面を基本要素とした３次元モデリング

五十嵐 健夫　　佐々木 直　　坂本 大介*

概要. 通常，３次元形状モデリングにおいては，空間中の頂点がもっとも基本的な要素として利用されている．ユーザはまず空間中に頂点を配置し，それらの頂点間を辺で連結し，さらに，辺によって閉じられた領域を面として定義する．SketchUp の push/pull 操作のように，面を基本とした操作も存在するが，内部表現はやはり頂点を基本としたものであり，複雑な形状を生成するためにはさまざまな操作を組み合わせて使用しなければならない．本稿では，これとは異なる方法として，面要素を基本とした形状表現を提案する．ユーザは空間中に面要素を配置するだけで３次元形状モデリングを行うことができる．面要素は，３次元空間中の方向を持った点として表現される．システムは，これらの面要素の表す平面同士の交線として辺を生成し，これらの辺に囲まれた領域を３次元モデルを構成する面としてユーザに提示する．特にモーションキャプチャなどの３次元入力装置を利用した場合，面要素の位置と方向を１操作で指定することができるので，提案手法によって３次元モデリングを非常に効率よく行うことができる．

1　はじめに

通常，３次元モデリングにおいては，空間中の頂点がもっとも基本的な要素として利用されている．ユーザはまず空間中に頂点を配置し，それらの頂点間を辺で連結し，さらに，辺によって閉じられた領域を面として定義することで３次元モデルを表現する (図 1 左)．このような手法は，計算機の内部表現を直接反映したものであるため，実装が容易であり，広く用いられている．しかし，建築物のように，大きな平面の組み合わせで構成されたような３次元モデルを表現する場合には，特定の面を平面性を保ちつつ平行移動したり回転したりといったことを行うために，関係する複数の頂点を正しく協調させて操作する必要があり，操作の負担が大きい．たとえば，立方体の面を１つ引き出すためには，四頂点を平行に移動することが必要である．SketchUp の push/pull 操作は，面を直接つかんでひっぱりだしたり押し込んだりしてモデリングを行うことを可能にしている．しかし，内部表現はやはり頂点を基本としたものであり，何もないところにいきなり面を置いたりすることはできず，また既存の面を回転したりするとまわりの面の平面性が失われるといった問題を内包している．

本稿では，このような頂点を基本とした表現手法に代わる，インタラクティブな３次元モデリングのための新たな表現手法として，面要素を基本とした形状表現を提案する (図 1 右)．提案手法では，ユーザは空間中に方向を持った点である面要素を配置するだけでモデリングを行うことができる．システムは，これらの面要素が表す平面同士の交線として辺を生成し，これらの辺に囲まれた領域を目に見える面領域としてユーザに提示する．このような手法を用いることにより，複数の頂点を個別に操作するのでなく，面を直接操作するようなモデリングが可能となる．本手法は，特に，面要素の位置と方向を１つの操作で指定することができるモーションキャプチャなどの３次元入力装置を利用した場合に，有用であると考えられる．

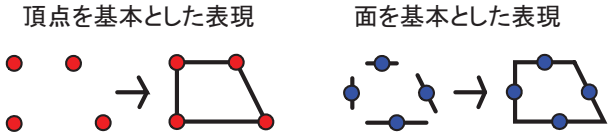

図 1. 点要素による表現と面要素による表現

2　関連研究

これまでに，さまざまな３次元形状の表現手法が提案されてきている．もっともよく使われている手法は，ポリゴンによる表現である．これは，３次元形状の表面を，多面体で表現し，その多面体を構成する頂点の座標によって形状を定義するというものである．しかし，このような手法は，前節で説明したように，直接面を操作するような操作には適していない．たとえば，面を移動したり回転したりしたい場合に，周囲の頂点を適切に移動しなければならず，操作の負担が大きい．また，面と面の間に隙間ができたりするという問題もある．そのほか，有機的な形状を表現するのに適した手法である陰関数表現 [1]，微細な内部構造を表現するのに適した手法としてボクセル表現 [7]，微細な凹凸を高解像度に表現する手法である点群表現 [8] などがあるが，建築物のように少数の平面から定義される物体の形状表現に適

