

# 切る

## Cut

大和田 茂      赤保谷 鮎美      Frank Nielsen      楠 房子      五十嵐 健夫\*

**Summary.** We propose a system that enables the user to cut 3D food models using a knife controlled by standard mouse operations. Cut is one of the most primary interactions with foods and has a long history. Therefore, there are a great number of artistic cut patterns. However, previous computerized cut interactions, which have mainly been explored in the context of volume visualization, are not sufficient for such complicated cuts. We designed a new cut interface by which the user can easily create artistic food models using simple mouse operations. Our system allows the user to cut along a surface of an object and also a part of a model is deformed, if desired. We show various models created by only cutting an original food model.

## 1 はじめに

三次元CG(コンピュータグラフィックス)技術の進歩は、多くの独創的なコンテンツを生み出してきた。そして、その仮想的空間とインタラクションを行うためのユーザーインターフェースも数多く提案されてきた。しかしながら、通常ユーザーが用いるのはマウスに代表される二次元デバイスであり、三次元空間と直観的にインタラクションすることは一般に難しい問題である。

我々は「食」をテーマとした、新たなCGコンテンツを製作している。このプロジェクトでは、ユーザーにコンピュータ上で料理をしてもらい、その過程で食材が動的に変化していく様子を楽しんでもらうことを目標としている。ここで必須となるのが、食材を切断するためのインターフェースである。

三次元物体を切断するインターフェースは、これまでは主にポリウムデータ可視化の分野で研究されてきた。なぜなら、物体を切断することはポリウムレンダリングと並んで、内部構造を把握するための最も基本的なインタラクションだからである。しかしながら、これまで提案されてきたシステムには、切断面が平面に限られていたり、平面以外の形状は外部からロードしなければならなかったり(つまり、対応するユーザーインターフェースがない)[1]、特殊な入力デバイスを必要とされるなど[2]、様々な欠点が存在する。食材の切断には長い歴史があり、また、それは極めて日常的に行われるインタラクションであるため、多くの手法が存在し、一種の芸術性を持つ

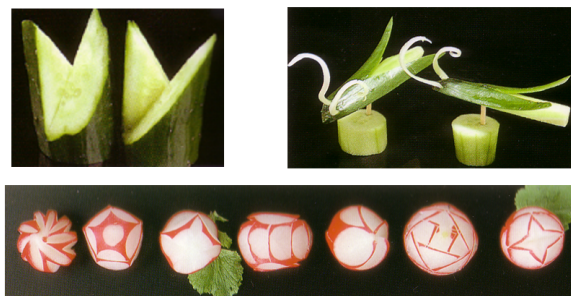


図 1. 複雑な切断の例

までになっている。このような切断を行うには、既存システムでは不可能である(図1)。

そこで我々は本稿において、いくつかのインターフェース上の工夫により、マウスだけでも多様な切断を行うことができることを示す。本手法ではユーザーがコンピュータ内のナイフを動かすことにより、より現実に則した直観的な操作を行うことができる。キーとなるアイデアは、物体表面に沿う切断、ナイフを途中で止めること、そして、物体の変形である。我々はこれらのアイデアを、実際の料理で用いられるナイフの動きを調査することから得た[3]。これまでのところ、食材の切断にしか応用していないが、同様の手法はポリウム可視化においても適用できると考えている。

## 2 ユーザーインターフェース

まず、ユーザーは自由曲線を用いて物体を切断することができる[4](図2)。

本稿で新たに提案するカット法では、明示的にナイフを表示して様々な切り方を可能にしている。この手法では一つの切断操作の自由度が高いため、これをいくつかのステップに分け、以下に順次説明す

\* Shigeru Owada, 東京大学大学院 情報理工学系研究科/Sony CSLabs,Inc., Ayumi Akaboya, 多摩美術大学 情報デザイン学科, Frank Nielsen, Sony CSLabs,Inc., Fusako Kusunoki, 多摩美術大学 情報デザイン学科, Takeo Igarashi, 東京大学大学院 情報理工学系研究科, 科学技術振興機構 さきがけ

