

## レゴブロックを用いた影のトリックアート構築システム

## LEGO Block Modeling System to Make a Trick Artwork Using Shadows

生田 寛和†

高井 昌彰††

高井 那美‡

Hirokazu Ikuta

Yoshiaki Takai

Nami Takai

## 1. はじめに

芸術作品の一つに影を用いたトリックアートがある。この作品は、Fig.1のようにその立体形状を直接見ただけではどのようなものか容易に判別できないが、ある方向から作品に光をあて、背景に影をつくることによって、意味のある様々な形状を創出するものである。Fig.1左のオブジェクトからはその影を判別できないが、光を当てることでFig.1右のように座り込んだ人物の影を確認することができる。

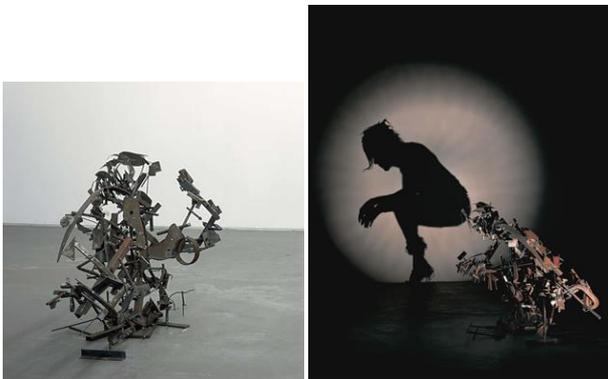


Fig1. 影のトリックアート HE/SHE[1]

一般に、影のトリックアートの製作を行う場合、立体形状を構成する素材の選択、素材の接着、影を確認しながらの製作等の観点から、トリックアートに興味を持った初心者が容易に製作し得るとは言い難い。

そこで本研究では、目標とする異なる複数の影画像を入力として与え、これを基に立体造形可能な3Dモデルを出力する影のトリックアート製作の支援システムを開発した。この3Dモデルに対していくつかの方向から光を当てることにより、目標とする影を映し出すトリックアートとなる。3Dモデルの構築にはレゴブロックを使用しており、それによって問題であった素材の選択、接着を解消し、システム側で3Dモデルを自動構築することで初心者でも容易に影のトリックアートを製作することが可能となる。

ブロック配置の最適化には、レゴブロックの連結性を考慮した山登り法、焼きなまし法、遺伝的アルゴリズムを適用し、異なる4方向からの影画像生成に対応可能である。本システムにより得られた3Dモデルと実際に構築したレゴブロック作品をもとに、システムの有効性や形状再現性について評価する。

## 2. 影の評価方法

本システムでは得られた影と入力画像がどの程度相似であるかを評価するため、再現率(0~100の範囲を有する値)を定義した。この再現率はテンプレートマッチングに基づいており、画像のエッジ部分のマッチングを重視するとともに、画像の枠外にブロックが置かれる場合のペナルティを考慮している。これは、画像が似ているかどうかを判断する際、エッジ部分の正否は画像の内側部分の正否よりも人の視覚認識において重要であると考えたためである。

各画像の再現率の積を3Dモデル全体の再現率とし、この値が高まるように3Dモデルの生成とブロックの追加・削除によるモデル形状の修正を行い、得られる影を入力画像に近づけていく。



Fig2. 再現率

## 3. システムの構成

本システムの全体の構成をFig.3に示す。ユーザは入力として複数の二値画像データと、3Dモデルの大きさ(ブロックの解像度)を与えることで、球形の初期形状で構成された3Dモデルに対し、システムによって自動決定された光源の位置からのシルエット法を適用する。次に、シルエット法で残った共通部分に対して、ブロックの追加・削除による配置の最適化を行った後、実際にレゴブロックで組み上げることが出来るようにするためのレゴブロック化処理を行う。以上の3つのステップを行い、影のトリックアートの3Dモデルと設計図の出力をする。

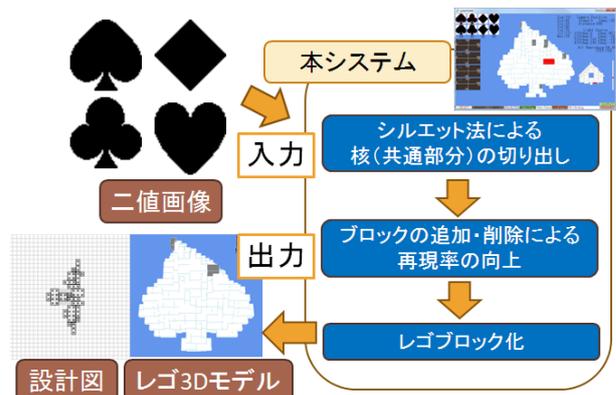


Fig3. システムの構成

†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 シルエット法の適用

本システムでは、はじめにレゴブロックの1×1ブロックを最小単位とした球で構成される3Dモデルに対して、光源の位置からシルエット法を適用する。この際の光源は、二値画像の枚数から複数の光源の干渉が最も少なくなる位置、もしくはシルエット法によって得られる全体の再現率が最も高くなる点に自動決定される(手動で光源の位置を決めることもできる)。これにより得られた3Dモデルは、全二値画像の共通部分となっているため、この3Dモデルを探索の出発点としてブロックの追加・削除を行い、配置の最適化を進める。

