

## 3次元点群を用いた深層学習による 発破ずりの粒度分布推定法

吉野航平<sup>1,a)</sup> 宍戸英彦<sup>2</sup> 北原格<sup>2</sup>

**概要:** 発破ずりとは、岩山を爆破した後に形成される岩石の山の名称である。鉱山工学の分野では、発破ずりに含まれる各岩石の粒度分布推定が重要な研究課題とされている。本研究では、発破ずりを多視点で撮影した画像群からその3次元点群を復元し、3次元点群を入力、発破ずりの粒度分布を出力とする深層学習を行うことにより、岩石群の粒度分布を推定する手法を提案する。

**キーワード:** 発破ずり, Structure from Motion, 3次元点群, 深層学習, 粒度分布推定

### 1. はじめに

発破とは、鉱山において火薬等を用いて岩石を爆破する作業であり、それによって砕けた岩の山を発破ずりと呼ぶ。発破の主な目的は、周囲の物体を損傷させることなく安全に岩石を爆破させ、用途に適した粒度に砕き、岩石に含まれる資源を採掘することである[1]。鉱山における資源採掘技術は、地下の浅部や地表に露出した鉱床を採掘する露天掘りと、地下に埋もれた鉱床を採掘する坑内掘りの二つに大別されるが、発破は、特に露天掘り鉱山の採掘において、作業性や経済性の点から重要な作業工程であるとされているため[1]、本研究では露天掘り鉱山を主たる対象とする。

発破ずりに含まれる岩石の大きさが目的の粒度に集中している、具体的には、砂などの微粒子のように小さ過ぎたり、運搬に支障をきたす程大き過ぎたりしなければ、効率的な発破が行われたといえるが、対象となる岩盤の内部構造は未知であることが多く、結果として火薬の量や爆破箇所などを適切に設定することは困難な課題である。この問題の解決を目的として、発破後に発破ずりの粒度分布を計測し、発破の設計を解析する研究[2]が進められている。それらの研究の基盤データとして発破ずりの粒度分布の正確な計測法の実現に注目が集まっている。従来の取り組みでは、発破ずりをふるいにかけることによって粒度分布を計測しているが、発破現場において大規模な計測を行う場合、岩石群を一々ふるいにかけることは、多大な処理と時間がかかる上に採掘作業を中断する必要があるため、生産効率低下の観点から現実的とは言い難い。

上述した課題を解決するための研究が盛んに行われている。発破ずりを撮影した2次元画像から岩の領域を検出し、検出領域の面積から粒度分布を推定する手法[3]が実現

されているが、岩石の3次元的な形状による見え方の変化を考慮していないという問題が存在する。レーザレンジセンサを用いて発破ずりの3次元形状を計測し、粒度を推定する手法[4]があるが、計測機材導入に要するコストや知識が、現場での利用時の問題となっている。出川ら[5]は、発破ずりの多視点画像を撮影し、そこから3次元点群を復元することにより、上述した問題の解決を試みた。発破ずりの3次元点群をSupervoxelクラスタリング[6]によって個々の岩に分割し、分割結果にBounding Boxを当てはめ、粒度を算出することにより、粒度分布を推定した。しかし、クラスタリング精度が不十分な場合、岩とクラスタが1対1に対応しないため、粒度分布推定精度が低下するという問題が存在する。

前述のクラスタリングのような機械学習では、人間がデータから抽出した特徴量をもとに、機械が学習を行うが、問題によって有効な特徴量は異なるため、人間は各問題に応じた特徴量を選択・設定する必要がある[7]。例えば、Supervoxelクラスタリングでは、発破ずりの3次元点群から座標、色、法線の情報を特徴量として岩石のクラスタリングを行ったが、結果として、これらの特徴量では正確な岩石の識別に必要な情報が不足しており、正しい粒度分布推定は困難であった。一方、深層学習は、入力データに関する重要な特徴量自体を機械的に設定・学習し、大量のデータから問題固有のパターンを発見することで、問題の種類に依存しない学習を可能とする。具体的には、パーセプトロンと呼ばれる構造を何層も重ねることにより、理論上、任意の関数を表現することが可能であることが知られている[7]。

本稿では、図1に示すように発破ずりの3次元点群を復元し、3次元点群を入力とした深層学習を用いて粒度分布を推定する研究について述べる。出川らと同様に発破ずりの3次元点群を復元することにより、点群の座標、色情報を取得し、それらの情報を入力とする深層学習を行う。深層学習の特性を活かし、発破ずりの3次元点群と粒度分布との複雑な関係性の獲得を試みる。

<sup>1</sup> 筑波大学 理工情報生命学術院 システム情報工学研究群  
Degree Programs in Systems and Information Engineering, Graduate School of  
Science and Technology, Tsukuba University  
<sup>2</sup> 筑波大学 計算科学研究センター  
Center for Computational Sciences, Tsukuba University

