

煙のアニメーションにおける輪郭線の描画を目的とした 曲線生成手法の検討

坂本 卓也*・竹下 大樹

Study on Curve Generation for Drawing Contour Lines in Animation of Smoke

Takuya SAKAMOTO* and Daiki TAKESHITA

(平成25年11月25日受理)

Recently, fluid and the other physical simulations have been used for not only industry but also CG for entertainment. They are often used to produce an image representing real shape and movement. On the other hand, they are becoming used to produce simplified or symbolized representation like a cartoon. Toon rendering is a technique of drawing 3DCG that brings cartoon like representation. In this technique, it is included a drawing contour lines. However, it is difficult to draw a fluid, such as smoke does not have a well-defined contour. In this study, for toon rendering, I define contour lines of smoke by using an isoline. In addition, I intend to add a unique effect to isoline. Finally, I have succeeded in make drawing a smoke's contour lines from output of 2D fluid simulation and add an effect of advection to contour lines by Hermit Curve

Keywords: CG, smoke, animation, toon rendering, curve generation

1. 序論

流体シミュレーションをはじめとする物理シミュレーションは主に写実的で現実の事象に則した造形や挙動を生み出すために研究されている。一方でそういったフォトリアルなコンテンツばかりではなく単純化・記号化されたトゥーンレンダリングもよく利用されるようになってきた。本研究では煙のトゥーンレンダリングにおける輪郭線の生成に関する検討を行った。

トゥーンレンダリングにおけるオブジェクトは輪郭線によって強調されているので、煙のようなもともと輪郭のないものとの相性は悪く、描画は難しい。そこで、煙に対して境界を持たせることでトゥーンレンダリングできるようにする必要がある。

3次元においては等値面を用いることで明確な面としての境界を与えることができる。本研究ではその基礎として2次元の煙に対して等値線を用いることで、2次元における境界線である輪郭線の描画を検討した。

また、煙としての輪郭線の特徴を際立たせるため

に、はっきりとした輪郭線をもちながら特徴的な挙動を示すように輪郭線を変形させる手法についても検討を行った。

2. 煙の輪郭線の描画

本研究では、流体のシミュレーションによって得られた計算結果を元に、その煙密度のデータを主に利用することで対象の輪郭線を描画する手法の検討を行った。

輪郭線描画までの流れは大まかに分けて3ステップで構成されており、以下の手順で行う。

- ①流体シミュレーションにより煙密度等を導出する。
- ②煙密度データを用いて等値線の元となる座標を導出する。
- ③等値線のデータを輪郭線のパスデータに変換し、補間関数によって平滑化する。

上記のうち、①におけるシミュレーション法については本研究報告の対象外であり、その計算手法やアルゴリズムについては触れず、計算結果として得られたデータの形式についてのみ記述する。

* 秋田高専専攻科学生

2.1. 流体のシミュレーションデータ

流体シミュレーションによって得られるデータは、2次元の格子点上に配置される煙密度、圧力、温度、回転のスカラー場と速度のベクトル場である。このうち、煙密度は煙の見た目をつかさどっているため、煙の輪郭線の描画には主にこのデータを使用する。