* Takeo Igarashi, Naoki Sasaki, Daisuke Sakamoto, JST ERATO 東京大学

しているとはいえない．機械部品のような形状を表現するのに適した手法として，CSG (Constructive Solid Geometry) と呼ばれる手法がある．しかし，通常の CSG は，直方体や円柱といった，大きさがあらかじめ決まったプリミティブの組み合わせでソリッド形状を表現するというものであり，特定の面だけを移動したり回転したりする操作には適していない．

形状操作の手法についても，これまでさまざまなものが提案されてきている．もっとも基本的なものは，ポリゴンを構成する頂点や辺，面を直接に追加・削除したりを移動したりするというものである．ただ，このような手法は，細かい形状の調整には適しているが，ざっくり形状を表現するには手間が大きすぎる．もう少し，大きな単位で作業を行う方法として，回転や掃引 (extrusion) といった操作がよく使われている．これらの方法は，複数の頂点にまたがった操作を一度に行えるという意味で効率がよいが，まず適切な機能（ツール）を選択した上で，操作対象の要素（頂点・辺・面）を指定しなくてはならないという手間がある．手早く簡単にモデリングを行う手法としてはスケッチ操作に基づくものがある [11, 10, 5]．しかし，これらのシステムは，２次元の入力デバイスを用いて３次元形状を入力するので，建築物のような様々な方向を向いた面から構成されるような形状を表現するためには，多くの操作が必要となる．３次元入力デバイスを用いたモデリング手法もいろいろと提案されているが [9, 3, 6]，主に有機的な形状を表現するものであり，少数の面で定義される多面体のモデリングを対象とした提案手法とは目的が異なる．３次元入力デバイスによる家具のモデリングシステム [4] も提案されているが，これも CSG に近い考え方であり，建築物のような大きな面から構成されるもののモデリングには適していない．

本研究にもっとも関連するものとして，Google 社のモデリングソフトである SketchUp の push/pull 操作 [2] がある．これは，面を直接つかんでひっぱりだしたり押し込んだりしてモデリングを行うことを可能にするものであり，本研究で提案している面要素にもとづくモデリングをある程度実現したものといえる．しかし，内部表現はやはり頂点を基本としたものであり，面を回転したりすると，まわりの面の平面性が失われるといった問題がある (図 2 左)．また，図 2 右のように壁を新しく１枚追加するときに，提案手法では面要素を１つ空間に追加するだけであるが，SketchUp では線で新しい面領域の輪郭を設定したあとに不要な面と線を削除するという数多くの手順を踏む必要がある．

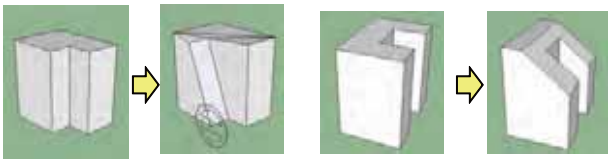

図 2. 通常のポリゴンモデリングの問題点．左：回転によって周囲の面の平面性が失われる．右：面を一つ追加するのに複雑な手順が必要．

3 ユーザインタラクション

本節では，提案手法によるモデリング操作の概要について，具体的な例を用いて説明する．内部での計算方法については，次節で詳しく説明する．

本手法は，建築物のような少数の平面から構成される３次元形状をインタラクティブに作成するための手法である．ユーザが行う主な操作は，向きを持った点として表現される面要素の空間中への配置とその回転・移動である．入力手法としては，３次元空間における向きと位置を直接的に一度に指定することのできる３次元入力装置を用いることを想定している．通常の２次元のマウスを用いることもできるが，３次元的な位置と向きを指定するには，３Ｄウィジェットやジェスチャなどを利用した間接的な操作が必要となる．

