



数値地盤モデル構築のためのボーリング柱状図からのデータ自動抽出

木崎, 崇雄
竹山, 智英
橘, 伸也
高山, 裕介

(Citation)

神戸大学都市安全研究センター研究報告, 29:13-22

(Issue Date)

2025-03

(Resource Type)

departmental bulletin paper

(Version)

Version of Record

(JaLCD0I)

<https://doi.org/10.24546/0100496646>

(URL)

<https://hdl.handle.net/20.500.14094/0100496646>



数値地盤モデル構築のためのボーリング柱状図からの データ自動抽出

Automated data extraction from borehole logs for the development of numerical
ground models

木崎 崇雄¹⁾

Takao Kizaki

竹山 智英²⁾

Tomohide Takeyama

橘 伸也³⁾

Shinya Tachibana

高山 裕介⁴⁾

Yusuke Takayama

概要:近年、都市を仮想空間上に再現し、防災計画や都市計画に活用する「都市デジタルツイン」が注目されている。その構築には地盤情報が不可欠であり、特にボーリングデータを基にした三次元地盤モデルの作成が重要となる。通常、Kunijiban¹⁾のボーリングデータは PDF 形式の柱状図とその XML 形式のデータがセットになっている。しかし、日本の海岸沿いでは XML 形式のデータが整備されておらず、モデル構築の障害となっている。そこで本研究では、PDF 形式のボーリング柱状図から地盤情報を抽出し、XML 形式へ変換する手法を開発した。国土交通データプラットフォーム²⁾が公開している国土交通データプラットフォーム API³⁾を活用し、Python による PDF 解析を用いて柱状図内の地層情報や N 値を抽出、ベクターデータの解析を通じて高精度なデータ取得を実現した。得られたデータを XML に変換し、DPP を用いて地盤モデル構築のテストを実施した結果、高精度な変換が可能であり、既存データの再利用が促進されることを確認した。本研究は都市デジタルツインの基盤となる地盤情報の整備に貢献する。

キーワード:ボーリング柱状図、PDF、XML、DPP、地盤モデル

1. はじめに

近年、都市デジタルツインが注目され、防災計画や都市計画の精度向上が期待されている。その構築には、地上だけでなく地下の詳細な情報が必要であり、特に地盤の物理的特性を正確に把握することが重要である。ボーリングデータは地盤モデルの構築に不可欠だが、日本全国の海岸沿いのデータの多くは PDF 形式の柱状図のみで提供され、XML 化されていないため、データ活用が困難となっている。この問題を解決するため、本研究では PDF 形式のボーリング柱状図から地層情報や N 値を自動抽出し、XML 形式へ変換する手法を開発する。具体的には、国土交通データプラットフォーム API を用いたデータ収集と、Python による PDF 解析技術を組み合わせ、情報抽出と変換の自動化を実現する。本研究により、既存のボーリングデータの再利用を促進し、都市デジタルツインのための地盤情報整備に貢献することを目指す。

2. データ抽出手法

(1) 国土交通データプラットフォームを活用したデータ収集

沿岸地域以外の通常の柱状図にはそのボーリングデータの緯度経度や孔口標高などの地盤モデル構築に必須のデータが記載されているが、本研究が対象とする沿岸地域のボーリング柱状図にはそれらのデータが記載されていない。不足データを補うため、国土交通データプラットフォーム API を利用してボーリングデータの属性情報を取得する。国土交通データプラットフォームは日本の国土に関するデータを検索できるサイトである。このサイトがプログラム上でもデータを取得できるように機能を API 化しておりそれを活用する。API は国土交通データプラットフォームに登録されている各種データに対して「プログラムを通じた検索・取得」を可能にする仕組みである。API を活用しない場合、国土交通データプラットフォームのサイト上で手動検索し、対象のボーリングデータの属性情報を 1 つずつ取得する作業が必要となる。これに対し、API を利用することで、特定の地域を指定し、XML が未整備になっているボーリングデータの属性情報を自動的に取得することが可能となる(図 1)。国土交通データプラットフォームは API を介して Kunijiban のデータベースにあるボーリングデータを取得する(図 2)。

