

バルーンアートのデザイン支援のための 複数本バルーンモデルの構造化と制作手順の自動生成

浦正広¹⁾ (正会員) 山田雅之²⁾ (正会員) 宮田一乗¹⁾ (正会員)
遠藤守³⁾ (正会員) 宮崎慎也²⁾ (正会員) 安田孝美³⁾ (正会員)

1) 北陸先端科学技術大学院大学知識科学研究科

2) 中京大学工学部

3) 名古屋大学大学院情報科学研究科

Formalization and Automatic Generation of Multiple Balloons Modeling for Supporting Balloon Art Design

Masahiro Ura¹⁾ (Member) Masashi Yamaeda²⁾ (Member) Kazunori Miyata¹⁾ (Member)
Mamoru Endo³⁾ (Member) Shinya Miyazaki²⁾ (Member) Takami Yasuda³⁾ (Member)

1) School of Knowledge Science, Japan Advanced Institute of Science and Technology

2) School of Engineering, Chukyo University

3) Graduate School of Information Science, Nagoya University

{ura, miyata} @ jaist.ac.jp

{myamada, miyazaki} @ sist.chukyo-u.ac.jp {endo, yasuda} @ is.nagoya-u.ac.jp

アブストラクト

バルーンアートは、細長いバルーンをひねる操作の繰り返しにより動物などの様々な形状を制作する造形である。著者らはこれまでに1本のバルーンからなるバルーンアートについて、グラフ理論を活用した構造化、制作手順導出、難易度評価の各手法と、それら手法に基づく制作支援システムを提案している。一方で、バルーンアートにおいて複雑な形状を制作する場合には、一般的に複数本のバルーンを用いる。その際、パーツごとに色分けを行うなど、配色に意味付けがなされる場合も多い。そこで本論文では、複数本のバルーンからなるバルーンアートの形状の性質を踏まえて、バルーンアートの構造を定義したグラフをその制作に必要な最少の本数に分割することで、制作手順を生成する手法について提案する。また、この提案手法に基づいて、入力した形状と指定した配色に応じてグラフを分割してその制作手順を生成する、バルーンアートデザイン支援システムを開発する。

Abstract

Balloon modeling is the artistic technique of twisting balloons into various shapes. We have proposed a method for formalization and rating difficulty of balloon modeling made of one balloon based on graph theory, and developed a system for supporting balloon modeling making based on the method. In this paper, we propose a method for separating a graph which is defined as a structure of balloon modeling made of multiple balloons into the minimum number that is needed for making. We also develop a design support system that users can set shape and color of balloons based on proposed method.

1. はじめに

材料を加工して形状を作り上げる造形には様々な種類があり、また、その目的も実用的な機能、作品としての芸術的な表現、あそびや知育など多岐にわたる。材料の特性に応じた加工手法により作られる造形物は、その造形特有の構造をもつ。個々の造形について構造の特徴がわかれば、その構造を数理的に表したり簡潔に記述できる。また、加工手法を有限種の操作として表現できれば、操作の組み合わせと造形物の構造との対応がわかってくる。これらは造形物や造形プロセスのコンピュータによる記録、分析、造形支援へと発展させられ、既に様々な造形についてこのような試みがなされている。例として折り紙を取りあげると、折り紙の各面をノード、折り目をエッジとしたグラフにより形状構造を表現し、コンピュータ上での対話的な操作を実現したり、完成した折り紙のグラフ構造から作成プロセスを推論する研究などが行われている[1,2,3]。

バルーンアート（正確にはバルーンモデリングであるが、本論文では一般に馴染みのある名称を用いる）は、細長いバルーンを用いて動物などの形状を制作する造形である[4,5,6,7]。1本もしくは複数本のバルーンにより様々な形状を制作する。バルーンをひねるという単純な操作の反復により複雑な形状が制作できるが、反面、完成形状から制作方法を推測することは容易ではない。一般的には解説書等で制作手順が示され、制作者はそれに基づいて形状を制作するため、コンピュータにより自由な形状の制作を支援するためには、概形を入力することでその構造を解析し、解析結果に基づいた制作手順を導出する、といった要素が必要となる。コンピュータ上でバルーンアートを支援する試みとして、バルーンアートを題材としたモデリング教育ソフトが存在する[8]。また、バルーンの物理的な特性をシミュレートし、コンピュータ上で対話的にバルーンを操作して形状を制作するものも存在する[9]。制作手順を導出するものとしては、多面体を制作するためのアルゴリズムが考案されている[10]。しかしながら、上記の事例では制作物が予め決められている、バルーンの物理シミュレーションに焦点が当てられており制作手順の構造化はなされていない、制作の対象が多面体に限定されるといった制約がそれぞれ存在する。そのため、自由で多種多様なバルーンアートのデザインは行えない。

以上より、著者らは1本のバルーンからなるバルーンアートについて、グラフ理論に基づいた構造解析および難易度評価手法を提案し[11]、その応用により、与えられた概形をバルーンアートとして整形してその制作手順を導出するシステムを開発した[12]。これにより、構造に関する知識がなくても様々なデザインをバルーンアートで制作可能となった。しかしながら、バルーンアートは複数本のバルーンが用いられるケースも多い。その際、たとえば花であれば花びらを赤、葉や茎を緑など、パーツの配色に意味を持たせるケースも数多く見受けられる。

そこで本論文では、複数本のバルーンを用いたバルーンアートを対象に、形状の性質や制作手順の生成手法を論じ[13]、配色も考慮した制作手順が生成可能なデザイン支援システムを提案する。

2. バルーンアートの特徴と構造化

バルーンアートは風船をひねる操作の反復という造形の特徴を持つ。既研究[11]では1本のバルーンにより構成されるバルーンアートを対象に、この特徴や構造をグラフにより表現している。ここではその概略、および、その定義に基づいた複数本バルーンにおける構造定義に必要な要素について論じる。

2.1 バルーンアートの定義と操作

本論文では1本1本の風船を「バルーン」、バルーンで形作られる形状を「バルーンモデル」とよぶ。バルーンアートではバルーンを分割して「バブル」とよぶパートを作り、これを組み合わせて形状を作っていく。

制作において、まず、「先端」に「余分」ができるようバルーンを膨らませ、空気口を結び「結び目」を作る(図1(a))。バブルを作るときは両手で握って逆方向にひねる。この操作を「ツイスト (twist)」とよび、ひねった部分を「ひねり目」と呼ぶ(図1(b))。バブルを作るたびに空気が余分へと流れ、余分の長さが短くなるため、空気口側からバブルを作る。これを「ツイストの方向性」とよぶ。必要数のバブルができたなら残った余分はそのまま使うか(図1(c))、結び目を作った後に切り除く(図1(d))。ここでは、議論を簡潔にするため後者のみ扱う。

バルーンアートでは2本以上のバルーンを使ってバルーンモデルを作る場合もある。一般に n 本のバルーンから構成されるバルーンモデルでは、結び目は合計 $2n$ 個ある。また、ひねり目とバブルの個数をそれぞれ n_t 、 n_b とすると $n_b = n_t + n$ である。

