

土木研究所資料

二次元レーザスキャナによる 土石流の流量観測手法

令和5年11月

国立研究開発法人土木研究所
土砂管理研究グループ火山・土石流チーム

Copyright © (2023) by P.W.R.I.

All rights reserved. No part of this book may be reproduced by any means, nor transmitted, nor translated into a machine language without the written permission of the Chief Executive of P.W.R.I. The program source code is licensed under CC BY-SA (Attribution-ShareAlike) 4.0.

この報告書は、国立研究開発法人土木研究所理事長の承認を得て刊行したものである。したがって、本報告書の全部又は一部の転載、複製は、国立研究開発法人土木研究所理事長の文書による承認を得ずしてこれを行ってはならない。ただし、プログラムソースコードのライセンスは CC BY-SA (表示-継承) 4.0 である。

二次元レーザスキャナによる 土石流の流量観測手法

| | | |
|-----------|--------|------|
| 火山・土石流チーム | 元主任研究員 | 今森直紀 |
| 火山・土石流チーム | 研究員 | 清水武志 |
| 火山・土石流チーム | 交流研究員 | 池島 剛 |
| 火山・土石流チーム | 上席研究員 | 伊藤誠記 |

要 旨 :

土石流の流量は土石流対策において重要なデータの一つであり、流速と流下断面積を計測することで把握される。火山・土石流チームでは、二次元レーザスキャナを使用し土石流の表面を計測し、土石流の流下断面積を従来よりも詳細かつ高い時間分解能で算定する方法を検討してきた。本資料は、これまでの観測結果にもとに、二次元レーザスキャナを使用した土石流の流量観測手法を解説する。また、現地での観測事例を紹介する。さらに、データ処理プログラムソースコードと計算に用いたデータを添付する。

キーワード：土石流観測、土石流の流下断面積、二次元レーザスキャナ、LiDAR

目次

| | |
|--|-----|
| 1. はじめに | 1 |
| 2. 土石流の流量観測手法..... | 2 |
| 2.1. 土石流の流量の観測手法..... | 2 |
| 2.2. 土石流の流速の計測手法とその課題..... | 2 |
| 2.2.1. ビデオカメラの映像から流速を判読する方法..... | 2 |
| 2.2.2. 非接触型流速計により流速を計測する方法..... | 3 |
| 2.3. 土石流の流下断面積の計測手法とその課題..... | 4 |
| 2.3.1. ビデオカメラの映像から流下断面積を判読する方法..... | 4 |
| 2.3.2. 水位計により流下断面積を計測する方法..... | 5 |
| 3. 二次元レーザスキャナによる土石流観測手法の概要..... | 6 |
| 3.1. 二次元レーザスキャナの計測原理..... | 6 |
| 3.2. 土石流観測に用いるレーザスキャナに必要な性能..... | 7 |
| 3.3. 計測機材の構成..... | 8 |
| 3.4. 設置位置選定にあたっての留意点..... | 9 |
| 3.5. 観測データの整理..... | 10 |
| 3.5.1. 流下断面積..... | 10 |
| 3.5.2. 土石流の流速..... | 12 |
| 3.5.3. 土石流の流量データ..... | 13 |
| 3.6. 二次元レーザスキャナによる計測結果と従来の計測手法の比較..... | 14 |
| 3.7. 二次元レーザスキャナの維持管理..... | 16 |
| 4. 桜島有村川流域における観測事例..... | 17 |
| 4.1. 観測箇所概要 | 17 |
| 4.2. 土石流の流量観測概要..... | 18 |
| 4.3. 観測データの概要..... | 20 |
| 4.4. 観測データの取得状況とデータの整理結果..... | 23 |
| 5. まとめと今後の課題..... | 25 |
| 5.1. まとめ | 25 |
| 5.2. 今後の課題 | 25 |
| 5.2.1. データ取得の安定性..... | 25 |
| 5.2.2. 土石流の流速計測手法..... | 25 |
| 謝辞 | 27 |
| 参考文献 | 28 |
| A. 付録 | A-1 |
| A.1. データ整理用プログラム..... | A-1 |

| | |
|-----------------------------|-----|
| A.2. プログラムの動作確認済み実行環境等..... | A-1 |
| A.3. 計測データの下処理..... | A-2 |
| A.4. 各プログラムの解説..... | A-4 |
| A.4.1 プログラムの実行..... | A-4 |
| A.4.2 「area.py」 | A-4 |
| A.4.3 「median.py」 | A-5 |
| A.4.4 「resampling.py」 | A-5 |
| A.4.5 「discharge.py」 | A-6 |
| A.4.6 「hydrograph.py」 | A-7 |
| A.4.7 「figure.py」 | A-7 |
| A.4.8 「animation.py」 | A-7 |
| B. 付録 プログラムソースコード..... | B-1 |

1. はじめに

土石流の観測は、土石流の流速や流量等の実態を把握し、その規模を想定するための情報を得る重要な調査の一つである。特に土石流の流量観測により得られるデータは土石流対策や土石流の数値計算において重要な土石流のピーク流量やその水深、土石流の継続時間を推定するために不可欠な調査である。土石流流量の観測は、計測した流速と流下断面積を時系列データの積として算定する。

我が国における土石流の観測が本格的に開始されたのは1970年代である^{例え1)、2)}。その後、新たな計測機器の開発や観測技術の進展に伴い、焼岳や桜島等、土石流が頻繁に発生する流域で観測されたデータが蓄積され、1980年代から1990年代にかけ、土石流の総流量と土石流のピーク流量の関係が整理されてきた^{例え3)}。これらの結果に基づき砂防堰堤の設計における土石流のピーク流量等の推定手法が提案されている⁴⁾。また、他の流域を調査対象の溪流に加え、土石流の調査結果等から土石流の総流量と土石流のピーク流量の関係に関する調査が継続されている^{例え5)}。

観測技術に注目すると、前述の土石流のピーク流量の整理に用いられた土石流の流量観測のデータは、焼岳の上々堀沢¹⁾、桜島の野尻川をはじめとする複数の流域⁶⁾、滑川の北股沢でビデオカメラにより観測された映像から得られたデータ⁷⁾が多くを占めている。近年は非接触方式の水位計及び流速計による自動計測技術も採用されているが^{例え8)}、ビデオカメラで記録した映像から土石流の流量データを整理している現場もある^{例え9)}。一方、ビデオカメラの映像の判読作業は労力を要しており、夜間や高降雨強度下等では映像が不鮮明で判読そのものが困難な場合もある。また、流速や水位等の定量的な判読を行うことは困難である¹⁰⁾。

そこで、火山・土石流チームでは、土石流の流下断面積の計測手法の改善を目的として、二次元レーザスキャナ（以下、「レーザスキャナ」という）を用いた計測手法を考案し、室内実験および現地での観測に取り組んできた^{11)、12)、13)、14)}。本資料はレーザスキャナを用いた土石流の流量観測手法について実際の観測事例も交えて解説するものである。また、巻末にはデータ処理プログラムソースコードと計算に用いたデータ（DVD-R）を添付する。

2. 土石流の流量観測手法

2.1. 土石流の流量の観測手法

ある時刻における土石流の流量 Q_s は次式で求めることができる⁴⁾。

$$Q_s = U \cdot A_d \quad (1)$$

ここに、 U : 土石流の流速 (m/s、以下、「流速」という)、 A_d : 土石流の流下断面積 (m²、以下、「流下断面積」という) である。

土石流の流量を求めるには、 U および A_d を時系列で計測する必要がある。以下、それぞれの計測手法について我が国で採用されている手法の概要を解説する。

2.2. 土石流の流速の計測手法とその課題

流速 U はビデオカメラの映像から判読する方法や、非接触型流速計¹⁰⁾による方法により計測される。以下、それぞれの方法の概要とその課題を述べる。

2.2.1. ビデオカメラの映像から流速を判読する方法

ビデオカメラの映像を判読する方法は、土石流の表面を流下する礫の移動速度をビデオ映像から判読するものである (図 1)。映像で記録された範囲において、複数の測定面で計測することが可能であるが、ビデオカメラの映像をコマ送りしながら目視で判読するため、データ整理には多くの労力が必要である。目視による判読作業に替わる方法として、粒子画像流速測定法 (PIV: Particle Image Velocimetry) を用いてビデオカメラの映像から流速を計測する方法等も提案されているが、モヤがかかり映像が不鮮明な場合や映像の輝度の差を捉えがたい場合には計測が困難である¹⁵⁾。

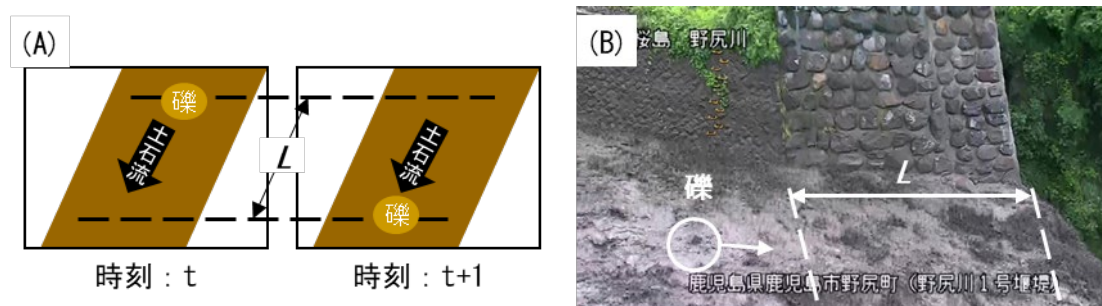


図 1 ビデオカメラの映像から流速を判読する方法

(A) 判読方法の概要、(B) ビデオカメラの映像例 (国土交通省提供資料に加筆)

2.2.2. 非接触型流速計により流速を計測する方法

非接触型流速計による計測手法は土石流の表面にマイクロ波や超音波を発射し、土石流表面で反射した波の周波数がドップラー効果により変化することを利用して表面流速を計測するものである（図 2）¹⁰⁾。かつてビデオカメラの映像を判読してきた現場においても、近年は非接触型流速計による計測手法が併用されている^{例えば7)}。なお、表面流速を自動で連続観測することが可能であるが、流速計 1 台につき溪流内の 1 点の流速しか計測できない^{14)、15)}。このほか、二次元レーザスキャナにより計測した土石流の水脈飛距離から表面流速を求める方法も提案されているが、実用化に向けて様々な現場における計測データの蓄積を通じ、検討と改善の必要がある¹⁶⁾。

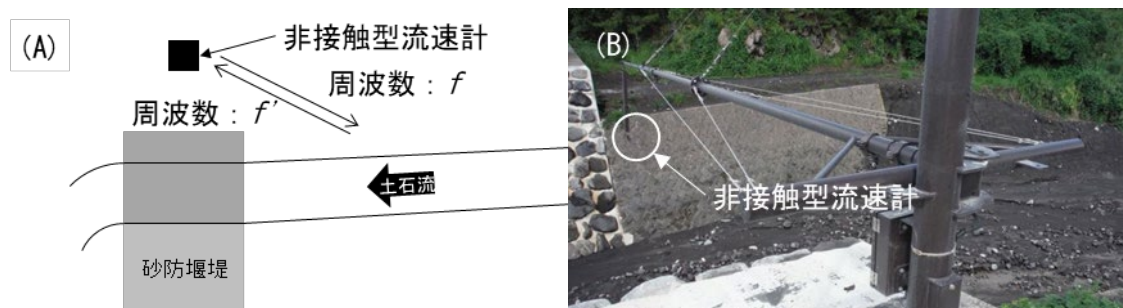


図 2 非接触型流速計により流速を計測する方法
(A) 計測方法の概要、(B) 実際の設置例

2.3. 土石流の流下断面積の計測手法とその課題

流下断面積 A_d はビデオカメラの映像から判読する方法や、水位計により計測する方法がある。以下、それぞれの方法の概要とその課題を述べる。

2.3.1. ビデオカメラの映像から流下断面積を判読する方法

ビデオカメラの映像を判読する方法は、砂防堰堤等の水通し断面を流下する際の流下断面を、袖にペンキで表示した目盛り等を参照し判読するものである（図3）¹⁷⁾。土石流の表面形状が複雑であっても、映像をトレースすることで形状を把握することが可能である。ただし、目視による判読であることから、特に水深方向では実際の流下断面とは異なる可能性がある⁹⁾。既設の監視カメラの映像を利用することができ、観測機材を追加で設置することは不要であるが、夜間や高降雨強度下等では映像が不鮮明で判読そのものが困難な場合もある。

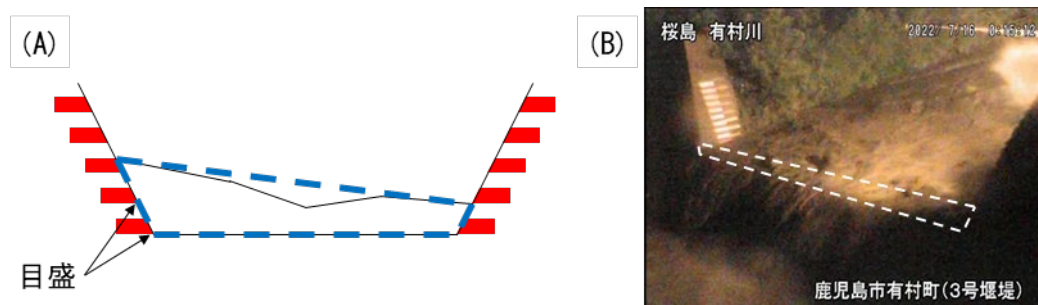


図3 ビデオカメラの映像から流下断面積を判読する方法

(A) 判読方法の概要、(B) ビデオカメラの映像例（国土交通省提供資料に加筆）

2.3.2. 水位計により流下断面積を計測する方法

水位計による計測は、マイクロ波や超音波が土石流表面に反射して戻ってくる時間差や位相差等から計測器から水面までの距離を計測する方法である（図4）¹⁰⁾。一般的には砂防堰堤等の水通し断面における水位を計測し、土石流の断面形状を台形と仮定して断面積を求める。自動で計測およびデータの記録が可能のため、判読などの作業は不要である。ただし、計測可能な情報は1点のみであるため、土石流の表面形状が複雑な場合には、正確な流下断面積を算出できない可能性がある。また、河床変動により土石流の滞筋が測定地点を外れると欠測が生じる¹⁴⁾。

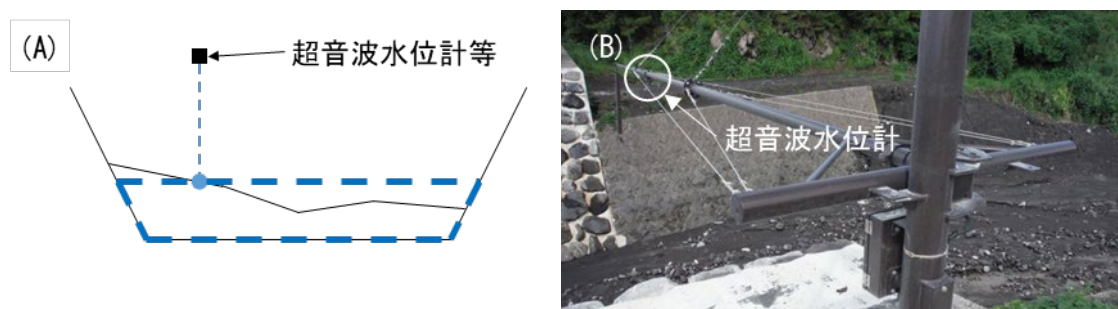


図4 水位計により流下断面積を計測する方法

(A) 計測方法の概要、(B) 実際の設置例（国土交通省提供資料に加筆）

3. 二次元レーザスキャナによる土石流観測手法の概要

前述のような課題を踏まえ、火山・土石流チームでは、流下断面積の計測に産業用ロボット分野で利用されるレーザ距離計のひとつであるレーザスキャナを使用する方法を考案し、室内実験試験および現地での観測に取り組んできた^{11)、12)、13)、14)}。この計測方法は、水位計によるある一点の水位を計測する方法と異なり、土石流の流下断面を計測することが可能である。以下にその概要を説明する。なお、本観測手法では、流速の計測に非接触型流速計を使用している。

