

消波ブロック据付けシミュレータにおける作業の正当性の判定方法

新潟大学 ○ 千葉瑠輝 新潟大学 学生会員 Chua Hao Min

新潟大学 山崎達也

1. はじめに

消波ブロックは、打ち寄せる波のエネルギーを減少させ、海岸線を波の浸食から保護することを目的として、港や岸壁に設置される。消波ブロックの据付け作業は、熟練技術者の経験則に頼る部分が大きく、技術の修得には多大な時間とコストがかかるため、効率的な技術継承が課題となっている。この課題を解決するために、Chuahらは仮想空間で消波ブロック据付け作業を訓練できる3D Break-Water Simulator（以降、単にシミュレータと呼ぶ）をUnityを用いて構築した¹⁾。

このシミュレータは、熟練技術者が理想的に構築した消波工の情報が得られた場合、それを手本として、練習者が同様の消波工を構築する訓練への活用が期待される。効果的な訓練を実現するためには、手本の消波ブロック（以降、手本ブロック）と練習者が据付けた消波ブロック（以降、設置ブロック）の3次元的位置・回転の一致度を定量的に評価する機能が不可欠である。しかし、現状のシミュレータには、一致度を評価する機能が実装されていない。

そこで本研究では、手本ブロックと設置ブロックの一致率をシミュレータ上で評価する方法として、複雑な形状を持つブロック同士の重なりを精密に計算できるボクセル化手法を用いた一致率評価方法を提案する。

2. ボクセル化手法によるブロックの一致率評価

2.1 ボクセル化手法

ボクセル化は、3次元物体をボクセルと呼ばれる微小な立方体あるいは直方体の集合として近似することで、物体をボクセルの集合体で表現する手法である。コンピュータグラフィックスや医療・科学データの可視化などの多岐にわたる分野でよく利用されている。

ボクセル化の主な利点は、対象の形状の複雑さ

に依らず、統一かつ単純なデータ構造で物体を表現できる点にある。このデータ構造の単純さにより、例えば物体間の重なり判定といった処理をボクセル単位の単純な比較で高速に実行できる。一方で、ボクセル化には表現精度と計算コストがトレードオフの関係にあるという制約が存在する。ボクセルのサイズを小さくするほど、元の物体形状を忠実に表現でき解像度は向上するが、必要となるメモリ容量や処理時間も増大する。

2.2 消波ブロックの一致率の計算

本稿における、シミュレータの対象は4tシェークブロックとするが、他のブロックへの適用も可能である。消波ブロックの一致率の計算手順は、主に以下の5ステップで構成される。

ステップ1. 計算対象の絞り込み：まず、手本ブロックと各設置ブロックの軸平行境界ボックスの重なりを判定する。重なりのある設置ブロックのみをリスト化して、ステップ2以降の計算対象とする。

ステップ2. ボクセル近似：次に、ステップ1で作成したリストから一致率を導出していない設置ブロックを選択してボクセルで近似する。

ステップ3. はみ出し体積の算出：更に、ステップ2の各ボクセルが手本ブロックと重なっているかを判定し、重なっていないボクセルの総体積 V_p を求める。

ステップ4. 一致率の導出：式(1)を用いて設置ブロックの手本ブロックとの一致率を導出する。

ステップ5. 一致率の決定：ステップ2からステップ4の処理を、ステップ1で作成したリスト内の設置ブロックすべてに対して行う。導出された一致率の中で最も高い値を、最終的な一致率として採用する。

ここで、4tシェークブロックの体積を V_0 （ $=1.735 \text{ m}^3$ ）、手本ブロックと重なっていないボク

セルの総体積を V_p とすると、一致率 R_{match} は式(1)のように定義される。

$$R_{match} = \frac{V_0 - V_p}{V_0} \times 100 \quad [\%] \quad (1)$$

2.2.1 計算対象の絞り込み

すべての設置ブロックに対してボクセル化の処理を行うと、計算コストが非常に高くなる。この問題に対処するため、計算の初期段階として、手本ブロックと明らかに重なっていない設置ブロックを効率的に除外するブロードフェーズを導入する。具体的には、各ブロックを完全に囲む軸平行境界ボックス同士を重ねりを判定し、重なりのあるブロックのみを後続のボクセル化処理の対象とする。軸平行境界ボックスとは、x 軸、y 軸、z 軸に平行な辺で構成される、対象の物体を完全に囲む直方体である。

本研究では、軸平行境界ボックスの生成に Unity の **bounds** 関数を、重なりの判定には **Intersects** 関数をそれぞれ用いた。消波ブロックを囲む軸平行境界ボックスを、図 1 として示す。なお、赤色のブロックが手本ブロック、灰色のブロックが設置ブロックであり、黄色の枠が各ブロックの軸平行境界ボックスである。

