

ボロノイ図を用いたステンドグラス風画像の生成とその応用*

1 F-02

芳賀 俊之[†]ヘンリー ジョハン[†]土橋 宜典[‡]西田 友是[†][†] 東京大学大学院[‡] 北海道大学大学院

1 はじめに

コンピュータの普及に伴い、専門でない人でも CG に触れ、作成する機会が増しており、フォトタッチソフト等も頻繁に利用されている。市販のソフトウェアには、入力画像に対して各種の効果を付加する機能が備わっており、すばやく、簡単に、様々な画像を作ることができる。しかし、それらの機能は、画像生成を低コストで行うことを前提としており、必ずしも満足 of いく結果を得ることはできない。

画像に対する効果の一つに、画像のステンドグラス化というものがある。その例を図 1 に示すが、図 1(b) の画像と比べて、市販のソフトウェア [1] による結果画像 (図 1(a)) では、元画像の特徴を読み取ることは難しい。また、市販のソフトウェア [2] では、マニュアルでガラス領域を指定するため、元画像の特徴を表現し易いが、結果画像は、ユーザーの技量に依存し、また、ユーザーの負担も大きくなる。そこで、ユーザーの労力が増加せず、短時間で画像生成が可能でありながら、より品質の高い結果画像が得られる手法が望まれる。

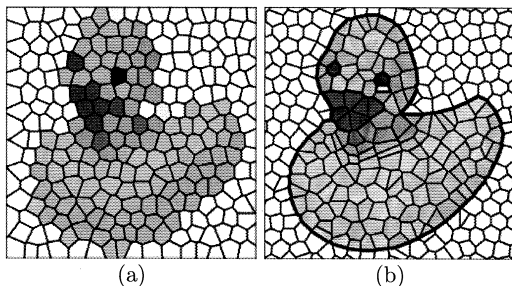


図 1: 市販のソフトウェア [1] での結果画像 (a) と提案法での結果画像 (b) との比較

ここで、図 2 に示す実際のステンドグラス¹を見てみると、ガラスとガラスの境界により、輪郭線等の特徴を表し、各ガラス領域は概ね凸多角形で構成されている。そこで、各ガラス領域をボロノイ図により近似し、画像を生成することを考える。

また、近年、コンピュータ関連機器は低価格化が進んでおり、高性能なグラフィックスハードウェアも広く一般に普及している。各ガラス領域をボロノイ図を用いて表現することで、グラフィックスハードウェアが持つ機能を生かして、画像生成の高速化が望める。

以上の点を踏まえ、アルゴリズムが簡潔であり、かつ、より高品質なステンドグラス風画像を自動的に生成可能である、グラフィックスハードウェアを利用した手法を提案する。また、その応用 (アニメーション化等) についての実験、考察も行った。

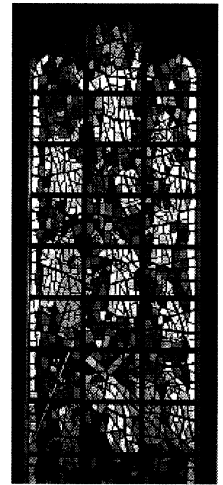


図 2: 実際のステンドグラスの例

2 関連研究

近年、ノンフォトリアリスティックレンダリングの研究は盛んに行われており、非常に様々な生成手法、描画スタイルが提案されている。本稿では、それらの一つとして、ボロノイ図を用いてステンドグラス風画像の生成を行う手法を提案する。

提案法と同じく、画像生成にボロノイ図を応用した手法としては、以下が挙げられる。Haeberli[3] は、ボロノイ領域を決めるための母点を単にランダムに配置し、画像のモザイク化を行い、Hoff III ら [4] はその手法の

*A Method for Creating Stained-Glass-Style Images Using Voronoi Diagrams and Its Applications.

Toshiyuki Haga, Henry Johan, Yoshinori Dobashi and Tomoyuki Nishita.

[†]The Graduate School of The University of Tokyo.

[‡]The Graduate School of Hokkaido University.

