

形状設計システムに物理シミュレーションを統合したキネティックアート・デザインシステム

Computer Aided Kinetic Art Design System Comprising Rigid Body Simulation

古田 陽介 三谷 純 五十嵐 健夫 福井 幸男*

Summary. 近年の製品設計の場においては、計算機による形状設計システム（CAD）が広く用いられている。さらに設計された形状に対して、その強度や機能を評価するためのシステム（CAE）も普及しつつある。しかし、これらは別のソフトウェアとして独立して運用されることが多く、また CAE ソフトの扱いには高度な知識を要するため、これらを一般ユーザが日常的に使用するには至っていない。我々は、CAD と CAE の機能を統合することで、一般消費者が自分の身近なものを手軽に設計できるシステムの提案を目指している。本稿では、その具体例の 1 つとして剛体物理シミュレーションエンジンを形状設計用のシステムに組み込むことで、形の編集作業の最中に、その物理的な挙動を瞬時に予測して、その結果を連続的に提示するシステムの開発を行った。このシステムでは、時間経過に伴って変化する物体の位置や動きを高速にかつユーザにわかりやすく提示するために、シミュレーション精度の動的な調整や軌跡や回転を表すアニメーション表示などの工夫を行った。提案システムを用いることで、従来は試行錯誤が必要であったキネティックアート作品（動きを伴うアート作品）の設計が容易に行えることを示す。

1 はじめに

キネティックアートとは風力や水力、あるいは観覧者の力によって、動きの変化を伴うアートのことである。ヤジロベエやモビール（図 1）は我々に身近な例であり、バラエティに富んだ作品がこれまでに発表されている[4][5]。キネティックアートは「動く彫刻」と呼ばれることもあり、20世紀初頭からさまざまな動きを作品に取り入れる試みがなされてきている。しかし、物理的に「動く」物体のデザインを行うときには、複数の物体間の相互作用を考慮しなければならず、完成までには数多くの試行錯誤が不可避である。本稿では、このような動きを伴う作品の設計を支援するシステムを提案する。

形状設計に CAD システムを用いることは専門家を中心に関わるところであり、近年ではユーザインターフェースを改善することで、専門家でなくとも、直観的な操作で形の設計を行えるようにする研究が盛んに行われるようになってきている。さらに、ここ数年のあいだでの、光造形法や粉末法を用いた三次元造形装置の低価格化は著しく、自分でデザインしたものを作りながら実際の手にできる環境が整いつつある。

しかし、実際の物作りにおいては「形」だけでなく、その物理的な挙動についても考慮しながらデザインを行う必要がある。CAE のシステムを用いれば



図 1. キネティックアートの例。
上:ヤジロベエ。下:モビール [4]

デザインされた形状が実世界でどのように振る舞うかを知ることが可能であるが、このようなシステムは未だ専門家を対象としたものであり、一般消費者が扱うことは困難である。さらに、CAD と CAE は独立に運用されることが多く、形の設計とその評価を対話的に繰り返し行うことができない。デザインにおいて最初から完成型が決まっていることはまれであり、多くの場合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新しい形が生み出される。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運用には適していない。

そこで本稿では、動きのある作品の制作を効率的に行うために、三次元 CAD システムに剛体物理シ

Copyright is held by the author(s).

* Yohsuke FURUTA, Jun MITANI and Takeo IGARASHI, JST ERATO 五十嵐デザインインタフェースプロジェクト, Yukio FUKUI, 筑波大学大学院システム情報工学研究科

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、それに動力が作用した際の結果を瞬時に計算し、それを連続的に提示しつづけることで、期待する動きを得るためにユーザが実世界で試行錯誤を繰り返すコストを低減できる。提案システムでは物体の動きを単に時間軸に沿ってアニメーションと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静止画として提示することで一目で理解しやすくする工夫を行った。詳細は文献[6]または添付のビデオも参照して頂きたい。それらの実現に必要となる、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、およびユーザへの動きの提示方法の詳細について第3章以降で述べる。

2 関連研究

本稿で提案するシステムは文献[6]のものであるが、本稿では特にユーザインターフェースの観点から提案システムの有効性を述べる。

CGの技術を使って実際の物作りを行うアプローチにはさまざまなもののが提案されている。Mitaniらは与えられた三次元モデルから曲面を持ったペーパークラフトの展開図を作成するシステムを提案している[8]。Moriらによるぬいぐるみの型紙作成システム[9]やFurutaらによるバルーンアートのデザインシステム[7]など、素材の物理的な特性を考慮した形状設計システムが提案されている。本稿では剛体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システムを提案するが、形状設計のステージに物理シミュレーションを組み込むという点が共通する。