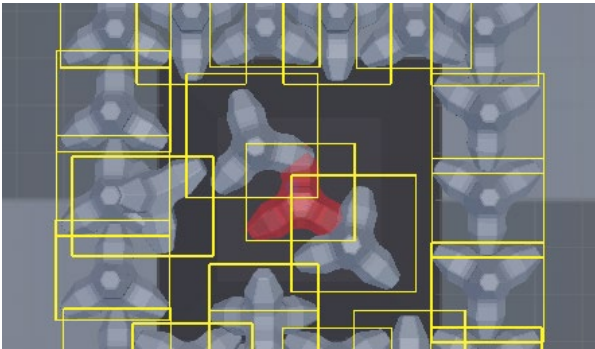


図 1 消波ブロックを囲む軸平行境界ボックス

2.2.2 ボクセル化によるブロック形状の近似

ブロードフェーズで絞り込んだ設置ブロックを、ボクセルを用いて立体的に近似する。この処理は以下の手順で行われる。

1. ボクセルグリッドの生成：まず、計算を行っていない設置ブロック一つを選び出す。このプロ

ックを囲む軸平行境界ボックスの各辺を、所定の分割数 N で分割し、格子状のボクセル群を生成する。例えば $N=10$ の場合には、軸平行境界ボックスは $10^3 = 1000$ 個のボクセルに分割される。

2. 内外判定：設置ブロックは、物理的な衝突範囲を定義する領域として **Collider** をもつ。生成した各ボクセルについて、ボクセルの中心座標が設置ブロックの **Collider** 内部に存在するか否かを判定する。

3. 形状の確定：**Collider** 内部に存在すると判定されたボクセル群のみを、最終的なブロックの近似形状として採用する。

内外判定の具体的な処理は、Unity の **ClosestPoint** 関数と **Distance** 関数を用いて実現する。ボクセルの中心座標を p_{center} とする。内外判定の処理は 2 段階に分けられる。まず最初に **ClosestPoint** 関数の、引数として与えた座標が **Collider** の外部にあれば **Collider** 表面上の最近接点を返し、内部にあれば引数として与えた座標自身を返す性質を利用する。ボクセルの中心座標 p_{center} が **Collider** の内部にある場合、**ClosestPoint** 関数は引数として p_{center} を与えると、 p_{center} の値を返す。次に **Distance** 関数を利用する。**Distance** 関数は、引数として与えた 2 地点間の距離を返す関数である。したがって **Distance** 関数に引数として、**ClosestPoint** 関数の返り値と p_{center} を与えた場合の返り値がごく僅かな閾値以下である場合、そのボクセルは **Collider** 内部に存在するとみなせる。本研究では閾値は、 0.001m とした。図 2 に、ブロックに割り当てられた **Collider** の形状を示す。薄緑の線で覆われた領域が、**Collider** の形状を表している。

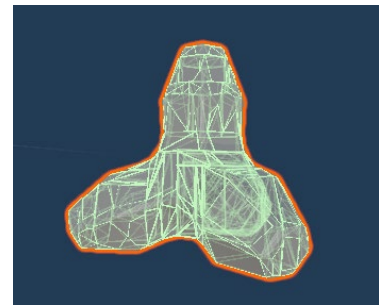


図 2 設置ブロックの Collider の形状

2.2.3 はみ出し体積の算出

前節で作成した，設置ブロックを近似するボクセル群を用いて，設置ブロックにおいて手本ブロックと重なっていない部分の総体積 V_p を算出する．

計算手順として，まず近似形状を構成する全ボクセルを対象に，個々のボクセルが手本ブロックの Collider 外部に存在するかを判定する．この内外判定の処理内容は 2.2.2 節で述べた手法と同様であり，ボクセルの中心座標 p_{center} と $ClosestPoint$ 関数が返す点との距離が，設定した閾値以上か否かで判断する．最後に，Collider 外部に存在すると判定された全ボクセルの体積を合計し，手本ブロックと重なっていない部分の総体積 V_p とする．

3. 一致率計算手法の妥当性検証

3.1 基本的な妥当性の検証

まず，本計算手法の基本的な妥当性を検証するため，二つの極端な場合において一致率を評価した．具体的には，手本ブロックと設置ブロックが完全に重なっている場合の一致率が 100%に，完全に重なっていない場合の一致率が 0%になることを確認した．軸平行境界ボックスの分割数は 20 とした．検証結果を図 3 および図 4 に示す．これらの結果から，本計算手法は想定通りに機能していることが確認できた．なお，赤色の消波ブロックが手本ブロックであり，一致率は Unity の console 上に表示している．また図 3 における青枠は，設置ブロックを近似するボクセルである．

