

イメージベースレンダリングのための 建築物テクスチャの対話的修正

塩田真之 古谷雅理 宮村(中村)浩子 斎藤 隆文
東京農工大学 大学院 生物システム応用科学府

本稿では、イメージベースレンダリングに使われるテクスチャを簡単に修正する手法を提案する。イメージベース手法で建築物の3次元形状を作成する場合、建築物周囲に木や車などの写り込みが作成される建築物に影響を及ぼす。そこで、校舎やビルなど繰り返し構造を持った建築物の特徴を利用し、テクスチャを格子状に分割し、情報が失われている格子に対して同じ構造の格子のテクスチャで置き換える。格子の生成は輪郭線抽出により自動的に行い、微調整、格子内のテクスチャの更新は手動で行う。これにより簡単かつ効果的な画像修正が可能となった。

Interactive Improvement of the Texture of Architecture for Image-based Rendering

Masayuki SHIOTA, Tadasuke FURUYA
Hiroko MIYAMURA(NAKAMURA), and Takafumi SAITO
Graduate School of Bio-Applications and Systems Engineering,
Tokyo University of Agriculture and Technology

In this paper, we propose interactive improvement of the texture of architecture for image-based rendering. In image-based techniques, the quality of input photographs affects the rendering result directly. Since obstacles such as trees appear in a photo are mapped onto the synthesized architecture, it is necessary to clear such obstacles. In our proposed method, the building texture is divided with rectangular grids by edge detection. Then, the user replaces each texture part including obstacles with another part that has the same structure but no obstacle. By using the building regularity, an effective correction can be easily done.

1. 緒言

コンピュータグラフィックス技術の発展により現実と見間違ふほど高品質な建築物のモデルをコンピュータグラフィックスで生成できる。その用途はゲームや映画、都市景観シミュレーションなど様々である。しかし、高品質なモデルを作成するには多大な時間や手間を要する。そこで、3次元物体の作成労力を削減するため実存写真を元にモデリング・レンダリングするイメージベースの研究が盛んに行われている。

イメージベース手法で3次元物体を作成する場合、レンダリングには撮影画像をテクスチャマッピングするため、物体の質感の精度は撮影

画像に依存する。そのため、使用する建築物の写真に、木や車などの障害物が写り込んでいると、生成される建築物モデルにも写り込みの影響が現れる。これは見栄えが悪くなるだけでなく、障害物によって建築物の情報が隠蔽されてしまう。

そこで本研究では、繰り返し構造を持つ建築物の特性を利用し、ユーザの簡単な操作で写り込みを修正する手法を提案する。

2. 既存研究と研究方針

イメージベース手法を用いて3次元物体の建築物を作成する研究として、建築物のエッジを

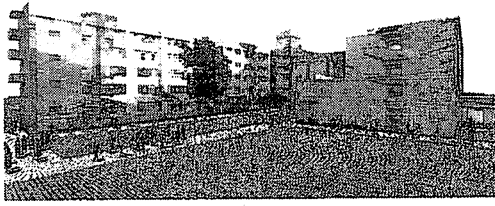


図1 写り込みが生じたレンダリング結果

指定する手法[1]や、消失点を抽出し形状を当てはめる手法[2][3]、航空写真を用いることで大規模な都市をモデリングする手法[4]がある。これらの研究は主に建築物のモデリングを簡易にすることに重点を置いており、テクスチャには目立った障害物が写り込んでいないことを前提としている。そのため、木などの障害物が大きく写り込んだ場合、レンダリング結果に影響を及ぼす(図1)。

ここで、写り込みなどによる画像の欠損部分修復の手法として山内らの研究[5]とパッチワークス[6]を挙げる。山内らの手法では周波数分解を用い、低周波成分に偏微分方程式、高周波成分に類似画像を利用し欠損部分補うことで、欠損部分をなめらかに補間する。パッチワークスでは8×8ピクセルなどのブロックに区切り、近傍の画像情報を利用し、他の修復手法に比べて高速かつ鮮やかな修正画像を自由自在に作成できる。しかし、これらの手法は自動・半自動により簡単で高速に修復が行えるという利点がある一方、木などの障害物により多くの情報が隠蔽されている画像に対しては対応できない。

