

## 水彩パース図風レンダリング法

中川 大介<sup>+</sup> 藤本 忠博<sup>+</sup> 村岡 一信<sup>++</sup> 千葉 則茂<sup>+</sup>

<sup>+</sup>岩手大学工学部 <sup>++</sup>東北工業大学

naka@cis.iwate-u.ac.jp fujimoto@cis.iwate-u.ac.jp muraoka@tohtech.ac.jp nchiba@cis.iwate-u.ac.jp

本論文では、建築等の完成イメージ図などに用いられるパース図を取り上げ、3次元幾何モデルから水彩パース図風画像を自動生成する手法について提案する。本手法は、水彩パース図の特徴である、輪郭が描画される、建物については彩色のはみ出しが無い、樹木などの建物以外の対象は簡略化されて描かれることが多い、淡い色彩が用いられることが多い、などの点を考慮したアルゴリズムの構成となっている。

### Watercolor-Style Rendering of Perspective Drawings

Daisuke Nakagawa<sup>+</sup>, Tadahiro Fujimoto<sup>+</sup>, Kazunobu Muraoka<sup>++</sup> and Norishige Chiba<sup>+</sup>

<sup>+</sup>Iwate University, Faculty of Engineering, <sup>++</sup>Tohoku Institute of Technology

We propose a method for automatically generating watercolor-style perspective drawings from three-dimensional geometric models. The method would be useful in the architectural-design area. It employs several technical approaches that realize the following general features of real watercolor-style perspective drawings: the outlines and edges of objects are drawn, there is no overflow of watercolor in the coloring especially of buildings, another objects like botanical trees are rendered often in simplified drawing styles and transparent light colors are used in most cases.

**キーワード:** ノンフォトリアリスティックレンダリング, 水彩画, パース

**Keyword:** non-photorealistic rendering, watercolor, perspective drawings

#### 1. はじめに

近年、コンピュータグラフィックス(CG)において非写実的なレンダリング法の研究が盛んに行わ

れている[1-21]が、水彩パース図に注目したものは少ない。本報告では水彩パース図風画像を3次元幾何モデルから自動的に生成する手法を提



図1 実際のパース図[22]

案する。

これまで提案されている非写実的なレンダリング法に関する研究としては、ペン画[1-8]、油絵[9-11]、パステル画[12]、水彩画[9,13]調画像を得る手法について良く研究されてきている。また、これらの絵画調画像を得るアプローチとしては、写真などの2次元画像をベースとして絵画調画像に変換する手法[1,10,14,15]と、3次元幾何モデルをレンダリングして生成された3次元画像から絵画調画像を生成するという3次元CGとしてのレンダリング法[2-9,12,16-21]がある。さらに、手描きツールを提供するための筆と顔料(pigment)の振る舞いを模擬する手法に関する研究[12,13]、筆などのストロークの生成に関する研究[10-12]などがある。

本研究の手法は、3次元画像から生成するものである。建築等の建設分野においては、通常の3次元CGによる写実的なレンダリングが可能な現在においても、完成予想図として水彩パース図が好んで用いられており、このようなパース図がCADデータから自動的に得られるような技術が開発されれば非常に有用であると言える。

多くの水彩パース図では、対象物の輪郭線がペン等で描画され、その後彩色が施されている(図1参照)。輪郭線は対象物を鮮明にするために、特に建築物については直線で描かれることが多く、彩色は暖かみのある印象を与えるため、一般に淡い色で、特に建築物については筆の跡が目立たないように施されることが多い[22]。また、輪郭をはみ出した彩色が行われることは僅少である。この点において、絵の具のにじみの効果や筆の線を活かした通常の手描き風水彩画[13]とは異なる特徴をもつ(ただし、文献[13]で提案されている水彩の発色計算法はパース図においても有効である)。一方、手描き風パース図の製作を支援する実用的なソフトウェアとして Piranesi[23]などがあるが、3次元幾何モデルから水彩パース図風画像を自動生成するものではない。

本論文では、3次元幾何モデルから水彩パース図風画像を生成する手法を提案する。本手法では、水彩パース図の特徴を踏まえ、輪郭線描画と彩色の2段階で画像生成を行う。輪郭線は、3次元幾何モデルにレイトレーシング法をベースとした隠面消去法を適用し、ピクセル毎の可視面の奥行き値(Z値)、法線ベクトル、およびサーフェイス情報から輪郭を構成するピクセルを求めること