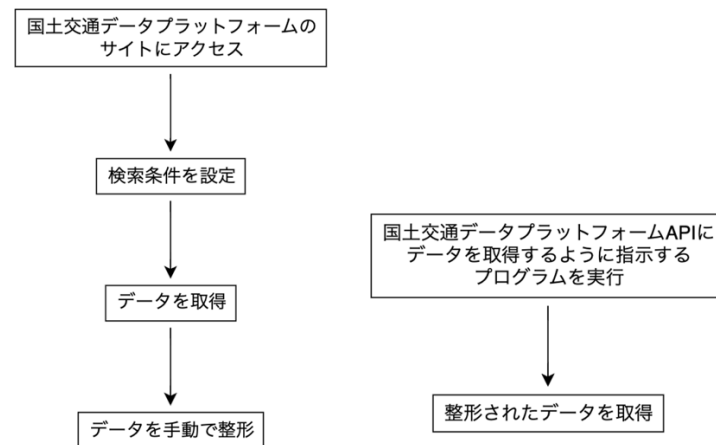


図 1 データ取得の工数の違い

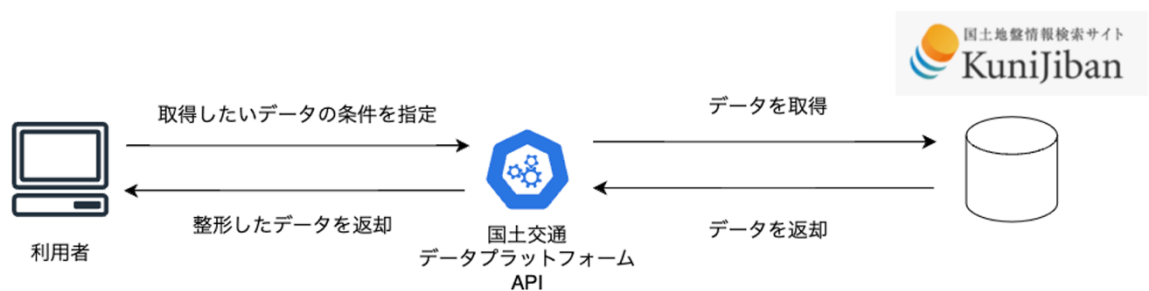


図 2 国土交通データプラットフォーム API を使用したデータ取得の流れ

また、国土交通データプラットフォーム API は GraphQL⁴⁾という仕組みを採用している。GraphQL はデータ取得を行うためのクエリ言語である。クエリは API 利用者がどのようなデータを取得したいかをリクエストするための文法構造である。以下の例では、地点 1(緯度が 34.697474、経度が 135.295467)と地点 2(緯度が 34.676735、経度が 135.295467)の範囲にあるボーリングデータの属性情報を取得するクエリである(図 3)。

```

1 query search {
2   search(
3     locationFilter: {
4       rectangle: {
5         topLeft: {
6           lat: 34.697474,
7           lon: 135.248610
8         },
9         bottomRight: {
10          lat: 34.676735,
11          lon: 135.295467
12        }
13      }
14    }
15    first: 0,
16    size: 100
17  ) {
18    totalNumber
19    searchResults {
20      id
21      title
22      dataset_id
23      metadata
24    }
25  }
26 }
27