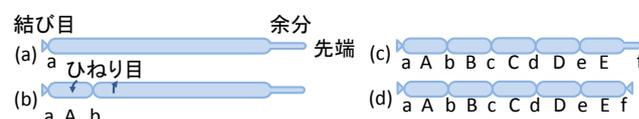


図1. バルーンアートにおける各部の名称。

図2(a)は図1(d)のひねり目bとeを絡み合わせた、図2(b)は図1(d)のひねり目cと結び目fを結んだ状態である。これら操作を行うと空気圧が作用し、ひねり目や結び目が外れなくなる。このとき、ひねり目や結び目が重なった部分を「複合目」、ひねり目、結び目、複合目を総称して「目」とよぶ。この目をつなぎ合わせてお互いを固定する操作を「ロック (lock)」とよぶ。

バルーンアートの書籍では上記操作に加え、図3のように環状になった部分にバブルを差し込んで位置を固定する「スラスト (thrust)」, ツイストとロックを複合した操作、ロックを状況に合わせて細分化した操作、バブルの側面をこすって曲がった状態にする操作なども紹介されている。本論文では、バルーンアートにおける形状および形状制作のプロセスの基本的な性質について論じるため、ツイストとロック操作のみを扱う。

また、必要に応じてバブルや目にラベルをつけ、バブルは大文字の、目は小文字のアルファベットで表記する。

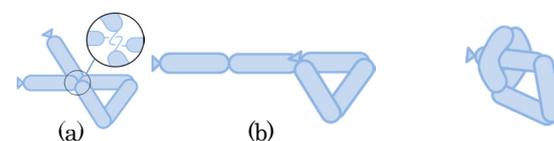


図2. ロックの例。

図3. スラストの例。

2.2 形状構造のグラフ表現

バルーンモデルの構造をグラフで表す。この際、バルーンモデルはこのグラフを3次元空間に埋め込んだものとみなせる。

グラフ G はエッジの集合 E とノードの集合 V の組 $G = (V, E)$ で定義される。バルーンモデルを構成するバブルをエッジに対応させ、目をノードに対応させたものをバルーンモデルのグラフとする。図4(a)の左側はバルーンモデル、右側はそのグラフである。以降ではバルーンモデル M のグラフを G_M で表す。図4(b)の2つのバルーンモデル M_1 と M_2 は図1(d)から作れる異なるバルーンモデルであるが、それらのグラフはラベルを考慮しない場合における同形、つまり位相が等しくなる。

バルーンアートでは、同一形状でも複数の制作手順が存在することが知られている。図4(b)の場合、 M_1 と M_2 は図1(d)から作られるため、ラベルの位置がバブルの位置に対応し、バブルは(d)のラベル順に配置される必要がある。そのため、このケースにおいては、同形であるグラフに変換される M_1 と M_2 はそのグラフの異なる制作手順として扱うことが可能である。

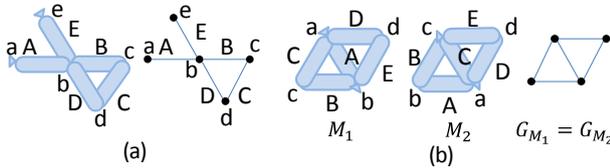


図4 バルーンモデルとグラフの例。

2.3 形状構造の性質

ノードに接続されるエッジの数をそのノードの「次数」といい、次数が偶数のノードを「偶ノード」、奇数のノードを「奇ノード」とよぶ。図5(a)においてb, c, fは偶ノード、a, d, e, gは奇ノードである。次数1のノードは1つの結び目に対応する。次数2のノードは1つのひねり目か、あるいは、2つの結び目をさらに結んだ目に対応する。次数が $2k + l \geq 3$ のノードは k 個のひねり目と l 個の結び目が重なった複合目に対応する。

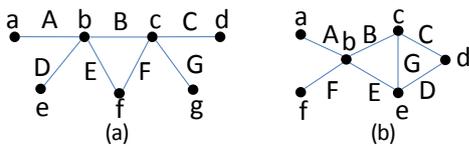


図5. 奇ノードと偶ノードの例。

バルーンモデルを構成するバルーンの本数と奇ノードの個数の間には次の関係がある。

性質1 n 本のバルーンで構成されるバルーンモデル M のグラフ G_M では、奇ノードの個数は偶数(0を含む)、かつ、 $2n$ 以下である。例えば2本のバルーンで構成されるバルーンモデルの場合、そのグラフの奇ノードの個数は0, 2, 4のいずれかである。

略証 いかなるグラフも奇ノードの数は偶数である。また、 M には結び目が $2n$ 個あり、結び目を偶数個含む目に対応するノードは偶ノードであるので、性質1が成り立つ。

性質1とは逆にグラフ G_M が $2n$ 個の奇ノードをもつとき、バルーンモデル M は n 本以上のバルーンを使って作れる。このことを形式的に表すと以下ようになる。

グラフ $G = (V, E)$ のノードとエッジをたどって得られる列 $v_1 e_1 v_2 e_2 v_3 \dots v_{l-1} e_{l-1} v_l$ ($v_i \in V, e_i \in E$)のうち、すべてのエッジが異なるものを「道」とよび、特に、両端のノードが同一である道を「閉路」という。互いに同じエッジを含まない複数の道を「互いに素な道」とよぶ。 m 本の互いに素な道 r_1, r_2, \dots, r_m に現れるエッジの集合 E_1, E_2, \dots, E_m の和が G のすべてのエッジを含むとき、すなわち、 $E = E_1 \cup E_2 \cup \dots \cup E_m$ のとき、これら道の集合 $R = \{r_1, r_2, \dots, r_m\}$ を G の「道分割」とよぶことにする。 R の大きさ $|R|$ について、次の性質がある。

性質2 グラフ $G = (V, E)$ における奇ノードの個数を $2n$ 、 G の道分割を R とすると、 $n \leq |R| \leq |E|$ である。

略証 奇ノードを n 個の組にわけ、各組の2つの奇ノードに接続する仮想的なエッジを n 本追加して得られるグラフを G' とすると、 G' のノードは全て偶ノードである。全て偶ノードであるグラフには全てのエッジを含む1本の閉路があることが知られている。これをオイラー道という。 G' のオイラー道から先に追加した仮想的なエッジを取り除くと、 n 本の道からなる G の道分割 R が得られる。 R の各道は $|R| = |E|$ になるまで分割できるので、 $n \leq |R| \leq |E|$ である。

