

建築物内部における簡易なイメージベース手法

5E-04

須田聰子 斎藤隆文†

東京農工大学大学院工学研究科電子情報工学専攻

†東京農工大学工学部情報コミュニケーション工学科

1. はじめに

著者らは、建築物内部をウォークスルーできるような仮想環境を、複数の実写画像を用いて低コストで簡易に作成する手法の開発を目指す。

実存する建築物をデジタルデータとしてコンピュータに取りこみたいという需要は多くある。例えば、学校や公共施設の紹介、映画やゲームなどである。その場合、コンピュータグラフィクスを用いて実際のデータから再構築した三次元モデルを作成したり、ビデオを利用することが最近の主流になっている。しかし、ビデオではインタラクティブな動きに対応できない。モデルを作成するためには対象物の三次元データが必要であり、対象が建築物の場合は特に、実際のデータを測量することはかなりの労力が必要となる。その上、コストをかけなければ結果が不自然になるといった欠点がある。

そこで本研究では、実写画像から三次元形状を再構築するイメージベース手法を利用する。実測データからモデル構築をするのは非常にコストがかかるため、実写画像から比較的容易にリアルな三次元形状を再構築できるこの手法は近年研究が盛んに行われている。この手法を用いることにより、個人単位などでも手軽に仮想建築物モデルを作成することが可能になる。

本研究では、人工的な建築物の内部を対象とした画像から直線と消失点を抽出、利用することで三次元情報の手がかりを得る。また、複数画像を入力に用いることで多くの状況に対応したウォークスルーを可能にする。これにより、手軽で精度の高い三次元仮想環境の再構築を行う手法を提案する。

"Simple Image-Based Method for Building Inside"

Satoko SUDA, Takafumi SAITO†

Division of Electronic and Information Engineering,
Graduate School of Technology, Tokyo University of
Agriculture and Technology†Department of Computer, Information and Com-
munication Sciences, Faculty of Technology, Tokyo
University of Agriculture and Technology

2. 従来手法

2.1. QuickTime VR

QuickTime VR[1]は、イメージベース手法の皮切りになった手法の一つである。複数の視点の異なる実写画像から円柱投影のパノラマ画像を作成することで、簡単な三次元仮想環境が構築できる。

この手法では、視点は離散的な動きに限定されるが、周囲を 360 度見渡すことと、ズーム動作ができる。他の視点にジャンプすることで広いシーン内を動くことに対応している。

2.2. TIP

特殊な装置を要さず、手軽に使えるイメージベース手法の一つとして **Tour Into the Picture(TIP)[2]**が挙げられる。TIP は一点透視の二次元画像 1 枚から画像内をカメラが三次元上に動くようなアニメーションを作成するシステムである。シーンを直方体の内面と仮定した疑似三次元構造を構築することで、少ない三次元情報から簡単に自然なアニメーションの生成を可能にしている。簡単な三次元構造しか持たないことから、厳密な三次元構造を構築し難い自然景観において、特に有効である。

TIP では、三次元構造を生成する際に **Spidery Mesh** と呼ばれる GUI が提案されている。これは消失点、消失点から放射状に出る線、内部の長方形、外部の長方形から成り立つ。これをユーザが画像上でマウス操作により指定する。**Spidery Mesh** を指定すると画像は五つのポリゴンに分割される。この各ポリゴンが直方体の五つの面(上下, 左右, 奥)に対応すると仮定し、三次元直方体モデルを得る。このようにして二次元画像から簡単に三次元モデリングを行うことができる。

2.3. 従来手法の問題点

紹介した二つのイメージベース手法は、どちらもシーンを簡単な疑似三次元構造で表すことで、手間のかかるモデリングや特殊な装置を要さずに仮想環境を生成できる。しかし、以下に示す問題点がある。