本研究ではシミュレーション用の格子の数は基本的に縦 100 × 横 100 として行った。

2.2. 等値線の導出

等値線とは同じ値を持つ点同士を結ぶことで得られる線であり、気象図における気圧の分布を視覚化した等圧線や、地図における高さを表現した等高線などがその一例である。

シミュレーションによって得られた煙密度場を輝度データに変換して利用し、等値線を導出することで通常は境界線がはっきりしない煙などに輪郭を持たせることができる。

2.2.1 等値線について

全体の等値線を導出するにはまず領域を三角形のメッシュに分割し、各三角形メッシュに対して線分を求める。メッシュ 1 つに対する線分は次のように求められる。

三角形メッシュは各頂点に値を持っており、本研究においては煙密度の値に相当する。

図 2.1 に 3 つの頂点があり各点に値を持つ三角形 $\triangle P_1P_2P_3$ を示す。

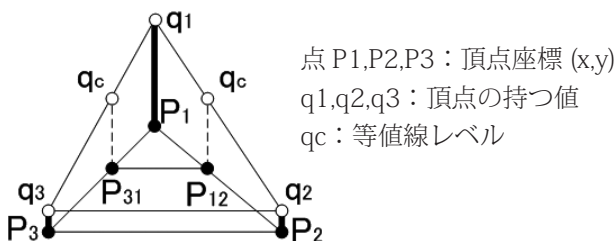


図 2.1 等値線を構成する線分

この三角形一つに対して 1 本の線分が得られる場合、どこか 2 辺に端点が存在することになる。

まず、辺 P_1P_2 について考える。辺 P_1P_2 上に端点が存在するには、

$$q_1 < q_c < q_2 \quad \text{または} \quad q_2 < q_c \leq q_1$$

が成り立つ必要があり、存在する場合にその点の座標を点 P_{12} とすると

$$P_{12} = \frac{q_c - q_2}{q_1 - q_2} (P_1 - P_2) + P_2 \quad (2.1)$$

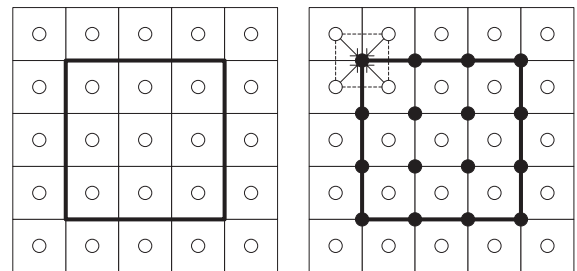
となる。

これをほかの 2 辺に対しても同様に行うと、点が 0 ~ 3 個出てくる。線分を構成するのは点が 2 個の場合のみなので、そのほかの場合は線分が存在しない。

2.2.2 シミュレーション結果への等値線の適用

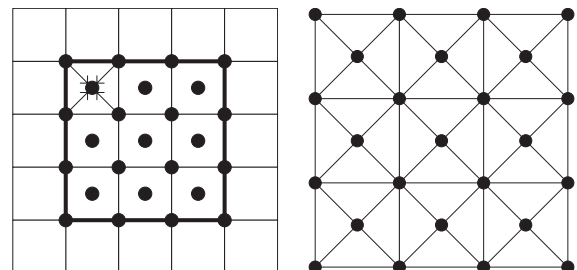
シミュレーション結果は 2.1 節の流体のシミュレーションによって得られるデータの概要で述べたように図 2.2(a) のような形で与えられる。これを等値線の描画のための三角形メッシュデータに変換するには以下の手順を用いた。

- 1) シミュレーションによって求めた煙密度のデータは図 2.2(a) のように格子状になっている。外周のデータは境界値である。
- 2) 描画領域に合わせたディスプレイ表示用格子のデータは、図 2.2(b) のようにその周囲 4 点のシミュレーションデータの平均より求める。格子点の煙密度を用いて、輝度値に変換する。輝度値の範囲は 0.0 ~ 1.0 である。
- 3) 四角形の格子を三角形のメッシュに分割するために、図 2.2(c) のように格子の頂点となる格子点の 4 つの値の平均より格子中心の値を求める。
- 4) 図 4.1(d) のように一つの格子に対して 4 つの二等辺三角形のメッシュに分割する。



(a) シミュレーションデータ

(b) 格子点データ



(c) 格子中心データ

(d) メッシュ

図 2.2 三角形メッシュの作成 (○: 煙密度, ●: 輝度)

あとは等値線レベルを設定し、このメッシュ一つ一つに等値線を導出することで全体の等値線が得られる。

2.3. 等値線の平滑化

等値線をそのまま描画すると画面に対する格子の大きさによっては滑らかでない角張った線になって見える可能性がある。そのため等値線を平滑化する必要がある。

等値線の平滑化には3次のBスプライン曲線を用いた。

2.3.1 Bスプライン曲線について

Bスプライン曲線は重み付け関数と制御点の座標を用いて曲線を描画する方法である。

3次のBスプライン曲線は4つの重み付け関数と4つの制御点座標によって構成され、重み付け関数 $N_1 \sim N_4$ 、制御点の座標 $P_1 \sim P_4$ とすると、曲線の関数 C は媒介変数 $t(0 \leq t \leq 1)$ を用いて次のように表される。

