2005－CG　118　(3)
2005／2／7

集約法による組み立て方を把握しやすい多面体の展開図生成手法

三谷　純[†]　鈴木宏正[††]

工作しやすい展開図を生成することを目的とした，多面体の新しい展開図生成手法を提案する．提案する手法では，はじめに全ての面を個別に同一平面上に配置し，これらの面を 3 次元空間での面の接続情報を元に徐々に集約してゆくことで展開図を構築する．本稿ではこのような展開手法を「集約法」，従来の手法を「逐次法」と呼ぶこととし，両者の手法によって生成される展開図のコストの比較を行う．また新しく提案する集約法を応用することで，面の向きの維持，左右対称性の維持などの特徴を持つ，組立て方を把握しやすい展開図の生成を行えることを示す．

A face-gathering approach for generating intuitive unfolded pattern of polyhedrons

Jun Mitani[†] and Hiromasa Suzuki[††]

We propose a new method for unfolding polyhedrons using a face-gathering approach targeting at generating patterns that are easy to be assembled. With our method, we firstly place all polygonal faces on an unfolded plane, then gather them by connecting faces regarding the connectivity in 3D. We call this method 'face-gathering approach' and the established method 'sequential approach'. We compare both methods in view of the ability of generating low-cost patterns. And we also denote that our approach is applicable for generating reasonable patterns such that are symmetrical and directions of faces are preserved.

1. はじめに

計算機の内部に保持された立体データを実世界でモノとして知覚できる形にすることは，工学的側面や教育的な観点からも有用である．立体の展開図を紙に出力して組み上げるペーパークラフトは，これを手軽に実現するための手段として有効であり，教育の一環で使用されたり[1]，またホビーの一つとして多くの人々に親しまれている[2)3)]．

ところで，CG の世界では立体形状の表現手法として，そのデータ構造の単純さと扱いの容易さから，多面体モデル（ポリゴンモデル）が用いられることが多い．多面体は平面多角形の集合であるため，単純なアルゴリズムで平面へ展開することができる．ただし，展開する際にどの稜線を切断し，どの稜線を接続したまま残すかを決定する必要があり，この決定方法に拠って展開図の形状は異なったものになる．

本稿では,「いかにわかりやすい展開図を作成するか」という観点から，多面体からペーパークラフトに適した展開図を生成するための新しい手法を提案する．

† 独立行政法人理化学研究所
The Institute of Physical and Chemical Research
†† 東京大学東京大学先端科学技術研究センター
Fine Digital Engineering, RCAST, The Univ. of Tokyo

2. 関連研究

3 次元空間上の曲面および多面体を平面に展開する手法は古くから研究されている．平面に展開できる曲面は可展面と呼ばれるものに限られ[4]，それ以外の曲面を展開するには，それらを可展面の集合で近似する必要がある．B-Spline 曲面や Bezie 曲面などのパラメトリック曲面を円錐曲面や円柱曲面の一部の集合で近似する手法は多く研究されている[5)6)7)]．この手法は船体などの複雑な凹凸をもたない形状へ適用できることが示されているものの，動物や車などの複雑な形状特徴を持つ自由形状を対象としたペーパークラフトへ応用することは難しい．この問題を解決するために，近年ではメッシュで表現された自由形状を Strip の集合で近似して展開図を作成する手法も提案されている[8)]．

一方，多面体を展開することは容易であるため，教育用ソフトや，市販ソフトにも，多面体を平面に展開する機能を組み込んだものがあり[9)10)]，実際に CG ソフトで作成された多面体からペーパークラフト用の展開図を生成することが行われている．

ところで，一つの多面体にも，その展開図は何通りも存在する．そのため，面の数が多い多面体を工作することを考える場合には，工作に要するコストの小さい展開図を作ることが重要となる．文献 11) では，切断する辺の長さやパーツ数を工作のコストの指標に用い，多面体に含まれる稜線の長さや角度を考慮することで，工作コストの小さい