QuickTime VR は、連続的な移動ができない。また、

シーンに基づいた三次元モデルを持っていないため、不自然な歪みが生じる。

一方 **TIP** は、簡単ではあるが、三次元モデルを作成しているため、よりリアルなレンダリングが可能になっている。しかし、入力画像が厳密な一点透視画像 1 枚であることから、視点移動操作が制限される上、入力画像への制限も大きい。また、**Spidery Mesh** が画像のエッジに正確に沿わせないと、生成画像に歪みが生じる。建築物のように直線が多いシーンに適用した場合、特にその歪みが目立ってしまうため、**Spidery Mesh** の指定の正確さが重要になる。この指定はユーザによる手動処理であるため、正確な消失点位置を模索しづらく、精度を追求するほどユーザへの負担が多くなる。

3. 提案手法

3.1. 提案手法の特徴

本研究は **TIP** に基づき、人工的な建築物の内部を対象として **TIP** を使用する際の問題点を改善することで、より手軽で精度の高い三次元仮想環境の再構築を行うシステムの開発を提案する。

ここで扱う入力画像の前提を以下に述べる。

- (1) 建築物内部を撮影したもので、画像中に三直交直線群を多く含んでいる。
- (2) カメラパラメータは未知とする。画像中心(光軸)の位置は、既知(通常は入力画像の中央)でも未知でもよい。
- (3) レンズ収差による歪みはない(補正済み)とする。
- (4) 直方体構造で、障害物の少ない廊下を対象としたものとする。廊下の分岐、階段等の対応は今後の課題とする。

提案手法の基本的な方針は、第一に、画像処理によってあらかじめ消失点や直線を抽出することで作業を手動から半自動化し、画像上での形状指定におけるユーザへの負担を軽減すると共に精度を向上させる。

第二に、画像から抽出した直線を、**Spidery Mesh** のエッジ候補として得ることで入力画像が厳密な一点透視でない画像でも正確な形状の指定を可能にする。これにより、入力画像が厳密な一点透視に限定されるという入力画像の制限が緩和される。

最後に、入力画像を 1 枚ではなく複数枚に対応できるようにすることで、より自由なワークスルーを実現させる。

3.2. 消失点および直線抽出

本研究では、**TIP** の前処理として有限円形領域を用いた消失点抽出システム[3]を加えることで問題点の改善を図る。

三次元空間中の平行直線群を透視投影により二次元平面に投影すると、直線は消失点で交わる。この性質から、画像中の直線を **Hough** 変換により抽出、直線群の交点となる消失点を得る。しかし、消失点を抽出するには、消失点が画像上では無限遠点もしくは有限でも極めて遠い点に位置することが多く、統一的に扱うことが難しいといった問題がある。そこで、このシステムでは無限平面を有限円形領域に投票することでこの問題を解決している。抽出された直線群を有限円形領域に変換して表示した画面上で、ユーザが画像から抽出された直線が交わる点をマウスでクリックして選択することにより消失点を指定する。この操作は、消失点が画像中に点として分かりやすく表示されるため、手動ではあるが、より直接的に消失点が指定できる。

さらに、抽出した消失点を用いて、消失点を正確に通る直線群だけを再抽出する。抽出した消失点、直線群を **Spidery Mesh** の候補とすることで、指定時におけるユーザの負担を減らす。

また、消失点 3 点と画像中心 1 点の計 4 点のうち、任意の 3 点が定まれば残りの 1 点は一意に決まるといえる。この性質から、画像中心が未知の画像から画像中心を抽出することができ、トリミング画像も扱える。

このシステムでは、入力画像中の直線を延長させ、交わる点を消失点として指定するので、当然画像中の直線は多いほどその精度は上がる。しかし、実際の画像ではうまく直線が抽出されない場合も多い。この問題は、輝度差の小さいエッジを強調させる画像処理[7]を行うことで改善される。

3.3. Spidery Mesh 指定

消失点抽出システムで入力画像中より抽出された消失点、直線を **Spidery Mesh** 指定に利用する。まず、消失点 3 点のうち画像内に現れる 1 点が **Spidery Mesh** 中の消失点となる。消失点から放射状に出る線は消失点と内部の長方形の頂点を通る線であるから、内部の長方形が指定されれば一意に決定する。そこで、ユーザが行う操作は内部の長方形の指定のみに限られる。

