

自立性と安定性を考慮したレゴモデル組み立て図の自動生成

狐崎 拓哉^{†1} 竹澤 正仁^{†1} 佐々木 雄飛^{†1} 前川 卓^{†1}

大きなレゴモデルを組立てる際、モデルを中空にし、ブロック数を減らすことによって、軽量化ならびにコスト削減を図ることができる。しかし、中空の仕方によって、自立性を失い、倒れてしまうことが予想される。また、自立するモデルを作成できた場合でも、倒れやすい安定しないモデルとなることがある。そこで、本研究ではモデルの内部ブロックの配置を考慮することによって重心位置を調整し、自立して倒れにくい安定したレゴモデルの組み立て図を自動生成するシステムを開発する。

Automatic generation of building instructions for stable and self-balanced LEGO sculptures

TAKUYA KOZAKI^{†1} MASAHITO TAKEZAWA^{†1} YUHI SASAKI^{†1}
TAKASHI MAEKAWA^{†1}

When building a large-scale LEGO model, we can reduce the number of bricks not only to lower the cost but also to lighten the model by hollowing it. However, the hollowed model may change its position of center of gravity significantly, and may not be able to stand by itself. In this paper, we present a method to control the position of the center of gravity of the large-scale hollowed LEGO model so that it can stand without any support.

1. はじめに

レゴブロックは安価であり、構造がシンプルであるが、ブロックを組み合わせることで複雑な形状を作り出すことができる。そのため楽しみながら大小様々な作品を組み立てることができ、子供から大人まで幅広い世代の人々に親しまれており、教育の分野にも多く導入されている。基本ブロックのみで構成されるサイズの大きな作品はレゴスカルプチャーと呼ばれ、インターネットを通して世界中の人々を魅了している。しかし、レゴをあまり扱ったことが無い初心者が複雑な形状の作品を作るには、時間と労力を要し、目的形状の作品を作ることは難しい。また、モデルの重心を考慮していないと、完成品が自立せず倒れてしまうことがある。そのため多くの場合、熟練したレゴビルダーが試行錯誤を積み重ねて作成した組立説明書が一般向けに付属品としてレゴセットに入っている。

一方、近年 3D プリンタは個人用から法人用まで価格も幅広く、目的形状を簡単に作れるため、注目を浴びている。この 3D プリンタの普及に伴って、形状モデリングの分野でも、造形コスト、自立性、強度、慣性モーメントなどの造形物の性質や内部構造を考慮したモデリング方法についての研究が盛んに行われている。

本研究ではこれらのモデリング技術を応用して、3 次元メッシュモデルから内部構造を考慮し、自立する大きなレゴモデルの組立図を自動生成するシステムを構築する。モデルの内部形状を変化させて重心位置を調整し、自立、安定したレゴモデルを生成する。ここで大きなレゴモデルと

はレゴブロック数が 500 以上のモデルを指す。

この論文の貢献として以下の二つが挙げられる。

- 自立するレゴモデルの組立図の自動生成
- レゴモデルの重心位置の制御

残りの節は以下に記載する。第 2 節では関連研究について述べ、第 3 節では、本研究方法について述べる。第 4 節に結果と考察、第 5 節に結論を述べる。

2. 関連研究

レゴ組立図自動生成の研究として、Kozaki ら[1]の提案した 3 次元形状復元と最適化処理を組み合わせたレゴモデルの組立図を自動生成する方法がある。この手法はまず、対象物をターンテーブルに乗せ、様々な角度から撮影し、ボクセルモデルとして 3 次元形状を復元する。その際、画像から得られた色情報をモデル表面に反映させている。次に強度を保ち、ブロック数を少なくすることを目的とし、最適化手法である焼きなまし法を用いてブロック配置を決めている。この最適化はモデル内部の不要なブロックを削り、その削った空間に柱を配置し、組み立て易さや強度を高める役割を果たしているが、モデルの自立性については言及していない。

