

2次元テクスチャを用いた 炎の簡易表示の試み

大島 登志一 * 板橋 秀一 **

筑波大学 * 理工学研究科 ** 電子・情報工学系

炎は、ある種の神秘性を有しているらしく、人の心に様々な感情を呼び起こす特殊な存在であるということができる。したがって、これをコンピュータ・グラフィックス(CG)の情景の中に表示可能にすることによって、CGが潜在的に持つ表現力をさらに豊かにすることができると期待される。しかしながら、純粹にアルゴリズムによってリアルな炎を再現できる一般的な手法は、いまだに確立されていない。筆者等は、たまたま2次元テクスチャを用いて炎の簡単な表示を試みたところ、その単純な処理から期待される以上に、炎の外観を良く表わすことを見出した。本報告では、2次元テクスチャを用いた炎の動的表示のための単純な手法(仮称、2次元テクスチャ法)を紹介する。

Easy Representation of Flame by Using 2-Dimensional Texture

Toshikazu OHSHIMA * Shuichi ITAHASHI **

* Master's Program in Science and Technology, Graduate School, Univ. of Tsukuba

** Institute of Information Sciences and Electronics, Univ. of Tsukuba

1-1-1, Tennodai, Sakuramura, Niharigun, Ibarakiken, 305 Japan

Form of flames often seems to be mysterious. Flame may be peculiar existence among other natural phenomena and objects, since it tends to remind us of various kinds of feeling. Thus, if it becomes possible to represent flames in computer generated imagery, it must improve further in its power of expression. A method of generating imagery of satisfactory real flames, however, has not been completed yet. The authors have tried to represent flames easily using 2-D texture, and it has been found that flames can be represented much better in this way than it could be expected. This report describes an easy way to represent flames using 2-D texture.

1. はじめに

筆者らは、炎の表示について興味を持っている。人類の歴史は火と共にあると言っても過言ではないほど、太古の昔から、火は、人の暮らしとあらゆる面で深く関わり、人々に恩恵を与えると同時に厄災をも与えてきた。火を神聖視する思想が古くから広く見られ、祭りや宗教的儀式において、火は、しばしば象徴的な意味を持たされている。このような時代を越えた関わりを通じて、炎は、人の心に様々な感情を呼び起こす特殊な存在になっている^{(6) (7)}。したがって、これをCG画像中表示可能にすることによって、CGが潜在的に持つ表現力をさらに豊かにすることができると期待される。

筆者らは、3次元乱流中の何らかの分布として表わされる雲・煙・炎等の対象を表示するための一般的かつ効果的な手法を模索する過程で、たまたま、2次元テクスチャを用いて炎の簡単な表示を試みたところ、その単純な処理から期待される以上に、炎の外観を良く表わすことを見出した。本報告では、この2次元テクスチャを用いた炎の動的表示のための単純な手法（仮称、2次元テクスチャ法）を紹介する。本手法は、一般性には欠けるものの、暖炉の火などの小道具として、合成された情景に趣を添えるのに役立ちそうである。

2. 炎の表示の現状

炎をはじめ雲や煙といった対象は、乱流中で刻々と変化する何らかの3次元分布として存在する。特に炎の場合は、その複雑な形状の激しい変化こそが炎としての特徴であり、それをいかにモデル化するかが重大な問題である。一般に、自然現象のシミュレーションを行なうには、当該専門分野における研究成果に頼らなければならないが、炎に関しては、現在の燃焼学において解明されていない問題が多く、乱れの激しい炎の形態のシミュレーションは今のところ不可能である⁽⁸⁾。流体力学における乱流の数値シミュレーションも、簡単な境界条件下でのみ可能であるに過ぎないのが現状であり⁽⁹⁾、この流用は難しい。いずれにせよ、この種の数値シミュレーションは、計算コストが大変高く、目的にもよるが、アニメーション制作には实际的でない。代わりに、炎の視覚的特徴を手続き的・確率論的に再現できるような手法が望ましい。現在、炎を含む画像を生成するための手段として、2次元ペイント・システムやビデオ・シンセサイザーによる作画・合成が一般に採用されているが、純粹にアルゴリズムによって炎をリアルに再現できる一般的な手法は、まだ確立されていない。

