

# 縮小画像を用いたポアソン画像合成の高速化

## A Fast Poisson Image Synthesis Algorithm using Image Subsampling

長友 陽介†  
Yosuke Nagatomo

八島 由幸†  
Yoshiyuki Yashima

### 1. 研究の目的と背景

ポアソン画像合成 [1][2]は切り取り部分と貼り付け先との色合いを調整し、違和感のない自然な画像合成を実現する技術であり、種々の画像合成アプリケーションのほか最近では超高压縮画像符号化への応用も検討されている[3]. しかし、合成処理の過程でポアソン方程式を数値解析的に解く必要があるために、膨大な回数の繰り返し処理が必要となり高速化が求められている. 演算量を削減するために、これまで、マルチコアによる並列化処理を工夫する手法[4]や、Mean value coordinatesを用いて近似解を求めることで高速化する手法[5]などが提案されている.

今回、縮小画像を利用して高速化を図る手法を試みた. 本手法では、画像に縮小処理を加えてから合成することで繰り返し処理に掛かる時間を短縮するとともに、合成済みの縮小画像を拡大する際に、合成前の貼り付け画像における隣接画素間の関係を利用して補間を行うことで高周波成分を復元し、ぼけのない合成画像を得る.

### 2. ポアソン画像合成の概要

図1にポアソン画像合成の原理を示す. ポアソン画像合成は、色味を貼り付け先画像のものに似せ、テクスチャ(画像の勾配)を貼り付け元画像に似せるような処理と考えてよい. すなわち、 $f$ を貼り付け先画像、 $g$ を貼り付け元画像(切り出し画像)、 $p$ を画素の座標、 $qi$ を $p$ に隣接する上下左右の4つの画素( $i=1,2,3,4$ )の座標とすると、貼り付け境界も含め、

$$\sum_{i=1}^4 (f_p - f_{qi}) = \sum_{i=1}^4 (g_p - g_{qi}) \quad (1)$$

が成り立つように $f_p$ を決める.  $g_p \cdot g_{qi}$ は切り出し領域の画素勾配に相当する. 式(1)を変形すると、

$$f_p = \left( \sum_{i=1}^4 f_{qi} + \sum_{i=1}^4 (g_p - g_{qi}) \right) / 4 \quad (2)$$

となり、(2)式は切り取り領域の画素数分だけできるので、この連立方程式をガウス・ザイデル法やSOR法などで解くことで各 $f_p$ を求める.

### 3. 縮小画像を用いた高速合成

#### 3.1 処理の概要

式(2)は画素数を未知数とした連立方程式となり、切り出し領域の画素数が大きいと解が収束するまでに要する繰り返し演算回数が膨大になる. そこで今回、縮小画像を用いてポアソン画像合成を高速化する手法を提案する. 図2に処理手順を示す. まず、切り取り画像Aと貼り付け先画像Bそれぞれ縦横 $1/n$ に縮小し、縮小画像 $A_s$ と $B_s$ を得る. ここで、縮小する際には折り返し除去などのフィルタ処理

は施さず単純間引きを行う. 次に、縮小した画像同士で2章に述べたのと同様の手法によりポアソン画像合成処理を行い、 $A_s$ を $B_s$ に合成できるよう輝度/色差を補正して $A'_s$ を縮小率に応じて $n$ 倍拡大して $A'$ とした後、貼り付け先画像Bの該当する領域に貼り付ける. この手法によってポアソン画像合成処理に伴う繰り返し演算が減るため、処理時間を大幅に削減することが可能になる.

#### 3.2 隣接画素関係を用いた拡大補間

縮小された合成画像を拡大する際にバイリニア補間やバイキュービック補間等の手法を適用すると、切り出し領域が元来持っていた高周波成分を表現することはできず、ぼけが発生する. そこで今回、補間の際にももとの切り出し領域A(合成前の原サイズ画像)を参照し、その隣接画素間の関係を利用して $A'_s$ を補完する手法を採用する. ポアソン画像合成に使われる画素と使われない画素を相補的サブサンプリングの関係になるようにすれば、それらを足し合わせることでもとの画像の持つ高周波成分を復元できる. このため、画像を縮小する際にはローパスフィルタを施さずに折り返し成分が含まれたまま単純にサブサンプリングする.