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おり、近年ではこれを形状設計のシステムと組み合わせて利用する例が提案されている。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[1]やPhun[3]があげられる。これらのシステムでは、ユーザがオブジェクト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[12]。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し、目的にあったアニメーション結果を生成する。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。

3 キネティックアート・デザインシステム

本システムは形状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ション、およびその結果の提示という機能を備える。図2は本システムを用いてオブジェクトのデザインを行う例である。

本システムでは三次元の形状設計のためのGUIに「Art of Illusion 2.6.1 (AOI)」[2]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オブジェクト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オブジェクトを糸で接続する機能を実装し、モビールなどをデザインできるように拡張した。

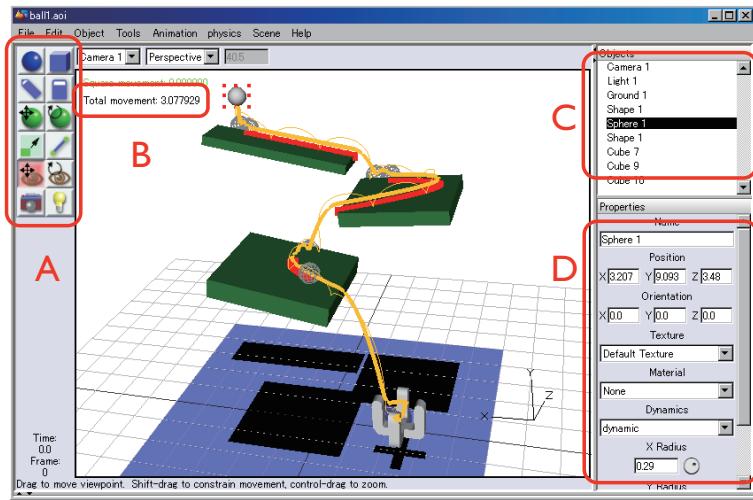
また、三次元仮想空間上での剛体シミュレーションには「PhysX 2.8.1」[10]を用いている。このライブラリは三次元の物理シミュレータとして多くの機能を持ち、一部のNVIDIA製GPUにおいてハードウェアによる支援を得ることが可能である。我々はそのうちの剛体シミュレーションおよび衝突判定の機能を利用して、ネイティブコードで記述されたPhysXのAPIをJavaで記述されたAOIから呼び出すため、JNI (Java Native Interface) を用いている。

図3は、我々のシステムを用いてモデリングを行う際の流れである。詳細については次節で説明する。

3.1 オブジェクトの生成と接続

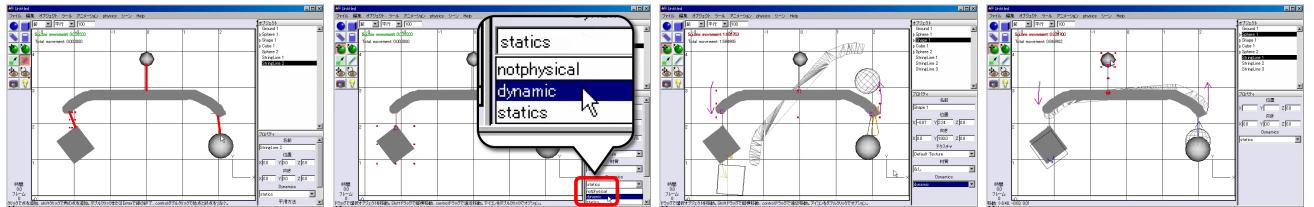
ユーザはマウスによる単純な操作で三次元空間上に球、直方体、板状立体の基本オブジェクトを生成することが出来る(図3(1))。板状立体とは閉じた二次元ストロークに一定の奥行きを加えて三次元化したものである。これらのオブジェクトは三角形メッシュで表現されていて、ユーザはその頂点をドラッグし変形させることができる。それぞれのオブジェクトは位置・姿勢を表すアフィン行列、密度、反発係数、静摩擦係数、動摩擦係数の値を持つ。

また、本システムは「接続」ツールを備えており、オブジェクトとオブジェクトの間をドラッグすることで二つのオブジェクトを仮想的な糸で接続することが出来る。この糸は最大長の情報を保持し、その値は生成時の長さが用いられる。