この性質は、 n 本以上 $|E|$ 本以下のバルーンにより、 G を構造とするバルーンモデルを作れることを意味する。たとえば、図5(b)のグラフの場合、エッジ数が7、奇ノード数が4である。このグラフの道分割の例として $R_1 = \{aAbBcGeEbFf, cCdDe\}$ や $R_2 = \{aAbFf, dDeEbBcCd, cGe\}$ などがあり、 $2 \leq |R_1| \leq |R_2| \leq 7$ を満たす。このグラフを構造とするバルーンモデルは、2本以上7本以下のバルーンを使って作ることができる。2本使う場合、1本のバルーンから道 $aAbBcGeEbFf$ の各ノード、各エッジに対応するひねり目、バブルを作り、同様に、もう1本のバルーンから道 $cCdDe$ に対応するひねり目、バブルを作り、ロックの操作を使って作り上げることができる。

2.4 形状維持可能かどうかの判定

ツイストで作られるひねり目は、なんらかの方法で固定しなければ元に戻る。例えば、図6のdがひねり目の場合、ひねりが戻ってしまう。一方、図6の目b, c, eはロックされているので固定され、ひねりを維持できる。また、ロックされていないひねり目でも固定されるものがある。複数のバブルが環状につながり、その中にひねり目が含まれる場合である。図6ではB, H, IおよびE, F, Gは上記を満たしており、ひねり目h, g, fは固定される。すべてのひねり目が固定されたバルーンモデルを「形状維持可能なバルーンモデル」とよぶ。

これをグラフを用いて再考する。次数が3以上の目はロックされてきた目であるので、その目に含まれるひねり目は固定される。次数が2のノードは、1つのひねり目、あるいは2つの結び目をさらに結んだ目に対応する。前者の場合、そのひねり目が固定される条件は閉路上にあることである。図6のグラフでは閉路**BcHhIb**, **eFfgGe**があるので、ノード**h, g**がひねり目であったとしても、これらのひねり目は固定される。以上より、次数が2のノードが全て閉路上にあるか、次数が2のノードを持たないグラフを「形状維持可能なグラフ」とよぶ。

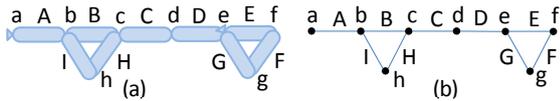


図6. 形状維持可能ではないバブルンモデル・グラフの例.

これを満たさない場合には、結び目同士を結ぶロックの必要が生じ、そこでグラフが分割される、つまり、制作に複数のバブルンが必要になる。なお、実際には図3のスラストによりひねり目を固定する方法もあるが、このような固定の判定にはグラフ構造に加え3次元的配置を考慮する必要がある。したがって、形状維持可能なグラフはこのグラフを構造とするバブルンモデルが形状維持可能であるための十分条件の1つである。

本論文では複数色のバブルンを使ったモデルを扱うため、色の制約と形状維持可能判定との関係について述べる。次数2のノードに接続する2つのエッジに対応するバブルンの色が異なる場合を考える。これは色の異なる2本のバブルンの端同士が繋がった状態に対応し、このような場合は、端同士を結んで接続するしかない。すなわち、この次数2のノードはひねり目ではなく結び目に対応する。したがって、形状維持可能判定は次数2のノードで同色のエッジが接続する場合のみを考えればよい。

2.5 操作ロックが実施可能かどうかに関する考察

図7(a)のひねり目cとgはロックでき、図7(b)が得られる。一方、図7(a)のひねり目cとiはロックできない。なぜなら、バブルンB, C, DとH, I, Jがともに環状で、cとiを絡めることができないからである。このように、2つの目のロックが実施できるときと、そうでないときがある。ロックの実施が可能か不可能かを判定する方法は現段階では不明であるため、ここではロックが実施可能であるための十分条件のみ述べることにする。

図7(d)に示すようなグラフの操作を考える。これは次数が2以上である1つのノードを2つにして、元のノードに接続するk本のエッジを k_1 本と k_2 本($k_1 + k_2 = k, k_1 \geq 1, k_2 \geq 1$)に分け、それぞれ分離した2つのノードに接続する操作である。 $k \geq 3$ の場合は複数通りのエッジの分け方がある。このような操作を行うと元のグラフが分離される(連結成分数が増加する)場合と分離されない場合がある。ここでは、この操作を施してグラフが分離されるエッジの分け方が存在するノードを「分離可能ノード」と呼ぶことにする。図7(c)に図7(a)のバブルンモデルのグラフを示し、ノードc, g, iに上記の操作を施した結果を図7(e)に示す。ノードc, g, iは次数2なのでエッジの分け方は一通りしかない。操作の結果、gは分離可能ノードであり、cとiは分離可能ノードではないことがわかる。ロックしようとする2つの目のうちどちらか1つでも分離可能ノードに対応するならば、ロックを実施できる。一方、両方の目が分離可能ノードではないとき、先述の例のようにロックを実施できない場合がある。

3. グラフからの制作手順の生成

バブルン本数が最少となる、すなわち、与えられたグラフGの奇ノード数を $2n$ とするとき、 n 本のバブルンで構成されるバブルンモデルを制作する手順を求める。求める手順はGの各ノード、エッジに対応する目とバブルンを生成するツイスト操作、および、生成した目に対するロック操作の順序列により表現する。

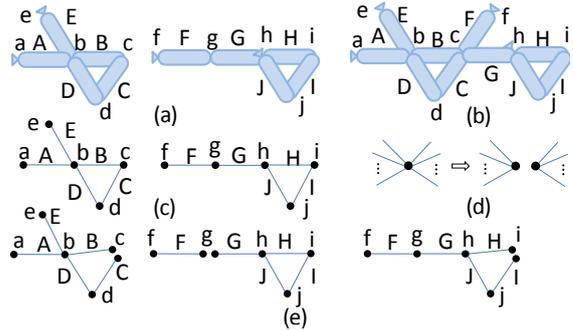


図7. ロック実施の可否と分離ノードの例.

また、ツイストの方向性とロックの実施可能な場合について上述したが、生成する手順はこれを遵守する。

グラフGのノードやエッジの数が大きい場合、可能な手順を全探索するには膨大な時間を要する。そこで、まず制作に必要なバブルンの本数 n とGの構造を n 本のバブルンで構成する道分割を求め、これに作り易さに基づく制限を加えて探索範囲を限定する。ここで求めたもののうち難易度の最も低いものを制作手順とすることで、実時間で処理を実現する。

3.1 道分割の生成

グラフGの奇ノードの個数を $2n$ とするとき、Gの道分割Rの大きさ $|R|$ の最小値は n である。 $|R| = n$ となる道分割Rを全て求める方法を述べる。まず、全ての奇ノードを n 個の組に分ける。各組の奇ノードを両端に持つ仮想のエッジを追加すると、グラフのノードは全て偶ノードとなる。図8(a)のグラフでは奇ノードが4個あり、2組への分け方は $OC_1 = \{(a, e), (d, g)\}$, $OC_2 = \{(a, d), (e, g)\}$, $OC_3 = \{(a, g), (d, e)\}$ の3通りがある。一般には、 $({}_{2n}C_2 \cdot {}_{2n-2}C_2 \cdot \dots \cdot {}_2C_2) / n!$ 通りの分け方がある。図8(b)(c)(d)は OC_1, OC_2, OC_3 の分け方に従って、各組の奇ノードを両端にもつ仮想のエッジH, Iを追加したものである。ノードは全て偶ノードとなるので、この場合はオイラー道が存在する。

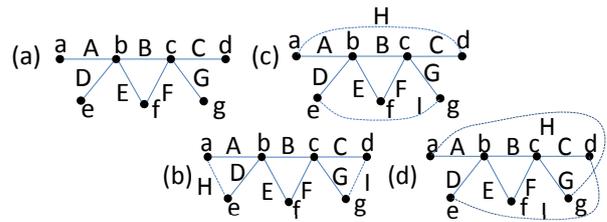


図8. 道分割の生成の例.