$$C(t) = N_1(t)P_1 + N_2(t)P_2 + N_3(t)P_3 + N_4(t)P_4 \quad (2.2)$$

$$N_1 = \frac{1}{6}(-t^3 + 3t^2 - 3t + 1)$$

$$N_2 = \frac{1}{6}(3t^3 - 6t^2 + 4)$$

$$(2.3)$$

$$N_3 = \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1)$$

$$N_4 = \frac{1}{6}(t^3)$$

媒介変数 t を0から1まで一定間隔で変化させることで図2.3のような曲線が描かれる。

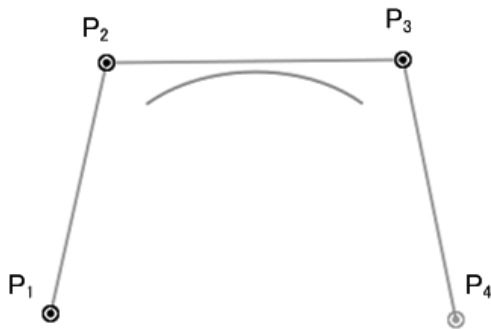


図 2.3 Bスプライン曲線 ($P_1 \sim P_4$: 制御点)

2.3.2 Bスプライン曲線による等値線の平滑化

導出された等値線のデータは、各メッシュに独立した線分のデータとして保存されるので、線分の探索および結合を行い、連続した座標データとしてのパスデータの1次元配列に変換しなければならない。

線分をパスデータに変換し、輪郭線を描画する手順は以下ようになる。

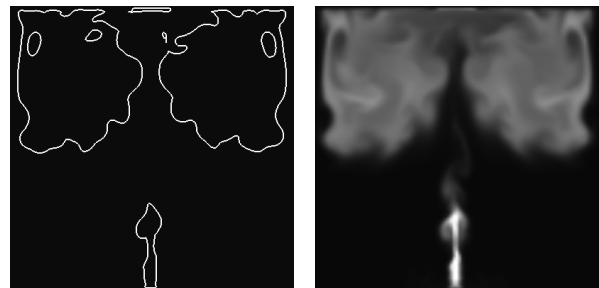
- 1) 線分が存在するメッシュを探し出す。
- 2) 線分の端点の一方の座標をパスデータ配列に格納し、もう一方の端点が存在する辺をもとに、その辺を共有するメッシュを次の探索メッシュとする。
- 3) 1つ前の探索メッシュと共有している辺上に存在する端点の座標をパスデータ配列に格納し、もう一方の端点が存在する辺をもとに、その辺を共有するメッシュを次の探索メッシュとする。
- 4) 3)を繰り返し、次の探索メッシュの候補が無くなるまたは探索始点のメッシュに戻ってきたら探索を終了する。
- 5) パスデータ配列をもとにBスプライン曲線を用いて輪郭線を描画する。
- 6) 1)に戻り、すべてのメッシュが探索済みになるまで繰り返す。

ここで注意しなければならない点は、輪郭線は閉じたもの（輪になっているもの）と開いたものの2つが存在することである。Bスプラインによる曲線の描画は閉じた曲線、開いた曲線の両方に対応しているが、線分の探索の際に開いた輪郭線の場合は始点が輪郭線の途中になってしまうことによりパスデータが途切れてしまう場合がある。この問題に関しては、領域の境界の辺に線分の端点を持つメッシュを先に全て探索してしまうことで解決できる。

2.4. 輪郭線描画の結果

輪郭線を描画した結果が図2.4(a)である。また、図2.4(b)が比較用に各メッシュに対してメッシュ頂点の輝度のデータを利用し、OpenGL標準の三角形描画によってグラデーションのかかった三角形を描画することで煙の表現したものである。

図2.4(a)における輪郭線は、(b)において輝度値が0.4を示す部分、つまり輝度値が等値線レベルと同じ値を示す点同士を結んだ線となっている。



(a) 煙の輪郭線の描画

(b) 煙の描画

図 2.4 輪郭線の描画結果
(等値線レベル 0.4, 400 ステップ目)