により定め、それをペン画風に描画する。同一物体の表面を構成するポリゴンの集合をサーフェイスと呼び、他のサーフェイスとはIDによって区別される。彩色のためには、まず、紙を表す画像を用意し、その上にRGB値と不透明度(0~1)を持つレイヤを重ねる。その際、初期値として、RGB値には通常のシェーディングモデルにより写実的にレンダリングしたCG画像を登録し、不透明度には0(完全透明)を登録しておく。そして、このレイヤ上で、不透明度を向上させる筆を物体形状に応じた(後述するように、本手法では輝度値のみ考慮した)ストロークに沿って走査させることにより、そのCG画像を浮き上がらせるというアプローチで彩色を行う。一般に、自動的な描画法では筆のコントロール、特に輪郭からの色のはみ出しを制御することは非常に難しいため、本手法では筆による色のはみ出しが起こらないアプローチを採用した。

また、水彩パース図では樹木などの添景や遠方の対象は簡略化して描かれることが多いため、本手法では、Z値の大きい輪郭線を簡略化する、下絵とする3次元CG画像にはZ値に応じた強さで平滑化を施す、また色彩をZ値に応じて変化させるなどのアプローチも用いている。

## 2. 建築物のレンダリング法

本手法では建築物のレンダリングは2つのステップで行う。

まず、幾何モデル(図2(a)参照)にレイトレーシング法を適用し、サーフェイス、Z値、法線ベクトルの情報から輪郭線を描画する(図2(b)参照)。次に、紙を表す画像を用意し、RGB値と不透明度(0~1)を持つレイヤを重ね、RGB値として通常のシェーディングモデルにより写実的にレンダリングしたCG画像を登録し、不透明度として0(完全透明)を初期設定する。このレイヤ上で、不透明度を向上させる筆を物体形状に応じたストロークに沿って走査させる(図2(c)参照)ことにより、CG画像を浮き上がらせる(図2(d)参照)。

### 2.1 輪郭線の描画

#### 2.1.1 基本手法

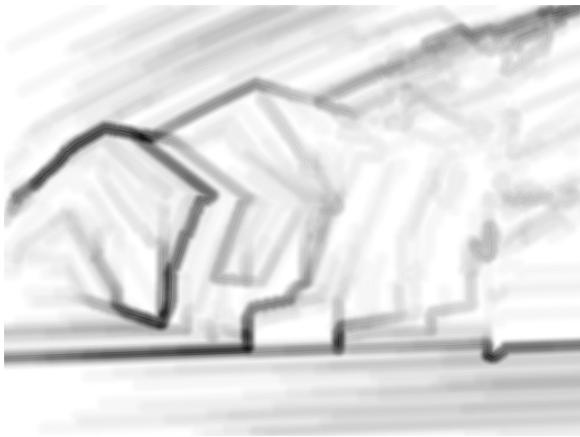
輪郭線の描画は、幾何モデルをレイトレーシング法によってレンダリングしながら、次の3つの条件



(a)シェーディング画像



(b)輪郭線の描画



(c)筆跡



(d)彩色処理により得られた画像

図2 本アルゴリズムの処理の流れ

- ・ 隣り合うピクセルの法線ベクトルのなす角が閾値以上である
  - ・ 隣り合うピクセルの Z 値の差が閾値以上である
  - ・ 隣り合うピクセルのサーフェイスが異なる
- のうちいずれかを満足するピクセルを検出し、それを輪郭線ピクセルとして登録する。

1番目の条件における閾値 (*angle*) は以下の式で求められる。

$$angle = smooth + zweight \cdot (180 - smooth) \quad (1)$$

$$zweight = \begin{cases} 0 & (zbuffer(x, y) < zmin) \\ \frac{zbuffer(x, y) - zmin}{zmax - zmin} & (zmin \leq zbuffer(x, y) \leq zmax) \\ 1 & (zbuffer(x, y) > zmax) \end{cases} \quad (2)$$

ここで、*smooth* は輪郭線を抽出する基本となる角度を調整するパラメタ、*zbuffer(x,y)* はピ

クセル(x,y)の Z 値、*zmin*, *zmax* はそれぞれ Z 値による輪郭線の簡略化の効果 (遠い物体ほど輪郭線が省略される) を調整するパラメタでデフォルトは Z 値の最小値と最大値である。なお、*zmin, zmax* は  $zmin < zmax$  を条件にユーザが任意に設定することもできる。

