

コンピュータ理工学特別研究報告書

題目: 勾配情報を利用したポアソン画像合成の性質と応用

学生証番号 947604

氏名 吉田 祐樹

提出日 平成 25 年 2 月 4 日

指導教員 蚊野浩 教授

京都産業大学

コンピュータ理工学部

要約

本研究は、画像の一部を切り出し、他の画像に貼付け合成する問題を扱う。このとき、切り出した領域を単純に貼付ける方法では、境界部分が目立ち不自然な画像になることがある。それに対して、ポアソン画像合成は境界部分が目立たない、自然な合成が可能である。この手法は、貼付け元画像の画素値の勾配と貼付け先画像の境界条件を利用する。本研究では、いくつかのサンプル画像にポアソン画像合成を適用することでその効果を確認した。ついで、ポアソン画像合成をより深く理解するために、簡易な実装と、境界部分がより目立ちにくくなる処理を行う詳細な実装の比較を行った。その結果、自然画像の合成では簡易な実装でも十分であると判断した。さらに、ポアソン画像合成と単純な画像合成の利用画面をさまざまに検討した。そして、これらの長所を生かす手法として、両者で計算される画素値を加重平均することが有効であることを見いだした。この提案手法は、境界部分が目立ちにくく、かつ背景の色味の影響を受けすぎない。本論文では、その効果を実画像によって実証した。

目次

1. はじめに	・・・ 2
2. 2画像の合成技術	・・・ 3
2. 1 単純画像合成	・・・ 3
2. 2 勾配情報を利用したポアソン画像合成	・・・ 4
3. ポアソン画像合成の原理と適用	・・・ 5
3. 1 ポアソン方程式と画像合成	・・・ 5
3. 2 ガウス・ザイデル法による連立一次方程式の解法	・・・ 7
3. 3 ポアソン画像合成の適用例	・・・ 9
3. 3. 1 中間顔画像の生成	・・・ 9
3. 3. 2 傷がついた画像の修復	・・・ 10
4. 簡易な実装と詳細な実装の比較	・・・ 12
4. 1 簡易な実装と詳細な実装の適用例と比較	・・・ 12
5. ポアソン画像合成と単純画像合成の加重平均による画像合成	・・・ 14
5. 1 ポアソン画像合成と単純画像合成の比較	・・・ 14
5. 2 改良ポアソン画像合成	・・・ 15
5. 3 実験例	・・・ 16
6. おわりに	・・・ 18

謝辞

参考文献

付録：開発したプログラムのドキュメント

1. はじめに

画像の編集・合成とは、既存の画像に処理を施し新たな画像を生成することである。その代表的な例として2枚の画像から1枚の画像を生成するものがある。その従来手法の一つは、第一の画像から貼付けたい部分をクロマキーや切り抜きツールで抽出し、それを第二の画像の適当な部分に貼付けるものである。この従来手法による合成では、両画像を取得したときの照明条件が大きく異なる場合、合成された画像が不自然になることがある。それを避けるために、なるべく単一の背景色で、かつその色が前景（切り出したい対象）のなかに現れないように、あらかじめ撮影しておくなどの工夫がある[6]。

従来手法における、これらの問題点を解決するために、様々な画像編集技術が考案され、それを利用したソフトウェアが開発されている。例えば、顔画像合成を行うFace Fit[1]や画像内の不要物を除去するフォレストタッチ[2]といったものがある。

このようなソフトウェアの元となった画像編集・合成技術の一つとして、画像の勾配と境界条件に着目したポアソン画像合成と呼ばれる手法がある[5]。ポアソン画像合成の特徴は、貼付け元画像と貼付け先画像の境界が目立たないことである。この特徴によって、多くの場合に自然な画像合成が可能になる。

本研究では、ポアソン画像合成を行うプログラムを開発し、サンプル画像に適用することでその効果を検証した。また、ポアソン画像合成の機能と効果をより深く理解するために、簡易な実装と境界がより目立ちにくくなる処理を行う詳細な実装の比較を行った。さらに、ペイントツールなどに実装されている単純な画像合成との用途の違いを明らかにし、両手法の長所を生かした画像合成法を考案した。

