

仮想空間で日本一の山への自律登頂を目指す富士山チャレンジ —大規模 DEM データを gazebo でスクロールする機能の検討—

○小島 匠太郎 (東北大学) 大野 和則 (東北大学) 軍司 健太 (東北大学)

西條 達慶 (東北大学) 田所 諭 (東北大学)

1. 緒言

本研究では、gazebo シミュレータ [1] 上でロボットが富士登山を実施する方法を検討する。建設機械の自動化を行う際、急な斜面を含む広大な土地を自動で走行することが求められる。本研究ではこうした地形の一例として富士山を選定し、シミュレータ上で山頂までの自動走行を目指す。富士山を題材として選んだ理由は、急な斜面を含む広大な地形である点と、国土地理院の公開データ [2] から地形情報を取得できる点である。シミュレータ上で富士山を登頂できれば、同じ方法で別の場所の地形をシミュレータ上に再現し、自動走行のテストを行うことが可能になる。

本稿では、Gazebo シミュレータ上に大規模な地形データを読み込む方法を開発する。富士山は直径 30km 以上の広大な裾野を有しており、それをシミュレータ上に再現する必要がある。同じ方法で建設現場をシミュレータ上で再現する場合にも、数 km ～数十 km 四方の広大な面積を再現する必要がある。

しかし、計算負荷などの観点から、シミュレータ上に一度に再現できる地形の範囲や解像度には制約がある。gazebo シミュレータには DEM (Digital Elevation Model) をロードすることで任意の地形を再現する機能 [3] が既に存在するが、DEM の解像度が 128x128 ピクセル以下に制限されるため、広大な範囲の地形を再現しようとするほど地形が粗くなるという課題がある。また、仮により高い解像度の DEM をロードできるようになったとしても、直径 30km を超える富士山の裾野の地形すべてをシミュレータ上に再現することは、処理負荷の観点から現実的ではない。

そこで、本研究では図 2 のようにロボットの周囲の地形のみシミュレータ上に再現し、ロボットの移動に合わせて地形をスクロールする方法を提案する。この方法では、ロボット周囲の約 10m 四方程度の範囲のみ地形を再現し、ロボットの移動に合わせて地形を再現する範囲をスクロールする。これにより、地形の範囲は常に 100 m² 程度に抑えられ、リアルタイム処理が可能な範囲に処理負荷を抑えることができる。地形スクロールの速度はロボットの移動速度以上である必要があることから、スクロールの速度がシミュレータ上でのロボットの移動速度の上限となる。本稿では、スクロールの最大速度を実測し、

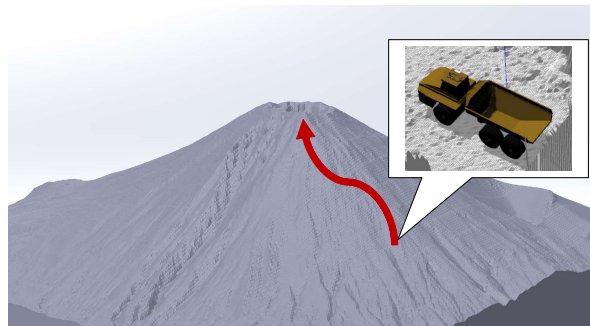


図 1 建設ロボットによるシミュレータ上での富士登山（イメージ）。これを実現するため、本稿では広域の地形データを gazebo シミュレータ上に再現する方法を検討する。

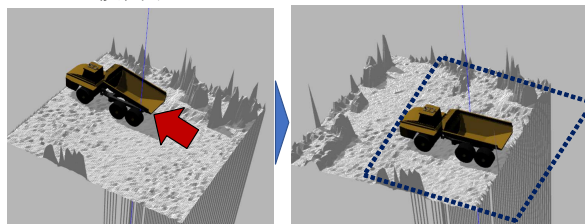


図 2 地形スクロールによる再現範囲の移動。ロボットの移動に合わせて地形を再現する範囲をスクロールすることで、広域のシミュレーションを実現する。

ロボットの移動速度の上限を求める。

2. Gazebo シミュレータ上での地形スクロール

機能の開発

図 3 に地形スクロール機能のプロセスフローを示す。処理の順序は、①地形再現範囲の決定、②再現範囲をブロックに分割、③追加するブロックをシミュレータにロード、④不要なブロックを削除、となる。地形データは ros のトピック (grid_map_msgs/GridMap 型) で与え、ロボットの位置・姿勢は gazebo から取得する。①ではロボット位置をもとに地形を再現する範囲を決め、②ではその範囲を一定サイズのブロックに分割する。③では、分割したブロックのうちシミュレータ上にロードされていないものをロードする。④では、ロボットから遠ざかった位置のブロックを削除する。このよう