内部の長方形を指定するにはユーザが画像上で三次元直方体の奥面になりうる対象を見ながらマウス、キ

一ボード操作でエッジの当てはめを行う。キーボードの操作方法は、矢印キーを用い、キーを押すとその方向で最も近い辺の候補を指定する。この候補は前節で再抽出した直線群である。これにより、画像中のエッジに沿った正確な当てはめを行うことが可能になる。また、マウスから手動による任意の位置決定も行える。この二つの操作を駆使することで、エッジの弱い部分や見えない部分をユーザに依存でき、柔軟な対応が可能である。

また、入力画像が厳密な一点透視でない場合も、指定した消失点を通る直線群を内部の長方形の候補線とすることにより、一点透視の場合と全く同じ操作でモデリングが行えるようになる。

この手法を用いることで **Spidery Mesh** の指定が簡易かつ正確に行えるようになり、同時に入力画像の制限も緩和される。

3.4. 三次元再構築

Spidery Mesh を指定したデータから三次元再構築を行う。透視投影された直方体の三次元座標は、直方体の 1 頂点に奥行き値を与えることで算出できる。三次元座標は、その座標値自体には意味がなく、直方体の各辺の比率が実際のシーンに一致したものである。よってここでは、内部の長方形の 1 頂点を奥行き値 0.0 であると仮定して他の三次元座標を算出し、三次元直方体モデルを作成する。

3.5. テクスチャマッピング

前節で再構築した三次元直方体モデルの各面にテクスチャを貼りつける、テクスチャマッピングを行う。ここでテクスチャは生成する面上の位置を逆変換して入力画像上での位置とその画素値を特定することで生成する。

3.6. 複数画像からのモデリング

TIP では、入力画像は 1 枚に限られていたが、建築物内部を自由にウォークスルーするためには複数画像からモデリングを行う必要がある。そこで、入力画像を複数枚扱えるように機能を拡張する。

3.6.1. 同一方向を向いて撮影した画像

違う位置から同一方向を向いて撮影した画像を入力画像として扱う場合を考える。1 枚の入力画像から復元した三次元モデルに対して奥行き方向に視点を移動さ

せると、テクスチャが粗くなり、不自然さが目立つ。同一方向を向いて撮影した画像を複数枚組み合わせることでテクスチャを作成することでこの問題を解決し、よりリアルな仮想環境の再構築を目指す。

入力画像全てに対して消失点抽出、**Spidery Mesh** 指定処理を行い、三次元直方体モデルの座標を算出する。モデリングには、奥面からの距離が最も遠い画像から得られた直方体モデルを採用する。各画像から得られた三次元直方体モデルを、直方体の奥面が同じサイズであるという条件からスケールを一致させる。これと同様に変形したテクスチャを、奥の面に接する辺が一致するように重ね合わせることで新しいテクスチャを作成する。境界部分は、徐々に画素の割合を変えて重ね合わせることで目立たないように配慮している。

同一方向を向いて撮影した画像を複数枚扱う場合は、消失点抽出、**Spidery Mesh** 指定の操作が画像数に比例して多くなる。

3.6.2. 逆方向を向いて撮影した画像

次に、逆方向を向いて撮影した画像を扱う場合を考える。これにより、振り向くような視点の動きにも対応できるようになる。

同一方向を向いた場合では、奥の面が同じであるという条件から、単純にテクスチャを重ねることが可能であった。しかし、逆方向を向いた場合には、その条件は成り立たず、ユーザが新たな条件を設定しなければならない。そこで、**Spidery Mesh** 中の外部の長方形を利用する。外部の長方形は三次元直方体モデルの手前の面に相当し、入力画像が 1 枚、または同一方向を向いて撮影した画像である場合は、入力画像の枠がその役目を果たしていたため、ユーザが指定する必要はなかった。ここでは、ユーザが内部の長方形同様、外部の長方形をマウス、キーボード操作をして、境界の目安になり得るエッジを指定することで境界位置を決定している。

モデリングは、得られる三次元直方体を組み合わせることで新しい直方体モデルを作成する。その際、各画像から作成された直方体のスケールを一致させる必要がある。一方の三次元直方体モデルを基準にし、互いの外部の長方形にあたる三次元座標が一致するように、もう一方の三次元直方体モデルを変換させる。変換作業はまず、スケールが一致するように拡大縮小させる。次に奥の面同士が向かい合うような向きに回転させる。各画像は撮影時の視線方向ベクトルが異なるため、ベク

