芸術科学会論文誌 Vol. 20, No. 1, pp. 40-44 (2021)

blobby: 手描きアニメーションにおける流体調表現の生成

藤田恭輔 [1)]　森本有紀 [1)]（正会員）　知足美加子 [1)]

1）九州大学芸術工学府

blobby: Generation of Fluid Style Effects in Hand-drawn Animation

Kyosuke Fujita[1)]　**Yuki Morimoto**[1)](Member)　**Mikako Tomotari**[1)]

1) Graduate School of Design, Kyushu University

fujita.kyosuke.761@s.kyushu-u.ac.jp,　{morimoto, tomotari}@design.kyushu-u.ac.jp

アブストラクト

手描きアニメーション作品において，伸縮を伴う流体的なエフェクト表現がよく用いられる．このエフェクト表現を，本稿では blobby と呼ぶ．blobby の作成は一枚一枚手描きで作成するため非常に大変であり，また，作成のための習熟した技術が必要である上，完成後の大きな修正は難しい．そこで本研究では，blobby を生成する疑似弾性体を用いた方法と，二つの動きの制御手法の提案を行う．また，本手法ではレンダリングにメタボール用いて blobby の特徴を表現する．これにより，手描きアニメーション特有の誇張された流体エフェクトの容易な作成を可能とする．

Abstract

In recent animations, we can find fluid-like effects that stretch. We call these effects “blobby”. Blobby is drawn by hand. Therefore, it is tedious to generate blobby and it requires skilled animator, and it is hard to fix it once blobbies are generated. Thus, we propose a method to generate blobby as pseudo elastic body and two kinds of control methods for blobby. Our method applies metaball to render blobby’s characteristics. As a result, our method enables to easily generate exaggerated fluid style which is unique effects of hand-drawn animation.

1. はじめに

近年のアニメーション作品では，伸縮性のある滑らかな動きの液体の塊のような表現が見られる（図1）．Kevin Baoが監督した「NEW ERA」のミュージックビデオ[1]では，インクを模したエフェクトとして用いられている．また，水江未来の「ピコトピア」[2]では，細胞の動きを結合・分離する液体の塊のように描かれている．同様の表現として，ユージンの「机上のダンサー」[3]など，類似の表現が多数ある．

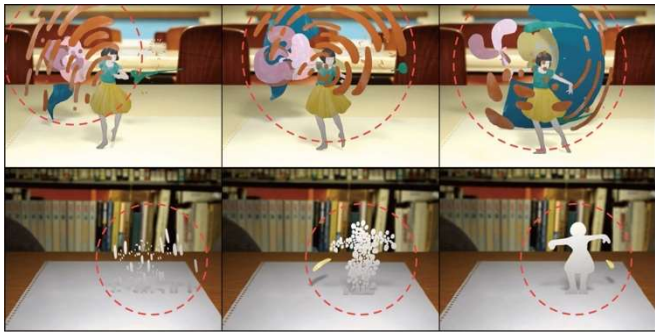


図1：手描き映像作品[3]で見られるblobby（赤線）．

このような流体表現を本稿ではblobbyと呼ぶ．blobbyは手描き作業によって作成される．そのため，一定の技術が必要であることや，完成後の大きな修正が難しいなどの問題がある．blobbyを観察すると，３次元的な軌道を描くものや，複雑に分岐するものなど様々なバリエーションが存在する．それらの中で特に共通して，以下の特徴を備えていることがわかる．

特徴① 元の体積に関わらず著しく伸縮し，結果として移動する際に尾を引く挙動となる．

特徴② なめらかに分離・結合する．

本研究ではこれらの特徴を再現することを目的とする．

2. 関連研究

アニメーションにおける誇張された水の振る舞いを，手書き作業を行う事なく再現することを目的としたJinhuiらの研究[4]がある．この研究では漫画における水の表現を波や水しぶき，波紋などに分類し複数のテンプレートを設計する．これを用いてユーザーが入力したパラメーターを元に水流の経路の作成や，波紋が広がるアニメーションの自動生成を可能にする．本手法で表現する対象は水ではないため，本研究とは目的が異なる．