建築物の構造特性を利用してモデリングを行うシステムとしてPeterらの手法[7]がある。この手法は自動処理で様々な種類の建築物が作成でき、高品質なレンダリング結果を得られる反面、実存する建築物の作成には多くの条件指定が必要となりユーザの負担が大きい。

本研究では多くの情報が欠損している画像に対し、簡単かつ効果的に写り込み問題を解決する手法について考える。

3. 提案手法

本節では、建築物の繰り返しの構造特性を利用した写り込みの修正法を述べ、予備実験とし

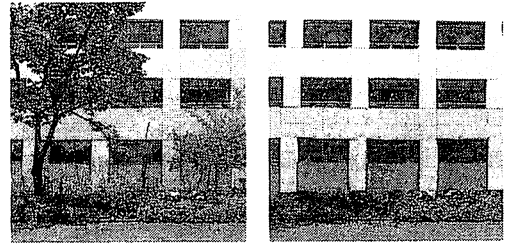


図2 建築物の構造特性による修正

て既存のソフトを用いて修正を行う。既存のソフトでは位置合わせ、拡大・縮小の設定が煩雑であり、修正に時間を要するため、操作を簡単にし、修正を簡便にするシステムを作成する。

3.1 手動によるテクスチャ修正

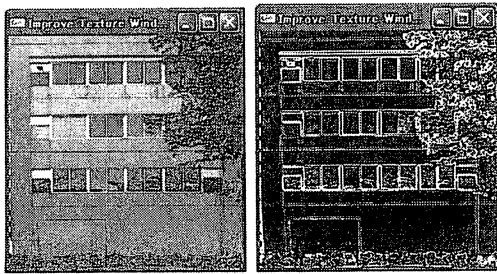
比較的大きな建築物である校舎やマンション、ビルなどの多くは、階層毎に部屋の数が決まっており、部屋の窓の構造も同一であるなど、構造に繰り返しの規則性を持つ。さらに、壁やタイルの色、窓の色などの色彩情報も単調であることが多い。この繰り返しの構造特性を利用し、木などにより情報が隠蔽されている部分に、同じ特徴が繰り返されている他の領域のテクスチャを貼り付けることで、簡単かつ効果的に広域の修正が可能になると考える。

図2に建築物の構造特性による修正を既存の画像ソフト(Microsoft Paint)*により手動で行ったものを示す。手順は、コピーするテクスチャの範囲選択、貼り付ける場所への位置合わせ、大きさの調整を複数回行うことで修正する。この処理により、建築物の構造を崩さず修正ができ、図2の結果からも違和感なく修正できていることがわかる。この修正には約十分の時間を要した。

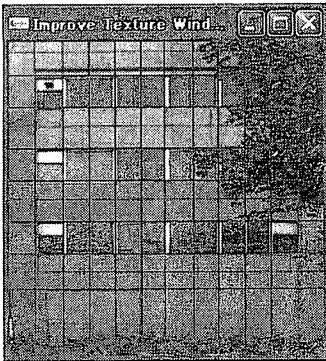
しかし、位置合わせや大きさの調整などを全てユーザが指定するため、操作が煩雑であるという問題がある。さらに、画像が歪んでしまっている場合、位置合わせが困難になる。

そこで本研究では、自動処理と対話的操作を組み合わせることで修正を簡単にし、歪みの修正を行うシステムを構築する。

* © Microsoft Corporation. All rights reserved.



(a) 原画像 (b) 輪郭線抽出後



(c) 修正ライン設定後

図3 修正ラインの抽出

3.2 画像修正システムの構築

既存の画像ソフトを用いて画像の切り貼りを行う際に最も困難な作業が位置合わせである。貼り付ける場所の指定や大きさの変更などが煩雑であり、ユーザがマウス操作で任意の場所に配置するため位置のズレも生じやすい。

そこで、ユーザの作業を軽減するため、位置合わせと大きさの指定の自動化を試みる。

3.2.1 修正ラインの抽出

今回の修正対象とする画像は長方形の窓枠やドアなど、 xy 方向に対し水平・垂直な構造を持っている。そこで、画像の輪郭線抽出を行い、水平・垂直な直線を抽出し、格子状に区切る。ユーザの操作により、障害物を含む格子内のテクスチャを障害物のない格子のテクスチャで置き換える。