上記のリバースエンジニアリングによる方法とは別に、メッシュモデルからボクセルモデルに変換し、自動的にレゴモデルに変換する研究がある[2-9]。Testuz ら[8]は、グラフ構造に基づき、単位ブロックを大きなブロックに変換している。しかし、実際にレゴブロックでレゴマンを作成し

^{†1} 横浜国立大学大学院
Yokohama National University

た結果の図はあるものの、内部構造については言及されていない。Hong ら[9]はボクセルモデルの内部ブロック配置を変え、重心位置を制御して自立するモデルを作成している。しかし、単位ブロックから大きなブロックへの変換は貪欲法を用いており、最適なブロック配置を得られる保証はない。また重心位置を制御したことによるモデルの安定性について言及していない。

一方近年の形状モデリングの研究では、3Dプリンタ等で実物を作ることを考慮した研究が行われている。Stava ら[10]は、メッシュモデルに対して強度の弱い箇所には厚みを持たせ、またモデルの内部や外部に柱を設けることで全体の強度を高めている。Bächer ら[11]は、モデルの内部形状をボクセル単位で変化させ、モデル全体の慣性モーメントを調整し、実際に3Dプリンタで様々な形状の独楽を作製している。Prévost ら[12]は、内部構造や表面形状を変化させ、メッシュモデルの重心位置を制御し、自立するモデルを作成している。

本研究ではこの技術の一部をレゴのモデリングに応用する。ここで3Dプリンタとレゴのモデリング技術の違いとして、自由度の違いが挙げられる。例えばレゴのモデリングにおいて、モデルの解像度によって内部構造を考慮できない場合や使用ブロックによって配置を制限される場合がある。そのため、本研究では自由度の低いレゴモデルに対して、最適化手法を用いて内部構造を変化させることで自立して安定するレゴモデルを自動生成する。

本研究で述べる自立性とは、レゴモデルを作成した際に倒れずに立ち続けることで、安定性とは立ち続けて、さらに倒れにくいことである。

3. 研究方法

以下に、研究の流れを紹介する。

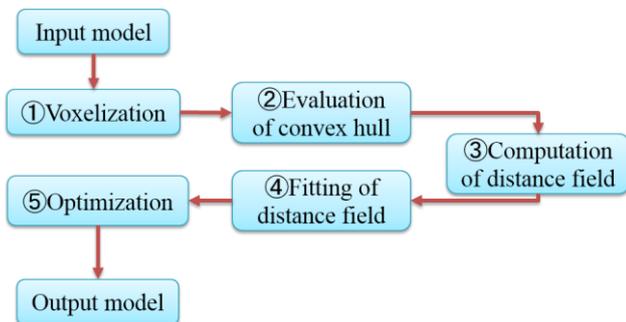


Figure 1 Flow chart of our research

- ① メッシュモデルをボクセルモデルに変換する。(3.1項)
- ② ボクセルモデルからconvex hullを作成する。(3.2項)
- ③ Convex hullの境界から距離場を算出する。(3.3項)
- ④ 離散的な距離場を連続したデータへ補間する。(3.4項)

⑤ 重心位置を考慮したレゴ配置を決定する。(3.5項)

今回、使用するレゴブロックは、Kozaki ら[1]と同様にFig.2に示すブロック(1×1, 1×2, 1×3, 1×4, 2×2, 2×3, 2×4)を使用する。