図8(b)の場合、4つのオイラー道 er_1, er_2, er_3, er_4 がある(図9)。各オイラー道から仮想のエッジH, Iを取り除くと、グラフGの道分割が得られる。オイラー道 er_1, er_2, er_3, er_4 からは、それぞれ道分割 R_1, R_2, R_3, R_4 が得られる。

$$er_1 = aAbBcCdIgGcFfEbDeHa \quad R_1 = \{aAbBcCd, gGcFfEbDe\}$$

$$er_2 = aAbBcGgIdCcFfEbDeHa \quad R_2 = \{aAbBcGg, dCcFfEbDe\}$$

$$er_3 = aAbEfffCcDlGcBbDeHa \quad R_3 = \{aAbEfffCcD, gGcBbDe\}$$

$$er_4 = aAbEfffCcGgIdCcBbDeHa \quad R_4 = \{aAbEfffCcGg, dCcBbDe\}$$

OC_2, OC_3 についても同様に行うと、それぞれ6個の道分割が得られる。3通りの奇ノードの分け方で合計16個の道分割が得られるが、その中でユニークな道分割は8個である。ここではグラフGから上記の手順で得られる道分割の全体を Φ_G で表す。一般に、 Φ_G の大きさは与えられたグラフに依存する。

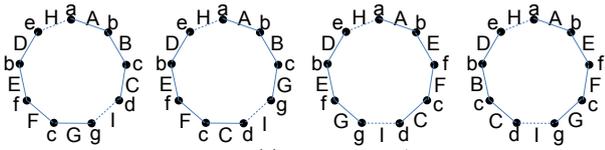


図9. 図8(b)のオイラー路.

3.2 制作手順候補の生成

上記で得られた ϕ_G の各道分割から、ツイストとロックを用いたバルーンモデルの制作手順を生成する。

制作手順の生成においては、道分割、つまりバルーンの本数とバブルの配置に基づいたすべての手順を得られることが望ましい。一方で、全ての制作手順を調べるのは、組み合わせの数が膨大になることから探索時間が指数関数的に増加し、実時間での導出が困難となる。そこで、今回は先述のバブル作成の方向性に従い、実際に制作可能な手順の候補を生成し、その中から、固定される目の数に関連した難易度の小さいものを提示するような方法を検討する。この際、道分割において奇ノードが0のものが含まれる場合、すべてのノードが始点・終点となり得るが、探索時間を考慮して任意の1ノードのみを始点・終点とする。また、制作中のある局面で、つぎにどのバルーンを操作するのが最適かを求めることも、組合せ数の問題から導出は困難である。そこで、後述のようにバルーンに選択の優先順位を設定し、高順位のパルーンのバブルの生成が行いやすくなるよう、順位が下のものへのロックを行うことを繰り返していくというアプローチを採用する。

上記に基づいて、図8(a)のグラフ G の1つの道分割である

$$R_1 = \{aAbBcCd, gGcFfEbDe\}$$

を例として制作手順の生成方法を説明する。 R_1 には2本の道があり、それぞれを1本のバルーンに見立てるが、先述のツイストの方向性を考慮して道に向きを与える。道 $r_1 = aAbBcCd$ の場合、 a から d への向きを与えた道を $\vec{r}_1 = aAbBcCd$ と表し、その逆向きは $\overleftarrow{r}_1 = dCcBbAa$ のように表す。道 $r_2 = gGcFfEbDe$ に対しても同様に向きを与える。さらに、後述の制作手順において制作に取り掛かる順、つまり各道における制作の優先順位を示す、向き付きの道を要素とするリスト(有効道の列)を作る。

向きと優先順位の与え方の組み合わせは8通りあり、優先順位の高い道を先に書いてリスト形式で表すと $[\vec{r}_1, \vec{r}_2], [\vec{r}_1, \overleftarrow{r}_2], [\overleftarrow{r}_1, \vec{r}_2], [\overleftarrow{r}_1, \overleftarrow{r}_2]$ となる。このように道分割 R の各道に向きと優先順位を与えて得られるリストを「道リスト」とよび、道分割と区別し、 $D = [d_1, d_2, d_3, \dots, d_n]$ 、 $d_i = v_1^i e_1^i v_2^i e_2^i \dots v_{i-1}^i e_{i-1}^i v_i^i$ のように表す。ここで l_i は d_i のノード数である。一般に大きさ n の道分割から $n! \times 2^n$ 個の道リストが得られる。

つぎに、図10のアルゴリズム *MakeOperationList()* を実行し、制作手順のリスト P を生成する。このアルゴリズムでは優先順位の高い道に対応するバルーンから作っていき、ツイストして生成されるひねり目がそのバルーン自身の目あるいは次の優先順位にあるバルーンの目とロックできるかどうかを常にチェックし、ひねり目をロックしていく。具体的には優先順位の高い道 d_i から、そのエッジ e_{i-1}^i とノード v_j^i に対応するバブルと目を作っていく、道 d_i に対応するバルーンを変形していく。

このとき、ノード v_j^i に対応する目がすでにあれば、ロックをする。また、道 d_{i+1} にノード v_j^i と同じノード v_k^{i+1} があれば、道 d_{i+1} を同様に v_k^{i+1} まで作っていき、 v_k^{i+1} をロックする。

道リストを $D = [\vec{r}_1, \vec{r}_2] = [aAbBcCd, eDbEffcGg]$ とした場合、図10を実行すると次の制作手順のリスト P が得られる。

$$P = [t(A), t(D), l(b), t(B), t(E), t(F), l(c), t(C), t(G)]$$

ここで、 $t(A)$ は*twist(A)*でツイストによりエッジ A を作ること、 $l(b)$ は*lock(b)*でノード b をロックすることを表し、紙面の都合上、以下の数式においても同様とする。

/ グローバル変数の説明 */*

$D = [d_1, d_2, d_3, \dots, d_n]$: 3.2節の方法で求めた道リスト

l_1, l_2, \dots, l_n : l_i は d_i のノード数

P : 求めたい制作手順のリスト

V : ノードの集合

E : エッジの集合

s_1, s_2, \dots, s_n : s_i は生成していない d_i のノード番号の最小値

/ DからPを求める */*

MakeOperationList()