図 3 に提案手法を利用したモデリングの例を示す．まず何もない空間に１つの面要素を配置すると，そこに面が１つ生成される．概念的には無限の広がりをもっているとの想定であるが，実装の都合上，対象シーンを覆うような大きな正方形に収まるようにトリミングを行っている．すべての面は地面によってもトリミングされる．次に，２つめの面要素を配置すると，すでに存在する面と新しい面の間の交線が計算され，その交線で区切られた半無限の面が生成される．その後さらに，２つの面要素を追加することによって，多面体が表現される．多面体を構成する面要素を移動したり回転したりすることによって，自由に多面体の形状を変更することができる．その際，頂点や辺の位置は，隣り合う面の間の交点として自動的に計算される．このようにすることで，図 2 左のような問題を根本的に回避することができる他，図 2 右のような編集も面要素を一つ追加するだけで実現できる．

なお，提案手法では，４面が１点で交差するような形状（４角錐の頂点）を表現するには，４つの面要素を特別な位置関係に置かなければならず，簡単に表現できるとはいえない．しかし，この問題は，点を基本とした表現において４角形を平面として表現するためには４点を特別な位置関係に置かなければならない問題と双対な関係にあるといえる．一般的な建築物では，３角形の面よりも４角形（あるいはそれ以上）の面が多く，また，３面の交わる角の

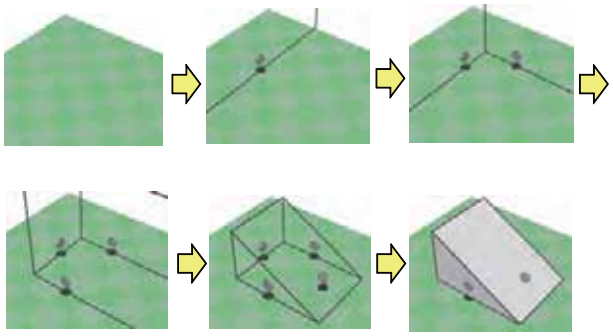

図 3. 面要素によるモデリング．ユーザが空間中に面要素を置いていくと，それらの面同士の交線によって面領域が与えられる．

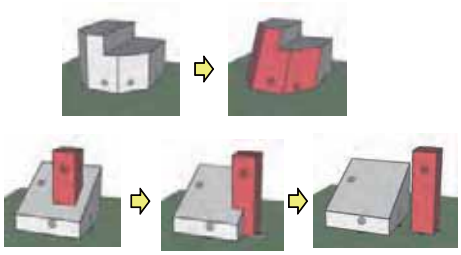

図 4. グループを利用した操作の例．一部の面の集合をグループ（赤く表示されている部分）として定義し，まとめて回転・移動することができる．

ほうが４面（あるいはそれ以上）の交わる角よりも頻繁に表れるので，面を基本とした表現の方が有利であると考えられる．

また，複数の面をグループとして定義し，それらをまとめて移動・回転させることで，より効率のよいモデリングを行うことも可能である．編集中の面要素をまとめて新たにグループを作成する他，よく使う複合プリミティブについては，あらかじめ定義されたグループとして用意しておくことで作業の効率化を図ることができる．現在の実装では，２枚の面を組み合わせて角を表現するもの，３枚の面を組み合わせて平面上に追加する柱を表現するもの，４枚の面を組み合わせて４角柱を表現するもの，５枚の面を組み合わせて平面上におく直方体を表現するもの，などを用意している．グループ化された面集合については，そのうちの一つの面が核として設定され，ユーザはこの核を移動・回転することで，グループに対する移動・回転を行う．図４にグループを利用したモデリング操作の例を示す．提案手法を用いると，このような操作をグループの回転や移動という１操作で行うことが可能となる．一方，頂点を基本とした表現におけるグループ機能ではこのような処理を簡単に実現することができないので，複数の頂点を手作業で個別に移動したり作成・削除したりしなくてはならず，手間が非常に大きいといえる．

現在の実装でサポートされている要素は平面要素だけであるが，今後，湾曲した面を表現する曲面要素や，円柱の側面を表現する円柱要素などを実装していく予定である．曲面要素は非常に大きな半径を持った円柱として表現する．角柱は平面要素を組み合わせたグループとして表現される．いずれの場合も，円柱要素や角柱要素自体は，上下方向に無限の広がりを持っていて，他の面要素との交差によってその広がりが決定される点が，要素自体があらかじめ決められた大きさを持っているＣＳＧ表現と根本的に異なっている．

