Vol.2012-CG-146 No.30
2012/2/8

クロス&グラフトに基づく 2D モデリングインタフェース

八坂 俊[†1] 藤代 一成[†1]

本稿では, 簡単に 2 次元オブジェクトをモデリングするための新しいパラダイムとして, クロス&グラフトを提案する. クロス（crossing, 交配）は 2 つのオブジェクトから両者の特徴を併せもった新たな世代のオブジェクトを生成する操作であり, グラフト（grafting, 接合）はオブジェクト同士を接合する操作である. ユーザはプリミティブからスタートし, クロスやグラフトを繰り返すことで, 目的のオブジェクトを得る. クロスは, グラフトにより得られる接合情報を利用して実行される. これにより, 簡単な操作であっても大幅な変更を施すことが可能になる. また, システムがユーザの予想を超えた結果を瞬時に返すことで, ユーザの発想が拡がる可能性もある.

2D Modeling Interface Based on Crossing and Grafting

Shun Yasaka and Issei Fujishiro

In this article, crossing & grafting is proposed as a new paradigm for effective modeling of 2D objects. Crossing is an operation to generate a new-generation object with features inherited from two objects whereas grafting is another operation that connects two objects geometrically. A user is allowed to obtain his or her target object through repetition of these two operations starting with a primitive. Since crossing is evaluated using the configuration information specified by grafting, even a simple operation can give rise to big change in appearance. Moreover, instant feedback of unexpected effects has a potential to inspire a user's way of design.

†1 慶應義塾大学 大学院理工学研究科
Graduate School of Science and Technology, Keio University

1. 序 論

モデリングインタフェースには, より少ない入力でモデリングを行うことや, 複雑なオブジェクトの作成, 高い修正・編集機能など, 多くの要求があるが, それらは互いにトレードオフの関係にあり, 同時に実現することが難しい. そのため, 近年さまざまな手法によるモデリングインタフェースが提案されている. しかし, 未だにユーザの発想自体を支援することを目的としたインタフェースは少ない. また, ほとんどのインタフェースは正確なモデルの作成を目的としているため, 目標とするモデルを明確に定めずにモデリングを行うスタイルを想定していない. そのため, 作成中のモデルに大幅な変更を与えることが難しくなってしまっている. これらは正確なモデルを作成するために課された制約であるため, 作成するモデルの正確さや変更の確実さをある程度諦めることで, それらを解決するモデリングインタフェースが実現可能と考えられる.

本稿では, 簡単にモデルの大幅な変更を実現し, さまざまなデザインを瞬時に示すことで発想を支援する新たなモデリングパラダイムとして, クロス&グラフトを提案する. グラフトは 2 つのオブジェクト同士を 1 点で接合する操作である. これはオブジェクトの特徴に本質的には変化を与えない操作であり, これにより得られる接合関係を利用することで, クロスをより効果的に機能させることができる. 一方クロスは, あるオブジェクトを他のオブェクトにドラッグ&ドロップすることで, 両者の特徴を併せもった新たな世代のオブジェクトを生成する操作である. 複数のオブジェクトから成るオブジェクトへクロスする際は, グラフトによる接合関係から決まる, ある規則に従ってその効果が決定される. これにより, ドラッグ&ドロップという簡単で直感的な操作にもかかわらず, オブジェクト全体へ大きな変化を与えることが可能になり, ユーザの入力の手間を減らすことができる. 例えば, 角張ったオブジェクト全体に丸みをもたせるような変更は, 1 回のクロスにより簡単に実現される. また, オブジェクトはそれぞれ遺伝子構造をもっている. クロスの際には両者の遺伝子を交叉するような操作が行われ, そこから生成されるオブジェクトの特徴が決められる. そのため, 同じ操作であっても結果が同じになるとは限らず, 多様なパターンのなかから 1 つが選ばれ, 生成されることになる. このように, さまざまなデザインを簡単な操作で示し, また, ときに思いもよらない結果を返すことで, ユーザの発想を拡げられる可能性もある.

本稿では, クロスとグラフトの考え方を分かりやすく示すために, まず 2D に限定した実装を試みる. 開発中のシステムのインタフェースを**図 1** に示す. 中央がモデリングスペースで, 上側と左側にそれぞれツールバーがある. 上側のツールバーにはカラーパレットや undo

© 2012 Information Processing Society of Japan

Vol.2012-CG-146 No.30
2012/2/8

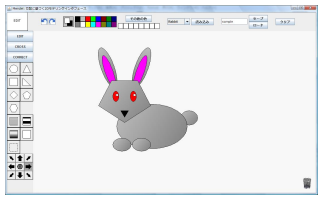


図 1 本システムのインタフェース画面
Fig. 1 A screenshot of the system interface.

