処理を行わせる．SearchHttpGetTask のインスタンスの作成時には，UI ハンドラである SearchHttpGethandler の抽象メソッドを実装する．

TouchCoreView クラス

public boolean onTouchEvent(MotionEvent event)

引数 **MotionEvent event** タッチイベント

戻り値 タッチイベントがあったかどうか

処理内容

ピンチ操作かドラッグ操作があったかを判断する。ピンチ操作の終了時にはピンチ距離によってズーム率を変更させたコア画像のリクエストを行う。ドラッグ操作の終了時には、ドラッグ方向・距離によってコアの回転角度を変更させたコア画像のリクエストを行う。

private float getDragDistance(MotionEvent event)

引数 **MotionEvent event** タッチイベント

戻り値 ドラッグ開始点からの距離

処理内容

ドラッグを始めた点からの距離を計算し返すメソッド。

private float GetPinchDistance(MotionEvent event)

引数 **MotionEvent event** タッチイベント

int len

戻り値 ピンチの距離

処理内容

ピンチを行っている2点の距離を計算し返すメソッド。

ColorMapView クラス

private void drawSpline(Canvas canvas,Paint paint)

引数 **Canvas canvas** キャンバス

Paint paint ペイント

戻り値 なし

処理内容

スプライン曲線を描画するメソッド。タッチ点が一点もなかった場合は線形に直線を描く。タッチ点がある場合は、タッチ点を補間点として3次スプライン曲線を描画する。

public void setCf()

引数 なし

戻り値 なし

処理内容

CT 値と色の対応を表す配列を CoreState に保存する。

public void ct2touchPoint(int[] ct,double[] colorNum)

引数 **int[] ct** 色値に対応した CT 値の配列

double[] colorNum 色値

戻り値 なし

処理内容

CT 値と色の対応付けをスプライン曲線の補間点を描画するためのパラメータに変換する。変換後のパラメータをタッチ点として登録する。

public void touchPoint2ct()

引数 なし

戻り値 なし

処理内容

タッチ点を色付けを行う CT の範囲を基に実際の CT 値と色値に変換する。

public boolean setTouchData(int x,int y)

引数 **int x** タッチ点の x 座標

int y タッチ点の y 座標

返り値 正しく登録されたかどうか

処理内容

引数の値をタッチ点として登録する．引数がタッチ点としてふさわしくない場合，タッチ点の個数が 8 個以上の場合は **false** を返す．

public void removeTouchData()

引数 なし

返り値 なし

処理内容

登録されてあるタッチ点をすべて削除する．

ToukaMapView クラス

public void drawHeatMap()

引数 なし

戻り値 なし

処理内容

CT 値に対応するヒートマップの Bitmap を作成する。ColorMapView で設定された 3 次スプラインの曲線を基に RGB に変換し、CT 値と色に対応する Bitmap を作成する。

private void drawToukaLine(Canvas canvas,Paint paint)

引数 なし

戻り値 なし

処理内容

CT 値と色の対応を表す配列を CoreState に保存する。

private int getRgbFromHeatmap(int i)

引数 **int[i]** 0～255 の値

戻り値 色値

処理内容

引数の値をヒートマップ関数に適用し、RGB に変換した値を返す。

public void setAf()

引数 なし

戻り値 なし

処理内容

CT 値と透過度の対応を表す配列を CoreState に保存する。

public void ct2map(int[] ct,double[] alpha)

引数 **int[] ct** 色値に対応した CT 値の配列

double[] alpha 透過度

戻り値 なし

処理内容

CT 値と透過度の対応付けを透過度を表す線を描画するためのパラメータに変換する.

public void map2ct()

引数 なし

戻り値 なし

処理内容

透過度を表す線で表現されているパラメータを透過度を実際の CT 値と透過度の値に変換する.

CoreGLView クラス

public CoreGLView(Context context)

引数 Context context

戻り値 なし

処理内容

コンストラクタ. CoreRenderer のインスタンスを作成する.

CoreCylinder クラス

CoreCylinder(float d, float s)

引数 **float d** 半径のピクセル数

float s 高さのピクセル数

戻り値 なし

処理内容

半径 d, 高さ s の円柱を表すデータ構造を作成する.

public void display(GL10 gl)

引数 **GL10 gl**

戻り値 なし

処理内容

CoreCylinder コンストラクタによって作成されたデータ構造に対して, 色付けを行う.