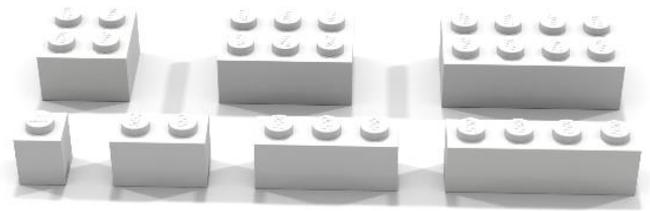


Figure 2: LEGO bricks

3.1 メッシュからボクセルの変換

縦、横、高さの比が5:5:6である1×1レゴブロックを単位ボクセルとして、閉じたメッシュモデル(Fig.3(a)参照)をボクセルモデル(Fig.3(b)参照)に変換する。まず、L×M×Nのボクセルを用意し、これらの各ボクセルの重心とメッシュを含む平面との距離が閾値以下であればボクセルを残し、それ以外は削除することにより、メッシュモデルの表面にボクセルを配置する。次に、配置されたボクセルを用いて、メッシュモデルの内部にボクセルを配置する。これにより、ボクセルが表面から内部まで全て詰まった状態となる。この後の操作で、モデル内部に配置したボクセルを削除し、重心位置を制御することでモデルを自立させる。モデルの自立性は生成したボクセルモデルの地面と接触している箇所のconvex hullを用いて、評価する。

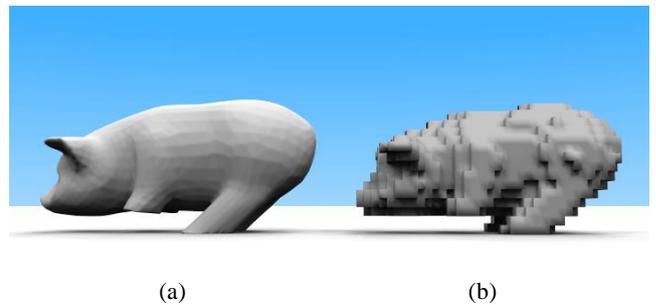


Figure 3: (a) Input mesh model. (b) Voxel model.

3.2 Convex hullの作成

ボクセルモデルの最下層の地面と繋がるボクセルの集合(Fig.4(a)参照)から、Andrew's Monotone Chain convex hull algorithm [13]を用いて、地面と繋がるすべての2次元ボクセルのconvex hullを作成する(Fig.4(b)参照)。この時、モデルの重心を地面に投影した点がこのconvex hull内にあればそのモデルは自立し、さらにconvex hullの境界から離れているほど、倒れにくく安定したモデルとなる。従って可能な限りモデルの重心がconvex hullの内部で且つ境界から離れるようにする。その為にモデルの最下層の2次元平面のconvex hullの境界からの距離場を計算する。

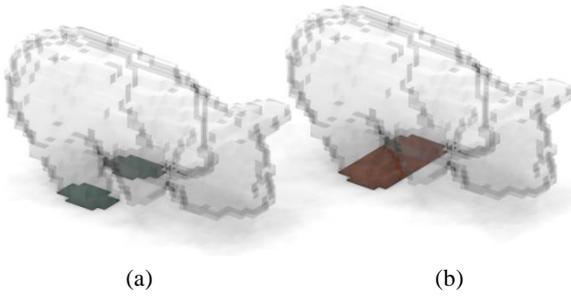


Figure 4: Computation of convex hulls. (a) Voxels having contact with the ground are in green color. (b) Convex hull of (a) is in brown color.

3.3 Fast Marching Method による距離計算

ボクセルモデルの最下層の2次元平面の距離場 $d(x)$ を求める。今回、convex hull 境界のボクセルから速度 F で境界が内側と外側に広がる場合を考える。この時各ボクセル境界の到着時間 $T(x)$ は、次の Eikonal 方程式を満たす。

$$|\nabla T(\mathbf{x})|F = 1 \quad (1)$$

本研究では、ボクセル分割された領域に対して、Sethian が提案した Fast Marching Method (以下 FMM)[14] を適用する。FMM は、境界が一方向に進むという条件を加えることで高速に方程式を解く数値解法である。Convex hull の境界ボクセルを初期位置として、到達時間の小さい場所から順に Eikonal 方程式を解き、各点の到達時間を確定する。これを対象となる空間全てを被覆するまで繰り返すことによって、各ボクセルにおける境界の到達時間を求めることができる。ここで $F=1$ とすることで、各ボクセルの到達時間が距離場 $d(x)$ と等しくなり、 $d(x)$ を求めることができる。