展開図を生成できることが示されている．文献 12) では，工作用紙に展開図を配置した際の余白の多寡で展開図の評価を行い，その評価値を高めるための手法が提案されている．文献 13) では，人が一定距離の直線をカットするのに要する時間，一定数の折れ線を折り曲げるのに要する時間などを実際に計測することで展開図のコストを評価する方法が提案されている．

展開図の評価手法には様々なものが考えられるが，これらの評価手法は全て，展開図の幾何的な値を用いて評価値を算出している．しかし，これらの方法では，数値評価が困難な，人が感性に基づいてよいと判断する展開図を生成することはできない．一方，市販ソフト 10) では，最初にシステムが作成した展開図を，後からユーザーが好みに応じて自由に編集することで，最終的にユーザーがよいと判断する展開図を手作業で作成できるようになっている．しかし，面の数が多くなると展開図の編集作業が大変になるため，この負荷を軽減するためにも，ある程度人がよいと判断する展開図をシステムが自動的に生成できることが望ましい．

そこで本稿では，面の向きや左右の対称性などを考慮することで，人がよいと判断するものに近い展開図を生成するための手法を提案する．

3. 多面体の展開手法

多面体の展開図は，はじめに面の 1 つを平面上に配置し，それに位相的に接続する面を逐次展開してゆくことで生成できる．このような方法を，以降「逐次法」と呼ぶこととする．2 章で述べた過去の研究においては，いずれもこの手法が用いられている．一方，我々が新しく提案する手法を，以降では「集約法」と呼ぶこととし，本章でこの 2 つの手法を説明する．

3.1 逐 次 法

逐次法は多面体を展開する一般的な手法である．最初に多面体を構成する面の 1 つを展開平面上に配置し，その後，展開平面上の面に 3 次元上で位相的に接続する面を 1 つずつ配置してゆく．新しく展開平面上に配置する面が，既に展開済みの面と干渉する場合には，干渉を起こさない別の面を選択して展開を続ける．このようにして全ての面が展開平面上に配置された時点で展開が終了する．展開が終了する前に干渉を起こさずに配置できる面が存在しなくなった場合には，まだ展開していない面を 1 つ選択し，別のパーツとして上記と同様の展開処理を行う．なお，展開図の中で互いに接続している一続きの面の集まりを本稿では**パーツ**と呼ぶ．

3.2 集 約 法

本稿で提案する集約法では，まず多面体の全ての面をバラバラに展開平面に配置してしまう．これはつまり，(パーツ数) = (面数) の展開図を作成することに等しい．その後，3 次元空間で位相的に接続している 2 面を展開平面上で接続し，徐々にパーツを大きくまとめあげてゆく．接続の際には，パーツの一方を固定し，他方を回転と平行移動によって展開平面上での位置を変更することで 1 つのパーツに統合する．例として，この集約法で正四面体の展開図を作成する様子を図 1 に示す．まず，正四面体のすべての面を個別に平面上に配置する (図 1(a))．続いて，3 次元空間で接続関係にある 2 面を共有する稜線で接続し，徐々にパーツを集約してゆく (図 1(b)〜(d))．この例では，パーツの数が 4 から 1 つずつ減少し，最後にパーツ数が 1 の展開図が作られる．

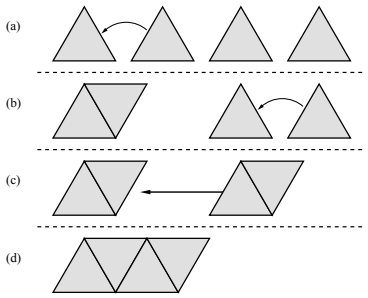


図 1 集約法による展開図作成の例
Fig. 1 An example of unfolding with 'gathering approach'.