Fig.4はシルエット法を適用した初期段階の結果を示している。図の右上のアルファベット(A,B,C,D)は入力として与えられた二値画像であり、右下は3Dモデルから得られた影のイメージとそれらの再現率である。

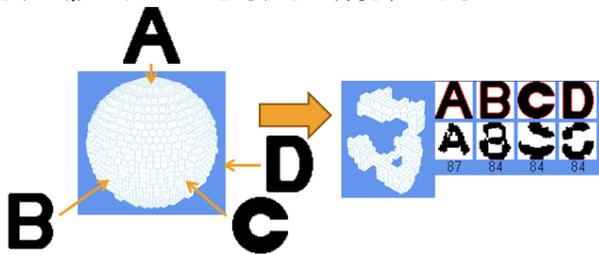


Fig.4 シルエット法の適用

### 3.2 ブロック配置の最適化

シルエット法で得られた3Dモデルに対して、レゴブロックの1×1ブロックに相当する大きさのブロックを単位とした追加・削除を繰り返し行うことで、モデルの影を入力画像に近づけていく。

ブロックの配置場所の探索においてはブロックの連結性を考慮し、各ブロックに六方向で隣接している部分を追加可能な場所、削除してもブロックが空間に孤立しない部分を削除可能な場所とし、これを近傍とみなした探索問題として再現率が最も高くなるよう最適化を行う。最適化手法には山登り法、焼きなまし法、遺伝的アルゴリズムの3手法を用意した。

焼きなまし法のパラメータは更新回数を500、初期温度を1とし、更新ごとに係数0.996を温度に乗じてアニーリングを行う。

遺伝的アルゴリズムは3Dモデルのブロックの有無を遺伝子とし、シルエット法を適用した状態の3Dモデルにランダムノイズを加えた60の個体が初期世代となる。これらの個体から全体の再現率が最も高い個体と入力画像毎の再現率が最も高い個体を入力画像の数だけ選択、残った個体の中から選択された個体の数の合計が15になるまでルーレット選択を行う。交叉は各個体同士の1部分を複数回交換することで行われる。

### 3.3 レゴブロック化

最適化された3Dモデルはレゴブロックの1×1サイズを基本として構築されているのでこのままでは実際に制作を行うことが出来ない。組み立て可能な作品を得るには、隣接との結合性を満たすように、様々な形状のレゴブロックを用いた形に変換する必要がある。本研究ではレゴブロックの制作においてなじみ深い1×1, 1×2, 1×3, 2×2, 2×3, 2×4の6種類のレゴブロックを扱う。

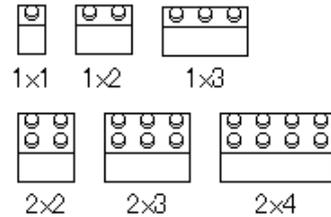


Fig.5 レゴブロックの種類

本システムは、3Dモデルをレゴブロック化する研究[2]の手法を元に、2×2幅の格子状に区切られた空間を奇数層と偶数層で1×1ずつずらし、この格子を基準に2×2以上の大きさのブロックを配置する。また奇数層と偶数層で配置するブロックの向き(長辺方向)を90度回転させる。これらの構築処理による形状生成後に全体の連結判定を行い、ブロック間の未接続部分が一定数以下に減少するまで、ブロックの再配置と構築処理を繰り返す。

### 3.4 レゴブロック組み上げ支援

本システムは、先に述べた3Dモデル出力の一連の処理以外に、ユーザが影のトリックアートを製作する上で支援となる機能を実装している。ブロックの未接続部分や3Dモデルが二つ以上のクラスターに大きく分かれてしまう場合には、これらをもとに色分けによって可視化し、作品の組み立て作業を視覚的に支援する機能や、得られた3Dモデルを実際にレゴブロックで制作する際に必要となる各層毎の配置図出力機能、ポインタを用いて手動でのブロック追加・削除によるレゴブロックの修正機能がある。