アルゴリズムによる炎の表示の例としては、Reevesのparticle system⁽³⁾によるものと、Perlinのソリッド・テクスチャ⁽⁵⁾によるものがある。particle systemは、炎を多数の発光性の微粒子からなる系であるとみなし、系内での各微粒子の生成・移動・消滅を確率過程に基づいて制御し、それらの飛跡を表示することによって、動的に炎を構成するものである。これは、SF映画⁽¹⁰⁾の惑星炎上シーンのために開発されたもので、充分遠距離から観察した大規模な火災や、花火の表現には向いている。しかしながら、身近かに見られる焚き火程度の小規模な炎は表現できない。一方、Perlinは、ソリッド・テクスチャを導入し、その例の一つとして、太陽のコロナを表示した。この例では、一点から放射状に伸びる単調な光の帯を $1/f$ ノイズで攪乱し、その中心を含む断面を表示することによってコロナを表わしている。しかしながら、この手法は、普通の意味での炎を表示できるに到っていない。どちらの手法も、乱流をなす炎（あるいはガス）の流れを統計的に処理している。

3. 2次元テクスチャ法

2次元テクスチャ法では、単色の2次元基本テクスチャを加工して色彩を持った原画像を生成し、それを都合良く変形することで炎を表現する。基本テクスチャは、このモデルにおける炎の実体として、その部分的特徴を表わし、マッピング関数による座標系の変換は、流れの全体的な乱れを表わす。

以下、その概要を説明する。条件を特に指定していない関数は全て、インプリメントの際、望むような炎が得られるように各々自由に決めて良い。

3.1 基本テクスチャ

まず、基本となる2変数のテクスチャ関数 $T(p, q)$ を用意する。これは、純粹に数学的に定義されたものでも、ルックアップ・テーブルを使用したものでも構わないが、時刻によって、テクスチャ関数の参照領域が q 軸上を移動することになるので、変数 q については、充分広い範囲で定義されている必要がある。ルックアップ・テーブルの場合、変数 q に対応するインデックスを循環させたり、インデックスが整数にならないときテーブルの値を補間するなどの工夫が必要である。変数 p については、 $0 \leq p \leq 1$ で定義されていればよい。また、関数の値域は、 $[0, 1]$ とする。

図1.に示すように、テクスチャ関数のパラメータ空間 (p, q) 上にテクスチャ座標系 (u, w) を設定し、

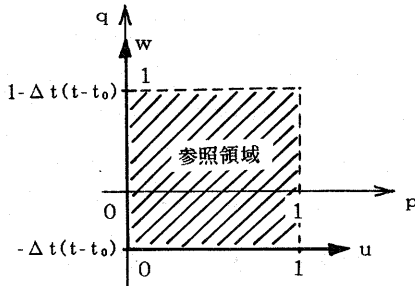


図1. テクスチャ空間とテクスチャ関数の参照領域

$0 \leq u, w \leq 1$ の範囲でテクスチャ関数を参照する。この参照領域のテクスチャが炎の原形となる。初期時刻を t_0 、フレーム間の時間刻みを Δt とすると、時刻 t における座標系 (u, w) から座標系 (p, q) への変換は、式(1)によって与えられる。

$$(p, q) = (u, w - \Delta t(t - t_0)) \quad \dots(1)$$

したがって、時刻 t のテクスチャ座標 (u, w) における基本テクスチャの濃度値 T_0 は、式(2)で与えられる。

$$T_0 = T(u, w - \Delta t(t - t_0))$$

ただし $0 \leq u, w \leq 1$...(2)