/ グローバル変数の初期化 */*

$P := []$; $V := \{\}$; $E := \{\}$; $s_1 := s_2 := \dots := s_n := 1$;

for $i := 1$ **to** n **do**

MakeOperationList(i, l_i);

end

/ d_i の s_i 番目のノードからright番目のノードまでに対応するバブルと目を作る手順を生成する */*

MakeOperationList(i, right)

while $s_i \leq right$ **do**

$j := s_i$; $s_i := s_i + 1$;

if $j > 1$ **then**

$P.Append(twist(e_{j-1}^i))$; */* 後方から追加 */*

$E := E \cup \{e_{j-1}^i\}$;

end

if $v_j^i \in V$ **then**

$P.Append(lock(v_j^i))$;

else

$V := V \cup \{v_j^i\}$;

if $i < n$ **then**

$k := s_{i+1}$;

while $v_j^i \neq v_k^{i+1}$ or $k \leq l_{i+1}$ **do**

$k := k + 1$;

end

if $k \leq l_{i+1}$ **then**

MakeOperationList(i + 1, k);

end

end

end

end

図10. 制作手順生成アルゴリズム.

表1はアルゴリズムの実行過程において V または E が変化するときの V , E の値および P に追加される操作を示す。また、図11(1)の(a)~(i)は、表1(a)~(i)の過程における各バルーンの状態を表す。

表1. アルゴリズムの実行過程.

	V	E	P に追加される操作
(a)	a		
(b)	a,b	A	$twist(A)$
(c)	a,b,e		
(d)	a,b,e	A,D	$twist(D)$, $lock(b)$
(e)	a,b,e,c	A,D,B	$twist(B)$
(f)	a,b,e,c,f	A,D,B,E	$twist(E)$
(g)	a,b,e,c,f	A,D,B,E,F	$twist(F)$, $lock(c)$
(h)	a,b,e,c,f,d	A,D,B,E,F,C	$twist(C)$
(i)	a,b,e,c,f,d,g	A,D,B,E,F,C,G	$twist(G)$

実線はバブル、鎖線は完成に至る過程においてバブルとしてまだ使われていないバルーンの残りの部分を表す。記号 \odot , \ominus , \oplus はそれぞれ結び目、バルーンの先端、固定されていないひねり目、固定されているひねり目を表す。なお、先述のように、ひねり目が固定されているかどうかは、ロックされているか、あるいは制作過程のバルーンモデルをグラフとしたとき閉路上にあるかを調べれば判定できる。制作過程のバルーンモデルのグラフは、 V と E に先端 δ とバルーンの残りの部分 Δ に対応するノードとエッジをそれぞれ追加すれば得られる。

図12(a),(c),(d),(e),(f)は図11(1)の(a),(c),(d),(e),(i)に対応するバルーンモデルである。このアルゴリズムで生成される制作手順はバブルを左から順につくるため、ツイストの一方方向性を満たしている。また、ロックしようとする2つの目のうち1つは分離可能ノードに対応するので、ロックは常に実行可能である。

別の例として、奇点の組を $OC_2 = \{(a, d), (e, g)\}$ として得られる道リストの1つである $D = [aAbBcFfEbDe, dCcGg]$ を入力として図10のアルゴリズムを実行すると、次の制作手順のリスト P が得られる。

$$P = [t(A), t(B), t(C), l(c), t(F), t(E), l(b), t(D), t(G)]$$

図11(2)には制作過程におけるバルーンの状態を示す。

以上の方法では、与えられたグラフにより得られる道リストから1つの制作手順を生成する。どの道リストであっても、生成した制作手順に従って作ったバルーンモデルの構造は全て同じである。また、制作手順に含まれるツイスト、ロックの回数も同じであり、その順序のみが異なる。

図13は3本のバルーンが必要なケースであり、(a)のグラフに対して同様に手続きを行うと1248個の制作手順が得られる。以下の手順はその1つである。

$$P = [t(L), t(H), t(I), l(h), t(J), t(K), l(b), t(A), t(B), t(C), t(G), t(F), l(d), t(D), t(E)]$$

図13(b)には、この手順で制作したときの各バルーンの状態を示す。

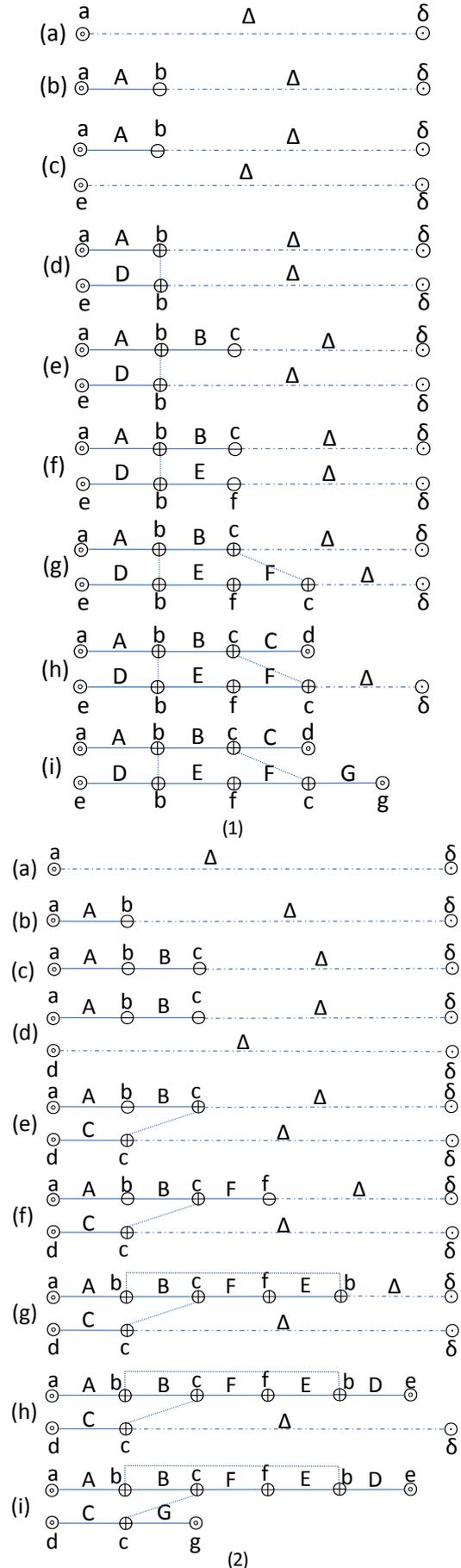


図11. 制作過程におけるバルーンの状態.

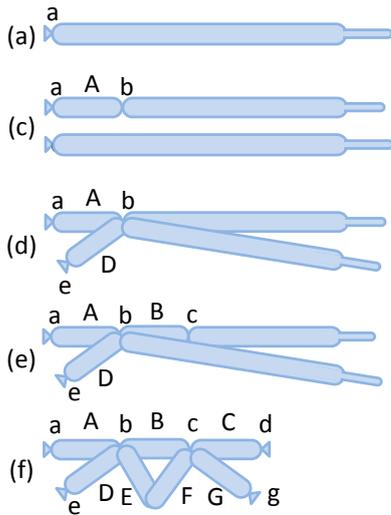


図12. 図11(1)に対応するバルーンモデル.