4　アルゴリズム

本研究の重要な貢献は上記のようなモデリング操作を実現するアルゴリズムデザインにある．面要素の数が少数かつ凸である場合には，得られる多面体の形状は一意に定まるため問題はないが，面要素の数が増えその空間的な位置関係が複雑になってくると，それらからどのような多面体を構成するかは自明でなく，慎重なデザインが要求される．特に，凹な (concave な) 部分をもつ多面体を適切に表現できるようなアルゴリズムとするためには工夫が必要である．以下，まずいくつかの考えうる計算方法の候補を紹介し，それらをもとに最終的なデザインについて説明する．なお，本アルゴリズムを適用することにより，面要素の集合が通常良く使われるポリゴンモデル（多角形の集合）に変換されるため，既存のレンダリング手法を適用して表示したり，外部ファイルに出力して他のアプリケーションで利用したりすることが可能となっている．

まず，最初に考えられるのが，それぞれの面要素からもっとも近い交線によって面領域を定義する方法である．ここでは，面要素の定義する平面の間の交線をすべて計算し，それらの交線によって表現される面領域のうち，それぞれの面要素を含む最小領域をその面要素の定義する面領域とする．このような手法は，凸多面体の場合にはうまくいくが，凹多面体の場合には開いた辺（辺がその辺を定義する面領域に接していない）が生じてしまうために分かりにくくなってしまう (図 5b, e)．これを防ぐ方法として，開いた辺があった場合には，その辺を削除して面を閉じた辺にぶつかるまで伸ばすという方法が考えられる (図 5c)．しかし，この方法では，図 5d のような３すくみのような関係に面要素が置かれている場合に，どのような形状を得るかが一意に定まらない．面要素の置かれた順番を考慮する方法も考えられるが，面の順番と生成される形状の関係が自明でなく，目的とする結果を得るためにどのような操作を行えばよいのかが分かりにくいという問題がある．

次に考えられるのは，純粋に面要素の位置だけでなく，交差辺をもつ面要素対を明示的に指定すると

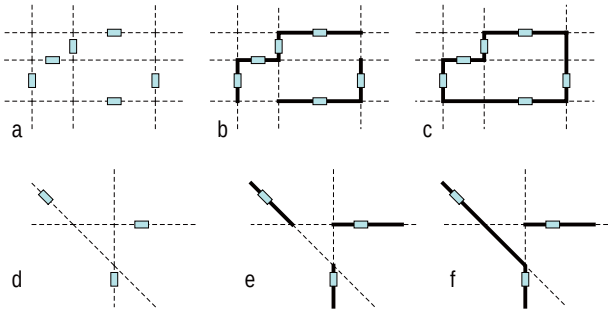

図 5. 面領域計算アルゴリズムの検討．すべての交線を調べて最小領域を得る方法 (b,e) と提案手法による結果 (c,f).

いう方法である．これは，通常の点要素を基本とした表現において，頂点と頂点の間に明示的に辺を定義しているのと同じようなものであり，実質的に面要素をノードとした無向グラフ構造を定義することと同値である．ただ，このような方法は，面を生成したり動かしたりするたびに，面要素間の関係をいちいち明示的に決めなければならず，ユーザの手間が大きくなってしまう．また，図 5d に示すような状況では，単に個別の面のペアの間の連結関係を指定するだけでは不十分であり，面のペアとペアの間の優先順位を指定しておく必要が生じる．このような状況を適切に表現する方法としては，他にも，すべての面の関係を有向グラフ構造で表現する方法などが考えられるが，位置や方向と別に複雑なトポロジ情報を管理する必要があることになり，ユーザにとってわかりやすいとは言えないものになってしまう．