また，手描きアニメーション調の液体や煙などの流体の描画を目的とする研究[5,6]がある．いずれも物理法則に基づいたシミュレーションによって生成された物体の動きを元に，手描きアニメーション調のレンダリングを可能にしている．前者[5]では，深度差のある部分に輪郭線をつけたり，視点に対する液体の厚みによって色を変更したりすることで強調表示を行なっている．後者[6]では，前者と同様に深度差による輪郭線追加に加え，パーティクルの密度や速度によってパーティクルの形状や大きさを変化させている．これらの手法では，動きに対して誇張表現が加えられていないため，手描きアニメーション特有の動きなどの表現には適していない．

Stephenらの研究 [7]では，速度や壁との衝突に合わせてボールを自動的に楕円形に誇張するなど，手描きアニメーション特有の形状の誇張を可能にしている．しかし，この誇張表現はモデル全体に対して同一の行列を掛けるアフィン変換のため，変形の柔軟さに欠ける．そのため，流体的な表現には適さない．

blobbyは「モーショングラフィックス」と呼ばれる類の映像においても散見される．Adobe After Effectsなどのソフトウェアを用いることで手描き作業を行うことなく，液体的な揺れの表現や，特徴②のような分離・結合表現を作成することができる[8]．この手法では固定された輪郭線からの距離を入力として，液体的な表現を付加する．しかし，輪郭線で囲まれた形状は単体では変形しないため、伸縮表現には適さない．一方，本手法では単一のblobbyを粒子の集合として定義し，擬似弾性体を適用する．これにより，本手法におけるblobbyは柔軟に変形でき，より伸縮表現に適している．

以上を踏まえて，本研究の新規性を以下に記す．

・擬似弾性体手法を用いた，伸縮を伴う誇張表現を行なう．

・blobbyの生成のためのLSMを制御するパス描画および速度場の2つの手法を提案する．

3. 本手法

本手法ではblobbyは擬似弾性体のシミュレーション手法であるLattice Shape Matching (LSM) [9]を基に表現する．blobbyの移動を制御する手法として，本手法ではパスとベクトル場による二種の制御手法を提案する．これらによりユーザーによる制御の簡易化と，第2章で挙げた特徴①の尾を引く挙動を実現する．レンダリングには特徴②の滑らかな分離・結合の表現のため，計算負荷を軽減したメタボールを用いる．本手法の処理の流れは以下である．

1. パスおよびベクトル場によるblobbyの移動指定.
2. LSMを用いたblobbyの変形.
3. メタボールを用いたblobbyのレンダリング.

この章では，まずLSMと頂点の配置について述べ（4.1節），次に制御手法（4.2節），レンダリング（4.3節）について述べる．

3.1 LSM を用いた blobby モデル

LSMを利用した変形手法はMüllerらが考案したShape Matching を用いる変形手法[10]を拡張したものである．Shape Matchingは頂点群で構成された物体が変形した際，元形状に戻るように位置と速度を更新する．この際，元形状と変形後の形状が出来る限り一致するよう剛体変換(平行移動と回転移動)を計算する．そして，この剛体変換を適用して元形状を求めることで元形状に戻る力を求める．

一方LSMでは，格子状に配置された頂点群を対象とする．頂点群に対し，複数の近傍頂点からなる局所領域を頂点と同じ数だけ生成する．局所領域は各頂点を中心とするmxmの頂点で構成される．本手法では$m = 5$とした（図2）．そして，各局所領域の頂点に対してShape Matchingの計算を行う．これにより求め

た理想位置に向かって各頂点が移動するよう位置と速度を更新する．この計算を繰り返すことで，物体の擬似的な弾性体表現を行う．このときmが小さいほど物体は柔軟になるため，mは柔軟さのパラメーターとして利用できる．この手法ではShape Matchingを用いた変形と比べ，より柔軟に変形させることができる．

blobbyは流体を誇張した表現であり，多くが円形や楕円形をしている．これは実際に働く表面張力によって水滴が丸くなる現象を模しているものだと思われる．よって本研究では何も力が加わっていない時のblobbyを円状とした．また，LSMの計算点は格子状に配置するよう，円を覆う正方格子上に頂点を配置する（図2左）．また，LSMモデルにおける各局所領域の大きさパラメーターmは任意の値を設定する．本稿では$m = 5$とした．円一つを表すLSMの頂点をp_i とおく．このとき$i = 1,2,3 \dots n$，nは頂点数である．