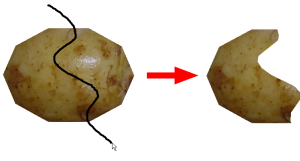


図 2. 自由曲線による切断

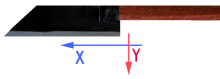


図 3. ナイフの局所座標系

る。以後の説明の便宜上、ナイフに局所座標系と定義する。柄の方向を  $x$  軸の正方向、刃が向いている方（進行方向）を  $y$  軸とする（図 3）。

## 2.1 ナイフを置く

ユーザーはまず、切断したい物体の上にナイフを配置する。これには二通りの方法がある。物体上でマウスを左クリックする方法と、物体上で左ボタンを長押ししたのちにドラッグという方法である。

一つ目の方法では、単に物体上でマウスを左クリックする。するとナイフが現れ、ナイフの  $x$  軸が上向き、 $y$  軸が視線方向（奥行き方向）に向くように置かれる。置かれたのがナイフであることをわかりやすくするために、アニメーションが付加される（図 4）。

二つ目の、長押ししたのちにドラッグという方法では、ドラッグ時にナイフの方向を決められる。この時に、物体表面に対するナイフの刃の方向がわかりにくくなるので、 $xy$  平面が半透明表示される（図 5）。ボタンを離せば、ナイフの方向が確定する。

## 2.2 ナイフを動かす

次に、ナイフを左ドラッグして動かす（図 6）。前ステップで左クリックによりナイフを置いた場合には、右ドラッグで物体をナイフごと物体を回転させてからドラッグする（さもないと、視線と平行な方向にナイフを動かさなければならなくなる）。

ナイフが進む方向には、内部的に二つのモードが存在する。ナイフの  $y$  軸方向を奥行きとしてシーンを見た場合、刃が物体の端近くに置かれている場合には皮むきモードとなり、そうでなければ通常の切

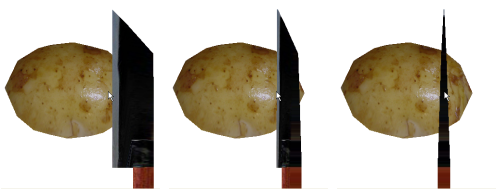


図 4. ナイフを置く方法 1

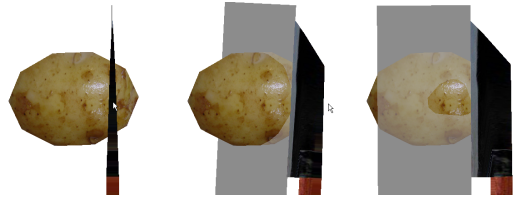


図 5. ナイフを置く方法 2

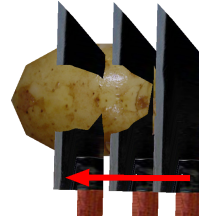


図 6. ナイフのドラッグ

断モードになる（図 7）。通常切断モードでは刃は  $y$  方向に直進するが、皮むきモードでは物体の表面に沿って動く（図 8）。しかし、どちらの場合にも切断中に  $x$  軸方向は変化しないようになっている。

## 2.3 変形する (Optional)

必要に応じて、切断した物体を変形することができる。切断中に（図 9(a)）ボタンを押したままナイフを一定時間停止させると（図 9(b)）、それまでにナイフが通った部分に実際に切り込みが入り、変形モードに入る（図 9(c)）。この状態でマウスをドラッグすると、ナイフによって切り込まれた部分が反り返るように変形される（図 9(d)）。マウスを刃の進行方向にドラッグした場合には手前側が、逆の場合には遠くの側の面が変形される。図 9(e) は、図 9(d) で変形された物体を別の角度から見たものである。

## 2.4 切断の終了

マウスの左ボタンを離せば、切断が終了する。結果の物体が複数のパーツからなる場合には、片方が捨てられる（図 10）。皮むきモードの場合は、薄い方（皮）が自動的に捨てられる（ただし、変形操作を行った場合には物体は二つに分かれないため、捨てられない）。また、すでに述べた自由曲線による切断の場