Convex hull の境界の距離を 0 として、convex hull 内部を正の距離値、外部を負の距離値にする。3.5 項で述べる最適化処理はローカルな処理である為、重心の変化量は1ブロックに対して小さい。従って算出した離散データではなく、重心の変化を敏感に捉えることのできる連続した距離データが必要になる。そこで、B-spline 関数を用いて離散データを補間し、連続したデータを生成する。

3.4 反復幾何処理手法による B-spline 曲面補間

離散的な点群データを入力として B-spline 曲面補間を行うことで、次式を得ることが出来る。

$$\mathbf{R}(u, v) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{R}_{i,j} N_{i,K}(u) N_{j,L}(v) \quad (2)$$

ここで、 K, L は階数、 $\mathbf{R}_{i,j}$ は制御点、 $N_{i,K}(u), N_{j,L}(v)$ は B-spline 基底関数を表す。一般に、B-spline 曲面補間は制御点 $\mathbf{R}_{i,j}$ を未知数とした線形システムを解く必要がある[15]。しかし、本研究で用いる反復幾何処理手法は線形システムを解かず、簡単な幾何学的操作により制御点を移動し、入

力データと B-spline 曲面との誤差を反復的に減少させる手法である[16]。この手法を用いるメリットとして、一般的な手法に比べると実装が容易である、精度に合わせたフィットが可能である等が挙げられる。また、佐々木ら[17]はグリッド上に与えられた特性データに対し、反復幾何処理手法を用いて unclamped の一様 B-spline 関数でフィットする手法を提案した。本研究では佐々木らの手法を用い、 $(M+1) \times (N+1)$ のグリッド上に与えられた離散データ $\mathbf{Q}_{i,j} = (x_i, y_j, d_{i,j})$ ($i = 0, \dots, M, j = 0, \dots, N$) (Fig.5(a) 参照) を unclamped の一様 B-spline 曲面で補間した (Fig.5(b)参照)。ただし $d_{i,j}$ は FMM で求めた (x_i, y_j) における距離場を表す。

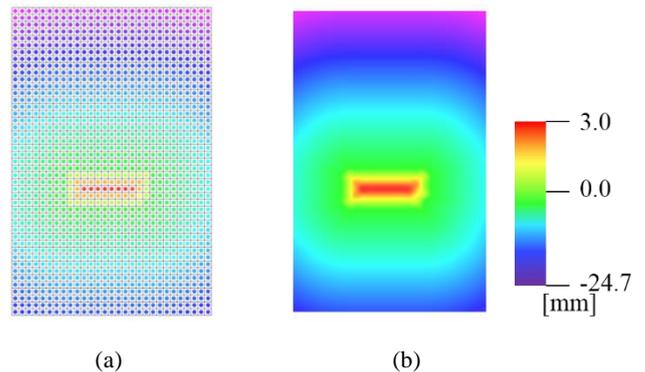


Figure 5: (a) Discrete distance field as a result of FMM. (b) Continuous distance field as a result of B-spline fitting.

3.5 最適化手法を用いたレゴブロック配置の決定

Kozaki ら[1]は、表面の形状と色情報を保ちつつ、内部を削り、強度を維持するように、ブロック配置を焼きなまし法で最適化している。初期状態は、 1×1 のブロックがモデル内部まで詰まった状態である。ブロック配置決め方法は単純で、ブロックの統合、分割、削除の3つの操作(Fig.6 参照)を繰り返すことである。

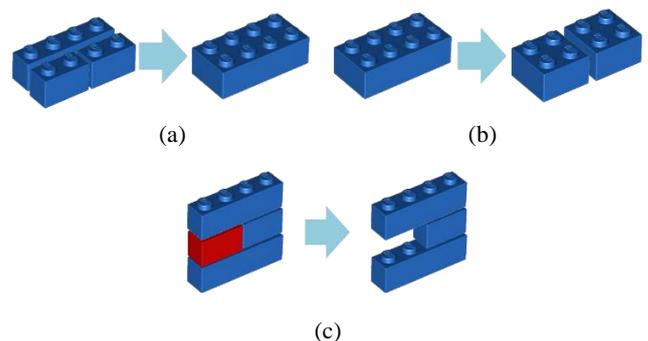


Figure 6: Three operations: (a) Merge. (b) Split. (c) Remove.