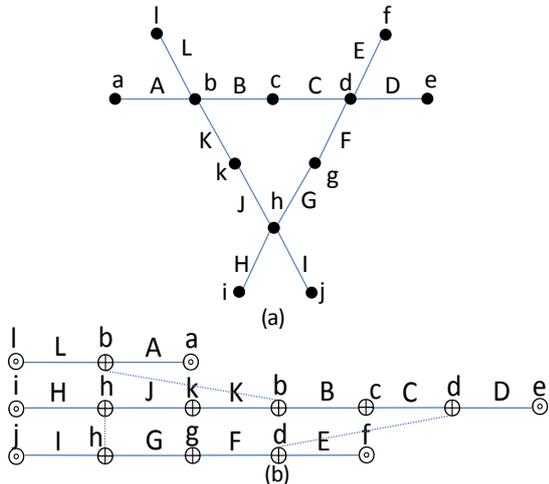


図13. 3本のバルーンの例.

3.3 制作手順の難易度評価

図11において、記号⊕は固定されていないひねり目を表す。制作の過程でこのようなひねり目がある場合、そのひねりが戻らないよう、非固定のひねり目が含まれる連続するバブルの両端をロックされるまで片手で押さえながら、次の操作を行わなければならない。この際、押さえが必要なのは2箇所のみであるが、バブルを作るたびに押えを移動しなくてはならず、押えが外れてしまうと作り直しとなる。したがって、制作過程で出現する非固定のひねり目の数が、難易度の一評価基準となると考えられる。ここでは、制作過程における非固定のひねり目の数の最大値および総和に着目する。

上述のように、非固定のひねり目はバブルの作成の度に手で押さえる箇所を変更する必要が生じる。そのため、バブルの作成を単位として、最大値および総和を算出する。具体的には、ツイストを1回行って新たにできた目がロックできる場合はロックした後、ロックできない場合はそのツイストの後にモデル全体で非固定の目の数 x を数え、完成に至るまでの x の最大値と総和を求める。

図11 (1), (2)の最大値はともに2であり、総和は4, 6である。また、図13(a)のグラフについて、先に示した制作手順の例では最大値が4、総和が20となる。最大値および総和は制作手順により様々な値をとるため、これらが小さい手順の方が作りやすいと考え、最大値を優先に難易度として設定する。

4. バルーンアートデザイン支援システム

前章の手法を実装した、バルーンアートデザイン支援システムを制作する。システムではユーザが作成したグラフを入力とし、形状維持可能なグラフかどうかを判定して、可能な場合はその制作手順の候補を出力する。

4.1 複数色のバルーンを使う場合への対応

2色のバルーンを使った単純な例を図14(b)に示す。これは図14(a)と同じ構造をもち、そのグラフは図14(c)となる。グラフは奇ノードが4個なので、図14(a)は2本のバルーンで作れるが、図14(b)の場合はバルーンの色を考慮すると3本のバルーンが必要である。システムは次のようにして複数色のバルーンを使う場合に対応する。

1. ユーザは入力したグラフ G のエッジに色 $1, 2, \dots, k$ から選んだ1つの色をつける (k の最大値: バブルの総数)
2. $G_i = (V_i, E_i)$ を色 i のエッジとその両端のノードからなるグラフとする
3. グラフ G_i から生成した m_i 個の道分割の集合 $\{R_1^i, R_2^i, \dots, R_{m_i}^i\}$ を Φ_{G_i} とする
4. $\Phi_G = \{R_{p_1}^1 \cup R_{p_2}^2 \cup \dots \cup R_{p_k}^k \mid R_{p_i}^i \in \Phi_{G_i}\}$ をグラフ G の道分割の集合とする
5. 集合 Φ_G のすべてについて道リストを生成し、制作手順を求める

図14(c)のグラフ G については、

$$G_1 = (\{a, b, c, d, f\}, \{A, B, C, D, E\}),$$

$$G_2 = (\{e, c, f, g, h, b, i\}, \{F, G, H, I, J, K\})$$

となり、それぞれのグラフの道分割の集合は次のようになる。

$$\Phi_{G_1} = \{\{aAbBcCdDfEa\}\}, \Phi_{G_2} = \left\{ \begin{array}{l} \{eFcGfHg, hIfjBKi\}, \\ \{eFcGfHh, gHfjBKi\}, \\ \{eFcGfjBKi, gHfHh\} \end{array} \right\}$$

なお、 G_1 は環状であるため、 Φ_{G_1} は本来すべてのノードが始点・終点となる6個の道が得られるが、先述のように任意のノード(ここではa)を始点終点とする。以上より、 Φ_{G_1} と Φ_{G_2} から道分割を1つずつ選んで得られる G の道分割の集合は以下となる。

$$\Phi_G = \left\{ \begin{array}{l} \{aAbBcCdDfEa, eFcGfHg, hIfjBKi\}, \\ \{aAbBcCdDfEa, eFcGfHh, gHfjBKi\}, \\ \{aAbBcCdDfEa, eFcGfjBKi, gHfHh\} \end{array} \right\}$$

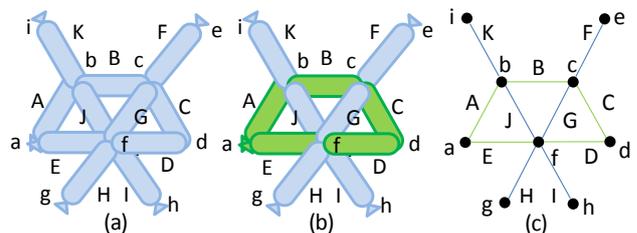


図14. 1色および2色のバルーンを使ったモデルの例.

4.2 制作手順の生成と提示

グラフのオイラー道は一般にグラフ探索により求めるが、オイラー道の本数はノードの個数や各ノードの次数が大きくなると指数的に増え、現実的な時間内で全てのオイラー道を探査することは困難となる。システムはこの問題を回避するため、全オイラー道探索とランダム探索の2つから1つをユーザに選択させ、後者の場合、ランダム探索により抽出するオイラー道の本数をユーザに入力させる。また、得られた制作手順の候補は難易度のやさしい順にソートする。ここでは、固定されているひねり目の数の最大値が小さいほど、また、これが等しい場合は総和が小さいほど順位を高くする。システム全体の処理の流れは次のようになる。