```

図 3 GraphQL のクエリ

このように、取得したいデータの条件をクエリに記し、データ取得プログラムに組み込むことで不足分の属性情報を取得することができる。図 4 は実際にクエリを実行して得られたデータの例である。

```

{
  "id": "31f8cf69-c94d-4ee9-a7ee-518cbe5bf0fd",
  "title": "神戸六甲アイランド",
  "dataset_id": "ngi",
  "metadata": {
    "NGI:id": "123174",
    "DPF:year": "2018",
    "NGI:code": "B4KJ201801001-1423",
    "DPF:title": "神戸六甲アイランド",
    "NGI:status": "1",
    "NGI:address": "兵庫県神戸市東灘区向洋町東四丁目",
    "NGI:created": "2018-12-31T23:59:59+09:00",
    "NGI:updated": "2018-12-31T23:59:59+09:00",
    "DPF:latitude": "34.68855574",
    "NGI:latitude": "34.68855574",
    "NGI:released": "2018-12-31",
    "DPF:longitude": "135.2823893",
    "NGI:longitude": "135.2823893",
    "NGI:client_code": "600",
    "NGI:client_name": "国土交通省近畿地方整備局神戸港湾空港技術調査事務所",
    "NGI:source_code": "PORT28001010028",
    "NGI:source_name": "Kunijiban(港湾)",
    "NGI:survey_name": "神戸六甲アイランド",
    "DPF:downloadURLs": [
      "http://www.sdb.pari.go.jp/data/doseizu/2800101/PORT28001010028.pdf",
      "http://www.sdb.pari.go.jp/data/Ichiran/2800101/PORT28001010028.pdf"
    ],
    "NGI:project_name": "港湾局",
    "NGI:boring_length": "28.5",
    "NGI:survey_finish": "1981-06-01",
    "NGI:rocksoil_names": "{}",
    "NGI:soiltest_codes": "{}",
    "NGI:soiltest_names": "{}",
    "DPF:prefecture_code": [
      28
    ],
    "NGI:boring_view_url": "http://www.sdb.pari.go.jp/data/doseizu/2800101/PORT28001010028.pdf",
    "NGI:client_class_id": 1,
    "NGI:boring_elevation": "-12.608",
    "NGI:link_boring_view": "http://www.sdb.pari.go.jp/data/doseizu/2800101/PORT28001010028.pdf",
    "NGI:rocksoil_symbols": "{}",
    "DPF:municipality_code": [
      280003
    ],
    "DPF:year_originalform": "YMDHms",
    "NGI:municipality_code": "28000",
    "NGI:soiltest_view_url": "http://www.sdb.pari.go.jp/data/Ichiran/2800101/PORT28001010028.pdf",
    "NGI:link_soiltest_view": "http://www.sdb.pari.go.jp/data/Ichiran/2800101/PORT28001010028.pdf",
    "DPF:completion_datetime": "1981-06-01T00:00+09:00",
    "DPF:last_update_datetime": "2018-12-31T23:59:59+09:00",
    "DPF:completion_datetime_originalform": "YM",
    "DPF:last_update_datetime_originalform": "YMDHms",
    "DPF:address": "兵庫県神戸市東灘区向洋町東四丁目",
    "DPF:dataset_id": "ngi",
    "DPF:catalog_id": "ngi",
    "DPF:dpf_update_date": "2024-01-30T13:54:44.450Z",
    "DPF:municipality_name": [],
    "DPF:prefecture_name": [
      "兵庫県"
    ]
  }
}

```

図 4 クエリの実行結果

ここで取得したデータと、PDF 解析による柱状図から直接取得したデータを合体させることで、地盤モデル構築に必要なデータを完成させる。

(2) PDF 解析によるデータ抽出

(1)では PDF の柱状図には記載されていない重要なデータを補うために国土交通データプラットフォーム API を用いて不足分のデータを収集した。ここでは、直接 PDF の柱状図から地盤モデル構築に必要な地盤情報を自動抽出する。

a) PDF の構造

PDF はベクターデータとラスターデータの 2 種類のデータを含む。ボーリング柱状図の PDF はベクターデータとして保存されている。ラスターデータは画像として埋め込まれているのが特徴である。ベクターデータは文字や図形が個別のオブジェクトとして埋め込まれている(図 5)。またそれぞれのオブジェクトが座標のデータも持っている。ラスターデータの場合、データを抽出する場合には OCR 解析(画像認識を用いた文字情報の抽出)が必要となる。本研究では、このベクターデータの特徴を活かし、PDF 内部の文字情報や線分情報を直接解析することで、OCR 解析を用いずに高精度なデータ抽出を実現する。