2番目の条件における閾値 (*zth*) は次式で求められる。

$$zth = zf + zweight \cdot (zb - zf) \quad (3)$$

ここで、*zf*, *zb* は輪郭線抽出を調整するためのパラメタで、それぞれ、最も手前に位置する物体、最も遠くに位置する物体に対する閾値である。

図3に *smooth* を変えた場合の輪郭線描画の例を示す。また、図4に Z 値による輪郭線の簡略化の効果を示す。

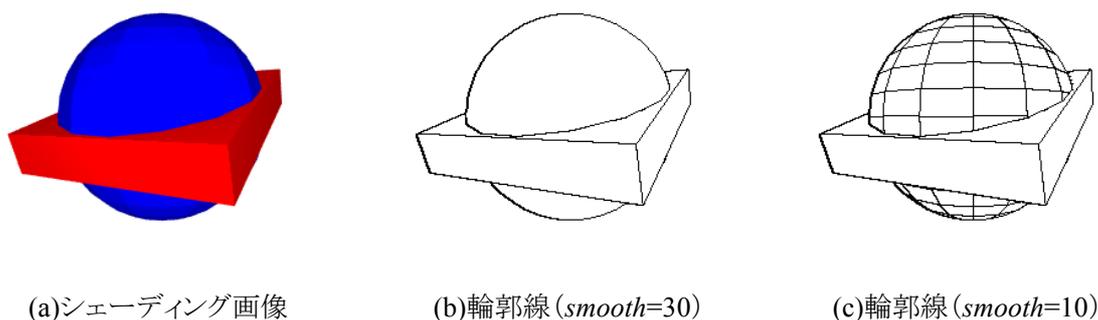


図3 輪郭線の例

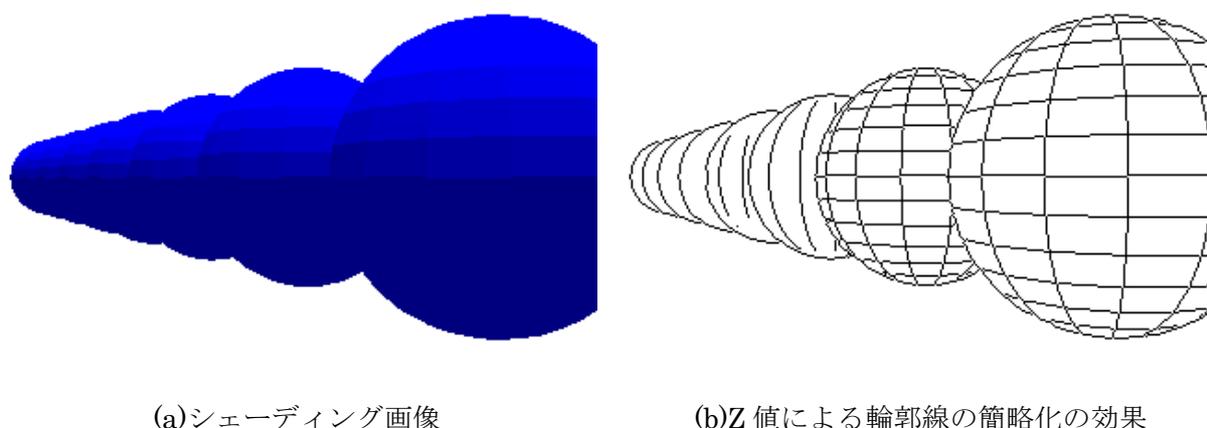


図4 輪郭線の簡略化

### 2. 1. 2 スーパーサンプリングを適用した手法

2. 1. 1による手法では、複雑な幾何モデルを用いた場合、エイリアシングが発生する(図5(a)(b)参照)。これを解決するために、1つのピクセルを格子状に細分割し、それぞれの格子点を仮想ピクセルとしてレイを放射しスーパーサンプリングを行うことによりアンタイエイリアシングされた輪郭線を求める。ピクセルを4×4格子に分けて16点にサンプリングした例を図5(c)(d)に示す。図5(a)(b)の解像度は640×480pixelsである。

### 2. 2 彩色処理

一般に水彩画では対象物の等濃度方向に沿って筆を走らせ彩色する。さらに、長いストロークで描く時は太い筆を使う。本アルゴリズムではこれらを考慮して、ストロークは光源ベクトルと法線ベクトルの外積の方向に走らせ、筆跡の幅は検出したストロークの長さに比例させる。ただし、筆跡の幅には有効範囲(最小幅から最大幅まで)を設け、長いストロークでも最大幅以上には広くせず、最

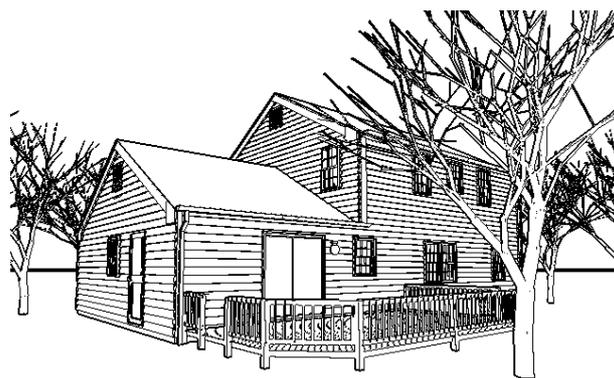
小幅より短くなるストロークは描かないようにしている。この手法によって、長く太い筆ストロークが優先的に描かれる(図2(c)参照)。

