

レゴブロックのための 3 次元近似形状組み立て支援システム A LEGO Block Modeling System with 3D Shape Approximation

北川 佑樹†
Yuuki Kitagawa

高井 昌彰‡
Yoshiaki Takai

高井 那美*
Nami Takai

1 まえがき

ユニット折り紙や缶アートなど、基本的に同一形状のプリミティブを立体的に多数組み合わせ、任意の 3 次元近似形状を生成する研究が行われている[1, 2]. 組み合わせの多様性から、同じ最終形を目標として構築しても制作手順、完成形状は制作者によって異なる. また拡張現実を利用し形状構築の支援を行う研究[3], 教育・知育に利用した研究[4]など, 実世界と関連付けられた 3 次元形状構築に関し現在幅広い分野から注目を集めている.

形状構築の玩具の一つとしてレゴブロックがある. レゴブロックはブロック同士を組み合わせることで立体形状を形成するものであり、一般的に子供の玩具として親しまれている. 近年では、芸術作品の一種としてレゴブロックによるオブジェが数多く制作されている(Fig.1). しかし実際に大規模な制作を行う場合、大量のブロックと広い制作場所を要するため、制作前にブロックの種類や組み合わせを十分に検討する必要がある.

そこで本研究では、3D ポリゴンモデルデータと使用ブロックの種類、個数、完成オブジェの解像度等の制約条件を入力として与え、物理法則に基づいて実際に配置する場合の安定性を考慮した上で、ポリゴンモデルをレゴブロックで近似表現した 3D モデルを自動生成するレゴブロック制作支援システムを開発した. レゴブロックの CAD ツールとして[5]があるが、近似形状の自動構築支援の機能はない.



©レゴ(R)ブロックワールド SAPPORO

Fig.1 レゴブロックによるアート作品

2 システムの対象

本システムでは、初心者にも最もなじみ深い、1×1, 1×2, 1×3, 1×4, 1×6, 1×8, 2×2, 2×3, 2×4, 2×6, 2×8 の 11 種類のレゴブロックを扱う. ポリゴンモデルのボクセル化では、1×1 ブロックをボクセル 1 個に対応させる.

ブロックの配色は、ユーザーの選択に基づき、モデルの色をそのままブロックに反映させるか、あるいは白、黄、赤、青、緑、黒、灰、茶、黄緑、橙のレゴブロックの代表的な 11 色のみを使用する. 11 色のみを使用する場合はブロックに対応するモデルの領域内で最も使用されている色とのユークリッド距離に近い色を選択する.

レゴブロックには一般に推奨されるような「正しい組み方」は存在しない. 本システムでは基本的にブロック間

の連結性をできるだけ保持するような配置方法をとる. なお、全体の重量バランスのチェックには、脆弱度の可視化機能を利用できるが、組み立て途中の配置方法に重量の影響を適応的に反映させることは現在未実装である.

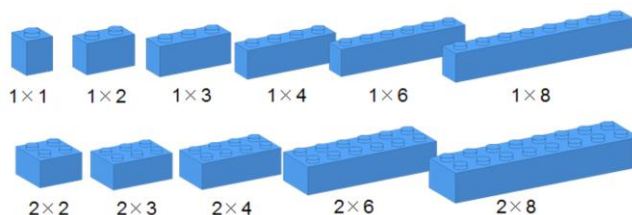


Fig.2 ブロックの種類

3 システムの概要

本システムの全体の処理の流れを Fig.3 に示す. 本システムはレゴブロックをユーザー自ら 1 ブロックずつ組み上げていくシステムではなく、ユーザーが与えた 3D モデルをレゴブロックのモデルに変換するシステムである. ポリゴンモデルは一般的な 3D モデリングソフトで作成し、実際に組んだ時のブロック解像度またはブロック総個数のいずれかをユーザーが制約条件として指定する.

システムはポリゴンモデルの形状とテキスト及び構築の制約条件から、ボクセル化、厚み付け、ブロック配置、連結性判定、再配置処理を行い、最終的にレゴブロックによる 3D モデルを出力する.

また本システムは、生成されたレゴブロックモデルに対してユーザーがインタラクティブに個々のブロックの配置を調整できる GUI も実装している. 実際に組み立て可能かどうかを判定するため、荷重とモーメントの計算による脆弱度の可視化の他、ブロック間の結合力を考慮した物理シミュレーションの機能も実装している.