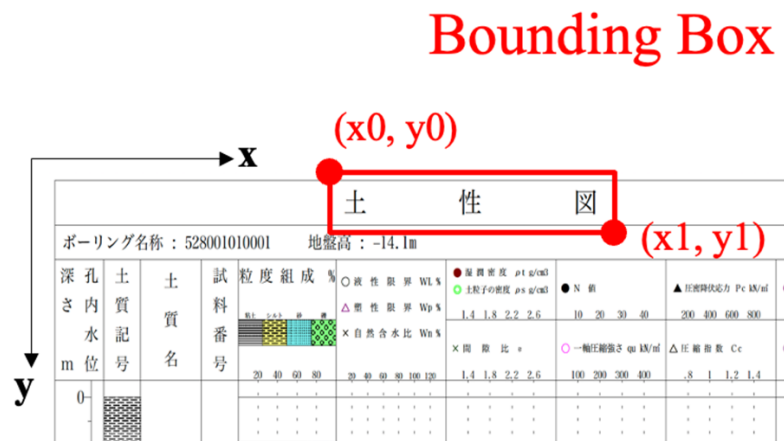


図 5 ベクターデータのイメージ

b) 解析の方針

本研究では、Python の PDF 解析ライブラリを使用することで、柱状図内のデータの取得を試みる。また、柱状図から抽出するデータは主に土質区分と N 値のデータである。DPP を用いて地盤モデルを構築する際、必須となるデータはボーリングデータの緯度経度、孔口標高、総掘進長、土質区分の深度と土質名、N 値の値とその深度である。緯度経度、孔口標高、総掘進長は柱状図に記載されていないため、(1)の国土交通データプラットフォーム API で補っている。よって、ここでは土質区分と N 値のデータの取得を目指す。また、図 6 は Python を用いて柱状図に埋め込まれているオブジェクトを可視化した結果である。赤枠はテキストデータを取得しており、青線は線などの図形を取得している。