前処理として、レンダリング画像と紙を表現するための画像を用意する。レンダリング画像はレイトレーシング法により輪郭線情報を検出する際に同時に生成する。この画像の各ピクセルはRGB値と不透明度(0:0~1)を持ち、不透明度の初期値は0(完全透明)とする。彩色処理アルゴリズムは以下のものである。

ステップ1. 紙の画像を下層、レンダリング画像を上層のレイヤとして配置する。

ステップ2. 筆跡の最小幅  $B$  に上限最小幅  $B_{max}$  の初期値を決める。今回は  $B_{max}=20$ ピクセル幅とした。

ステップ3. 筆ストロークの開始点の候補となるピクセルについて、以下のステップ 3-1~3-3を行う。その候補ピクセルとして、最初は、画像の縦サイズ×横サイズ個のピクセルをランダム



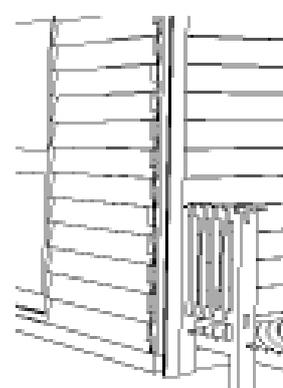
(a)基本手法



(b)基本手法(拡大)



(c)スーパーサンプリング



(d)スーパーサンプリング(拡大)

図5 複雑なモデルの輪郭線検出

に選び、その後、画像の左上のピクセルから走査線順に選んでいく。

ステップ3-1. 物体を構成するピクセルが見つかった場合、これを開始点として、筆ストロークを生成する(後述)。

ステップ3-2. ステップ 3-1 で生成した筆ストロークから筆跡の幅を求める(後述)。

ステップ3-3. ステップ 3-2 で求めた筆跡の幅が最小幅 $B$ より大きな場合、筆ストロークの周囲部分の不透明度を増加させる(後述)(図6参照)。

ステップ4. 最小幅 $B$ を $\Delta B$ だけ小さくする。その新たな最小幅 $B$ が下限最小幅 $B_{min}$ 以上である場合にはステップ3に戻り、 $B_{min}$ より小さくなった場合にはステップ5に進む。なお、今回は、 $\Delta B=1$ ピクセル幅、 $B_{min}=5$ ピクセル幅とした。

ステップ5. 紙( $Pa$ )とレンダリング画像( $I_{rend}$ )のレイヤを、上記で求めた不透明度( $O$ )によって次式で合成し、彩色画像( $I$ )を得る。

$$I = I_{rend}O + Pa(1 - O) \quad (4)$$

ここで、彩色処理アルゴリズムについて補足する。

ステップ3の補足: 筆ストロークの開始点をランダムな候補ピクセルのみから選ぶとすると、画像全体を覆うのに処理時間がかかってしまうため、一定個数のピクセルをランダムに選んだ後、左上端から画像をスキャンするという手法を用いた。

ステップ 3-1 の補足: 筆のストロークの生成は以下の手順で行う。

- (1) 物体を構成するピクセルが見つかった点をストロークの開始点とする。
- (2) 開始点に筆ストロークの先端を置き、次の

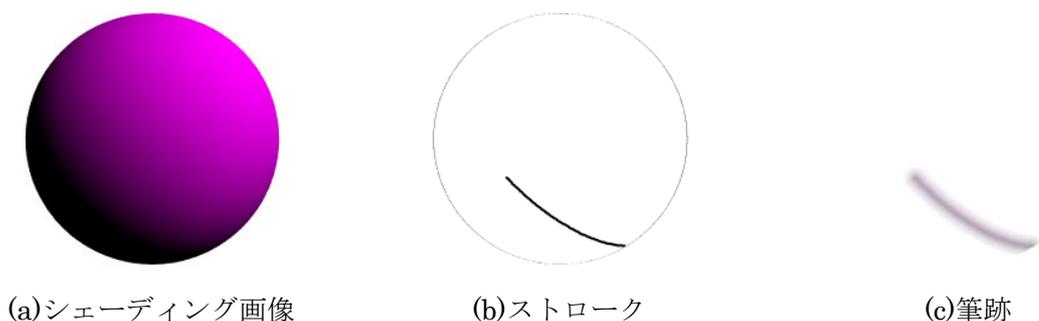


図6 筆跡の生成

(2-1),(2-2)を繰り返すことで、ストロークの終了点まで先端の位置を更新していく。

- (2-1) 先端の光源ベクトル  $L$  と物体の法線ベクトル  $N$  との外積を求める。このベクトルをスクリーンに投影したベクトルを正規化しその先端を、筆ストロークの新しい先端とする。
- (2-2) 筆ストロークの先端が、次の何れかの場合、その直前までの先端を筆ストロークの終点とする。
- 物体の無い領域にある。
  - スクリーンの外にある。
  - 先端位置の不透明度が閾値より高い。
- なお、閾値 ( $threshold$ ) は以下の式で求める。

$$threshold = base + (1.0 - p(I_{rend})) \cdot cont \quad (5)$$

ここで、 $base$  と  $cont$  は輝度値とコントラストを調整するパラメタ、 $p(x)$  は RGB 値  $x$  について NTSC 加重平均法により、輝度値を返す関数である。 $cont$  の値を上げると、レンダリング画像上の輝度値の低い(暗い)ところをよりこすり出す効果が得られる。