以上のような考察を踏まえ，提案手法では，すべての面の間の対を列挙し，それらを面要素の間の距離によってソートし，距離の小さい対から順に処理して面領域を決定していくことで，最終的な多面体 (ジオメトリ) を得るアルゴリズムを利用している．このようにすることで，グラフや順序といった目に見えない隠れた構造を導入することなく，面要素の置かれている空間的な位置関係という目に見える情報だけで面領域を一意に定めることが可能となる (図 5c,f)．この計算に当たっては，各面要素に関連づけられている面領域は最初，無限の広がりを持つ面を，シーン全体を囲む大きな正方形でトリミングしたものとして与えられ，その後，その面を含む対が処理される際にその対によって与えらえる交線によってトリミングされる（これを面の対の範囲決定処理と呼ぶ）．図 6 に例を示す．n 個の面要素があった場合には，n^2 個の対を列挙してソートするため，計算量的に多少問題があるが，本手法の対象が，デザインの初期段階における比較的簡単なモデルの作成であること，および，複雑なモデリングの際はグループ化が利用されると考えられることなどから，実用上は大きな問題にはならないものと考えている．

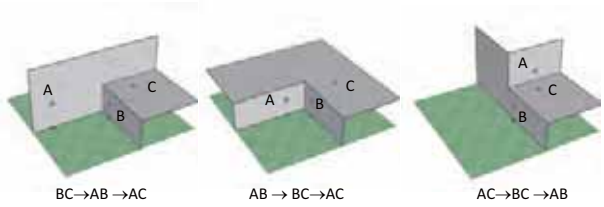

図 6. 面要素間の距離に基づく面領域の構成．面要素間の距離関係によって結果が異なっている.

グループ構造がある場合には，まずグループ内の面要素同士で範囲決定処理を行い，それらを結合してグループのジオメトリとする．その後で，できあがったジオメトリとグループ外の面要素との間で範囲決定処理が行われる．そのため，面要素の位置関係が同じでも，グループ構造によって，最終的に得られる形状が異なってくる（図 7, 8）．なお，グループにはそれぞれ核となる面要素が 1 つ設定されており，グループ外の面要素との間の範囲決定処理では，その核に近い領域が残される．図 8 では，面要素Ａが，面要素Ａ，Ｃからなるグループの核となっている．グループ化されていない場合には，ＢＣの間の範囲決定処理が最初に行われているが，グループ化されている場合には，まずＡＣが結合され，次にＡを核とした結合ジオメトリとＢとの間の範囲決定処理が行われている．

より一般的には，グループのグループといったような階層的な構造となるため，全体は木構造として表現される．末端の葉ノードが面要素を，中間ノードがグループを，根がシーン全体を表現する．シーン全体のジオメトリを決定するにあたっては，この木構造をボトムアップにたどりながら，それぞれの中間ノードの下についている子ノード群の間で範囲決定処理をしながら全体のジオメトリを決定していく．

1 つの中間ノードの下についている子ノード群をまとめる際には，面要素群をまとめる場合と同じように，すべてのノード間の対を列挙し，それらをノードの核の間の距離によってソートし，距離の小さい順に対を範囲決定処理してノード内の面の領域の決定をしていく．対となったノード（グループ）の範囲決定処理については図 9 を用いて説明する．まず，グループを構成するジオメトリの中の面の間のすべての対について交線を計算し，その交線によって各面を領域に分割する．その上で，各グループの核を起点としてその周囲に範囲を広げながら，残すべき領域を決定していく．このとき，核からの距離の近い順に処理を行い，交差する相手に隣接するまで，領域を広げていく．

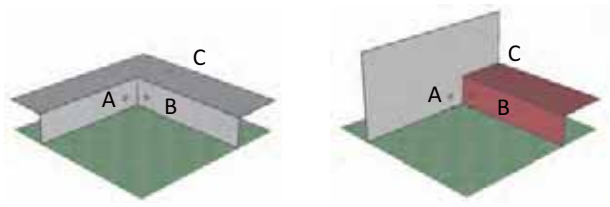

図 7. グループ化の例 1．グループ化しない場合 (左) と BC をグループ化した場合 (右).

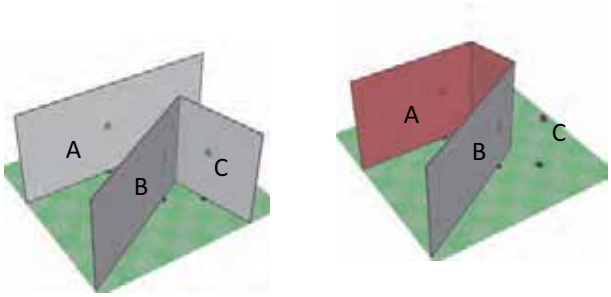

図 8. グループ化の例 2．グループ化しない場合 (左) と，AC を A を核としてグループ化した場合 (右).