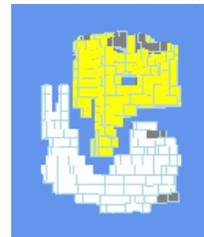


Fig.6 未接続部分とクラスター毎の色分け

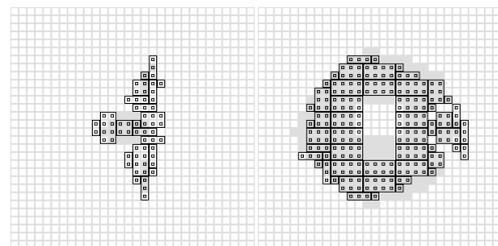


Fig.7 トリックアート作品の設計図

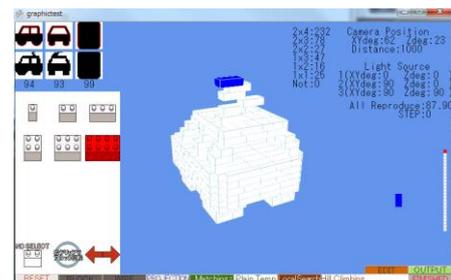


Fig.8 手動でのレゴブロック修正

さらに、3D モデルへのポインタ指示による対応する層の位置表示機能や、得られた 3D モデルデータのファイル保存機能等を実現しており、ユーザが影のトリックアートを製作する上で実用的な諸機能を提供している。

#### 4. 実装と実行結果

本システムの実装は、開発言語に C# を、GUI 作成と 3D モデル描画に XNA4.0[3] を使用した。システムの実行環境は Table.1 の通りである。

CPU	Intel Xeon 5160 2.99GHz
RAM	12.0GB
OS	Windows 7 Professional x64
GPU	NVIDIA Quadro FX 3500

Table.1 実行環境

本システムを評価するため、3D モデル生成に要する処理時間等の計測に加え、いくつかの二値画像群からの 3D モデル生成と、そのモデルに基づいたレゴブロックによるトリックアート作品の製作を行った。それらの結果を以下に示す。

##### 4.1 実行時間

Fig.9 の入力画像群に対する 3D モデル構築に要する処理時間 (ステップ数) を、初期形状の層数ごとに Table.2 に示す。ここでステップ数とはレゴブロックの追加・削除を行った回数であり、最適化手法には山登り法を用いている。



Fig.9 入力画像

層数	12	18	24
ステップ数	99	151	374
実行時間[s]	6.69	16.94	51.27
1 ステップ の実行時間 [ms]	67.6	112.2	137.1
層数	30	36	42
ステップ数	707	1129	1823
実行時間[s]	111.84	250.86	515.36
1 ステップ の実行時間 [ms]	158.2	222.2	282.7

Table.2 モデル構築の実行時間 (山登り法)

この結果から、層数  $x$  に対する実行時間  $f(x)$  を最少二乗法で近似すると以下ようになる。

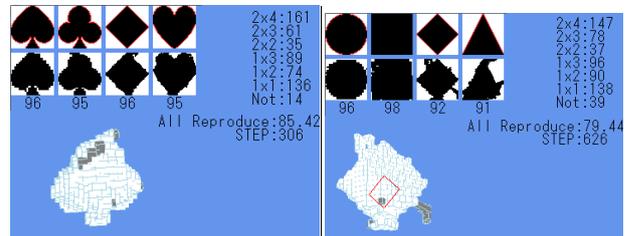
$$f(x) = 257 - 29.5x + 0.838x^2$$

初心者支援の観点から 1 時間以内でレゴブロックの最適化を完了させる場合を想定すると、本システムで 83 層まで対応可能であり、高さ約 80cm のレゴブロック作品まで適用できるものと推定される。

##### 4.2 4つの入力画像を用いた例

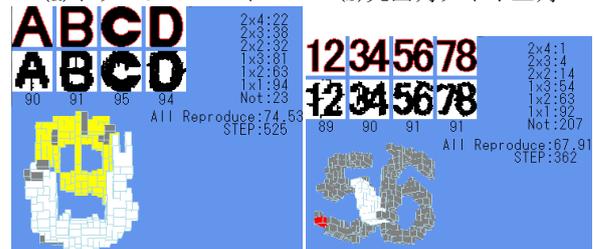
光源から出る光線同士の干渉が大きい 4 つの入力画像を使用した例を示す。Fig.10 では(a)トランプのスイート、(b)円、四角形、ダイヤ、三角形、(c)アルファベットのフォント形状、(d)1 から 8 のアラビア数字のフォント形状を入力としている。入力画像の数が 4 つであっても、どれも概ね類似した影画像が得られている。

しかし Fig.10(c)は入力画像がやや複雑であるためか、3D モデルが二つのクラスターに分割され、Fig.10(d)は殆どのブロックが未接続状態となっている。これより、複雑な入力画像でも類似した影画像は得られるが、複雑さが増すにつれ実際に制作することが困難になっていくことが分かる。このような場合、再現率の向上よりも連結性を重視した最適化手法の考案、またはレゴブロック化の際に未接続の部分を繋げるような処理を実装することで解決することが可能である。