3.1. 二次元レーザスキャナの計測原理

レーザスキャナは、内部の反射板を高速で回転させ、赤外線レーザ光の照射角度を変化させながら回転面内に存在する対象までの距離を計測する光波距離計である。通常、変化させる照射角度は一定であり、これをステップ角という。1軸走査の場合、走査面内の物体形状は図5のように計測点間を結んだ線として得られ、平面的に形状を把握することが可能である。

また、赤外線レーザ光を使用するため、日中夜間を問わず物体の形状を把握することが可能である。

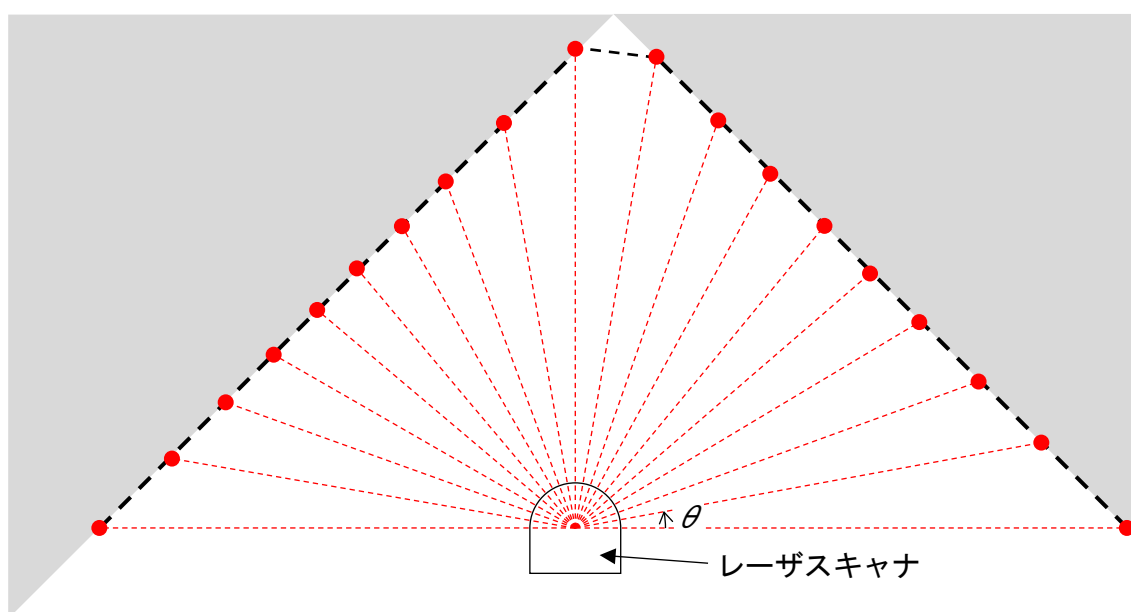


図5 レーザスキャナによる物体形状の検知イメージ

3.2. 土石流観測に用いるレーザスキャナに必要な性能

土石流観測に用いるレーザスキャナに必要な性能は以下のとおりである。

- 流下断面積はプログラムを使用して整理するため、レーザスキャナで計測したデータをテキストデータで出力できること。もしくは、バイナリデータであってもテキストデータに変換が可能であること。
- マルチエコー機能¹⁸⁾を搭載していること。レーザが雨滴や霧等に当たると、これらに反射するレーザとこれらを透過して奥にある物体に反射するレーザの2種類がセンサへ返る場合がある。このため、同じステップ角上で返ってきた2以上のレーザ光から距離データを取得することができるマルチエコー機能を搭載していることが必要である。
- 1秒間に複数回の走査が可能であること。マルチエコー機能を搭載していても、降雨強度が強い場合には雨滴を計測する場合がある。このため、後述の通り、雨滴によりノイズとなったデータを統計的手法により除却するために1秒間に複数回の走査が可能である必要がある。
- 屋外での計測に使用するため、防水・防塵の性能を有すること。具体的には、IP67相当以上の防水・防塵性能を有することが望ましい。

3.3. 計測機材の構成

土石流観測は、レーザスキャナを含め次の機材で構成される。

- レーザスキャナ
- レーザスキャナの制御、記録用コンピュータ
- 非接触型流速計
- 電源

レーザスキャナは前述の性能を満たす機材を用いる。一般的に、レーザスキャナはコンピュータにインストールしたソフトウェアにより諸設定、計測の開始、停止およびデータの記録といった一連の操作を行う。レーザスキャナとコンピュータはイーサネットケーブルで接続する場合が多い。コンピュータは現地に設置するため、防水・防塵性能を有するボックスの中に収納する等の対応が必要となる。現地ではボックス内も多湿な環境となることも想定されるため、防水性能を有するコンピュータを使用したり、ボックス内に設置した除湿剤を定期的に交換したりするなど適宜対策を講じる必要がある。また、ボックス等の内部が各種機器の動作保証温度を超えるような高温または低温とならないよう、適宜対策を講じる。コンピュータに搭載するオペレーティングシステム（OS）はソフトウェアに対応したものを選定する。また、データを記録するために十分な容量の記憶領域を確保する。

非接触型流速計はレーザスキャナおよびコンピュータとは独立した計測機材である。一般的に、非接触型流速計とデータロガーがセットである場合が多い。非接触型流速計のサンプリングレートはレーザスキャナと同程度かそれよりも大きいことが望ましい。

電源はレーザスキャナ、コンピュータおよび非接触型流速計へ電力を供給するために必要であり、商用電源が使用できることが望ましい。山間部では頻繁に豪雨時に瞬時停電が発生することも想定されるため、商用電源を利用する場合は無停電電源装置（UPS）を設置し、各機材への電力供給はUPSを経由することが望ましい。また、落雷サージへの対策も適宜検討する。商用電源の使用が困難な箇所では、上記の機材を安定的に稼働させるために必要な電力量を計算し、太陽光発電とバッテリーによる給電を検討する。

3.4. 設置位置選定にあたっての留意点

既に土石流観測を実施している場合、過去の観測データについてその正確度の検証が可能となるよう、既設の観測箇所や CCTV（Closed-circuit Television）等の現場監視カメラが設置されている箇所が候補となる。

土石流観測を実施していない箇所で新たに観測を開始する場合、土石流の流下に伴う河床の侵食や土砂の堆積による河床変動が流下断面積へ影響を及ぼすことを避けるため、砂防堰堤の水通しなど固定床の箇所を選定する。また、レーザスキャナとともに非接触型流速計を設置することから、洪水流量観測に準じ¹⁹⁾、以下についても留意する。

- ・ 非接触型流速計のセンサ面からの対水面間距離が機器仕様の範囲内であること。
- ・ 非接触型流速計の電波照射範囲や超音波放射範囲に護岸、岩、河床、樹木やその枝葉等、流水以外がないこと。
- ・ 対象洪水時の水表面流速が機器の流速計測範囲内であること。
- ・ 河道が湾曲していない地点で、かつ大きく流速の異なる水域がなく、渦を巻くような流れにならない場所であること。

レーザスキャナは極力砂防堰堤の水通しからの比高が高く、水通し天端の中央部に近い位置へ設置することが望ましい（図 6 (A)）。なお、レーザスキャナから照射されたレーザ光が、ある物体で反射すると、物体の陰となる範囲は計測することができない。特に、入射角が小さい範囲では、土石流の表面形状が著しく乱れる場合や水通しを巨礫等が通過する場合に土石流の表面一部が計測できない場合がある（図 6 (B)）。また、レーザスキャナから遠距離となる範囲では、相対的に水平方向の点群の密度が疎となる（図 6 (B)）。

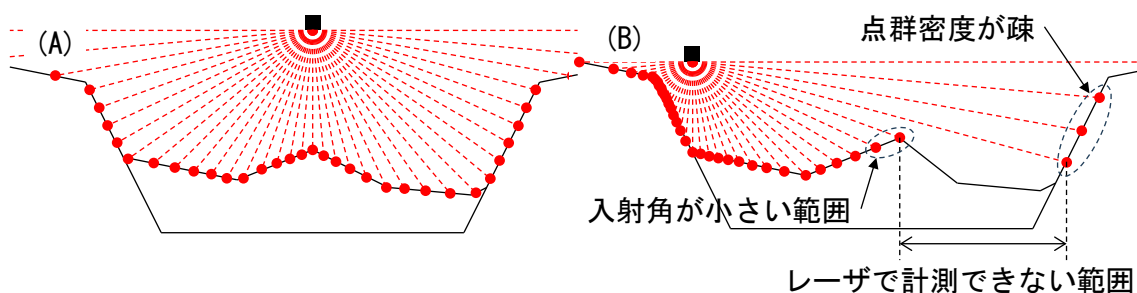


図 6 レーザスキャナの設置位置による点群データ取得の違いのイメージ

3.5. 観測データの整理

3.5.1. 流下断面積

レーザスキャナで観測したデータから、土石流の表面と水通し断面で囲まれた多角形の面積を求めることで流下断面積のデータを時系列で整理する。

水通し天端の左岸端または右岸端を原点とし、水平方向を x 軸、 x 軸と直行する軸を y 軸とすると、流下断面積 S は各点群と水通し天端および袖で囲まれた多角形として求めることができる。図 7 の点 $P_{k1}=(x_k, y_k)$ 、点 $P_{k+1}=(x_{k+1}, y_{k+1})$ 、点 $P'_k=(x_k, 0)$ 、点 $P'_{k+1}=(x_{k+1}, 0)$ の 4 点で囲まれた斜線部の台形の面積 S_k は台形公式を使えば次式で求めることができる。

$$S_k = (x_{k+1} - x_k) \cdot \frac{y_k + y_{k+1}}{2} \quad (2)$$

同様の方法で水通し天端内の台形の面積を順次求め、合計すると S' を得ることができる。

$$S' = \sum_{k=1}^n S_k \quad (3)$$

袖部は計測点が袖であるか土石流の表面であるかの判定が困難なため、水通し天端端部の点のデータから、簡便に次式で求める。

$$S_L = \frac{(x_0 - x_L) \cdot y_0}{2} \quad (4)$$

$$S_R = \frac{(x_R - x_n) \cdot y_n}{2} \quad (5)$$

I を袖小口の勾配とすると、左右岸でそれぞれ下記の通りである。なお、一般的に $I=0.5$ である²⁰⁾。

$$I = \frac{x_0 - x_L}{y_0} \quad (6)$$

$$\text{よって } x_0 - x_L = y_0 \cdot I \quad (7)$$

$$I = \frac{x_R - x_n}{y_n} \quad (8)$$

$$\text{よって } x_R - x_n = y_n \cdot I \quad (9)$$

式(4)に式(7)を、式(5)に式(9)をそれぞれ代入すると、 S_L および S_R はそれぞれ次式となる。

$$S_L = \frac{y_0^2 \cdot I}{2} \quad (10)$$

$$S_R = \frac{y_n^2 \cdot I}{2} \quad (11)$$

以上より、流下断面積 S は次式のとおりである。

$$S = S' + S_L + S_R \quad (12)$$

なお、降雨強度が強い場合、レーザ光が雨滴で反射する場合がある。この場合、土石流の表面より高い位置で点群が取得される (図 8)。このため、流下断面積が過大に計算され、ノイズデータとなる (図 9 (A))。雨滴を計測した点群データのみを除却することは困難であるが、1 秒間の複数の流下断面積のデータのうち、中央値を抽出することで、ノイズとなるデータを除却することができる (図 9 (B))。1 秒毎に同様の処理を行うことで、流下断面

