

隣接するビルの影響を考慮したビル正面テクスチャ合成手法

Facade Texture Composition Considering Neighboring Buildings

岸 麻美†
Asami Kishi†

中津川 直輝‡
Naoki Nakatsugawa‡

森谷 友昭†
Tomoaki Moriya†

高橋 時市郎†
Tokiichiro Takahashi†

1. まえがき

都市景観を3DCGで再現するために、再現対象のビルを撮影し、実写画像をモデリングしたビルの表面にテクスチャとしてマッピングする手法が多く用いられる。しかし、撮影した実写画像には障害物や隣接するビルの影響で生じた陰影が写り込んでいるためテクスチャとしての質を損なう。そのため、陰影や障害物の写り込みが少ない良質なテクスチャを生成する必要がある。

本研究では、再現対象のビルの全景画像と1階部分の低層部画像を合成することによって、街路樹などの障害物の写り込みが少ないテクスチャ画像生成を目指す。

2. 従来の合成の問題点

ビルが林立する都市部では、隣接するビルが天空光を遮蔽するため、再現対象となるビルの外壁面の明度が高度によって異なる(図3(a))。3DCG空間内で照明を設定する際、テクスチャに明度の違いが残っていると不自然さを生じるため、全景画像から隣接するビルの影響で生じた陰を取り除き、明度を均一化する必要がある。

また、全景画像は再現対象のビルから離れた地点で撮影されるため、全景画像の低層部部分には街路樹や植込み、道路を走っている車両などの障害物が写り込む。それらの障害物を除去するためには、ビルに近づき、歩道からビルの低層部を撮影した低層部画像を全景画像と合成する必要がある。

従来の画像の合成作業には、Photoshopなどの画像編集処理ソフトを利用している。しかし、手作業での障害物除去作業は煩雑である。全景画像と低層部画像を合成するために部品ごとに細かく色の修正を行わなくてはならないため、時間と労力がかかるという問題がある。

3. 提案手法

前節で述べた問題点を解決するため、全景画像と低層部画像の陰と色調を自動調整する手法を提案する。処理の流れを図1に示す。なお障害物を除去するには、障害物の写っているビルの部品画像を置き換える手法[1]や、ビルの低層部を撮影した動画像に長時間露光技術を適用して歩行者等の障害物を除去する技術[2]等がある。これらの手法は有効であるが、本稿では紙面の都合で割愛する。

まず隣接するビルが天空光を遮蔽することによる影響を考慮し、全景画像の明度を調整する。さらにヒストグラム伸張により、明るくなりすぎた全景画像の低層部の明度を再補正する。

次に全景画像と低層部画像の色調補正を自動で行い、

† 東京電機大学

† Tokyo Denki University

‡ 東京電機大学大学院

‡ Tokyo Denki University Graduate School

低層部画像のサイズを調整して全景画像と合成し、1枚のテクスチャ画像を生成する。

3.1 全景画像の明度補正

図2のグラフに示すように、隣接するビルによる遮蔽の影響により、ビルの低層部ほど、全景画像の任意の画素の明度 $v(x, y)$ の値は小さくなる。また、図2より、明度値の変化は線形近似と見なすことができるので、全景画像の壁面の明度が均一になるように、式(1)で補正する。

$$v_{(x,y)} = v_{(x,y)} + \frac{\sum_{i=0}^{width} v(i,0) - \sum_{i=0}^{width} v(i,height-1)}{width \times (height - y)} \quad (1)$$

ここで、 $width$, $height$ は全景画像の幅と高さである。

また、ビルの低層階は隣接するビルだけでなく街路樹等の障害物にも天空光がさえぎられているため、明度値が一層小さくなる傾向にある。そのため、式(1)の方法で明度値を補正した場合、低層部が明るくなりすぎてしまう。そこで、ビルの全景画像を低層部と高層部に分け、式(2)を用いて、それぞれの明度のヒストグラムを伸張し、色調を調整する。ここで、 v_{min} は明度の最小値である。

$$v_{(x,y)} = (v_{(x,y)} - v_{min}) \times \frac{255}{255 - v_{min}} \quad (2)$$

3.2 全景画像と低層部画像の合成

全景画像と低層部画像では色調が異なるため、そのまま合成すると違和感がでてしまう。そこで、両者の色調を自動補正した後、全景画像と低層部画像とを合成する。

色相などのばらつきがあるため、低層部画像の色調を一様に補正することは難しい。そこで、各画素の色相、彩度、明度の値により、全画素を6色域に分類しラベリングする。次に、ラベルごとに全景画像の低層部と低層部画像の色調を合わせる。具体的には、まず、全景画像と低層部画像の色を合わせたい部分の画素を選択する。次に、その画素の近傍のRGB値の平均値を求め、その差分を各画素のRGB値に足し合わせることで色調を補正する。