$$0 \leq T(p, q) \leq 1$$

このとき、式(3)に示す範囲がテクスチャ関数の参照領域となる。

$$0 \leq p \leq 1$$

$$-\Delta t(t - t_0) \leq q \leq 1 - \Delta t(t - t_0) \quad \dots(3)$$

基本テクスチャは、炎の視覚的特徴を直接表わすものであり、炎の表示に関する問題の本質と関係している。したがって、本手法の性格上、その構成法について、特に理論的な正当性を求めることはしない。ただ一つの条件としては、特定の値が部分的に集中しないこと、つまり平坦な部分のないことである。

3.2 基本テクスチャの加工

基本テクスチャの加工は、閾値処理、濃度補正、彩色の3段階の処理から成る。

3.2.1 閾値処理

炎の断片が上昇するにつれ、しだいに小さくなり消えていく様子を表現するために、単調増加関数 $V(w)$ により垂直方向の距離 w に応じて閾値を設定し、閾値以下を切り捨てる。また、左右の端が不自然に切れないように、水平方向にも同様の処理を行う。水平方向の閾値関数を $H(u)$ とすると、テクスチャ座標 (u, w) における閾値 L と、閾値処理後の濃度値 T_1 は、それぞれ式(4)、(5)で与えられる。 $V(w)$ と $H(u)$ は、おおよそ図2に示すような関数である。

$$L = 1 - (1 - V(w))(1 - H(u))$$

ただし $0 \leq V(w), H(u) \leq 1$...(4)

$$(0 \leq u, w \leq 1)$$

$$T_1 = \begin{cases} 0 & T_0 \leq L \\ T_0 - L & T_0 > L \end{cases} \quad \dots(5)$$

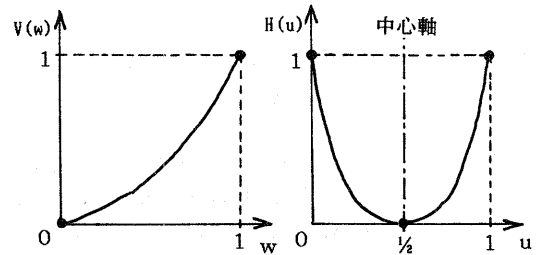


図2. 閾値関数 (a) $V(w)$ (b) $H(u)$

3.2.2 濃度補正

閾値処理しただけのテクスチャは、曖昧な断片から成っている。ところが、実際の炎の断片は、比較的くっきりした境界を持っているように見える。そこで、図3に示すように、入力 s が最大値1より低い値 s_p で出力が最大となるような関数 $C(s)$ を用意して濃度補正を行う。補正されたテクスチャの濃度値 T_2 は、式(6)で与えられる。

$$T_2 = C(T_1)$$

ただし $0 \leq C(s) \leq 1$...(6)

$$(0 \leq s \leq 1)$$

図3で、入力 s_p において出力がピークに達した後、傾きを逆転させているが、これによって、画像が平坦に見えるのを、ある程度修正することができた。

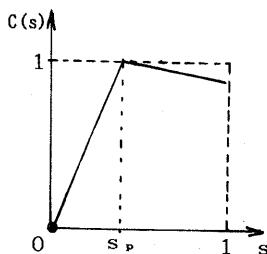


図3. 濃度補正関数 $C(s)$

3.2.3 彩色

単色のテクスチャを彩色するために、図4のような関数 $R(s)$, $G(s)$, $B(s)$ を用いて、補正したテクスチャの濃度値 T_z に応じて色ベクトル (r, g, b) を決定する。

$$(r, g, b) = (R(T_z), G(T_z), B(T_z)) \quad \cdots(7)$$

$$\text{ただし } 0 \leq R(s), G(s), B(s) \leq 1 \\ (0 \leq s \leq 1)$$

身近に見られる焚き火やろうそくなどの黄色い炎は、熱せられた炎中の煤が黒体放射によって輝いて見えるものである。今回の実験では、黒体放射にこだわらず、赤から黄白色へと感覚的に配色したが、比較的良好な結果を得ている。