焼きなまし法では、温度を下げることで式 (3) に示すエネルギーの増加を防ぎ、3つの操作回数が次第に0に収束し、ブロックの配置が決定する。これにより、モデル内部を空洞にし、柱を配置したモデルが生成できる。本研究では、Kozaki ら[1]の用いたエネルギー関数に第5項として、重心の位置に影響を与える項を付加する (式 (3))。それぞれの

項は線形結合になっており、独立している。第1項目は、表面からの距離を考慮したもので、モデルの厚みに影響を与える。第2項目と第3項目はそれぞれ水平、鉛直方向のブロックの繋がりを考慮したものでモデル全体の強度に影響を与える。第4項目はモーメントを考慮したもので、作る際の荷重や強度に影響を与える。本研究で付加する第5項目は、重心位置の変化を考慮したものである。

$$E(M) = w_1 E_D(M) + w_2 E_V(M) + w_3 E_H(M) + w_4 E_M(M) + w_5 E_C(M) \quad (3)$$

式(3)の E_C は重心によるエネルギーの項である。 $w_i (i = 1, \dots, 5)$ は重みを示している。式(4)に示すようにモデルの重心位置を \mathbf{C} として、 xy 平面に投影した点をバウンディングボックスで正規化した値を (u, v) として式(2)に代入し、 xy 平面の距離の値を得る。Fig.7に示すように、変更前後の重心を投影することで距離値を得ることができる。

$$\Delta E_C = \mathbf{R} \begin{pmatrix} C_x \\ C_y \end{pmatrix} - \mathbf{R} \begin{pmatrix} C'_x \\ C'_y \end{pmatrix} \quad (4)$$

$$\mathbf{C} = (C_x, C_y) = \frac{1}{n} \sum_{i=1}^n \mathbf{P}_i \quad (5)$$

$$\mathbf{C}' = (C'_x, C'_y) = \frac{1}{n \pm n'} \left(\sum_{i=1}^n \mathbf{P}_i \pm \sum_{j=1}^{n'} \mathbf{P}'_j \right) \quad (6)$$

ここで $\mathbf{B} = (B_x, B_y)$ はモデルのバウンディングボックスで、 \mathbf{C}' は変更後の重心の位置である。 n' は、ローカルに増減する単位ブロックの数、 n は、変更前の使用している単位ブロックの数で、 \mathbf{P}_i 、 \mathbf{P}'_j は単位ブロックの重心位置である。ローカルな処理により、式(4)から変化量を得ることで重心の位置をconvex hull内に移動する。

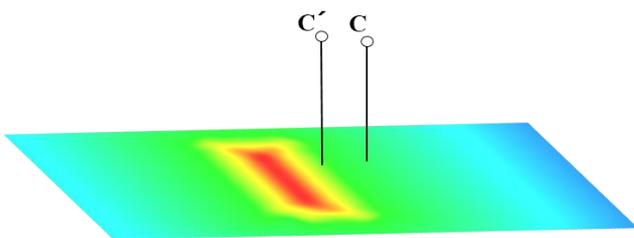


Figure 7: Change of the centroid positions.