- (3) 開始点から終了点までのベクトルの集合を筆ストロークとする。

なお、手描き風の印象を与えるため、上の手順で得られた筆ストロークにノイズを付加する。

ステップ 3-2 の補足: 筆跡の幅はストロークの長さの  $\times 0.3$  とする。ただし、筆跡の最小幅  $B$  の上限最小幅  $B_{max}$  より大きい場合はその筆跡の幅を  $B_{max}$  とする。

ステップ 3-3 の補足: 不透明度の増加量はストロ

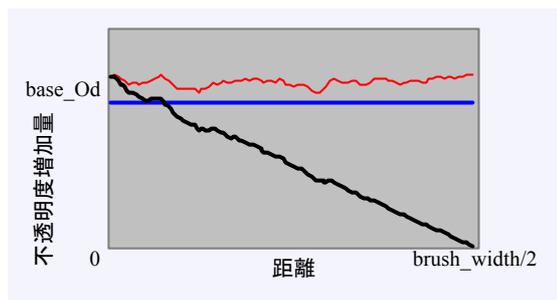
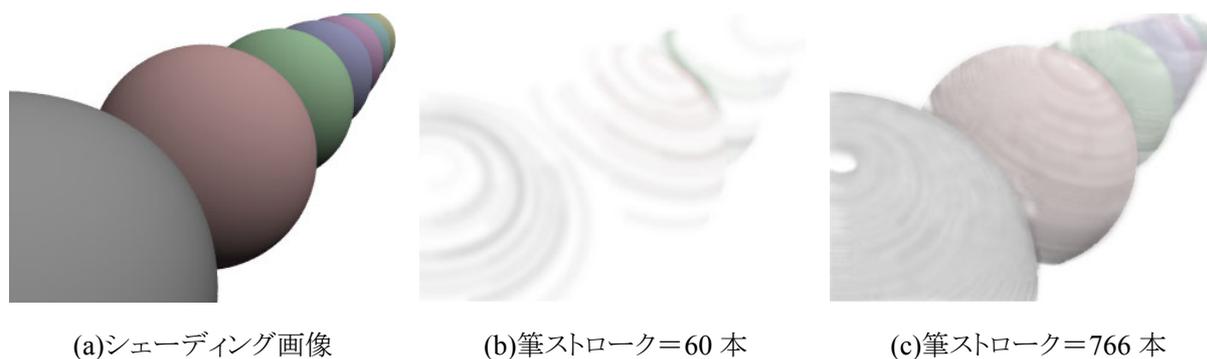


図7 距離と不透明度増加量

ークと各ピクセルの最短距離によって定める。距離と増加量との関係は、筆の不均一性を表現するためにノイズを入れ、ここでは図7のグラフで示すような不透明度を用いた。図で、青の線が基本的な不透明度のグラフで、それにノイズを加えたのが赤の線である。この不透明度は、

$$base\_Od \cdot (1.0 + c \cdot random) \cdot \left( 1.0 - \frac{x}{\frac{brush\_width}{2}} \right) \quad (6)$$

で定めている。ここで、 $base\_Od$  は基本的な不透明度の大きさであり(図では青の線)、 $c \cdot random$  は加えたノイズ成分(加えた結果は赤の線)、 $(1.0 - x / (brush\_width / 2))$  は、筆跡の幅(大きさは  $brush\_width$ ) 方向に、ストロークからの距離  $x$  に応じて線形に減衰させるための係数であり、結果の不透明度は黒の線で表される。なお、パース図では筆跡を目立たせないように彩色することが多いので、ここでは筆の両端での筆跡を目立たなくするために減衰させている。どのような減衰が良好な効果を生むかに関する詳細な検討は今後の課題である。また、 $random$  には、値の範囲が(0.0 ~ 1.0)であるような一様乱数列を移動平均法で平滑化したものを用いた。なお、今回は  $c=0.25$ 、 $base\_Od=0.08$  とした。