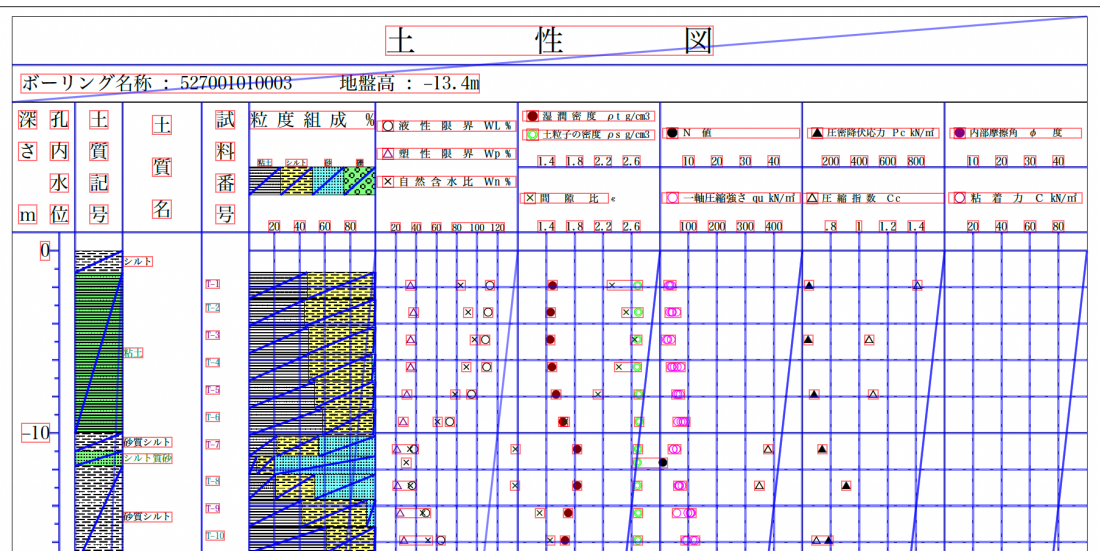


図 6 PDF のオブジェクトを可視化

c) 土質区分の抽出

柱状図には深度ごとの地層情報が記載されている。これを自動抽出するために次の処理を行なった。1 つ目は土質名のテキストを取得した。図 6 のようにテキストは **bounding-box** で取得することができた。2 つ目は深度範囲の算出である。柱状図には、y 軸方向に深度が記載されている。この目盛りも線オブジェクトとして取得できており、目盛りの線の交点座標を活用して目盛りの座標を取得する。また、土質記号は図形のオブジェクトとして抽出できており、図形オブジェクトは対角線に線オブジェクトが走るようになっている。土質記号の両脇の縦線と対角線の交点座標をもとにその土質記号の座標範囲を特定する。目盛りの座標と土質記号の範囲座標をもとに数値として深度を算出した。そして、テキストの **bounding-box** の座標情報から、土質名と深度のデータを紐付けた。

d) N 値の抽出

N 値は柱状図の右側に●マークとして記載されている。このマークは図 6 にもあるように、1 文字のテキストデータとして抽出できている。この 1 文字の **bounding-box** から座標を取得し、c)の深度算出と同じ方法で N 値を割り出す。N 値は x 軸方向に目盛りがあるためこれらの座標を取得し、●の座標から N 値を算出した。深度との対応づけに関しても c)と同じように、●マークの y 座標と目盛りの座標から深度を算出した。

(3) XML 変換

(2)の c)、d)で抽出した土質区分や N 値のデータを地盤モデルの構築に用いるためには、他のシステムが読み取りやすい形式へ整形する必要がある。本研究では、国土交通データプラットフォームで取得したボーリングデータの属性情報を組み合わせ、最終的に XML として保存する。これにより、既存の DPP による地盤構築プログラムが参照可能なデータフォーマットを実現する。Python に XML 関連のライブラリがいくつかあるのでそれを活用する。これまでに取得したデータを合体させ、XML のライブラリを活用することで図 7 のように、XML データを作成した。これにより、PDF から XML への自動変換を実現した。


```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ボーリング情報 SYSTEM "BED0210.DTD">
<ボーリング情報 DTD_version="2.10">
  <標題情報>
    <調査基本情報>
      <事業工事名>神戸六甲アイランド</事業工事名>
      <調査名>神戸六甲アイランド</調査名>
      ...
    </調査基本情報>
    <経度緯度情報>
      <経度_度>135</経度_度>
      <経度_分>15</経度_分>
      <経度_秒>11.61</経度_秒>
      <緯度_度>34</緯度_度>
      <緯度_分>41</緯度_分>
      <緯度_秒>13.96</緯度_秒>
      <測地系>WGS84</測地系>
    </経度緯度情報>
    <調査位置>
      ...
    </調査位置>
    <発注機関>
      <発注機関名称>国土交通省近畿地方整備局神戸港湾空港技術調査事務所</発注機関名称>
      <テクリスコード>
    </発注機関>
    <調査期間>
      ...
    </調査期間>
    <調査会社>

```

図 7 生成された XML の例

(4) 抽出データの精度検証

ここでは、(2)で取り組んだ柱状図からのデータ抽出のプログラムが単に動作するだけでなく、土質区分や N 値の情報をどの程度正確に取得できるかを検証する。もし抽出データに大きな誤差が含まれると、DPP などを用いた地盤解析において正確な結果が得られず、誤った設計や判断につながる恐れがある。そこで、複数のデータを用いて抽出プログラムの精度がどの程度担保されているかを検証した。

今回対象としたエリアは兵庫県神戸市の六甲アイランドのボーリングデータである。11 個をランダムに選定し、評価データとした。また、正解データに関して、本研究では人手で正解データを作成し、抽出データと比較するようにした。沿岸地域のボーリングデータは PDF の柱状図しか存在しないため、正確な正解データが存在しない。そこで、柱状図から目視で測定してデータを秋起こしてそれを正解データとして作成した。正確性を担保するために二人体制で測定を行った。