2. 2 画像の合成技術

2章では、最もよく利用される画像合成技術である貼付け元画像の画素値を単純に貼付ける方法と、本研究の素材であるポアソン画像合成の違いを例示する。

2.1 単純画像合成

図1の顔画像Aの目鼻口部分を、顔画像Bの目鼻口部分に合成する場合を例として、ポアソン画像合成と単純画像合成を比較する。このとき、2画像の目鼻口部分を指定するために、別途、マスク画像Cを用意する。

顔画像Aのマスク画像Cで指定される領域を顔画像Bと合成する方法の中で、最もよく用いられる方法は、Aの画素をBの画素に置き換える方法である。その結果、図1の結果画像を得る。この方法を「単純画像合成法」と呼ぶことにする。単純画像合成法は、顔全体や人物像全体など、物体の全体を背景に貼り付ける場合には向いているが、この例のように、目鼻口などの物体内部の一部を合成するような場合には向いていないと思われる。結果画像では、合成した境界部分が目立っておりうまく画像合成が行えていないことがわかる。

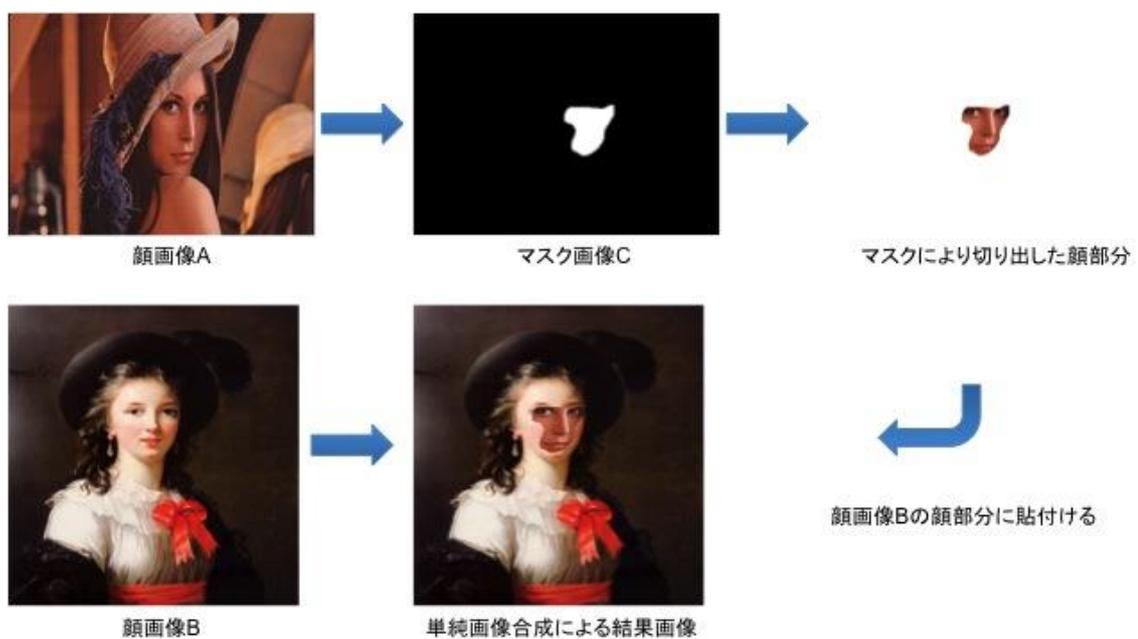


図1. 単純画像合成の処理の流れ

2. 2 勾配情報を利用したポアソン画像合成

ポアソン画像合成は、画像中の合成・修復したい領域の画像値をポアソン方程式から求める手法である。このとき、領域内の画素値の関係として、貼付け元画像の画素値の勾配を用いることで画像合成が行われる。ポアソン画像合成の処理の流れを図2に示す。結果画像において、合成の境界部分がなめらかに接続していることがわかる。