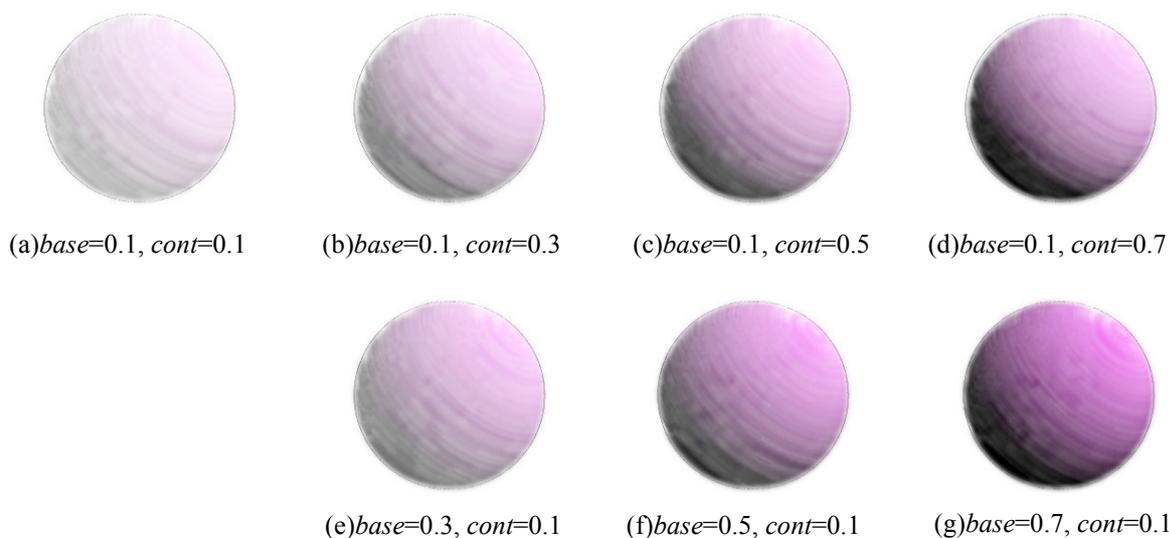


(a)シェーディング画像

(b)筆ストローク=60本

(c)筆ストローク=766本

図8 彩色の例



(a)base=0.1, cont=0.1

(b)base=0.1, cont=0.3

(c)base=0.1, cont=0.5

(d)base=0.1, cont=0.7

(e)base=0.3, cont=0.1

(f)base=0.5, cont=0.1

(g)base=0.7, cont=0.1

図9 不透明度の閾値のパラメタ変更による効果

本手法では、画像にZ値に応じて平滑化処理を施す。平滑化は、ピクセル(x,y)を中心とした、一辺の長さが  $reach(x,y)$ の正方形内部のピクセル値の平均値を、ピクセル(x,y)の新しいピクセル値とすることにより行う。

$$reach(x, y) = zweight \cdot range \quad (7)$$

ここで、 $range$ は平滑化の範囲を調整するためのパラメタである。 $zweight$ には(2)式を用いた。

図8に実際のモデルへの適用した例を示す。画像解像度は全て  $320 \times 240$ pixels である。(a)は通常のレンダリング画像。(b)は彩色処理中の画像で、筆ストロークが 60 本のものである。等濃度の方向に筆が走っているのがわかる。(c)は処理後の画像で、筆ストロークは 766 本である。図9に(5)式中の  $base$  と  $cont$  の値を変えた例を示す。入力モデルには図6(a)を用いた。画像解像度は全て  $320 \times 240$ pixels である。 $cont$  の値を上げる事で、

レンダリング画像上の輝度値の低い(暗い)ところをよりこすり出す効果が得られている。

水彩パース図では遠くの物体ほど形も色も簡略化されて描かれるが、本手法ではZ値に応じた平滑化処理で色の簡略化を簡便に表現している。なお、これによって境界でわずかに色が混じりあい、色のはみ出ているように見える場合もあるが、輪郭線が描かれることで目立たなくなる。また、視点に近い物体ほど色の混じり方は少なく、輪郭線も描かれ易いため、実用上は問題ないと思われる。

### 3. 樹木のレンダリング法

一般的にパース図では対象物が樹木の場合、葉などには輪郭線を加えずに簡略化して描かれる。このため、本手法では、葉のみをレンダリングした画像(Leaf レイヤ)を生成し、葉以外のオブジェクトを2節による手法で描画した画像に $\alpha$ ブレンディングすることで、これを実現する。

樹木の描画では、近くの葉を鮮明に描き、遠くの葉をぼかす、という描き方をすることが多い。そこで、まず、Leaf レイヤに対して以下の手順で視点からの距離に応じた平滑化を行ない、その後、αブレンディングを行う。

(1) Leaf レイヤの全ピクセルをスキャンし、葉ではないピクセルの不透明度(O)と RGB 値を0にし、葉を構成するピクセル(x<sub>i</sub>,y<sub>i</sub>)の不透明度 O(x<sub>i</sub>,y<sub>i</sub>)を 1.0-frange にする。なお、frange は (11)式で決定する。