CoreRenderer クラス

public void onDrawFrame(GL10 gl)

引数 **GL10 gl**

戻り値 なし

処理内容

画面の描画を行う。視点方向に応じて CoreCylinder を回転させた描画を行う。

public void onSurfaceChanged(GL10 gl, int width, int height)

引数 **GL10 gl**

int width 端末のウィンドウの横のピクセル数

int height 端末のウィンドウの縦のピクセル数

戻り値 なし

処理内容

画面が変化したときの処理を行う。

public void onSurfaceCreated(GL10 gl, EGLConfig config)

引数 **GL10 gl**

EGLConfig config

戻り値 なし

処理内容

画面が作成時の処理を行う。

CoreParams クラス

private void buildQueryString()

引数 なし

戻り値 なし

処理内容

レンダラ API で使用する TouchCoreView 用のコア画像取得用のクエリの URI を作成する.

private void buildCutQueryString()

引数 なし

戻り値 なし

処理内容

レンダラ API で使用する CutCoreView 用のコア画像取得用のクエリの URI を作成する.

private void resetCoreParam()

引数 なし

戻り値 なし

処理内容

レンダラ API で使用するパラメータの値をリセットする.

public void zoom(double zoom)

引数 **double zoom** ズーム率

戻り値 なし

処理内容

ズーム率の値をセットする. ズーム率が適切であるかどうかの判断もこのメソッド内で行う.

public void rotate(float x,float y,float z)

引数 **int x** 回転成分 x

int y 回転成分 y

int z 回転成分 z

戻り値 なし

処理内容

引数の値を基にコアの回転角度を計算する.

public void drawHistogramMap()

引数 なし

戻り値 なし

処理内容

ヒストグラム図を作成する.

RendererHttpGet クラス

public boolean requestCore()

引数 なし

返り値 コア画像が取得できたかどうか

処理内容

TouchCoreView 用のコア画像を取得するためのメソッド。まず JSON 形式のデータを取得するためのリクエストを行い、コア画像を取得するためのリクエストを行う。

public boolean requestCutCore()

引数 なし

返り値 コア画像が取得できたかどうか

処理内容

CutCoreView 用のコア画像を取得するためのメソッド。まず JSON 形式のデータを取得するためのリクエストを行い、コア画像を取得するためのリクエストを行う。

private void initBasicRequest()

引数 なし

返り値 なし

処理内容

HTTP Get に関するパラメータを設定するメソッド。接続時間のタイムアウト、データ取得のタイムアウト、Basic 認証のユーザ名、パスワードについての設定する。

private boolean getRequestedJson(String uri)

引数 **String uri** レンダラ API で使用する URI

返り値 データが正常に取得できたかどうか

処理内容

引数 uri が示す URL で JSON 形式のデータを取得するリクエストを行う。

private boolean getRequestedBitmap(String url)

引数 **String url** 画像の位置を表す URI

戻り値 画像が正常に取得できたかどうか

処理内容

引数 `uri` が示す URL にある画像を取得する。

private void parseJson(String json)

引数 **String json** JSON 形式のデータ

戻り値 なし

処理内容

引数で表される JSON 形式のデータの解析を行う。JSON 形式の中の画像の URL、スケール、CT 値のヒストグラムを表す配列などの項目を取得し、`CoreState` クラスに格納する。

SearchHttpGetTask クラス

private void initBasicRequest()

引数 なし

返り値 なし

処理内容

HTTP GET に関する情報を定義する。接続のタイムアウト時間、データ取得のタイムアウト時間、ユーザ ID、パスワードなどの設定を行う。

protected void onPreExecute()

引数 なし

返り値 なし

処理内容

タスクの開始時に呼び出される。通信中ダイアログを表示させる、HTTP のレスポンスを解析して、正常に受信できているかいないかを判断し、その結果に応じたメッセージを作成する。

protected Void doInBackground(Void... unused)

引数 なし

返り値 なし

処理内容

バックグラウンドで実行する処理を記述する。HTTPGET を行う URL の確認、GET リクエストを行う。

protected void onPostExecute(Void unused)

引数 なし

返り値 なし

処理内容

タスク終了時に呼び出される。通信中ダイアログを消す、受信結果を渡すためにメッセージを作成する。

SearchHttpGetHandler クラス

public abstract void onGetCompleted(String response)

引数 **String response** 通信成功時の内容を表す文字列

戻り値 なし

処理内容

通信成功時の処理を記述する。抽象メソッドであるのでインスタンスを生成するときにその処理内容を記述する。

public abstract void onGetFailed(String response)

引数 **String response** 通信失敗時のエラー内容の文字列

戻り値 なし

処理内容

通信失敗時の処理を記述する。抽象メソッドであるのでインスタンスを生成するときにその処理内容を記述する。

SearchParams クラス

public String getQuery(int n)

