

剛体シミュレーションエンジンを統合したキネティックアート・エディタ

古田 陽介 Yohsuke FURUTA

三谷 純 Jun MITANI

五十嵐 健夫 Takeo IGARASHI

福井 幸男 Yukio FUKUI

概要: モビールやヤジロベエなどのキネティックアートを新たに創作するためには、現状では現実の素材を用いて幾度もの試行錯誤を行う必要があり大きな労力を必要とする。コンピュータを使えば素材を組み合わせた際の動きをシミュレーションで求めることが可能であるが、既存の CAD と CAE のシステムは互いに独立してデータの交換に手間がかからってしまうという問題点がある。そこで我々は 3D CAD システムに物理エンジンを統合し、形状のモデリングと行うと同時に設計の妥当性の検証を自動的に行い、その結果をわかりやすくユーザに提示するというシステムを考案し実装した。本システムを用いることで、現実の素材を用いた試行錯誤を行わなくても必要とする動きを容易に得られるようになり、形状を構築する作業を効率的に行なうことが可能となった。

キーワード: CAD・CADD／シミュレーション／キネティックアート

1. はじめに

アートの分野の一つにキネティックアートがある。これは風力や水力、あるいは観覧者などの力が加わることによって形が変わったり動いたりする作品を対象とした美術のジャンルのことである。ヤジロベエ(図1)やモビール(図2)はその代表的なものであり、多数の個性的な作品が発表されている^[1-4]。また、キネティックアートは「動く彫刻」ともよばれ、20世紀初頭からさまざまな動きを作品に取り入れるという試みがなされてきている。しかし、このような物理的に「動く」物体のデザインを行う際は多数の物体同士の相互作用を考慮しなければならず、

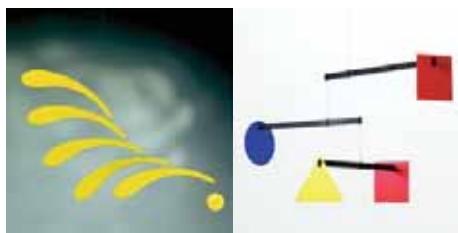


図1 モビールの例^[1-2]



図2 ヤジロベエの例^[3]

完璧に意図通りのものを作ることは非常に難しい。本論文では、そのような作品をつくるのに適したコンピュータシステムを提案する。

コンピュータを使って物理的なものを設計するということは非常にポピュラーであり、とくに CAD システムは物作りの専門家にとってなくてはならないものである。また、近年では非専門家であっても形を作りやすいインターフェースやインタラクション技術を盛り込んだ CAD システムの研究が盛んに行われるようになってきている。さらに、最近では光造形法や粉末法を用いた三次元造形装置の低価格化と高性能化により、自分がデザインしたものを実際に手にすることができる環境が整いつつある。

しかし、既存の CAD システムでは形を作ることができるだけで、その挙動をコンピュータが評価するような機能は備わっていない。そのような用途には別途 CAE のシステムを用いてシミュレーション解析を行う必要があり、その結果を受けて再度 CAD 上で形状の修正を行うというプロセスを何度も行う必要がある。また、多くの美術作品は最初から完成型が決まっていることはまれであり、ほとんどの作品は実際にものを作りながら、度重なる試行錯誤や場合によっては完全な偶然によって形が生み出されたものである。そのようなプロセスは創作活動の初期の段階においては非常に重要なものであるが、上記の理由により、既存の CAD/CAE システムはそのような使用には適していない。

そこで、本論文ではそのような動きのある作品の制作を効率的に行なうために、三次元 CAD システムに物理シミュレーションを取り入れたシステムを提案する。両者を組み合わせ、モデリングを行うと同時にその動きを瞬時に計算し提示することで、目的となる動きを得るためにユーザが現物でなんども試行錯誤を繰り返す必要がなくなる。その際、オブジェクトを単純に時間軸に沿って動かして提示するのではなく、すべての動きを一つのフレームに圧縮して提示することでその動きを一目で理解しやすくてある。そのために必要となる、インタラクティブな操作に適したシミュレーションのアルゴリズム、およびユーザへの提示方法の詳細について本論文で述べる。

2. 関連研究

CG の技術を使って実際の物作りを行うというアプローチにはさまざまなもののが提案されている。三谷らは与

えられた三次元モデルからペーパークラフトの展開図を作成するシステムを開発している^[5]. また Weyrich らは三次元モデルから浮き彫りを自動的に生成するシステムを開発している^[6]. 他にも, 森らによるぬいぐるみの型紙を作成するための研究^[7]など, 物理的な物体を対象とした特別製のモデリングシステムがいくつか提案されている. 我々のシステムは物理シミュレーションを用いてキネティックアートの設計を行うものであるが, 計算機を用いて現実のオブジェクトの設計を行う際の支援を行うという目的は共通するものである.