本研究で重視する評価指標は 3 つである。1 つは層数・測点数の合致である。土質区分や N 値のデータが合致しているのかどうかを確認する前に、そもそも正解データと抽出結果で土質区分の層数や N 値の測点数が一致しているを確認する。2 つ目は土質区分の正解率である。土質名が完全に一致し、かつ層の上端・下端深度が $\pm 0.1\text{m}$ の許容範囲内にある場合を正解とみなし、その割合を正解率として算出する。3 つ目は N 値の正解率である。土質区分の深度と同じように、N 値の測定深度が $\pm 0.1\text{m}$ の範囲内、かつ N 値自体が ± 1 以内であれば正解とみなし、その割合を正解率として求める。以下はここで実施した精度検証の結果である(表 1)。

表 1 検証結果

ファイル名	土質区分の 層数	土質区分の 正解率	N 値の 測点数	N 値の 正解率
PORT28001010046.pdf	4/4	100	3/3	100
PORT28001010050.pdf	5/5	100	5/5	100
PORT28001010084.pdf	32/32	100	42/49	85.7
PORT28001010051.pdf	5/5	100	3/3	100
PORT28001010047.pdf	6/6	100	4/4	100
PORT28001010021.pdf	10/10	100	8/8	100
PORT28001010077.pdf	6/6	100	7/7	100
PORT28001010019.pdf	15/15	100	13/13	100
PORT28001010048.pdf	9/9	100	3/3	100
PORT28001010049.pdf	6/6	100	4/4	100
PORT28001010045.pdf	19/19	100	14/17	82.4
合計	117/117	100	106/116	91.4

表 1 からわかるように、層数や測点数などデータ数はすべてのファイルで完全一致していた。また、11 ファイルのうち 9 ファイルでは土質区分・N 値ともに抽出結果が 100% 正解となり、深度や数値の誤差も許容範囲内であった。また、全体の集計を見ても土質区分の正解率は 100%、N 値は 91.4% と高い精度で抽出できている。この結果から、作成した抽出プログラムは高精度かつ、汎用性も高く柱状図からデータを抽出できることがわかった。

(5) 一連の処理を自動化

これまでの成果から、単一の PDF を対象として PDF から XML へ変換する手法は確立できた。しかし、XML が未整備のデータは多数存在する。そこで、図 8 のように指定した範囲に含まれるすべての柱状図に対して、XML を自動生成できるよう、これまで作成した各種プログラムを統合し、一括処理を可能とするプログラムを構築した。