(A) ツールボタン (B) オブジェクトの移動量 (C) オブジェクトリスト (D) 選択されたオブジェクトの属性

図 2. システムのユーザインターフェース . ボールの落ちる軌跡が計算され , システムによって提示されている



(1) オブジェクトを生成し , 互いを糸で接続する (2) 生成したオブジェクトに「dynamic」属性を付与する (3) シミュレーションを有効にする (4) シミュレーション結果を見るながら修正する

図 3. モデリングの流れ . ワイヤフレームで表示されているものがシステムによって提示されたシミュレーション結果

3.2 オブジェクトのシミュレーション属性

各オブジェクトは「dynamic」「static」「non physical」の三通りのうちのいずれかのシミュレーション属性値をとる . 動的 (dynamic) なオブジェクトはシミュレータによって力が加えられた際の挙動が計算され , 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 . この値は GUI 上でユーザが変更することが可能となっており (図 3(2)) , 初期状態では static が用いられる .

3.3 反応性の高い物理シミュレーション

一般に , 物理シミュレーションは計算コストが高い . 全体の処理にはオブジェクト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 , さまざまな初期化処理が含まれてあり , それらの計算にも大きなコストが必要となる . 本システムではそれらを削減するために , 次のような高速化手法を取り入れた .

1. シーンが変化した際は形状が変化したオブジェクトのみを初期化する
2. ユーザの操作に合わせて , シミュレーションの時間刻み幅を動的に変化させる

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縮の方を重視し , 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 . 対方に , ユーザがオブジェクトを微小距離だけ移動させている場合は正確な位置を探るために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 , 時間刻み幅を 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 , 時間刻み幅を 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 , ユーザは編集の作業に集中することが可能である .

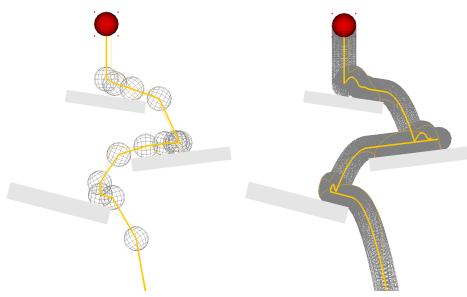


図 4. 動的な時間刻み幅の変更の例。
左:刻み幅が大きい場合。右:刻み幅が小さい場合

3.4 シミュレーション結果の可視化

シミュレーションによって得られたオブジェクトの位置情報をどのように可視化するかについて考える。本システムでは物体の動きを一目で理解しやすくするために、単に時間軸に沿ってアニメーションするのではなく、すべての動きを一つのフレームに静止画として提示することとした。

しかしながら、図 5(A) のように、すべての場所に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中に同時に存在していたりすると、画面が煩雑になってしまい状態を把握できなくなってしまう。そのような事態を避けるため、時間軸に沿った位置情報の集合から動きの変化した時刻 d_t を求め、その時刻のオブジェクトの状態だけを描画することで動きの様子を把握しやすくなる。

d_t はつぎの式によって求める

$$\mathbf{v}_t = \mathbf{M}_t \mathbf{p} - \mathbf{M}_{t-1} \mathbf{p} \quad (1)$$

$$\mathbf{a}_t = \mathbf{v}_t - \mathbf{v}_{t-1} \quad (2)$$

$$d_t = \mathbf{a}_t \cdot \mathbf{a}_{t-1} \quad (3)$$

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 5(B))。

また、よりオブジェクトの挙動を把握しやすくするため、重心の軌跡もあわせて描画する。さらに、オブジェクトの各頂点について、移動した総距離を

求め、最大となった点の軌跡も描画する(図 5(C))。軌跡の総距離が最大となる頂点を探すために、システムはすべての頂点について軌跡の長さを計算し保持する。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。加えて、衝突があった際にはその座標に赤い点を表示する(図 6)。

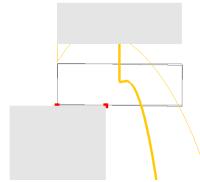


図 6. 箱が障害物の縁に衝突する様子。衝突点を赤で描画

3.5 動きの少ないオブジェクトの動きの可視化

前節で述べた可視化手法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形状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る。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、画面が煩雑になってしまいかえって状態を把握しにくくなる。

そこで、我々のシステムには動きの少ない形状のデザインを行うための描画モードを搭載している(図 7)。このモードでは、オブジェクト内で移動距離の最も大きい頂点と、重心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示する。それ以外の場合は二本の黄色の矢印で、各軌跡を表示する(図 6)。

