

手書きスケッチに基づく 3 次元シーンのデザインシステム

Interactive Design of 3D Scenes from Freehand Sketches

辛 孝宗 岡部 誠 五十嵐 健夫*

概要. 複数のモデルを含んでいる 3 次元シーンのデザインは煩雑な仕事である. 現在使われている通常の 3 次元ツールはユーザが適切なモデルをロードし, 移動, 拡大/縮小, 回転などのオペレーションを反復しながら配置しなければならず無駄な時間がかかるため, 特にプロトタイプデザイン作業には適していない. この問題を解決するため, ユーザがモデルの 2 次元スケッチをしていくだけで 3 次元シーンが構成できるシステムを提案する. このシステムはユーザのスケッチを入力とし, モデルのデータベースから検索を行い, 姿勢推定後画面にスケッチにフィットするように検索されたモデルを配置する. このシステムを利用することによってデザインは直観的に 3 次元シーンのプロトタイプを作ることが可能になる. 本手法の応用としては, 娯楽・会議・教育などの分野における 3 次元シーンプロトotypingツールなどが考えられる.

1 はじめに

3 次元シーンは, ゲームやインテリアデザイン, アニメーション, 映画など, 様々な分野で必要とされる. 3 次元シーンのデザインは, デザイナの描いたコンセプトスケッチから始まるが, 2 次元のスケッチを 3 次元シーンに変換する作業に既存の 3 次元ツールを用いると, 多数のメニューやコマンドを操作する必要があり, 素早く簡単にプロトタイプをデザインする作業には適していない.

本研究では, 手書きスケッチに基づき, 簡単に 3 次元シーンのプロトタイプをデザインするためのシステム, Magic Canvas を提案する. 本システムを用いると, ユーザはデザインしたいシーンのスケッチを描くことで, 3 次元シーンをデザインできる (図 1). ユーザが配置したいモデルの輪郭をスケッチすると, システムは自動的にそのスケッチに合うモデルを検索するとともに, そのスケッチにフィットするように, 検索されたモデルの姿勢を修正して配置する. 本システムを用いることで, ユーザは既存のツールのように, モデルのロードや編集コマンドを繰り返し使用する必要はない.

スケッチに基づく 3 次元形状モデリングの手法としては, SKETCH[14] や Teddy[6] などが有名である. また, 3 次元モデル検索エンジンとしては [4] などがある. 本研究の目的は, これら既存手法の長所を組み合わせることによって, 3 次元シーンをデザインするための直観的なユーザインタフェースを提案することである.

本手法の応用としては, インテリアデザインなど, 3 次元シーンのプロトotypingツールとしての利用をはじめ, 初心者のための娯楽用ソフトウェアと

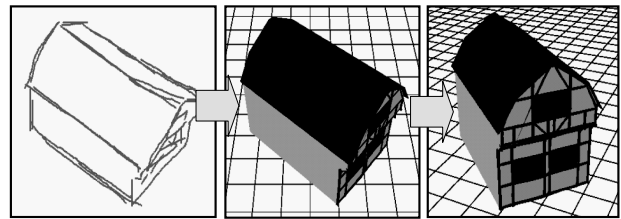


図 1. Magic Canvas システムの例: ユーザが 3 次元モデルをスケッチすると (a), システムは自動的にスケッチに合うモデルを検索し, 姿勢を修正して配置する (b). 3 次元シーンを別の視点から見た図を (c) に示す.

しての利用, 立体感を養うための教育用ソフトウェアなどが考えられる.

以降では, 関連研究に触れた上で, Magic Canvas 上で 3 次元シーンをデザインするためのユーザインタフェースの詳細について述べ, ついでアルゴリズムについて述べる.

2 関連研究

2.1 スケッチベースモデリング

2 次元の入力, 主に線画から 3 次元モデルを構築する手法は, 制約を解く Egli らの手法 [1] や, 最適化ベースの Lipson らの手法 [11][12], 最小化問題を解く Pentland らの手法 [9], 対称性を利用した Tanaka らの手法 [13] があるが, この他にも数多く存在する. 我々が特に興味をもっているのは 2 次元ジェスチャを利用した, インタラクティブ性の強いスケッチベース 3 次元モデルであり Zeleznik らの SKETCH[14] や, Igarashi らの Teddy[6][8] などの手法である.

Copyright is held by the author(s).