5　実装

ソフトウェアは Java で実装されており，Windows 7 上で動作している．３次元入力装置としては，MotionAnalysis 社の３次元動作解析システムを用いている (図 10)．この３次元動作解析システムは，13 個のカメラを用いた光学式モーションキャプチャ装置であり，計測対象に取り付けられたマーカを毎秒 100Hz，誤差 1mm で３次元的な位置計測が可能である．本稿で使用するシステムでは底辺が 2.5m x 3m，高さ 2m の空間中の物体を測定可能である．

面要素を指定する道具としては，Wii リモコンの先にマーカーを付した円板を取り付けた道具を作成して用いている (図 11)．マーカーは，方向を持った面要素を表現する半径 20cm の円板の円周上に４つ配置されている．３点があれば平面が一意に決まるが，１つでもユーザの陰になるとトラッキングができなくなるため，マーカーを４つ用いている．ユーザは，wii リモコンのボタンを押すことで，円板の中心に新しい面要素を配置する．

なお，３次元入力装置を利用できない場合のために，２次元の入力装置向けに，SKETCH システムのような簡単なジェスチャーと直接操作を組み合わせたインタフェースを提供している．

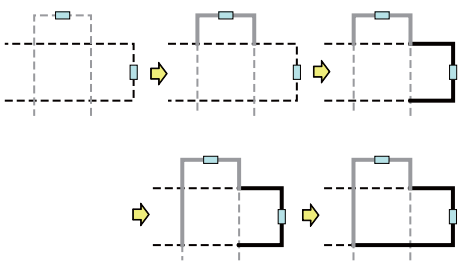

図 9. グループの対を処理する際のアルゴリズム．縦になったコの字型のグループと横になったコの字型のグループが処理されるプロセスを図示している.

6　結果

図 12 に，提案手法を用いてモデリングした結果の例を示す．ほとんどの例について，特殊な編集操作やグループ構造を組み合わせることなく，単に面要素を必要なだけ空間中に配置するだけでモデリングできている．既存のツール (Google SketchUP など) で同様なモデリングを行う場合には，線の描画や push/pull 操作などのツールを適切な順番で組み合わせて使用する必要があり，ユーザにとっての負荷が大きいと考えらえる．特に，作成するべきゴール形状が明確に決まっておらず，試行錯誤しながらデザインを行っていくような場合，ツールの切り替えをせずに要素の配置だけで多様な形状をモデリングできることは，自由な思考を妨げないという意味で非常に重要であると考えられる．

7　まとめと今後の課題

本稿では，面要素を空間においていくことによる３次元形状のモデリング手法について紹介した．従来の頂点をベースにした手法に比較して，面の追加や移動など，面を単位とした操作をより少ない手間で行うことができる．特に，従来のモデリングシステムのように線の描画や掃引など複数のツールを組み合わせてモデリングを行うのでなく，面要素を配置するという一種類の操作のみでモデリングを行うことができる点が重要であると考えている．また，

図 10. モーションキャプチャ装置を利用したモデリング

図 11. wii を用いた入力装置

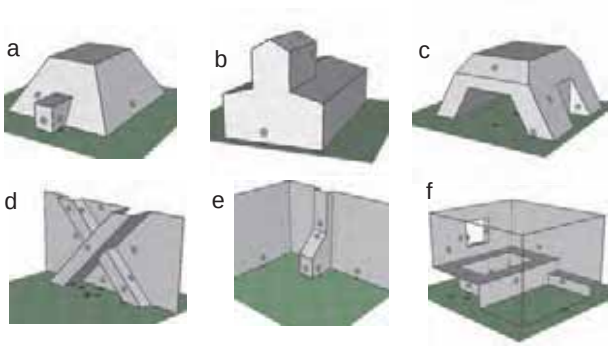

図 12. 面要素によるモデリングの例.