積を求めることができる。同様の方法を適用してもノイズが生じている場合、当該時間帯のデータを断面図として可視化し、確認する。

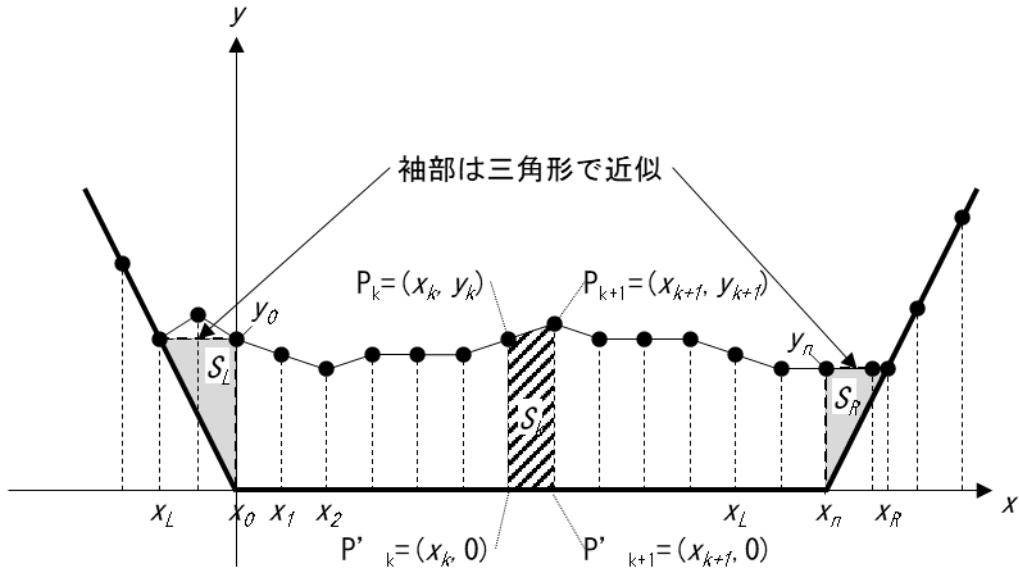


図 7 流下断面積の計算方法

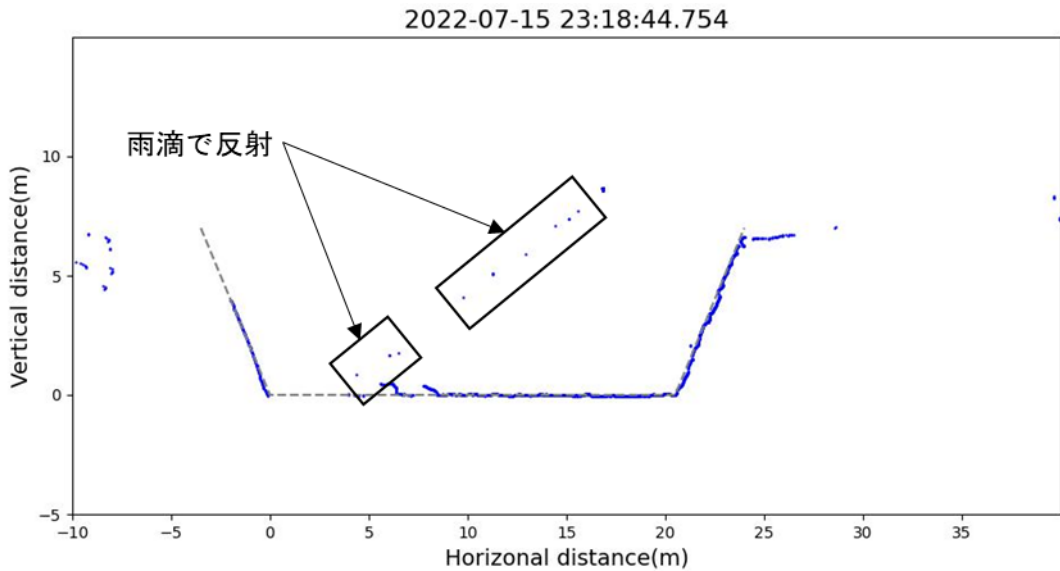


図 8 雨滴によるレーザ光の反射を物体として検出した事例

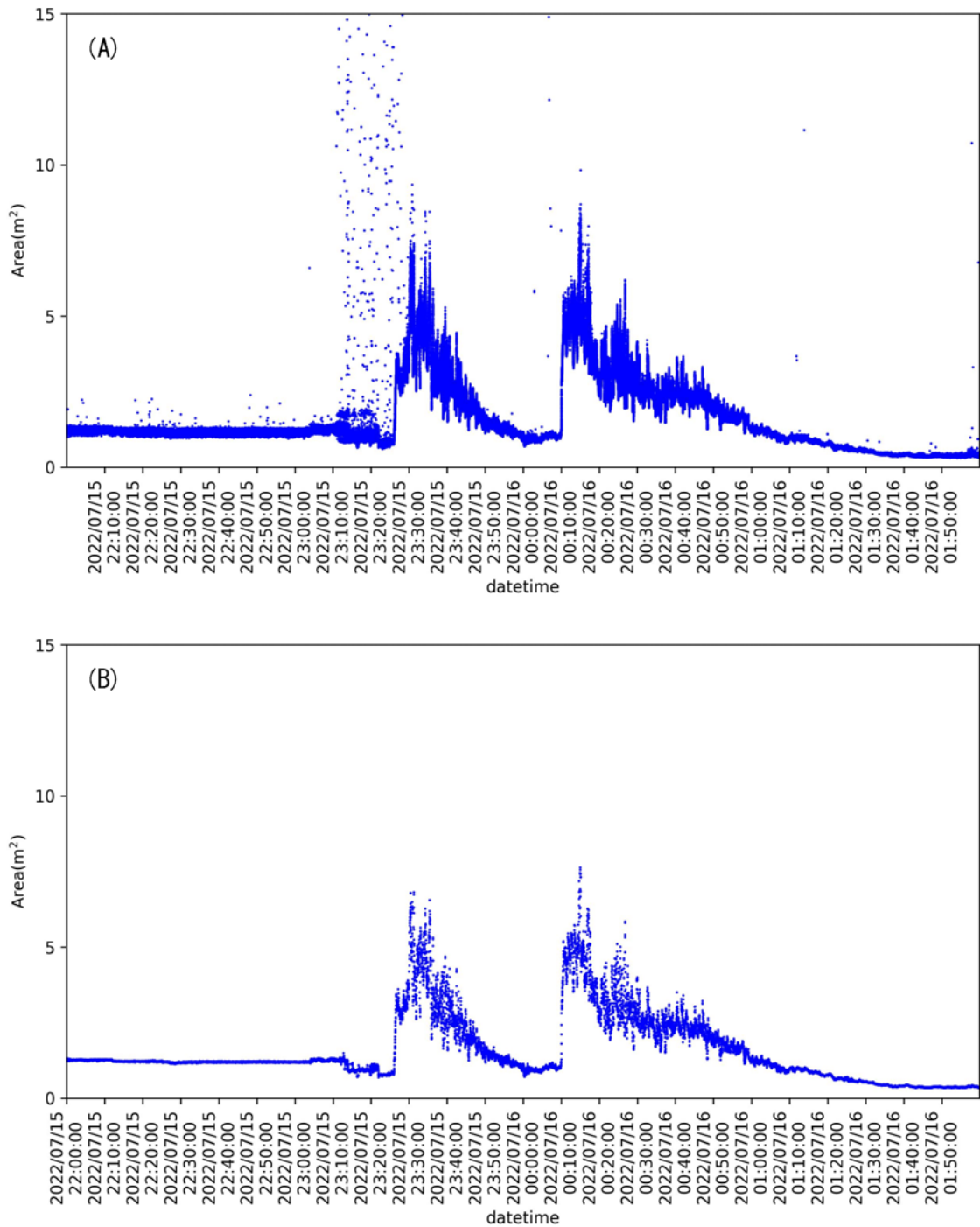


図 9 ノイズ除去前後の流下断面積の時系列データ

(A)ノイズ除去前、(B) ノイズ除去後

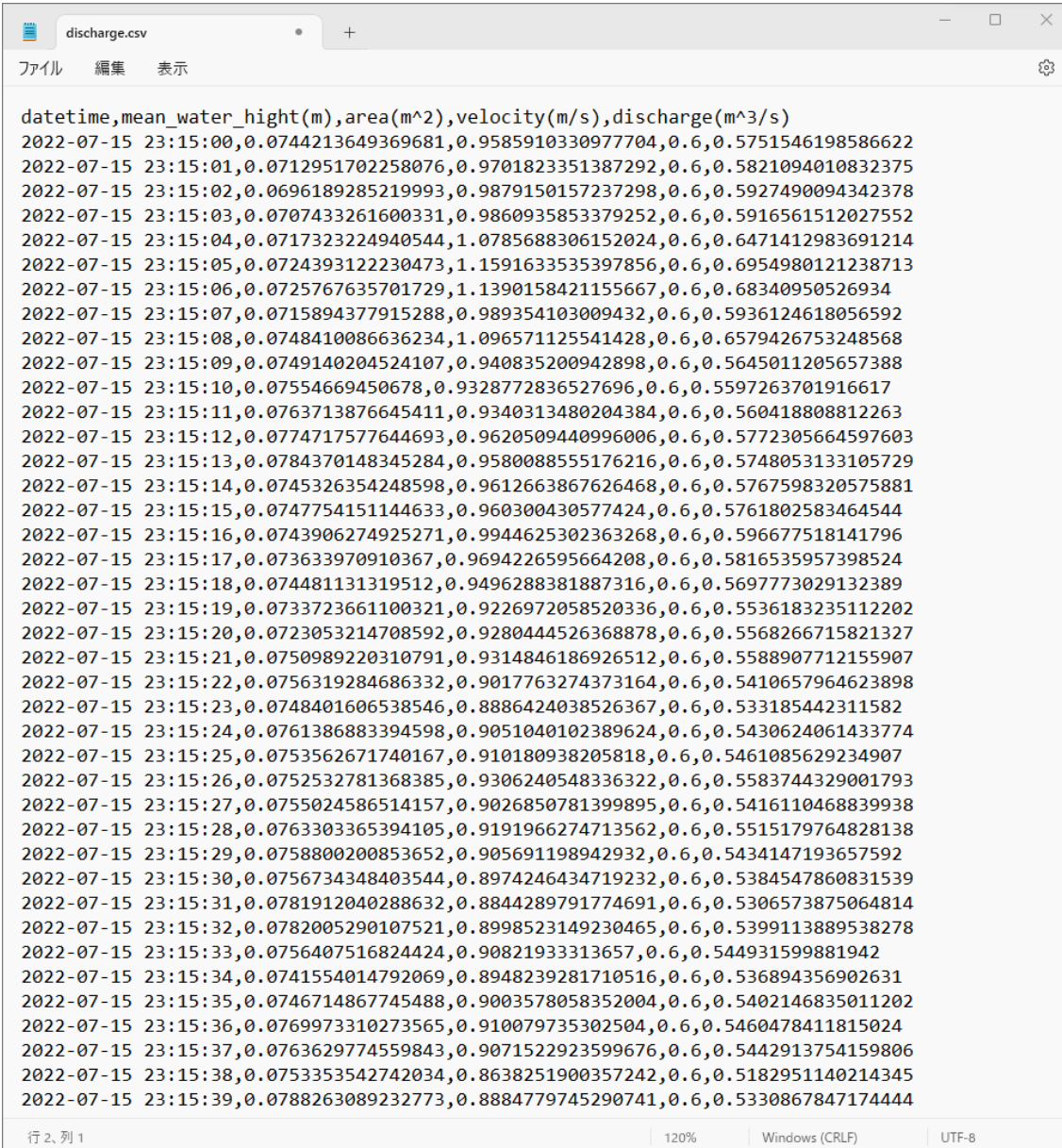
3.5.2. 土石流の流速

非接触型流速計により得られる流速は表面流速であることから、土石流の平均流速を求める必要がある。既往の研究では、土石流の流速分布式²¹⁾から、土石流の表面流速に 3/5 を

乗じて土石流の流速を求めている²²⁾。

3.5.3. 土石流の流量データ

整理したデータは、土石流数値計算における検証作業にも使用できるよう、**図 10** に示すとおり `datetime`、平均水位 (m)、流下断面積 (m^2)、土石流の平均流速 (m/s)、流量 (m^3/s) を時系列データで整理する。データの形式はプログラムへの入力データとして扱いやすいよう、`csv` 形式や `txt` 形式の共通フォーマットで保存することが望ましい。



```
datetime,mean_water_high(m),area(m^2),velocity(m/s),discharge(m^3/s)
2022-07-15 23:15:00,0.0744213649369681,0.9585910330977704,0.6,0.5751546198586622
2022-07-15 23:15:01,0.0712951702258076,0.9701823351387292,0.6,0.5821094010832375
2022-07-15 23:15:02,0.0696189285219993,0.9879150157237298,0.6,0.5927490094342378
2022-07-15 23:15:03,0.0707433261600331,0.9860935853379252,0.6,0.5916561512027552
2022-07-15 23:15:04,0.0717323224940544,1.0785688306152024,0.6,0.6471412983691214
2022-07-15 23:15:05,0.0724393122230473,1.1591633535397856,0.6,0.6954980121238713
2022-07-15 23:15:06,0.0725767635701729,1.1390158421155667,0.6,0.68340950526934
2022-07-15 23:15:07,0.0715894377915288,0.989354103009432,0.6,0.5936124618056592
2022-07-15 23:15:08,0.0748410086636234,1.096571125541428,0.6,0.6579426753248568
2022-07-15 23:15:09,0.0749140204524107,0.940835200942898,0.6,0.5645011205657388
2022-07-15 23:15:10,0.07554669450678,0.9328772836527696,0.6,0.5597263701916617
2022-07-15 23:15:11,0.0763713876645411,0.9340313480204384,0.6,0.560418808812263
2022-07-15 23:15:12,0.0774717577644693,0.9620509440996006,0.6,0.5772305664597603
2022-07-15 23:15:13,0.0784370148345284,0.958008855176216,0.6,0.5748053133105729
2022-07-15 23:15:14,0.0745326354248598,0.9612663867626468,0.6,0.5767598320575881
2022-07-15 23:15:15,0.0747754151144633,0.960300430577424,0.6,0.5761802583464544
2022-07-15 23:15:16,0.0743906274925271,0.9944625302363268,0.6,0.596677518141796
2022-07-15 23:15:17,0.073633970910367,0.9694226595664208,0.6,0.5816535957398524
2022-07-15 23:15:18,0.074481131319512,0.9496288381887316,0.6,0.5697773029132389
2022-07-15 23:15:19,0.0733723661100321,0.9226972058520336,0.6,0.5536183235112202
2022-07-15 23:15:20,0.0723053214708592,0.9280444526368878,0.6,0.5568266715821327
2022-07-15 23:15:21,0.0750989220310791,0.9314846186926512,0.6,0.5588907712155907
2022-07-15 23:15:22,0.0756319284686332,0.9017763274373164,0.6,0.5410657964623898
2022-07-15 23:15:23,0.0748401606538546,0.8886424038526367,0.6,0.533185442311582
2022-07-15 23:15:24,0.0761386883394598,0.9051040102389624,0.6,0.5430624061433774
2022-07-15 23:15:25,0.0753562671740167,0.910180938205818,0.6,0.5461085629234907
2022-07-15 23:15:26,0.0752532781368385,0.9306240548336322,0.6,0.5583744329001793
2022-07-15 23:15:27,0.0755024586514157,0.9026850781399895,0.6,0.5416110468839938
2022-07-15 23:15:28,0.0763303365394105,0.9191966274713562,0.6,0.5515179764828138
2022-07-15 23:15:29,0.0758800200853652,0.905691198942932,0.6,0.5434147193657592
2022-07-15 23:15:30,0.0756734348403544,0.8974246434719232,0.6,0.5384547860831539
2022-07-15 23:15:31,0.0781912040288632,0.8844289791774691,0.6,0.5306573875064814
2022-07-15 23:15:32,0.0782005290107521,0.8998523149230465,0.6,0.5399113889538278
2022-07-15 23:15:33,0.0756407516824424,0.90821933313657,0.6,0.544931599881942
2022-07-15 23:15:34,0.0741554014792069,0.8948239281710516,0.6,0.536894356902631
2022-07-15 23:15:35,0.0746714867745488,0.9003578058352004,0.6,0.5402146835011202
2022-07-15 23:15:36,0.0769973310273565,0.910079735302504,0.6,0.5460478411815024
2022-07-15 23:15:37,0.0763629774559843,0.9071522923599676,0.6,0.5442913754159806
2022-07-15 23:15:38,0.0753353542742034,0.8638251900357242,0.6,0.5182951140214345
2022-07-15 23:15:39,0.0788263089232773,0.8884779745290741,0.6,0.5330867847174444
```

図 10 データの整理方法の例

3.6. 二次元レーザスキャナによる計測結果と従来の計測手法の比較

後述の桜島有村川流域の計測データを用いて従来の計測手法による流下断面積のデータと比較した(図11)。比較対象は以下のとおりである。計測データは2022年7月15日23時から7月16日1時にかけての2時間のデータを用いた。

Case1 CCTVによる映像から判読した水位から計算した流下断面積

Case2 レーザスキャナ直下の水位データから計算した流下断面積

Case3 二次元レーザスキャナで計測したデータから計算した流下断面積

なお、Case1は国土交通省九州地方整備局大隅河川国道事務所(以下、「事務所」という)が整理したデータを使用した。また、観測箇所では超音波水位計が設置されていないため、Case2ではレーザスキャナの直下の横断方向1mの水位データの平均値を超音波水位計のデータに見立てた。

Case1の流下断面積はCase2およびCase3に対して過大である。有村川3号ダム付近に設置したCCTVは図12のように左岸から水通し断面を撮影しており、映像から水位を判読する場合、誤差が生じやすいと考えられる。Case2およびCase3は時系列的な傾向は同調するが、Case2はCase3に対して過大な流下断面積となる。これは、観測箇所では土石流の水位が左岸側に偏る場合が多いため、水通しの左岸側から3.5mの位置に設置されているレーザスキャナ直下のデータを使用することで右岸側の水位が過大評価されるためである。なお、Case3で初期の値がゼロでないのは、水通し上の巨礫の断面を計測したためである(図13)。