引数 **int n** 検索で用いる項目の ID

戻り値 URL の文字列

処理内容

コアを検索するために必要なクエリの URL を返す。引数の値によって、検索の URL に追加するパラメータを変化させる。

public boolean parseJson(String json)

引数 **String json** JSON 形式のデータ

戻り値 正常に終了したかどうか

処理内容

JSON 形式のデータをパースして、検索で用いるための識別番号を保存する。

ColorModelDB クラス

public void setDefaultColorModel()

引数 なし

返回值 なし

処理内容

デフォルトで呼び出すことのできるカラーモデルをデータベースに保存する．初回起動時のみ呼び出す．

private int nextColorModelNum()

引数 なし

返回值 カラーモデル番号

処理内容

現在のデータベースに保存されているカラーモデル番号から，次に登録するカラーモデル番号を求めるメソッド．カラーモデルを追加するときに呼び出される．

public void insertColorModel(int modelNum,String modelName,int[] colorCt,double[] colorNum, int[] alphaCt,double[] alpha)

引数 **int modelNum** カラーモデル番号

String modelName カラーモデル名

int[] colorCt 色値に対応する CT 値の配列

double[] colorNum 色値の配列

int[] alphaCt 透過度に対応する CT 値の配列

int[] alphaNum 透過度の配列

返回值 なし

処理内容

データベースにカラーモデルを追加するメソッド．insertColorModelTable メソッド，insertCtColorTable メソッド，insertCtAlphaTable メソッドを用いてテーブルに保存する．

private void insertColorModelTable(int modelNum,String modelName)

引数 **modelNum** カラーモデル番号
 modelName カラーモデル名

戻り値 なし

処理内容

データベースのカラーモデル番号とカラーモデル名を保存するテーブルにカラーモデル番号, カラーモデル名を保存する.

private void insertCtColorTable(int modelNum,int[] colorCt,double[] colorNum)

引数 **int modelNum** カラーモデル番号
 int[] alphaCt 色値に対応する CT 値の配列
 double[] alphaNum 色値の配列

戻り値 なし

処理内容

データベースの CT 値と色に対応付けを保存するテーブルにカラーモデル番号, CT 値と透過度を保存する.

private void insertCtAlphaTable(int modelNum,int[] alphaCt,double[] alphaNum)

引数 **int modelNum** カラーモデル番号
 int[] alphaCt 透過度に対応する CT 値の配列
 double[] alphaNum 透過度の配列

戻り値 なし

処理内容

データベースの CT 値と透過度に対応付けを保存するテーブルにカラーモデル番号, CT 値と透過度を保存する.

public void deleteColorModel(int modelNum)

引数 **int modelNum** カラーモデル番号

戻り値 なし

処理内容

カラーモデル番号に対応したのカラーモデルをデータベースから削除する.

public Cursor getModelListCursor()

引数 なし

返回值 カラーモデルの番号とその名前が含まれた Cursor

処理内容

カラーモデルのリストをデータベースから取得し、カラーモデル番号とカラーモデル名が含まれている Cursor を返す.

public void reloadColorModel(int modelNum)

引数 **int modelNum** カラーモデル番号

返回值 なし

処理内容

カラーモデル番号に対応した CT 値と色の対応付け, CT 値と透過度の対応付けを取得するためのメソッド. getCtColor メソッド, getCtAlpha メソッドを呼び出す.

private void getCtColor(int modelNum)

引数 **int modelNum** カラーモデル番号

返回值 なし

処理内容

カラーモデル番号に対応する CT 値と色値の対応をデータベースから取得する. 色付けを行う CT 値の最大値, CT 値の最少値の計算も行い保存する.

private void getCtAlpha(int modelNum)

引数 **int modelNum** カラーモデルの番号

返回值 なし

処理内容

カラーモデルの番号に対応する CT 値と透過度の対応をデータベースから取得する.