トルが一致するように回転をさせる。こうした工程を得た上で合体させ、新しく作成された直方体モデルを採用する。

直方体モデルと同様に変形されたテクスチャを外部の長方形が一致するようにつなぎ合わせて新しいテクスチャを作成、マッピングすることで三次元仮想環境が完成する。

逆方向を向いて撮影した画像を複数枚扱う場合は、消失点抽出、外部の長方形指定を加えた **Spidery Mesh** 指定の操作が画像数に比例して多くなる。

4. 結果

本手法を用いたシステムの実験結果を述べる。まず、入力画像が 1 枚の場合、本手法の過程を交え、結果を示す。入力画像を図 1(a)とする。この入力画像に関して消失点抽出システムを施した結果が図 1(b)である。この画面で消失点である直線群の交点をユーザが 3 点指定する。ここでは、画像中心は入力画像の中央と定まっている。消失点から直線の再抽出を行い、これらのデータを利用して **Spidery Mesh** を指定する。**Spidery Mesh** 指定の初期状態を図 1(c)に示す。初期状態では、消失点がすでに一意に定まっており、内部の長方形は消失点を通る直線で成り立った適当な大きさになっている。この状態からマウス、キーボード操作をして

Spidery Mesh を指定した状態を図 1(d)に示す。この指定したデータを利用して三次元構築を行う。視線方向および視点を変更した結果を図 1(e)に示す。この図は視点を撮影時から左方向、前方向に移動させ、視線方向ベクトルを左上方向に回転、つまり左上を見上げたようになっている。

次に、入力画像が複数枚の場合で、同一方向を向いて撮影した画像を扱った結果を述べる。ここでは入力画像を 3 枚とする。入力画像を図 2(a)~(c)に示す。各画像に消失点抽出、**Spidery Mesh** 指定を行い、三次元構築後、視点を移動させた結果を図 2(d)に示す。比較のため、1 枚の入力画像、ここでは図 2(c)に示した画像から三次元再構築し、同じ位置に視点を移動させた結果を図 2(e)に示す。

最後に、逆方向を向いて撮影した画像を扱った結果を述べる。ここでは入力画像を 2 枚とする。この場合は **Spidery Mesh** 指定に外部の長方形の指定も必要となる。入力画像 2 枚に対して **Spidery Mesh** を指定した結果を図 3(a), (b)に示す。画像中のドアのエッジを利用して外部の長方形を指定している。このデータから三次元構築を行った結果が図 3(c), その視点位置から視点を前方に移動させて振り向いた状態が図 3(d)である。