輪郭線抽出にはソーベルフィルタを用いる。ソーベルフィルタとは濃淡情報を利用して輪郭線抽出を行うフィルタであり、発生するノイズを抑える効果がある。ソーベルフィルタにより得られた輪郭線情報(図 3(a) (b))を使い、濃淡

差が大きくなっている輪郭線に対し、輪郭線を xy 軸に平行に延長し、修正ラインを作成する。

以上により自動で修正ラインを抽出し、格子を作成することができる。しかし、自動抽出のみでは誤った場所にラインが設定される場合や、必要な場所にラインが存在しないなどの危険性がある。そこで、自動抽出後にユーザのマウス操作により任意の場所に修正ラインを設置・削除できるようにすることで、正確なラインの作成を可能にする(図 3(c))。

3.2.2 割り当て画像の拡大・縮小

修正ラインを使って格子内のテクスチャを置き換える際に、移動元の格子と移動先の格子の間に大きさの差が生じ、テクスチャサイズを変更する必要がある。本システムではバイリニア補間を用いて画素を補間することで自動的に拡大・縮小する。

3.3 建築物テクスチャの歪み補正

修正ラインを自動抽出する際の問題点としてテクスチャの歪みへの対応が挙げられる。通常のイメージベースレンダリングでは、テクスチャ画像の歪みがレンダリング結果に影響しない事も多い。しかし、格子を決定してテクスチャを置き換える場合、わずかな歪みでも格子とのズレが生じてしまう。

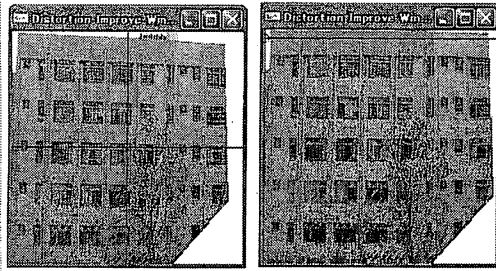
そこで、本システムでは歪みのあるテクスチャに対し、対話的な操作により補正する。

ここでは、歪みの原因を撮影時の建築物面が 3 次元的に傾いているためと仮定し、透視投影変換を利用することで歪みを補正する。

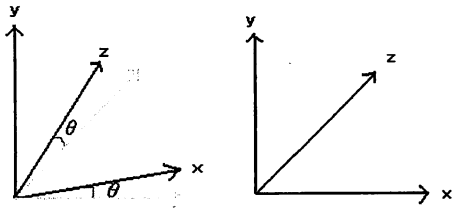
図 4(a)のように、建築物面が xyz の 3 次元座標において、図 4(c)のように y 軸を中心に座標軸が θ 度回転している場合、透視投影によって歪みが発生する。そこで、原画像に対してアフィン変換を用い、3 次元座標内で画像を回転させ、透視投影を用いて再度描画することで発生した歪みを補正する。

作成された画像は二次元平面に描画するため、 $z=1$ の平面に描画されていると仮定する。 (x,y) を補正前画像の画素の座標値とすると、補正後の画素の座標 (x',y') は(1)(2)式にて求められる。

(1)式は x 軸を中心に回転させる処理、(2)式は y 軸を中心に回転させる処理である。



(a) 歪み画像 (b) 補正画像



(c) 歪み画像の座標軸 (d) 補正画像の座標軸

図4 歪みの補正

$$\begin{cases} x' = \frac{x}{\cos\theta - y \sin\theta} \\ y' = \frac{y \cos\theta + \sin\theta}{\cos\theta - y \sin\theta} \end{cases} \dots (1)$$

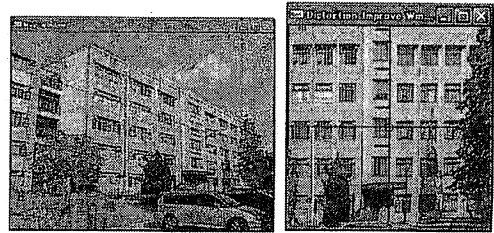
$$\begin{cases} x' = \frac{x \cos\theta - \sin\theta}{x \sin\theta + \cos\theta} \\ y' = \frac{y}{x \sin\theta + \cos\theta} \end{cases} \dots (2)$$