1. ユーザはグラフGを入力しエッジに配色
2. Gが形状維持可能か判定し、可能ならば3以降へ
3. $G_i = (V_i, E_i)$ を色iのエッジとその両端のノードからなるグラフとする
4. ユーザが全オイラー道探索を選択した場合は5へ、ランダム探索を選択した場合は6へ
5. G_i について全オイラー道を求め、道分割の集合 Φ_{G_i} を生成
6. G_i について、ユーザがオイラー道の本数 $cmax_i$ を入力し、 $cmax_i$ 回ランダム探索を行って $cmax_i$ 本のオイラー道を求め、道分割の集合 Φ_{G_i} を生成
7. 上記5または6で得られた Φ_{G_i} からGの道分割の集合 Φ_G を求める
8. 集合 Φ_G のすべてについて道リストを生成し、制作手順を求める
9. 得られた手順を難易度の低い順に並べ替えユーザに提示

4.3 実行例

制作したシステムを利用してバルーンアートをデザインした例を示す。なお、以下の3種類のバルーンアートは、いずれも著者のシステム操作による制作である。

まず、複数本バルーンに対応した構造化と制作手順の導出が可能であるか、図15(a)の1色のバルーンを使ったバルーンモデルを用いて確認する。図15(b)のグラフについて、 $cmax_1 = 100$ としてランダム探索によりオイラー道を求めた結果696個の道リストが得られた。その中で最も難易度が低かったものの1つを示す。

$$D = [fFcDdCbKhMkNhIjjiPlOihGgH, iLcBaAbEe]$$

これは2本のバルーンに対応し、この道リストから得られる制作手順は次のようになる。

$$P = [t(F), t(L), l(c), t(D), t(C), t(B), t(A), l(b), t(K), t(M), t(N), l(h), t(I), t(J), l(i), t(P), t(O), l(i), t(H), t(G), l(h), t(E)]$$

この手順において、固定されないひねり目の最大値は3、総和は14である。図15(c)に当該手順で制作したときの各バルーンの状態を、(d)には実際に制作したバルーンアートを示す。

図が示すように、提案手法およびシステムにより複数本バルーンによるバルーンアートの制作手順が導出でき、その手順に基づいて実際のバルーンを用いた制作が可能であることを確認した。

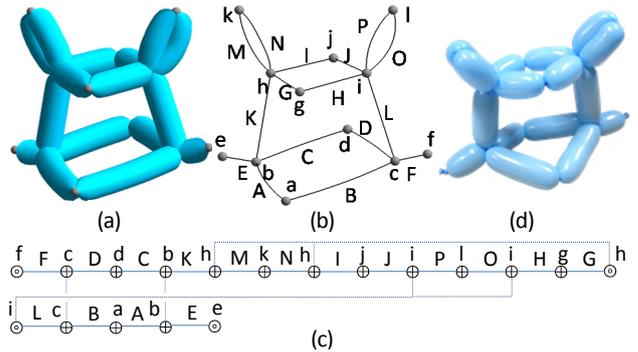


図15. 単色のバルーンアートの制作手順の生成例

つぎに、複数色のバルーンを用いた事例について、図16(d)の2色のバルーンを使ったバルーンモデルで確認する。また、システムによる制作手順も示す。(a)から(c)はこのモデルを入力する過程である。まず、(a)のようにノードを空間に配置し、その間にエッジを挿入し、ベジェ曲線の制御点と接続点を移動してエッジの形を定める。これを繰り返すことで(b)のような完成形状のグラフを作成し、(c)のようにエッジをバブルに変換した後で配色を行う。(e)に示すノードやエッジのラベルは入力した順に基づいて与える。 $cmax_1 = cmax_2 = 10$ としてランダム探索によりオイラー道を求めた結果、400個の道リストが得られた。その中で、最も難易度が低かったものの1つを次に示す。

$$D = [aEbFaBcAaGeHaCdDa, fHaIgmJNgKiLgJh]$$

これは2本のバルーンに対応し、この道リストから得られる制作手順は次のようになる。

$$P = [t(H), l(a), t(E), t(F), l(a), t(B), t(A), l(a), t(G), t(H), l(a), t(C), t(D), l(a), t(I), t(M), t(N), l(g), t(K), t(L), l(g), t(J)]$$

この手順において、固定されないひねり目の最大値は2、総和は8である。図16(f)に当該手順で制作したときの各バルーンの状態を、(g)には実際に制作したバルーンアートを示す。

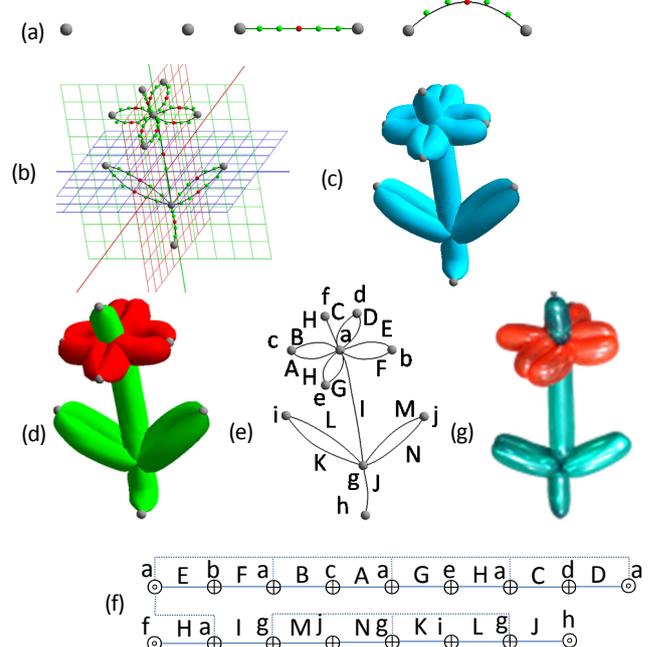


図16. 2色のバルーンアートの制作手順の生成例。

図17は4色を使った場合の、複数色の別例である。全道リスト384個のうち最も難易度が低かったものの1つを次に示す。

$D = [LgSnPmMbAa, bBcDeFgGfEdCb, hHcNmOdIi, kKfRnQeJj]$
 また、この道リストから得られる制作手順は次のようになる。

$P = [t(L), t(B), t(H), l(c), t(D), t(F), l(g), t(S), t(P), t(M),$
 $l(b), t(A), t(G), t(E), t(N), l(m), t(O), l(d), t(C), l(b), t(I), t(K),$
 $l(f), t(R), l(n), t(Q), l(e), t(J)]$