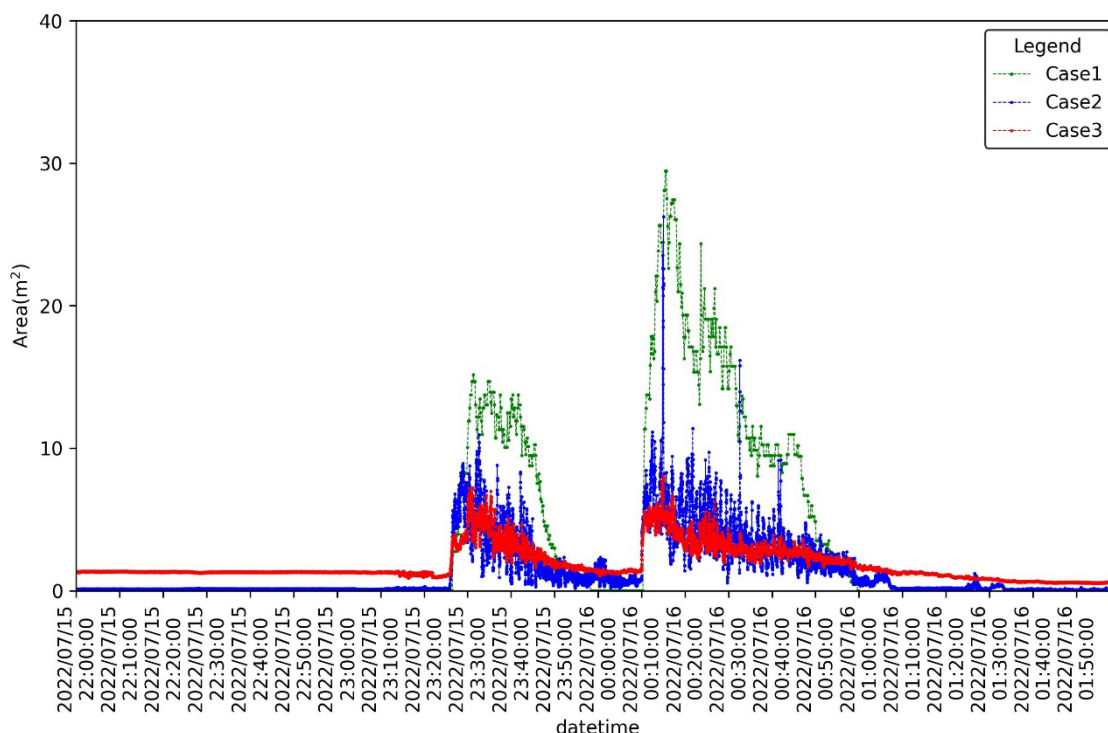


図11 各計測手法による流下断面積の比較



図 12 判読に用いる映像の例 (国土交通省提供)

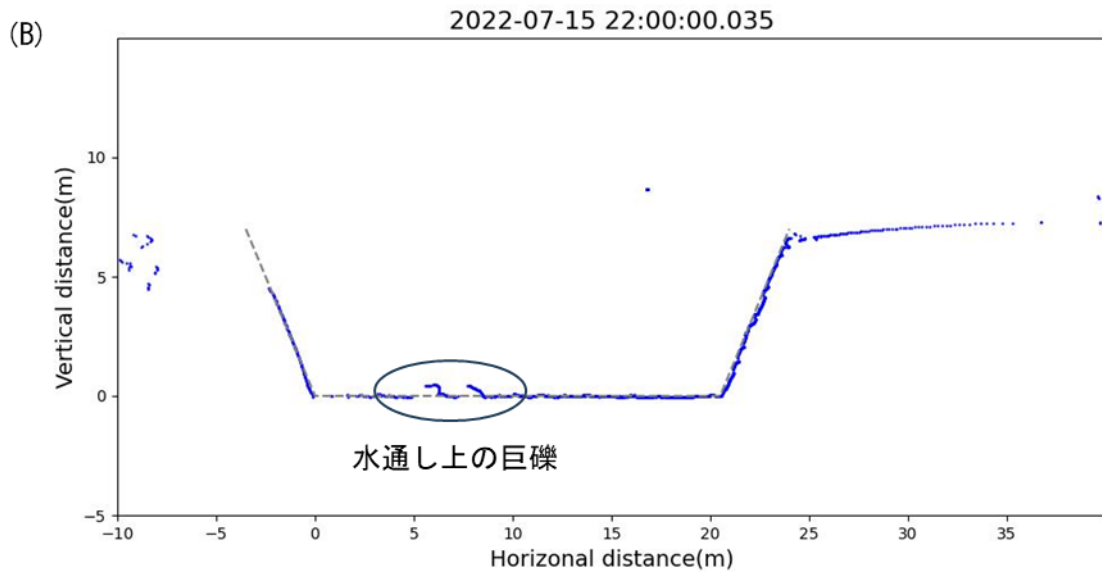


図 13 現地の状況

(A)現地状況 (2022年7月14日)、(B)レーザスキャナで計測した形状 (2022年7月15日)

3.7. 二次元レーザスキャナの維持管理

屋外でレーザスキャナを使用する場合、火砕物や砂塵の付着、虫の営巣等によりレーザスキャナの光学窓が汚れることがある。この場合、照射されたレーザ光が汚れで反射し、正しく計測できない。このため、定期的にレーザスキャナの光学窓の清掃を実施し、汚れを取り除く必要がある（図 14）。また、砂塵の付着を防止するため、レーザスキャナの上空に屋根を設置することも有効と考えられる。

なお、火砕物が付着している場合、ウエス等で直接拭き取ると火砕物中のガラス質による研磨作用で光学窓を傷つける可能性がある。この場合も光学窓の傷部でレーザ光が反射し、正しく計測できないことがある。火砕物が付着している場合には、火砕物をブロワーで吹き払うか、水で洗い流してから光学窓を清掃することが望ましい。



図 14 レーザスキャナの保守点検状況

4. 桜島有村川流域における観測事例

4.1. 観測箇所概要

土石流観測を実施している有村川3号堰堤（以下、堰堤という）上流の地形図を、**図 15**に示す。有村川流域は桜島南東斜面に位置しており、堰堤上流の流域面積は約 1.6 km²である。



図 15 観測流域概要

(A)有村川流域位置図および平面図（地理院タイルに流域界、砂防堰堤位置、縮尺、方位記号を追記して掲載）、(B)有村川流域斜め写真（国土交通省提供）、(C)土石流観測箇所概観

4.2. 土石流の流量観測概要

レーザスキャナは、堰堤水通し部の横断方向の土石流の表面形状を計測している。計測に使用したレーザスキャナは、北陽電機（株）の測域センサ UXM-30LAH-EWA で波長 905 nm の近赤外線レーザ光を 0.125 度ステップで照射し回転面内の 1520 点（190 度走査）において距離（測距範囲 0.1～30m）を測定分解能 1mm で計測して、物体の形状を計測する。また、サンプリングレートは 20Hz であり、1 断面を 50ms（ミリ秒）毎に計測可能であることから、時系列的に詳細な断面変化を追跡可能である¹⁴⁾。

計測機材の構成の概略を図 16 に示す。計測機材設置用の鋼製片持ち梁にはレーザスキャナおよび流速計が設置されており、それぞれの計測機材の電源は左岸の袖天端に設置したボックス内の電源盤から供給される。赤色太実線は火山・土石流チームが設置した機材、黒色太実線は国土交通省九州地方整備局大隅河川国道事務所が設置した機材である。なお、計測機器への電源供給は UPS を経由しており、瞬時停電が生じても計測が継続できる。電源ボックス内には図 17 のとおりデータ記録用のラップトップ型パーソナルコンピュータ（以下、「ノート PC」という）が収納されており、レーザスキャナとはイーサネットケーブルにより接続されている。レーザスキャナの制御とデータの記録はノート PC にインストールした専用ソフトにより行う。

現地は携帯電話の電波を安定的に使用することができないため、リモートデスクトップ等を利用することができない。このため、概ね 2 か月に 1 度の頻度で記録したデータを回収している。また、この際、レーザスキャナ等の各種機材の保守点検を実施している。

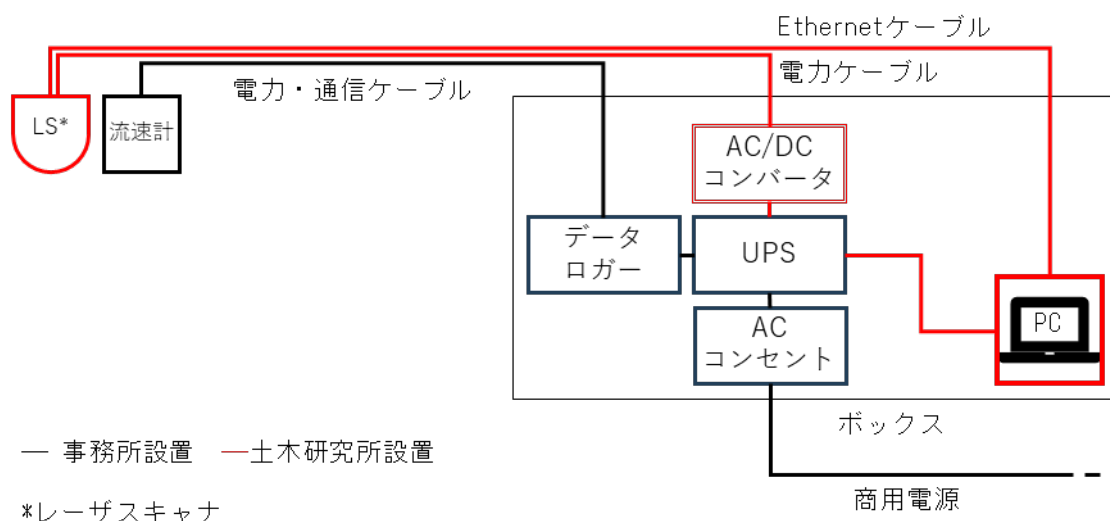


図 16 観測機材の構成図

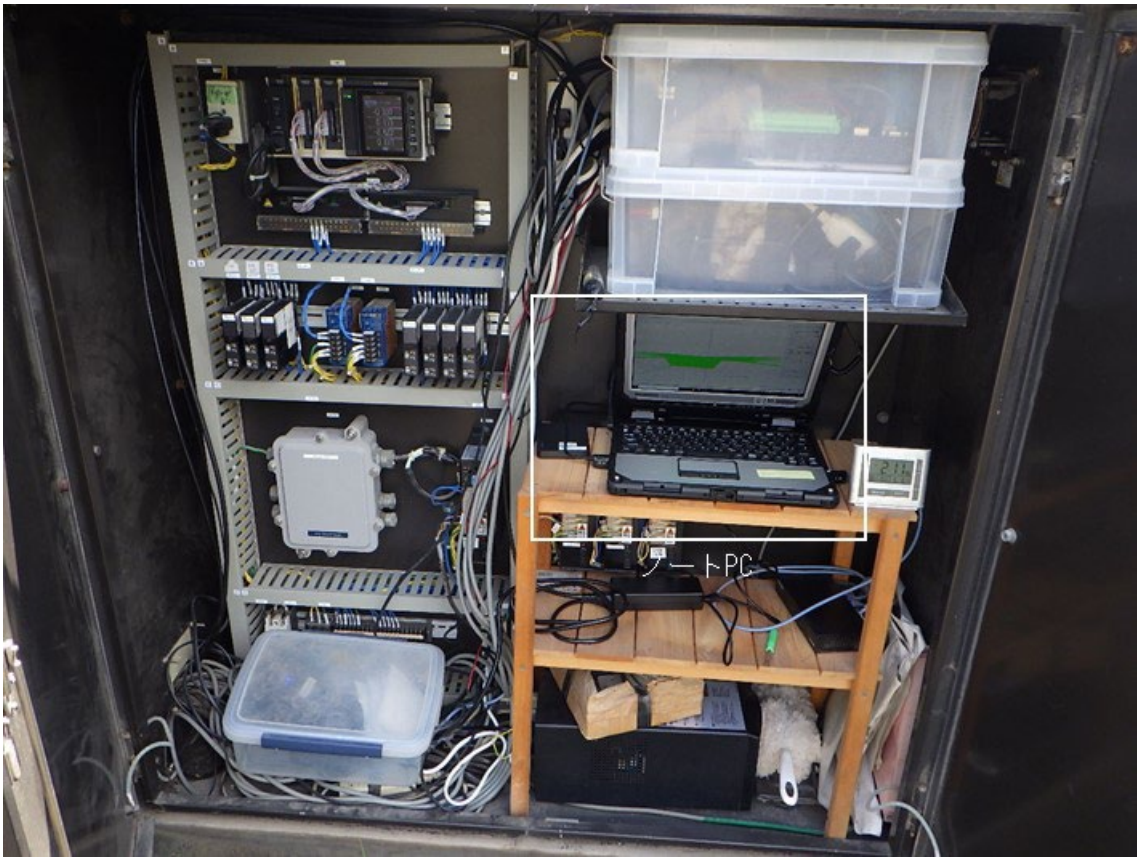


図 17 電源ボックス内の状況

4.3. 観測データの概要

レーザスキャナで計測したデータは極座標系で記録される。使用したレーザスキャナでは、**図 18** のとおり、偏角 $\theta=0^\circ$ からステップ角 θ_s 毎においてレーザ光が反射した地点までの距離（動径） r が記録される。ここで、極座標と平面直角座標の関係は**図 18** のとおりであるため、例えば**図中**の点 P の座標は $(r\cos\theta, r\sin\theta)$ と表すことができる。

レーザスキャナは土石流を計測するため、砂防堰堤の水通しに向けて設置しており、「3.5.1. 流下断面積」に示す方法で流下断面積を整理するため、**図 19** に示すように点 P を点 P' の位置へ回転させる必要がある。ここで、据え付け角を α 、レーザスキャナの計測可能範囲の角度を β とすると、回転後の偏角 θ' は**図 19** のとおりであり、次式で求めることができる。なお、有村川の場合、 $\alpha=270^\circ$ 、 $\beta=190^\circ$ である。

$$\theta' = \theta + \alpha - \beta/2 \quad (7)$$

また、反時計周りに走査する場合は次式のとおりである。

$$\theta' = \alpha + \beta/2 - \theta \quad (8)$$

以上の方法により平面直角座標に変換し、「3.5.1. 流下断面積」で解説した方法により各時刻における流下断面積を求める。なお、実際には**図 20 (A)** に示すように、据え付け誤差等によりレーザスキャナが傾いている場合があるため、**図 20 (B)** のように据え付け角を補正する必要がある。また、レーザスキャナで計測したデータはレーザスキャナの位置が原点であるため、**図 7** に示す方法で流下断面積を整理する場合、砂防堰堤の水通し右岸端が原点となるよう、水平方向に H 、鉛直方向に V 移動させる必要がある。具体的なデータの処理は付録としてプログラムのコードを添付した。