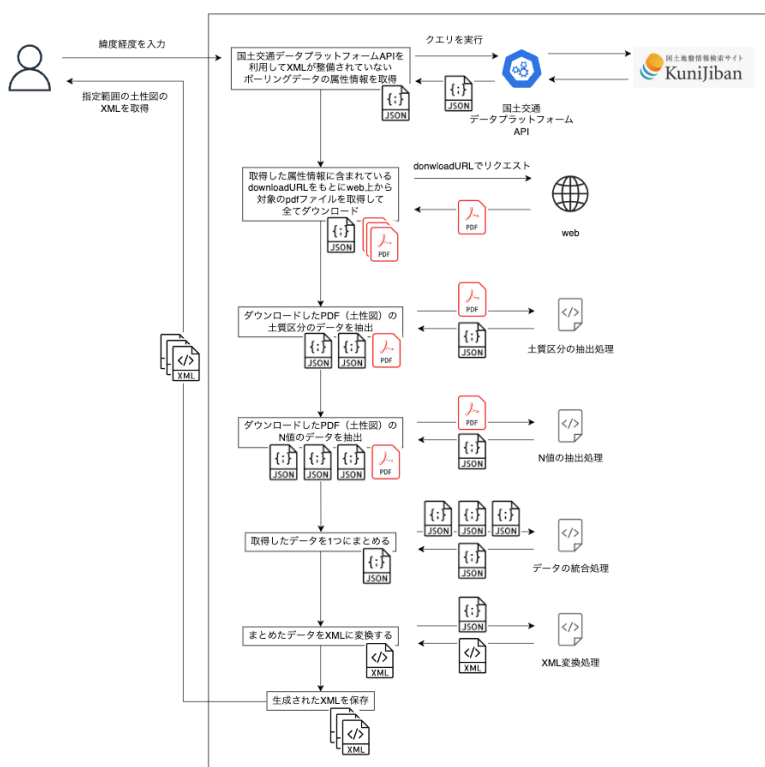


図 8 PDF から XML に変換する処理の構成図

ユーザーは、取得したい範囲の緯度経度を指定するだけで、その範囲にあるすべての XML が未整備のボーリングデータの XML を自動で取得することができる。これにより、作業時間を 1/10 に削減することができた。

3. 数値地盤モデルへの適用

ここでは、これまでに構築してきた「PDF から XML への変換フロー」を実際の数値地盤モデルに適用する方法と、その結果の検証を行う。生成された XML データを DPP で読み込み、地盤モデルを構築して可視化や精度の観点から評価を試みる。目的は、本研究で開発したデータ抽出手法が時際の地盤モデルの構築に有用であることを示すとともに、適用過程で生じる課題や改善点を明らかにすることである。これにより、XML が整備されていないボーリングデータにおいても、生成した XML データを活用し、地盤解析へ容易に接続できる可能性を示す。

Takeyamaら⁵⁾が開発した DPP による地盤モデル生成のプログラムを使用して、三次元の地盤モデルを構築する。対象エリアはこれまでと同じ六甲アイランドとし、六甲アイランド全体の地盤モデルを作成することにする。六甲アイランド全体からランダムにボーリングデータを集めることにした。図 9、図 10 に生成された地盤モデルを示す。凡例としては、白が砂、青が粘土、赤が礫を表している。