ボタンが, 左側のツールバーにはシェイプパレットとテクスチャパレット, ディレクションパレットがそれぞれ配置されている.

本稿は次節以降, 以下のように構成されている. まず次節でモデリング手法の関連研究を挙げ, 3 節で本手法の特長について述べる. 続く 4 節でデザインの出発点であるプリミティブ, 5 節でグラフトについて詳細な説明を行った後, 6 節でクロスに焦点をあてる. 7 節でシステムの評価を行い, 8 節で本稿をまとめ, 今後の課題に言及する.

2. 関 連 研 究

前節で述べたように, モデリングインタフェースにはさまざまな手法を用いたものがある.

ジェスチャベースのモデリングインタフェースは, ジェスチャを用いてオブジェクトを作成する. 最も代表的なシステム SKETCH[1] では, 点を打つ, 線を描くといったジェスチャによりプリミティブを作成する. 例えば, 打った点の周りを囲むように線を描くことで, 球体を作成する. ジェスチャにはさまざまな種類があり, それにより, 管状のプリミティブやひょうたん型のプリミティブなどを含む多様なプリミティブを作成することができる. このように, ジェスチャベースのモデリングは, 少ない入力で複雑なモデルを作成することに適している反面, どのジェスチャがどの操作に対応するかを覚える必要があり, 慣れていないと操作が難しい.

スケッチベースのモデリングインタフェースは, ユーザのスケッチを入力としてモデリングを行う. その先駆的なシステムに Teddy[2] がある. Teddy はひじょうにシンプルで親しみやすいインタフェースを提供し, モデリングに不慣れなユーザでも簡単に扱うことができる. システムはユーザが描いた 2D の輪郭に合わせて 3D モデルを作成する. しかしながら, スケッチベースのモデリングインタフェースは, 直感的に操作ができる反面, 作成できるオブジェクトの種類が少ない傾向がある. 例えば, Teddy では丸みのあるオブジェクトしか作成できない.

これらのインタフェースを含め, ほとんどの研究ではユーザの発想自体を支援することに主眼をおいていない. たとえば Xu らの研究[3] では, システムがあらかじめもっている 3D 候補モデルを, ユーザが用意した画像上のオブジェクトに合わせて変形することで, ユーザの気に入ったデザインのモデルを簡単に作成することができる. このシステムは, 自ら何もない状態からモデルを作成することはできず, 他のインタフェースを用いて作成したモデル（3D 候補モデル）を画像に合わせて変形することでモデリングを行う. 3D 候補モデルはあらかじめ構造情報が解析されており, 変形する際の誤りの修正や, 2D の画像と 3D のモデルの間にある情報量の不足を補うために, その情報を利用している. 一般には単一視点の画像から完全な 3D モデルを復元することはできないが, 候補モデルのもつ構造情報を利用することでそれを可能にしている. また, その構造情報は変形後も保存され, モデルの修正や加工に利用することができる. さらに, 複数の 3D 候補モデルを画像に合わせて変形し, それらをユーザに提示することで, ユーザに対してさまざまなデザインのモデルを示すことができる. このように, 画像と候補モデルから多様なデザインのモデルを作成することで, ユーザの創造性を刺激することができる.

本システムは, 文献 4) で提案したシステムの拡張であり, ジェスチャベースのモデリングインタフェースにクロス&グラフトの考え方を加えたものと位置づけられる. さらに, Xu らの研究[3] と同様に, 発想を支援することを目的の 1 つとし, クロス&グラフトを用いてそれを実現する. ただし, 文献 3) のシステムは入力として画像と 3D 候補モデルを必要とするが, 本システムは何もないところからモデリングを行える, 独立したインタフェースを提供する. 加えて, ジェスチャベースのインタフェースではあるが, プリミティブはオートシェイプにより作成され, クロス, グラフトともにドラッグ&ドロップにより実行されるため, 直感的な操作でモデリングを進めることができる. クロス&グラフトは既存の研究にはない新しい考え方で, これによりモデルに大幅な変更を与える際でも複雑な操作を行う必要がなくなる.

3. アプローチ

クロスは既存のインタフェースにはない新たな手法であり, グラフトはクロスをより有効に機能させるための補助操作である. グラフトによりオブジェクトの接合関係を指定することで, クロスする際にその構造情報を利用して, よりユーザの期待に沿った変化を与えることができる. ここでは, プリミティブや, プリミティブとオブジェクトをクロスしたもの, オ

© 2012 Information Processing Society of Japan

Vol.2012-CG-146 No.30
2012/2/8

ブジェクト同士をグラフトしたものを, オブジェクトと定義する. また, 複数のプリミティブやオブジェクトから構成されるオブジェクトを複合オブジェクトと定義する. プリミティブから複合オブジェクトに対してクロスを行う際には, クロスの効果は構造情報をもとにオブジェクト全体へと伝搬する. 複合オブジェクト同士のクロスの際には, 構造情報をもとにオブジェクト間での各部位の対応を決定し, オブジェクト全体に変化を与える.