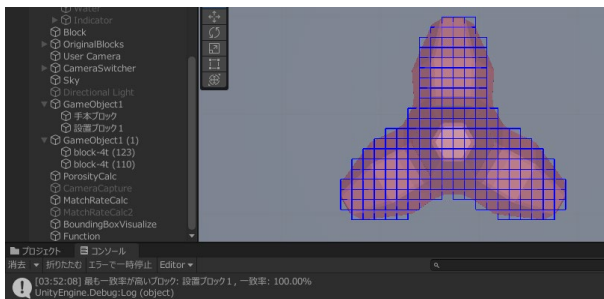


図 3 ブロックが完全に重なっている場合の一致率

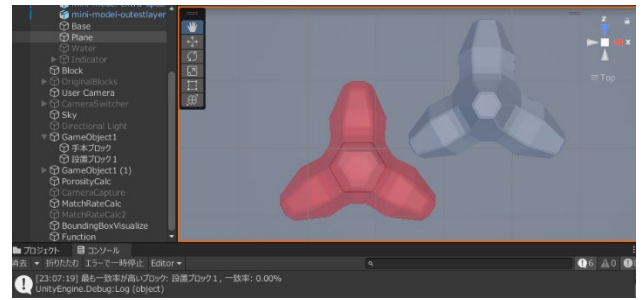


図 4 ブロックが完全に重なっていない場合の一致率

3.2 回転角度と一致率の関係性評価

次に，手本ブロックと設置ブロックを完全に重ねた状態から，設置ブロックのみを y 軸周りに 1 度ずつ 60 度まで回転させた際の一致率の変化を測定した．設置ブロックは y 軸に対して 120 度の回転対称性をもつため，回転角度が 1 度から 60 度の範囲において理想的には，一致率は回転角度に比例して線形に減少すると予想される．

検証結果を図 5 に示す．図より，回転角度の増加に伴い一致率が減少する傾向が見られた．しかし，減少率は一定ではなく非線形な変化を示した．設置ブロックをボクセルで近似しているため，回転角度の増加に伴う，手本ブロックとの重なり領域の減少は，滑らかな減少ではなくボクセル単位での段階的な減少となる．これが一致率の減少が非線形な変化を示した原因であると考えられる．

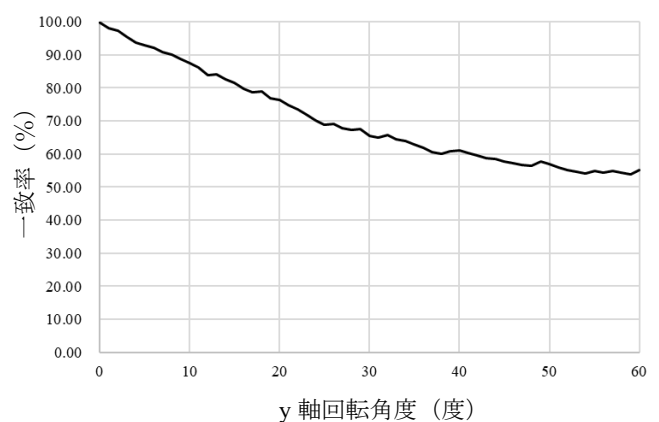


図 5 設置ブロックの y 軸回転角度と一致率の関係

3.3 基本的な処理性能評価

本計算手法の処理性能を評価するため、一致率の計算開始から UI テキストとして画面に表示されるまでの時間を計測した。計測は、軸平行境界ボックスの分割数を 20 に設定し、100 回試行した平均値を採用した。手本ブロックと設置ブロックが完全に重なっている場合と、両ブロックの軸平行境界ボックスが完全に重なっていない場合の二つの場合において、計測を行った。

完全一致条件下での処理時間は 255ms であり、完全不一致条件下での処理時間は 0.030ms であった。完全不一致条件下では、処理がブロードフェーズで終了するため、処理時間に約 8500 倍の差が出ることを確認出来た。

4. おわりに

本研究では、手本ブロックと設置ブロックの一致率を精密に評価するため、ボクセル化を用いた一致率の計算手法を提案し、その妥当性と実用性を検証した。本手法の妥当性検証では、ブロックが完全に重なる場合と完全に重ならない場合に、一致率がそれぞれ 100%と 0%になる基本的な正確性を確認した。さらに、ブロックの回転に伴う一致率の変化が物理的に妥当な傾向を示すことも確認した。これらの結果から、本手法はブロックの一致率を評価する上で一定の妥当性を有すると考えられる。また、処理性能の評価では、手本ブロックと設置ブロックが完全に重なっている場合における処理時間が 255ms であることを確かめた。単位ブロックあたりの処理時間が 1 秒を大きく下回ることからシミュレータでの実用に値する処理性能であると考えられる。今後の課題としては、現在は単一の一致率表示に留まっている機能を拡張することが挙げられる。具体的には、手本ブロックが複数存在する場合に、アプリケーション画面上で効果的に一致率を可視化する方法を検討し、これを実装することで、より実用的な評価ツールとしての完成度を高めていく予定である。

参考文献

- 1) Chuah, H. M., Yamazaki, T., Iwasawa, R. and Suto, T.: Development of Wave-Dissipating Block Installation Simulation for Inexperienced Worker Training, Engineering and Technology International Journal of Computer and Systems Engineering, Vol.18, No.9, pp. 611-168, 2024.