

スケッチに基づくメッシュモデルの類似検索システム

-概略構造スケッチに基づくモデルの類似検索手法-

Content-based Mesh-Model Query based on sketch image

鎌田 雄喜[†] Yuki Kamata 金井 理[†] Satoshi Kanai 岸浪 建史[†] Takeshi Kishinami

1. はじめに

1.1 背景・目的

意匠デザインにおける工業デザイナーの形状発想プロセスは、主に2段階に分けられる。第一段階は、製品内部の機能部品レイアウトの違いに対応した、様々な概略構造（プロポジション）の案を発想する段階であり、第二段階は、特定のプロポジションに属する詳細形状（バリエーション）の案を発想する段階である。これら概略構造と詳細形状の案を、短時間で出来るだけ多く提案する事が、工業デザイナーに要求される。

現在、3次元形状モデル(CADモデル、メッシュモデル等)の蓄積が企業内およびWeb上で進められている。このような多数の3次元形状モデルを登録したデータベース(以下DB)内から類似形状のモデルを検索する類似検索機能は、工業デザイナーが上述したバリエーションの案を発想する際の作業支援に有効な機能となると思われる。

そこで本研究では、工業デザイナーが発想段階で描く手書きラフスケッチ画像を入力とし、3次元メッシュモデルが登録されたデータベース内から、このスケッチに類似した投影画像を有するモデルを効率的に検索する手法の提案を目的とする。

1.2 関連研究

メッシュモデルの類似検索手法には、様々な方法が提案されているが、殆どが入力として3次元モデルが必要な手法である。一方、投影画像を入力とするメッシュモデルの類似検索手法としては、視点毎の距離画像間の類似性を比較する手法[Nakazawa03]や、手書きの輪郭画像とメッシュモデルの視点毎の輪郭画像同士の類似性を比較する手法[Funkhouser03]等が既に提案されている。しかし[Nakazawa03]の手法は、手書きスケッチ画像を検索入力とした類似検索には適用できない。また、[Funkhouser03]では、モデルの輪郭画像のみしか類似性の比較に利用しないため、デザイナーがラフスケッチ上に描く製品表面上の特徴線等の重要な情報を、類似検索に有効活用できない。

2. 類似検索手法の概要

提案する3次元モデルの類似検索処理の手順概要を図1に示す。まず前処理として、メッシュモデルDB内のn個の全モデル(M₁, ..., M_n)に対し、それらの輪郭画像と Non-Photorealistic Rendering (以下 NPR)により生成された画像とを、予め規定された視点毎に作成し、画像DBに格納しておく。さらにこれら画像毎の特徴量を計算し、特徴量DBに格納しておく。一方、類似検索処理では、デザイナーがペンタブレット等を用いてスケッチ画像を入力すると、検索システムはその画像の特徴量を計算し、特徴量DB内の全ての特徴量との類似度を総当り的に算出する。この結果に基づき、類似度の高いモデルを順に表示する。検索結果は画像毎に元のメッシュモデルへのリンクを持っており、最終的に入力スケッチ画像に類似する投影画像を持つメッシュモデル群を得ることができる。

3. Non-Photorealistic Rendering (NPR)を用いたメッシュモデルからの投影画像生成手法

3.1 NPRに基づく投影画像の描画方法

予め与えられたメッシュモデルからの NPR 画像生成には[Rasker01]の手法を用いた。この手法は(1)輪郭線・凸凹稜線・面交線を単純かつ独立な手続きでレンダリング可能、(2) Polygon