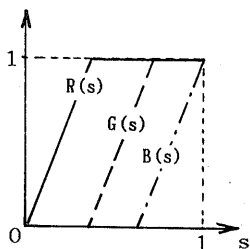


図4. 色ベクトル成分関数 $R(s), G(s), B(s)$

3.3 マッピング

図5のように、テクスチャ空間の原画像を、世界座標系中に定義した2次元の直交座標系 (x, y) の $0 \leq x, y \leq 1$ の領域にマッピングする。座標系 (x, y) から座標系 (u, w) への写像を定義するマッピング関数は、炎の上昇速度および加速度と、左右の炎と大気との境界のゆらぎを直観的な形で導入し、決定される。

なお、アンチエイリアシングの問題と共に、スクリーン座標系から座標系 (x, y) までの変換については、ここでは触れない。

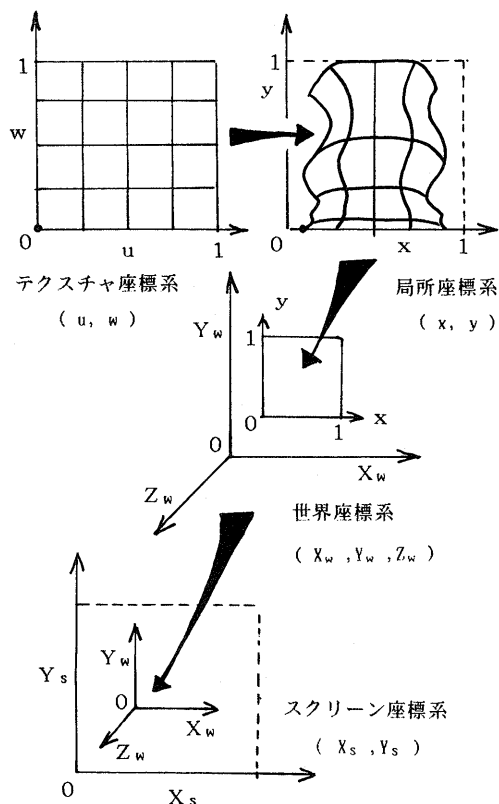


図5. 各座標系の包含関係

マッピング操作を説明する。

炎が加速しつつ上昇する様を表わすために、まず式(9)のようにパラメータ w を決める。

$$w = (1 - k) \sqrt{y} + k y \quad \cdots(9)$$

$$\text{ただし } 0 \leq k \leq 1$$

k	初速度	加速度
小	小	大
大	大	小

表1. 式(9)の定数 k と炎の速度

式(9)で、定数 k は、表1.に示すように、炎の上昇速度を制御するパラメータである。

今、パラメータ x を式(10)のような x' で置き換えて、炎を中心軸 $x = 1/2$ で左右対称にして考える。

$$x' = 2|x - 1/2| \quad \cdots(10)$$

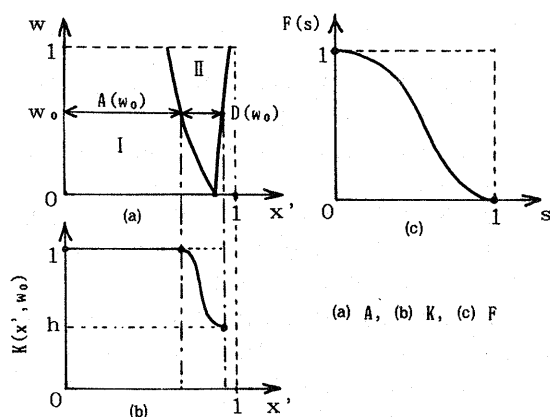


図6. 式(1)の関数群

炎を図6.(a)のように、性質の異なる2つの領域Ⅰ・Ⅱから構成する。領域Ⅰは、炎が一樣な加速度で上昇して行く領域、領域Ⅱは、炎と周囲の空気との境界が大きく乱れを生じている領域で、その上昇加速度は、端に行くほど小さくなるものとする。