a) yoshino.kohei@image.iit.tsukuba.ac.jp

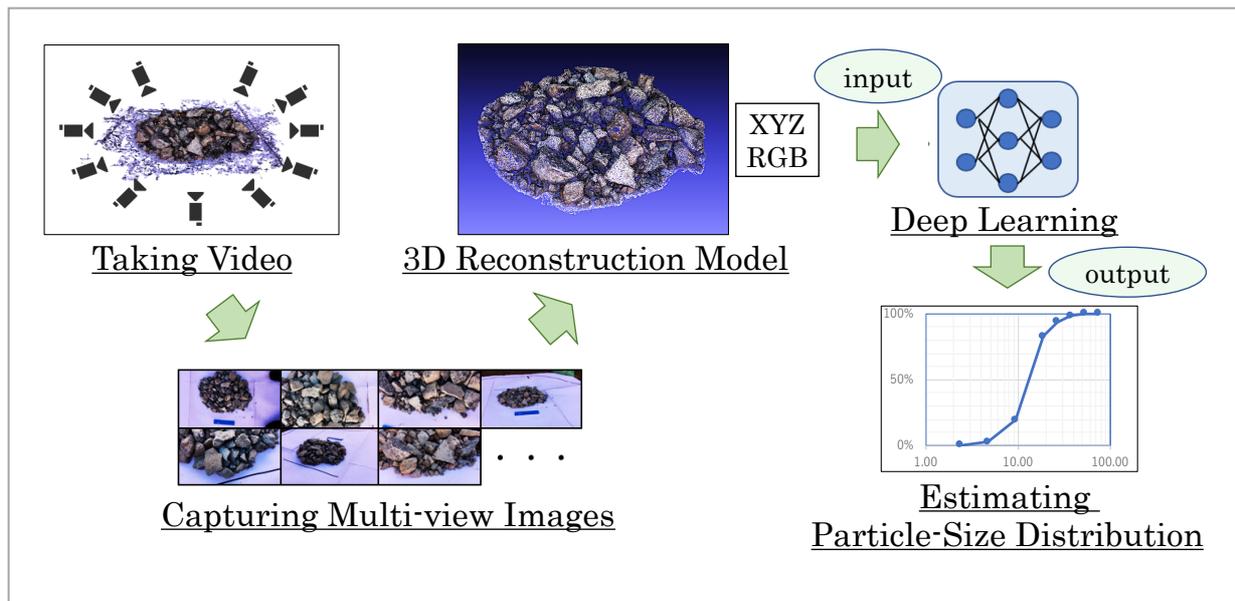


図1 発破ずりの粒度分布推定

以降、2章で発破ずりの粒度分布推定に関する従来手法とその関連手法について述べる。3章で本研究の概要を述べる。4章では発破ずりの3次元モデルの復元、5章では3次元点群を入力とした深層学習について述べる。6章で発破ずりのサンプルによる学習と推定について述べ、7章で本論文をまとめる。

## 2. 関連研究

### 2.1 画像を利用した粒度分布推定手法

発破ずりの粒度分布を推定する代表的な手法として、カメラで撮影した2次元画像を利用する手法がある。Maerzら[3]は、発破ずりの画像から岩の閉領域を検出し、その領域の画素数と同じ面積の円の半径を粒度として粒度分布の推定を行う。しかしこの方法の場合、岩が球状であることを仮定しており、実際の岩が球状でない場合は誤差が大きくなるうえ、岩石の3次元形状を考慮していないため、正確な粒度分布の推定が困難であると考えられる。さらに、粒度を計測するためには、発破ずりの付近に大きさが既知の物体を置いて撮影する必要がある、発破ずりの崩壊によって作業員が怪我をする危険性がある。

### 2.2 3次元モデルを利用した粒度分布推定手法

発破ずりの3次元形状を計測して粒度分布を推定することで、2次元画像情報に基づいた粒度分布推定の問題を解決することが可能である。Onedderaら[4]は、レーザレンジセンサを用いて発破ずりの3次元形状を復元し、発破ずりの3次元モデルを生成し、粒度分布を計算している。この手法では、3次元モデルを並行投影したレンダリング画像を領域分割することで、各岩の閉領域を計算し粒度分布を推定している。しかし、並行投影によって生成した2次元

情報における推定処理のため、3次元形状による見え方の変化は考慮していない。Liuら[8]は、ステレオカメラを用いて発破ずりを撮影し、粒度分布を推定しているが、粒度分布計測の原理はOnedderaらと同様な方法であるため、3次元形状を考慮していないという問題が残る。

上述のレーザレンジセンサを用いた3次元形状の復元では、物体にレーザを照射し、反射して戻ってくるまでの時間を計測することで距離を推定するTime-of-Flight(ToF)[9]という計測原理を利用している。この方法ではオクルージョンによって物体の一部が計測できないという問題があるため、複数地点から物体を計測する必要がある。しかし、実際の発破ずりの現場のような大規模空間を対象とした計測を行う場合、レーザレンジセンサを何度も移動させながらの計測が必要となる。

一方で、持ち運びが簡単な可視光カメラのみを用いて3次元形状を復元する研究も行われている。Structure from Motion (SfM)[10]は、対象の物体を様々な角度から撮影した多視点画像から撮影したカメラの位置姿勢と、物体の3次元形状を同時に復元する手法である。この手法を利用することによって、大規模な発破現場においても効率的に撮影を行い、3次元復元が可能になる。SfMによる復元結果は、空間的に疎であることが多いが、Patch-based Multi-View Stereo (PMVS)[11]を適用することで、より密な3次元形状を復元することが可能である。

出川らは、上述のSfMとPMVSによって3次元形状を復元し、発破ずりの粒度分布を推定している。復元された3次元形状は点の集合である3次元点群によって表現され、Supervoxelクラスタリング法によって、3次元的に各岩石に分割した後、物体を囲む最小直方体であるBounding Boxを当てはめることで、各岩石の大きさを求め、3次元的に