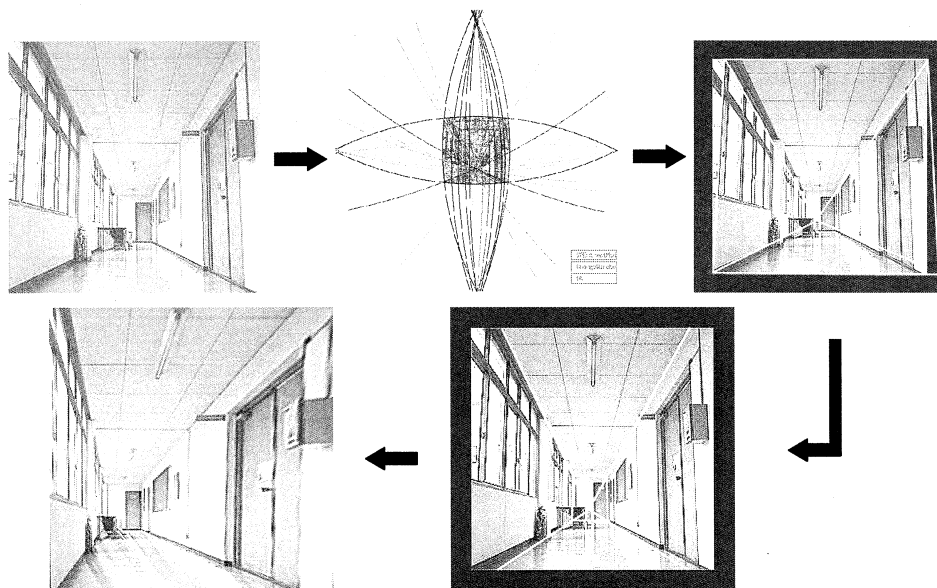


図 1. (a:左上) 入力画像 (b:上中) 直線群の円形領域表示 (c:右上) **Spidery Mesh** 初期画面 (d:右下) **Spidery Mesh** 指定画面 (e:下左) 三次元再構築結果

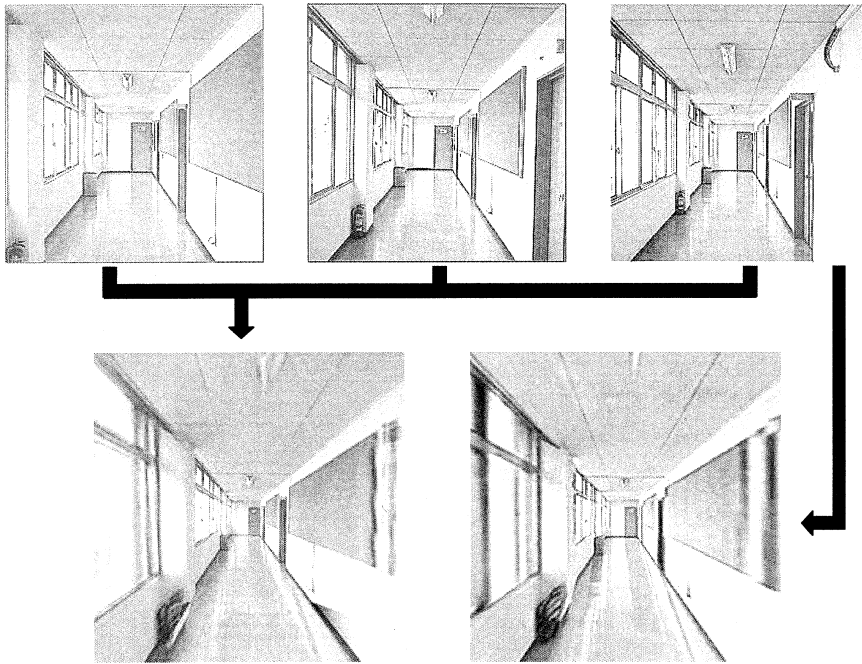


図 2. (a:上左) (b:上中) (c:上右) 入力画像
 (d:下左) 図 2(a)~(c)を入力画像とした三次元再構築結果
 (e:下右) 図 2(c)を入力画像としたの三次元再構築結果