この処理の流れは図 2 のようになる．なお，この流れ図では稜線毎に干渉 (Intersection) の有無を表すフラグ（I フラグ）を設定することで，干渉計算の重複を避けている．

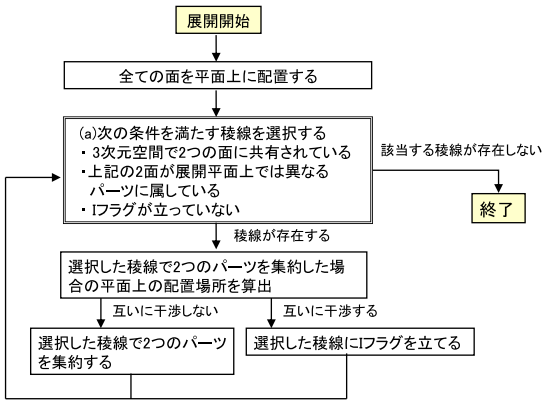


図 2 集約法による展開図作成の流れ
Fig. 2 Flow of 'gathering approach'.

4. 工作のコスト軽減への応用

従来の研究[11)12)] では，組立ての手間に着目し，切断する稜線の総延長や，折れ線の長さ，パーツの数，または用紙に展開図を配置した場合の余白の率など，様々な幾何的な値に基づいた展開図の評価が行われてきた．また，文献 13) では，人が一定距離の直線をカットするのに要する時間，一定数の折れ線を折り曲げるのに要する時間などを実際に計測することで導出した，式 (1) による展開図のコス

ト評価法が提案されている．

$$Cost = 0.08(L_{cut\_straight} + L_{break}) + 1.1L_{cut\_curve} + 3.2N_{stoit} + 4.7N_{break} + 7.7N_{paste} \quad (1)$$

式 (1) 中の変数はそれぞれ，$L_{cut\_straight}$（直線切断線の総延長），L_{break}（折れ線の総延長），L_{cut_curve}（曲線切断線の総延長），N_{stoit}（切断線上の鋭角点の数），N_{break}（折れ線の数），N_{paste}（貼り合せ箇所の数）である．

ところで，多面体の展開図の取りうるパターンの数は膨大であるため，すべての展開図のパターンから最適なものを見つけ出すことは現実的でない．また，既存のコスト評価法では，展開図が完成するまでその評価値を決定できない．そのため，展開の進め方に選択肢が存在したときに，その時点でよい展開図になりそうだと予想されるものを選ぶ**欲張りアルゴリズム**が妥当なアプローチである．2 章で述べた過去の手法においても，この欲張りアルゴリズムが使用されており，本稿で提案する集約法においても，集約を進める上でこのアルゴリズムを採用する．

4.1 選択肢の数の比較

展開図を作成する時に，逐次法では「最初に配置する面」の決定と「次の面の選択」，集約法では図 2(a) の「次の稜線の選択」において複数の選択肢が存在する．選択の場面において，それぞれの選択肢に（例えば稜線の長さなどの）評価値を設定し，最も評価のよいものを選択することによって，最終的にある程度評価のよい展開図を得ることができる．一般に，選択肢が多い中から選ばれたものの方が，少ない選択肢から選ばれたものよりも評価が高い可能性が高いため，選択肢の数は最終的な結果に影響を与えると考えられる．

そこで，逐次法と集約法での，それぞれ「次の面の選択」と「次に接続する稜線の選択」の選択肢の数が，どのように異なるかを調べた．例として，図 3 と図 4 に示す面数が 216 の単純な球形の多面体と面数が 280 のウサギの例を用いて実験を行った．結果はそれぞれに示すグラフのようになった．このグラフより，逐次法では選択可能な選択肢の数は展開の途中でほぼ一定であることがわかる．他方，集約法では，多面体に含まれる全ての稜線が最初に選択可能であり，展開処理が進むごとに減少してゆく．この結果より，逐次法よりも集約法の方が選択肢の数が格段に多いことがわかる．特に展開開始直後では，逐次法は最初に展開平面に配置した面に隣接する面（三角形の場合は 3 つ）だけであるのに対し，集約法では多面体に含まれる全ての稜線が選択可能であるため，大きな差となっている．従って，欲張りアルゴリズムで展開図を作成する場合には，集約法の方が効率よく評価の高い選択肢を選択できると考えられる．