¹東北学院大学ホームページ (下記 URL) より転載。

<http://www.tohoku-gakuin.ac.jp/gakuin/outline/reihai.html>

応用例として、母点の位置に揺らぎを与えることで、リアルタイムのモザイク効果を示した。しかし、これらの手法 [3, 4] では、元画像の特徴があまり表現できない。Hausner[5] は、ユーザーの指定したエッジに沿うように、小さな正方形タイルを敷き詰めたモザイクの生成を行った。一方、提案法では、元画像と結果画像との色の誤差が減少するように、各母点を自動的に移動させ、結果として、元画像の特徴を表現している。また、領域形状を正方形に限定していないため、より少ない領域数で、元画像の特徴を表現可能である。宮田ら [6] は、有機的な表面形状 (例: 爬虫類のうろこ) の詳細をボロノイ分割を用いて自動的に生成する手法を提案した。しかし、提案法とは目的が異なっており、スタンドグラス風画像の生成に適用することはできない。

3 画像のスタンドグラス化

3.1 ハードウェアを用いたボロノイ図生成

提案法で採用している、離散ボロノイ図²をグラフィックスハードウェアを用いて高速に生成する方法 [7] について、ここで簡単に説明する。また、ボロノイ図の各部の名称については、図 3(a) を参照のこと。

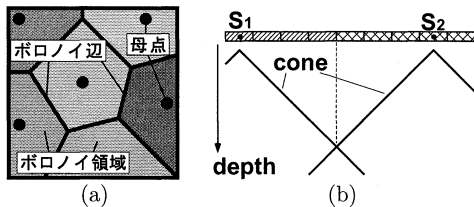


図 3: ボロノイ図 (a) と、ハードウェアによる離散ボロノイ図生成の概念図 (断面図)(b)

離散ボロノイ図を作成するには、まず、母点はスクリーン平面上にあるとし、母点の位置を頂点として、底面がスクリーンと平行であるような円錐を考える。こうすることで、円錐の表面の奥行きは母点からの距離と見なすことができる。そこで、各母点について、隠面消去を施しつつ、その領域の色で円錐を描画する。図 3(b) は、隣接した二つの母点 S_1, S_2 を通り、スクリーンに垂直な方向の断面図を示しており、cone は円錐の表面である。depth の判定により、各画素は塗り分けられる。こうして、生成されたボロノイ図では、解析的なボロノイ辺の情報は得られないが、その代わりに、生成を高速に行うことができる。

²ボロノイ図の境界線を厳密に計算するのではなく、各画素がどの領域に含まれるか判定して表示したもの。

3.2 提案法の概要

まず、以降の説明で使用する用語を幾つか説明する。

ボロノイ画像: 生成中のボロノイ図で表現された画像

参照画像: 画像生成において目標とする、入力画像の色数を減らした画像 (4.1 節参照)

E_{color} : ボロノイ画像と参照画像との対応する画素毎の色を比較した場合の二乗誤差の合計 (式 1)

$$E_{color} = \sum_x \sum_y \sum_c \left(P_{(x,y,c)}^V - P_{(x,y,c)}^R \right)^2 \quad (1)$$

但し、 P^V, P^R は各々、ボロノイ画像と参照画像中の画素を表し、添え字 (x, y, c) は、位置が (x, y) 、色成分 (RGB) が c であることを示す。 (x, y) は画像全体で定義される。

計算途中のボロノイ図においては、各母点は画素の中心に位置しており、また、その母点を含むボロノイ領域は、その母点と同じ位置の参照画像の画素の色で塗りつぶされるものとする。

提案法は大きく二つのプロセス (パターン生成プロセスと効果付加プロセス) から成る (生成画像の変化の過程は図 4 参照)。前者は自動で行われ、後者はインタラクティブな操作によって行われる。