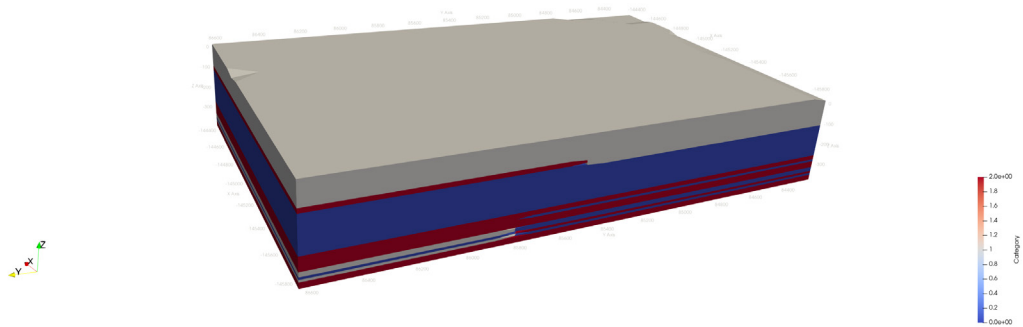


図 9 六甲アイランド周辺の地盤モデル

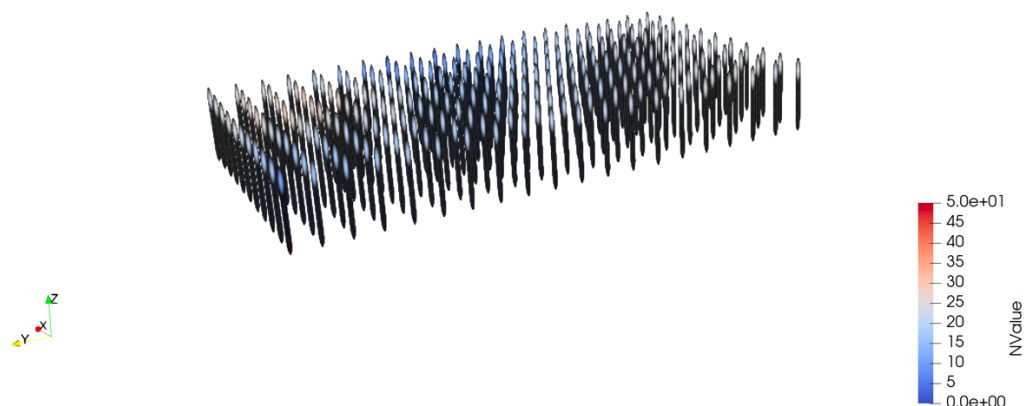


図 10 六甲アイランド周辺の N 値の三次元モデル

図 9 より、PDF から変換した XML をもとに六甲アイランド周辺の地盤モデルを構築できていることがわかる。また、図 10 のように N 値のデータに関しても三次元かできていることがわかる。これにより、六甲アイランド周辺の地下の各層、N 値がどのように連続しているのかを立体的に把握することができた。

4. まとめ

(1) 本研究の成果

本研究では、都市デジタルツインの地盤情報整備を目的として、PDF 形式のボーリング柱状図から地盤データを自動抽出し、XML 形式へ変換する手法を開発した。従来の手作業によるデータ入力の課題を解決し、地盤モデル構築の効率化を実現することができた。また、実際に生成された XML を用いて三次元地盤モデルを構築できることも確認することができた。

(2) 今後の展望

今後は検証データの数をさらに増やしてあらゆるパターンのデータを準備し、それらを検証することでより精度、汎用性がともに高い抽出プログラムを作成したい。人手で正解データを作成したこともあり、十分なテストデータを用意できず検証できていないのが課題である。あらゆる地域のボーリングデータを収集し、検証結果を集めて改善する必要があると考える。

参考文献

- 1) 国土地盤情報検索サイト Kunijiban、(最終閲覧日:2025 年 3 月 14 日)、
<https://www.kunijiban.pwri.go.jp/jp/>
- 2) 国土交通データプラットフォーム、(最終閲覧日:2025 年 3 月 14 日)、
<https://www.mlit-data.jp/#/>
- 3) 国土交通データプラットフォーム API リファレンス、(最終閲覧日:2025 年 3 月 14 日)、
https://www.mlit-data.jp/api_docs/reference/general/introduction.html
- 4) GraphQL, GraphQL 公式ドキュメント、(最終閲覧日:2025 年 3 月 14 日)、
<https://graphql.org/>
- 5) Takeyama T., O-Tani H., Oishi S., Hori M., Iizuka A.: Automatic Construction of Three-Dimensional Ground Model by Data Processing, *IEEE/ASME Transactions on Mechatronics*, DOI: 10.1109/TMECH.2021.3105062, 2021.

筆者:1) 木崎 崇雄、神戸大学大学院工学研究科市民工学専攻、学生; 2) 竹山 智英、神戸大学大学院工学研究科市民工学専攻、教授; 3) 橘 伸也、神戸大学都市安全研究センター、教授; 4) 高山 裕介、神戸大学都市安全研究センター、准教授

Automated data extraction from borehole logs for the development of numerical ground models

Takao Kizaki
Tomohide Takeyama
Shinya Tachibana
Yusuke Takayama

Abstract

In recent years, urban digital twins have gained attention for disaster prevention and urban planning. A key component of these twins is a detailed three-dimensional (3D) ground model, which relies on borehole data available in two formats: PDF-based borehole logs and XML data. While XML facilitates straightforward 3D modeling, many borehole logs along Japan's coastlines exist only as PDFs, creating challenges in constructing comprehensive models. To address this, we propose a method to extract geological data from PDF borehole logs and convert it into XML for subsurface modeling via DPP. Using the Ministry of Land, Infrastructure, Transport and Tourism's Data Platform API and GraphQL queries, we efficiently collected borehole data. We then developed a Python-based PDF parser to interpret vector data, performing coordinate transformations and error corrections to map geological layers and N-values accurately to an XML schema compatible with DPP. Validation with real data demonstrated high precision in converting PDF logs to XML, enabling the reuse of legacy data for 3D ground modeling. This approach simplifies data preparation for urban digital twins, providing a robust and detailed representation of underground conditions, ultimately improving disaster mitigation and urban planning.

©2025 Research Center for Urban Safety and Security, Kobe University, All rights reserved.