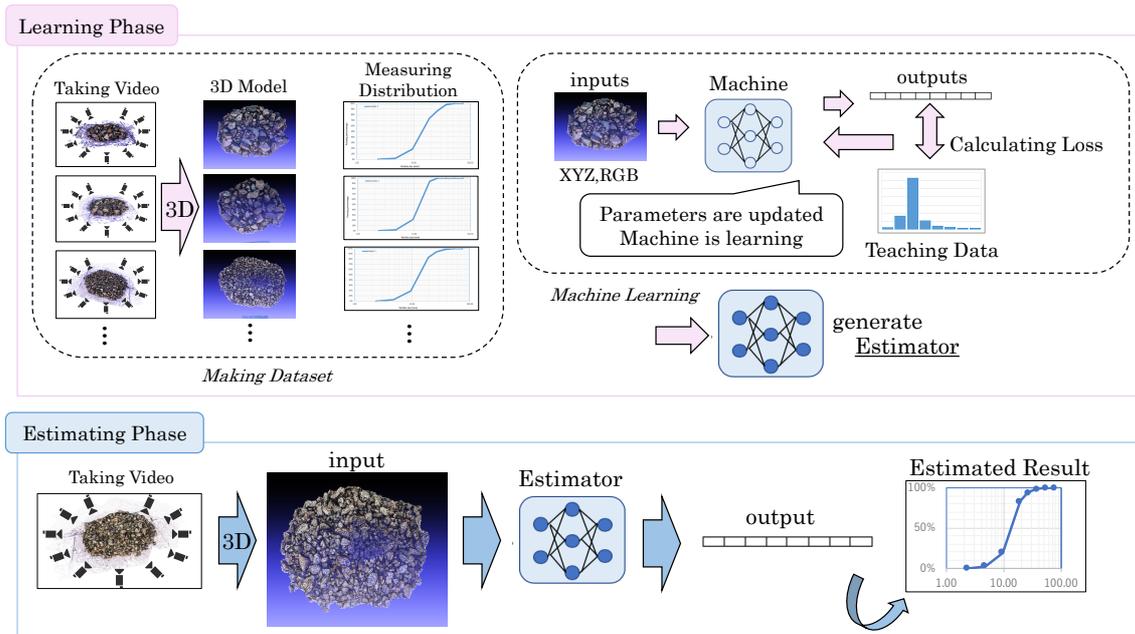


図2 粒度分布推定の流れ

粒度分布を推定している。Supervoxel クラスタリングとは、点同士のユークリッド距離、色、法線ベクトルに任意の重みを与え、k-means 法[12]の目的関数を用いて点をクラスタリングする方法である。しかし、岩とクラスタが一対一に対応しないという問題があり、正確な粒度分布を計測することは困難であった。そこで、同一の岩石にある複数のクラスタを統合するために、発破ずりの多視点画像に LOG (Laplacian Of Gaussian) フィルタを適用し、岩のエッジの検出を行った。二つのクラスタの境界線上に、ある閾値以上の箇所があれば、そのクラスタ同士は別の岩であると判断し、逆に閾値以上の箇所が無ければ、二つのクラスタは同じ岩であると判断して二つのクラスタの統合を行う。査ら[13]は、岩の表面の法線ベクトルを利用し、岩の凹凸を調べることで、隣接する平面の接続関係に基づいたクラスタリングを行い、出川らの手法の改良に取り組んだ。しかし、粒度分布推定の精度には以前改善の余地がある。

### 2.3 3次元点群を入力とした深層学習

深層学習を含むニューラルネットワークにおいて3次元形状のデータを扱う場合、点群をボクセルに置換し、3次元 Convolutional Neural Network (CNN)により畳み込みを行う手法が一般的であったが、置換する過程で点群情報の欠損があることが問題であった。Charles ら[14]は、3次元点群を加工することなく点群のまま深層学習を行い、3次元形状のクラス分類を実現する PointNet と呼ばれるモデルを提案した。PointNet は点群のまま深層学習を行う場合に起こる、点の順不同性の問題等を考慮した構造になっており、近年ではこの手法をはじめとした3次元点群を入力とする深層学習の研究が盛んに行われるようになった。

3次元点群を深層学習の入力として扱うことができると

いう点で、PointNet は本研究でも有効であると考え、このモデルを応用して、発破ずりの粒度分布を推定する手法を提案する。

## 3. 3次元点群を用いた深層学習による発破ずりの粒度分布推定法

本節では、図2に示すように、多視点画像から発破ずりの3次元点群を復元し、それを入力とした深層学習によって粒度分布を推定する手法の概要について述べる。

まず、持ち運びが容易なモバイルカメラによって発破ずりを多方向から撮影し、多視点画像を取得する。その後、3次元復元手法によって発破ずりの3次元点群を生成する。さらに、3次元点群を入力とした深層学習を用いて、粒度分布を推定する。深層学習には学習フェーズと推定フェーズがある。入力を3次元点群の座標と色情報、出力を離散確率分布である8次元ベクトルとして学習を行う。発破ずりの粒度分布は八つの粒度に分割されており、これを教師データとして利用する。学習を終えた後、学習フェーズには含まれないデータで推定を行い、精度を検証する。

## 4. 発破ずりの3次元復元

### 4.1 多視点画像の撮影

SfM および PMVS で3次元形状復元するための多視点画像を撮影する。実際の発破現場の大きさは数10m から数100m に及ぶことが多く、カメラを移動しながら一枚ずつ静止画像を撮影するのは効率的ではない。我々は、ビデオカメラを利用して図3に示すように発破ずりの周囲を移動