さらに作品の実制作上の支援として、完成イメージを掴めるよう AR 表示機能、ならびに、各層ごとの詳細なブロック配置図を表示する機能を実装している.

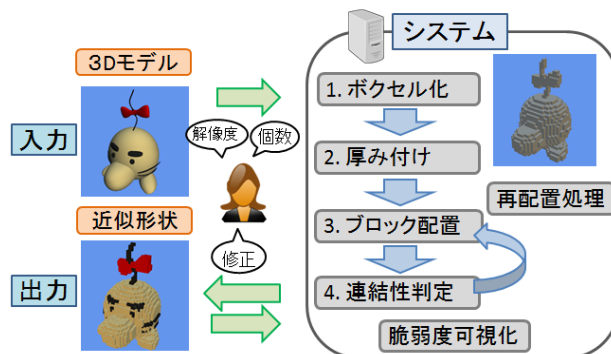


Fig.3 システム概要

†北海道大学大学院情報科学研究科, Graduate school of Information Science and Technology, Hokkaido University

‡北海道大学情報基盤センター, Information Initiative Center, Hokkaido University

*北海道情報大学, Hokkaido Information University

3.1 ボクセル化と厚み付け

ボクセル化と厚み付け処理の流れを Fig.4 に示す。入力された 3D ポリゴンモデルの表面に対して、1×1 ブロックと同サイズ、すなわち縦、横、高さの比が 5:5:6 の直方体によるボクセル化を行う。

内部の厚み付け処理では、層ごとに 1 ブロックずつ走査していき、直下層の同一平面位置にブロックが無く、かつモデルの内部である場合、その部分に 1×1 ブロックを配置する。この処理を最下層から最上層まで繰り返す。

外部の厚み付け処理では、内部の厚み付け同様に、層ごとに 1 ブロックずつ走査していき、直上及び直下の両方の層の同一平面位置にブロックが無く、かつモデルの外部である場合、その層の直下層にブロックを配置する。

また 40 層以上の構築を行う場合には、内部の厚み付けの前に 6 層ごとに内部を全て埋める層を作り、内部を補強する。

これらの内外部の厚み付け操作により、作品全体の強度が増し、非連結ブロックの発生を抑制できる。内部補強の処理は単に強度を上げるだけでなく、実制作時にモデル内の奥にブロックを落とすことを防ぐためでもある。



Fig.4 厚み付け

3.2 ブロックの配置

ブロックの配置方法は基本的に 2×8 ブロックから 2×6, 2×4, 2×3, 2×2 ブロックという順に、2×N ブロックを先に配置し、次に 1×8 ブロックから 1×6, 1×4, 1×2, 1×1 ブロックという順に、1×N ブロックを配置する。どのブロックも一定方向から走査していき、配置可能性を判定する。

2×N ブロックの配置は、はじめに空間を 2×2 ブロック幅の格子に区切ってから行う。その際、奇数層と偶数層は 1×1 ブロック分だけ縦横に格子をずらし、この格子を基準に配置を行う。1×N ブロックは、奇数層・偶数層ともに 1×1 ブロック幅の格子により配置を行う。Fig.5 に奇数層と偶数層のブロック配置例を示す。

また 2×2, 1×1 ブロック以外の縦横の長さの違うブロックは、奇数層と偶数層でそれぞれ優先して配置させる向きを変更する。

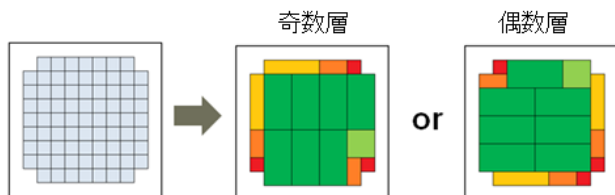


Fig.5 ブロックの配置

3.3 連結判定

1ブロックを1ノードとし、レゴブロックで構築されたモデルを1つのグラフ構造として考える。上下両方に1ポッチでも接続している場合、その部分を枝とする。した

がって2×8ブロックの場合は最大で32個の枝を持つことになる。このグラフ構造に対して幅優先探索を行い、各ブロックが最下層のブロックのいずれかと連結しているかどうかを調べる。配置予定の全てのブロックの連結性が確認された時、ブロックの配置を決定する。