4. 結果と考察

本手法をFig.8に示すpig, kittenとStanford bunnyの3つのモデルに適用し、本研究の有効性を確認する。本手法とKozakiら[1]の結果をTab.2に記載し、図をFig.9, その断面図をFig.10に載せる。Tab.2のbeforeはモデルの初期状態(中身が詰まった状態)の重心位置のconvex hullの境界からの距離を示す。結果の後、過去の研究と我々の手法の違いを記載する。

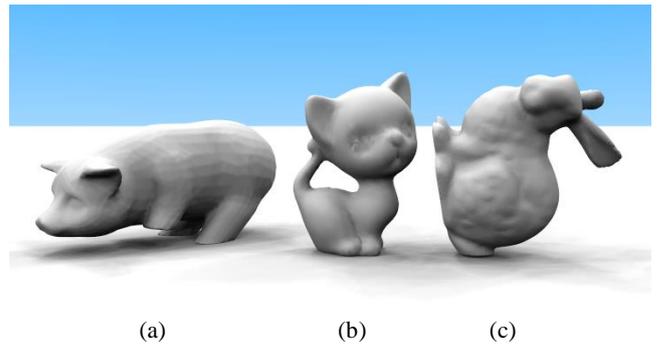


Figure 8: Input mesh models. (a) pig. (b) kitten. (c) Stanford bunny.

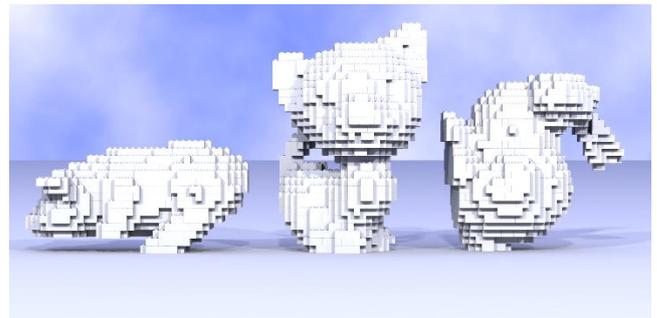


Figure 9: Computed LEGO models. (a) pig. (b) kitten. (c) Stanford bunny.



Figure 10: Cross sections of Figure 9. Bricks in purple color are inside bricks.

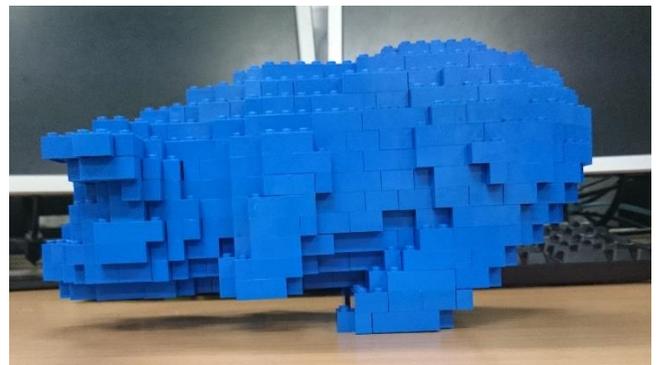


Figure 11: Real pig LEGO model.

Table 1: Comparative data between our method and other methods [1,8,9].

	Input	Real LEGO	Convert method	Color information	Thickness adjustment	Centroid adjustment	Building instructions
Hong et al.[9]	Mesh	Yes	Greedy	No	No	Yes	No
Testuz et al.[8]	Mesh	Yes	Greedy	No	No	No	No
Kozaki et al.[1]	Real object	Yes	SA	Yes	Yes	No	Yes
Our method	Mesh	Yes	SA	No	Yes	Yes	Yes

Table 2: Comparative data between Kozaki method [1] and our method.