3.1 グラフトによる接合関係の指定

上記の通り, クロスの操作に対するユーザの期待にはさまざまな解釈があるが, オブジェクトの構造の観点からその解釈をサポートするものがグラフトである. グラフトは, オブジェクト同士を 1 点で接合する操作であり, また, 一方から他方へと方向性をもつ接合を行う. このように方向性のある接合を行うことで, オブジェクト同士の関係性の情報をクロス結果の決定に利用することができる. この場合, 接合する側のオブジェクトよりも, 接合される側のオブジェクトの方が目的のモデルにおいて本体部分に近いものである場合が多いと考えられる.

図 2 では, 犬の胴体を中心に, 末端から中心に向かって各部位が順にグラフトされている.

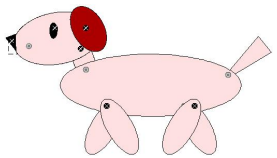


図 2 接合関係の例. 黒い円が接合している点を, 灰色の円が接合されている点を示す
Fig. 2 An example of configuration. A black circle depicts a grafting point, and a gray circle a grafted point.

3.2 クロス結果の伝搬

クロスは, 2 つのオブジェクトから, 両者の特徴を併せもった新たな世代のオブジェクトを生成する操作である. また, クロスされる側が複合オブジェクトの場合, クロスの効果はオブジェクト全体へと伝搬する. グラフトから得られるオブジェクトの構造情報からユーザの期待する結果を推測し, 適切にクロス効果の伝搬を行う. 本システムで扱うクロスの例として, **図 2** の犬のオブジェクトに対してクロスを行った際の変化を**図 3** に示す. この例では, 胴体に対して行ったクロスの効果が足や顔へ伝搬している.

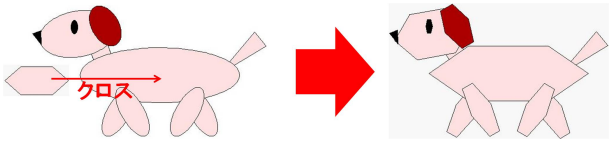


図 3 クロス効果の伝搬例
Fig. 3 An example showing the propagation of crossing effect.

3.3 複合オブジェクト同士のクロス

複合オブジェクト同士のクロス場合, 構造情報をもとにオブジェクト間の各部位の対応を決定し, オブジェクト全体に変化を与える. **図 4** は, 複合オブジェクト同士のクロスの例である. この例では, オブジェクトの各部位の形状や色が同時に変化している.

クロスの結果を決定する方法には, いくつかの考え方がある. 例えば, オブジェクト A とオブジェクト B をクロスする際に, A と B の中間的な性質をもつオブジェクトを生成するという考え方や, A と B のもつデータのうち優先度の高い方を残すという考え方などがある. また, 情報慣性の観点から, ユーザの履歴が多いものほど重要なオブジェクトであり, ユーザの最終目的に近い性質をもつと考えられる. つまり, クロスする側とクロスされる側とを区別することで, ユーザの期待をより正確に予測することができると考えられる. この場合は, クロスする側のプリミティブよりもクロスされる側のオブジェクトの方が目的のモデルに近い性質をもっていることが多いと予想される. 本手法では, これらを総合して規則を定め, その規則からクロス結果を決定する.

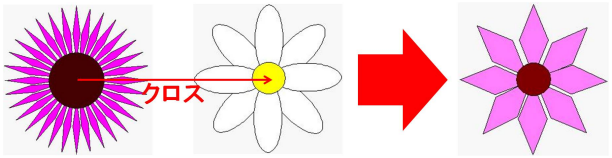


図 4 複合オブジェクト同士のクロス例
Fig. 4 An example of crossing compound objects.