4 作例

本システムを用いてモバイルのデザインを行い、現実の材料で実際にそれらを作成した。

図 8 はパーツの接続関係にループを含んだ構造をもつモバイルであり、ユーザはパーツの形状だけでなく、その釣り合いと衝突についても注意しながら

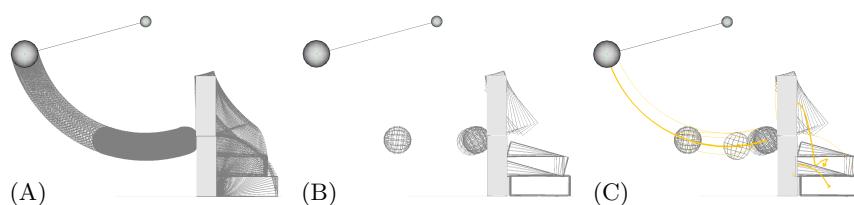


図 5. 重なり合った二つの箱に対して球をぶつける様子を描画した様子

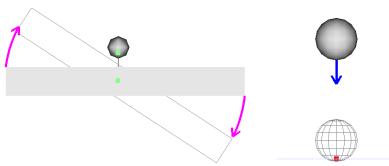


図 7. 左:回転の表現、右:平行移動の表現

デザインを行わなければならない。本システムを用いることでユーザーはおよそ 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 8 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 9 も同様に接続関係にループを含み、かつ三次元的な構造を持ったモビールである。図 10 のデザインを行ったユーザーからは「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」という評価を得た。

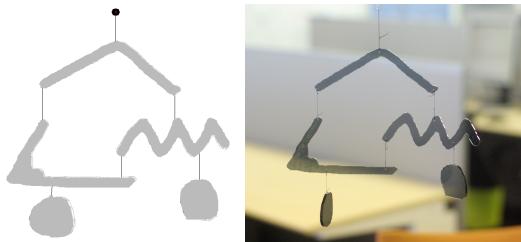


図 8. ループ構造を含むモビールの例

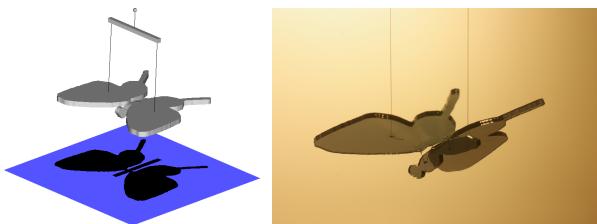


図 9. ループ構造を含む三次元モビールの例

図 11 はバランスが崩れた状態のモビールの糸の位置を被験者が移動させて、バランスがとれた状態へと修正するという試験を実行した結果である。6人の被験者に対して実験を行ったところ5人は下段左図のような形に修正を行ったが、1人の被験者は右図のように錘を右へと寄せることでバランスを取るという解を発見しそれを回答とした。これは我々も当初は想定していなかった答えであったが、青棒にも質量があるため、正しい解であると言える。このように本システムを使いユーザーが試行錯誤することによって、予想外の形を生み出すことも可能であるだろう。

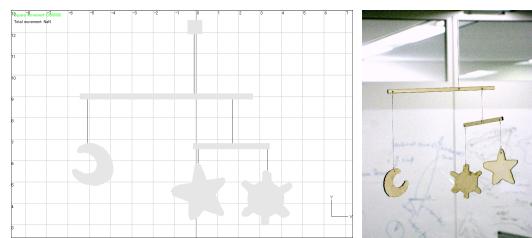


図 10. ユーザによってデザインされたモビール

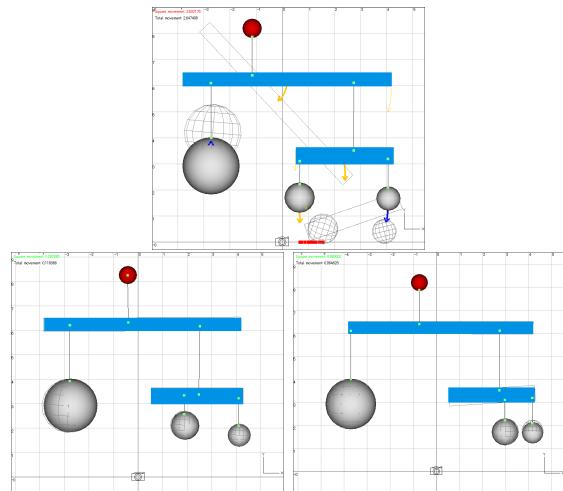


図 11. 本システムを用いてバランスを修正した結果

5 まとめと今後の課題

本稿では一般消費者向け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シミュレーションの時間刻み幅を調整する工夫、および得られた結果の提示方法について述べた。こ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している。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ー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。衝突をたくさん含むような複雑なアート作品の場合、その提示方法にも改善が必要になると予想される。