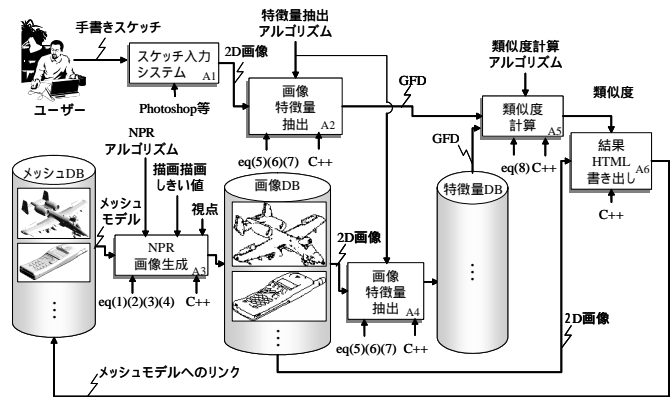


図1: 本研究の手順概要

Soup形式のメッシュモデルでもレンダリング可能、(3)視点位置に依存した輪郭線を実時間レンダリング可能、といった特徴を持つ。

3.1.1 輪郭線の描画

輪郭線は、メッシュモデルの全面分に対して、前方面の面分を白く描画し、後方面の場合は視点位置に応じた大きさの拡張面分を、元の面分の外周に追加し、追加面分を黒で描画することによって生成できる。拡張面分幅 w_{of} は式(1)で計算できる。

$$w_{of} = w_{sil} \{z \sin \alpha (\mathbf{V} \cdot \mathbf{C}) / (\mathbf{V} \cdot \mathbf{N})\} (\mathbf{E} \times \mathbf{N}) \quad (1)$$

ここで、 w_{sil} : 輪郭線幅制御パラメータ、 z : 面分の頂点のカメラに対する深さ、 \mathbf{V} : 視線方向単位ベクトル、 \mathbf{C} : 視軸方向単位ベクトル、 \mathbf{N} : 描画面分単位法線ベクトル、 \mathbf{E} : 面分稜線方向単位法線ベクトル、 $\cos \alpha = \mathbf{V} \cdot \mathbf{E}$ 、である。

3.1.2 凸稜線の描画

凸稜線は、メッシュモデルの全面分に対し、前方面なら面分を白く描画し、面分の稜線周りに角度 θ 回転した拡張面分を黒で描画することで生成できる。拡張面分幅 w_{of} は式(2)(3)(4)で計算できる。

$$w_{of} = w_{rid} \{z \sin \alpha (\mathbf{V} \cdot \mathbf{C}) / (\mathbf{V} \cdot \mathbf{R})\} \mathbf{Q} \quad (2)$$

$$\mathbf{R} = -\mathbf{N} \cos \theta + (\mathbf{E} \times \mathbf{N}) \sin \theta \quad (3)$$

$$\mathbf{Q} = -\mathbf{N} \sin \theta - (\mathbf{E} \times \mathbf{N}) \cos \theta \quad (4)$$

ここで、 w_{rid} : 凸稜線幅制御パラメータ、 θ : 凸稜線表示制御パラメータ(面分-拡張面分間の二面角)、 \mathbf{R} : 拡張面分単位法線ベクトル、 \mathbf{Q} : 面分稜線に垂直な拡張面分上の単位ベクトル、である。

3.1.3 凹稜線の描画

凹稜線は、メッシュモデルの全面分に対して、前方面分の稜線周りに角度 ϕ 回転した拡張面分を付け加え、前方面分の深さ値からなるデプスバッファ $Z1$ と、 $Z1$ の深さ値に比べ僅かに大きな深さ値を持つデプスバッファ $Z2$ との間に拡張面分が含まれていた場合に限り、その拡張面分を黒で描画することで表現できる。拡張面分幅 w_{of} は凸稜線の場合と同様に計算する。

3.2 描画時の投影方向

描画時の投影方向は、デザイナーがラフスケッチを描く際に標準的に利用する、(1)3面図・(2)アイソメ図をもとに、(1)正投影となる6個の投影方向、(2)等側投影となる16個の投影