3.4 本手法の特長

クロスでは, オブジェクトのもつデータや接合情報などから, ユーザの期待を推測して結果を決定する. それらの情報を利用することで, ユーザに複雑な操作を要求することなく適

© 2012 Information Processing Society of Japan

切な結果を返すことができる．通常，作成中のモデルの全体に変更を与える場合，ユーザの操作の手間も多くなってしまいがちであるが，本手法では，そのような場合でも変わらず簡単な操作で変更を与えることができる．

また，クロスのもう 1 つの大きな特長として，ときにユーザにとって意外な結果を返す点があげられる．例えば，**図 4** の操作の結果は，実際**図 5** に示すような複数のパターンから確率的に選ばれている．これは，各オブジェクトが表面に表れない遺伝子情報をもっており，それらの遺伝子情報から表現型が決定されるためである．このようにシステムが多様な結果を瞬時に返すことで，ユーザの発想を作業中に拡げることができると考えられる．

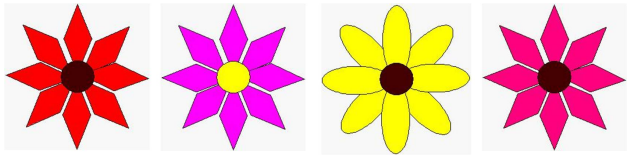


図 5 図 4 でのクロスの結果候補．ここでは全 48 通りある候補中の 4 例だけを示している
Fig. 5 Possible results of crossing in Fig. 4, where only 4 representatives out of all the 48 cases are shown.

4. プリミティブ

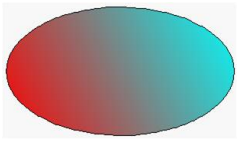
プリミティブはオートシェイプにより作成する．システムインタフェース （**図 1**）の左側にあるシェイプパレットから作成したい形状を選択し，画面内をドラッグすることで軸に沿った長方形の枠が描かれ，その枠に内接するようにプリミティブが作成される．形状や色，テクスチャなどの属性値は，作成時に選択されているものになる．各プリミティブは，**形状**，**色 1**，**色 2**，**テクスチャ**，**テクスチャ方向**の 5 つの属性をもつ．**形状**属性はプリミティブの幾何学的な形の種類を示し，**色 1** 属性はプリミティブを着色する色を示す．2 つの色を用いる**テクスチャ**属性が貼られている場合，**色 1** 属性と**色 2** 属性の 2 色でテクスチャパターンを作り着色する．テクスチャのパターンは**テクスチャ**属性と**テクスチャ方向**属性から決定される．それぞれの属性がとる属性値は**表 1** に示す通りである．

4.1 表現型と遺伝子

プリミティブは属性ごとに 2 つずつ遺伝子をもっており，表面に表れる表現型はこれらの遺伝子から決定される．**図 6(a)** に楕円プリミティブの例を示す．これは**図 6(b)** に示す表現型・遺伝子をもつプリミティブである．各表現型は後述する決定規則に従って，クロスの際に遺伝子をもとにして決定される．

表 1 プリミティブの属性値
Table 1 Atribute list of a primitive.

属性	属性値
形状	三角形，ひし形，五角形，六角形，楕円形，長方形
色 1	R, G, B それぞれ 8 ビット符号なし整数値
色 2	R, G, B それぞれ 8 ビット符号なし整数値
テクスチャ	透明，ドット，縞，グラデーション，無地
テクスチャ方向	周囲 8 方向



(a) プリミティブの見え
(a) Appearance.

属性	表現型	遺伝子 1	遺伝子 2
形状	楕円	楕円	長方形
色 1	赤	黒	赤
色 2	シアン	青	緑
テクスチャ	グラデーション	無地	グラデーション
テクスチャ方向	右上	上	右

(b) 表現型と遺伝子
(b) Phenotype and genes.

図 6 楕円プリミティブの例
Fig. 6 An example of ellipsoid primitive.

4.2 表現型の決定規則

表現型は，クロス後に遺伝子としてもつ 2 つの属性値から決定される．**形状**属性では，三角形，ひし形，五角形，六角形，楕円形，長方形の順に優先度を高く設定しており，2 つの遺伝子のうち優先度の高いものが表現型として選ばれる．**色 1**，**色 2** 属性では属性値ごとの優先度は定めておらず，遺伝子 1，遺伝子 2，遺伝子 1 と遺伝子 2 の中間のどれかがランダムに表現型として選ばれる．**テクスチャ**属性では，透明テクスチャの優先度が最も高く，次いでドット，3 番目に縞とグラデーションが同じ優先度で並び，最後に無地，という順になっている．遺伝子として縞とグラデーションの 2 つをもつ場合は，表現型は両者の特徴を併せもつ，滑らかに変化する縞模様になる．**テクスチャ方向**属性は，2 つの遺伝子と，**テクスチャ**属性の表現型との組合せから決定される．

現行システムではこのように規則を定めたが，前述のようにクロスの結果にはさまざまな考え方がある．このように遺伝子構造を用いた規則を定めることも 1 つの解釈にすぎない．