全域のDEMデータ(rosトピック経由)



地形スクロールプロセス

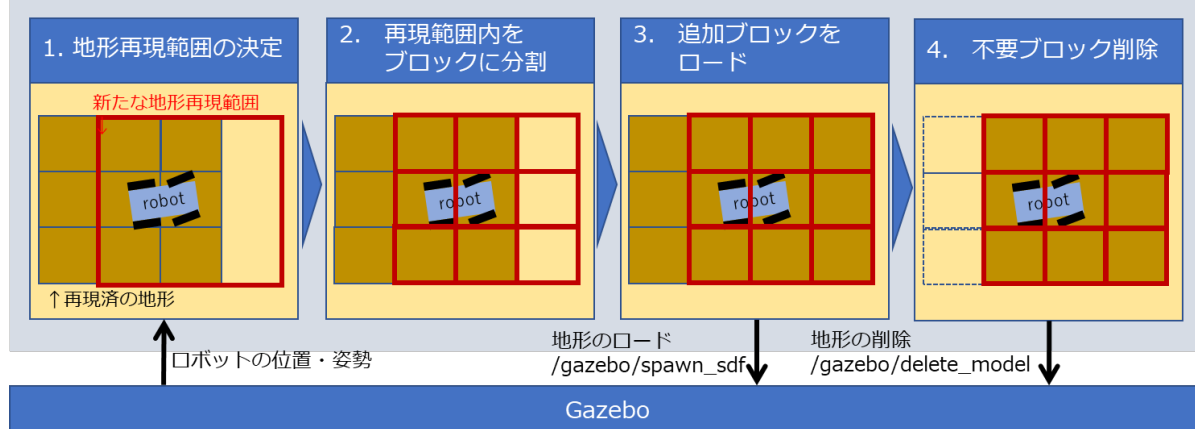


図3 地形スクロール機能のプロセスフロー. ロボット周囲の地形をブロックに区切り, ロボットが一定以上近づいた位置のブロックをシミュレータにロードし, 遠ざかった位置のブロックを削除する. これにより, 地形を再現する範囲の面積を一定以下に抑える.

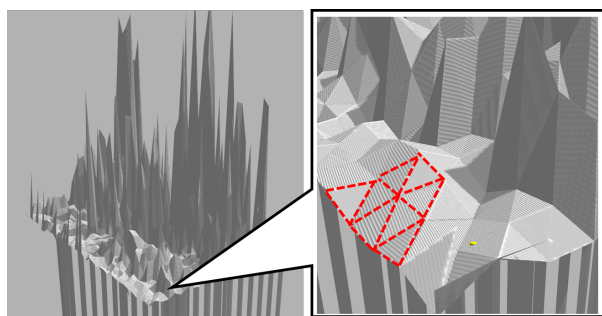


図4 三角形要素のサーフェイスメッシュ(赤点線)を用いた地形の表現. Sdf フォーマットの polyline 形式 [5]を用いて三角形要素を表現した.

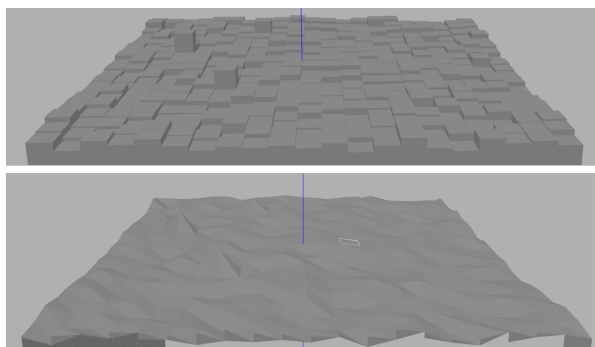


図5 表現方法による gazebo 上での地形の違い. (上)直方体要素による表現, (下)三角形要素のサーフェイスメッシュによる表現. より滑らかに地形を表現するため, 今回は後者を採用することとした.

に, ロボットの移動に合わせて逐次的に地形を再現する範囲を移動することで, 地形を再現する範囲を一定以下に抑える. DEM データの受け渡し・分割等の操作では ros 上で占有格子地図を扱うためのラ

表1 スクロール速度に関する評価結果.