* HyoJong Shin and Makoto Okabe and Takeo Igarashi, 東京大学大学院情報理工学系研究科

2.2 スケッチベース検索

Funkhouser らは、ユーザのスケッチを入力に、3次元モデルを検索するシステムを提案している [4]。ユーザがクエリーとして入力できるのは、検索したい3次元モデルの3面図のスケッチか、Igarashi らの Teddy[6] を用いてモデリングした3次元モデルである。これに対して、我々のシステムでは、任意の視点から見たモデルのスケッチを与えることができる。また Fonseca らは、ユーザが描いたスケッチをもとに、データベースからクリップアートを検索するシステムを提案している [3]。しかし、ここでは検索対象が2次元ベクトル画像に限られている。これら既存研究に対し、我々は、検索対象のモデルを任意の視点から見たスケッチをクエリーに3次元モデルを検索し、更にスケッチにフィットする3次元モデルの姿勢を求めるシステムを提案する。

2.3 3次元姿勢推定

3次元世界に仮想的なカメラを置き2次元画像をレンダリングするのに対し、逆に最終的なシーンの見えを2次元画像で与え、カメラの姿勢を推定する手法にカメラ・キャリブレーション [2] がある。また、Gleicher らは、3次元シーン上に必ずカメラに写る点を指定し、それらを制約としながらカメラの姿勢を制御する手法を提案している [5]。提案手法では、これらの既存研究をもとに、3次元モデルの姿勢をユーザが与えたスケッチに基づいて推定する手法を提案する。

3 ユーザインタフェース

3.1 システム操作の概要

既存の3次元ツールが多くのメニューやボタンを提供しているのに対し、Magic Canvas ではユーザの手書きスケッチを入力としているので、ユーザは絵コンテのようなイメージを描きながら3次元シーンをデザインしていく。Magic Canvas システムのスクリーンショットを図2に示す。

スケッチウィンドウに表示されている床の上に、3次元モデルのラフなスケッチを描く(図3a,c,i)。ユーザはマウスの左ボタン+ドラッグ操作によってスケッチを入力することができる。スケッチが終わった後、“検索ボタン”を押すことで、3次元モデル検索によって検索されたモデルが、モデルウィンドウ内に検索スコアの順に表示される。また、一番検索スコアが高いモデルは、ユーザのスケッチにフィットするように配置される(図3b,d,j)。描いたストロークや配置されたモデルを削除する場合は、削除したい対象の上で“消しゴムジェスチャー”をすることで削除することができる(図3e)。配置されたモデルの姿勢が間違っただけの場合は上の姿勢ウィンドウから姿勢を選ぶことができる(図3f)。既に存在するモデル

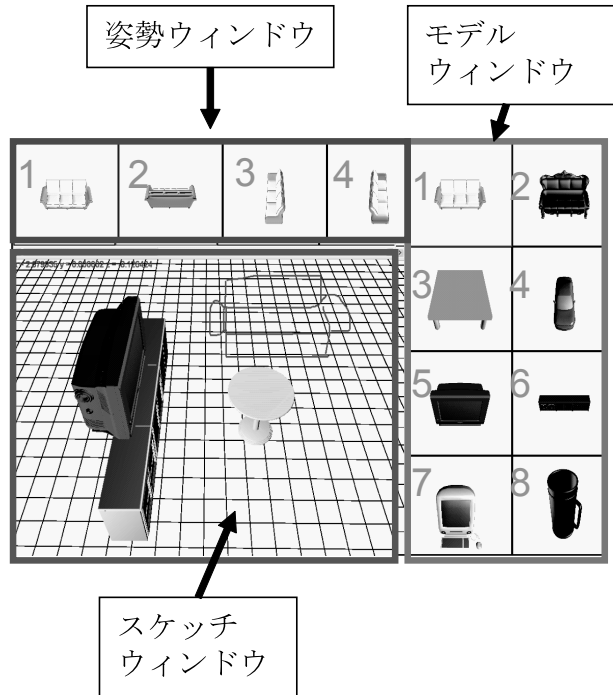


図2. Magic Canvas システムのスクリーンショット。モデルウィンドウ内、および、姿勢ウィンドウ内の数字は、そこに表示されるモデルの検索順位を表している。

の上にスケッチを重ねて行くと(図3g)、モデルの上に新たなモデルを配置することができる(図3h)。システムが自動的に配置したモデルが間違っていた場合は、他の候補モデルをモデルウィンドウから選んで変更することができる(図3i-k)。