ColorModelDBHelper クラス

public void onCreate(SQLiteDatabase db)

引数 SQLiteDatabase db データベース

戻り値 なし

処理内容

テーブルの作成を行う。このメソッドはデータベースを作成したタイミングで呼び出される。

public void onUpgrade(SQLiteDatabase arg0, int arg1, int arg2)

引数 SQLiteDatabase arg0

int arg1

int arg2

戻り値 なし

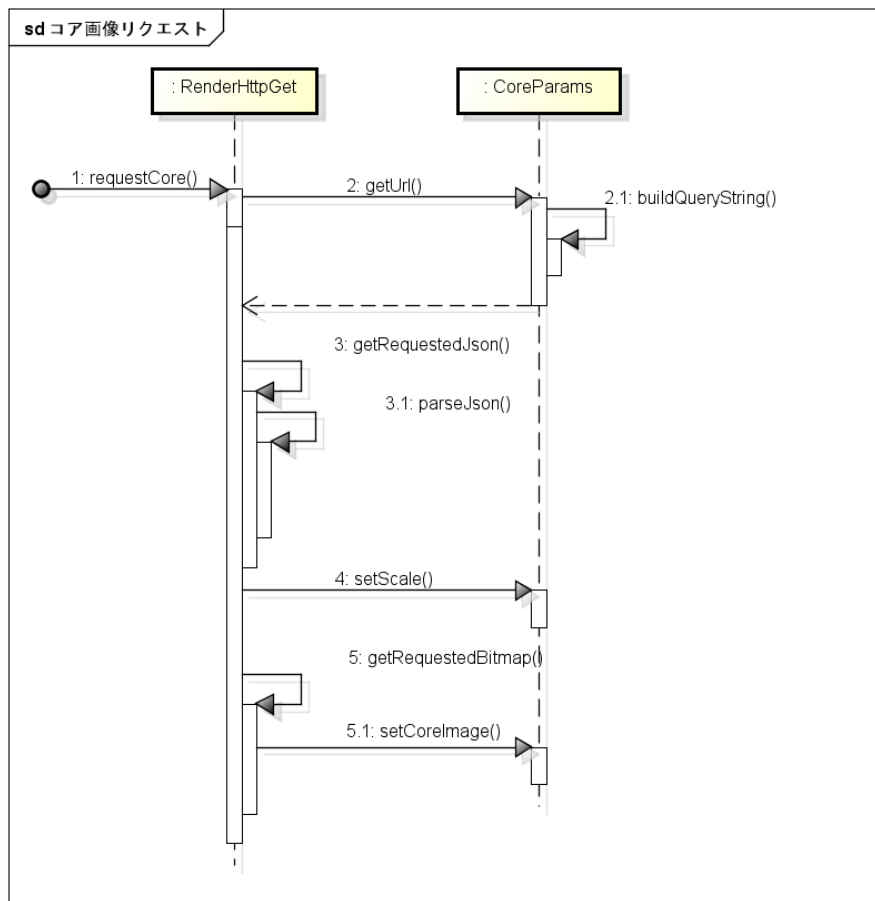
処理内容

処理なし、SQLiteOpenHelper は抽象クラスのためオーバーライドしてある。

B.4 シーケンス図

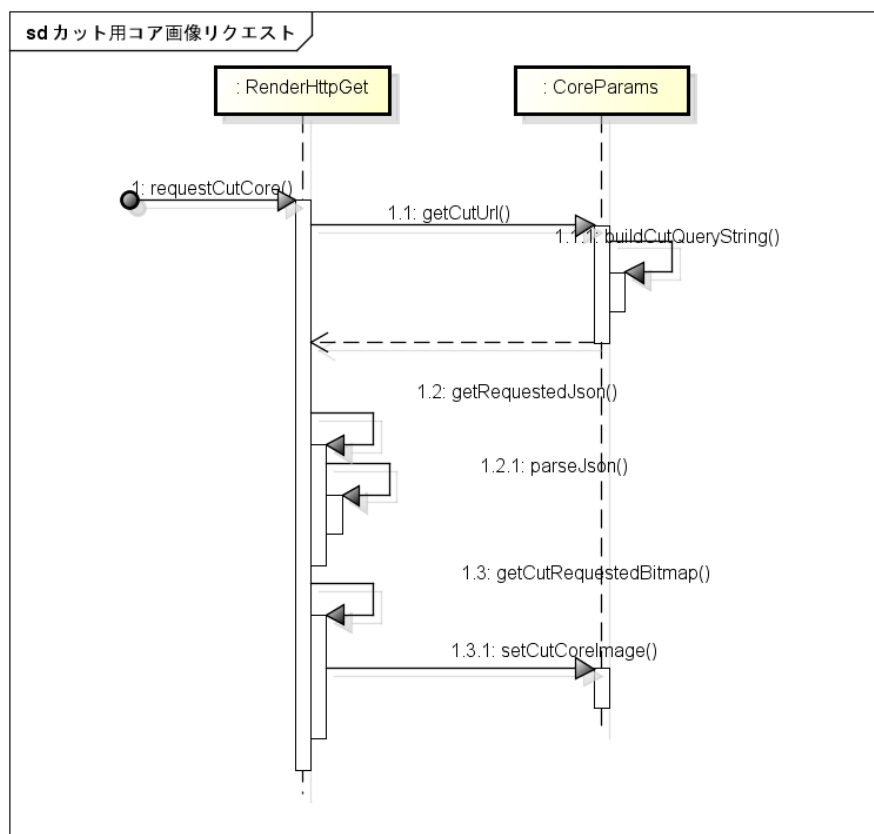
コア画像リクエスト

コア画像を取得するリクエストのシーケンス図を示す.