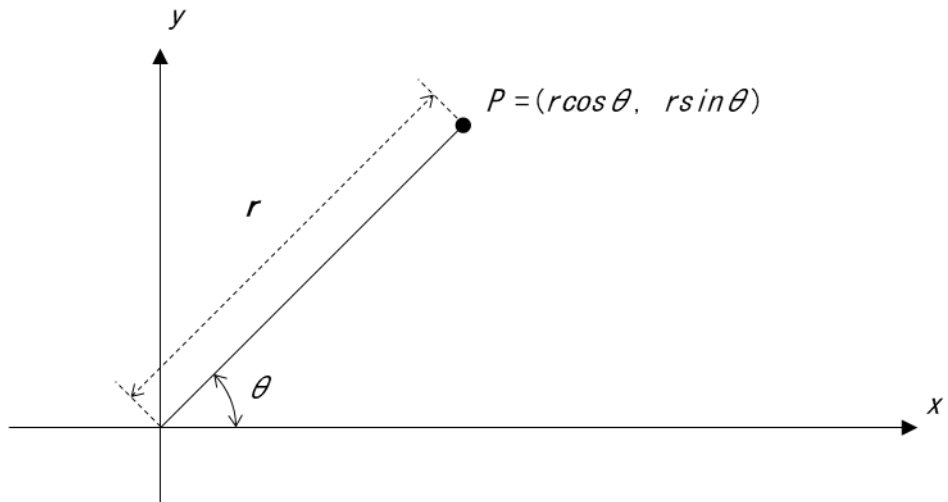


図 18 極座標の概要

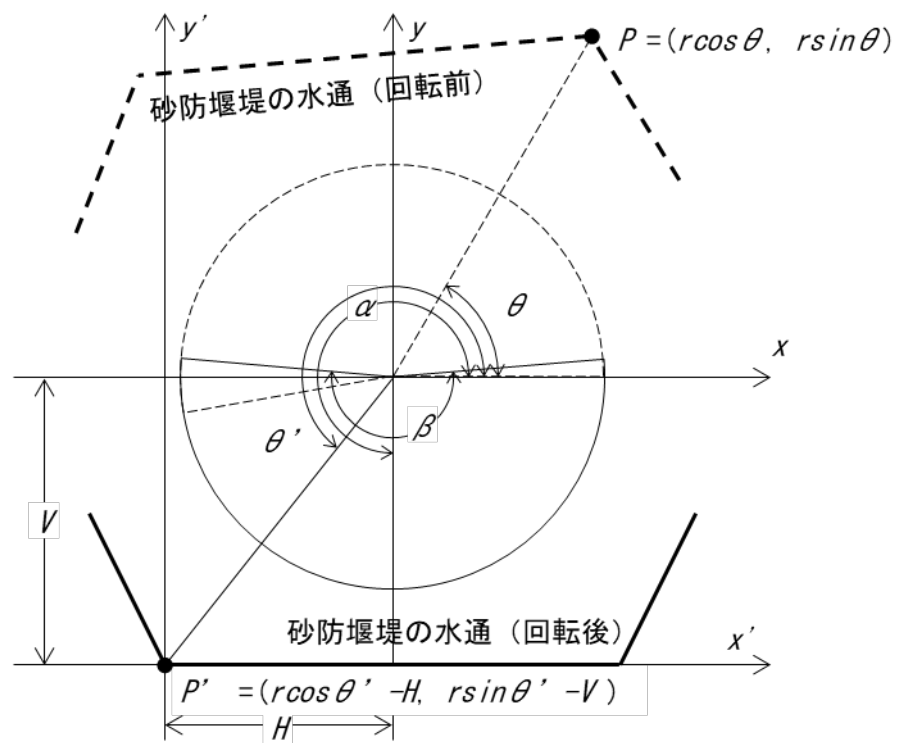


図 19 座標変換の概要

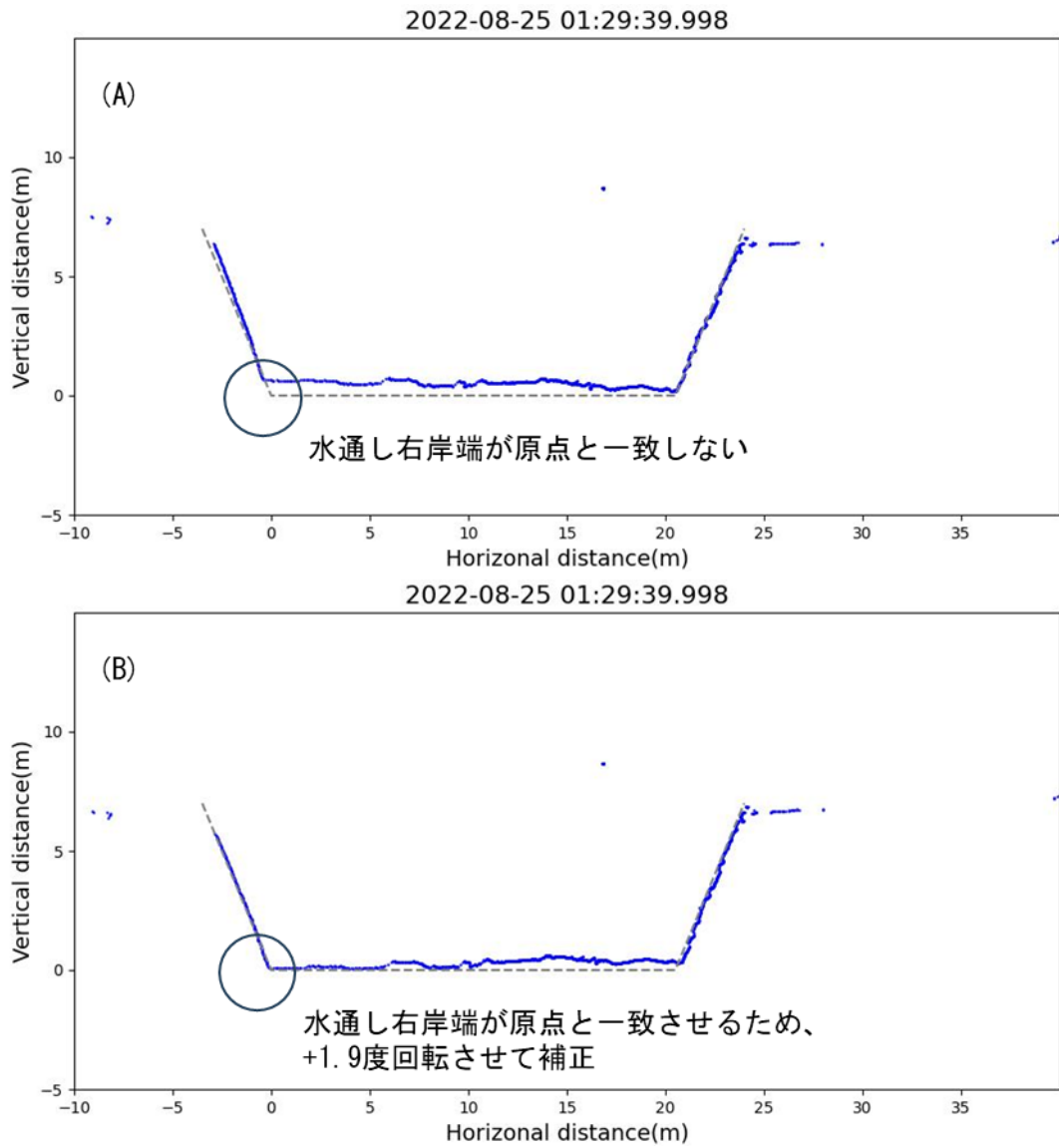


図 20 LiDAR の据え付け誤差の補正有無の比較

(A) 据え付け角の補正前

(B) 据え付け角の補正後

4.4. 観測データの取得状況とデータの整理結果

2022 年に有村川で計測したデータの事例を示す。2022 年のデータの取得状況は図 21 のとおりである。なお、1 月から 6 月の間は計測機材更新のため、観測を中断しており、6 月上旬に観測を再開した。

事務所の観測によると 2022 年は有村川では 4 回の土石流が発生している²³⁾。このうちの 2 回の土石流をレーザスキャナにより計測した。計測データを前述の方法により土石流のハイドログラフとして整理したものが図 22 である。

概ね 1 か月程度は連続でデータを取得できる場合もあるが、計測開始後数日で計測が中断している場合もある。計測の中断には様々な要因が考えられるものの、エラーログから主な要因はノート PC に接続したポータブルタイプの HDD (Hard Disk Drive) にあると推測した。HDD に起因するエラーが生じる要因はいくつか考えられるが、ここでは以下の 2 つを原因と推察した。まず、電源に起因する問題である。現地で使用している HDD はポータブルタイプであり、電力はノート PC の USB から供給される。何らかの原因でノート PC から HDD への電源供給が不安定となった際に、ノート PC と HDD の接続が切断された可能性がある。この問題に対しては、より消費電力の小さい、SSD (Solid State Drive) に換装することで解決できる可能性がある。次に、温度上昇の問題である。前述のとおり、測域センサは 1 秒間に 20 回の計測を行うため、計測 1 回あたりのデータ量は約 17KB と小さいものの、HDD へは常時データの書き込みが行われる。このため、HDD に負荷がかかり、温度が上昇した結果、リミッターが作動して動作が不安定となった可能性がある。この問題に対しても、スピンドルモータ等動力機構を有さない SSD に換装することで解決できる可能性がある。

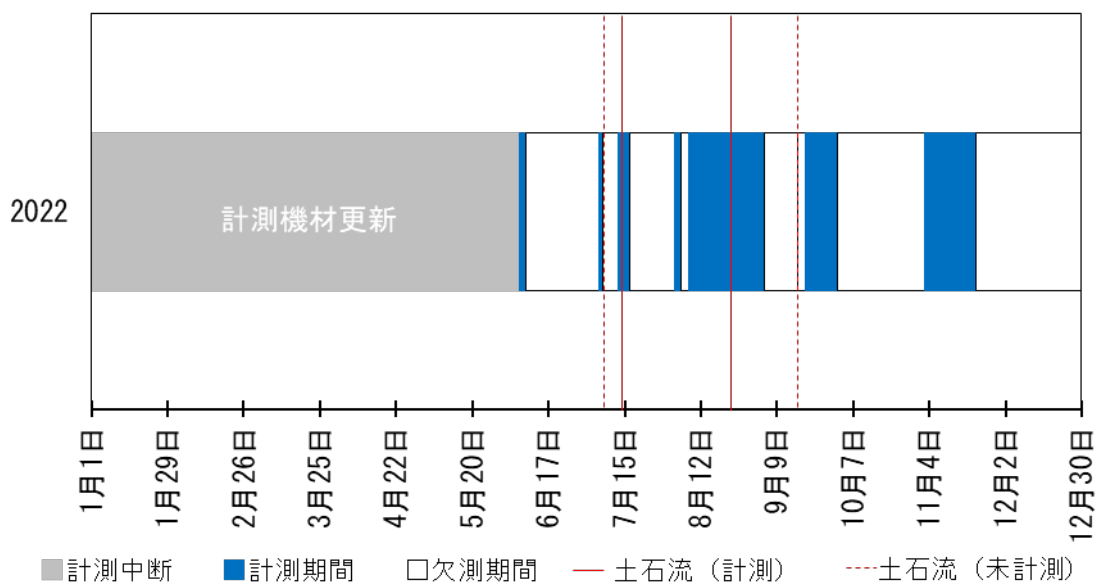


図 21 レーザスキャナによる計測の実施状況

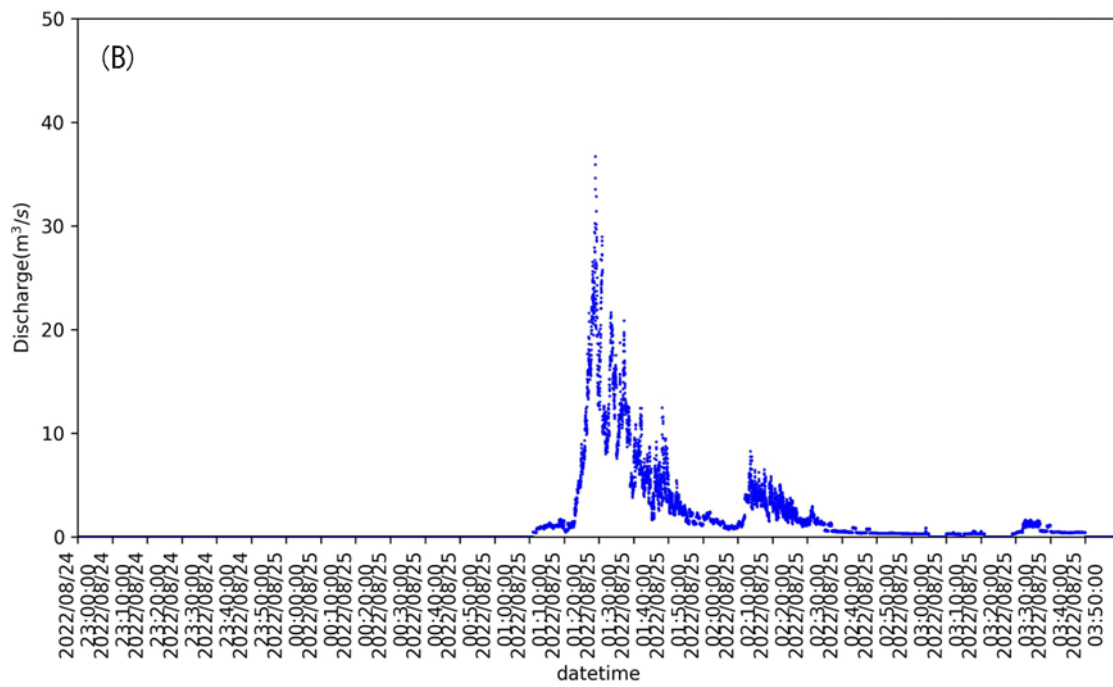
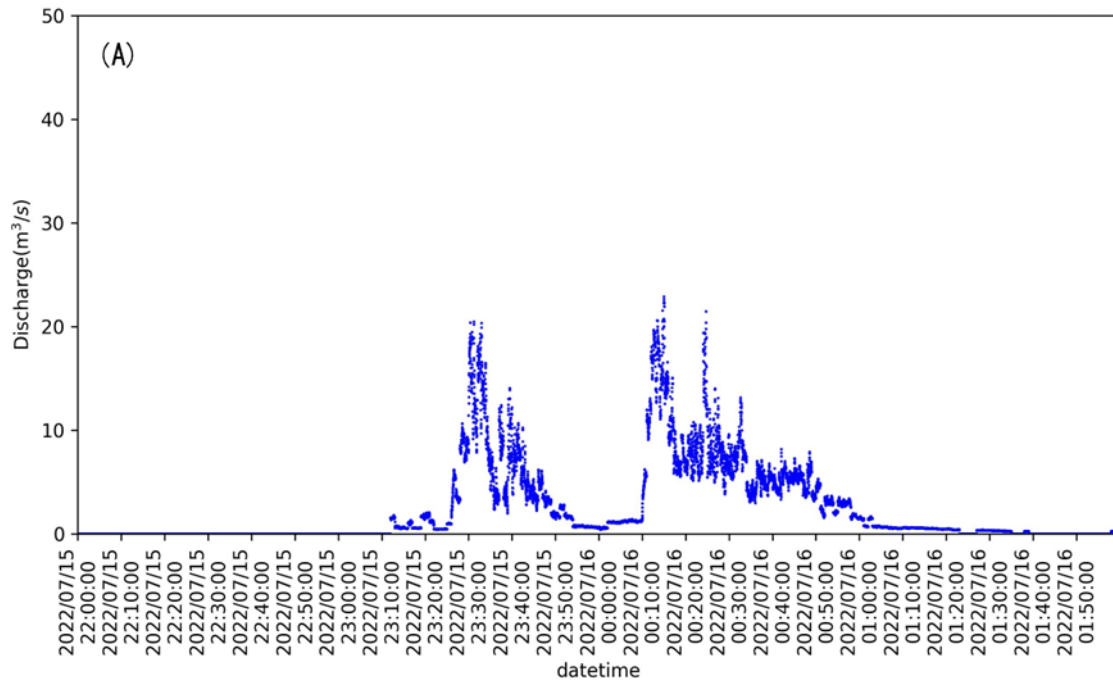