この手順での非固定のひねり目の最大値は3、総和は20である。

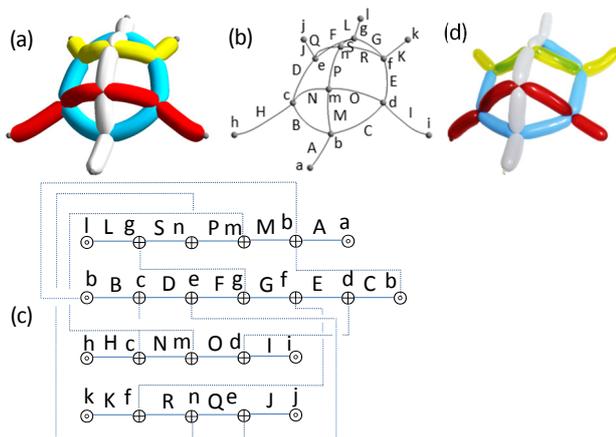


図17. 4色のバルーンアートの制作手順の生成例。

図16, 17で示すように、複数色のバルーンアートについても、実際にバルーンを用いて制作可能である構造化、および、制作手順の導出を行えることが確認できた。

4.4 考察

本論文では、グラフの道分割や手順の組み合わせという、全探索が難しい対象を扱っている。そのため、制作の方向性など作り易さの観点に基づく制限を加えることで探索範囲を限定し、実時間で処理を実現した。したがって、作り易さに基づく一定の解が得られていると考えられるが、理論上では存在する制作手順を網羅できておらず、探索範囲外により最適な解が存在する可能性は排除できない。また、図17の亀の形状を1色のものとしてアルゴリズムを適用した場合、 $aAbBcHh$ というような足と尾と甲羅の一部に1本のバルーンを割り当てるような出力が得られる。このような場合、制作の際にどの部分を制作しているのかがわかりづらいといった、制作技法とは異なる問題も発生する。そのため、操作とは別の観点での作り易さの指標についても検討が必要である。

提案システムについては、1章で述べたように複数本・色のバルーンによるバルーンアートのコンピュータによる制作支援を前提に、構造や制作手順の出力を主目的としている。そのため、前節で示したように主目的は達成されたと考える一方、今回はあらかじめ完成形状がイメージできているケースを想定しており、バルーンアートやコンピュータ上での造形に不慣れな者にとっては利用し易いインタフェースとなっていない。バルーンアートでは、たとえば動物の耳など、ある形状を制作するバブルの組み合わせがおおよそパターン化されているものもある。そのため、現状のものを使いやすさを改善させるだけでなく、そういった要素を取り入れた支援についても検討が必要であると考えられる。

5. おわりに

本論文では、複数本で構成されるバルーンモデルのグラフ構造に関する性質、制作手順の生成手法、および、これらに基づいた配色も考慮したデザイン支援システムについて論じた。

今回の提案手法では、導出される制作手順が理論上の全手順と比較して、どの程度をカバーしているかは明らかとなっていない。また、今回は考慮しなかったバルーンの長さや個々のバブルの配置などを考慮することで、形状維持可能性をより厳密に判定でき、スラストを含めた詳細な制作手順の提示が可能となる。これらの解決は実用性の高い支援につながると考えられるため、考察の内容も含めて今後の課題としたい。

謝辞

本研究の一部はJSPS 科研費 25280131, 15K00448 による。

参考文献

- [1] 内田忠, 伊藤英則, 折り紙過程の知識表現とその処理プログラムの作成, 情報処理学会論文誌, Vol.32, No.12, pp.1566-1573, Jul.1991.
- [2] Miyazaki S, Yasuda T, Yokoi S, and Toriwaki J, An Interactive Simulation System of ORIGAMI Based on Virtual Space Manipulation, Proc. of ROMAN '92, pp.210-215, Sep.1992.
- [3] Tomohiro Tachi., 3D Origami Design Based on Tucking Molecule, In Origami4: Fourth International Meeting of Origami Science, Mathematics, and Education, pp. 59-272, 2009.
- [4] 寺崎美保子, 今すぐできる! バルーンアート ひねってつくる魔法の風船, 新星出版社, Apr.2008.
- [5] 高田佳子, ファンタジー・バルーン, 大月書店, Apr.1999.
- [6] 大石亜由美, 壁画・イベントに生かす 子どもと楽しむバルーンアート, いかだ社, Mar.2008.
- [7] はいけんた, 楽しくつくろう! バルーン工作, プティック社, Jul. 2008.
- [8] 畑中啓行, 生田大介, 田中敏光, 杉江昇, 仮想バルーンをモデリングするエデュテイメントソフトの提案, 2000年度電気関係学会東海支部連合大会論文集, p.269, Sep.2000.
- [9] 小坂友里, 風船の物理特性を考慮したバルーンモデリングの制作支援に関する研究, 東京工科大学2008年度卒業論文, 2009.
- [10] Erik D. Demaine, Martin L. Demaine, Vi Hart, Computational Balloon Twisting: The Theory of Balloon Polyhedra, in Proc. of CCCG 2008, Aug. 2008.
- [11] 浦正広, 山田雅之, 遠藤守, 宮崎慎也, 安田孝美, バルーンアートの構造解析と難易度評価手法の提案, 芸術科学会論文誌, Vol.8, No.4, pp.143-150, Dec.2009.
- [12] 浦正広, 山田雅之, 遠藤守, 宮崎慎也, 安田孝美, バルーンアートの構造解析と作成支援ツールの開発, 第8回 NICOGRAPH 春季大会論文集, Mar.2009.
- [13] 浦正広, 山田雅之, 遠藤守, 宮崎慎也, 安田孝美, 複数本バルーンにより構成されるバルーンアートの構造解析, 電子情報通信学会技術研究報告, Vol.112, No.473, pp.295-300, Mar.2013.

浦 正広



2012年名古屋大学大学院情報科学研究科博士課程修了。同年MAI Lab.開業，14年北陸先端科学技術大学院大学知識科学研究科助教，現在に至る。アート，エンタテインメントと科学との融合に関する研究に従事。博士（情報科学）。

山田 雅之



1994年名古屋工業大学大学院工学研究科修士課程修了。同年同大学知能情報システム学科助手，98年中京大学情報科学部助手，99年同講師，07年同大学情報理工学部准教授，12年同教授，13年同大学工学部教授，現在に至る。人工知能とメディア技術を融合利用した応用研究に従事。博士（工学）。

宮田 一乗



1986年東京工業大学大学院総合理工学研究科修士課程修了。同年，日本アイビーエム（株）東京基礎研究所入社，98年東京工芸大学学術部助教，02年陸先端科学技術大学院大学知識科学教育研究センター教授，12年同大学知識科学研究科教授，現在に至る。コンピューターグラフィックスおよびメディア表現に関する研究に従事。博士（工学）。

遠藤 守



2003年名古屋大学大学院人間情報学研究科博士課程修了。同年中京大学情報科学部講師，08年同大学情報理工学部准教授，14年名古屋大学大学院情報科学研究科准教授，現在に至る。仮想空間とネットワークの融合利用に関する研究，およびオープンソースの社会利用に関する研究に従事。博士（学術）。

宮崎 慎也



1994年名古屋大学大学院工学研究科博士課程修了。93年中京大学情報科学部助手，97年同講師，99年同助教，07年中京大学情報理工学部教授，13年同大学工学部教授，現在に至る。CG，VRの特にソフトウェアによる仮想環境構築，空間操作に関する研究に従事。博士（工学）。

安田 孝美



1987年名古屋大学大学院工学研究科博士課程修了。同年同大学助手，93年同大学情報文化学部助教，03年同大学大学院情報科学研究科教授，現在に至る。ネットワークを利用したマルチメディアにCG，VRの新たな可能性に関する研究に従事。博士（工学）。