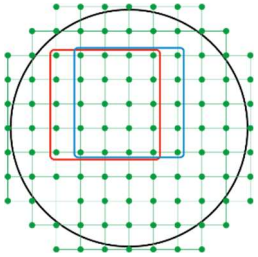


図 2：blobby の頂点の配置と局所領域.

3.2 制御

本手法では，blobby の動きをユーザが指示する二つの制御手法を提案する．一つ目は blobby をユーザーが描画指定したパスに沿って移動させるものである．これによって，ユーザーが一つ一つの blobby のパスや速度を自由に指定できる．二つ目はベクトル場を用いて外力を加えることで移動させるものである．後者では複数の blobby を一度に指定することができる．

3.2.1 パスを用いた制御

本手法ではまず，パスをユーザーが描画で入力する．パス一つにつき，blobbyは一つ配置される．パスは当分割し，各頂点の位置をl_j，$j = 0,1,2 \dots k-1$，kは点の数とする．

次に，blobbyの中心点cがパスl上を通るように移動する．この時，一部の頂点がパスに対して常に等しい相対位置を保つように平行移動させる．これによってLSMの変形を適用しつつ，経路に沿った移動を行う（図3）．

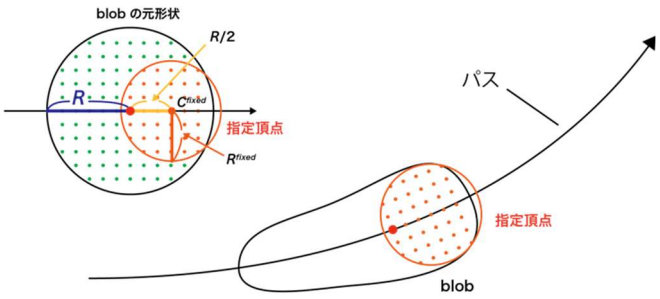


図 3：指定頂点を進行方向前方に位置させる.

パスに平行させる一部の頂点を指定頂点と呼び，以下のように選ぶ．blobbyの初期形状である円の半径をRとすると，時間t = 0におけるcを通る軸上の半径$R/2$の点c^{fix} より半径R^{fix}以内の点を指定頂点とする．経験的に，$R^{fix} = 3R/5$としたときに最も手描きアニメーションの表現に近い挙動となった（図3）．

blobbyの中心位置$c(t')$は以下のように求められる．

$$c(t') = l_j + (l_{j+1} - l_j)\{h\}, \qquad (1)$$
$$t' = \{t/T\}, \quad h = kt', \quad j = \lfloor h \rfloor$$

このときtは時間，$\{*\}$は$*$の小数部分，$\lfloor * \rfloor$は$*$の整数部分であり，Tは一つのパスの始点から終点までの移動時間を任意で与えたもの，$t' \in [0,1]$は時間を正規化したものである．式(1)により，$t' = 0$の時blobbyはパスの始点に，$t' = 1$の時は終点に位置する計算となる．また，時間を[0,1]で正規化したことにより，t'を使った速度関数を定義することが容易である．例えば

$$t' = \{t/T\}^3 \qquad (2)$$

などとして，移動に緩急をつけることができる．これは，全ての線分の長さは等しくt'の変化量とパス上をblobbyが移動する速度は比例するためである．

3.2.2 ベクトル場を用いた制御

ベクトル場を用いた制御手法では，まず任意のベクトル場を入力する．図4左にベクトル場の例を示す．次に，ベクトル場から各頂点へ力を加える．このとき加える力は，blobby内の部位によって重み付けする（図4右）．頂点p_iがベクトル場から受ける力F_iは，

$$F_i = w_i G(p_i), \qquad (3)$$
$$w_i = \min\left(\alpha \frac{|p_i^0 - c^0 + c^{fix}|}{R}, 1\right)$$