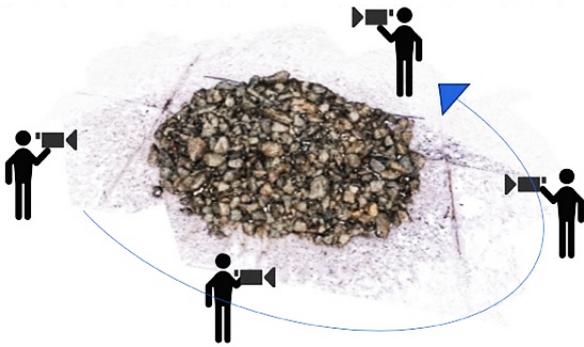


図3 発破ずりの多視点画像撮影

しながら動画を撮影し、動画をフレーム分割することで多視点画像を効率的に取得している。

#### 4.2.3 多視点画像からの3次元復元

発破ずりの多視点画像から3次元形状を復元する。まず、SfMによる復元を行う。画像間で特徴点の対応点探索を行い、対応点をもとに5点アルゴリズム[15]等を用いてカメラのモーション推定を行う。その後、求めたカメラモーションをもとにステレオビジョンを利用して3次元形状を復元する。

SfMによって復元された3次元点群は、空間的に疎であることが多く、粒度分布を推定するのに十分とは言えない。そこで、PMVSを用いてより密な3次元点群を生成する。PMVSでは、画像間の対応点探索を行い、対応点の3次元位置に物体の表面を近似する小領域であるパッチを生成し、密な3次元点群を復元する。図4に復元結果の一例を示す。

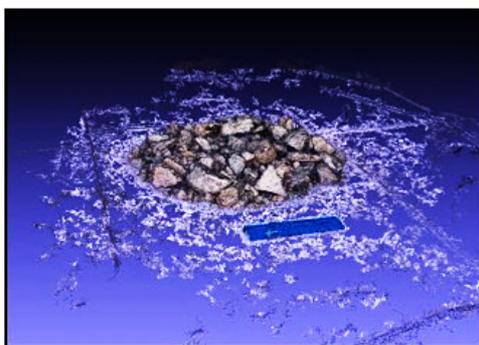
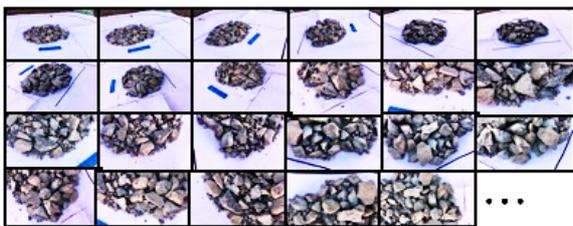


図4(上) 発破ずりの多視点画像  
 (下) 発破ずりの3次元復元結果

## 5. 3次元点群を入力とした深層学習

### 5.1 PointNetの概要

近年、3次元点群データを入力とした深層学習の研究が盛んに行われており、PointNetはその一つである。従来では、点群情報は深層学習において、標準的なフォーマットではないとされ、ボクセルグリッドや画像を利用した手法[16][17][18]が典型的であったが、PointNetは点群情報を入力とした深層学習により、3次元物体のクラス分類、セグメンテーション、セマンティックセグメンテーションを実現した。

点群情報を扱う場合に生じる問題がいくつかあるが、その一つに点の順不同性がある。ボクセルグリッドや、ピクセルが並ぶ画像と異なり、点群は単なる点の集合であり、規則的に並んでいるわけではない。同じ点群でも、N点で表現されている場合では点を順番に並べるときN!通りの並べ方があり、それらは全て同じ点群を表現しているということを理解する必要がある。この問題を解決するために、PointNetでは対称関数を利用している。対称関数とは、和や積などの計算結果が数値の順に依存しない関数であり、具体的な方法は図5に示す通りである。各点ごとに独立した変換や学習を行った後、Max Pooling処理を行うことで、点の順序に依存せず、点群のグローバルな特徴を抽出している。

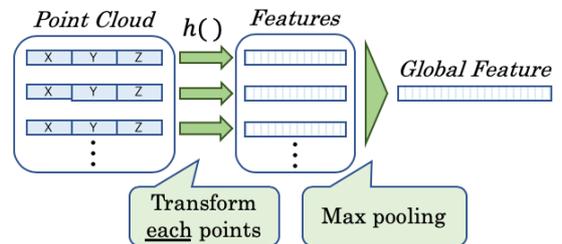


図5 対称関数の利用

点群は順不同性の問題の他に、剛体変換の普遍性の問題がある。同じ形状の点群でも、位置や姿勢が異なる場合があるが、同じ点群として認識する必要がある。そこでPointNetでは、T-Netと呼ばれるネットワークを利用して、3×3の行列を推定し、点群の剛体変換を行っている。T-Netの構造はPointNetのネットワークの構造と類似しており、Max Pooling層、MLP (Multi-Layer Perceptron)、畳み込み層等から構成される。行列の推定による損失を、クラス分類の損失に加えることで、行列の推定についても学習が行われ、実際に論文の中では、T-Netによる推定精度の向上が認められている。