[†] 北海道大学大学院工学研究科

Graduate School of Engineering, Hokkaido University

方向とした。

4. 2次元画像の類似度に基づくメッシュモデルの類似検索手法

画像間の類似検索には[Zhang02]の手法を用いた。この手法は、元画像を極座標変換した後、さらに極座標画像をフーリエ変換し、そのパワースペクトルに基づく画像特徴量間の類似度を算出する手法である。これにより、手書きスケッチを検索入力画像とする際に必須となる、画像の平行移動、拡大・縮小、回転に影響を受けない画像間類似検索が可能となる。

4.1 画像特徴量の抽出

画像特徴量には、(5)(6)(7)式で定義される多次元ベクトルであるGeneric Fourier Descriptor(GFD)を用いた。

$$\mathbf{GFD} = \{E(0,0), E(0,1), \dots, E(0,n), \dots, E(m,0), \dots, E(m,n)\} \quad (5)$$

$$E(0,0) = \frac{|PF_2(0,0)|}{\text{area}}, E(k,l) = \frac{|PF_2(k,l)|}{|PF_2(0,0)|} \quad (6)$$

$$PF_2(\rho, \phi) = \sum_r \sum_i f(r, \theta_i) \times \exp[-j2\pi(\frac{r}{R}\rho + \frac{2\pi i}{T}\phi)] \quad (7)$$

$$\text{area} = R \times T \quad (8)$$

ここで、 m, n : 類似度を算出する半径、角度周波数成分の最大個数、 (ρ, ϕ) : 周波数の半径成分、角度成分 $0 \leq \rho < R, 0 \leq \phi < T$ 、 (r, θ_i) : 画像の半径座標、角度座標 $0 \leq r < R, \theta_i = i(2\pi/T)$ 、 $0 \leq i < T$ 、 R, T : 極座標の半径解像度、角度解像度

4.2 画像間の類似度計算

画像間類似度は、式(8)のGFD間のユークリッド距離Dとし、低い値ほど類似している事を示す。

$$D(\mathbf{GFD}_Q, \mathbf{GFD}_{I_x}) = \sqrt{\sum_{k=0}^m \sum_{l=0}^n \{E_Q(k,l) - E_{I_x}(k,l)\}^2} \quad (9)$$

ここで、 $\mathbf{GFD}_Q, \mathbf{GFD}_{I_x}$: 問い合わせ画像、検索対象画像のGFD、 E_Q, E_{I_x} : 問い合わせ画像、DB内の画像のGFDの各要素。