と計算する．このとき Gはベクトル場，αは定数，はw_iは重み付け係数である．重み付けは元形状（図3左上）における各頂点の座標p^0に基づく．その結果w_iが小さい部位は移動が遅れ，軌跡を描くように伸長する．本稿の結果ではα = 11/6を用いた．

ベクトル場Gは位置関数や画像によって指定できる．本稿では，以下のような位置関数によって与える．ベクトル場のサイズベクトルをsとすると，

$$G(p_i) = \hat{q}_i, \qquad (4)$$
$$q_i = \mathrm{Ref}(p_i') - \beta \hat{p_i'}, \qquad p_i' = \frac{2p_i}{s} - 1$$

このとき ^ は単位ベクトル化，Ref()はy軸に対する鏡映変換，βは定数，p_i'はp_iを$|p_i| < 1$となる領域内に正規化したものを表す．これらにより，blobbyはベクトル場に沿って移動する．βが大きいほど，ベクトルは領域の中心方向を向く．本稿では$\beta = 1/20$を用いた．

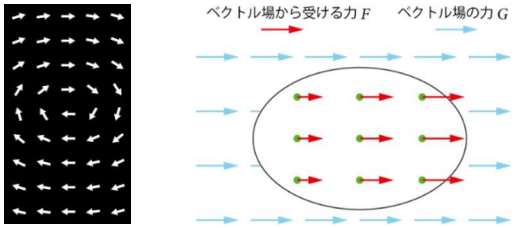


図 4：ベクトル場の例（左）と頂点への影響（右）.

3.3 レンダリング

レンダリングにはメタボールを用いる．まず，blobbyの各頂点p_iを中心としてその周囲の画素毎に濃度分布を配置する（図5）．本手法では計算負荷を減らすため，計算する濃度分布を各頂点から半径rの範囲内に限定する．そのため，p_iからr内の各画素pixへの距離をrで除算し正規化した$d(i,pix)$を用い，p_iから発生する濃度Dは以下の式で計算する．

$$D(i,pix) = \gamma\left(\cos\frac{\pi d(i,pix)}{2}\right)^{\kappa} \quad (5)$$

このときγ, κはそれぞれ濃度分布の濃さと勾配を調節する定数である．γが大きいほど，濃度分布全体の濃度が大きくなり，κが大きいほど，dによる濃度の変化が急になる．更に，各画素から半径r内にある頂点による濃度を加算したものが任意の閾値以上の値となる画素を指定色にすることでレンダリングする．本研究では，$\gamma = 1/5$, $\kappa = 5.5$，レンダリングの濃度閾値は1とした．

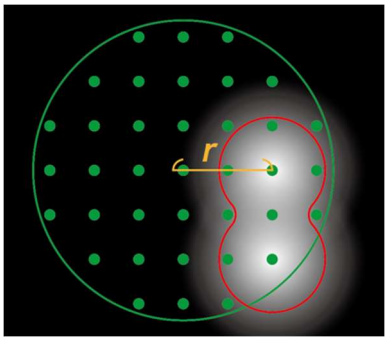


図5：グラデーションは白いほど濃度が高い．
赤線は濃度が閾値と等しい画素を示す．

4. 結果

本手法を用いた結果の生成にはCPU2.9 GHz Intel Core i5, メモリ 8 GB 1867 MHz DDR3, GPU Intel Iris Graphics 6100 1536 MB，C++フレームワークである openFrameworks を用いた．

本手法による結果アニメーションを図6,7に示す．それぞれ，パスおよびベクトル場による制御によって，横方向および円状に単一または複数のblobbyが動く様子を示す．また，実際の作中のblobbyとの比較のため作品[3]の一部を示し，以下考察する．

パス制御によるアニメーションでは，式3によりパスの終点に近づくほど，速度が大きくなるよう緩急をつけているため，作品と同様にblobbyが伸縮していることがわかる．