条件	分割単位 1m x 1m	分割単位 3m x 3m
処理時間 (実時間)	64 s	43 s
スクロール 速度 (実時間)	0.14 m/s	0.21 m/s
実時間比 (平均値)	0.39	0.06

イブラリである grid map library [4]を使用した.

Gazebo 上での地形を滑らかに再現するため, 図4に示すように地形を三角形要素のサーフェイスメッシュとしてロードすることとした. 三角形要素は sdf フォーマットの polyline 形式 [5]を用いて三角形の板をつくり, DEM の3つの格子が示す点に三角形の頂点が重なるよう位置・姿勢を調整することでサーフェイスメッシュを表現した. なお, 他にも DEM の各格子の高さを直方体で表現する方法も検討したが, その場合には図5の上の図のように表面の凹凸が大きく表現されてしまうという欠点があることが分かった. 表面形状をより滑らかに再現するため, 三角形要素のサーフェイスメッシュを採用した.

3. 地形スクロールの速度に関する検証

地形スクロール機能を用いて gazebo 上でロボットを走行させる際には, ロボットの移動速度は地形のスクロール速度によって制限される. ロボットが地

形のスクロール速度を超えた速度で移動すると、地形が再現された範囲外にロボットがはみ出してしまうためである。

地形スクロールの速度を評価するための検証を行った。大きさ 9m x 9m、解像度 0.5m の DEM をブロックに分割して gazebo にロードし、一辺の方向に 9m スクロールするのにかかる時間(実時間)を計測した。また、移動距離を時間で割ることで、スクロールの平均速度を求めた。さらに、処理に要した実時間とシミュレータ上での経過時間を比較することで、処理中の実時間比(real time factor)の平均値を求めた。ブロックの分割単位は 1m x 1m と 3m x 3m の 2 条件で計測を行った。

検証結果を表 1 に示す。9m 四方の DEM を 9m スクロールした際の平均速度は、分割単位 1m 四方と 3m 四方でそれぞれ 0.14m/s と 0.209m/s、実時間比はそれぞれ 0.39 と 0.06 となった。スクロール速度では 3m 四方のほうが高速に移動できるが、実時間比を比較すると 1m 四方のほうが高いという結果となった。

4. 考察

3 節の結果から、現状の実装におけスクロール速度は 0.14m/s から 0.20m/s 程度であることが分かった。この速度は、クローラロボットなど人よりも小さいロボットにとっては十分な速度と考えられる。一方で、自動走行ダンプトラック等の建設ロボットの場合は、移動速度が 1.0m/s から 2.0m/s である。そのため、建設ロボットに適用するためには、スクロール速度を一桁程度速くする必要がある。

3 節の検証での処理時間について、1m x 1m サイズのブロックを gazebo にロード・アンロードする時間を詳しく確認したところ、ロードにかかる時間は 1 ブロックあたり 0.4 秒程度、アンロードにかかる時間は 1.0 秒程度であった。アンロードに特に多くの時間がかかっており、そこには時間短縮の余地があると考えられる。

5. 結言

本稿では、富士山のような広域のシミュレーションを gazebo 上で実施するための地形スクロール機能の実装およびスクロール速度の評価について述べた。ロボットの移動に合わせて地形を再現する範囲をスクロールすることで、広域を移動するロボットのためのシミュレータ上での地形再現の方法を構築した。検証結果から、現状の実装では 0.14m/s から 0.20m/s 程度の速度でスクロールできることが確認できた。今後の課題としては、gazebo のモデルのロードとアンロードの高速化によるスクロール速度の向上が挙げられる。また、実際に国土地理院が公開している

富士山の地形データを gazebo 上にロードし、ロボットで富士登山を実施する予定である。

謝辞

本研究は、JST【ムーンショット型研究開発事業】グラント番号【JPMJMS2032】の支援を受けたものです。

参考文献

- [1] Koenig, Nathan, and Andrew Howard. "Design and use paradigms for gazebo, an open-source multi-robot simulator." 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566). Vol. 3. IEEE, 2004.
- [2] 国土地理院, “地理院地図 (電子国土 Web)”, <https://maps.gsi.go.jp/>
- [3] GazeboSim.org, “Digital Elevation Models”, <http://gazeboSim.org/tutorials/?tut=dem>
- [4] Fankhauser, P  ter, and Marco Hutter. "A universal grid map library: Implementation and use case for rough terrain navigation." Robot Operating System (ROS). Springer, Cham, 2016. 99-120.
- [5] sdfORMAT.org, “polyline”, http://sdfORMAT.org/spec?ver=1.8&elem=geometry#geometry_polyline