パターン生成プロセスでは、母点位置を離散的な画素位置に限定し、参照画像 (4.1 節) にできるだけ類似したボロノイ画像を生成する。すなわち、母点の初期位置を格子状に配置してボロノイ画像を作成 (4.2 節) し、参照画像との誤差 E_{color} を計算する。そして、 E_{color} が減少するように母点を移動する。この処理を繰り返し、参照画像の特徴を反映したボロノイ画像を生成する。一回の反復での母点の移動は、隣接する 8 画素のいずれかへの移動 (8 方向の移動) を考える。また、 E_{color} が減少しない場合には、その母点は移動させない。この際、処理の高速化のため、母点移動は以下のように二段階で行う。(i) すべての母点に関して、母点が移動した際のその母点に対応した領域の変化を近似的に計算する。そして、それに基づき E_{color} の変化を計算し、すべての母点を一斉に移動させる (4.3 節)。(ii) 個々の母点に関して、母点の移動に伴う領域の変化を厳密に計算する。そして、それに基づき E_{color} の変化を計算し、その母点を移動させる (4.4 節)。自動的な母点移動後に、インタラクティブに母点位置を調節可能である (4.5 節)。

効果付加プロセスでは、パターン生成プロセスで求めた母点配置の位置情報を基に、解析的な (離散的でない) ボロノイ図を計算し、描画に必要なボロノイ辺の情報を得る (5.1 節)。そして、結果画像をよりスタンドグラス風にするための効果を付加する (5.2 節)。

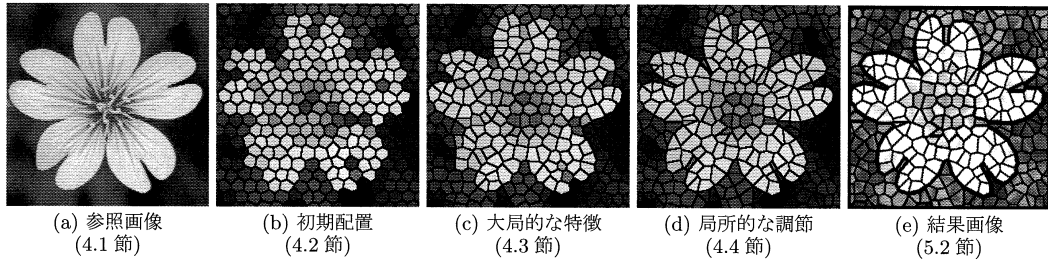


図 4: 各ステップ後のボロノイ画像の変化

4 パターン生成プロセス

4.1 ベクトル量子化

入力画像の各画素の色 (R, G, B) を三次元ベクトルとみなし、ベクトル量子化 [8] を行うことで、参照画像を表現するための色のテーブルを作成する。一般に、ベクトル量子化の目的は、データ圧縮であるが、提案法では以下の二点である³。

- 参照画像の色数を減らすことで、生成されるボロノイ領域に使用される色数を減らす。
- 参照画像とボロノイ画像の各画素毎での色の誤差の計算 (E_{color} の算出に必要) を高速化 (差の二乗の値を格納した参照テーブルを作成し、利用) する。

4.2 母点の初期配置

ボロノイ画像の初期状態として、ハニカム (六角形格子) 状に母点の初期配置を行う。生成される結果画像は概ね、初期配置時の母点間隔に依存したものとなる。ユーザーが指定できるパラメータとしては、母点間の間隔と、「ぶれ」(規則的なハニカムの中心点からのずれ) の大きさがある。また、画像の細かな特徴を表現し易くするために、ユーザーが指定した範囲でアダプティブに、配置する母点の密度を変化させられる。

4.3 大局的な領域計算

ボロノイ画像を参照画像に高速に近づけるために、近似的な領域形状の計算を行い、すべての母点を一斉に動かすということを繰り返す。そして、ボロノイ画像が参照画像の特徴を大まかに表すようにする。

近似的な領域形状の計算方法を以下で説明する。

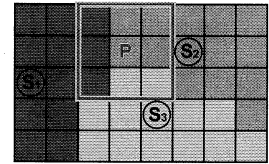
母点を移動させる際には、 E_{color} の変化量が分かれば、各母点を移動させるべき方向が判断できる。そし

³実際のスタンドグラスでも使われるガラスの色数はそれほど多くはない、という観察に基づいている。また、提案法では、経験的に 256 色の色のテーブルを使用した。

て、 E_{color} の変化量の計算に必要である領域形状の変化を知ることは、扱っているのが離散ボロノイ図であるために、ある母点のある方向への移動と、画像中の任意の画素との関係 (詳細は後述) を知ることと等価である。