高さ $w = w_0$ における領域Ⅰ・Ⅱの幅をそれぞれ、関数 $A(w_0)$ 、 $D(w_0)$ によって与える。このとき、中心 $x' = 0$ から端 $x' = A(w_0) + D(w_0)$ へ向かう水平断面での上昇速度分布 $K(x', w_0)$ は、式(1)で与えられ、図6.(b)のようになる。

$$K(x', w_0) = \begin{cases} 1, & x' \leq A(w_0) \\ h + (1-h) F\left(\frac{x' - A(w_0)}{D(w_0)}\right), & A(w_0) < x' \leq A(w_0) + D(w_0) \end{cases}$$

$$A(w_0) < x' \leq A(w_0) + D(w_0)$$

ただし $0 \leq h \leq 1$

$$D(w) = 0, w = 0$$

$$D(w) > 0, w > 0$$

$$A(w) > 0$$

$$A(w) + D(w) \leq 1 \quad \dots(1)$$

式(1)で、定数 h は、最低速度の最高速度に対する比を表す。そして、関数 F は、図6.(c)のように、最高速度から最低速度へ推移する部分を表す。

そして、 x' による上昇速度の上昇速度の変化を反映させるために、式(9)で与えられたパラメータ w の値を調整する。式(9)の w を w_p と表して、式(12)により、新しい w を決める。

$$w = w_p / K(x', w_p) \quad \dots(12)$$

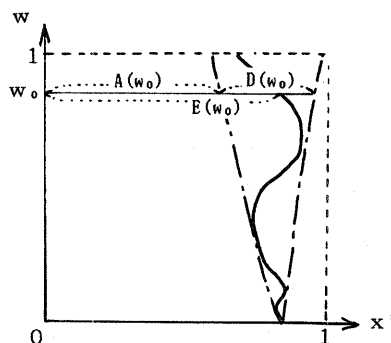


図7. 境界のゆらぎ

最後に、炎のゆらぎを得るために、図7のように、領域Ⅱの範囲で左右の境界にゆらぎを与えた上で、パラメータ u を決定する。

まず中心軸の右側 ($\frac{1}{2} \leq x$) を考え、パラメータ x と u を、それぞれ式(3)、(4)のように x' と u' で置き換える。

$$x' = 2x - 1 \quad \dots(3)$$

$$u' = 2u - 1 \quad \dots(4)$$

右側の境界のゆらぎを表す関数を $P(t)$ とし、右側の境界の形状を表す関数 $E(w, t)$ を式(5)のように決める。

$$E(w, t) = A(w) + D(w)P(w - \Delta t(t - t_0)) \quad \dots(5)$$

ただし $0 \leq P(t) \leq 1$

u' を式(6)で求める。

$$u' = x' / E(w, t), \quad x' < E(w, t) \quad \dots(6)$$

式(4)と式(6)より、 u が求まる。

中心軸より左側 ($x < \frac{1}{2}$) の場合、パラメータ x と u を、それぞれ式(7)、(8)のように x' と u' で置き換え、 $P(t)$ を左側の境界のゆらぎを表すものに置き換えて、右側と同様に、 u を決める。

$$x' = 1 - 2x \quad \dots(7)$$

$$u' = 1 - 2u \quad \dots(8)$$

以上で、座標系 (x, y) から座標系 (u, w) への写像が決定された。

4. 炎の表示例

実験に使用したテクスチャ関数は、雲の表示⁽⁴⁾のために作られたものに、一部手を加えたもので、式(20)によ

って定義される。

$$T(x, y) = \frac{\sum C_i^2 (\cos \Phi \cos \Psi + 1)}{\sum C_i^2} \quad \dots (20)$$