(2) Leaf レイヤの全ピクセルを再びスキャンし、葉を構成する全ピクセルを探索する。葉を構成するピクセル(x<sub>i</sub>,y<sub>i</sub>)が見つかった場合、そのピクセルの周囲のピクセルに対して、葉の不透明度および RGB 値をピクセル間の距離と Z 値に応じて拡散させるため、Leaf レイヤ上の全ピクセル(x<sub>j</sub>,y<sub>j</sub>)について以下の処理を行う。

(2-1) 式(8)により、不透明度Odの値を計算する。この値がピクセル(x<sub>j</sub>,y<sub>j</sub>)の不透明度より大きければ、この値をピクセル(x<sub>j</sub>,y<sub>j</sub>)の不透明度とする。

$$Od = \text{Density}(distance) \cdot O(x_i, y_i) \quad (8)$$

ここで、

$$distance = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (9),$$

$$\text{Density}(d) = \begin{cases} e^{-(d-flat)^2 / frange} & (d \geq flat) \\ 1 & (d < flat) \end{cases} \quad (10),$$

$$frange = \frac{zbuffer(x_i, y_i) - ozmi}{ozma - ozmi} \quad (11)$$

ozmi,ozma はZ値に応じた拡散を調整するパラメタである。これらのデフォルト値は葉オブジェクトのZ値の最小値と最大値であるが、ozmi<ozmaを条件にユーザが任意に設定することもできる。式(10)中の flat(≧0)は拡散のしかたを調整するパラメタである。これにより、関数 Density は d<flat の範囲でフラットな部分を持つ。その結果、葉を構成するピクセル(x<sub>i</sub>,y<sub>i</sub>)の近傍のピクセル(x<sub>j</sub>,y<sub>j</sub>)では、不透明度が急に減少することなく、ピクセル(x<sub>i</sub>,y<sub>i</sub>)の不透明度 O(x<sub>i</sub>,y<sub>i</sub>)が与えられる。これ

は、flat の値を調整することで、異なる筆の太さによるぼかしの効果を生み出す。

(2-2) Leaf レイヤのピクセル(x<sub>j</sub>,y<sub>j</sub>)の不透明度

O(x<sub>j</sub>,y<sub>j</sub>)が Density (distance) より小さければ

ピクセル(x<sub>j</sub>,y<sub>j</sub>) の RGB 値を葉を構成するピクセル(x<sub>i</sub>,y<sub>i</sub>)の RGB 値で置き換える。

本手法では、関数 Density と frange によって、遠くの葉ほど大きく薄く描かれるようにしている。また、flat を大きくすると葉は大きく描かれるようになる。

一般に、水彩パース図では樹木の葉などは淡い色で描かれる事が多い。本手法では Leaf レイヤに淡い色を持つレイヤ(RGBO(1.0,1.0,0.4,0.3))をブレンドしてから処理を施した。図10に生成画像を示す。画像解像度は全て 480×320pixels である。

#### 4. 画像生成結果

本手法による画像生成例を図11に示す。画像解像度は全て 640×480pixels である。(a)は通常のレンダリングを行った画像であり、(b)は本手法により輪郭線と彩色のシミュレーションを行った画像である。(c)は紙の画像で、(d)は(c)を下層レイヤに設定して本アルゴリズムを適用した例である。

次章で述べるように今後の課題は多いものの、生成例により、本手法は水彩パース図の特徴である、輪郭が描画される、建物については彩色のみ出しが無い、樹木などの建物以外の対象は簡略化されて描かれる、淡い色彩が用いられることが多い、紙の色彩の影響を受ける、などが反映された手法であることが確認できる。

計算には DOS/V Celeron-666MHz を使用した。図11(d)の生成に要した時間をステップ毎に表1に示す。

表1 図11(d)の生成時間

各処理	生成時間(s)
前処理	13
レンダリング	1502
彩色処理	4548
葉のレンダリング	719
葉レイヤの平滑化	757
総時間	7539



(a)シェーディング画像

(b)生成画像

図 10 樹木の生成画像例

## 5. おわりに

本論文では3次元幾何モデルから水彩パース図風画像を生成するための基本的な手法と、水彩画調パース図の特徴を踏まえた、樹木などの添景や遠方の対象を簡略化して描く手法について提案し、その効果を画像生成例により示した。