© 2012 Information Processing Society of Japan

Vol.2012-CG-146 No.30
2012/2/8

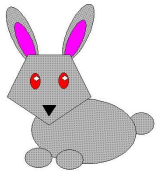
5. グラフト

本システムには, 編集・クロス・グラフトの 3 つのモードがあり, 各モードへはそれぞれに対応するボタンを押すことでいつでも切り替えることができる. グラフトモードでオブジェクトを別のオブジェクトにドラッグ&ドロップすると, ドロップした位置で両者がグラフトされる. グラフトされたオブジェクト同士は, ドラッグによる移動の際には 1 つのグループとしてまとめて移動する. また, この際, グラフトされる側のオブジェクトを “上位”, グラフトする側のオブジェクトを “下位” とする, 接合の木構造における上下関係が生まれる. グラフトは, 接合の木構造を, 新たな接合により繋げる効果がある. クロスの仕様上, 接合関係は木である必要があるため, 複数の下位オブジェクトをもつことはできるが, 複数の上位オブジェクトをもつことはできない.

5.1 構造情報

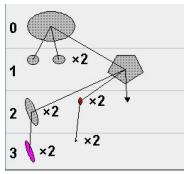
グラフトモードでオブジェクトを選択することで, ユーザはグラフトの際に得られる木構造を表示することができる. 例えば, **図 7(a)** に示すモデルの木構造は図 7**(b)** のようになる. グラフ上では, 全属性の表現型と, 大きさや縦横比のすべてが一致するものはまとめて表示している.

また, 各オブジェクトは, 前述した 5 つの属性それぞれについて**図 7(b)** と同様の木構造をもっている. そこでは, ある 1 つの属性のみに注目し, その属性の表現型及び遺伝子の組が一致するものを 1 つの組としてまとめて扱う. 例えば, **図 7(a)** の形状属性木は**図 7(c)** のようになる.



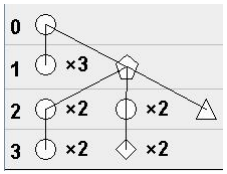
(a) モデル（ウサギ）

(a) A model (rabbit).



(b) (a) の木構造

(b) The tree of **(a)**.



(c) (a) の形状属性木

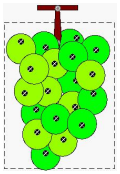
(c) The shape tree of **(a)**.

図 7 グラフトを用いたモデル例（ウサギ）
Fig. 7 An example of grafted model (rabbit).

5.2 離れたプリミティブ同士のグラフト

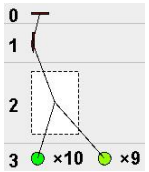
グラフトの仕様上, 離れたオブジェクト同士はグラフトすることができない. また, 重なる部分が小さいオブジェクト同士も, 仕様上グラフトすることが難しい. そこで, そのようなオブジェクト同士のグラフトを実現するために, **テクスチャ**属性値の一種として透明テクスチャを提供する. 透明テクスチャにより透明化したオブジェクトはグラフトモード以外では見えなくなる. これにより, 見た目上離れているオブジェクト同士でも透明化したオブジェクトを介してグラフトすることができる.

図 8 では, 透明化したオブジェクトを介して枝にマスカットの実をグラフトしている. 透明化したオブジェクトを使うことで, 小さな枝にコネクトしやすくなるだけでなく, 共通の上位オブジェクトをもつというマスカットの実同士の関係を簡単に指定することができる. 同階層にあることを簡単に指定できることで, クロスがより使いやすくなる. 詳しくは次節で解説する.



(a) 透明なオブジェクトを使用したモデル

(a) A model with a transparent object.



(b) (a) の木構造

(b) The tree of **(a)**.

図 8 透明なオブジェクトを用いた接合の例
Fig. 8 An example of grafting with a transparent object.

6. クロス

クロスモード時に, あるオブジェクトをドラッグして別のオブジェクトにドロップすることで, それらのオブジェクトがクロスされる. クロスでも交換法則は成り立たず, オブジェクト A をオブジェクト B にクロスした場合とその逆の場合とでは結果が異なる. ある 2 つのオブジェクトをクロスすると, 両者の属性をもつ新たな世代のオブジェクトが規則に従って生成される. オブジェクト A をオブジェクト B にクロスする場合, 生成されるオブジェクトは, 情報慣性の観点から, クロスされる側のオブジェクトである B の性質をより強く引き継ぐ. 具体的にはクロスは B のデータを一部変更することで実現される. 大きさや縦横比,

© 2012 Information Processing Society of Japan

座標, 角度といった基本的なデータは変化させずに, 前述した 5 つの属性のみを変化させる.

ここでは, オブジェクトが複合オブジェクトか否かに着目し, 簡単な例から順にクロスの効果を説明していく.