Fig.6 で一例を示す。左図のブロック配置をグラフ構造として見たものが右図である。最下層である1番のブロックとの連結のない3, 6番のブロックを非連結ブロックとして扱う。この非連結ブロックが実際に構築不可能なブロックであり、これらを解消する処理を以降で行う。

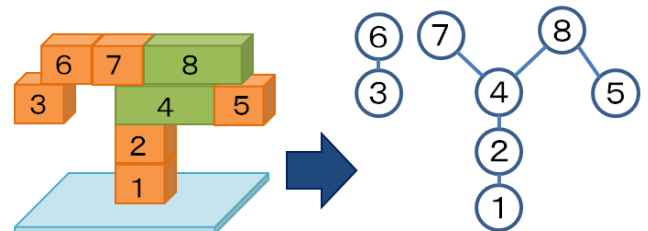


Fig.6 連結判定

3.4 再配置処理

連結性判定により非連結なブロックが 1 つでも確認された場合、非連結部分の 1 ヶ所を 1×2 ブロックで置き換え、先述したブロック配置ルールに従って、その層のブロックを全て組み直す。次に再配置前と再配置後の総非連結数を比較し、再配置前より再配置後が多い場合または置き換え部分の非連結性が解消されなかった場合、配置を置き換え前に戻す。非連結部分が解消されない場合、1×2 ブロックの置き換え方を変え、組み直しの処理をする。したがって同一箇所でも最大 4 回再配置処理を試すことになる。全ての配置でも解消されない場合、置き換える対象のブロックを変えて、組み直しの処理を繰り返す。再配置処理の流れを Fig.7 に示す。

1×2 ブロックで置き換える理由は、内部・外部の厚み付け並びにブロック配置ルールにより非連結部分が必ず 1×1 ブロックになっているためである。本システムの評価実験で使用したモデルに関しては、再配置処理を繰り返すことで非連結ブロックは全て解消された。

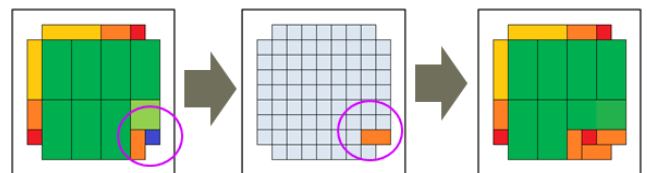


Fig.7 再配置処理

3.5 脆弱度の可視化

ブロックにかかる荷重とモーメントを計算することによって、水平断面ごとの脆弱度を可視化する。脆弱度を計算する断面の図心を P1, その断面より上側全体の重心を P2, かかる重力を W, 上の断面と連結しているポッチ数を N とする。ここで P1P2 ベクトルと W ベクトルによる外積を N で除した値を脆弱度とし、その値を可視化する。また P2 が脆弱度を計算する断面の内部なら、脆弱度

はゼロとする。可視化の例を Fig.8 に示す。色が濃いほど、脆弱度の高い場所である。

摩擦力などによるブロック間の結合力を考慮した物理演算には Physx を利用している。衝突判定を行うプリミティブに円柱型が用意されていないので直方体を組み合わせたプリミティブで代用している。レゴブロックの材料と同様の摩擦係数を設定することで、現実に近い検証が可能となっている。

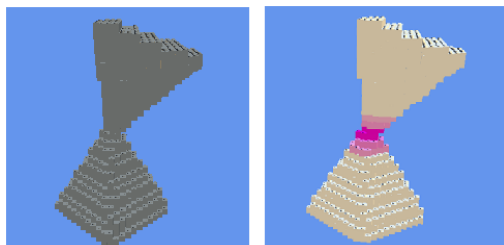


Fig.8 脆弱度の可視化

3.6 手積み機能

Fig.9 に手積み機能の表示を示す。右側にモデル全体が表示され、ユーザーはマウス操作により任意の層を選択することができる。選択した部分は赤く示され、左側に選択した層の水平断面が表示される。また、左側の水平断面ではどの種類のブロックが配置されているかの見やすさを考慮し、ブロックのサイズを小さくすることでブロック間に隙間ができるようにしている。ブロックの追加では、使用するブロックの種類をアイコンから選択し、水平断面の任意の個所に合わせることで配置可能である。また任意の個所のブロック削除も可能である。左側の水平断面への操作の結果が右側のモデルにも反映して表示される。