5. メッシュモデルの類似検索実験及びその結果

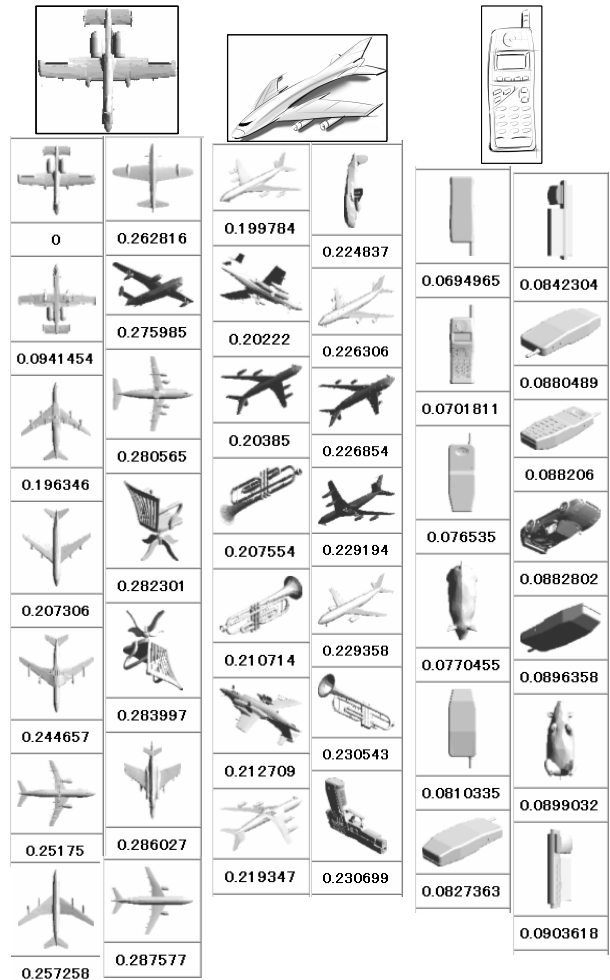
5.1 類似検索アルゴリズム

類似検索アルゴリズムは C++を用いて実装する。検索対象は、3ds, .max, .stl, .ply, .wrl形式のメッシュモデルとした。まず前処理として、メッシュ DB 内の全メッシュモデルに対し、3.2 で設定した各視点に対し、輪郭のみの画像、および NPR 画像の 2 種類を用意し、画像 DB 内に格納する。さらにこれらに対し GFD を算出し、特徴量 DB に格納する。

次にユーザーが入力した.bmp 形式の問合せ画像から、モルフォロジーの closing 処理にて輪郭画像を生成し、元画像と輪郭画像両方に対して GFD を計算する。DB 内の全モデルの輪郭画像の GFD と、入力輪郭画像の GFD 間の類似度を計算することで、物体のある視点から見た輪郭画像が類似したモデルを、複数個検索する。その中でさらに詳細な物体の特徴を見るために、DB 内の NPR 画像の GFD と、入力元画像の GFD 間の類似度を計算し、類似度の高い(D の値が小さい)順に検索された画像およびメッシュモデルを HTML 形式で表示することができる。

5.2 実験結果

前述した形式の 90 個のメッシュモデルを用意し、輪郭画像に基づき検索を行った。図 2、図 3、図 4 にその結果を示す。画像下の数字は問い合わせモデルとの類似度 $D(\mathbf{GFD}_Q, \mathbf{GFD}_{I_x})$ である。図 2 はメッシュモデルから生成した画像を問い合わせとし、図 3 はデザイナーが DB 内のメッシュモデルを参照し、描いたスケッチを問い合わせとし、図 4 はデザイナーが何も参照せずに描いたスケッチを問い合わせとして検索を行った。類似検索は、 $R = 64, T = 256, m = 3, n = 15$ とし、P4, 2.5GHz で、10 秒以内で完了できた。



(左) 図 2 : DB 内のメッシュモデルから生成した画像を問い合わせとした検索結果

(中) 図 3 : デザイナーが DB 内のメッシュモデルを参照し、描いたスケッチを問い合わせとした検索結果

(右) 図 4 : デザイナーが何も参照せずに描いたスケッチを問い合わせとした検索結果

6. まとめと考察

図 2 から、入力である飛行機のモデルが上位の大部分を占めていることや、図 3 で DB 内に格納された携帯電話 2 個のうち 2 個が上位に現れていることや、図 4 でモデルを参照せずに描いた飛行機の画像を入力したにもかかわらず、上位に多数飛行機のモデルが現れていることから、本手法が妥当であるといえる。今後の課題として、NPR 画像の生成時の線幅や、画像特徴量抽出の際の様々なパラメータと検索結果の関係を調査すること等が挙げられる。

参考文献

- ・ [Nakazawa03] : 中澤昌俊 et al.
"見掛けに基づく特徴量を用いた 3 次元形状の形状類似検索"
情報処理学会第 65 回全国大会, 平成 15 年 pp263 - 266
- ・ [Funkhouser03] : Thomas Funkhouser et al.
"A Search Engine for 3D Models"
ACM Transactions on Graphics, Vol. 22, No. 1, 2003, pp 83-105
- ・ [Raskar01] : Ramesh Raskar et al.
"Hardware Support for Non-photorealistic Rendering"
SIGGRAPH/Eurographics Workshop on Graphics Hardware (HWWS), August 2001, pp41-47
- ・ [Zhang02] : Dengsheng Zhang et al.
"Shape-based image retrieval using generic Fourier descriptor"
Signal Processing: Image Communication 17, 2002, pp825-48.