そこで、「ある母点が移動しても (周りの母点の移動は無視して)、対応するボロノイ領域の形状は変化しないで、母点と同じだけ平行移動する」と仮定する。また、3.2 節で述べたように、一回の母点の移動を隣接した 8 方向の画素のみに限定しているため、 E_{color} の変化の計算が低コストで行えるようになる。

これを図 5 を用いて説明する。図 5 の、 P を注目する画素、 S_1, S_2, S_3 を P の近傍の母点とする。また、母点 S_i に対し、方向 D_j に移動した

図 5: E_{color} の変化量の計算際の E_{color} の変化量 $dE_{(i,j)}$ を準備する。

P は S_2 の領域に含まれており、また、 P の上、右上、右の画素も S_2 の領域に含まれる画素である。よって、 S_2 が下、左下、左に移動する場合には、 P は S_2 の領域に含まれたままの画素であり、 S_2 がそれ以外の方向に移動する場合には、 P は S_2 の領域から外れる画素である。そこで、これらの情報から、 S_2 が保持する各方向に対応した dE を更新する。 dE の更新は、誤差の値を格納した参照テーブル (4.1 節) によって計算できる誤差の変化量を追加することで行われる。 P の下、右下の画素は、 S_3 の領域に含まれる画素である。よって、 S_3 が上、左上に移動する場合には、 P は S_3 に対応した領域に新たに含まれる画素である。ゆえに、 S_3 の上方向と左上方向の移動に対応した dE の値を更新する。

他の場合についても同様であり、全画素について、それと隣接した 8 つの画素に対して、上記の計算を行うと、結果として、すべての (i, j) について、つまり、母点 S_i の方向 D_j への移動について、 $dE_{(i,j)}$ の値が計算

される。よって、各母点について、 dE が最小となる方向、つまり、移動すべき方向が計算できる。

入力画像の大局的な特徴を表現するための、全母点を一斉に移動させるステップは以下の通りである。このステップでの E_{color} の変化量の計算は近似的であり、母点の位置が収束せず、振動してしまうこともある。そこで、 E_{color} の変化率が閾値未満になるまで、このステップを繰り返す。

- i. 各母点に異なる色を割り当てて、円錐を描画することで、フレームバッファにポロノイ図を描画
- ii. フレームバッファからポロノイ画像の読み出し
- iii. すべての母点に関して、各方向へ移動した場合の E_{color} の変化 (つまり、 dE の値) を近似的に計算し、最も E_{color} が減少する方向を決定
- iv. 全母点を移動

4.4 画像の局所的な調節

本節で説明するステップでは、一度に移動させる母点の一つのみとする。そして、各母点について、最適な母点の位置を探索するために、隣接する 8 画素へ移動した際の色誤差 (E_{color}) の変化量を算出し、誤差の最も減少する方向へ母点を移動させる。しかし、移動方向毎にポロノイ領域の計算、および、フレームバッファの読み出しを行うと時間を要する。一般的に、グラフィックスハードウェアを用いた手法では、バッファの読み出しが計算時間の点でボトルネックとなり易い。

そこで、フレームバッファの主メモリへの読み出し回数を減らし、かつ、母点に対応した円錐の描画回数を減らせる方法を提案する。ポロノイ画像は、各母点を頂点とする円錐を描画することで得られる (3.1 節参照) が、移動方向毎にすべての母点に対する円錐を描画すると時間を要する。そこで、あらかじめ、移動させる母点を除くすべての母点に対する奥行きを記憶しておき、次に、移動させる母点の円錐を描画すると、移動後のポロノイ領域が抽出される。この際、フレームバッファおよびデプスバッファを書き換える必要はなく、高速に領域が抽出できる。また、そのとき、ポロノイ領域の記憶には、通常、マスクとして使用されるステンシルバッファを利用する。ステンシルバッファの各ビットを母点の各移動方向に対応させて記憶させることで、複数の移動後の領域形状を一回のバッファの読み出しで得ることができるため、高速化を図ることができる。