4.2 稜線の長さを考慮した展開を行ったコスト比較

前節では，集約法の方が選択肢の数が多く，効率よく評価のよい選択を行えることを示した．ここでは，実際に稜線の長さを評価値に用いた展開図の作成を行い，それによって，実際に工作のコストをどの程度小さくできるかを検証する．

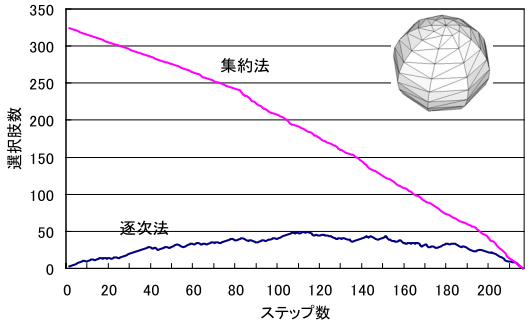


図 3 球形の多面体の展開図作成時に現れる選択肢の数の推移
Fig. 3 Transition of the number of choices during unfolding the spherical polyhedron.

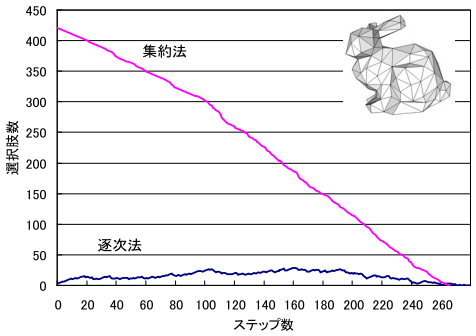


図 4 ウサギの多面体の展開図作成時に現れる選択肢の数の推移
Fig. 4 Transition of the number of choices during unfolding the bunny polyhedron.

展開図を工作する際には，切り取り線の総延長が短い方が展開図の切り出しの手間が少なく，またそれと同時に貼り合せの手間も少なくてすむ．そのため，工作のコストを小さくするには，多面体に含まれる稜線のうち，なるべく長い短い稜線で面が分割され，なるべく長い稜線で面が繋がるとよい．従って，逐次法で次に展開する面を選択する際には，選択可能なもののうち，最も長い稜線を介して接続する面を選択し，集約法で次に接続させる稜線を選択する際には，最も長い稜線を選択することとして実験を行った．なお，球形の多面体は容易に 1 つのパーツに展開でき，どちらの手法でも同じ結果となるため，球体の代わりにより複雑な形状として，面数 272 の恐竜のモデルを使用した．

その結果は表 1 の通りである．なお，例題の大きさはどちらも高さ 20cm とし，比較対象として稜線の長さを評価せずにランダムに次に展開する面を生成する手法も行った．このランダムな手法の値は，5 回の試行の平均とした．コストの算出には式 (1) を用いた（貼り付け箇所は切断線 2cm に 1 箇所とした）．この結果より，逐次法と集約法の両方とも，ランダムに生成した場合よりも切断線長の短い展開図を生成できることがわかる．また，集約法の方が逐次法よりもさらに約 5% コストを軽減できた．

表 1 手法による切断総延長と工作コストの比較
Table 1 Comparison of calculation time between different methods.