PointNetは主に3次元物体のクラス分類を実現する学習ネットワークであるが、上述のような3次元点群を扱う際

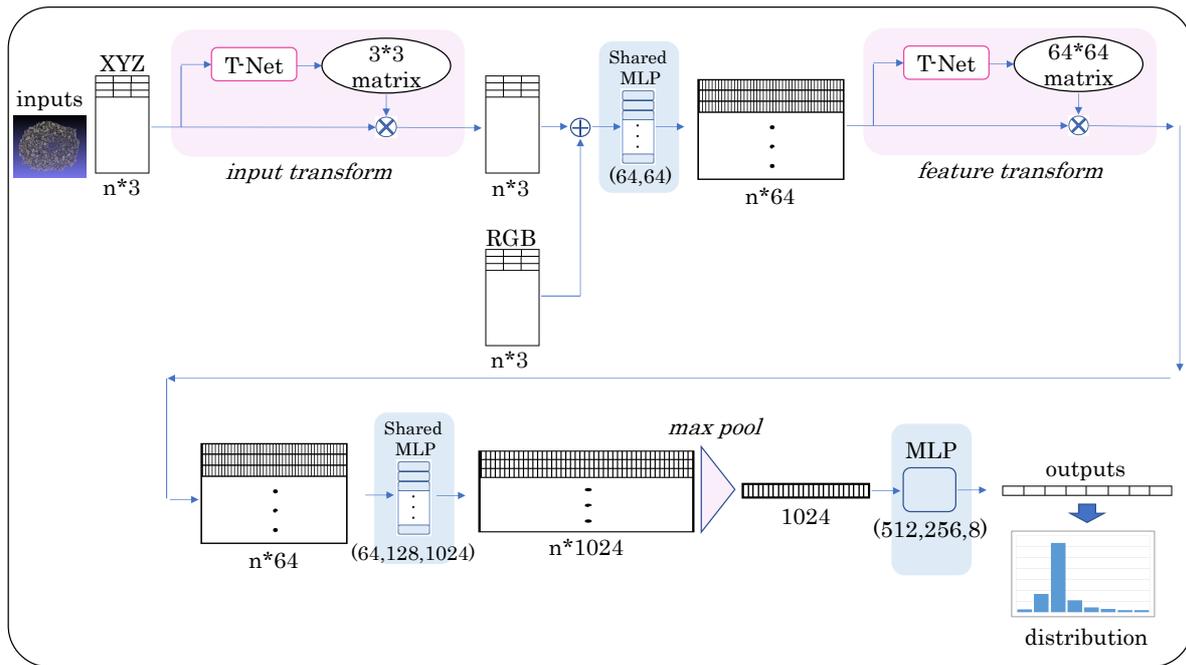


図6 構築したネットワーク

に必要な不可欠な処理を活用するとともに、推定に適した構造に改変することにより、発破ずりの粒度分布を推定するネットワークを構築する。5.2節では具体的な構成について述べる。

## 5.2 本研究への適用

### 5.2.1 入力層

図6に構築したネットワークを示す。PointNetでは、最大4,096点で構成される3次元点群のXYZ座標情報を入力とした。本論文で扱うネットワークは、XYZ座標とRGB情報の6チャンネルを有した点群を入力とする。発破ずりの多視点画像から復元された3次元点群は、本来30万~100万程度の点から構成されており、4,096点にダウンサンプリングをすると、情報が大きく欠損する可能性があるため、点数は10万点にランダムにダウンサンプリングする。また点群データはHDF5[19]というファイル形式でまとめる。

### 5.2.2 中間層

入力されたXYZ座標は、T-Netにより剛体変換が適用された後、RGB情報と結合され、64個の1×3のフィルタにより畳み込みが行われた後、さらに64個の1×1のフィルタにより畳み込みが行われる。このような処理を、ここではshared MLPと呼ぶことにする。その後、64×64の行列が推定され、特徴量空間の変換が行われる。実際にPointNetではこの変換により、精度の向上が認められた。変換後のベクトルをshared MLPによって畳み込み、n×1024に次元を拡張する。Max poolingにより、1024次元のベクトルに点群のグローバルな特徴量を集約する。さらに全結合のMLPにより学習を行うことで、最終的に8次元ベクトルを出力する。

### 5.2.3 出力層

出力されたベクトルはスコア値として考えることができる。そこでソフトマックス関数によりベクトルを離散確率分布に変換することで、粒度分布の推定値とする。正解データである粒度分布との損失関数はクロスエントロピー誤差を計算し、誤差逆伝播法によりパラメータを更新する。

## 6. 粒度分布推定実験

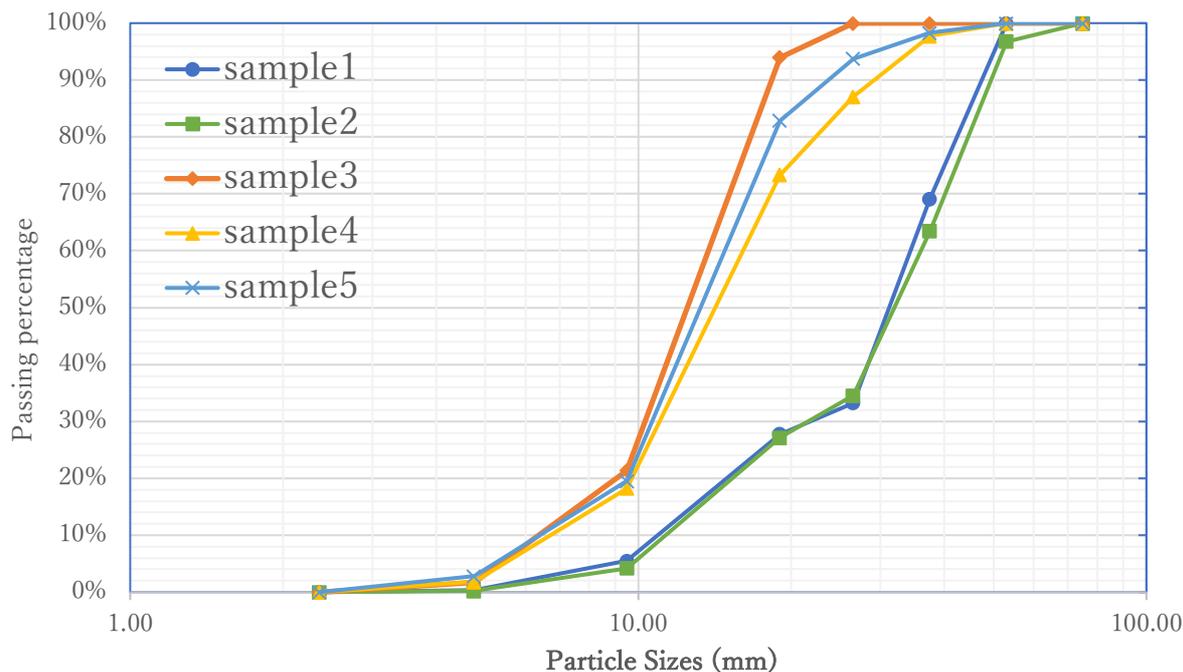
本節では、発破ずりの3次元点群のデータセットを準備して深層学習を行い、未知の発破ずりの粒度分布を推定する実験について述べる。発破ずりのデータは40~50cmほどの幅のサンプルを用いることとする。