また、あらかじめスケッチしたものがビットマップ画像としてあれば、それを読み込むこともでき、その場合は、検索に掛けたいスケッチ領域をストロークで囲んで選択することによって3次元モデルを検索、配置することができる(図3l-p)。

3.2 モデルの配置と削除

3次元モデルの輪郭を手書きスケッチで描くことによって、3次元モデルを配置できる。ユーザは検索したい3次元モデルの輪郭を描く必要があるが、1つの3次元モデルに対して、スケッチを行う際、ストロークの数や、ストロークを描く順序などに制限はない。

Magic Canvas では、シーン中に既に存在する3次元モデルの上に、新たなスケッチを重ねて描くことで、3次元モデルを既に存在する3次元モデルの上に配置することができる。通常の3次元ツールでは、モデルを他のモデルの上に乗せるためには、モデル同士の位置関係に気を配りながら移動、拡大/縮小、回転の作業を何度も繰り返さなければならず、

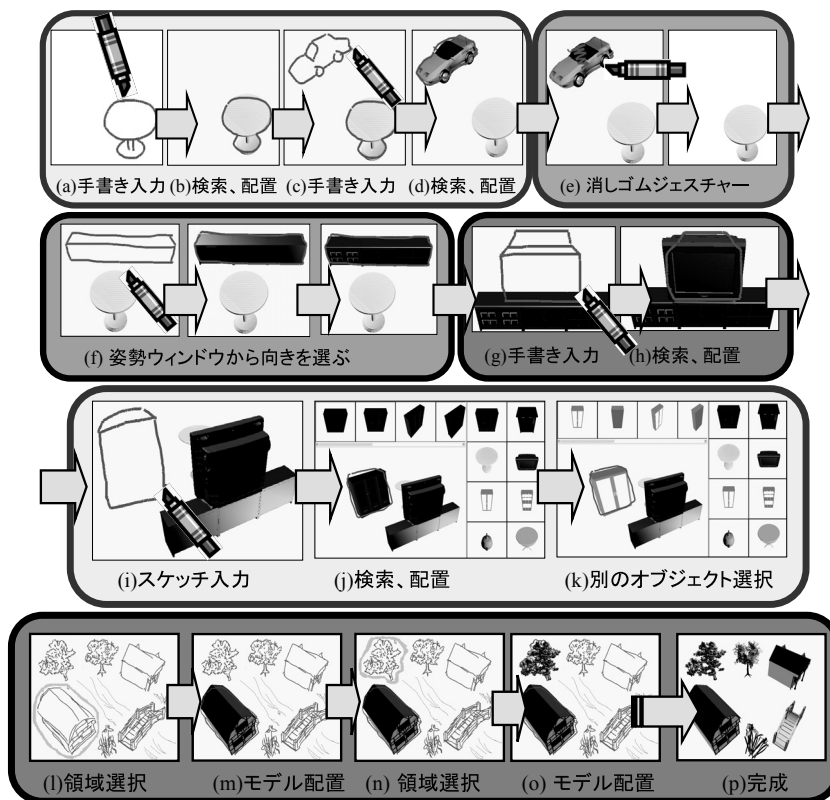


図 3. 3次元シーンデザインの概要

かなり時間がかかる作業である。

スケッチしている最中に、不要なストロークを見つけ削除したくなった場合や、一旦配置した3次元モデルを削除したくなった場合は、それぞれ、削除対象のストロークおよび3次元モデルの上で、“消しゴムジェスチャー”を行うことで、消去を行うことができる。

3.3 3次元モデルの変更

3次元シーンをデザインしながら、既に配置した3次元モデルを、一旦他の類似したモデルで置き換えて比較したい場合がしばしばある。たとえば、ベッドルームをデザインしている際に、ベッドのモデルが多数存在する場合などである(図4)。通常の3次元ツールを利用した場合には、新たなベッドを選択して最初のベッドを配置したのと同様の作業を繰り返さなければならない。ベッドのモデル数が多いほど、モデルの変更に対する欲求は多くなり、その結果、冗長な操作をたくさん行わなければならない。

Magic Canvas では、3次元モデルの変更を行う場合は、変更したい3次元モデルをクリックする。クリックされたモデルが配置された時の2次元スケッチが記憶されているので、モデルウィンドウに、再び検索結果の候補一覧が表示される。この候補モデル