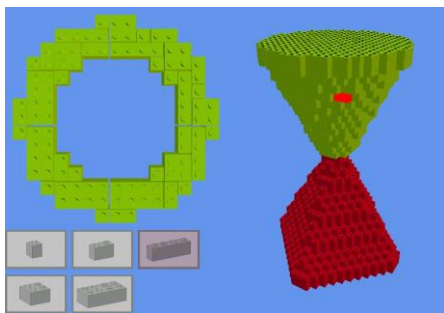


Fig.9 手積み機能

3.7 AR 表示機能

マーカーの位置に完成オブジェ、各層の断面を表示させることで実際の制作場所にどれくらいの空間を占めるのか等のシミュレーションが行える。30層で構築したモデルのAR表示を Fig.10 で示す。

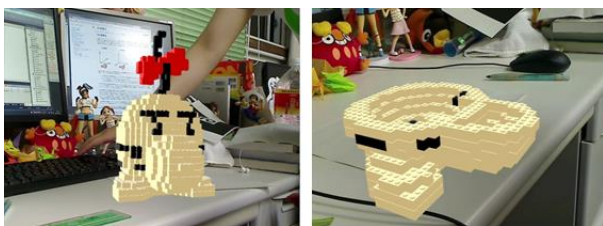


Fig.10 AR 表示機能

3.8 配置図

実際に作品を構築する場合に参考にするものとして、層ごとのブロック配置図の出力機能を用意している。ある層の断面図の Fig.11 に示す。左が k 層目、右が $k+1$ 層目である。ユーザーの作りやすさを考慮し、指定層の下層の配置についても青く表示している。また下の層から順に構築していく場合に、1つ上の層のブロックを配置してからのみ実際に配置可能なブロックについての表示も同時に行っている。

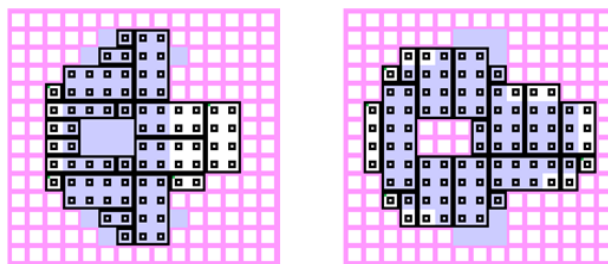


Fig.11 配置図(k 層(左), $k+1$ 層(右))

4 システムの実行結果

4.1 実装環境

本システムの実装は、開発言語にC#言語、3DCGの描画にはXNA4.0, Physxを呼び出すためのラッパーライブラリにStillDesignPhysX.Net, AR表示にはNyARToolkitを使用した。

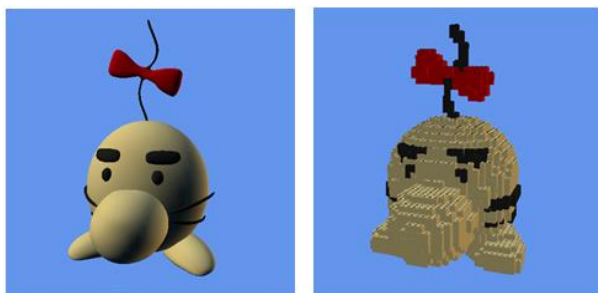
また OS : Windows 7 Professional x64, GPU : NVIDIA GeForce GTX 460 の PC 上で本システムを実装し、以降の結果も同 PC 上で実行したものである。

4.2 自動構築

本システムを評価するため、いくつかの3Dポリゴンモデルデータと解像度を本システムに入力し、レゴブロックモデルを生成した。40層で構築した結果をFig.12に示す。ここでは使用ブロックの個数は制限として与えず、本システムのブロック配置方法において最適な配置結果である。概ね正確に本システムによるレゴブロックの変換が行われたことがわかる。ブロック間の連結性はすべて満たされているが、Fig.12(a)のリボンの部分やFig.12(b)の耳の部分など、実際に組み立てた場合に崩壊する可能性のある脆弱な部分がある。