手法	ウサギ（280 面）		恐竜（272 面）	
	コスト	切断長 (cm)	コスト	切断長 (cm)
ランダム	3086	868	4178	989
逐次法	2621(0.85)	659(0.76)	3676(0.89)	785(0.79)
集約法	2480(0.80)	602(0.69)	3554(0.85)	741(0.75)

括弧内はランダムな手法の値を 1.0 としたときの相対値

5. パーツ数を考慮した展開

一般に凹凸の多い複雑な多面体は，1 つのパーツに展開できないことが多い．そのような場合は，複数のパーツに分割された展開図となるが，工作する上では，パーツの数は少なく，あまりに小さなパーツは存在しないことが望まれる．そこで，ここではパーツ数を少なく抑え，小さなパーツを減らすことを目的とした展開を行った場合の結果を逐次法と集約法で比較する．

逐次法では，展開が終わるまで全体を構成するパーツ数がわからないため,「谷折りとなるエッジを切断する稜線として選択すると，パーツが少なくなりやすい」などの経験則に基づいて展開を進める必要がある[11)]．ここでは，文献 11) で紹介されている方法で展開図の作成を行った．

一方，集約法では最初に (パーツ数) = (面の数) の展開図が存在し，これらを集約によってパーツを減らすことが行われる．そこで「最も小さなパーツの組を集約する」という，簡単なアルゴリズムを用いた．なお，ここでのパーツの大小は，パーツに含まれる面の数を用いる．

この結果は表 2 のようになった．経験則に基づいた逐次法では，恐竜の例でランダムな場合よりもパーツ数が増えてしまう結果となった．また，パーツに含まれる面数の分散が大きく，極端に大きなパーツと小さなパーツが混在していることがわかる．逐次法では，次に展開可能な面が存在する限り，別のパーツに分けることを行わないため，結果として 1 つ目のパーツばかりが大きくなり，面の数が 1 つや 2 つしか含まれない細かいパーツが後から生成されることになる．一方，集約法を用いると 1 つ 1 つのパーツの大きさをほぼ均一にすることができ，さらにパーツ数を少なく抑えることができたことがわかる．以上より，パーツ数を少なく抑え，小さなパーツを減らす目的でも，集約法は優れていることがわかる．

表 2 手法によるパーツ数の比較
Table 2 Comparison of the number of parts between different methods.

手法	ウサギ（280 面）		恐竜（272 面）	
	パーツ数	分散	パーツ数	分散
ランダム	10 (5)	5889	15 (9)	1637
逐次法	7 (3)	6371	20 (19)	3113
集約法	4 (0)	1739	12 (2)	581

カッコ内は含まれる面数が 3 以下のパーツの数

6. 組立て方を把握しやすい展開図の生成

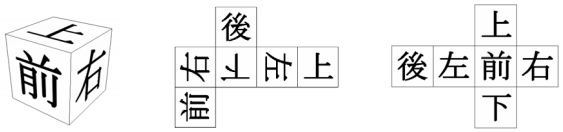
展開図を組み立てるための難易度について，従来の研究では切断線の長さや折れ線の長さなどの，展開図から得られる幾何的な値のみを考慮していた．ところで，立方体のような単純なものであれば，展開図を見ただけで組み立て方を容易に把握できるが，対象とする立体が複雑になると，どの辺同士を貼り合せるべきか，どのパーツ同士を貼り合せるべきかがわかり難く，その組み立て方を把握することが難しくなる．展開図のコスト評価式から組立てに要する時間を予測する手法を提案した文献 13) でも，面の数が増えると，思考に要する時間が増える傾向にあることが言及されている．そのため，展開図を作成する際には，実際に手を動かして展開図を切断したり貼り合せたりする作業以外に，組立て手順を思考する時間についても考慮すべきである．

そこで，本章では容易に組立て方を把握できる展開図を，集約法を用いて生成する手法を提案する．ここでは，幾何的な値からのコスト評価ではなく，主観的な評価によって，よいと判断される展開図を集約法を用いて生成することを目的とする．なお，ここで示す手法は，逐次法では実現できず，集約法によって可能となったものであり，集約法の有用性を示すものといえる．