### 6.1 発破ずりサンプルの粒度分布推定実験

発破ずりサンプルの粒度分布を計測する。Kalgoorlie, AustraliaのCurtin Universityにおいて、五種類の発破ずりのサンプルに対し、ふるい分けにより粒度分布の計測を行った。ふるいは八つのふるいで構成されており、発破ずりのサンプルは八つの粒度に分割される。計測結果は図7に示す通りである。図7の上のグラフは、計測した発破ずりの各サンプルにおいて、各粒度のふるいを通過する岩石の割合を示したものであり、このグラフを一般的に粒度分布と呼ぶ。図7の下の方の表は、各粒度のふるいにかかった岩石の割合を示すものである。この結果を正解データとして、深層学習を行う。

### 6.2 発破ずりサンプルの3次元復元

6.1節で粒度分布の計測を行ったサンプルの3次元形状を復元する。モバイルカメラを用いて、発破ずりサンプル



	sample1			sample2			sample3		
Ss	Mr	Percentage	PassPer	Mr	Percentage	PassPer	Mr	Percentage	PassPer
75.00	0	0.00%	100%	0	0.00%	100%	0	0.00%	100%
53.00	0	0.00%	100%	210.2	3.25%	97%	0	0.00%	100%
37.50	2056	30.88%	69%	2156.4	33.32%	63%	0	0.00%	100%
26.50	2388.8	35.88%	33%	1868.1	28.87%	35%	0	0.00%	100%
19.00	366.6	5.51%	28%	481.3	7.44%	27%	248	6.06%	94%
9.50	1480.3	22.23%	6%	1485.3	22.95%	4%	2970.5	72.56%	21%
4.75	344.9	5.18%	0%	255	3.94%	0%	807.4	19.72%	2%
2.36	21.4	0.32%	0%	14.9	0.23%	0%	67.8	1.66%	0%

	sample4			sample5		
Ss	Mr	Percentage	PassPer	Mr	Percentage	PassPer
75.00	0	0.00%	100%	0	0.00%	100%
53.00	0	0.00%	100%	0	0.00%	100%
37.50	138.7	2.27%	98%	122.9	1.71%	98%
26.50	651.6	10.67%	87%	326.3	4.54%	94%
19.00	840	13.75%	73%	784.8	10.91%	83%
9.50	3363.5	55.06%	18%	4555.3	63.35%	19%
4.75	1004.8	16.45%	2%	1202.7	16.72%	3%
2.36	109.7	1.80%	0%	199.1	2.77%	0%

Ss : Size of sieves [mm]  
Mr : Mass of rock [g]

図 7 (上) 粒度分布計測結果のグラフ  
(下) 粒度分布計測結果の表

の多視点画像を撮影し、サンプルの形状を手作業で崩し、また積み直すという作業を、一つのサンプルにつき 10 回行うことで、計 50 種の多視点画像のセットを取得する。それぞれに対して、4 章で述べた手法を用いて 3 次元復元し、発破ずりサンプルの 3 次元点群を取得する。復元結果の例を図 8 に示す。復元した 50 個の点群を入力として深層学習を行う。撮影カメラには、Apple 社の iPhone6s を使用し、画像の解像度は 1,080 画素 × 1,920 画素である。復元には OpenSfM[20] というライブラリを使用した。

### 6.3 深層学習

3 次元復元した発破ずりサンプルに、5 章で述べた深層学習のネットワークを適用し、学習を行う。40 個のデータを学習(train)データ、10 個のデータを検証(test)データとして利用する。それぞれの点群を 10 万点にランダムにダウンサンプリングし、座標を標準化することによってスケールングを行っている。さらに学習時には、ランダムに点群を微小に回転、また点を微小に移動させることで、学習データのオーグメンテーションを行う。計算機は、CPU : Intel Core i7-3930K 3.20GHz, GPU : NVIDIA GeForce GTX 1060