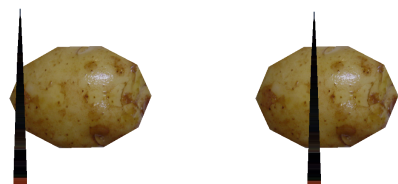


図 7. 皮むきモードになる場合（左）と通常モードになる場合（右）

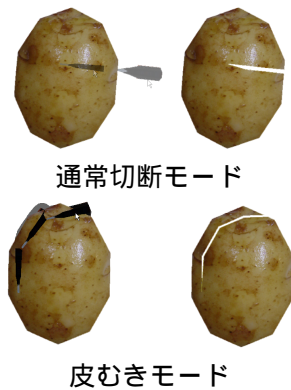


図 8. モード別の刃の動き (ナイフの先端方向から見た場合)

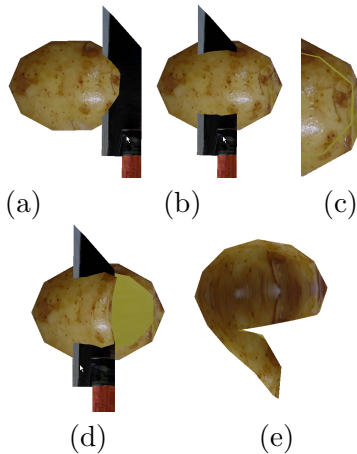


図 9. 変形操作

合は、ストロークの左側が捨てられる [4].

### 3 実装

実装は C++ と OpenGL を用いて行った。レンダリングは三角形ポリゴンメッシュによって行われ、切断には CSG [5] を用いる。断面画像を生成するアルゴリズムは本稿の主題ではないので何を用いてもよいが、我々の実装では Volumetric Illustration [6] をベースにモデルを作成した。原論文は二次元のテクスチャ合成技術を応用し、あたかも中身が詰まっているかのような切断面テクスチャを動的に生成する手法を提案している。我々は物体が複数回切断さ

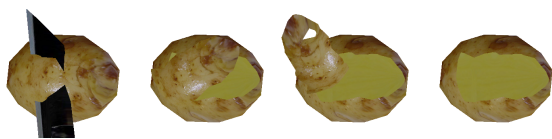


図 10. パーツの破棄



図 11. 結果 (上二段は実物体との比較. 左が実物体.)

れた際にも、前回に生成された断面と一貫性を持つように拡張を行った。

物体の変形アルゴリズムとしては、まずユーザーが動きを止めた時点でのナイフの刃を中心軸として、新たに生成された切断面上の頂点を回転変換する。この時に軸から遠いほど大きく回転させる。これにより、図 9(e) のようにめくれ上がるような変形を行うことができる。次に、その変形された断面形状に応じて、それ以外の頂点の位置を変更する。これには、各頂点に対して (変形前の) 切断面上での一番近い点を調べ、その点の変形結果をそのまま適用することとする。

皮むきモードでのナイフのモーションパス計算については、Appendix にアルゴリズムを載せた。

### 4 結果

このシステムを用いて、図 11 に示したような形状が簡単に製作できた。参考のため、キュウリとラディッシュの例については実際に食材を切った結果も併せて載せた。また、我々のカットインターフェースを用いたコンテンツも開発中である (図 12)。図 12 左は、Flash と協調動作する教育用コンテンツで、小学生を対象に食育 (食事の教育) を行うためのものである。また、図 12 右は、野菜切断ゲームである。ウィンドウ内左上に表示されている時間内に左下の形状を作ることが求められる。このようなゲームは、我々の切断システムの操作法を短時間で覚えるためにも役立つ。

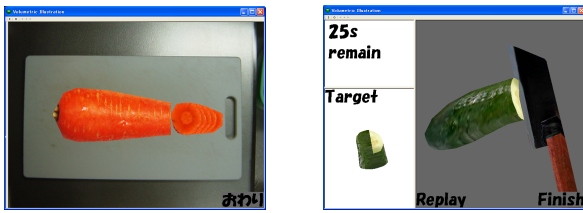


図 12. 本インターフェースを用いたコンテンツの例

## 5 まとめと今後の予定

本稿では、マウスを用いて自由に食材を切断するユーザーインターフェースを提案した。物体形状に沿った切断、ナイフを途中で止める、物体の変形を行う、というアイデアにより、物体切断のバリエーションが増加した。今後はこのインターフェースを拡張して、CT スキャンデータなどの、より一般的なモデルの切断変形を行う研究を行いたいと考えている。