本手法は，３次元的な位置と方向を同時に入力できるような環境に特に適した手法である．

提案手法の問題点としては，面を中心としたモデリングには適しているものの，頂点を中心としたモデリング作業には適していないという点が挙げられる．特に，動物やキャラクタのような有機的な形状の表現を行う場合には，頂点を基本とした表現の方が適していると考えられる．本手法はあくまでも，大きな面から構成される建築物のようなモデルのラフスケッチを対象として考えている．

アルゴリズムについては，４節で議論したように，いろいろな可能性を検討した上でのベストな選択として，面と面の間の距離の小さい順に処理するというアルゴリズムを採用しているが，これがあらゆる場面，あらゆるユーザにとってもっともよいものとは限らない．特に，面要素同士がほぼ平行な場合に，要素間の位置関係の微妙な変化によって，最終的な構造が大きく変化することがあり，混乱のもととなる可能性がある．

今後の発展について，まずは基本要素である曲面要素の実装を行う予定である．さらに，単純な平面や曲面だけでなく，岩肌やブロック塀のようなテクスチャやレリーフなどのデコレーションを持った平面要素の導入を検討している．いずれの場合についても，単純な平面と比べて交線の計算が複雑になるので，その処理に工夫が必要となると考えられる．長期的な課題としては，本稿で紹介したものと異なる面領域計算アルゴリズムについても検討していきたい．たとえば，現状の実現方法では，T字状の形状を表現するためには，上辺の両側にそれぞれ面要素を置かなくてはならない．別の方法として，下辺は上辺に従属するという関係を与えるというアルゴリズムも考えられる．また，今回の実装では，面の両側を区別しない薄い板のような存在として，面を定義している．しかし，別の方法として，面をソリッドの境界として定義し，常に，面の表側（ソリッド外部）と面の内側 (ソリッド内部）を区別するという表現も考えられる．

参考文献

[1] J. F. Blinn. A Generalization of Algebraic Surface Drawing. *ACM Trans. Graph.*, 1(3):235–256, July 1982.

[2] J. E. U. Brad Schell, Joe L. Esch. System and method for three-dimensional modeling, US Patent 6628279, nov 2000.

[3] T. A. Galyean and J. F. Hughes. Sculpting: an interactive volumetric modeling technique. In *Proc. SIGGRAPH '91*, pp. 267–274, 1991.

[4] M. Lau, M. Hirose, A. Ohgawara, J. Mitani, and T. Igarashi. Situated modeling: a shape-stamping interface with tangible primitives. In *Proc. TEI '12*, pp. 275–282, 2012.

[5] H. Lipson and M. Shpitalni. Optimization-based reconstruction of a 3D object from a single freehand line drawing. *Computer-Aided Design*, 28(8):651 – 663, 1996.

[6] I. Llamas, B. Kim, J. Gargus, J. Rossignac, and C. D. Shaw. Twister: a space-warp operator for the two-handed editing of 3D shapes. *ACM Trans. Graph.*, 22(3):663–668, July 2003.

[7] H. Pfister, J. Hardenbergh, J. Knittel, H. Lauer, and L. Seiler. The VolumePro real-time ray-casting system. In *Proc. SIGGRAPH '99*, pp. 251–260, 1999.

[8] H. Pfister, M. Zwicker, J. van Baar, and M. Gross. Surfels: surface elements as rendering primitives. In *Proc. SIGGRAPH '00*, pp. 335–342, 2000.

[9] S. Schkolne, M. Pruett, and P. Schröder. Surface drawing: creating organic 3D shapes with the hand and tangible tools. In *Proc. CHI '01*, pp. 261–268, 2001.

[10] R. Schmidt, A. Khan, K. Singh, and G. Kurtenbach. Analytic drawing of 3D scaffolds. *ACM Trans. Graph.*, 28(5):149:1–149:10, Dec. 2009.

[11] R. C. Zeleznik, K. P. Herndon, and J. F. Hughes. SKETCH: an interface for sketching 3D scenes. In *Proc. SIGGRAPH '96*, pp. 163–170, 1996.