3. 輪郭線の加工

2章では輪郭線を見た目に忠実に描画することを目的としてきたが、ここでは煙の輪郭線に特殊な効果を加えることで煙の特徴を際立たせる手法を検討した。

具体的には、等値線の平滑化に用いるBスプライン曲線の代わりに、パスデータ化された等値線に対して速度場ベクトルを用いたエルミート曲線によって輪郭線を波打たせるように描画することで、煙の輪郭線と移動速度・方向の両方を同時に視覚化することを試みた。

3.1. エルミート曲線

エルミート曲線は重み付け関数と制御点の座標、そして制御点におけるベクトルを利用して曲線を描画する方法である。

3次エルミート曲線においては1区画の曲線の描画に対して制御点2つ、それに付随するベクトル2つ、そして重み付け関数4つで構成される。制御点に対する重み付け関数 $N_1 \sim N_2$ 、ベクトルに対する重み付け関数 $M_1 \sim M_2$ 、制御点の座標 $P_1 \sim P_2$ 、制御点におけるベクトル $R_1 \sim R_2$ すると、曲線の関数 C は媒介変数 $t (0 \leq t \leq 1)$ を用いて次のように表される。

$$C(t) = N_1(t)P_1 + N_2(t)P_2 + M_1(t)R_1 + M_2(t)R_2 \quad (3.1)$$

$$\begin{aligned} N_1 &= 2t^3 - 3t^2 + 1 \\ N_2 &= -2t^3 + 3t^2 \\ M_1 &= t^3 - 2t^2 + t \\ M_2 &= t^3 - t^2 \end{aligned} \quad (3.2)$$

一見Bスプラインと比較して重み付け関数が異なっているだけに見えるが、 R_1 および R_2 がベクトルである点が大きく異なる。以上の式を用いてエルミート曲線を描画すると図 3.1 のようになる。

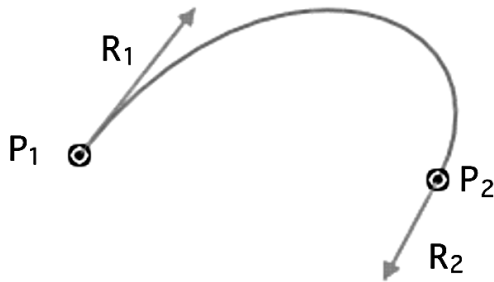


図 3.1 エルミート曲線

図 2.5 からわかるように、ベクトル R_1 および R_2 は曲線の制御点における接線方向に伸びていることがわかる。また、ベクトルの大きさは曲線の曲がり具合に影響を与え、ベクトルが大きいほどその方向に向かって大きく引っ張られるように湾曲する。

3.2. エルミート曲線の応用

本研究ではこのエルミート曲線を利用して煙の輪郭線に特徴づけを行う手法として以下のような検討を行った。

例えば、図 3.2 のようなBスプラインによる平滑化が行われた輪郭線が得られたとする。

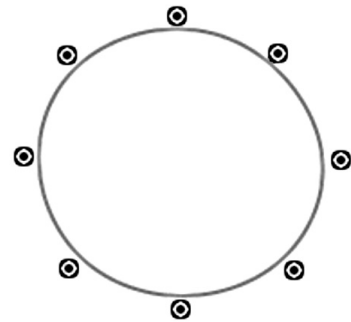
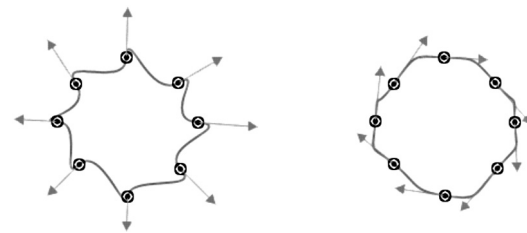


図 3.2 Bスプラインによる輪郭線の例

この状態では確かに平滑化が行われているためきれいな曲線が描画できているが、静止画の状態ではこれが時間変化していくとどのような形状に変化をしていくのかは読み取ることが出来ない。そこで、エルミート曲線を用いると、制御点におけるベクトルによって図 3.3 のような違いが現れてくる。