剛体物理シミュレータは, 今日ではゲームの世界で一般的なものとなっている. また, 近年ではシミュレーションエンジンを形状モデリングと組み合わせて利用しているシステムがいくつか提案されている. スケッチベースのモデリングシステムに剛体エンジンを組み込んだ例としては, ASSIST^[8]や Phun^[9]があげられる. これらのシステムでは, ユーザがモデルを作成すると即座にシミュレーションが実行され, その結果にそってオブジェクトがアニメーションする. また, プロメテックソフトウェアはモデリングインターフェースに粒子法解析による流体シミュレーションを統合した PhysiCafe を開発している^[10]. しかしながら, これらのシステムでは結果を一連のアニメーションとして提示するため, それが完了するまでユーザは待たなくてはならない. 我々のシステムではシミュレーションとモデリングがより密接に統合されており, ユーザがモデルをドラッグしている最中にもシステムはバックグラウンドでシミュレーションを行い, その結果を常にユーザに提示しつづけることが可能である.

剛体物理シミュレーションによる結果をユーザに提示したまま, その結果を直接編集することが可能なものとしては, Popović らによる CG アニメーション生成のための

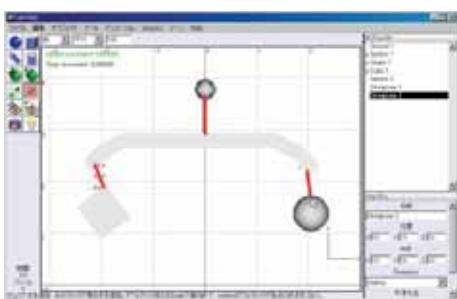
システムがある^[11]. このシステムではユーザは対象となるオブジェクトの状態を対話的に変更することができ, それに合わせてシステムがオブジェクトの初期値や速度といったパラメータを自動で最適化する. 我々はこのシステムのコンセプトと物体の挙動の可視化の手法を参考にしている. また, Justin Needham らはボールの軌跡を予測するデジタルアシスト・ビリヤードシステムを提案している. このシステムではボールの場所とキュの向きを画像解析で求め, ボールを打った際の動きをシミュレーションしその動きをプロジェクトで台に投影するというシステムを提案している^[12].

3. システムの概要について

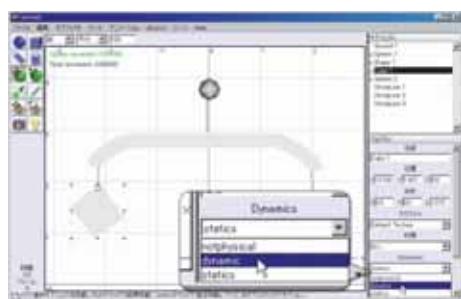
本システムはモデルの生成という一般的な CAD ツールとしての機能に加え, 構築されたオブジェクト群の挙動のシミュレート, およびその結果の図示という機能を備える.

図 3 は本システムを用いてモデルのデザインを行う例である. 本システムではモデル形状の構築のためのインターフェースに「Art of Illusion 2.6.1 (AOI)」^[13]のソースコードを利用している. AOI は GPL の元でオープンソースとして公開されている三次元モデリングソフトであり, 球, 直方体, または閉じた三角形メッシュといった基本的な形状を対話的に構築する機能を備えている. 我々はそれに加えて二つの形状を糸で接続する機能を実装し, モバイルなどをデザインできるように拡張してある. また, 剛体シミュレーションには「PhysX 2.8.1」^[14]を用いている. ネイティブコードで記述された PhysX の API を Java で記述された AOI から呼び出すため, JNI (Java Native Interface) を利用している.