$$\begin{aligned} \Phi &= F_{xi} x + P_{xi} \\ \Psi &= F_{yi} y + P_{yi} \\ F_{xi+1} &= 2 F_{xi} & F_{x0} &= 16 \\ F_{yi+1} &= 2 F_{yi} & F_{y0} &= 26 \\ P_{xi} &= \frac{1}{2} \pi \sin(\frac{1}{2} F_{yi} y) \\ P_{yi} &= \frac{1}{2} \pi \sin(\frac{1}{2} F_{xi} x) \\ C_{i+1} &= 0.707 C_i & C_0 &= 1 \end{aligned}$$

画像1.に式(20)により生成されたテクスチャを示す。

画像1.を閾値処理した結果を画像2.に示す。

画像2.を濃度補正した結果を画像3.に示す。

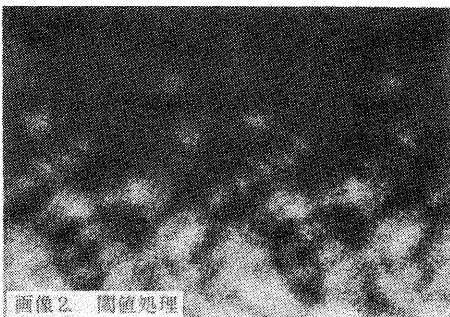
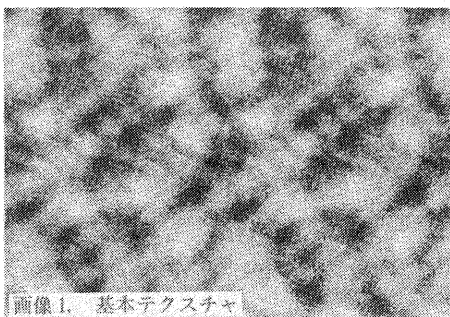
画像3.を彩色した結果を画像4.に示す。

完成した炎の出力例を画像5.～8.に示す。

なお、環境は、以下に示す通りである。

使用言語 FORTRAN77
 ホスト・コンピュータ FACOM S-3300
 グラフィック・ディスプレイ SEILLAC-3
 データ転送 RS-232C 9600bps

計算時間は、256 × 256 画素で、一枚約3分である。



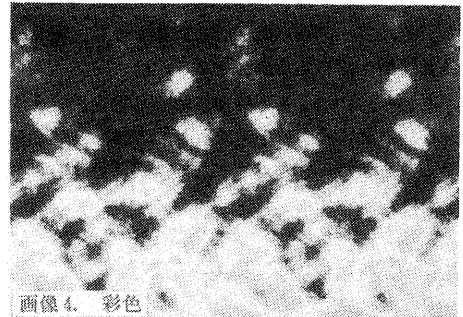
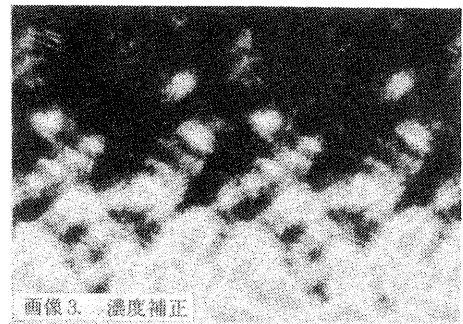
5. むすび

