

データベースに基づく車体形状デザイン GUI の構築

Data Driven GUI Development for Car Shape Design

仲田真輝
Masaki Nakada

早川達順
Tatsunori Hayakawa

杉本志織
Shiori Sugimoto

森島繁生
Shigeo Morishima

早稲田大学理工学部
Science and Engineering, Waseda University

1. はじめに

近年、自動車の開発過程や広報活動において CG は不可欠な技術となっている。一般的に、自動車の車体形状デザインは、デザイナーの 2 次元のデザイン画をベースに、3 次元モデリングソフトを用い、手作業で立体的な形状をモデリングする。この作業には長い時間と多大なコストを要し効率化が求められている。このような背景から、本研究では、過去の車種の 3 次元形状データベースに基づいて、その内挿や主成分によって新しい車体形状を直接的かつ効率的にデザイン可能なツールの構築を目的とする。

2. 関連研究

ここで提案するカーデザインツールは、車体の特徴を全体的なシェイプとして捉え、フロントガラスやヘッドライトなどの細かいパーツに関しては、別途後処理でデザインするというアプローチを取る。従来研究でも、車体形状を既存車種の内挿や主成分により表現する方法は既に提案されている[1]。しかし、いずれも全てのパーツ類を考慮しているために、共通のワイヤフレームにより全車種を網羅してデータベース化することは難しく、手動モデリングが避けられないために、手間のかかる作業であった。そこで本手法では、全自動でのデータベース化を実現し、車体形状空間の構築を容易に実現可能とした。

3. 車体形状データベース構築

本研究では、まず既存の車種の CAD データからデータベース構築を行う。個々の CAD データは、異なる形式で記述されており、表現される形状の複雑さによって割り振られるポリゴン数などまちまちである。そこで既存車の車体形状データを、同次元数の統一された形式に変換し、データベースを構築する必要がある。

まず始めに図 1 のように車体に対する平行面上にグリッドを用意し、各グリッド交点から車体に垂直に降ろした直線と車体の交わる点の座標をそのグリッド交点に対応付ける。上記のプロセスを車体の前後、左右、上下の 6 方向から行い、最終的に 1 つのモデルとして統合することで、1 台分の車体データとした。グリッドの分割数はすべての車種で同数とし、グリッド交点に統一の番号を割り振る。この番号を用いて車体間の対応付けを行うことにより、分析・合成可能な形式でデータベース構築が可能である。図 1 に上面からのモデルを示す。また図 2 は、6 方向からのモデルを統合し、1 台の車体としてモデル化した結果である。このプロセスは、6 方向からの頂点のみを混合して、ポリゴンを自動的に生成している[2]。少ない頂点数にも関わらず各車種の特徴が顕著に表現されていることが分かる。



図 1 上面モデル

図 2 車種データベースの例

4. データベースの分析方法

共通次元ベクトルとして得られたデータベースを用いて、6 つの各方向のデータについて、まず別々に PCA を行うことで次元圧縮を行う。次にその各方向の圧縮されたデータを結合して 1 つの車体を表現し、このデータに対し再度 PCA を施すことで、形状の特徴空間を抽出した。

5. 車体形状の合成

PCA 分析の結果としての固有空間上に、既存車種の車体形状を写像し、固有空間上での形状特徴の分布の様子を可視化した。さらに、その固有空間上で主成分値の変化により任意の車体形状の合成が可能になった。

本カーデザインツールでは固有空間でスライダを用いて各主成分値を操作することにより、連続的に車体の変形を施すことが可能である。この手続きをインタラクティブに行うことによって、新たな車種の形状が決定される。

6. まとめ

本研究では、既存の CAD データから、自動的に車種データベースを構築する手法を提案した。この手法は、元のデータ形式に依存せず、たとえばレンジデータ等を入力とすることも可能なため極めて有効である。また必要に応じて、ポリゴンの密度を変化させて最適なデータベース構築が可能である。今後は、新車種の合成に関して、デザイナーの使いやすい GUI の構築を目指す。

7. 参考文献

[1] Randall C. Smith, "Navigating in a Shape Space of Registered Models", IEEE Transactions on Visualization and Computer Graphics, vol.13, No.6, 2007.

[2] C. Bradford Barber and David P. Dobkin and Hannu Huhdanpaa: The Quickhull Algorithm for Convex Hulls, ACM Transactions on Mathematical Software, Vol. 22, No. 4, pp. 469-483 (1996).