6.1 1 対 1 のクロス

最も簡単な例は, 単一のプリミティブ同士のクロスである. この場合は特に複雑な操作は行われず, 単純に 5 つの属性の遺伝子を交叉した後, 表現型決定規則に従って表現型が決定される. ここでの交叉とは, 属性ごとに, 各プリミティブから 1 つずつ遺伝子を選択する操作である. 各属性において, 選択された 2 つの遺伝子が新たに生成されるオブジェクトの遺伝子となり, それらから表現型が決定される.

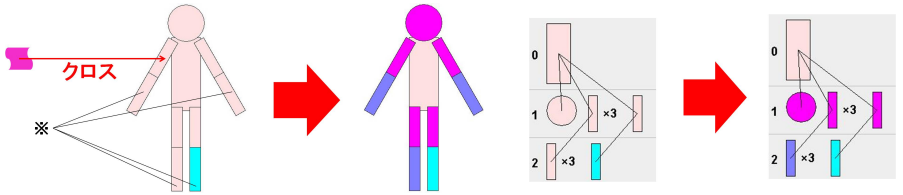
6.2 1 対多のクロス

1 対多のクロスとは, 単一のプリミティブを複合オブジェクトに対してクロスする場合である. あるプリミティブを複合オブジェクト上のあるプリミティブに対してクロスすると, そのプリミティブ同士で 1 対 1 のクロスと同様の操作が行われた後, クロスの効果がオブジェクト全体へと伝搬する. 複合オブジェクト上のプリミティブ A に他のプリミティブがクロスされた場合, A と表現型が一致し, A 以下の階層にあるプリミティブが伝搬の対象となる. また, クロス前に表現型と遺伝子の組がともに一致していたプリミティブの間では, クロス後の表現型と遺伝子の組も同じになるように調整し, 結果が不自然なものにならないようにする. このようにすることで, オブジェクト上の似た位置付けにある部位に対して, 簡単に共通の効果を与えることができる. また, 複合オブジェクトに対するクロスであっても伝搬を行わないようにする操作も提供する.

図 9 に, クロス効果の伝搬例を示す. ここでは, 説明の簡略化のために**色 1** 属性だけについて示している. この例では人型のオブジェクトの腕の部分に対してマゼンタをクロスしているが, その効果が前述の規則により手足の先へと伝搬している. ただし, 図中の※印付きのプリミティブは他のプリミティブと違い, 遺伝子としてシアンの属性値をもっているため, 伝搬の仕方に差が表れている. また, 表現型と遺伝子の組がともに同じプリミティブの間では, クロス後の結果も同じになっていることがわかる.

さらに, 本システムでは, 1 対多のクロスの一種として, ツールバーからユーザの欲しい属性を選んでクロスする機能も提供する. 属性を指定したクロスでは, 交叉の際に選択される遺伝子のうち一方はクロスする属性値になる.

カラーパレット, シェイプパレット, テクスチャパレット, ディレクションパレットの何れかから任意の属性値をドラッグしオブジェクトにドロップすることで, そのオブジェクトの



(a) クロス前後のモデル　　**(b)** **(a)** の木構造

(a) The models before and after crossing.　　**(b)** The trees of **(a)**.

図 9 1 対多クロスの効果伝搬
Fig. 9 The propagation of 1-to-many crossing effect.

(a) クロス前のモデル　　**(b)** クロス後のモデル

(a) The models before crossing.　　**(b)** The models after crossing.

図 10 ツールバーからのクロス.
Fig. 10 Crossing through toolbar.

対応する属性に対してだけクロスを行うことができる. カラーパレットからのクロスの場合, マウスの左ボタンでドラッグした場合は**色 1** 属性に対して, 右ボタンでドラッグした場合は**色 2** 属性に対してクロスが行われる. **図 10** では, **(a)** に示す 2 つのモデルの花弁部分に対して, ツールバーからクロスを行った結果, **(b)** が作成されている. ここで, **(a)** 左側のモデルに対しては, **形状**属性にひし形, **色 1** 属性に赤をそれぞれクロスし, **(a)** 右側のモデルに対しては, **形状**属性に六角形, **色 1** 属性にマゼンタ, **テクスチャ**属性にグラデーションをクロスしている.

6.3 多対多のクロス

多対多のクロスとは, 複合オブジェクト同士のクロスを示す. 本システムでは, 限られた条件のもとで, 多対多のクロスを実現している. 多対多のクロスは, 次の 3 つのステップで実行される.