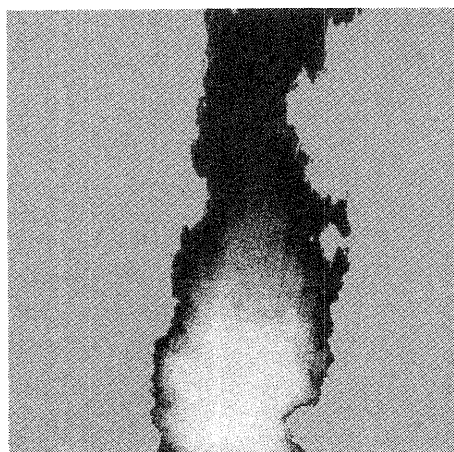
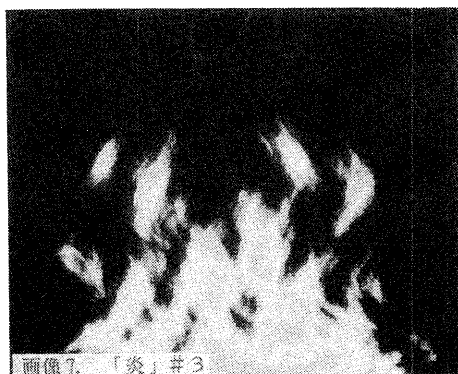
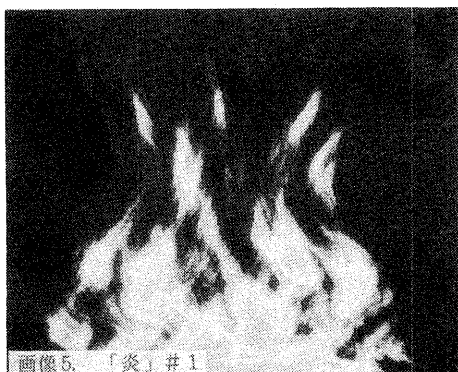
2次元テクスチャを用いた単純な炎の表示手法を紹介した。モデルとしては不十分なものであるが、計算コストの低さを考慮すれば、比較的良好な出力結果が得られた。同様の手法によって、火の粉や煙を生成し、今回の炎と一緒に表示することも可能であると思われる。炎を表示するための満足すべき一般的な手法が開発されていない現在、本手法は、用途によっては想像以上に役に立つかも知れない。今後、本手法に関しては、今回触れなかった3次元情景中での実際の表示の問題について検討すると共に、さらに手を加えて、それなりの使い方ができる手法にまとめたいと考えている。

現在、筆者等は、炎を3次元空間における煤の温度分布および濃度分布として定義し、その流れをフラクタル的なゆらぎの場によって攪乱することにより、炎と煙を同時に表現しようと試みている。画像9.に2次元の予備実験による出力結果を示す。今後は、炎の表示のための正当なモデルの構築と、その評価方法について研究を進めて行きたい。

謝辞

本研究を進めるにあたり、燃焼実験や出力画像の評価に協力して頂いた、パターン認識研究室の諸氏に感謝する。





画像9. 炎と煙の表示

参考文献

- (1) Mandelbrot, B.B., "The Fractal Geometry of Nature", W.H. Freeman & Co. (1982).
- (2) Fournier, A., Fussell, D., and Carpenter, L., "Computer Rendering of Stochastic models", Comm.ACM 25, 6 (June 1982), 371-384.
- (3) Reeves, W.T., "Particle Systems-A Technique for Modeling a Class of Fuzzy Objects", SIGGRAPH'83, Computer Graphics 17, 3 (July 1983), 359-376.
- (4) Gardner, G.Y., "Visual Simulation of Clouds", SIGGRAPH'85, Computer Graphics 19, 3 (July 1985), 297-303.
- (5) Perlin, K., "An Image Synthesizer", SIGGRAPH'85, Computer Graphics 19, 3 (July 1985), 287-296.
- (6) 秋田一雄, 「火のはなし」、技報堂出版(1985)。
- (7) 秋田一雄, 「火のはなしⅡ」、技報堂出版(1986)。
- (8) 金原寿郎, 「応用物理学選書5: 気体の燃焼物理」、裳華房(1985)。
- (9) 巽友正, 「乱流現象の科学」、東大出版会(1986)。
- (10) Paramount, "Star Trek II: The Wrath of Khan" (film), (June 1982).