図 22 土石流のハイドログラフ

(A) 2022 年 7 月 20 日～21 日に発生した土石流

(B) 2022 年 8 月 25 日に発生した土石流

5. まとめと今後の課題

5.1. まとめ

レーザスキャナを用いることで従来の土石流観測手法と比較し、以下の点について改善を図ることが可能である。

- ビデオ映像の判読作業が不要であり、データ整理に係る労力の大幅な省力化が可能である。
- 土石流の流下断面積を定量的かつ正確に計測することが可能であり、また、高時間分解能である。
- 夜間及び高降雨強度下等の悪条件であっても、土石流の流下断面積の計測が可能である。
- 土石流の流下中に濘筋が変化した場合であっても、欠測は生じず、流下断面積を正確に計測することが可能である。

5.2. 今後の課題

5.2.1. データ取得の安定性

レーザスキャナの計測データは容量が非常に大きく、長期間にわたる時系列データを安定的に取得できない場合がある。この点については、以下の方法により解決できる可能性があると考えられる。

- データの記録に使用する記憶装置は HDD ではなく、SSD を使用することで、省電力化を実現するとともに、データの書き込み速度を向上させ、安定化を図る。
- 通信環境が整った観測箇所においては、レーザスキャナを制御する PC をインターネットに接続し、データが取得できていない場合に通知するプログラムを実装したり、リモートデスクトップにより稼働状況を定期的に、あるいはまとまった降雨が予報された場合に確認したりする。レーザスキャナによる計測が停止している場合、リモートで PC を操作し、計測を再開することで、データの取得が可能となる。特に、CCTV が既に設置されている地点の近傍で観測を行う場合には、既設の光ケーブルを利用することが有効であると考えられる。
- レーザスキャナの制御用プログラムを開発し、タスクスケジューラによるプログラムの起動およびデータを一定時間ごとに保存できるようにする。あるいは、雨量計と連動させ、一定以上の降雨量に到達した時のみ計測を実行するようにする。近年、レーザスキャナをはじめ多くの機械は ROS (Robot Operating System) に対応しており、自由に制御用アプリケーションを開発することが可能となりつつある。

5.2.2. 土石流の流速計測手法

本資料における土石流の表面流速は、砂防の観測現場で採用実績が多く、既設の非接触型流速計により計測することを念頭に置いている。この方法では前述のとおり 1 点のみの流

速しか計測できない。

一方、前述の通りレーザスキャナを用いた流速の計測方法が提案されている他¹⁶⁾、近年急速に3次元レーザスキャナの技術開発と普及が進んでおり、3次元レーザスキャナによる土石流の観測が可能であるとの報告が行われている²⁴⁾。今後、これらの方法の実用化についても検討していく必要がある。

謝辞

本観測手法の開発にあたっては、室内実験および桜島有村川流域において試験的な計測を開始した当時のチームのメンバーである高橋佑弥氏、能和幸範氏、藤村直樹氏、森田耕司氏、山越隆雄氏、吉永子規氏（五十音順）をはじめ、過去に火山・土石流チームに在籍した多くの関係諸氏の努力と貢献によるところが大きい。

本資料で使用了流速データは国土交通省九州地方整備局大隅河川国道事務所よりご提供いただいた。また、現地観測における機材設置等様々な便宜を図っていただいた。現地計測機材の設置および保守点検は、電通工業株式会社に尽力いただいた。

ここに記して、関係者の皆様に感謝申し上げます。

参考文献

- 1) 奥田節夫、奥西一夫、横山康二、諏訪浩、矢野勝正、大同淳之、奥村武信、中島暢太郎、枝川尚資：土石流の総合的観測 その1. 土石流計測システムの開発と焼岳東麓における1970年の実測例、京都大学防災研究所年報、Vol.14、No.B、pp.691-705、1971
- 2) 里深好文：京都大学防災研究所穂高砂防観測所、砂防学会誌、Vol.59、No.4、pp.60-63、2006、https://doi.org/10.11475/sabo1973.59.4_60
- 3) 水山高久：土石流ピーク流量の経験的な予測、文部省科学研究費重点領域研究、「自然と災害の予測と防災力」研究成果、土石流の発生及び規模の予測に関する研究、pp.54-57、1990
- 4) 国土交通省国土技術政策総合研究所土砂災害研究部砂防研究室：砂防基本計画策定指針（土石流・流木対策編）解説、国土技術政策総合研究所資料第904号、77p、2016、<https://www.nilim.go.jp/lab/bcg/siryounn/tnn0904.htm>、2023年10月2日参照
- 5) Akihiko Ikeda, Takahisa Mizuyama, Takahiro Itoh : Study of prediction methods of debris-flow peak discharge, 7th International Conference on Debris-Flow Hazards Mitigation, pp.709-715、2019、<http://dx.doi.org/10.25676/11124/173152>
- 6) 田原正清：桜島の土石流とその調査、新砂防、Vol.99、pp.26-31、1976、https://doi.org/10.11475/sabo1973.28.4_26
- 7) 池田暁彦、門馬直一、堀内成郎、山田利治：滑川北股沢で発生する土石流について、砂防学会誌、Vol.51、No.2、pp.31-38、1998、https://doi.org/10.11475/sabo1973.51.2_31
- 8) 岡野和行、諏訪浩、植野利康：焼岳上々堀沢における土石流の総合的観測、砂防学会誌、Vol.60、No.6、pp.62-65、2008、https://doi.org/10.11475/sabo1973.60.6_62
- 9) 鈴木豊、鈴木崇：滑川北股沢における土石流の観測体制について、砂防学会誌、Vol.61、No.6、pp.43-46、2009、https://doi.org/10.11475/sabo.61.6_43
- 10) 小川達則、佐々木寿、能和幸範、青山裕：非接触センサの連続観測による土砂動態モニタリングの可能性、砂防学会誌、Vol.61、No.6、pp.27-30、2009、https://doi.org/10.11475/sabo.61.6_27
- 11) 大坂剛、高橋英一、國友優、山越隆雄、能和幸範、木佐洋志、石塚忠範、宇都宮玲、横山康二、水山高久：桜島における土石流荷重計による単位体積重量測定、砂防学会誌、Vol.65、No.6、pp.46-50、https://doi.org/10.11475/sabo.65.6_46
- 12) 清水武志、吉永子規、藤村直樹、石塚忠範：山地河川における土砂移動現象への測域センサの活用、土木技術資料、Vol.57、No.5、pp.10-13、2015
- 13) 清水武志、吉永子規、水谷佑：桜島有村川流域における土石流観測と斜面侵食観測、砂防学会誌、Vol.67、No.5、pp.71-74、2015、https://doi.org/10.11475/sabo.67.5_71
- 14) 清水武志、吉永子規、水谷佑、藤村直樹、高橋佑弥：第6章 2次元レーザスキャナを用いた土石流の流速と表面形状の計測ー鹿児島県桜島有村川流域における土石流観測ー、「砂防の観測の現場を訪ねて」出版プロジェクト委員会編、「砂防の観測の現場を訪

- ねて 3～水の動きの不思議～」、砂防学会、pp.77-86、2022
- 15) 平川泰之、諏訪浩、福田克之、小林範之：土石流の表面流速計測における PIV 適用の試み、砂防学会誌、Vol.59、No.2、pp.49-54、2006、https://doi.org/10.11475/sabo1973.59.2_49
 - 16) 吉永子規、清水武志、水谷佑、高橋佑弥、藤村直樹、泉山寛明、石塚忠範：レーザ測距儀を用いたナップ飛距離及び水深の計測方法の提案と流速推定への応用、砂防学会誌、Vol.70、No.1、pp.46-53、2017、https://doi.org/10.11475/sabo.70.1_46
 - 17) 奥田節夫、諏訪浩、横山康二、枝川尚資：土石流の総合的計測システム、写真測量、Vol.11、No.1、pp.13-20、1972、<https://doi.org/10.4287/jsprs1962.11.13>
 - 18) 佐藤功太、大矢晃久：マルチエコーを取得可能な測域センサの性質に関する研究 -雨、煙、シャッターがある環境における測定結果、ロボティクス・メカトロニクス講演会講演概要集、2012、https://doi.org/10.1299/jsmermd.2012._1A1-A07_1
 - 19) 国土交通省水管理・国土保全局河川計画課河川情報企画室：非接触型流速計測法の手引き（案）、p.3-6、2023、https://www.mlit.go.jp/river/shishin_guideline/kasen/pdf/ryusoku_tebiki.pdf、2023年10月2日参照
 - 20) 国土交通省国土技術政策総合研究所土砂災害研究部砂防研究室：土石流・流木対策設計技術指針 解説、国土技術政策総合研究所資料第905号、p.18、2016、<https://www.nilim.go.jp/lab/bcg/siryounn/tnn0905.htm>、2023年10月2日参照
 - 21) 高橋保：土石流の発生と流動に関する研究、京都大学防災研究所年報第20号B-2、pp.405-435、1977
 - 22) 水山高久、上原信司：土石流の水深と流速の観測結果の検討、砂防学会誌、Vol.37、No.4、pp.23-26、1984、https://doi.org/10.11475/sabo1973.37.4_23
 - 23) 国土交通省九州地方整備局大隅河川国道事務所：令和5年土石流調査情報（桜島地域）第1報（通算第272号）、2023、<https://www.qsr.mlit.go.jp/osumi/files/Content/1322/pdf/230105report.pdf>、2023年10月2日参照
 - 24) Jordan Aaron、Raffaele Spielmann、Brian W. Mc Ardell、Christoph Graf：High-Frequency 3D LiDAR Measurements of a Debris Flow: A Novel Method to Investigate the Dynamics of Full-Scale Events in the Field、Geophysical Research Letters、Volume 50、Issue 5、2023、<https://doi.org/10.1029/2022GL102373>

A. 付録

ここでは、事例紹介で用いたデータ処理および可視化プログラムを紹介する。

A. 1. データ整理用プログラム

ディレクトリ構成と各プログラムにおける作業内容は以下のとおりである。

└data : 各プログラムによる出力結果を保存するディレクトリ

├txt : 各時刻における計測データを格納したディレクトリ

└jpg : 可視化用の jpeg ファイルを保存するディレクトリ

└area.py : 各時刻における計測データから流下断面積を求める

└median.py : 1 秒間の流下断面積データの中央値を求める

└resampling.py : 1 分間隔の流速データを 1 秒間隔の流速データにリサンプリングする

└discharge.py : 各時刻の流下断面積と流速を乗じて土石流の流量を求める

└hydrograph.py : 土石流のハイドログラフを描画する

└txt2jpg.py : 各時刻における計測データから流下断面図を作成する

└animation.py : 各時刻の流下断面図を結合して mp4 形式のアニメーションを作成する

A. 2. プログラムの動作確認済み実行環境等

コードの記述および動作確認は Windows 10 Professional にインストールした Python 3.9 で行った。プログラムの実行にあたり、標準ライブラリ以外に以下のライブラリをインストールする必要がある。

- pandas
- Matplotlib
- NumPy
- OpenCV
- Pillow

A. 3. 計測データの下処理

桜島に設置したレーザスキャナで記録したデータの構造は図 23 のとおりである。データの先頭 30 行分はレーザスキャナ本体や制御ソフトに関する情報が格納されており、流下断面面積を計算するにあたり不要な情報であるため、削除する。ファイルは独自の拡張子「.ubh」を有するデータであるが、テキストファイルであるため、テキストエディタで閲覧したり、編集したりすることが可能である。ただし、長時間にわたり計測した場合、データ容量が 1GB を超え、Windows に搭載されているメモ帳等標準的なテキストエディタで編集することはできない。このため、大容量のファイルの表示、編集が可能なエディタを使用するか、プログラムで処理する必要がある。なお、後述の「area.py」および「figure.py」は入力データの文字コードを utf-8 と指定しているので、データ保存時の文字コードの指定に留意が必要である。

データの先頭 30 行を削除すると、1 回の計測毎に 6 行分の情報が記録されている。データの構造を詳述すると、下記のとおりである。

```
[timestamp]
1839741
[logtime]
2022-08-25 01:00:00.001
[scan]
176;176;184;184;194;182;190;182;182;176;172;166;156;156;176;...
```

桜島で使用したレーザスキャナのサンプリングレートは 20Hz であるため、例えば 10 秒間の計測データであれば、上記 6 行分のデータが 20 組、合計 120 行記録されている。

「area.py」および「figure.py」ではテキストデータの形式で保存された上記の情報を使用する。このため、計測データを 6 行毎に分割し、「.txt」形式のファイルで保存する必要がある。この作業は、Windows の環境では PowerShell を使用することで実行が可能である。上述のとおり、データの先頭 30 行分を削除したデータ「data.txt」を 6 行ごとに分割する場合、PowerShell で実行するコードは下記のとおりである。

```
~~~~
mkdir txt
$i=0; cat ./data.txt -ReadCount 6 | % { $_ | Set-Content -Encoding UTF8 ./txt/0.05sec_$.txt;$i++ }
~~~~
```

なお、[captureMode] が距離だけを計測する GD_Capture_mode ではない場合、[scan] の計測データフォーマットは「距離|受光強度;」となる。本資料で添付したプログラムの利用に

あたっては注意が必要である。

```
[appName]
UrgBenriPlus
[appVersion]
2.2.0
[firmwareVersion]
1.0.1 (2016-10-12)
[model]
UXM-30LAH-EWA
[frontStep]
760
[minDistance]
23
[maxDistance]
120000
[motorSpeed]
1200
[totalSteps]
2880
[captureMode]
GD_Capture_mode
[scanMsec]
50
[startStep]
0
[endStep]
1520
[grouping]
1
[serialNumber]
H1704615
[timestamp]
14016499
[logtime]
2017-10-20 14:33:35.117
[scan]
37776;37666;37618;37434;37380;37104;36906;36852;36852;36800;36740;36720;36698;36658;36588;36522;36464;36346;36248;3
6208;36178;36150;36142;36148;36148;36148;36082;36010;35986;35986;35976;35976;35932;35906;35880;35830;35702;35684;35
588;35340;35341;120000;120000;120000;120000;34400;34380;34380;34318;34226;34026;33928;33890;33866;33848;33840;33840
;33828;33828;33780;33588;33418;33366;33366;33398;33398;32284;32285;32284;32424;32424;32424;32412;32406;32406;32350;3
30898;30898;30918;30918;30906;30906;31032;31032;30902;30902;31008;31084;31084;30594;30594;31172;31268;31268;31190;3
1008;30760;30722;30650;30544;30232;30232;30312;30312;30248;30184;30110;30010;30011;29914;120000;29172;28734;28612;2
8508;28352;28180;28181;28062;27322;27314;27312;27296;27254;27255;27254;27214;26866;26750;26750;120000;120000;120000
;26544;26544;120000;120000;120000;120000;120000;120000;120000;120000;120000;120000;120000;120000;120000;1200
00;120000;120000;120000;120000;18358;17542;17504;17504;17504;17460;17394;17278;17202;17146;16924;16834;16682
;16626;16574;16560;16508;16412;16340;16194;16156;16098;15962;15848;15802;15770;15672;15628;15508;15382;15346;15222;
14974;14656;14602;14490;14400;14374;14240;14174;14132;14116;14026;14010;13966;13924;13864;13760;13724;13678;13642;1
3576;13528;13498;13466;13456;13432;13432;13454;13454;13440;13388;13388;13414;13436;13544;13544;13514;13506;13494;13
476;13476;13462;13458;13400;13394;13400;13366;13340;13292;13210;13194;13160;13160;13144;13128;13126;13098;13076;130
52;13040;13040;13042;13042;13050;13058;13058;13070;13102;13140;13176;13204;13228;13236;13272;13334;13340;13382;1344
4;13468;13468;13458;13458;13438;13438;13358;13272;13148;13120;13016;12954;12944;12936;12922;12894;12828;12802;12732
...