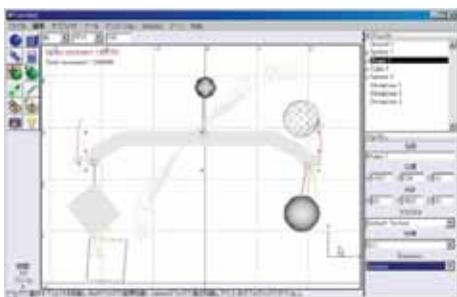
モデリングシステムと剛体シミュレータを密接に統合



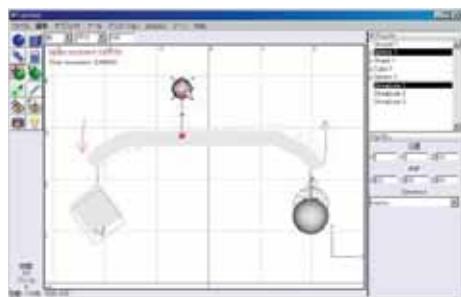
1. 形状を生成し, 互いを糸で結ぶ



2. 形状を dynamic 属性に変更する



3. シミュレーションを開始



4. シミュレーションの結果を見ながら位置を修正

図3 デザイン手順の例

するために、各オブジェクトには速度や密度、シミュレーションモードなどの情報が付加されている。シミュレーションモードは「dynamic」「static」「non physical」の三通りのうちのいずれかの値をとる。動的(dynamic)なオブジェクトはシミュレータによって挙動が計算され、その結果がユーザに提示される。静的(static)なオブジェクトは動くことはないが、他の動的なオブジェクトに対して衝突などの影響を及ぼす。Non physicalなオブジェクトはシミュレーションの対象にはならず、オブジェクトを一時的にシーンから外したりする際に利用する。他にも、オブジェクトの形状に合わせて表1のような属性を持つ。また、シミュレーションの正確さと計算完了までに要する時間を決定する重要なパラメータとして、シミュレーション間隔というパラメータが存在する。この大きさがどのように決定されるかは後の節で述べる。

オブジェクトの属性と位置は、JNI経由でモデルからシミュレータに転送され、規定の回数(=n)の反復計算が行われた後に、各時刻におけるオブジェクトの位置がn個の配列に格納され呼び出し元に返される。もしシミュレーション中に衝突が発生した場合は、その座標も配列として取得することが可能である。

表1 形状と属性

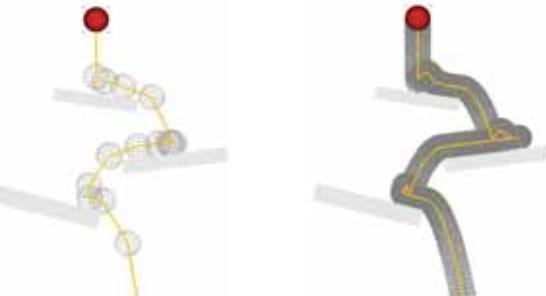
形状	属性
球	半径
直方体	幅、高さ、奥行き
平らな板	厚み、輪郭を表す二次元の頂点リスト
三角形メッシュ	三次元の頂点リスト、頂点インデックス

3.1. 反応性の高い物理シミュレーション

一般的に、物理シミュレーションは非常に時間を要する計算である。また、シミュレーションの初期化の際にも形状の凸多角形分割や凸包の計算、頂点インデックスの再割り当てなど、さまざまな処理を行う必要があり、それにも大きな計算コストが必要となる。

そこで、本システムではそれらを削減するために、シーンが変化した際、形状が変化したオブジェクトのみを初期化し、またユーザの操作に合わせて、動的にシミュレーション間隔を変化させるといった高速化の手法を取り入れている。後者については、ユーザがオブジェクトの位置を大きく変化させている最中はあまり正確な位置調整を行っている状態ではなく、そのためシミュレーションの正確さよりはシミュレーションの完了までに要する時間の短さの方を重要視し、近似解を高速に提示した方が良いという考えに基づくものである。反対に、ユーザがオブジェクトをほとんど移動させていない場合は正確な位置を探るために微調整を行っている段階であると考えられ、そのためより正確なシミュレーションが必要であるということを意味する。もしすべてのオブジェクトに対