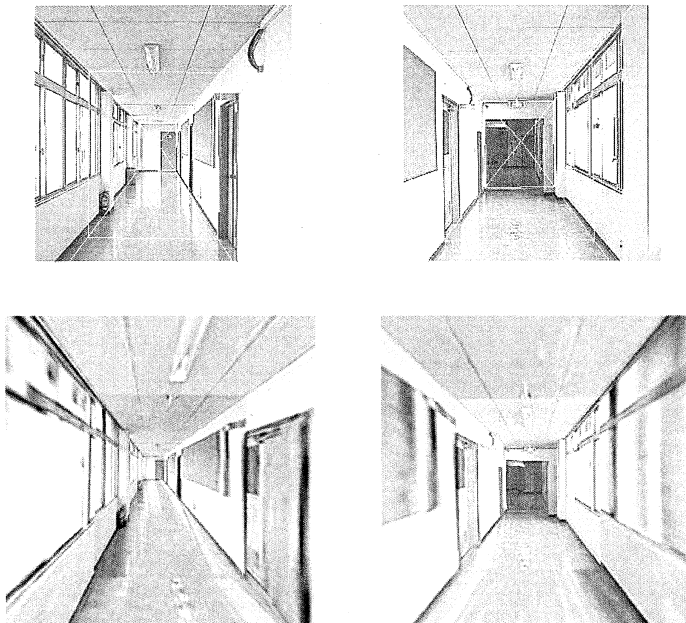


図 3. (a:上左) (b:上右) Spidery Mesh を指定した入力画像
 (c:下左) (d:下右) 三次元再構築結果

5. 考察

結果をふまえ、考察を行う。複数画像を入力とすることでより様々な状況に対応できるようになった。1枚の画像から作成した直方体側面のテクスチャを図4(a)に示す。この図から特に、奥の面に近い部分が粗くなっていることが分かる。そこで、同一方向から撮影した画像を組み合わせてテクスチャを作成することで、視点が前に移動すると目立つ画像の粗さが改善した。複数画像から作成したテクスチャの図4(b)に示す。図2(d), (e)の比較から同様、改善されたことが分かる。

逆方向から撮影した画像を扱うことで振り向くという動作が可能になった。ここでは、テクスチャの境界に平均化などの処理をしていないので少々不自然さが残るが、この点をは同一方向を向いて撮影した場合同様の処理を加えることで改善される。

複数画像を用いた場合には、画像の境界部分が目立つ問題が残る。入力画像の鮮度や解像度の違いがその境界を強調する一因となっている。また、Spidery Mesh 指定時の誤差、側面の凹凸などからずれが生じる場合がある。それらの改良が今後の課題となる。

入力画像数が多いほどより自然なレンダリングができるが、全ての画像に消失点抽出、Spidery Mesh 指定が必要になるため、手間が多くなる。この自然さと手間とのトレードオフのため、画像間の距離等のガイドラインが必要となる。

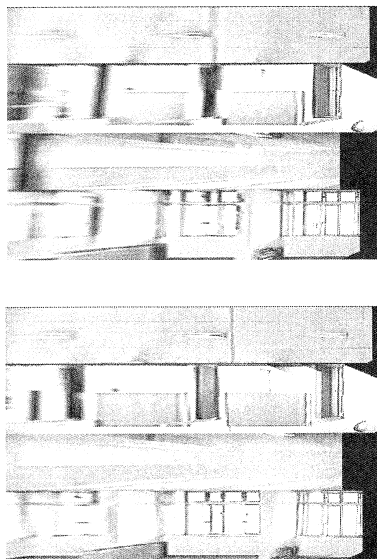


図4. (a:上) 1枚の入力画像からのテクスチャ
(b:下) 複数画像からのテクスチャ

6. まとめ

本研究では、実写画像を用い、消失点および直線抽出結果に基づく候補直線群からユーザが選択操作を行うことで、従来手法に比べより簡単な操作で、効率的な建築物内部をウォークスルーするためのモデリングを実現した。

今後は、より多くの状況に対応できるように改善すると共に、GUIの操作方法の更なる改良を目指す。

参考文献

- [1] Shenchang Eric Chen: "QuickTime VR -An Image-Based Approach to Virtual Environment Navigation", Proc. SIGGRAPH '95, pp. 29-38 (1995).
- [2] Y. Horry, K. Anjyo, K. Arai: "Tour Into the Picture: Using a Spidery Mesh Interface to Make Animation from a Single Image", Proc. SIGGRAPH '97, pp. 225-232 (1997).
- [3] 松藤和夫, 斎藤隆文: "無限平面内での消失点抽出", 情報処理学会研究報告, Vol. 99, No. 70, pp. 25-30 (1999).
- [4] P. Debevec, C. J. Taylor, H. Nalik: "Efficient Modeling and Rendering Architecture from Photographs: A hybrid geometry- and image-based approach", Proc. SIGGRAPH '96, pp. 11-20 (1996).
- [5] Paul Debevec, Yizhou Yu, George Borshukov: "Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping", Proc. Eurographics Workshop on Rendering, 1998.
- [6] 松藤和夫, 須田聡子, 斎藤隆文: "消失点抽出に基づくイメージベースモデリング", Visual Computing グラフィクスと CAD 合同シンポジウム 2001 予稿集, pp.157-162 (2001)
- [7] 白倉健太郎, 瀬川大勝, 斎藤隆文: "多重解像度解析を用いた非写実的画像生成", 情報処理学会第64回全国大会, 1F-01, 2002