(a) 外側に向いている場合

(b) 回転するように向いている場合



(c) ベクトルが小さい場合

図 3.3 エルミート曲線におけるベクトルの違いによる輪郭線の変化の例

図 3.3 はいずれも制御点の位置は同じでベクトルの向きおよび大きさのみを変えたものである。図 3.3(a) は (b)(c) と比べて輪郭線が大きく波打っていることが分かる。このことを煙のシミュレーションに当てはめると、煙が勢い良く拡散または収束していく場合にその輪郭が大きく波打つ効果が表れるということになる。このことより、煙の輪郭線の静止画においてはその変化が予想できるようになり、動画においてはより特徴的な挙動が期待できる。

3.3. 煙の輪郭線への適用

輪郭線のパスデータは B スプラインによる平滑化のときのものがそのまま使えるが、それに加えて制御点におけるベクトルのデータを用意する必要がある。

ベクトルはパスデータおよび速度場データを利用して導出する。以下はその手順である。

- 1) 図 2.2(a) から (b) への変換と同様、シミュレーションデータ 4 点の平均より速度ベクトル場の格子点データを作成する。
- 2) パスデータの座標を一つ読み出し、どこの格子の内側に属するかを探索する。
- 3) 図 3.4 のように格子の 4 点の座標 $P_1 \sim P_4$ とその速度ベクトル $v_1 \sim v_4$ および制御点座標 P_c を用いて制御点座標での速度ベクトル v_c を導出する。

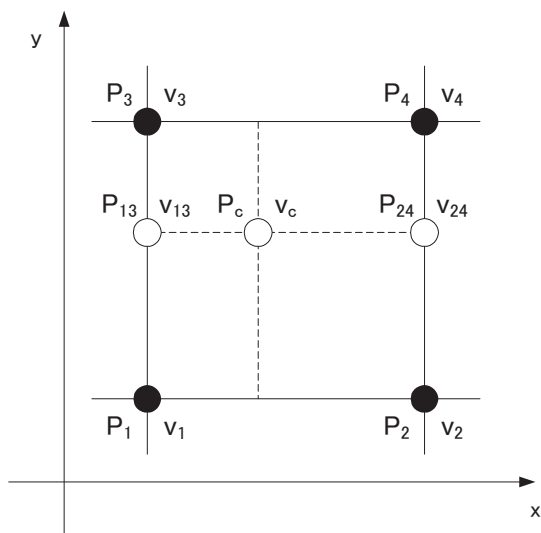


図 3.4 速度ベクトルの補間

v_c を直接求めるのは困難なので、まず y 方向の補間を行って v_{13} と v_{24} を導出する。それらはそれぞれ以下の式で求められる。

$$V_{13} = \frac{(P_{cy} - P_{1y})}{(P_{3y} - P_{1y})} \cdot (V_3 - V_1) + V_1$$

$$V_{24} = \frac{(P_{cy} - P_{2y})}{(P_{4y} - P_{2y})} \cdot (V_4 - V_2) + V_2$$
(3.3)

次に x 方向の補間を行って v_c を導出する。ここで x 方向の座標に関して $P_{13x} = P_{1x} = P_{3x}$ および $P_{24x} = P_{2x} = P_{4x}$ であるので、

$$V_c = \frac{(P_{cx} - P_{1x})}{(P_{2x} - P_{1x})} \cdot (V_{24} - V_{13}) + V_{13}$$
(3.4)

とすることで制御点における速度ベクトルが求まる。

これをすべてのパスデータの座標に対して行い、パスデータの座標数と同数の速度ベクトルデータを計算する。

- 4) 求めた速度ベクトルデータとパスデータを用いてエルミート補間を行う。

これでエルミート補間による曲線描画は可能になったが、速度ベクトルのスケールが小さい場合効果が分かりづらく、またスケールが大きすぎる場合表示が乱れてしまう。そこで、式 3.1 の速度ベクトルを含む項に係数を新たに追加してそのスケールを調整できるようにする必要がある。それを踏まえて式 3.1 を書き換えると次のような式になる。

$$C(t) = N_1(t)P_1 + N_2(t)P_2 + (M_1(t)R_1 + M_2(t)P_2) \cdot C$$
(3.5)