して何の変化もないまま規定の回数のシミュレーション計算が完了した場合、時間間隔を1/2倍して再度最初からシミュレーションを開始する。もしオブジェクトに対して何らかの変化が加えられた場合、時間間隔を2倍してシミュレーションをやり直す(図4)。一連のパラメータ変更はシステムが自動的に行うため、ユーザはモデリングの作業に集中することが可能である。



**図4 動的なシミュレーション間隔の変化の例。
左: 間隔が長い場合。右: 間隔が短い場合**

3.2. シミュレーション結果の提示

上記の処理によって得られた位置情報をどのように可視化するかについて考える。図6のように、すべての場所にワイヤフレームでオブジェクトを描画した場合、シミュレーション時間が長かったり多数のオブジェクトがシーンの中に同時に存在していたりすると、画面が非常に煩雑になってしまい、状態を把握できなくなってしまう。そういう事態を避けるため、位置情報の集合から動きの変化した場所 d_t をもとめ、その場所のみにオブジェクトを描画することで動きの様子を把握しやすくなる。

d_t は以下の式によって求める

$$\begin{aligned} v_t &= M_t p - M_{t-1} p \\ a_t &= v_t - v_{t-1} \\ d_t &= a_t \cdot a_{t-1} \end{aligned}$$

M_t はシミュレータによって求められた、時刻 t におけるオブジェクトの位置情報を表すアフィン行列である。 $p = (x, y, z)$ はオブジェクトの中心座標である。 $d_t < 0.0$ のときに位置情報に変化があったとみなし、その時の位置にオブジェクトを描画する(図7)。

また、よりオブジェクトの挙動を把握しやすくするため、中心の移動の軌跡もあわせて描画する。同時に、オブジェクトモデルのすべての頂点について合計の移動距離を求め、最大となった点の軌跡も描画している(図8)。加えて、衝突があった際にはその座標に赤い点を表示している(図5)。これらの手法を組み合わせることで、オブジェクトが回転しながら移動する様子も視覚的に把握することが可能となる。



図5 衝突点を赤で描画

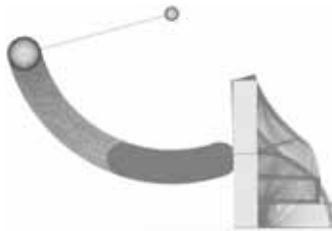


図6 シミュレーションで求められたすべての位置にワイヤフレームでオブジェクトを描画した場合

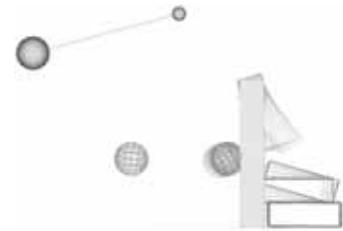


図7 動きに変化のあった場所のみに描画した場合

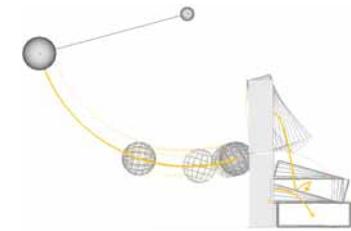


図8 図7にオブジェクトの移動の軌跡を描画した場合

3.3. 動きの少ないオブジェクトの動きの可視化

上記の表現はオブジェクトが大きく移動する際には有効に作用する。しかし、ヤジロベエなど全体のバランスがとれていてあまり動かないものをデザインする場合は、それほど長い時間のシミュレーションは不要であり、かつ多数のオブジェクトのすべての動きの軌跡を表示させてしまうとディスプレーが煩雑になってしまい、かえって状態を把握することが難しくなる。そこで、図9のように動きの少ないオブジェクトのデザインを行うための描画モードを搭載している。このモードでは、各オブジェクトの中心点と最大移動点の移動距離を描画前に比較し、ほとんど差がない場合は平行移動、中心点がほとんど動いていない場合は回転と判断し、それぞれブルー、マゼンタの軌跡で表現する。



図9 左:回転 右:平行移動

4. 評価実験

提案手法の有効性を評価するために、被験者を用いて次に述べるような二つの異なるタスクを本システムのインターフェース(A)および一般的なシミュレーション表示(B)とで行ってもらい、完了までに要する時間を比較した。被験者は計算機を日常的に用いているが製品設計等は行ったことがない20代前半の男性4人とした。タスクを行う前に10分程度のレクチャーを行ってシステムの使い方に慣れてもらった後に実験を行っている。また各タスクにおいてAとBを行う間隔は一日以上あけて記憶の影響を減らすとともに、Aから先に実行する組とBから先に実行する組を同人数になるように調整してある。テストに用いた環境は以下の通りである。