## 6 謝辞

本研究の一部は、文部科学省科学研究費補助金(基盤研究(C)(2) 16500311)による補助によるものである。また、研究に対する有用なコメントを頂いたアーク情報システムの佐々木孝氏に感謝する。

## 参考文献

- [1] D. Weiskopf et al. Volume Clipping via Per-Fragment Operations in Texture-Based Volume Visualization. Proc. IEEE Visualization, pp.93-110, 2002.
- [2] K. Hinckley et al. Passive Real-World Interface Props for Neurosurgical Visualization. Proc. CHI, pp.452-458, 1994.
- [3] 材料の下ごしらえ百科, 主婦と生活社
- [4] T. Igarashi et al. Teddy: A Sketching Interface for 3D Freeform Design. Proc. ACM Siggraph, pp.409-416, 1999.
- [5] C. Hoffman. Geometric and Solid Modeling. Morgan Kaufmann Pub. 1989.
- [6] S. Owada et al. Volumetric Illustration: Designing 3D Models with Internal Textures. Proc. ACM Siggraph, pp.322-328, 2004.

## A Appendix: 皮むきモードでのナイフのモーションパス計算法

実装上問題となるのが皮むきモードでのナイフの動きである。基本的には物体の表面に沿うように動かせばよいのだが、細かく見ると、まずナイフは物体表面からある深さまで動き、次にその深さを維持するように動き、最後にまた表面まで動かさなければならない。このために、まず我々は次のアルゴリズムを用いて二値画像を作成する。

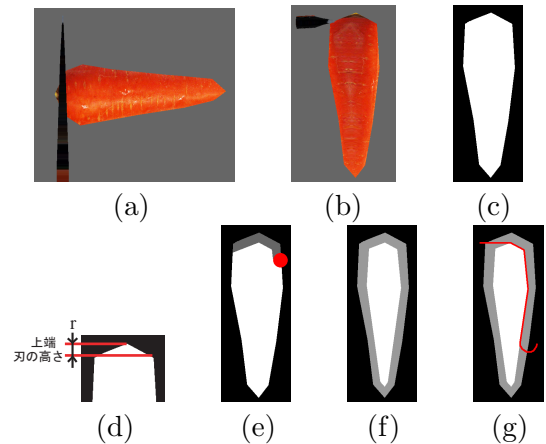


図 13. 皮むきモードでのナイフパス計算

1. ナイフが置かれたら(図 13(a)), ナイフの  $x$  軸が手前方向,  $y$  軸方向が右方向となるように、シーンを正射影でオフスクリーンレンダリングし(図 13(b). 実際にはナイフは除く), 結果を背景と前景に二値化する(図 13(c)).
2. この画像上における前景の上端と下端の縦方向座標のうち、ナイフの位置により近い方を選び、その位置の差を  $r$  とおく(図 13(d)).  $r$  がある閾値より大きければ通常の切断モードであり、刃は直進するだけである。そうでなければ 3 に進む。
3. 二値画像を、半径  $r$  の円で erosion<sup>1</sup>する(図 13(e,f)). 二値画像の前景は、 $r$  だけ縮む。

この二値画像をナイフが置かれた時に計算し、ユーザーがナイフをドラッグした時には、刃がこの図上で前景の上端あるいは下端(ステップ 3 で選ばれた方)に達するまでは直進し、達したら前景の縁をなぞるように刃を動かせばよい(図 13(g)).

ユーザーがマウスボタンを離したら、パスの残りを生成しなければならない。ナイフは erosion されたあとの表面に沿って動いているので、ユーザーがボタンを離した時点では、刃は物体内部にあるはずである。ここで大切なのは、ナイフが物体を確実に切り離すことと、ユーザーの意図しない不連続な表面が現れないことである。そこで、まずは前述の二値画像上でとりあえず刃をそのまま直進させてみて、ナイフが物体の外に出るかどうかを調べる。我々は  $4r$  程度まで探索を行っている。これで物体の外に出るようなら、この動き(最後は直進させる)を採用する。外に出ないようなら、パスの最後に、物体の外方向に向かう半径  $r$  の円弧を付け加える。パスの最後でこの円弧を物体の外に出るまで進めれば、スムーズかつ物体を切り離すパスが生成できる(図 13(g)).

<sup>1</sup> モルフォロジーオペレータ