ある母点 S を動かす場合のアルゴリズムは、以下の通りである。これを、移動により E_{color} が減少する可

能性のあるすべての母点について行い、そして、 E_{color} の変化が閾値未満になるまで繰り返す。

- i. S 以外の母点で構成されるポロノイ図を生成
- ii. 各方向へ S を移動した後での S のポロノイ領域を、ステンシルバッファに記録 (ステンシルバッファの値の各ビットを各移動方向に対応させて)
- iii. フレームバッファ (S 以外で描画されたポロノイ画像) と、ステンシルバッファ (各方向へ移動後の S に対応した領域形状) の読み出し
- iv. 領域形状の変化の厳密な計算に基づき、 S の各移動方向毎の E_{color} の変化を計算
- v. 最も E_{color} が減少する方向へと S を移動

4.5 細かな特徴の表現

前ステップまでの自動的な母点の移動だけでは、初期配置 (4.2 節) の際に母点が配置されなかった元画像の特徴を表現することは難しい。そこで、インタラクティブな母点の移動、追加、削除を可能にしている。具体的には、ユーザーはマウス操作を通じて、前述の 3 つの母点操作を行い、それを受けて、システムは再び、自動的に、母点配置の最適化 (4.4 節) を行う。

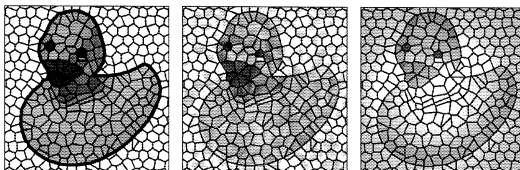
5 効果付加プロセス

5.1 ポロノイ図の計算

ポロノイ画像は離散的なので、ポロノイ辺やポロノイ領域同士の隣接関係を正確に計算できない。そこで、4 節で求めた母点の位置情報から、通常の (離散的でない) ポロノイ図生成アルゴリズムにより、それらを求める。ポロノイ図の生成手法は、提案法で採用した逐次添加法も含め、Aurenhammer[9] が詳しく論じている。

5.2 ステンドグラス風画像の生成

よりステンドグラス風な画像を生成するために、5.1 節で得られた情報を用いて、以下の操作を行う。各々の操作はユーザーがインタラクティブに行える (図 6)。



(a) 輪郭の強調 (b) ガラスの色むら (c) 透過光の表現

図 6: ステンドグラス風の表現のための効果

- ボロノイ辺 (境界線) の太さを、それに隣接した二つのボロノイ領域の色の誤差が大きいほど、太くし、参照画像中の輪郭や特徴をより強調する (図 6(a))。
- 実際のガラスでは歪み等の原因で、部分毎に明るさや透過率の変化がある。そのような材質感を擬似的に表現するため、ボロノイ領域の各頂点で乱数を加味して輝度を計算して塗りつぶす (図 6(b), (c))。

6 アニメーション

本節では、画像シーケンスを入力とした際の、ステンドグラス風アニメーションの作成について説明する。

ノンフォトリアスティックレンダリングにおいてアニメーションを作成する場合には、一般的に、フレーム間の相関関係が最も問題となる。そこで、我々は、計算コストが低く、かつ、コヒーレンスを保つことができる方法として、キーフレーム法を採用した。

まず、入力シーケンスから、数枚おきに、キーフレームとなる画像を選び出し、前節までで説明した手法により、ステンドグラス風画像を生成する。その際、全キーフレームに対して、初期状態では同じ母点配置を用いる。また、各母点には、初期位置に対応して、全キーフレームで共通の ID を割り当てておく。そして、同じ ID を持つ母点について、その位置と色を線形補間することによって、キーでないフレームの生成を行う。

結果のアニメーション中では、各母点の状態は、(i) 色の変化は少なく、移動することで、ほぼ同じ部分を表現する場合、(ii) 位置、色ともに変化し、別の部分への表現に切り替わる場合、の二つがある。フレームが進むにつれて、これらの状態を入れ替わることで、アニメーション中の各部分の様々な変化が表現される。すなわち、(i) に属す母点で、平行移動や回転が表現され、(ii) に属す母点で、出現や消滅する部分が表現される。