CPU : Intel Xeon E5430 (2.66GHz FSB: 1333MHz)

RAM : 4GB FB-DIMM ECC 667MHz

VGA : NVIDIA Quadro FX 4600 (VRAM : 768MB)

Microsoft Vista sp1, Java 1.6.0_11 + JOGL 1.1.1

4.1. Task1: 球の初期位置を探す

このタスクでは、図10のように落下させたときにすべてのオブジェクトと接触するような球の初期位置を探す。このタスクはピタゴラ装置やコリントゲームといった、多数の衝突を含むもののデザインを模したものである。すでに位置の決定している3つのオブジェクトすべてに自然落下する球が接触する場所を試行錯誤で求める必要がある。

図11はタスクの完了までに要した時間を表したグラフであるものである。本システムによってタスクの完了までに要する時間を平均値で1/3近くに短縮することができた。また、「シミュレーションの結果を見ながらのほうが正しい場所を探す際の当たりをつけやすかった」「いちいちシミュレーションを実行しなくても自動で結果を見せてくれるため効率的に行えた」といった評価を得ることができた。

4.2. Task2: モビールのバランスの修正

このタスクでは、図12のような糸でつながれた二種類のモビルを、正しくバランスがとれた状態になるよう

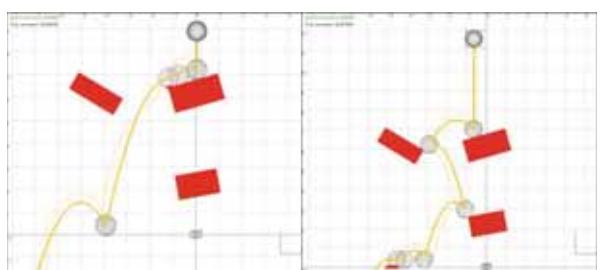


図10 タスク1: 落下させた際に全てのブロックに接触するような球の位置を探す

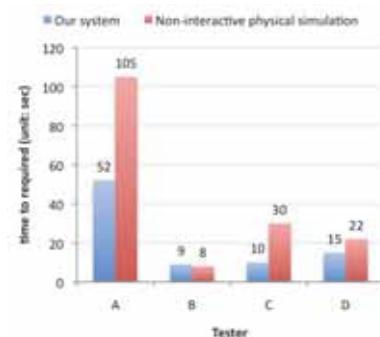


図11 タスク1の完了に要した時間

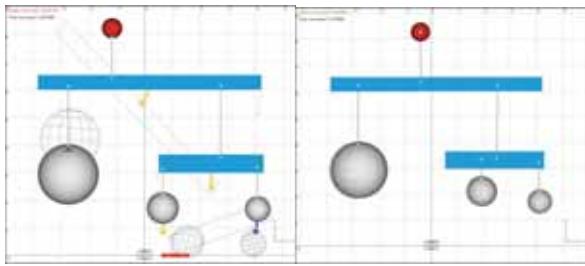


図12 タスク2：ヤジロベエの釣り合いの修正

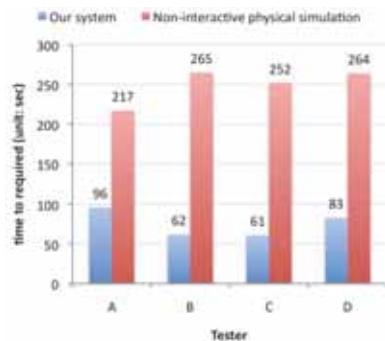


図13 タスク2の完了に要した時間

に糸やおもりの位置を調整するというタスクである。バランスがとれているかどうかは紐でつながれた長方形のオブジェクトの頂点の移動量の平均で判断し、もっとも移動量の多かった頂点の移動量が 1mm/sec 以下となったら合格としている。このとき、動かしてよいのは糸と球形のおもり、および固定点のみとし長方形のオブジェクトは移動させてはならないという制約を加えた。