		# of bricks	Void ratio(%)	Centroid	
				before	after
Pig	Kozaki [1]	591	25.3	0.167	-0.187
	our method	632	24.5		
Kitten	Kozaki [1]	957	27.9	4.71	4.87
	our method	975	30.1		
Staford bunny	Kozaki [1]	813	25.2	-0.761	-0.952
	our method	846	30.6		

それぞれのモデルの重心位置は Tab.2 より convex hull 内に移動したことが確認できる。これより、3 つのモデルで自立するモデルとなった。一方、Kozaki らの手法では Stanford bunny と pig モデルにおいて、それぞれの重心位置が convex hull 内に存在せず、自立しないモデルとなった。我々は実際に pig モデルを作り、自立し安定したモデルを作成できることを確認した (Fig.11)。Kitten のモデルでは、重心位置を更に中央付近まで移動した配置となり、安定性が増した。しかし Kozaki らの手法と比較して、本手法では総ブロック数が増した。その理由として、Kozaki らは一定の厚みを持たせていたが、我々の手法では重心移動のために、厚みのある箇所や厚みの薄い箇所が生じた為である。そのため、空隙率の高いモデルとなったが、部分的に強度の低い箇所がある。

過去研究の手法との違いを Tab.1 に記載する。我々の手法は、メッシュから始まっており、色情報を持っていない。また多くの研究は組立図を記載しておらず、実際に作ることが可能かどうかの判断ができなかった。そこで、我々は付録に pig の組立図を記載する。Fig.11 はこの組立図を基に組立てた。

5. おわりに

本研究において、倒れそうなメッシュモデルから重心位置を考慮し、自立したレゴモデルを作成できる方法を提案した。また、自立するモデルであっても重心位置を制御することで倒れにくい安定したレゴブロック配置を示すことができた。これにより、完成品が自立せず倒れてしまう問

題を解決することができた。

5.1 制限

モデルの腕が細い場合、組立てた時に自重で壊れてしまうので組立てできない。内部に空隙を設けても重心移動に限界があるので、必ずしも全てのモデルで自立はできない。

5.2 今後の展望

重心特性による強度を考慮したブロック配置を考える必要がある。また、重心移動では自立できないモデルに対して、モデル自体の形状を変化させるか、モデルの外部にサポートの柱を入れることで安定した姿勢にすることができると考えられる。

謝辞

本研究は科学研究費補助金(15H03944)の助成を受けたものである。レゴブロックを作成するにあたり、研究室のメンバーの多大なる協力を頂いた。ここに感謝の意を表す。モデルは AIM@SHAPE Shape Repository によって提供されたものを、Meshmixer を使って形状を変化させて使用した。

参考文献

- 1) Kozaki, T., Tadenuma, H., and Maekawa, T.: Automatic generation of LEGO building instructions from multiple photographic images of real objects, *Computer-Aided Design*, Vol.70, pp.13-22 (2016). [Proceedings of the GD/SPM 2015]
- 2) Gower, R.A.H., Heydtmann, A.E., and Petersen, H.G.: LEGO: Automated model construction, 32nd European Study Group with Industry - Final Report, pp.81-94 (1998).
- 3) van Zijl, L., and Smal, E.: Cellular automata with cell clustering, *Proceedings of AUTOMATA 2008 Workshop*, pp.425-440 (2008).
- 4) 北川佑樹, 高井昌彰, 高井那美: レゴブロックによる近似形状制作支援システム, *情報処理学会全国大会*, Vol.2013, No.1, pp.285-287 (2013).
- 5) 小野純明, Andre Alexis, 張英夏, 中嶋正之: 作りやすさを考慮したブロック玩具作品組立手順の自動生成に関する研究, *映像情報メディア学会技術報告*, Vol.36, No.16, pp.23-26 (2012).
- 6) 林慧, 前川卓: 複雑な物体の複数画像からのレゴブロックによる再構築, *情報処理学会研究報告 グラフィクスと CAD 研究会報告*, Vol.2013, No.9, pp.1-6 (2013).
- 7) 狐崎拓哉, 前川卓: 組み立てやすさを考慮したレゴブロックモデルによる複雑な物体の再構築, *日本機械学会 関東支部第 21 期総会・講演会*, (2015)
- 8) Testuz, R., Schwartzburg, Y., Pauly, M.: Automatic generation of

constructable brick sculptures, Proceedings of Eurographics 2013 (short paper), pp.81-84 (2013).