補間拡大方法を図3に示す. 図3は縮小率 $1/4$ の場合の例であり、説明を簡単にするため水平方向のみを示している. 縮小画像でポアソン方程式を解いて得られた基準画素(図3の▲画素)を拡大画像の対応する座標に配置する. 基準画素 $x,y$ の画素値を $f(x), f(y)$ とすると、求めるべき補間画素はその間にある3つの座標位置 $a,b,c$ における画素値 $f(a), f(b), f(c)$ である. まず、基準画素の中間に存在する $f(b)$ を次式を解いて求める.

$$\begin{aligned} \alpha &= \tan^{-1}((g(b) - g(x))/2) - \tan^{-1}((g(y) - g(x))/4) \\ &= \tan^{-1}((f(b) - f(x))/2) - \tan^{-1}((f(y) - f(x))/4) \quad (3) \end{aligned}$$

ここで $g(\cdot)$ は切り出し画像ももとの持つ画素値である.

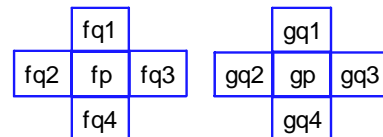


図1.ポアソン画像合成の原理

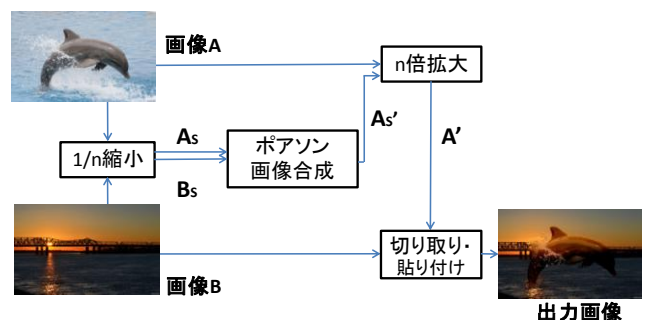


図2.提案手法

† 千葉工業大学大学院 情報科学研究科

この処理は言い換えれば図3において  $x, b$  間の勾配と  $x, y$  間の勾配の差 (角度  $\alpha$ ) を合成前と同じにすることを意味する。同様の処理を再帰的に繰り返し、 $a$  の画素値を  $x$  と  $b$  から、 $c$  の画素値を  $b$  と  $y$  から求める。2次元的に行うためにはまず水平方向に本手法を適用し、補間された画素を用いて垂直方向に同様の補間をすればよい。この手法により、切り出し領域が本来保有していた高周波成分を再現することができる。

4. 実験と考察

提案手法の有効性を確認するため、以下の3つの手法を実験し合成後の画像の画質と処理時間を比較した。

- (1)従来のポアソン画像合成
- (2)縮小画像合成+バイキュービック補間
- (3)縮小画像合成+原画素参照 (提案手法)

今回の実験では切り出し領域の指定はオフラインでPhotoshopを用いて行った。画像1, 画像2, 画像3における切り出し領域サイズ  $N$  (画素数) はそれぞれ 500904, 682460, 48560 である。処理時間の測定においては、ポアソン方程式を解く反復演算において、 $k$  回目に得られた合成画像  $f_p(k)$  に対して  $k-1$  回目に得られた合成画像  $f_p(k-1)$  との差が、

$$\frac{1}{N} \sum_p (f_p(k) - f_p(k-1)) < \epsilon \quad (4)$$

を満たす場合に処理を終了した。 $\epsilon$  は各画像の反復過程において画質が変化しない十分な値を設定し、各  $\epsilon$  の値を  $3 \times 10^{-5}$  と設定した。

縮小率  $n$  を 2, 4, 8, 16 とし測定を行った結果を図4に示す。処理時間は縮小率の増加とともに急激に減少し、1/2 縮小で 3~4%, 1/4 縮小で 0.1~0.01%, 1/8 縮小で 0.04~0.01%, 1/16 縮小で 0.03~0.01% で処理が終了する。また、図5には縮小率2の場合について実際の合成画像の例を示す。バイキュービック補間では明らかにぼけてしまうのに対し、提案手法では色見を自然に保ちつつ高周波エッジを良好に再現できており、間引きなしの通常の合成手法に比べても視覚上大きな劣化は認められなかった。しかし、縮小率を4以上にすると若干ブロック状のノイズが発生することがあった。その理由として今回は基準画素を固定しているため、基準画素の左側と右側における連続性が欠如していたためと考えられる。