1 クロスの可否判定

まず, 前述したオブジェクトの木構造をもとに, クロス可能かどうかを判定する. 5 つ

© 2012 Information Processing Society of Japan

の属性に対応する木と，全属性を合わせた木の計 6 つの木を両オブジェクト間でそれぞれ比較し，すべての属性においてクロス可能と判定された場合，続いてクロスを実行する．この判定では, 前述のように表現型と遺伝子の組がともに同じとしてまとめられているプリミティブは，1 つの組として扱う．対応する木同士の長さやノード（組）の個数を比較して, 長さが一致し，かつクロスする側のオブジェクトの最下層のノードの個数がクロスされる側の最下層のノードの個数よりも少ない場合，クロス可能と判定する．また，木の長さが異なる場合は, クロスされた箇所をヒントに部分木同士で同様の比較を行い, そこで交配可能と判定された場合はその部分木に対応する箇所だけをクロスする．どちらにも当てはまらない場合は，それらのオブジェクト間ではクロスは実行できない．

2 対応付けの決定

クロス可能と判定された場合, 続いて対応付けを決定する．多対多のクロスを行うためには，両オブジェクトのどの部分とどの部分が対応するのかを決める必要がある．対応付けは上の階層から順に進められ, 対応関係が明確な箇所はシステムが自動で対応付けを決定し, そうでない場合はユーザの入力を要求する．システムは, その階層のノードの個数とノードを構成するプリミティブの個数から，対応が一意に定まる場合に対応付けを行う．

3 クロスの実行

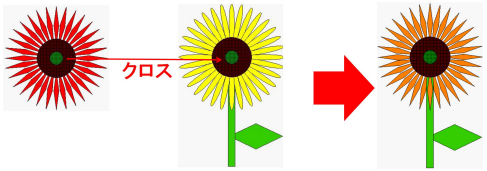
最後に，2 つ目のステップで決定された対応付けをもとにクロスを実行する．まず対応する箇所ごとに 1 つずつプリミティブを選んで, その後は 1 対多のクロスと同様に, クロスを行い, 各ノード内で結果が同じになるように調整する．**図 11** に多対多のクロスの例を示す．この例では, 全体の階層の長さが異なるため，花の部分だけをクロスしている．

7. モデリング例と評価

7.1 結　　果

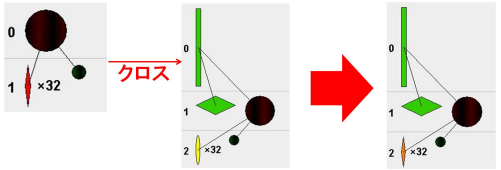
本システムは, 開発ソフトウェアとして eclipse を使用し, Java 言語の swing を用いて開発している．

多様なデザインのモデルを作成する例として, 本システムを使用して著者が作成したモデルを**図 12** に示す．ここでは, 花のモデルを 10 種類作成することを目的としてモデリングを行った．作成手順は次のようになる．まず, プリミティブの作成とグラフトを繰り返し, もと



(a) 複合オブジェクト同士のクロス

(a) Crossing a compound object with another compound object.



(b) **(a)** の木構造

(b) The trees of **(a)**.

図 11　多対多のクロス．構造が一致する花の部分だけが変化している

Fig. 11　Crossing a compound object with another compound object. Only the portion with a similar structure has been changed.

となる花を 2 種類作成した（**図 10(a)**）．次に, それぞれの花弁部分に対しツールバーからクロスを行い, **形状**属性, **色 1** 属性, **テクスチャ**属性の値をそれぞれ変化をさせた（**図 10(b)**）．ここで, ツールバーからのクロスにより, この 3 つの属性における遺伝子 1 と遺伝子 2 がもつ属性値が別のものになっている．最後に, 両オブジェクト間で互いにクロスを繰り返すことで, **図 12** に示す 10 種類を作成した．**図 10(a)** の作成には合計で 15 分かかったが, その後**図 12** のモデルを作成するまでには 1 分ほどしかかからなかった．作成時間のほとんどはユーザの対話時間である．

7.2 評　　価

本システムの評価を行うために, 著者自身がオートシェイプで同じようなモデルを作成し, かかった時間を比較した．**図 10(a)** に似たモデルを作成するまでには 15 分程度を要し, 本システムを用いて作成した場合とほぼ同じ時間がかかった．それから**図 12** のような 10 種類のモデルを作成するまでには, さらに 1 時間程度の時間を必要とした．このように, デザインの異なるモデルを多数作成するためには, 通常多大な時間と手間を必要とするが, 本システムでは, そのような時間と手間ほとんど必要とせずに, さまざまなデザインのモデルを作成することができる．

© 2012 Information Processing Society of Japan

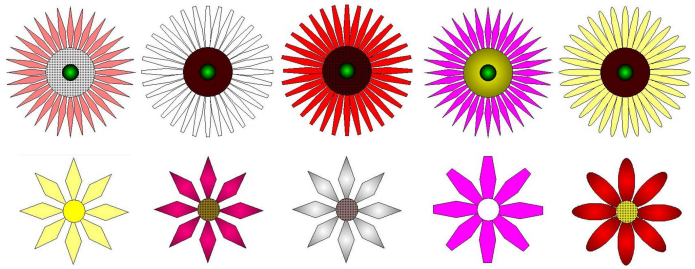


図 12 モデル作成例．デザインの異なる 10 種類の花を作成した
Fig. 12 A variety of crossbred flowers. Ten different designs have been generated.