ベクトル場による制御法では，全ての位置で同じ大きさの力を加えるベクトル場を用いた．図7下段では式4で定義した渦状のベクトル場を用いた．ベクトル場制御においても，移動の際に伸長している．また，複数のblobbyが接近した際に滑らかに結合・分離する様子がわかる．

以上により，本手法によりblobbyの特徴①・②を再現できたことがわかる．また，本手法で用いた式やパラメータなどを変更することで，作品ごとに異なるblobbyの表現も可能と考える．そのような検証は今後の課題としたい．

その他，本手法では課題がいくつか挙げられる．まず，速度が大きい場合，カーブにおいてblobbyの後方部がパスに沿わないことがある(図7上段5列目)．また，ベクトル場を用いた制御法では移動速度に緩急がつけられない．この他，複雑な分岐・３次元的な軌道・色や模様の考慮なども今後の課題である．

5. おわりに

本稿では手描き作業によって作成される流体表現の生成手法を提案した．柔軟な変形を行うため，擬似弾性体の手法を適用した．また，メタボールを用いたレンダリングにより，輪郭をなめらかに変化することができた．また，パスとベクトル場を用いて制御し，同時にシンプルな速度式の使用により，動きを強調することができた．

将来課題として，本稿では取り扱わなかった更に幅広いblobbyの特徴の再現がある．例えば，水滴が面上を通った跡に残る水滴の表現や，色や模様の変化を考慮した制御手法などがあげられる．今後，様々なバリエーションのblobbyを再現するために，本稿で提案したモデルや制御手法の拡張を行っていきたい．

参考文献

[1] Kevin Bao ほか, Nulbarich - NEW ERA (Official Music Video). https://www.youtube.com/watch?v=5pkBqmX2ymc, 2016.

[2] 水江未来, ピコトピア, https://www.youtube.com/watch?v=fwMhX3TTWzA, 2013.

[3] ユージン, 机上のダンサー, https://gifmagazine.net/post_images/2990708?locale=ja, 2018.

[4] Jinhui Yu, Xinan Jiang, Haiying Chen, and Cheng Yao, Real-time cartoon water animation, In Computer Animation and Virtual Worlds, pp. 405–414, 2007.

[5] Ashley M. Eden, Adam W. Bargteil, Tolga G. Goktekin, Sarah Beth Eisinger, and James F. O'Brien, A method for cartoon-style rendering of liquid animations, In Proceedings - Graphics Interface, pp. 51–55, 2007.

[6] Andrew Selle, Alex Mohr, and Stephen Chenney, Cartoon rendering of smoke animations, In NPAR Symposium on Non-Photorealistic Animation and Rendering, pp. 57–60, 2004.

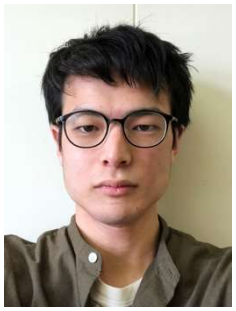
[7] Stephen Chenney, Mark Pingel, Rob Iverson, and Marcin Szymanski, Simulating Cartoon Style Animation, In NPAR Symposium on Non-Photorealistic Animation and Rendering, pp. 133‐138, 2002.

[8] 川原健太郎, 鈴木成治, プロが教える! After Effects モーショングラフィックス入門講座, ソーテック社, 2019.

[9] Alec R. Rivers and Doug L. James, FastLSM: Fast lattice shape matching for robust real-time deformation, In Proceedings of the ACM SIGGRAPH Conference on Computer Graphics, ACM Press, New York, New York, USA, pp. 82, 2007.

[10] Matthias Müller, Bruno Heidelberger, Matthias Teschner, and Markus Gross, Meshless deformations based on shape matching, ACM Trans. Graph. 24, 3 (July 2005), pp. 471-478, 2005.

藤田　恭輔



2020年 九州大学芸術工学部芸術情報設計学科卒業. 同年より九州大学芸術工学府芸術工学専攻修士課程在学中.

森本　有紀

2008 年九州大学芸術工学府修了，同年東京大学にて日本学術振興会特別研究員，2009 年より独立行政法人理化学研究所研究員，2012 年より東京電機大学未来科学科講師，芝浦工業大学助教などを経て，2016 年より九州大学助教．コンピュータグラフィクスに関する研究に従事．情報処理学会，ACM SIGGRAPH 各会員．博士(芸術工学).