今後の課題としては、以下のようなことがあげられる。

- (a) 特に彩色アルゴリズムの効率化を図る。
- (b) 等輝度方向に求めているストロークの生成法について、物体の輪郭やテクスチャの方向性を考慮できるようにする。
- (c) 筆の不透明度増加量の望ましいグラフに関する詳細検討を行う。
- (d) ペンや筆の質感を向上させるための紙質等を考慮したモデルへ拡張する。
- (e) 実際の水彩パース図に見られるような、より淡い色で彩色するための、既定値の物体色から水彩調の色彩へ自動変換する手法を開発する。
- (f) 絵の具の光学特性を考慮した発色について検討する。

なお、課題(b)については、開発済の線描表現法[1]におけるストロークの生成法を拡張して適用することを検討している。

## 参考文献

[1]中川大介, 山口恵介, 藤本忠博, 村岡一信, 千葉則茂, 濃淡画像からペン画調画像への変換法 —線描, 点描—, 画像電子学会誌, 2001,

Vol30, No4, pp.350-361.

- [2]G.Winkenbach and D.H.Salesin, Computer-Generated Pen-and-Ink Illustration, SIGGRAPH'94, pp.91-100, 1994.
- [3] L.Markosian, M.Kowalski, S.Trychin and J. Hughes, Real Time Non-Photorealistic Rendering, SIGGRAPH'97, pp.415-420, 1997.
- [4]M.P.Salisbury, M.T.Wong, J.F.Hughes, and D.H.Salesin, Orientable Textures for Image-Based Pen-and-Ink Illustration, SIGGRAPH'97, pp.401-406, 1997.
- [5]G.Winkenbach and D.H.Salesin, Rendering Parametric Surfaces in Pen and Ink, SIGGRAPH'96, pp.469-476, 1996.
- [6]J.Hamel, S.Schlechtweg and T.Strothotte. An Approach to Visualizing Transparency in Computer-Generated Line Drawings. Proceedings of Information Visualisation'98, pp. 151-156, 1998.
- [7]J.Hamel and T.Strothotte, Capturing and Re-Using Rendition Styles for Non-Photorealistic Rendering, EUROGRAPHICS '99, pp.C-173-C-182, 1999.
- [8]G.Elber. Interactive Line Art Rendering of Freeform Surfaces, EUROGRAPHICS '99, pp. C-1-pp.C-12, 1999.
- [9]B.J.Meier, Painterly rendering for animation. Holly Rushmeier, editor, SIGGRAPH'96, pp. 477-484, 1996.
- [10]A.Hertzmann, Painterly rendering with curved brush strokes of multiple sizes, SIGGRAPH'98, pp.453-460, 1998.



(a)シェーディング画像



(b)生成画像



(c)紙画像



(d)生成画像

図 11 水彩画風パース図の生成例

- [11]M.hiraishi and Y.Yamaguchi. Image Moment-Based Stroke Placement, SIGGRAPH'99 Conference Abstracts and Applications, pp.247-248, 1999.
- [12]村上 恭子, 鶴野 玲治, 顔料及び支持体の特性を考慮したパステル画風レンダリング, 芸術科学会論文誌, vol.1, pp.89-96, 2002.
- [13] C.J.Curtis, S.E.Anderson, K.W.Fleischer, and D.H.Salesin. Computer-Generated Watercolor, SIGGRAPH'97, pp.421 - 430, 1997.
- [14]O.Verevka and J.W. Buchanan, Halftoning with Image-Based Dither Screens, GI'99, pp. 167-174, 1999.
- [15]L.Streit and J.Buchanan, Importance Driven Halftoning, EUROGRAPHICS'98, pp.C-207-C-217, 1998.
- [16]J. D. Northrup and L.Markosian, Artistic silhouettes: A hybrid approach. Non-photorealistic Animation and Rendering. SIGGRAPH'00, pp.31-38, 2000.
- [17]R.C.Zeleznik, K.P.Herndon and J.F.Hughes SKETCH: an interface for sketching 3D scenes, SIGGRAPH'96, pp. 163-70, 1996.
- [18]L.Streit, O.Veryovka and J.Buchanan, Non-photorealistic Rendering Using an Adaptive Halftoning Technique, SKIGRAPH'99.[<http://www.cs.ubc.ca/spider/streit/papers/Skigraph99.pdf>]
- [19]O.Veryovka and J.Buchanan, Comprehensive Halftoning of 3D Scenes, EUROGRAPHICS'99, pp.C-14-C-22, 1999.
- [20]O.Deussen, J.Hamel, A.Raab, S. Schlechtweg and T.Strothotte, An illustration technique using hardware-based intersections and skeletons, GI'99, pp.175-182, 1999.
- [21]T.Saito and T.Takahashi, Comprehensive

Rendering of 3-D Shapes, SIGGRAPH'90,  
pp.197-206, 1998.

[22]宮後浩, “緑のプレゼンテーション-GREEN PRESENTATION-”, 学芸出版社, 2000.

[23]Priranesi [<http://www.informatix.co.uk/piranesi/index.shtml>]