以上の変換式により修正を行った結果を図 4(b)に示す。以上の操作により、図 4(c)のように歪んでいた座標軸も図 4(d)に示すように正しい位置に補正することが可能となる。

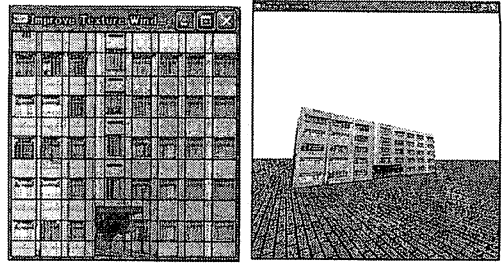
4. 提案システムにおける操作手順

我々が作成したシステムでは、モデリングからレンダリングまでをサポートしている。モデリングデータの作成は荒井らの研究[2]を用いる(図 5(a))。荒井らの研究では撮影画像を入力画像とし、画像の消失点を求め 3 次元の座標軸を検出し、直方体などの形状当てはめにより、モデリングする。作成した形状からテクスチャを切り抜きレンダリングする。

切り抜いた画像に歪みが発生している場合(図 5(b))、提案手法によりマウス操作にて角度を指



(a) メイン画面 (b) 歪み補正画面



(c) 画像修正画面 (d) レンダリング画面

図5 システム画面

定し、歪みを修正する。回転の中心となる点はユーザが任意に指定可能である。

写り込みの修正を行う場合は、修正ラインにより区切られた格子中の切り抜きたい場所をクリックし、貼り付けたい場所にドラッグすることで大きさの設定や位置合わせが可能である。また、建築物には左右の対象性を持つ場合が多いため、左右どちらかだけ隠されている場合は左右対称な画像を作成し、修正に利用できる。さらに、複数画像から修正も可能になっており、主となる画像中で修復できない写り込みが発生した場合、他の画像情報を用いた修正も可能である。

以上の処理により、歪みの補正、写り込みの修正を行い、仮想環境内に建築物のレンダリングを行う(図 5(d))。

5. 結果・考察

既存の画像ソフトを用いて障害物の写り込みを修正したものを図 6(b)に示す。提案手法により画像の修正を行った結果を図 6(c)に示す。提案手法による修正結果を用い、仮想環境内にレンダリングを行った結果を図 7・8に示す。

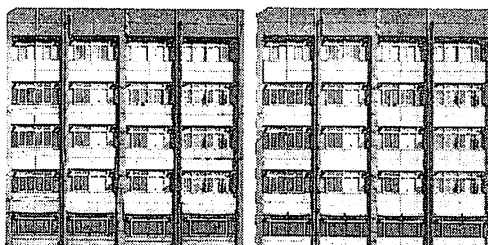
1つの建築物に対し、モデリングデータの作成に8分程度要した。モデル作成後、歪みの修正、修正ラインの設定、画像の修正を行うには、1枚の画像に対し1分強の時間で実現した。画像修正に既存の画像ソフトを用いた場合、10分程度かかったため、大幅な作業時間の短縮を実現した。

図6(b)を見ると、既存ソフトでは細かい範囲選択が可能であるため、細部まで修正が可能となっているが、操作が困難であり、位置や大きさに小さなズレが生じている。図6(c)では修正ラインを細かく設定することで細部まで修正でき、既存のソフトと変わらない品質で修正できている。

図7,8のレンダリング結果から、修正前には大きく写り込んでいた木が消え、見栄えが良くなっている。しかし、問題点として、光が当たる部分と影となる部分を持つ入力画像を用いた場合、明暗に大きな差が生じるため、図9のように色合いに不自然さが生じる。全体の明度を調整してから修正を行う処理が必要である。