7 結果

(1) ステンドグラス風画像

提案法による結果画像を、図 1、4、6、7、10 に示す。図 1(b) の例では、市販のソフトウェアの結果画像 (図 1(a)) では表せていない、輪郭等の元画像の特徴がよく表現されている。なお、図 4 に示した生成画像の変化の様子は、実際の画像生成時にもユーザーは見ることが出来る。提案法では、図 7 にも示すように、様々な画像を入力とすることができる。また、図 6 のような各種の効果を結果に付加することも可能である。

画像名 (図番号)	画像サイズ	母点数	計算時間
ダック (図 1(b), 6)	256 × 256	341	2.0 秒
花 (図 4(e))	256 × 256	347	3.4 秒
アゲハ蝶 (図 7(b))	512 × 512	830	8.6 秒
ひまわり (図 7(d))	512 × 512	1271	12.7 秒
競走馬 (図 7(f))	512 × 512	1278	14.0 秒
聖人 (図 10(b))	360 × 640	1594	20.8 秒

表 1: 計算結果

それぞれの画像の生成に要した計算時間等を表 1 に示す。計算時間の測定は、Pentium III 1GHz、GeForce3 搭載 PC で行い、グラフィックスライブラリとしては OpenGL を用いた。表 1 にあるように、リアルタイムではないが、どの画像も秒単位での生成が可能であった。

(2) アニメーション

6 節で述べた、ステンドグラス風アニメーションの結果のキーフレームの幾つかを図 8 に示す。

(3) 三次元シーンへの適用

ステンドグラス風画像をテクスチャとして利用した応用例として、スタンドの置かれた部屋 (図 9) と教会 (図 10(c)) のシーンを示す (それぞれ、図 4(e)、10(b) をテクスチャとして使用)。レンダリングには、土橋らの方法 [10] を用いており、図 9、10(c) 共に、ほぼリアルタイムでの描画が可能であった。

8 終わりに

ノンフォトリアスティックレンダリングの技法の一つとして、ボロノイ図を利用することで入力画像の特徴をよく表した、ステンドグラス風画像の生成を行う手法を提案した。提案法では、ユーザーの介入が必要なく自動的に、また、グラフィックスハードウェアを効率的に用いて高速に、画像生成を行うことができる。

参考文献

- [1] Adobe Photoshop[®].
<http://www.adobe.com/products/photoshop/>
- [2] Dragonfly Software Glass Eye[®].
<http://www.dfly.com/>
- [3] P. Haeberli. "Paint by Numbers." Proc. SIGGRAPH 90, pp.207-214.
- [4] K. E. Hoff III, T. Culver, J. Keyser, M. Lin and D. Manocha. "Fast Computation of Generalized Voronoi Diagrams Using Graphics Hardware." Proc. SIGGRAPH 99, pp.277-286, 1999.
- [5] A. Hausner. "Simulating Decorative Mosaics." Proc. SIGGRAPH 2001, pp.573-580, 2001.
- [6] K. Miyata, T. Itoh and K. Shimada. "Organic Texture with Controlled Anisotropy and Directionality." SIGGRAPH 2001, Sketch, Technical, pp.240, 2001.