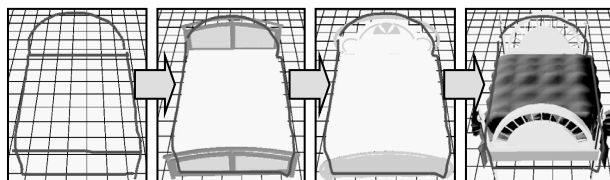


図 4. ベッドの選択画面

をクリックすることで、類似モデルとの変更することができる。

3.4 ビットマップ画像の利用

あらかじめ描かれた絵コンテなどのビットマップ画像を読み込み、画像中のスケッチを選択することで、選択されたスケッチをクエリーに、3次元モデルを配置することができる。画像中のスケッチの選択は、領域を囲むストロークを描くジェスチャーを使用する。

4 アルゴリズム

提案システムは、ユーザのスケッチ入力に基づき適切な3次元モデルをデータベースから検索した上で、スケッチにフィットするようにその3次元モデルの姿勢を推定して3次元シーンに配置する。以下

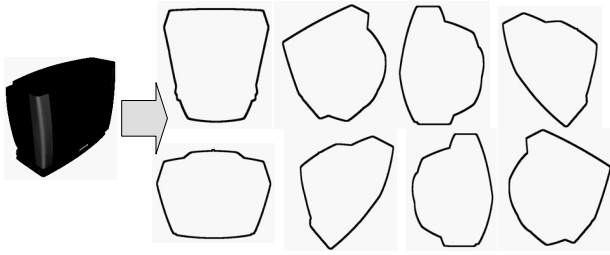


図 5. テレビモデルの斜め上から 8 つの視点で見た際の輪郭情報

に、3次元モデル検索、3次元モデルの姿勢推定の各々のアルゴリズムについて述べる。

4.1 3次元モデル検索

本システムでは、検索対象となる3次元モデルのデータベースを作成する。ユーザから与えられる2次元スケッチは3次元モデルの輪郭を意図して描かれているので、データベース内にも、同様に3次元モデルの輪郭情報を保持して、マッチングに使用する必要がある。大体の3次元シーンは全体的な特徴を表現するため、斜め上(クォータビュー)の視点を利用している。データベースには、1つの3次元モデルに対して、図5に示すように、斜め上から8つの視点で見た際の輪郭情報を保持しておく。8つの視点を選択したのは実験の結果、最も検索効率が良いからである。

個々の輪郭情報は、画像そのものではなく、特徴ベクトルに変換されてデータベースに格納される。本システムでは、検索効率を上げるために、2種類の特徴ベクトルを利用している。

1つ目の特徴ベクトルはCentroid Fourier Descriptor[10]を用い、輪郭の全体的な特徴を捉えるものである。まず、図6(a)の3次元モデルの輪郭線を抽出し、図6(b)に示すように、中心点を輪郭線画像の真ん中に取り、その中心点から最も遠い輪郭線までの距離を測定する。測定を中心点の周りに360度行うことで、図6(c)に示すようなデータが得られる。このデータに対して離散フーリエ変換を行った結果から得られるデータが特徴ベクトルとなる(図5(d))。

2つ目の特徴ベクトルは、輪郭の局所的な特徴に注目するIpらの手法を用いた[7]。Ipらの手法では、図7(a)に示すように、輪郭線上に特徴点(コーナーに当たる点など)を求め、中心点からそれら特徴点までの距離のみをサンプリングしてデータ配列を作る(図7(b))。本システムではサンプリングを10度の範囲で36回行った。サンプリング値は10度の範囲での合計値を用いている。

ユーザの手書きスケッチが入力されると、システムはスケッチに対して、同様に、2種類の特徴ベク

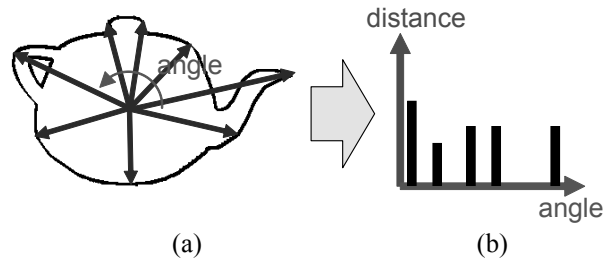


図 7. Ip らの手法による特徴ベクトル: (a) 中心点から輪郭線上の特徴点への距離を左周りの回転角に対してサンプリング (b) サンプリングされたデータの特徴ベクトルとして用いる