6.1 面の向きを考慮した展開

立方体や球体などの単純な幾何形状には，上下左右という概念が無いが，車や人型のキャラクターなど，ペーパークラフトの対象となる形状の多くには，どちらが上でどちらが下になる，という空間上の「向き」の概念が存在する．また，このようなキャラクター形状では，面にテクスチャ画像を張りつけることで，よりリアルなペーパークラフトを作成することがよく行われる．このような場合，展開図でも 3 次元空間内の上下の向きが正しく維持されていることは重要である．例えば展開図内でキャラクターの顔が上下反転している場合など，組立て時に誤解を招く原因となる．

図 5 の (b) と (c) は，ともに (a) に示す立方体の展開図であるが，面の向きがバラバラな (b) よりも，面の向きが揃い，3 次元空間での上下が展開図でも維持されている (c) の方が，直感的にわかりやすい展開図である．



(a) (b) (c)

図 5 (a) テクスチャ画像の張られた立方体 (b) 面の向きを考慮していない展開図 (c) 面の向きを考慮した展開図
Fig. 5 (a) A cube with texture images. (b) An unfolded pattern without consideration for face directions. (c) An unfolded pattern with consideration for face directions.

そこで，以下に述べる方法で多面体を構成する各面に面の向きを表すベクトルを定め，その向きが展開図内でできるだけ揃うように集約法による展開を行う．

まず，面の向きを表すベクトルを，3 次元空間の z 軸の正の方向に平行なベクトルを各面に投影して定める．なお，xy 平面に平行な面については面の向きは x 軸の正の方向に平行になるものとする（図 6）．

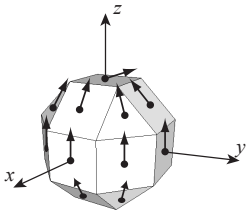


図 6 面の向きの設定
Fig. 6 Putting directions on faces.

面の向きを定めた後，集約法で展開図を作成するときに，稜線を介して接続する 2 面の向きベクトルの成す角の大きさを評価値に設定し，なるべく面の向きベクトルが同じ方向である 2 面を優先的に接続するようにする．

最後に，各パーツに含まれる全ての面の向きベクトルの平均が展開図の中で真上を向くように回転させる．

以上の方法で，図 7 に示す中央列の展開図を得ることができた．なお，左側の展開図の列は，面の向きを考慮せずに従来の遂次法で展開図を作成したものである．これにより，この手法を用いることで，面の向きが揃った展開図を生成できることがわかる．なお，展開図の中で，各パーツの配置に関しては考慮しておらず，パーツの外接四角形を端から詰めていくことを行っている．

6.2 左右対称性を考慮した展開

車やロボット，電車，動物など，ペーパークラフトの対象となるものには左右対称な形状が多い．このような場合，展開図も左右対称であれば，組み立て方を把握するのが容易となる．また，左右とも同じ組立て方をできるという利点がある．

そこで，左右対称な立体に対しては，前節で述べた面の向きに対する考慮とともに，左右対称な展開図になるような制約を設けて展開図の作成を行う手法を提案する．これは次のようにして行うことができる．

まず，3D データを読み込んだときに，左右対称な位置にある稜線の組を探索し，互いに対象位置にある稜線を記憶しておく．なお，ここでの左右対称とは，3 次元空間座標において，yz 平面に関して対称な位置に存在することをさす事とする．その後，前節と同様に面の向きを考慮した集約法による展開図の作成を行うが，ある稜線を介して異なるパーツの結合を行った時に，その稜線に対して左右対称な対となる稜線が存在する場合，その対となる稜線においても，同様のパーツの結合処理を同時に行う．

以上の方法で，図 7 に示す右側の列の展開図を得ることができた．これにより，この手法で各パーツが左右対称であり，かつ面の向きも揃った展開図を生成できることがわかる．展開図を視覚的に比較することで，この手法によって生成された展開図が，よりわかりやすい展開図であると判断できる．

7. 結　　論

本稿で新しく提案した集約法による展開図の作成は，従来用いられてきた遂次法よりも展開時の選択肢の数が多いため，同じ欲張りアルゴリズムを用いた場合には，遂次法よりも評価のよい展開図を生成できることが確かめられた．