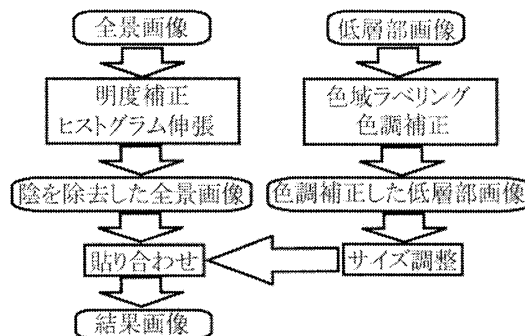


図1. 提案手法

その後、画像編集処理ソフトを使って色調補正済みの低層部画像を全景画像に貼り合わせる。

4. 実験結果

図3にビル全景画像の原画像と前章で述べた補正処理後の画像を、図2に全景画像の明度分布を示す。図2から、ビルの低層階で低くなっていた明度値が、補正処理によってほぼ一定の値に補正できたことがわかる。図3(a)で暗く、同じ高さでも微妙に明るさが異なっていたビルの低層階は、図3(b)に示すように、壁面の色が均一に見えるように補正されていることがわかる。

次に、明度補正を行った全景画像と低層部画像の色調補正を行った。まずは低層部画像の全画素を赤、緑、青、白、灰、黒の6色域にラベリングする。彩度sが閾値より高い画素は色相を $-60\sim 60^\circ$ 、 $60\sim 180^\circ$ 、 $180\sim 300^\circ$ に分類し、それぞれ赤、緑、青とする。彩度sが閾値より低い画素は無彩色であるとし、その明度を白、灰色、黒の3段階に分類する。低層部画像の全画素を色域分類した結果を図4に、ラベリングした6つの色域ごとに色の補正を行った結果を図5に示す。図5(b)と図5(c)を比較すると、色調補正をすることによって全景画像と低層部画像の色の境界が目立たなくなっている。また、図5(a)の原画像には障害物が大量に写り込んでいるが、障害物の写り込んでいない低層部画像を合成することによって、より良質テクスチャ画像を生成することができた。

5. むすび

都市景観再現の効率化のため、ビルの全景を撮影した画像とビルの低層部を撮影した画像を用いて、3次元ビルモデルの側壁にマッピングするテクスチャの生成手法を提案した。歩行者や街路樹等の障害物を自動・半自動で除去した後、全景画像と低層部画像の色調を自動的に補正して合成する手法を提案し、良好な合成結果を得た。

参考文献

- [1] 中津川 他: “都市景観シミュレーションのための建築物テクスチャの自動修正”, 信学総大 2008, D-11-98.
- [2] 岸 他: “都市景観再現のための建築物低層部のテクスチャ生成”, 信学総大 2010, D-11-83.

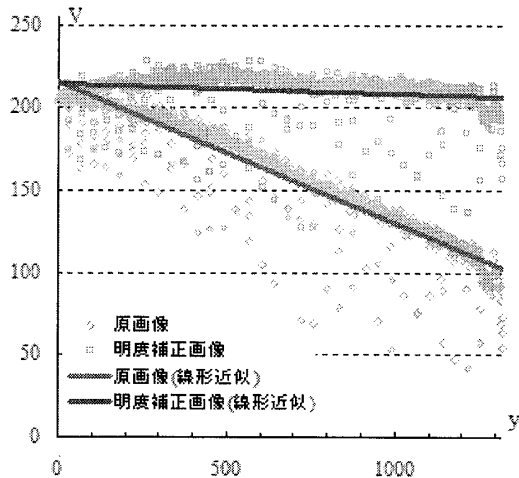
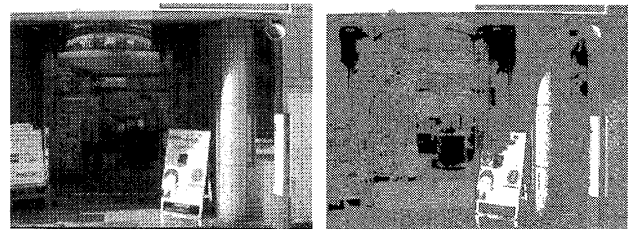


図2. 全景画像の明度値のグラフ



図3. 全景画像の明度を補正した結果



(a) 原画像 (b) 領域分割画像
図4. 低層部画像を6色に領域分割した画像



図5. 全景画像と低層部画像の合成結果