トルを計算する。計算された特徴ベクトルに対し、データベース内の特徴ベクトルとのユークリッド距離をスコアとして比較する。このとき、スコアは2種類の特徴ベクトルに対するスコアの線形和をとったものとする。システムはスコアの順にモデルウィンドウに3次元モデルを表示する。

4.2 3次元モデルの姿勢推定

検索されたモデルをユーザのスケッチにフィットさせるアルゴリズムについて述べる。

図8に示すように、ユーザのスケッチが p_1, p_2, \dots, p_n からなる点列で与えられているとする。一方、3次元モデルの各頂点 v_i は、3次元モデルの姿勢を決める変換 M_{object} と、カメラの姿勢を決める変換 M_{camera} によって、 $q_i = M_{camera} \times M_{object} \times v_i$ として画面上に射影される。ここで解くべき問題は、対応する p_i と q_i の距離がなるべく近くなるように、 M_{object} を推定することである。すなわち、

$$E = \sum_i (M_{camera} \times M_{object} \times v_i - p_i)^2, \quad (1)$$

を最小にするような M_{object} を最小二乗法によって推定する。

p_i と v_i は特徴ベクトルを求めるときに得られた点列である。本システムでは、変換 M_{object} として、アフィン変換のうち移動、拡大/縮小、回転だけを取り扱う。さらに、3次元モデルは常に床、もしくは他の3次元モデル上に接していると仮定する。そのため、 M_{object} ではy軸周りの回転だけをサポートする。また、拡大/縮小に関しては、縦横高さ方向に等倍率の拡大/縮小だけをサポートする。これは、ユーザのスケッチ入力の不正確さによって、容易に3次元モデルが、歪んでしまうのを防ぐためである。以上、移動、拡大/縮小、y軸周りの回転に相当する変数を未知数として、式(1)の最小化することによって、3次元モデルの姿勢を決定する変換 M_{object} を得ることができる。

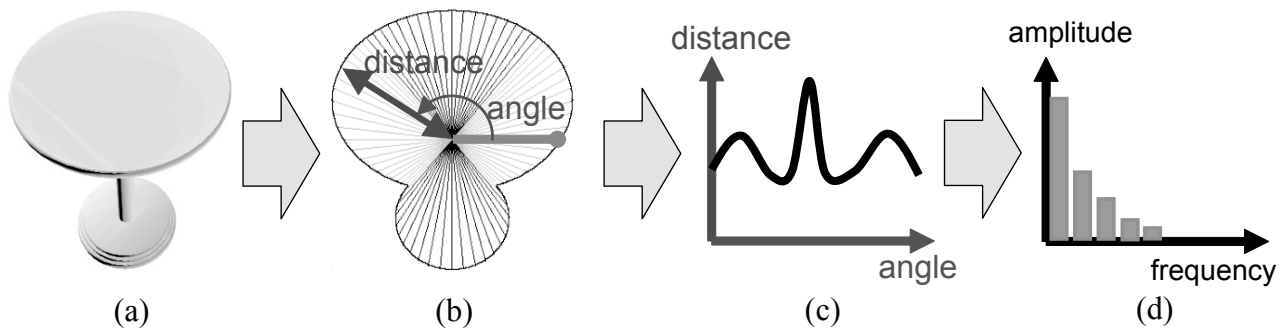


図 6. Centroid Fourier Descriptor による特徴ベクトル: (a) 3次元モデル (b) 中心点から輪郭線への距離を左周りの回転角に対してサンプリング (c) サンプリングされたデータ (d) データをフーリエ変換した結果を特徴ベクトルとして用いる