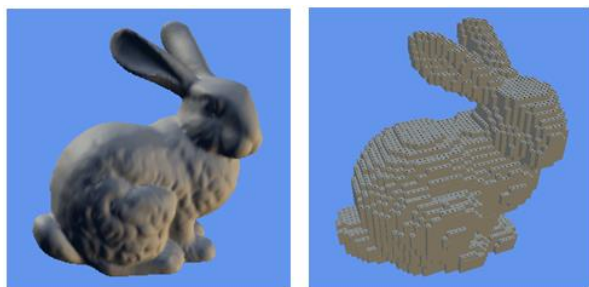
Fig.12の2つの作品の見積もり情報をFig.13に示す。入力モデルデータの形状により、必要となるブロックの種類が大きく違うことがわかる。構築時間の大半はボクセル化の処理にかかる時間である。Fig.12(a)はポリゴン数9240, Fig.12(b)はポリゴン数34384であり、同じ解像度でも構築時間の差が出ているのはこのためである。ブロック配置から再配置処理にかかる時間はどちらのモデルも2秒前後である。また概算費用については、ブロックの1ポッチにつき2円で計算している。つまり1×1ブロックは2円で、最大が2×8ブロックの32円という計算である。

Fig.12(a)のモデル1を異なる解像度で構築した結果をFig.14に示す。解像度を上げると入力モデルにより近づくのは明らかであるが、モデル1のリボンの部分のように、少ないブロックで上のブロックを支える箇所がある場合には必ずしも解像度を上げるのが良いとは限らない。



©任天堂

(a) モデル 1 (40 層)



(b) モデル 2 (40 層)

Fig.12 システムの実行結果

1×1	306個	1×1	552個
1×2	277個	1×2	500個
1×3	54個	1×3	162個
1×4	106個	1×4	249個
1×6	26個	1×6	52個
1×8	30個	1×8	76個
2×2	52個	2×2	161個
2×3	50個	2×3	87個
2×4	183個	2×4	192個
2×6	51個	2×6	60個
2×8	113個	2×8	112個
総数	1248個	総数	2203個
高さ	38cm	高さ	38cm
重さ	2220g	重さ	3011g
概算費用	12,468円	概算費用	18,336円
構築時間	10秒	構築時間	28秒

モデル 1

モデル 2

Fig.13 作品の見積み結果とシステム処理時間



Fig.14 解像度の違いによる比較 (左から 20, 30, 50 層)

4.2 実製作

本システムがポリゴンモデルから自動生成するブロック配置図をもとに、Fig.12 の 2 つのモデルを実際にレゴブロックで構築した結果を Fig.15 に示す。作品の完成まで

に要した制作時間（組み立て作業人数 1 名）は、モデル 1 が 20 時間、モデル 2 が 30 時間であった。モデル 1 のリボン部分やモデル 2 の耳の部分などに崩れやすい脆弱な箇所があるが、ブロックのノリ付けなどの特殊処理は一切せずに全体を構築できている。



Fig.15 作品の実制作の結果 (モデル 1(左),モデル 2(右))

5 まとめと今後の課題

本研究では、仮想空間上にレゴブロックを表現し、与えられたポリゴンモデルの近似形状を自働構築し、適切な組み合わせ・配色・脆弱性・総数(コスト)を見積もることにより、作品制作を支援するシステムを構築した。

今後、脆弱度や組みやすさを反映したブロックの配置法や、使用できるブロック種類の拡充、完成モデルの評価方法の拡充を目指す。

参考文献

- [1] 高橋和茂, 高井昌彰, 高井那美: “近似形状の構築が可能な缶アート制作支援システム”, 情報処理学会研究報告, Vol. 2011-GC-145, No. 26, pp. 1-6 (2011)
- [2] 田村友和, 高井昌彰, 高井那美: “ユニット折り紙を用いた3次元メッシュモデルの近似形状構築”, 情報処理学会研究報告, Vol. 2010-CG-141, No. 3, pp. 1-6 (2010)
- [3] 川島高志, 加藤博一, 橋啓八郎: “拡張現実感を用いた3次元部品組み立てマニュアルとその評価”, 電子情報通信学会技術研究報告, Vol. 2001-マルチメディア-101, No. 389, pp. 1-6 (2001)
- [4] 後藤頭一, 鮫島朋美, 高橋三男, 松原静郎: “レゴブロックの組み立て再現を利用した表現力育成の基礎的研究: 効果的な記録方法の基礎的なトレーニング”, 日本理科教育学会第59回全国大会, Vol. 日本理科教育学会全国大会要項-202, No. 59, pp. 1-4 (2009)
- [5] BlockCAD <http://www.blockcad.net/>