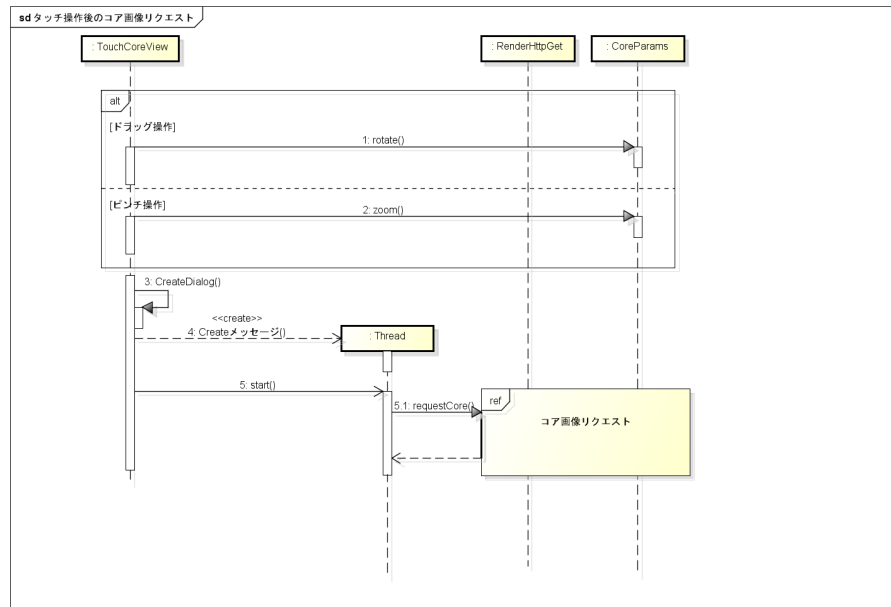
カット用コア画像リクエスト

カット用のコア画像を取得するリクエストのシーケンス図を示す.



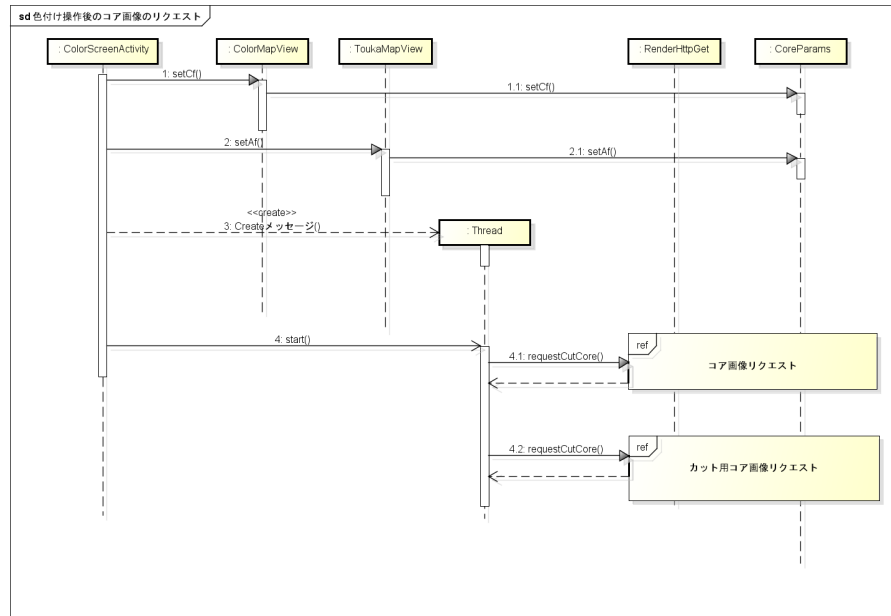
タッチ操作後のコア画像リクエストのシーケンス図

タッチ操作をおこなったコア画像のリクエストのシーケンス図を示す.



色付け後のコア画像リクエストのシーケンス図

色付けを変更させたコア画像のリクエストのシーケンス図を示す.



切断後のコア画像リクエストのシーケンス図

切断を行った際のコア画像リクエストのシーケンス図を示す.