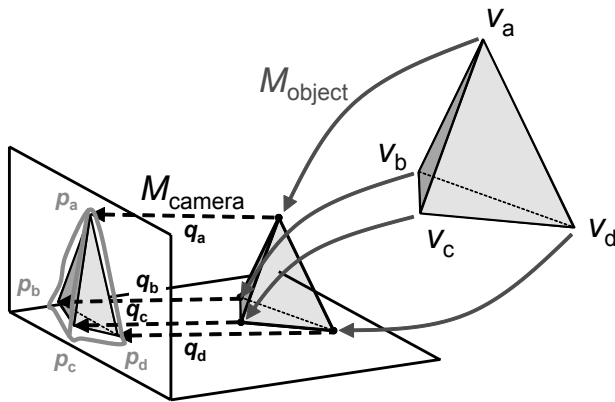


図 8. ユーザのスケッチに基づく姿勢推定: ユーザが描いたオレンジ色のスケッチに、レンダリング画像がフィットするように、3次元モデルの姿勢=アフィン変換 M_{object} を推定する。

5 実装

Magic Canvas システムは Visual Studio .NET 2003 の MFC を用いて実装されている。3次元グラフィックスシステムの実装には、DirectX SDK 9.0 を利用した。

ユーザの2次元スケッチに基づく3次元モデルの検索や配置は瞬時に実現されるが、データベースに3次元モデルを登録する際の事前処理には、モデルごとに2～3秒を費やす。これは登録処理で、ビデオメモリを頻繁にアクセスする必要があるためであり、今後、最適化してパフォーマンスの向上を図ることが可能である。

6 結果

図9に、提案システムを用いてデザインした3次元シーンを示す。1,2行目の図は筆者ら、3行目の図はユーザ1名がデザインしたものである。野外、リビングルーム、ベッドルーム、乗り物のカテゴリ

を分けて、カテゴリ別20個ずつのモデルを入れて実験した。実験の参加者は1人だったがモデルウィンドウの上位2個以内にユーザが望むモデルが入るケースは約90%を示し、ユーザは次の候補を表示する操作をほとんどせずにシーンを構成することができた。ユーザのスケッチ時間は約25分、3次元シーンの構成の時間は約5分かかった。

7 まとめと今後の発展

本稿では、手書きスケッチに基づく対話的な3次元シーンのデザイン手法、およびそのアルゴリズムについて紹介した。本手法を利用することにより、通常の3次元ツールの扱いに熟練したデザイナーだけでなく、初心者も使用法を学習する手間をそれほどかけずに、直観的に3次元シーンをデザインすることが可能となる。

今後の発展としては、検索アルゴリズムの改良による検索率の向上に取り組む予定である。また、今回は3次元モデル単位での検索のみをサポートしたが、メッシュセグメンテーションにより3次元モデルを代表的なパーツごとに分割し（例えば、家のモデルを、屋根、壁、窓、ドアに分割するなど）、部品単位での検索、組み合わせによる3次元モデルデザインなどにも取り組みたい。また、複数の絵コンテをキーフレームとして利用することで、アニメーションを生成できるようなユーザインタフェースの研究も行う予定である。

参考文献

- [1] L. Egli, C. yao Hsu, B. D. Brüderlin, and G. Elber. Inferring 3D models from freehand sketches and constraints. *Computer-Aided Design*, 29(2):101–112, 1997.
- [2] O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, MA, USA, 1993.