また，面の向きや左右対称性を考慮することで，組み立て方を理解しやすい展開図を生成できることが確かめられた．これらの方法は，集約法が最初に全ての面をバラバラに配置し，それらを稜線を介して集約してゆくアプローチであるために実現したことであり，1 つ 1 つの面を順に展開してゆく，従来の遂次法では実現できなかった方法である．

以上より，集約法では組立てコストの小さな展開図を作成したり，組み立て方を把握しやすい展開図を生成するための，様々な応用が可能であると言える．

8. 展　　望

本研究では，展開図の各パーツの形に注目して，組立て方を把握しやすい展開図を考慮したが，各パーツをどのように展開図に配置するか，という問題も同様に重要である．パーツの配置については，工業製品の型抜きのように，如何に無駄な領域を少なくするか，という観点からの研究は過去になされているが[14]，ペーパークラフトの展開図として，理解しやすい配置，切り取りやすい配置などに関する研究はまだされていない．

また，従来の展開図の評価方法では，展開図の切り取りや折り曲げ作業に要する時間をベースに値を算出していたが，本研究で提案したように，組立て方を把握しやすいか否かも，展開図を評価するうえでは重要である．これらも考慮に入れて展開図の評価を定量的に行えるようにすることは今後の課題である．

参 考 文 献

1) 石松丈佳: 立体描写・制作課題の問題点, 日本図学会 2003 年度本部例会学術講演論文集, pp. 71–74 (2003).
2) ジャストシステム出版部: デジタルペーパークラフト, ジャストシステム (2001).
3) スルッと KANSAI 協議会: 電車・バスペーパークラフト本, 阪急電鉄 (2001).
4) 磯田浩, 鈴木賢次郎: 図学入門, 東京大学出版会 (1986).
5) Elber, G.: Model fabrication using surface layout projection, *Computer-aided Design*, Vol. 27, No. 4, pp. 283–291 (1995).
6) Pottmann, H. and Farin, G.: Developable rational