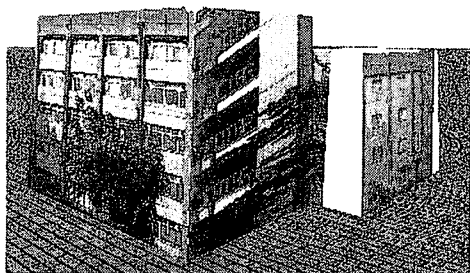


(a)修正前

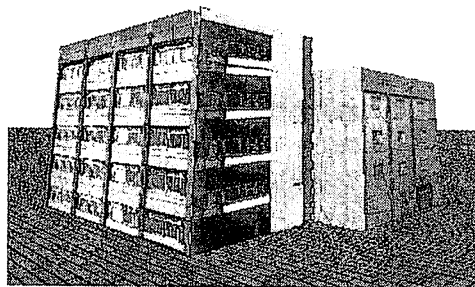


(b)既存ソフト修正 (c)提案手法修正

図6 画像の修正

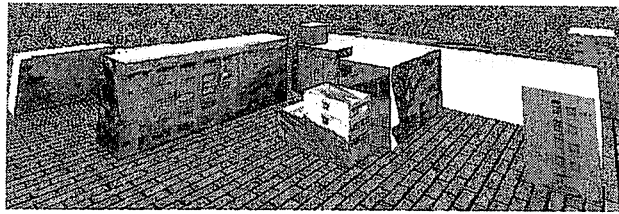


(a)修正前

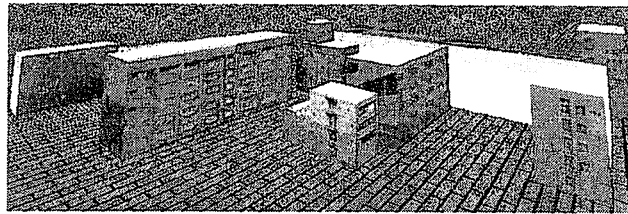


(b)修正後

図7 レンダリング結果1



(a) 修正前



(b) 修正後

図8 レンダリング結果2

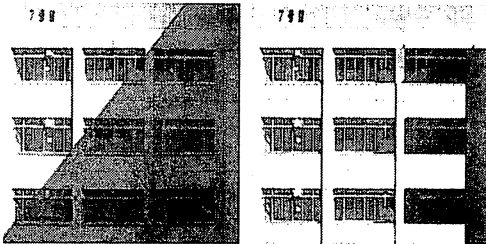


図9 色合いの問題

6. 結言

建築物の構造特性を利用し、簡単かつ広域に写り込みの修正を可能にするツールを開発した。修正の際に歪みを補正し、自動的に修正ラインを抽出するため、ユーザは格子内のテクスチャを置き換えるという、わずかな操作で修正が可能である。また、修正画像を用いてレンダリングを行った結果、修正前と比べて見栄えの良い結果を得る事ができた。

以上より、建築物の構造特性を利用するという簡単な処理のため、既存のイメージベースモデリングの研究に広く適用でき、テクスチャの精度を大幅に上げることができると考える。

今後の課題として、画角による歪みの修正、色合いの問題の対処、インタフェースの改善、有効性の検討・評価などが挙げられる。

参考文献

- [1] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik, Modeling and Rendering Architecture from Photographs: A Hybrid Geometry and Image Based Approach, *Proceedings of SIGGRAPH 96*, pp.11-20, 1996.
- [2] 荒井領太, 斎藤隆文, 複数画像に基づく簡易な建築物形状モデラ, FIT2002, LJ-6, 2002.
- [3] 松井健児, 斎藤隆文, 宮村(中村)浩子, 建築物のイメージベースモデリングにおける精度に関する考察, 情報処理学会グラフィクスとCAD研究報告, 2004-CG-117-3, 2004
- [4] Milan Ftačnik, Peter Borovsky, and Martin Samuelcik, Low Cost High Quality 3D Virtual City Models, *Proceedings of CORP 2004*, pp.305-308, 2004.
- [5] Hitoshi Yamauchi, Jörg Haber, and Hans-Peter Seidel, Image Restoration using Multiresolution Texture Synthesis and Image Inpainting, *Proceedings of CGI 2003*, pp.120-125, 2003.
- [6] Patrick Perez, Michel Gangnet, and Andrew Blake, Patch Works: Example-Based Region Tiling for Image Editing, *Technical report, MSR-TR-2004-04*. <ftp://ftp.research.microsoft.com/pub/tr/tr-2004-04.pdf>, 2004.
- [7] Peter Wonka, Michael Wimmer, Francois Sillion, and William Ribarsky, Instant Architecture, *Proceedings of SIGGRAPH 2003*, pp.669-677, 2003.