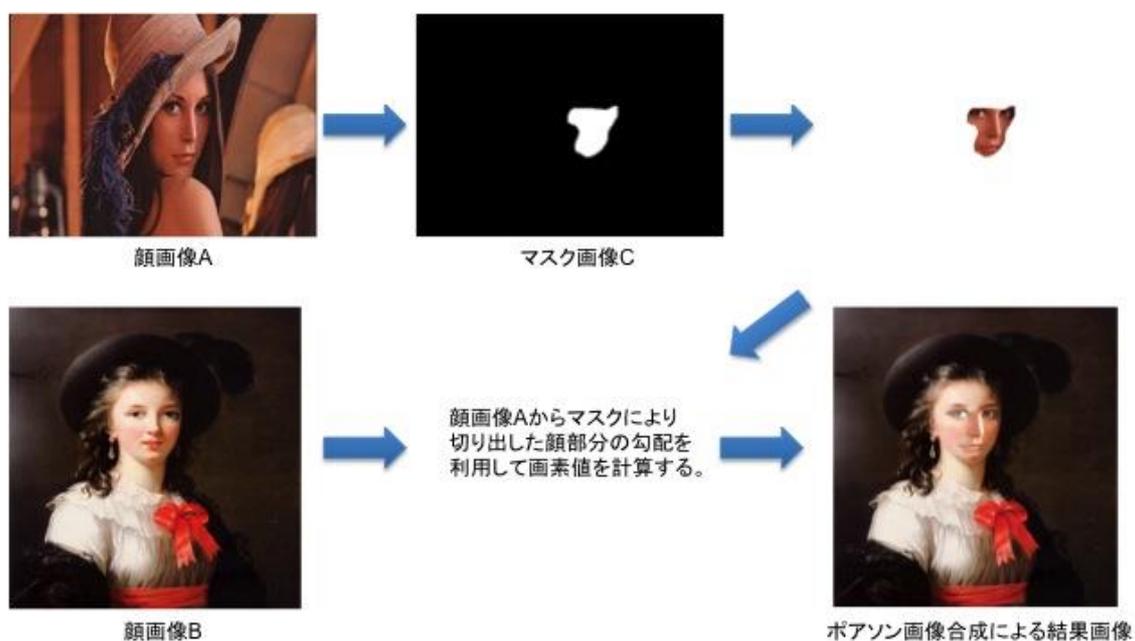


図2. ポアソン画像合成の処理の流れ

3. ポアソン画像合成の原理とその適用

ポアソン画像合成の原理、実装法、適用例について述べる。

3. 1 ポアソン方程式と画像合成

ポアソン方程式は2階の偏微分方程式である。ポアソン方程式で記述される物理現象として、電磁気学における静電ポテンシャルがある。そこでは、内部電位の状態を記述する2階の偏微分方程式を、境界部分の電位を条件として解くことで、電位分布を求めることができる[4]。図3のように、赤い点の集まりで示した境界部分の電位が既知で、黒い点の集まりで示した内部電位の関係がポアソン方程式で記述されれば、領域内部の電位の値を計算することができる、ということである。