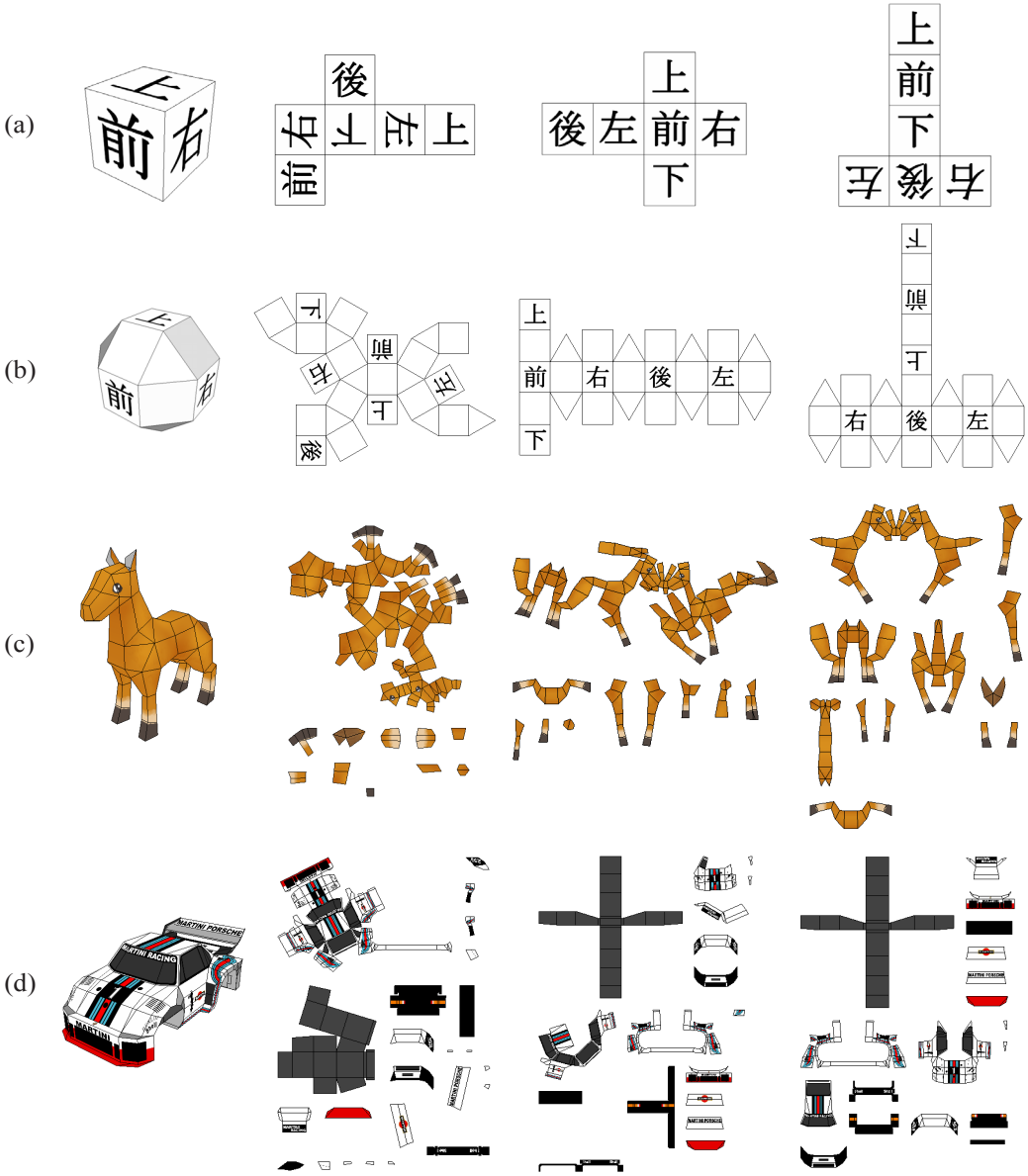


図 7 組立て方の把握しやすさ考慮して生成した展開図．左から順に多面体，逐次法を用いた展開図，集約法で面の向きを考慮したもの，集約法で面の向きと左右対称性を考慮したもの

Fig. 7 Generated patterns by considering the understandability of the procedure of assembling.

Bèzier and B-spline surfaces, *Comput. Aided Geom.*, Vol. 12, No. 5, pp. 513–531 (1995).

7) Hoschek, J.: Approximation of surfaces of revolution by developable surfaces, *Computer-aided Design*, Vol. 30, No. 10, pp. 757–763 (1998).

8) Mitani, J. and Suzuki, H.: Making Papercraft Toys from Meshes using Strip-based Approximate Unfolding, Vol. 23, pp. 259–263 (2004).

9) 穂坂・木村: GEOMAP ソースプログラム仕様書. http://www.ke.ics.saitama-u.ac.jp/kondo/Geomap/ GeomapProgSpecJ/Contents.html.

10) 多摩ソフトウェア: ペパクラデザイナー. http://www.tamasoft.co.jp/pepakura/.

11) 三谷純, 鈴木宏正, 木村文彦: 3 次元ポリゴンモデルの展開図作成, グラフィックスと CAD 研究報告, Vol. 96, グラフィックスと CAD 研究会, pp. 13–18 (1999).

12) 和田弘重, 高井那美, 高井昌彰: 組立ての容易さと曲げ変形を考慮した 3D ポリゴンモデルの展開図生成, 情報処理学会研究報告, Vol. 2003, 情報処理学会, pp. 45–50 (2003).

13) 三谷純, 鈴木宏正: ポリゴンモデルの展開図組み立てに要するコスト評価法, 日本図学会 2003 年度本部例会学術講演論文集, pp. 27–32 (2003).

14) Milenkovic, V. J.: Rotational polygon containment and minimum enclosure using only robust 2D constructions, Vol. 13, pp. 3–19 (1999).