またキネティックアートのデザインにおいて、「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

のをユーザは好む場合がある。現在のシステムでは両者を見分ける仕組みがないため、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ことも今後の課題である。

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アプローチは、キネティックアート作品に限らず、広範囲の物作りにおいて有効であると考えられる。ただし、対象とする物体の目的にあわせて有限要素法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るべきかについても検討の余地があるだろう。

参考文献

- [1] C. Alvarado and R. Davis. Resolving ambiguities to create a natural computer-based sketching environment. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, p. 24. ACM, 2006.
- [2] P. Eastman. Art of Illusion. artofillusion.org.
- [3] E. Ernerfeldt. Phun. www.phunland.com.
- [4] O. Flensted. Bauhaus mobile, Flowing rhythm.

未来ビジョン

我々が目標としている「一般ユーザ向けのモノ作りシステム」とは、実際のモノ作りにおけるさまざまな制約を未経験者にも意識させ、ユーザの創造力の発揮をサポートするようなものである。また最終的には設計したものを実際に実在の素材を加工して出力するようにもしたいと考えている。

モノ作りにおける専門家とそうでない一般ユーザとの間にある違いには、大きく2つのものがあると考えている。一つはツールの使いこなしの術を持っているかどうか、もう一つは現実世界のモノ作りにおけるさまざまな制約に関する知識を十分備えていて、その回避のためのセオリーを知っているかどうかという点である。前者の改善はユーザインターフェースの研究としてすでに一般的になりつつあるが、後者はユーザインターフェースの研究としてはあまり認知されていないと思われる。

制約にはさまざまなものがあり、強度、大きさ、材質などといった対象物体自体の物理的なものから、すでに存在する他のモノとの位置関係や大小関係、色のマッチングのような周りの環境に依存したものもある。従来からの研

<http://www.flensted-mobiles.com/>.

- [5] B. Frank. Kineticus. www.kineticus.com.
- [6] Y. Furuta, J. Mitani, T. Igarashi, and Y. Fukui. Kinetic Art Design System Comprising Rigid Body Simulation. In *Computer-Aided Design and Applications*, Vol. 7, pp. 533–546, 2010. <http://www.npal.cs.tsukuba.ac.jp/~furuta/>.
- [7] Y. Furuta, N. Umetani, J. Mitani, T. Igarashi, and Y. Fukui. A Film Balloon Design System Integrated with Shell Element Simulation. In *Proc. of EUROGRAPHICS 2010*, May 2010.
- [8] J. Mitani et al. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Trans. Graph.*, 23(3):259–263, 2004.
- [9] Y. Mori and T. Igarashi. Pillow: interactive pattern design for stuffed animals. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Sketches*, p. 74. ACM, 2006.
- [10] NVIDIA corp. PhysX 2.8.1. <http://developer.nvidia.com/object/physx.html>.
- [11] J. Popović et al. Interactive manipulation of rigid body simulations. In *Computer Graphics (Proc. of SIGGRAPH '00)*, pp. 209–217. ACM, 2000.
- [12] プロメテック・ソフトウェア株式会社. PhysiCafe. <http://www.prometech.co.jp/physicafe/>.

究では、そのような制約を事前に全て集めておいて、それをいかに計算機で効率的かつ正確に問題を解くかという点に議論が集中していたが、そこで得られた回答はユーザにとって十分満足できるようなものとは言えなかった。なぜならほとんどの場合、目的とする完成形を得るために十分な制約を事前に入力しておくことは、量が多すぎて現実的ではないためである。

そのような問題は、計算機上にさまざまな制約を入力するためのインターフェースを実装することで改善できるだろう。また、システムは正確な解を一つだけ提示するのではなく、複数の幅のある回答を提示しユーザがその中から最適なものを選択するという作業を繰り返すことで、解の正確性を高めることが出来るのではないだろうか。そのような繰り返しの作業をユーザが効率的に実行できるようにすることで、たとえユーザが現実世界のモノ作りにおける様々な制約に関する知識・セオリーを備えていなくても、ユーザにとって満足度の高いデザインを行うことが可能となると考えている。そのようなシステムを実現するためのインターフェースの開発に我々は取り組んでいきたい。