図 13 はタスクの完了までに要した時間を表すグラフである。本システムによってタスクの完了までに要する時間を平均値で半分近くに短縮することが可能であった。また、「見やすく、オブジェクトの挙動を理解しやすかつた」といった評価を得ることができたが、インタラクティブでないシミュレーションを用いた手法であっても「オブジェクトの動きを連続的にアニメーションとして見ることができるのでわかりやすい」という評価もあった。

ちなみに被験者によっては図 14 のように中心のおもりを右側に寄せた形でバランスをとらせた場合もあった。このシステムでは青色の棒も質量を持つためこのような解も妥当なものであるが、我々の想定にはなかったものである。この結果はすなわち、本システムが多数存在する妥当な解を探す際の支援として十分に機能するということを示している。

計算式によって機械的に妥当な解を求める方法では、このようにシステム設計者の意図を超えた解をユーザに提示することは難しいだろう。

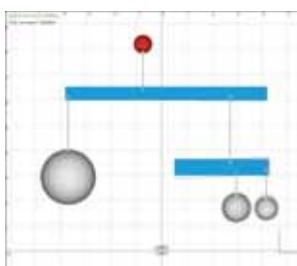


図14 重りを右に寄せた別解

4.3. 他の例

他にもいくつか、本システムを用いてキネティックアートのデザインを行い、現実の材料で実際にそれらを作成した。

図 15 はループを含んだ複雑な構造をもつモビールである。本システムを用い、およそ 15 分程度の時間でデザインを行うことが可能であった。図 15 右はそれを 5mm 厚のアクリル板で作成したものである。シミュレーション結果と同じく、バランスのとれたものとなった。また、図 16 も同様にループを含み、かつ三次元的な構造を持ったモビールである。

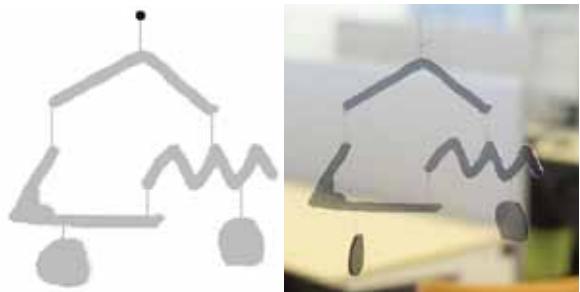


図15 モビールの例

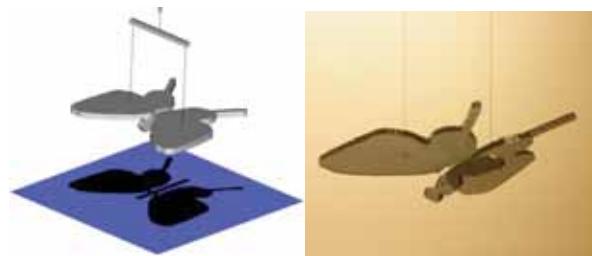


図16 モビールの例

図 17 は他の被験者によって作成されたものである。最初に 5 分程度使用方法の説明を行い、その後時間制限や質問制限などない状態で自由にデザインしてもらった。完了までに要した時間は一時間程度で、「ほぼ希望通りの形状をデザインできたが、取り消しや穴あけの機能が欲しい」という評価が得られた。

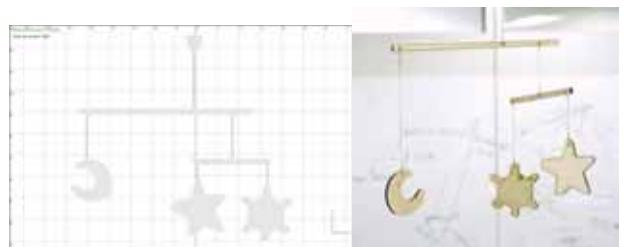


図17 被験者によってデザインされたモビール

5. まとめと今後の課題

本論文では動きのある作品を作るために、モデリングと物理シミュレーションを統合したシステムを提案し、それを実現するための物理演算のアルゴリズム、および得られた結果の提示方法について述べた。それらを実装することで、従来のモデリングとシミュレーションが分か

れているシステムよりも効率的に設計作業を行えることが確かめられた。また、設計した形状を実際に作成し、設計したとおりにバランスがとれたものを手にすることが可能であった。