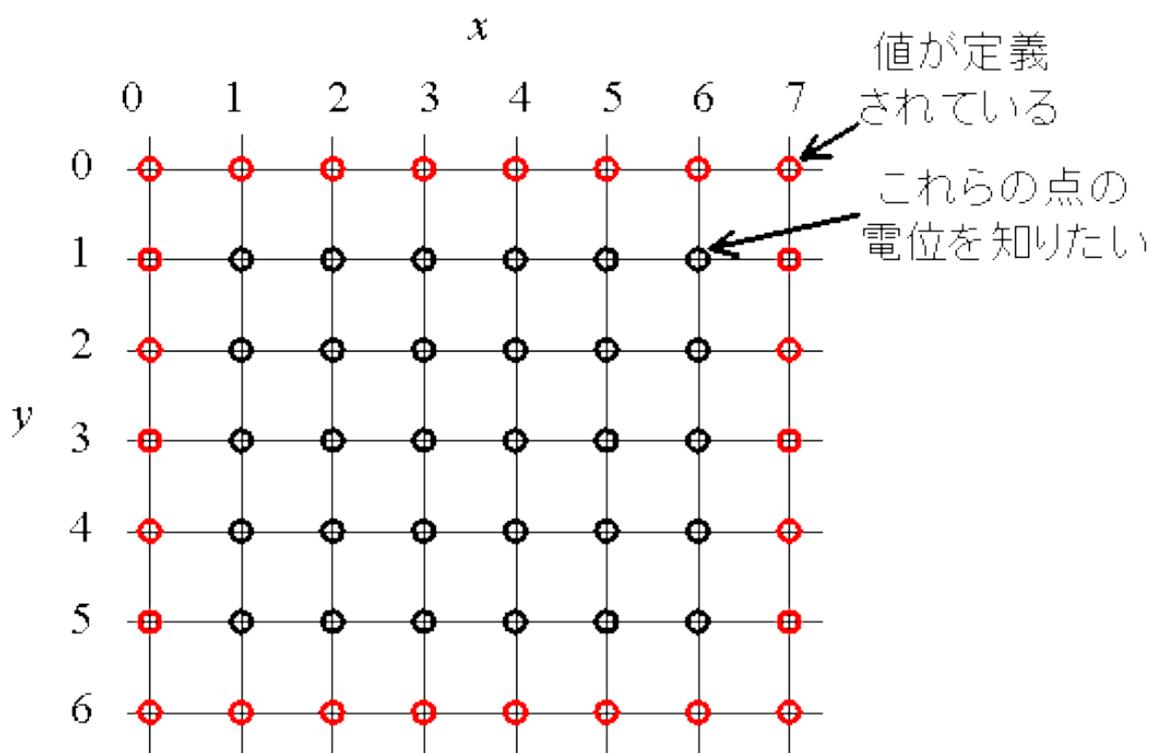


図3. 境界条件を定めるセルと内部電位を計算するセル[4]

本研究で扱うポアソン画像合成は、領域内部の画素値の関係をポアソン方程式で記述し、領域の境界部分の画素値を条件として、領域内部の画素値を求めるものである。そして、領域内の関係に貼付け元画像の勾配を用いることによ

り画像合成を行う。図4にポアソン画像合成の概念を示し、定義式を式(1)に記述する。

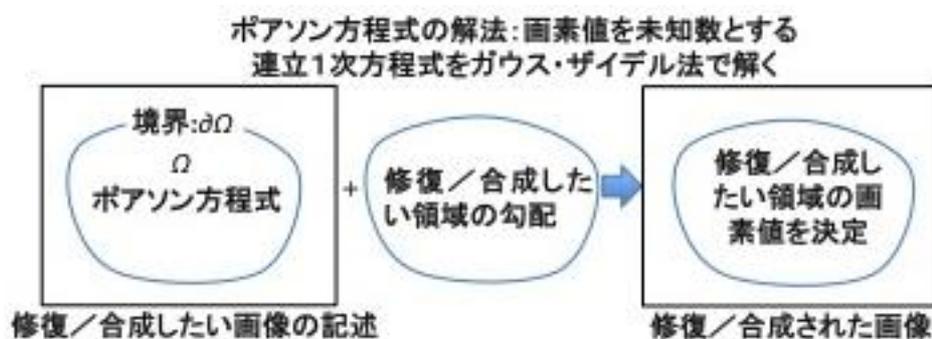


図4. ポアソン画像合成の概念

修復/合成したい領域に、切り取り済みの貼り付け元画像の勾配情報を与え、それをポアソン方程式で記述する。また、貼付け先画像の領域境界における画素値を境界条件として与える。なお、画像合成の具体例として画像の一部を修復する場合があるので、「修復/合成」という言葉を用いた。ポアソン方程式の離散領域における解法は、連立一次方程式に帰着される。それをガウス・ザイデルで解くことで、修復/合成したい領域の画素値を得ることができる。この手法の全体をポアソン画像合成とよぶ。

$$\frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y) = f(x,y) \cdots (1)$$