上式における C がスケール調整用の係数である。 C を大きくするとベクトルのスケールが大きくなり、曲線の波打ちが大きくなる。

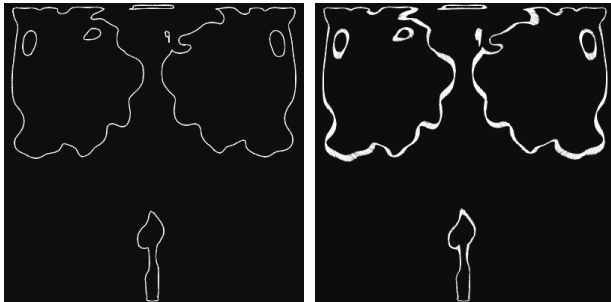
また、波打ちの間隔（山の数）自体を調整するには、パスデータを n 個おきに読み込んで制御点の数自体を減らすことで実現できる。ただしデータを読み捨てているということなので元の形状からは遠ざかってしまうという欠点がある。

3.4. エルミート曲線による描画の結果

エルミート補間による結果を描画したものが図 3.5 である。図 3.5 はそれぞれ速度ベクトルのスケール係数 C とパスデータのステップ幅 n (n 個ごとにデータを読み込み、それ以外は破棄) を変えて 3 パターン描画したものである。

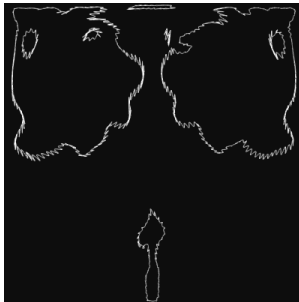
図 3.5 のうち (a) はスケールが小さすぎるため、輪郭線の波打ちが全く表れていない。(b) はスケールを大きくしたため波打ちの効果は表れているが、波の間

隔が狭過ぎて潰れて見える。(c)では波の間隔を調整し、ちょうど良い効果が得られた。このように、より良い効果を得るにはベクトルスケールの調整と波の間隔の調整が重要である。



(a) スケール C=1
ステップ幅 n=1

(c) スケール C=50
ステップ幅 n=1



(b) スケール C=50
ステップ幅 n=5

輝度値の範囲：0.0 ~ 1.0
等値線レベル：0.4
シミュレーション
：400 ステップ目

図 3.5 エルミート曲線による輪郭線の描画

4 結論

本研究においては B スプライン曲線を用いて輪郭線をきれいに描画する手法と、エルミート曲線を用いて輪郭線を特徴的に描画する手法という 2つの手法について検討した。その結果以下のことが分かった。

- (1) 煙の輪郭線の描画は「流体シミュレーション」「等値線の導出」「等値線の平滑化および変形」の 3つのステップに分けることができる。
- (2) 上記の 3ステップのうち「等値線の平滑化および変形」で輪郭線の形状に与える影響を決定することができる。
- (3) エルミート曲線を用いることで煙の輪郭線と煙の輪郭線の変化を同時に視覚化することができる。

これらのことより、何か輪郭線に特徴を持たせるのは最終的な曲線描画の段階で行えることが分かった。また、エルミート曲線による輪郭線の描画に関して次のような課題が見つかった。

- (1) 波の間隔を調整する手段を本研究のようなデータの読み飛ばしで行うと、段階的な調整しかできない。パスデータを再構成することで任意の間隔に調整するような手法を検討する必要がある。
- (2) エルミート曲線は対称ではなく、例えば図 3.3(a)は制御点・ベクトルのどちらもほぼ線対称かつ点対称な図形であるにも関わらず、それを使用して描かれる曲線は線対称ではない。そのため、対称化する場合には補間曲線の改良が必要である。

参考文献

- [1] 白山晋,「計算力学レクチャーシリーズ 8 知的可視化」,丸善株式会社, 2006.5. pp.47-55.
- [2] James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes,「コンピュータグラフィックス 理論と実践」,オーム社, 2001.3, pp.471-495.