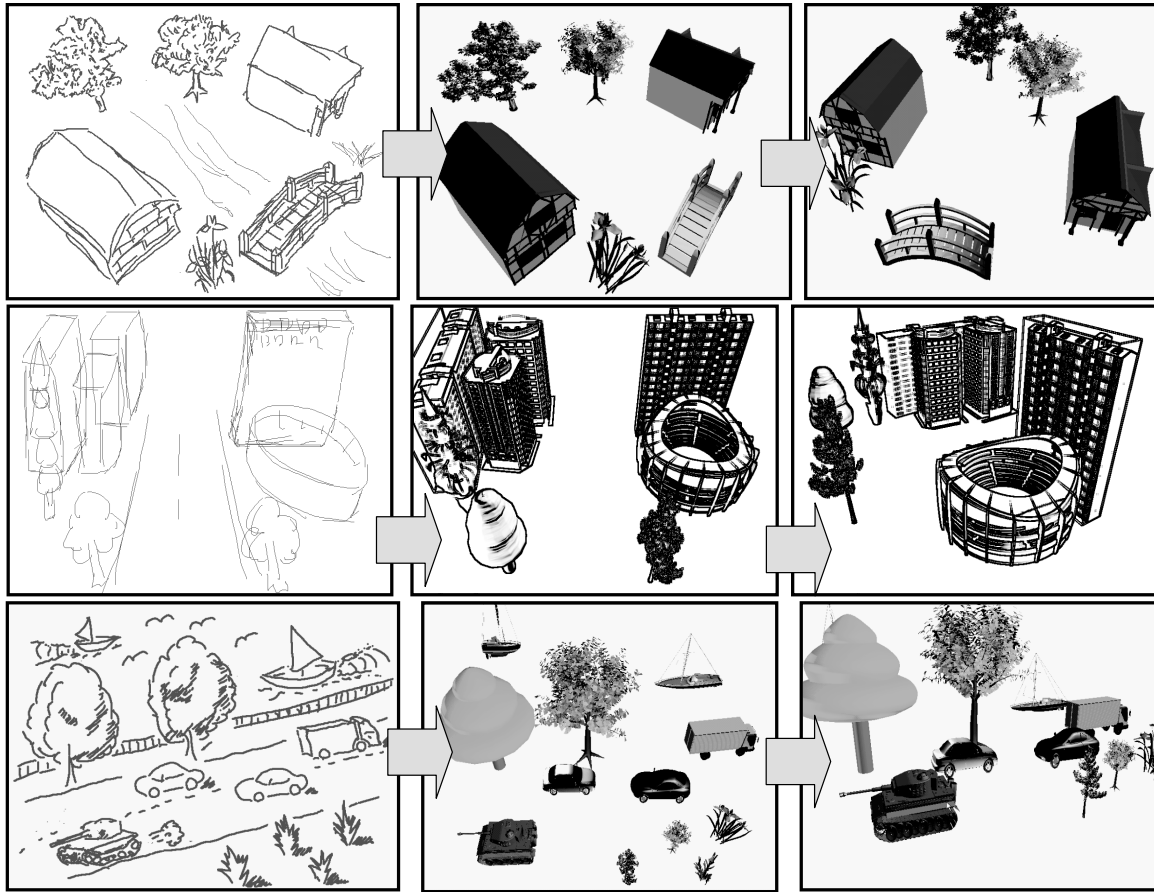


図 9. 2次元スケッチをもとにデザインした3次元シーン結果．左列にユーザが与えた手書きスケッチと中央列に完成した3次元シーン，右列に回転されたシーンを示す．

- [3] M. Fonseca, B. Barroso, P. Ribeiro, and J. Jorge. Sketch-based retrieval of ClipArt drawings. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pp. 429–432, New York, NY, USA, 2004. ACM Press.
- [4] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A search engine for 3D models. *ACM Trans. Graph.*, 22(1):83–105, 2003.
- [5] M. Gleicher and A. Witkin. Through-the-lens camera control. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pp. 331–340, New York, NY, USA, 1992. ACM Press.
- [6] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: a sketching interface for 3D freeform design. In *Proceedings of SIGGRAPH '99*, pp. 409–416. ACM Press, 1999.
- [7] H. H. S. Ip, A. K. Y. Cheng, W. Y. F. Wong, and J. Feng. Affine-Invariant Sketch-Based Retrieval of Images. In *CGI '01: Proceedings of the International Conference on Computer Graphics*, p. 55, Washington, DC, USA, 2001. IEEE Computer Society.
- [8] S. Owada, F. Nielsen, M. Okabe, and T. Igarashi. Volumetric illustration: designing 3D models with internal textures. *ACM Trans. Graph.*, 23(3):322–328, 2004.
- [9] A. Pentland and J. Kuo. The Artist at the Interface. *Vision Science Technical Report 114.*, 1989.
- [10] M. Safar, C. Shahabi, and X. Sun. Image retrieval by shape: a comparative study. *IEEE International Conference on Multimedia and Expo*, pp. 141–144, 2000.
- [11] M. Shpitalni and H. Lipson. Identification of Faces in a 2D Line Drawing Projection of a Wireframe Object. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(10):1000–1012, 1996.
- [12] M. Shpitalni and H. Lipson. Optimization-based reconstruction of a 3D object from a single free-hand line drawing. *Computer-Aided Design*, 28(8):651–663, 1996.
- [13] T. Tanaka, S. Naito, and T. Takahashi. Generalized symmetry and its application to 3D shape generation. *The Visual Computer*, 5(1&2):83–94, 1989.
- [14] R. C. Zeleznik, K. P. Herndon, and J. F. Hughes. SKETCH: an interface for sketching 3D scenes. In *Proceedings of SIGGRAPH '96*, pp. 163–170. ACM Press, 1996.