本論文で提案した手法では移動するオブジェクトの移動の様子を可視化することでデザインを行いやすくなっている。しかし多数のオブジェクトが同時に、かつ長い距離を移動するような場合、それを可視化すると画面が非常に煩雑となる。そういう場合は可視化するオブジェクトをユーザが任意に選択できるようにするなどの対策をとる必要があるだろう。

また、同じ「バランスがとれている」という状態であったとしても、安定していて多少の外力では動かないようなものよりは、不安定で少しの外力でも大きく動くようなものをユーザは望む場合がある。現在のシステムでは両者を見分ける仕組みはないため、作品の安定性を評価する手法および外力を与えるインターフェースを考案し実装する必要があるだろう。

本システムで実現したモデリングとシミュレーションを融合させるというアプローチはより一般的な物作りにおいても有効であると考えられる。その場合は作る対象にあわせて FEM や BEM などの適切なシミュレーションの手法を導入する必要がある。そのためには、それら構造解析の計算を高速に行うための手法を開発しなければならない。また、そういう場合は変形の様子をどのように可視化すべきかについても検討の余地があるだろう。

参考文献

- [1] Bernward Frank, “Kineticus”,
<http://www.kineticus.com/>
- [2] Ole Flensted, “Flowing rhythm”,
<http://www.flensted-mobiles.com/>
- [3] Ole Flensted, “Bauhaus mobile”,
<http://www.flensted-mobiles.com/>
- [4] (株)イシグロ, “テーブルモビールヤジロベエ”
- [5] Jun Mitani and Hiromasa Suzuki, “Making papercraft toys from meshes using strip-based approximate unfolding”, ACM Trans. Graph., Vol.23, No.3, pp.259-263, 2004.
- [6] Tim Weyrich, Jia Deng, Connelly Barnes, Szymon Rusinkiewicz, and Adam Finkelstein, “Digital bas-relief from 3D scenes”, ACM Transactions on Graphics (Proc. SIGGRAPH), Vol.26, No.3, August 2007.
- [7] Yuki Mori and Takeo Igarashi, “Pillow: interactive pattern design for stuffed animals”, In SIGGRAPH ’06: ACM SIGGRAPH 2006 Sketches, p.74, New York, NY, USA, 2006. ACM.
- [8] Christine Alvarado and Randall Davis, “Resolving

ambiguities to create a natural computer-based sketching environment”, In SIGGRAPH ’06: ACM SIGGRAPH 2006 Courses, p.24, New York, NY, USA, 2006. ACM.

- [9] Emil Ernerfeldt, “Phun”,
<http://www.phunland.com/wiki/Home>
- [10] プロメテック・ソフトウェア株式会社, “Physicafe”
<http://www.prometech.co.jp/physicafe/>
- [11] Jovan Popović, Steven M. Seitz, Michael Erdmann, Zoran Popovic, and Andrew Witkin, “Interactive manipulation of rigid body simulations”, In SIGGRAPH ’00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pp.209-217, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [12] Matthew Straub, Justin Needham, “Digitally assisted billiards”, <http://www.prism.gatech.edu/gtg279x/Digitally Assisted Billards/Welcome.html>
- [13] Peter Eastman, “Art of illusion”,
<http://www.artofillusion.org/>
- [14] NVIDIA corp, “Physx 2.8.1”,
<http://developer.nvidia.com/object/physx.html>

著者紹介

- 古田 陽介：筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻 博士一年, JST ERATO 五十嵐デザインインタフェースプロジェクト 研究補助員, 305-8573, 茨城県つくば市天王台 1-1-1
三谷 純：筑波大学大学院システム情報工学研究科 コンピュータサイエンス専攻, JST ERATO 五十嵐デザインインタフェースプロジェクト 研究推進委員, (兼任所属機関)科学技術振興機構さきがけ, 305-8573, 茨城県つくば市天王台 1-1-1
五十嵐 健夫：東京大学大学院情報理工学系研究科 コンピュータ科学専攻, JST ERATO 五十嵐デザインインタフェースプロジェクト 研究総括, 113-0033 東京都文京区本郷 7-3-1
福井 幸男：筑波大学大学院システム情報工学研究科 コンピュータサイエンス専攻, 305-8573, 茨城県つくば市天王台 1-1-1