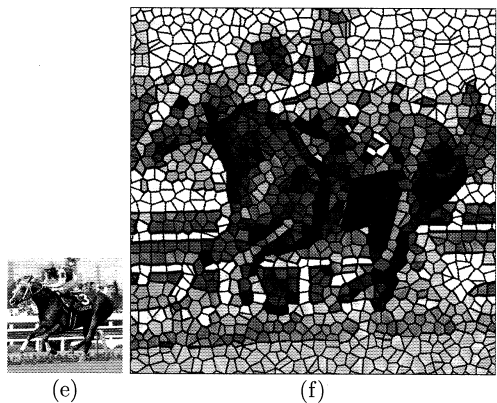
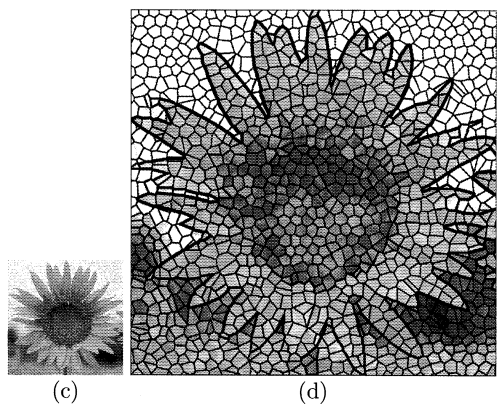
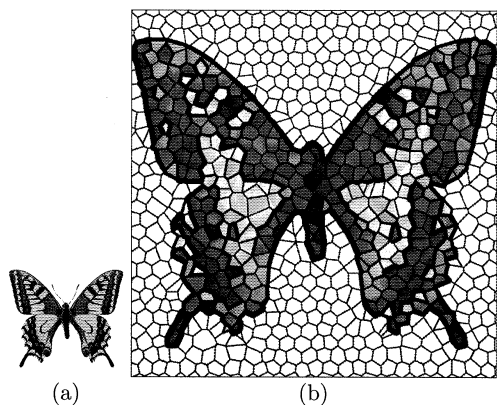


図 7: 入力画像 (a,c,e) と結果画像 (b,d,f)

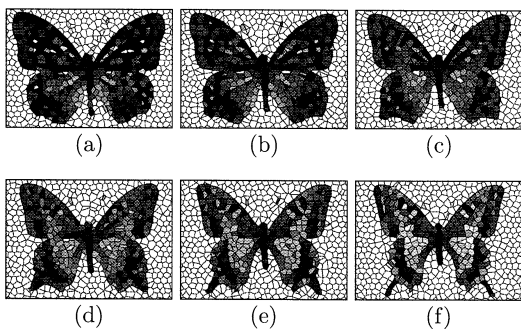


図 8: 結果のアニメーションのキーフレーム

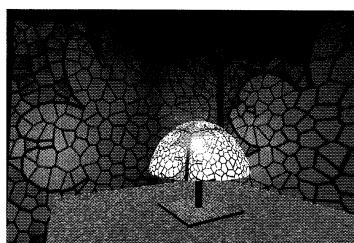


図 9: シーンへの応用 (1): スタンドランプのある部屋

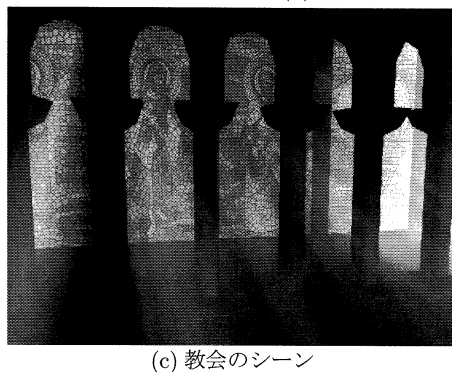
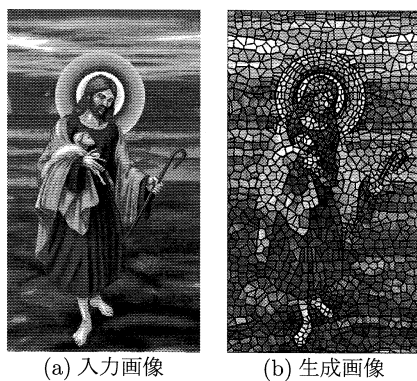


図 10: シーンへの応用 (2): 教会のステンドグラス

- [7] M. Woo, J. Neider and T. Davis. *OpenGL Programming Guide, Release 1*. Addison Wesley, 1993
- [8] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.
- [9] F. Aurenhammer. "Voronoi Diagrams: A Survey of a Fundamental Geometric Data Structure." *ACM Computing Surveys*, Vol 23, pp.345-405, 1991.
- [10] Y. Dobashi, T. Yamamoto and T. Nishita. "Interactive Rendering Method for Displaying Shafts of Light." *Proc. the 8th Pacific Conference*, 2000-10, pp.31-37.