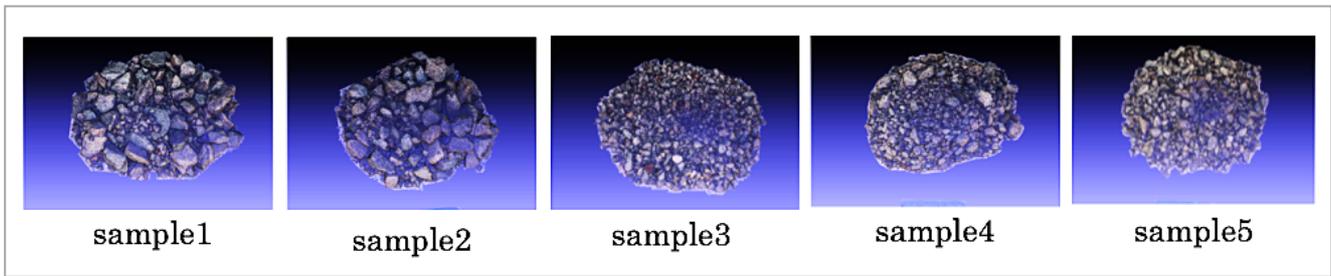


図8 発破ずりサンプルの復元結果

6GBを搭載したデスクトップ PCを用いる。学習のバッチサイズは1, 初期学習率は0.001, パラメータの更新は adam 法[21]とし, エポック数は200回で学習を行った。

学習の結果を図9に示す。各エポックごとの損失関数の平均の推移を示している。学習データの損失関数については, 0付近に収束することはなかったが, 減衰がみられた。検証データの損失関数は, 値が14から16の間付近で振動しており, 収束することはなかった。そこで, 図6における1番目の Shared MLP 層を2層から10層, 2番目の Shared MLP 層を3層から5層, 最後の MLP 層を3層から21層に増やし, 学習を行ったところ, 図10に示す結果になった。学習データについては大きな差異はなかったが, 検証データの損失関数は大きく変化し, 学習データの損失関数に重なるように減衰していることが確認できる。

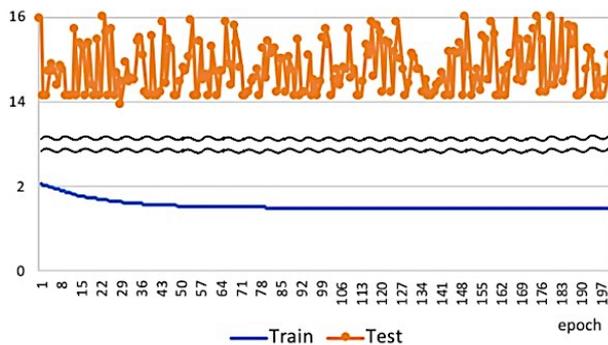


図9 損失関数の推移

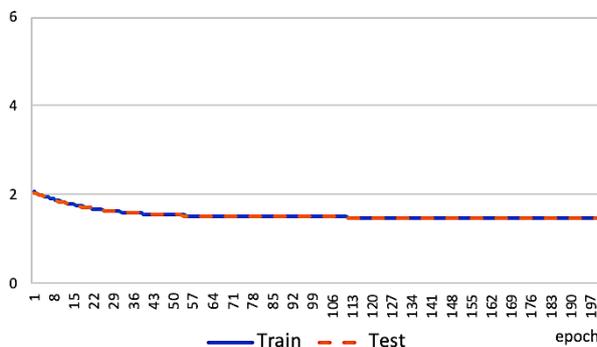


図10 層を増やした場合の損失関数の推移

#### 6.4 粒度分布の推定

学習データと検証データを利用し, パラメータ, またハイパーパラメータの更新をするため, 学習ネットワークは学習データと検証データに依存した構造になってしまう。そこで両データにも使用していないデータを入力として, 粒度分布を推定する。一つの発破ずりのサンプルを入力として粒度分布を推定した結果を図12に示す。また粒度ごとに誤差を計算した結果を図11に示す。

6.3節の結果では, 損失関数の減衰は不十分であり, ネットワークの表現力の乏しさが危惧されたが, 学習データには75mmなどの大きい粒度や2.36mmの小さい粒度の岩石がほとんど含まれていないため, 推定結果も極めて小さい値になることが考えられる。さらに学習データの sample1, sample2 は19mmの粒度の岩石が著しく少ない。このデータの影響を受け, 推定結果においても小さい値になると考えられる。

Size	Sieving	Proposed	Error
75.00	0.00%	0.04%	0.04%
53.00	0.00%	0.68%	0.68%
37.50	0.00%	13.60%	13.60%
26.50	7.73%	15.97%	8.24%
19.00	17.65%	8.73%	8.92%
9.50	43.23%	47.21%	3.98%
4.75	22.18%	12.40%	9.78%
2.36	9.20%	1.36%	7.84%
Average			6.64%

図11 粒度分布推定の誤差

#### 7. おわりに

本稿では, 発破ずりの3次元点群を入力とした深層学習による粒度分布の推定手法を提案した。SfM, PMVSを用いて多視点画像から3次元復元を行い, 発破ずりの3次元点群を生成した。点群の座標, 色情報を入力として, PointNetを応用した深層学習ネットワークによって, 発破ずりの粒度分布を推定した。実験では, 粒度分布推定結果と実際の粒度分布との損失関数が減衰することを確認した。さらに, 学習データには含まれない未知の発破ずりサンプルに対して推定が可能であることを確認した。本研究は CREST JPMJCR16E3 の助成を受けたものである。