8. 結論と今後の課題

8.1 結　　論

本システムでは, 2D ではあるが, クロス&グラフトのアイディアによりオブジェクト全体に変化を与える操作を簡単に実現することに成功した．特にツールバーからのクロスは使い勝手がよく, ユーザが意図した通りにオブジェクトを変化させたい場合に有効に機能した．また, 構造が完全に一致するオブジェクト間のクロスでは，簡単に多様な変化を実現できた．限られた条件のもとではあるが，簡単な操作で多様なデザインを示すことで，ユーザの発想に対する支援を行う道筋をつけたといえる．

クロスは，正確なモデルを作る場合ではなく，漠然と作成する対象を決めた後，モデリングを行いながら詳細なデザインを決めていくような場合に, より有効に機能する操作である．逆に，作成するモデルの形やデザインを正確に決めてからモデリングを行う場合は，ユーザが大幅な変化を必要としないため，クロスを使う機会は少ない．また，操作後の変化が予想しにくいことも，そのような場合にクロスを使いづらい理由の一つとなっている．

8.2 今後の課題

今後の課題としては，まず, 各機能を拡張し，より使いやすいインタフェースにすることがあげられる．特に，多対多のクロスの際の対応付けを決定するステップでは，対応付けをユーザの指定に頼ってしまうことが多く，入力の手間が増えてしまっている．クロスの際にユーザの入力を何度も要求してしまうと，一瞬で変化を与えられるというクロスの長所が失われてしまう．現在，システムが対応付けを判断する際には木構造に現れる接合関係だけを用いているため，それに加えて接合している位置や各プリミティブの大きさ，縦横比などの幾何学的な特徴を利用すれば，ユーザの入力の手間を減らすことができるであろう．

次に取り組むべき課題に，交配モードの改良があげられる．現在の各属性の木を用いた手法では，クロス可能な対象は構造が似ているオブジェクトに限定されてしまい，それ以外のオブジェクトに対してはクロス自体が行えない．現在の手法は構造が大きく異なるオブジェクト同士をクロスするのには不向きであり，そのようなクロスを実現するためには，新たな原理を採り入れる必要がある．そこで，新たな交配モードとして，オブジェクト全体がもつ幾何学的な特徴をもとにクロスを行う方法を現在検討中である．幾何学的な特徴とは，オブジェクト全体に占める色の割合や，各プリミティブの接合位置，大きさなどである．この交配モードが実現できれば，花を動物にクロスし，花のような模様をもった動物を生成する，といった操作も可能になり, ユーザはより自由にクロスを行うことができる．

また, 本システムでは属性値の優先度を一意に定めていたが, 属性値の優劣は本来は一意に定まるものではなく, ユーザが作成したいモデルによって変わるものであると考えられる．そこで, 現在, ユーザが作成したいモデルのイメージに合わせて属性値の優先度を決められる操作を検討している．例えば, ユーザが “暖かみのあるモデル”, “インパクトのあるモデル”などのイメージを指定することで, 優先度を変更する操作を考えている．他にも, ユーザが特定の属性値の優先度を低く設定し, 好ましくない特徴を淘汰するような操作も検討中である．

最後に，3D での実装も課題として残っている．近年のモデリングは 3D が主流であり，インタフェースとしての実用性を考えると, この実装は必須といえる．

謝辞　本研究の一部は，平成 23 年度科学研究費補助金基盤研究 (B)20300033 により実施された．

参　考　文　献

1) Zeleznik, R. C., Herndon, K. P. and Hughes, J. F.: SKETCH: An Interface for Sketching 3D Scenes, *SIGGRAPH 96 Conference Proceedings*, pp.163–170 (1996).
2) Igarashi, T., Matsuoka, S. and Tanaka, H.: Teddy: A Sketching Interface for 3D Freeform Design, *SIGGRAPH 99 Conference Proceedings*, pp.409–416 (1999).
3) Xu, K., Zheng, H., Zhang, H., Cohen-Or, D., Liu, L. and Xiong, Y.: Photo-Inspired Model-Driven 3D Object Modeling, *ACM Transactions on Graphics*, Vol.30, No.4, pp.80–80 (2010).
4) 八坂　俊，大野義夫，藤代一成：交配に基づく 2 次元形状モデリング，Visual Computing/グラフィクスと CAD 合同シンポジウム DVD 予稿集 2011, No.47.

© 2012 Information Processing Society of Japan