知足　美加子

1995年筑波大学大学院芸術研究科美術専攻彫塑コース修了,国画会彫刻部会員，美学会会員，日本山岳修験道学会評議員，九州大学芸術工学研究院教授，博士（芸術学）

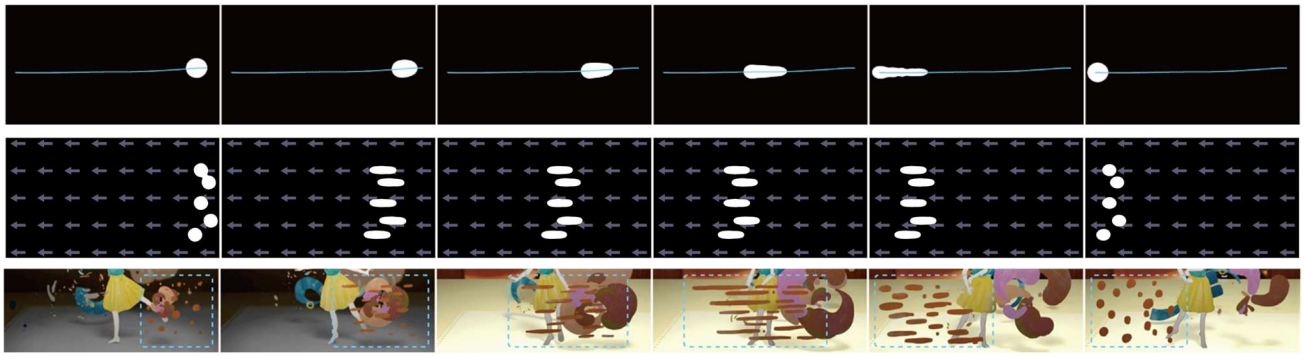


図 6 : パス（上）およびベクトル場（中）の制御によって横方向に移動する blobby と比較のための作中例（下）.

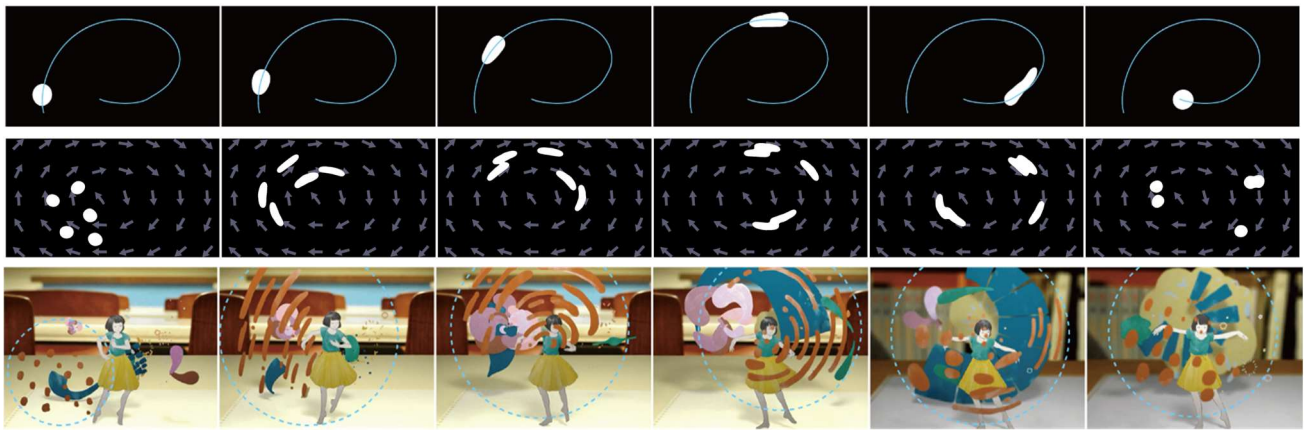


図 7 : パス（上）およびベクトル場（中）の制御によって円状に移動する blobby と比較のための作中例（下）.