5. まとめ

今回、縮小画像を用いたポアソン画像合成と画素の隣接関係を用いる補間手法により、大幅な高速化を図ることができた。今後の展開として、本方式の画像符号化への適用や、動画像への拡張が挙げられる。なお、本研究は JSPS 科研費 25330204 の助成を受けて実施したものです。

[参考文献]

[1]P.Perez etc., "Poisson Image Editing," ACM SIGGRAPH 2003, pp.313-318, 2003.  
 [2]宮岡伸一郎, "画像の勾配空間フィルタリング," 情報処理学会研究報告, 2009.  
 [3] 吉野貴士, 八島由幸, "顕著性マップとポアソンレンディングを用いたテクスチャ合成画像符号化," 電子情報通信学会画像工学研究会, IE-2012-119, pp.227-232, 2012.  
 [4]渡辺賢悟, 伊藤和弥, 近藤邦雄, 宮岡伸一郎, "Poisson Image Editingを用いたキャラクターカラーシステムの開発," 芸術学

会論文誌, Vol.9, No.2, pp.58-65, 2010.

[5]Zeev Farbman, Gil Hoffer, Yaron Lipman, Daniel Cohen-Or and Dani Lischinski: "Coordinates for Instant Image Cloning", ACM SIGGRAPH 2009, 2009.

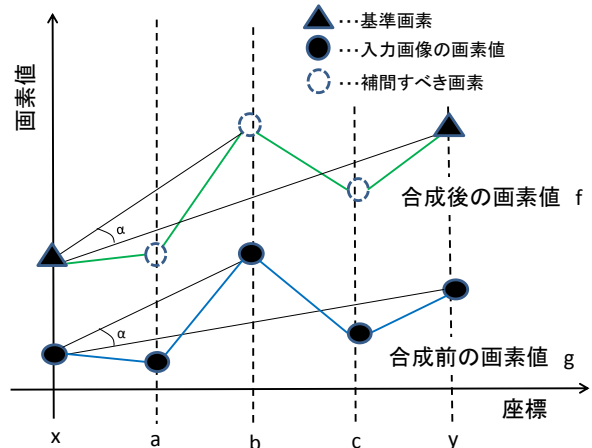


図3. 補間処理 (縮小率4の場合の例)

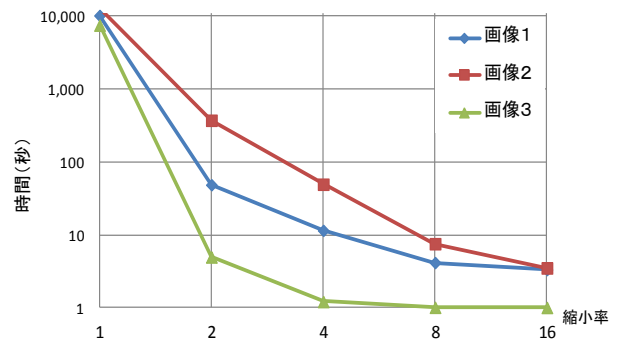


図4. 処理時間の比較

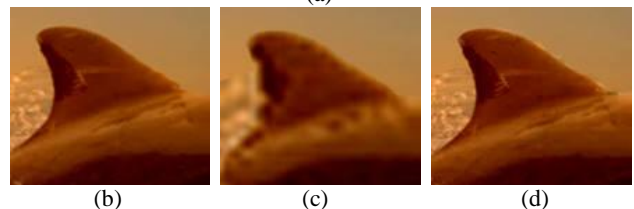


図5. 画質比較 ((a)提案手法による合成画像, (b)従来のポアソン画像合成, (c)縮小画像合成+バイキュービック補間, (d)縮小画像合成+原画素参照補間 (提案手法) )