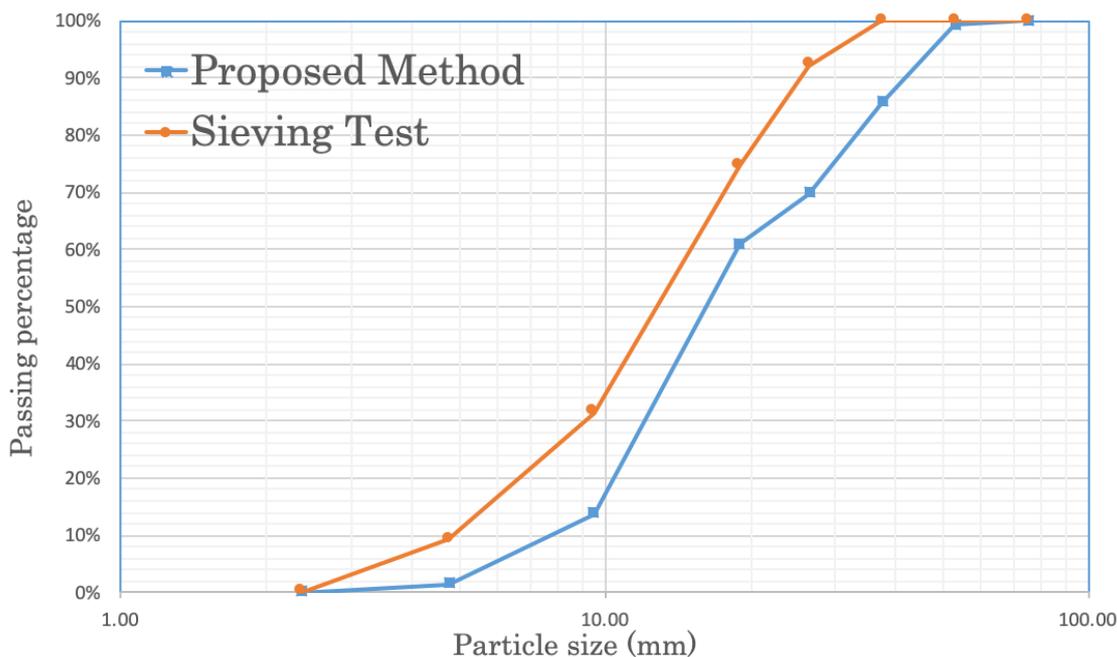


図 12 発破ずりの粒度分布推定結果

### 参考文献

- [1] “高橋良堯, 山口耕太郎, 笹岡孝司, 濱中晃弘, 島田英樹, 一ノ瀬政友, 久保田士郎, 佐分利禎, “露天掘り鉱山における発破規格及び岩盤状態が起砕物の飛翔挙動に与える影響”, Journal of MMIJ, Vol.135, No.10, pp. 94-100, 2019.
- [2] Cunningham, C.V.B. “Fragmentation estimations and the Kuz-Ram model-Four years on” Proc. 2nd Int. Symp. on Rock Fragmentation by Blasting. 1987.
- [3] Maerz, N. H., Franklin, J. A., Rothenburg, L., & Linncoarsen, D., “Measurement of rock fragmentation by digital photoanalysis”, 6th ISRM Congress, 1987.
- [4] Oneddera, Itaro., M. J. Thurley, Alex Catalan., “Measuring blast fragmentation at Esperanza mine using high-resolution 3D laser scanning”, Mining Technology 124.1, pp. 34-36.
- [5] R. Degawa, Hyongdoo Jang, Youhei Kawamura, Itaru Kitahara, “A Conceptual Study for Development of 3D Rock Fragmentation Analysis System with Stereo-photogrammetry Technologies”, 9th Asian Rock Mechanics Symposium, 2016.10
- [6] J. Papon, A. Abramov, M. Schoeler, F. Worgotter, “Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013.
- [7] 齋藤 康毅, “ゼロから作る Deep Learning—Python で学ぶディープラーニングの理論と実装”, O'Reilly Japan, 2018.
- [8] Liu, Y., Nadolski, S., Elmo, D., Klein, B., Linncoarsen, D., “Use of Digital Imaging Processing Techniques to Characterise Block caving Secondary Fragmentation and Implications for a Proposed Cave-to-Mill Approach”, 49th US Rock Mechanics/Geomechanics Symposium, 2015.
- [9] B. Curless, “From Range Scans to 3D Models”, ACM SIGGRAPH Computer Graphics, Vol.33, No.4, pp38-41, Nov, 2000.
- [10] N. Snavely, S. M. Seitz and R. Szeliski, “Photo Tourism: Exploring Photo Collections in 3D”, ACM Transactions on Graphics, Vol.25, No.3, pp.835-846, 2006.
- [11] Y. Furukawa, J. Ponce, “Accurate, Dense, and Robust Multi-View Stereopsis”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009.
- [12] Hartigan, John A., and Manchek A. Wong “Algorithm AS 136: A k-means clustering algorithm”, Journal of the Royal Statistical Society, Series C (Applied Statistics) 28.1, pp.100-108, 1979.
- [13] 查万志, Hyongdoo Jang, 宍戸英彦, 亀田能成, 北原格, “岩石の多視点画像から推定した3次元点群のクラスタリング”, 情報処理学会第80回全国大会, 3R-05, Vol.2, pp.475-476 (2018.3)
- [14] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. “Pointnet: Deep learning on point sets for 3D classification and segmentation” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
- [15] Hongdong Li, R. Hartley, “Five-Point Motion Estimation Made Easy” International Conference on Pattern Recognition, 2006.
- [16] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. Journal of Machine Learning Research, 9(Nov):2579–2605, 2008.
- [17] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 37–45, 2015.
- [18] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz. Align-ing point cloud views using persistent feature histograms. In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3384–3391. IEEE, 2008.
- [19] The HDF Group, “THE HDF5® LIBRARY & FILE FORMAT”, <https://www.hdfgroup.org/solutions/hdf5/>
- [20] Mapillary, “mapillary/OpenSfM”, <https://github.com/mapillary/OpenSfM>
- [21] Diederik P. Kingma, Jimmy Lei Ba, “ADAM: METHOD FOR STOCHASTIC OPTIMIZATION” arXiv:1412.6980v9. 2017