(a)トランプのスイート

(b)丸四角ダイヤ三角



(c)アルファベット

(d)1 から 8 の数字

Fig.10 モデル構築の実行例

##### 4.3 焼きなまし法の効果

Fig.11 は焼きなまし法の活用例である。同じ入力画像群に対し Fig.11(a)は山登り法、Fig.11(b)は焼きなまし法を用いて最適化を行っており、山登り法では月と星の左下部分が欠けているのに対し、焼きなまし法では同部分の形状が再現されている。



(a)山登り法

(b)焼きなまし法

Fig.11 山登り法と焼きなまし法の比較

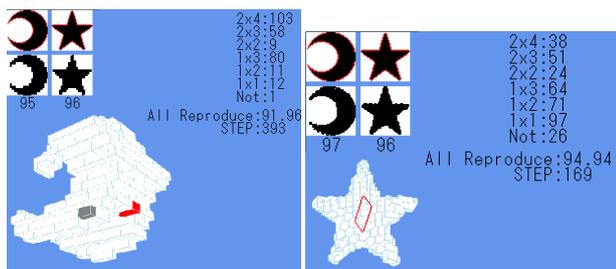
##### 4.4 ジャギーの発生と解消

使用されるレゴブロックは直方体であるため、3D モデルから得られる影もレゴブロックの四角が反映されたジ

ジャギーが発生することがある。

このジャギーは光源がレゴブロックに対し、各面に垂直である場合に発生し、それ以外の位置に光源を配置することで、レゴブロックの立体的な並びがジャギーの補間を行い解消することができる。ジャギーが発生した場合と発生しない場合では、発生しない場合の方が再現率は高くなりやすく、本システムではシルエット法を適用する際に光源を再現率の高まる位置に自動決定するため、ジャギーは発生しにくくなっている。

Fig.12(a)はジャギーが発生した場合であり、レゴブロックの四角が反映されているのが分かる。Fig.12(b)は光源の位置をずらしてジャギーを解消した場合であり、Fig.12(a)に比べて輪郭が滑らかであり、合計の再現率も高い。



(a)ジャギーの発生した場合(b)ジャギーの発生しない場合  
Fig.12 ジャギー発生の有無

#### 4.5 実際のレゴブロックを用いて制作した例

Fig.13はFig.10(a)のモデルを実際のレゴブロックで構築したものである。この制作例のパラメータはTable.3に示す通りである。このレゴブロックの立体形状を見てもどのような影を映し出すかは判別できない。

初期形状である球の層数	25
最適化に要した時間[s]	49
ステップ数	305
使用したブロック数	556
完成した制作例の層数	18
制作例の大きさ[cm]	17.25
組み立て作業時間[h]	3

Table3. 制作例のパラメータ

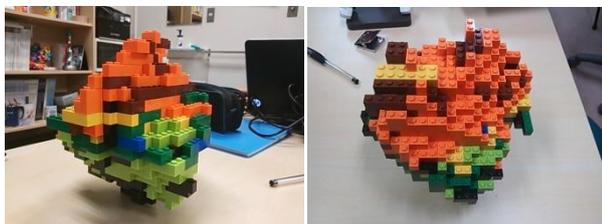


Fig.13 レゴブロックで構築したトリックアート

このレゴブロックのオブジェに対し4方向から光を当てて得られた実際の影の様子をFig.14に示す。4つの影がそれぞれトランプのスイートを表しているのが分かる。



Fig.14 トリックアートが映し出す影

#### 5. まとめと今後の課題

本稿では、光を当てることで目標とする複数の影画像を映し出すトリックアートの制作支援として、レゴブロックを用いて連結性を考慮したトリックアート生成システムを構築し、実際にレゴブロック作品を構築することで本システムの有効性を示した。

今後の課題としてはレゴブロックの連結性を高める探索手法の改善、入力画像の回転や拡大・縮小への対応によるより柔軟な評価方法の追加、入力の影画像の形と探索手法が得られる影に及ぼす影響の考察が挙げられる。

#### 参考文献

- [1] Tim Noble & Sue Webster - HE/SHE, (Diptych) 2004 [http://www.timnobleandsuewebster.com/he\\_she\\_2004.html](http://www.timnobleandsuewebster.com/he_she_2004.html)
- [2] 北川佑樹, 高井昌彰, 高井那美: “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”, 第12回情報科学技術フォーラム(FIT2013)論文集, R0-001, Vol.4, pp.71-72, (2013).
- [3] XNA Developer Center <http://msdn.microsoft.com/en-us/aa937791.aspx>
- [4] 生田寛和, 高井昌彰, 高井那美: “レゴブロックを用いた影のトリックアート構築システムの開発”, 情報処理学会第76回全国大会, 12C-5, (2014).