```

図 23 桜島に設置したレーザスキャナで記録したデータの構造

A. 4. 各プログラムの解説

A. 4.1 プログラムの実行

プログラムを保存したディレクトリでコマンドプロンプトまたは PowerShell で実行する。以下に実行コードの例を示す。

```
python area.py
```

また、各プログラムはモジュール化されているため、別のプログラムにおいてモジュールとしてインポートおよび実行することが可能である。以下に Python のプログラムにおいて「area.py」をインポート及び実行するコードを例示する。

```
from area import area  
area()
```

A. 4.2 「area.py」

各時刻の計測データから流下断面積を求めるプログラムである。

多数のテキストファイルの読み込みおよび各計算を高速に実行できるよう、数値計算を効率的に行うためのライブラリである NumPy を主に使用している。具体的なコードの解説は以下のとおりである。

ループ処理で以下の作業を繰り返し行い、「./data/txt」内の全 txt ファイルについて処理する。

- 「./data/txt」内の txt ファイルのデータを読み込み、極座標を直行直角座標に変換する。
- 直行直角座標に変換したデータから、砂防堰堤の水通し天端の範囲の点群データを抽出し、任意の計測点と隣り合う点および x 軸で構成される台形について台形公式を適用して面積を求める。
- 水通し天端内の点群データについて同様の計算を行い、積算して流下断面積とする。
- 得られた流下断面積と時刻のデータは順次リストに格納する。

「./data/txt」内の全 txt ファイルの処理が完了したら、リストを csv ファイルに書き込んで出力する。出力ファイルは「./data/area.csv」として保存される。以上の手順をフローチャートで整理すると図 24 のとおりである。

なお、レーザスキャナの据え付け時の傾き γ を補正するための値は現状では試行錯誤的に求める必要がある。改善方策として、袖天端等の不動点の情報を元に、ICP (Iterative Closest Point) によるマッチングを実装することで自動化が可能であると考えられるが、この点は今後の課題である。

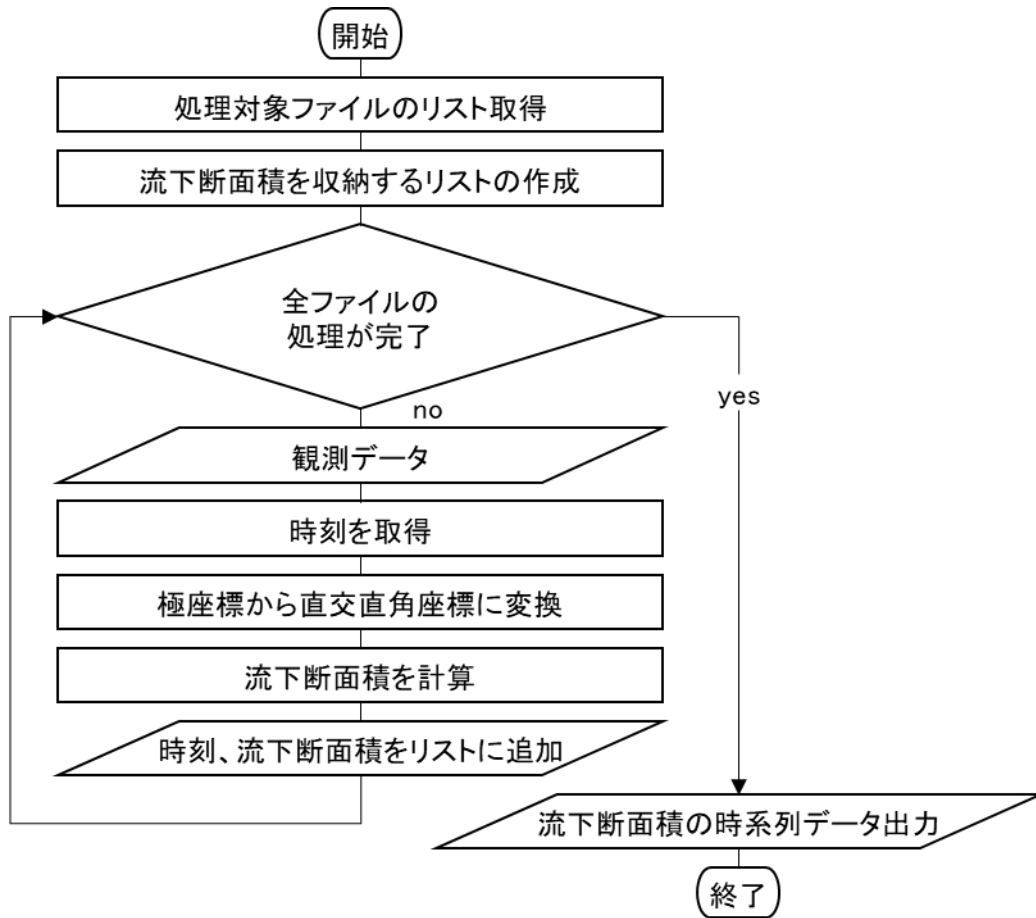


図 24 「area.py」の処理に関するフローチャート

A. 4.3 「median.py」

「area.py」で求めた流下断面積等の時系列データから、1秒間における流下断面積等の中央値を求めるプログラムである。

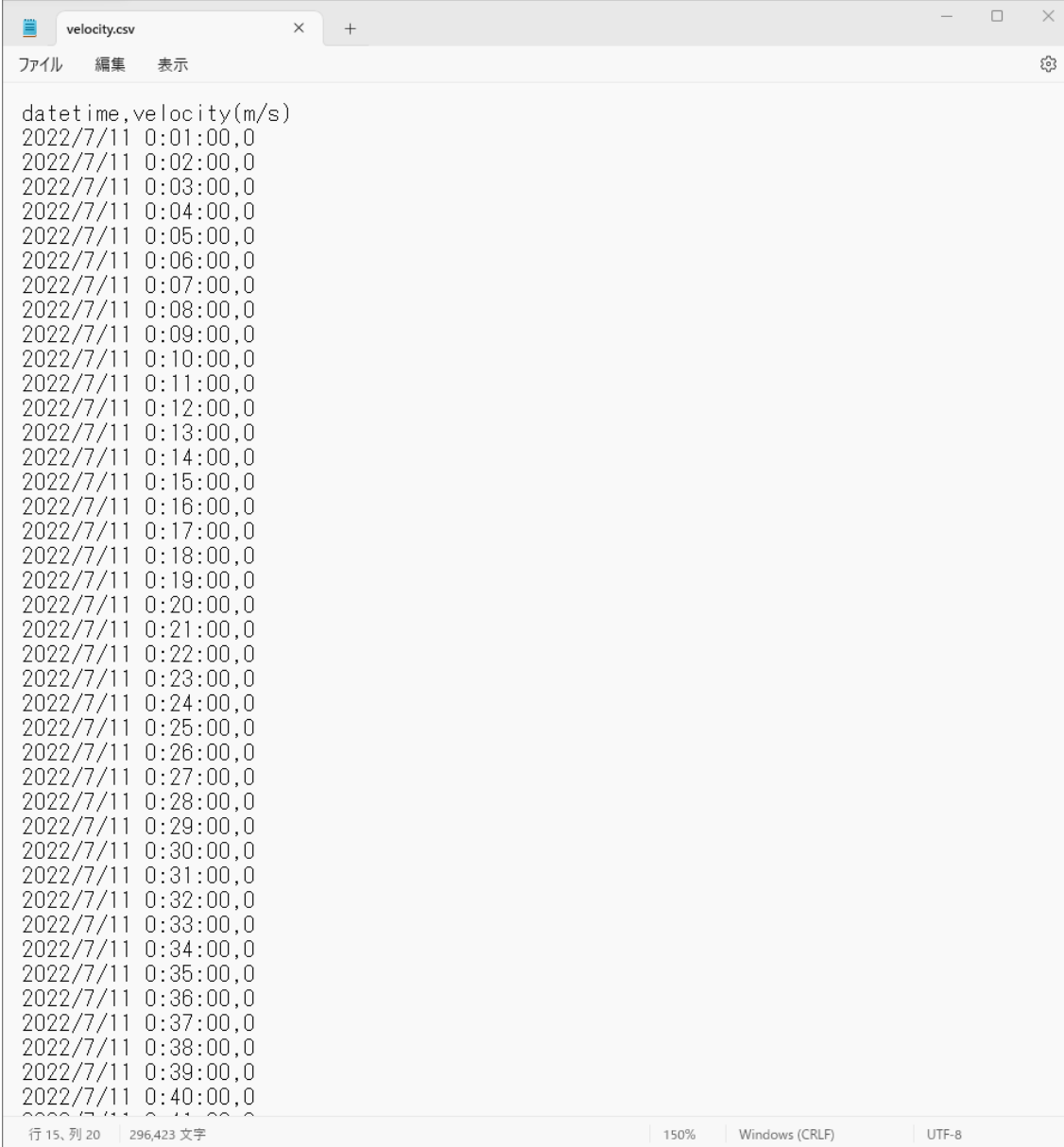
時系列データを容易に扱うことが可能なライブラリである pandas を使用している。入力データは「area.py」により出力される「./data/area.csv」である。出力データは「./data/area_med.csv」として保存される。

A. 4.4 「resampling.py」

超音波流速計のデータが1分間隔であるため、1秒間隔のデータに補完処理するプログラムである。また、超音波流速計のデータは土石流の表面流速であるため、既往の方法²²⁾に倣い表面流速に3/5を乗じて土石流の平均流速とする。

時系列データを容易に扱うことが可能なライブラリである pandas を使用している。入力データとして「./data」ディレクトリに図 25 のとおり datetime 形式のデータ、流速データの順でコンマ区切りの順となるよう、下処理を行った「velocity.csv」を保存する。出力データ

は「./data/velocity_rs.csv」として保存される。



```
velocity.csv
ファイル 編集 表示
datetime,velocity(m/s)
2022/7/11 0:01:00,0
2022/7/11 0:02:00,0
2022/7/11 0:03:00,0
2022/7/11 0:04:00,0
2022/7/11 0:05:00,0
2022/7/11 0:06:00,0
2022/7/11 0:07:00,0
2022/7/11 0:08:00,0
2022/7/11 0:09:00,0
2022/7/11 0:10:00,0
2022/7/11 0:11:00,0
2022/7/11 0:12:00,0
2022/7/11 0:13:00,0
2022/7/11 0:14:00,0
2022/7/11 0:15:00,0
2022/7/11 0:16:00,0
2022/7/11 0:17:00,0
2022/7/11 0:18:00,0
2022/7/11 0:19:00,0
2022/7/11 0:20:00,0
2022/7/11 0:21:00,0
2022/7/11 0:22:00,0
2022/7/11 0:23:00,0
2022/7/11 0:24:00,0
2022/7/11 0:25:00,0
2022/7/11 0:26:00,0
2022/7/11 0:27:00,0
2022/7/11 0:28:00,0
2022/7/11 0:29:00,0
2022/7/11 0:30:00,0
2022/7/11 0:31:00,0
2022/7/11 0:32:00,0
2022/7/11 0:33:00,0
2022/7/11 0:34:00,0
2022/7/11 0:35:00,0
2022/7/11 0:36:00,0
2022/7/11 0:37:00,0
2022/7/11 0:38:00,0
2022/7/11 0:39:00,0
2022/7/11 0:40:00,0
行 15、列 20 296,423 文字 150% Windows (CRLF) UTF-8
```

図 25 下処理を行った超音波流速計のデータ

A. 4.5 「discharge.py」

流下断面積および超音波流速計の時系列データから土石流の流量時系列データを計算するプログラムである。

時系列データを容易に扱うことが可能なライブラリである pandas を使用している。入力データは一連のデータ処理で出力される「./data/area_med.csv」および「./data/velocity_rs.csv」である。出力データは「./data/discharge.csv」として保存される。

A. 4. 6 「hydrograph. py」

土石流の流量時系列データから土石流のハイドログラフを作成するプログラムである。入力データは「./data/discharge.csv」である。出力データは「./data/hydrograph.py」として保存される。

A. 4. 7 「figure. py」

各時刻の計測データから土石流の流下断面図を可視化するプログラムである。「3. 7. 二次元レーザスキャナの維持管理」で述べたとおり、レーザスキャナの光学窓が火砕物等の付着により汚れている場合、流下断面の一部が欠測している場合がある。このため、土石流の発生前後で欠測なく計測ができていないか、確認する必要がある。なお、多量の計測データを処理する場合、オーバーフローする場合がある。このため、流下断面図は1～10秒毎程度の時間間隔で作成する。

ループ処理で以下の作業を繰り返し行い、「./data/txt」内の全txtデータについて処理する。なお、砂防堰堤の構造一般図に倣い、砂防下流側より上流側を臨んだ場合の視点で描画する。

「./data/txt」内のtxtデータを読み込み、極座標を直角座標に変換する。直角座標に変換したデータから、土石流の表面形状を描画する。また、併せて砂防堰堤の水通し断面も描画する。出力データは「./data/jpg」ディレクトリ内にjpeg形式の画像ファイルが順次保存される。ファイル名はレーザスキャナによる計測時刻を抽出して命名される。