9) Hong, J.Y., Way, D.L., Shih, Z.C., Tai, W.K., and Chang, C.C.: Inner engraving for the creation of a balanced LEGO sculpture, The Visual Computer, pp.1-10 (2015).

10) Stava, O., Vanek, J., Benes, B., Carr, N., and Měch, R.: Stress relief: improving structural strength of 3D printable objects, ACM Transactions on Graphics, Vol.31, No.4 (2012). [Proceedings of SIGGRAPH 2012]

11) Bäcker, M., Whiting, E., Bickel, B., and Sorkine-Hornung, O.: Spin-it: optimizing moment of inertia for spinnable objects, ACM Transactions on Graphics, Vol.33, No.4 (2014). [Proceedings of SIGGRAPH 2014]

12) Prévost, R., Whiting, E., Lefebvre, S., and Sorkine-Hornung, O.: Make it stand: balancing shapes for 3D fabrication, ACM Transactions on Graphics, Vol.32, No.4 (2013). [Proceedings of SIGGRAPH 2013]

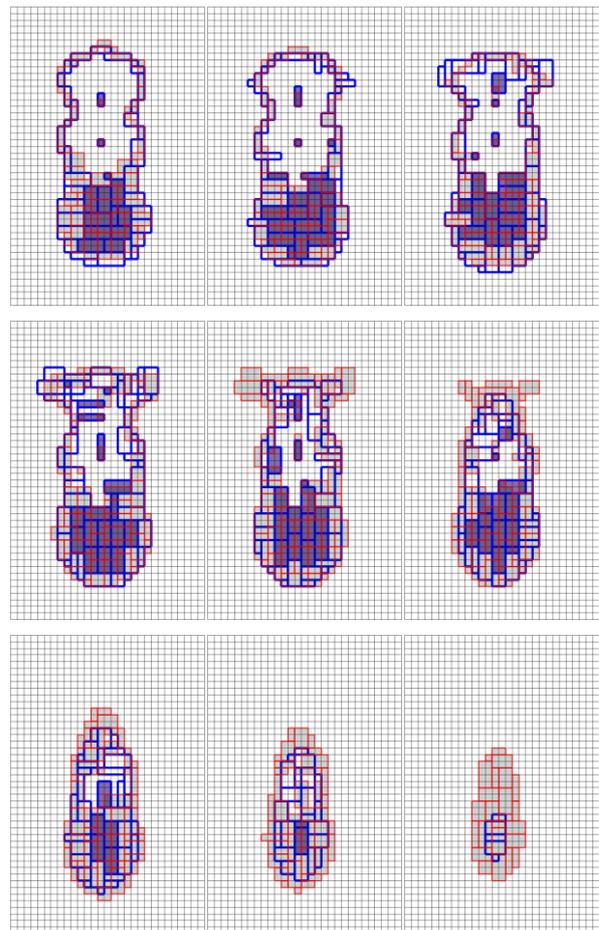
13) Andrew, A.M.: Another efficient algorithm for convex hulls in two dimensions, Information Processing Letters, Vol.9, No.5, pp.216-219 (1979).

14) Sethian, J.A.: Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science, Cambridge University Press, (1999).

15) Piegl, L. and Tiller, W.: The NURBS book 2nd Ed. Springer-Verlag New York, Inc., (1997).

16) Kineri, Y., Wang, M., Lin, H., and Maekawa, T.: B-spline surface fitting by iterative geometric interpolation/approximation algorithms, Computer-Aided Design, Vol.44, No.7, pp.697-708 (2012).

17) 佐々木雄飛, 竹澤正仁, 金勝基, 前川卓: 幾何処理による離散データの B-spline volume 補間. 日本機械学会 第 24 回設計工学・システム部門講演会論文集, (2014).



付録

Fig の組立図を以下に示す. 青枠が現在の層のブロック配置を示し, 赤枠が一つ下の層のブロック配置を示す. 青枠の内部の色が使用するレゴブロックの色であり, 紫色で示すブロックは内部ブロックである.