A. 4. 8 「animation. py」

「figure.py」により出力したjpegファイルを時系列で結合し、mp4形式のアニメーションファイルを作成することで欠測の有無の確認が容易となる。

入力データは「./data/jpg」ディレクトリ内のjpeg形式の画像ファイルである。出力データは「./data/animation.mp4」として保存される。

B. 付録 プログラムソースコード

目次

| | |
|--------------------|----|
| area.py..... | 2 |
| median.py..... | 6 |
| resampling.py..... | 7 |
| discharge.py..... | 8 |
| hydrograph.py..... | 9 |
| figure.py..... | 10 |
| animation.py..... | 13 |

area.py

```
1 from datetime import datetime
2 import csv
3 import glob
4 import numpy as np
5 import os
6
7 ALPHA = 270
8 BETA = 190
9 GAMMA = -1.9
10 STEP_NUMBER = 1521
11 STEP_ANGLE = 0.125
12 DAM_X0 = -3.5
13 DAM_Y0 = -8.68
14 DAM_I = 0.5
15
16 DAM_XLU = -3.5
17 DAM_XLD = 0
18 DAM_XRU = 24.0
19 DAM_XRD = 20.5
20 DAM_YD = 0
21 DAM_YU = 7
22
23 def area():
24     print('断面積の計算')
25     print('初期設定中')
26
27     CSV_OUTPUT = 0
28
29     def check_data_directory():
30         if os.path.exists('./data'):
31             pass
32         else:
33             os.makedirs('./data')
34     check_data_directory()
35     files = glob.glob('./txt/*.txt')
36
37     theta_list_ar = (ALPHA + BETA/2 + GAMMA) - STEP_ANGLE *
np.arange(STEP_NUMBER)
38
39     area_list = []
40
41     total = len(files)
42     count = 0
43     bar_width = 100
44
45     print('計算開始')
46
47     for file in files:
48         with open(file, 'r', encoding='utf-8') as textfile:
49             content = textfile.readlines()
```

area.py

```

50
51     scan_index = content.index('[scan]\n') + 1
52     scan_value = content[scan_index].strip()
53
54     scan_list = [float(x) for x in scan_value.split(';')]
55     scan_list_ar = np.array(scan_list) / 1000
56
57     x_axis_ar = scan_list_ar * np.cos(np.radians(theta_list_ar))
- DAM_X0
58     y_axis_ar = scan_list_ar * np.sin(np.radians(theta_list_ar))
- DAM_Y0
59
60     data_ar = np.vstack([theta_list_ar, scan_list_ar, x_axis_ar,
y_axis_ar]).T
61
62     datetime_index = content.index('[logtime]\n') + 1
63     datetime_value = content[datetime_index].strip()
64     dt = datetime.strptime(datetime_value, '%Y-%m-%d
%H:%M:%S.%f')
65     dt_str = dt.strftime('%Y%m%d_%H%M%S.%f')
66     filename = str(dt_str) + '.csv'
67
68     data_ar_trim = data_ar[(data_ar[:, 2] >= DAM_XLD)
69                             & (data_ar[:, 2] <= DAM_XRD)
70                             & (data_ar[:, 3] >= DAM_YD)
71                             & (data_ar[:, 3] <= DAM_YU)]
72
73     try:
74         mean_wh = np.mean(data_ar_trim[:, 3])
75
76         value_rh = data_ar_trim[0, 3]
77         value_lh = data_ar_trim[-1, 3]
78
79         trr = value_rh * value_rh * DAM_I / 2
80         trl = value_lh * value_lh * DAM_I / 2
81
82         diff_col = np.diff(data_ar_trim[:, 2])
83         diff_col = np.insert(abs(diff_col), 0, 0)
84         data_ar_trim = np.insert(data_ar_trim, 4, diff_col,
axis=1)
85
86         avg_col = (data_ar_trim[:, 3] + np.roll(data_ar_trim[:,
3], 1)) / 2
87         avg_col = np.insert(avg_col[:-1], -1, 0)
88         data_ar_trim = np.insert(data_ar_trim, 5, avg_col,
axis=1)
89
90         mult_col = data_ar_trim[:, 4] * data_ar_trim[:, 5]
91         mult_col = np.insert(mult_col[:-1], -1, 0)
92         data_ar_trim = np.insert(data_ar_trim, 6, mult_col,
axis=1)
93

```

area.py

```

94         area = np.sum(data_ar_trim[:, 6]) + trr + trl
95
96         ts_data = [datetime_value, mean_wh, area]
97         area_list.append(ts_data)
98
99         if CSV_OUTPUT == 1:
100             def check_data_directory():
101                 if os.path.exists('./data/csv_org'):
102                     pass
103                 else:
104                     os.makedirs('./data/csv_org')
105             check_data_directory()
106             def check_data_directory():
107                 if os.path.exists('./data/csv_area'):
108                     pass
109                 else:
110                     os.makedirs('./data/csv_area')
111             check_data_directory()
112             col_names_org = ['theta(degree)', 'distance(m)',
113 'x_axis(m)', 'y_axis(m)']
114             col_names_area = ['theta(degree)', 'distance(m)',
115 'x_axis(m)', 'y_axis(m)', 'hight(m)', 'length(m)', 'Area(m^2)']
116             np.savetxt('./data/csv_org/' + filename, data_ar,
117 delimiter=',', header=','.join(col_names_org), comments='')
118             np.savetxt('./data/csv_area/' + filename,
119 data_ar_trim, delimiter=',', header=','.join(col_names_area),
120 comments='')
121
122         except IndexError:
123             ts_data = [datetime_value, 0, 0]
124             area_list.append(ts_data)
125
126         count += 1
127         progress = count / total
128         progress_width = int(progress * bar_width)
129         bar = '[' + '#' * progress_width + '.' * (bar_width -
130 progress_width) + ']'
131         percent = int(progress * 100)
132         print('\r' + '進捗: [' + str(percent) + '%] ' + bar,
133 end='')
134
135         print('計算完了')
136
137         with open('./data/area.csv', 'a', encoding = 'utf-8', newline =
138 '') as csvfile:
139             writer = csv.writer(csvfile)
140             header = ['datetime', 'mean_water_hight(m)', 'area(m^2)']
141             writer.writerow(header)
142             for row in area_list:
143                 writer.writerow(row)
144
145         print('csvファイルの出力完了')

```

area.py

```
138 |  
139 | if __name__ == '__main__':  
140 |     area()
```

median.py

```
1 import pandas as pd
2
3 def median():
4     print('1秒間中の平均水深、断面積データの中央値を抽出')
5     df = pd.read_csv('./data/area.csv')
6
7     df['datetime'] = pd.to_datetime(df['datetime'])
8
9     df_1min = df.resample('1S', on='datetime').median()
10    df_1min.to_csv('./data/area_med.csv')
11    print('csvファイルの出力完了')
12
13 if __name__ == '__main__':
14     median()
```


resampling.py

```
1 import glob
2 import os
3 import pandas as pd
4 import sys
5
6 ALPHA = 3/5
7
8 def rs():
9     def check_data_directory():
10         if os.path.exists('./data/velocity.csv'):
11             pass
12         else:
13             print('流速データが見つかりません。ファイル名を確認するか、ディレクトリ「./data」に流速データを保存してください')
14             sys.exit()
15     check_data_directory()
16
17     print('流速データのリサンプリング')
18
19     df = pd.read_csv('./data/velocity.csv', index_col=0,
20 parse_dates=True, names=['datetime', 'velocity(m/s)'], header=0)
21
22     resampled_df = df.resample('1S').ffill() * ALPHA
23     resampled_df.to_csv('./data/velocity_rs.csv')
24     print('csvファイルの出力完了')
25
26 if __name__ == '__main__':
27     rs()
```

discharge.py

```
1 import pandas as pd
2
3 def dc():
4     print('流量の計算')
5     print('初期設定中')
6
7     df_a = pd.read_csv('./data/area_med.csv', parse_dates=True,
8 index_col=0)
9     df_v = pd.read_csv('./data/velocity_rs.csv', parse_dates=True,
10 index_col=0)
11
12     df_q = df_a.copy()
13     df_q['velocity(m/s)'] = None
14
15     total = df_v.shape[0]
16     count = 0
17     bar_width = 100
18
19     print('計算開始')
20
21     for index, row in df_v.iterrows():
22         if index in df_q.index:
23             df_q.at[index, 'velocity(m/s)'] = row['velocity(m/s)']
24
25             count += 1
26             progress = count / total
27             progress_width = int(progress * bar_width)
28             bar = '[' + '#' * progress_width + '.' * (bar_width -
29 progress_width) + ']'
30             percent = int(progress * 100)
31             print('\r' + '進捗: [ ' + str(percent) + '%] ' + bar,
32 end='')
33
34     print('流量の計算完')
35
36     df_q['discharge(m^3/s)'] = df_q['area(m^2)'] *
37 df_q['velocity(m/s)']
38     df_q.to_csv('./data/discharge.csv')
39
40     print('csvファイルの出力完了')
41
42 if __name__ == '__main__':
43     dc()
```

hydrograph.py

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import matplotlib.dates as mdates
5
6
7 FIG_XSIZE = 10
8 FIG_YSIZE = 5
9 FONT_SIZE = 10
10 MAKER_SIZE = 1
11 YLIM_MIN = 0
12 YLIM_MAX = 50
13 XLIM_STEP = 10
14 YLIM_STEP = 10
15 ROLL = 90
16
17
18 def hg():
19     print('ハイドログラフを作成')
20     df = pd.read_csv('./data/discharge.csv')
21     df['datetime'] = pd.to_datetime(df['datetime'])
22
23     fig, ax = plt.subplots(figsize=(FIG_XSIZE, FIG_YSIZE))
24
25     ax.plot(df['datetime'], df['discharge(m^3/s)'], '.', color='blue',
26 markersize = MAKER_SIZE)
27     ax.set_xlabel('datetime', fontsize = FONT_SIZE)
28     ax.set_ylabel('Discharge($\mathrm{m}^3/\mathrm{s}$)', fontsize = FONT_SIZE)
29     ax.set_xlim(df.iloc[:, 0].min(), df.iloc[:, 0].max())
30     ax.set_ylim([YLIM_MIN, YLIM_MAX])
31     ax.xaxis.set_major_locator(mdates.MinuteLocator(interval =
32 XLIM_STEP))
33     ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y/%m/%d' +
34 '\n' + '%H:%M:%S'))
35     plt.xticks(rotation = ROLL, fontsize = FONT_SIZE)
36     plt.yticks(np.arange(YLIM_MIN, YLIM_MAX + 0.1, YLIM_STEP),
37 fontsize = FONT_SIZE)
38
39     plt.savefig('./data/hydrograph.jpg', dpi=300, bbox_inches='tight')
40
41     print('jpegファイルの出力完了')
42
43 if __name__ == '__main__':
44     hg()
```

figure.py

```
1 from datetime import datetime
2 import glob
3 import matplotlib.pyplot as plt
4 import numpy as np
5 import os
6
7 ALPHA = 270
8 BETA = 190
9 GAMMA = 1.9
10 STEP_NUMBER = 1521
11 STEP_ANGLE = 0.125
12 DAM_X0 = -17.0
13 DAM_Y0 = -8.68
14
15 DAM_XLU = -3.5
16 DAM_XLD = 0
17 DAM_XRU = 24.0
18 DAM_XRD = 20.5
19 DAM_YD = 0
20 DAM_YU = 7
21
22 X_SIZE = 10
23 Y_SIZE = 5
24 PLOT_SIZE = 0.5
25 XLIM_MIN = -10
26 XMIM_MAX = 40
27 YLIM_MIN = -5
28 YLIM_MAX = 15
29 LIM_FONT_SIZE = 10
30 LABEL_FONT_SIZE = 14
31 TITLE_FONT_SIZE = 16
32 DAM_X = [-3.5, 0, 20.5, 24.0]
33 DAM_Y = [ 7.0, 0, 0, 7.0]
34
35 def mkfig():
36     print('断面図の作成')
37     print('初期設定中')
38
39     def check_data_directory():
40         if os.path.exists('./data/jpg'):
41             print('')
42         else:
43             os.makedirs('./data/jpg')
44     check_data_directory()
45
46     files = glob.glob('./txt/*.txt')
47
48     theta_list_ar = (ALPHA - BETA/2 + GAMMA) + STEP_ANGLE *
49     np.arange(STEP_NUMBER)
```

figure.py

```

50     total = len(files)
51     count = 0
52     bar_width = 100
53
54     print('データ処理開始')
55
56     for file in files:
57         with open(file, 'r', encoding='utf-8') as textfile:
58             content = textfile.readlines()
59
60             scan_index = content.index('[scan]\n') + 1
61             scan_value = content[scan_index].strip()
62
63             scan_list = [float(x) for x in scan_value.split(';')]
64             scan_list_ar = np.array(scan_list) / 1000
65
66             x_axis_ar = scan_list_ar * np.cos(np.radians(theta_list_ar)) -
DAM_X0
67             y_axis_ar = scan_list_ar * np.sin(np.radians(theta_list_ar)) -
DAM_Y0
68
69             datetime_index = content.index('[logtime]\n') + 1
70             datetime_value = content[datetime_index].strip()
71             dt = datetime.strptime(datetime_value, '%Y-%m-%d %H:%M:%S.%f')
72             dt_str = dt.strftime('%Y%m%d_%H%M%S.%f')
73
74             fig = plt.figure(figsize=(X_SIZE, Y_SIZE))
75             plt.plot(DAM_X, DAM_Y, linestyle='--', c = 'gray')
76             plt.scatter(x_axis_ar, y_axis_ar, s = PLOT_SIZE, c = 'blue')
77             plt.xlim(XLIM_MIN, XMIM_MAX)
78             plt.ylim(YLIM_MIN, YLIM_MAX)
79             plt.xticks(np.arange(XLIM_MIN, XMIM_MAX, step=5), fontsize =
LIM_FONT_SIZE)
80             plt.yticks(np.arange(YLIM_MIN, YLIM_MAX, step=5), fontsize =
LIM_FONT_SIZE)
81             plt.xlabel('Horizontal distance(m)', fontsize = LABEL_FONTSIZE)
82             plt.ylabel('Vertical distance(m)', fontsize = LABEL_FONTSIZE)
83             plt.title(str(datetime_value), fontsize = TITLE_FONTSIZE)
84             plt.subplots_adjust(left=0.12, bottom=0.1, right=0.99,
top=0.94, wspace=0.2, hspace=0.2)
85             plt.savefig('./data/jpg/' + str(dt_str) + '.jpg')
86             plt.close()
87
88             count += 1
89             progress = count / total
90             progress_width = int(progress * bar_width)
91             bar = '[' + '#' * progress_width + '.' * (bar_width -
progress_width) + ']'
92             percent = int(progress * 100)
93             print('\r' + '進捗: [ ' + str(percent) + '%] ' + bar,
end='')
94

```

figure.py

```
95     print('jpegファイルの出力完了')
96
97     if __name__ == '__main__':
98         mkfig()
```

animation.py

```
1 import numpy as np
2 import cv2
3 from PIL import Image
4 import os
5
6 def mkmov():
7     print('mp4アニメーションの作成')
8     print('初期設定中')
9
10    directory = "./data/jpg"
11
12    file_names = sorted(os.listdir(directory))
13
14    total = len(file_names)
15    count = 0
16    bar_width = 100
17
18    print('フレームリストの作成中')
19
20    frames = []
21    for file_name in file_names:
22        if file_name.endswith(".jpg"):
23            file_path = os.path.join(directory, file_name)
24            frame = Image.open(file_path)
25            frame = cv2.cvtColor(np.array(frame), cv2.COLOR_RGB2BGR)
26            frames.append(frame)
27
28            count += 1
29            progress = count / total
30            progress_width = int(progress * bar_width)
31            bar = '[' + '#' * progress_width + '.' * (bar_width -
progress_width) + ']'
32            percent = int(progress * 100)
33            print('\r' + '進捗: [' + str(percent) + '%] ' + bar,
end='')
34
35            print('mp4アニメーションの作成中')
36            count = 0
37
38            size = frames[0].shape[1::-1]
39
40            fourcc = cv2.VideoWriter_fourcc(*'mp4v')
41            video_writer = cv2.VideoWriter("./data/animation.mp4", fourcc, 30,
size)
42            for frame in frames:
43                video_writer.write(frame)
44
45                count += 1
46                progress = count / total
47                progress_width = int(progress * bar_width)
```

animation.py

```
48     bar = '[' + '#' * progress_width + '.' * (bar_width -
progress_width) + ']'
49     percent = int(progress * 100)
50     print('\r' + '進捗: [ ' + str(percent) + '%] ' + bar,
end='')
51
52     video_writer.release()
53     print('mp4ファイルの出力完了')
54
55 if __name__ == '__main__':
56     mkmov()
```

土木研究所資料
TECHNICAL NOTE of PWRI
No.4445 November 2023

編集・発行 ©国立研究開発法人土木研究所

本資料の転載・複写の問い合わせは

国立研究開発法人土木研究所 企画部 業務課
〒305-8516 茨城県つくば市南原1-6 電話029-879-6754