式(1)がポアソン方程式である。式(1)左辺の $\frac{\partial^2 u}{\partial x^2}(x,y)$ は水平方向の2次偏微分を表す。 $\frac{\partial^2 u}{\partial y^2}(x,y)$ は垂直方向の2次偏微分を表す。左辺全体はこの2項の和になっている。この形式の式をラプラシアンと呼ぶ。右辺は任意の関数であり、左辺がある関数になることを示す。

式(1)が画像合成とどのように関係しているかを説明する。式(1)左辺の $u(x,y)$ は貼付け先画像である。左辺全体は注目画素のラプラシアンを計算する。右辺は、一般には任意の関数であるが、勾配情報を用いるポアソン画像合成では、貼付け元画像の勾配を用いる。

ポアソン方程式を画像合成に応用する時、簡易な実装と境界部分がより目立ちにくくなる処理を行う詳細な実装がある[3]。本研究では、主に詳細な実装を適用する。ただし、実装法について、詳細な実装の式で説明すると難解であるため、まず、簡易な実装の式を説明する。4章で詳細な実装を説明し、簡易な実装との比較を行う。

式(2)は、文献[3][5]に記載されているポアソン画像合成の簡易な実装を表現する式である。

$$|N_p|f_p - \sum_{q \in N_p} f_q = \sum_{q \in N_p} v_{pq} (v_{pq} = g_p - g_q) \dots (2)$$

式(2)において、 f は求めたい結果画像、 g はマスク画像で切り取った貼り付け元画像である。 p は注目画素の位置、 N_p は p の上下左右の隣接画素を表す。 N_p に絶対値記号を付けたものは隣接画素の画素数を表す。 q は N_p に含まれる隣接画素のどれか一つを表す。 f_p 、 f_q は f における p と q の画素値を表す。 v_{pq} は貼り付け元画像における注目画素と隣接画素の画素値の差、すなわち勾配を表す。式

(2)の左辺は画像処理におけるラプラシアンフィルタ(2次微分フィルタ)になっている。右辺は切り取り済み貼り付け元画像の勾配を示し、先ほど説明したように式(1)の右辺に置き換える。

式(2)の左辺は求めたい結果画像の注目画素のラプラシアンを表し、右辺は切り取り済みの貼り付け元画像の注目画素の勾配を表している。つまり、貼り付け先画像のラプラシアンを貼り付け元画像の勾配と同じにする計算を行っていることになる。その結果、 f_p (求めたい結果画像の注目画素値)を求めることができる。領域全体では、式(2)が画素数の数だけ存在する連立一次方程式になる。多次元連立一次方程式の計算にはガウス・ザイデル法を用いる。次の節でこの計算手順について述べる。

3. 2 ガウス・ザイデル法による連立一次方程式の解法

式(2)のように実装されたポアソン方程式は、画素値を未知数とする連立一次方程式になる。式(3)に連立一次方程式の一つの式を示す。

$$4 \times f_p - (f_{q1} + f_{q2} + f_{q3} + f_{q4}) = c_p \dots (3)$$

式(3)において、 f_p の係数4は上下左右の隣接ピクセルの総数を示している。未知数は $f_p, f_{q_1}, \dots, f_{q_4}$ の計5つで、 c_p は定数である。

ここで、例えば画素がM個あったとする。すると、式(2)がM個あることになる。行列を使って、式(3)を簡潔に記述すると式(4)になる。

$$\mathbf{Ax} = \mathbf{c} \dots (4)$$

式(4)のAは係数行列、 \mathbf{x} は未知数のベクトル、 \mathbf{c} は定数のベクトルを示している。ここで、未知数 \mathbf{x} と式の数が同じであり、式(4)を連立方程式として解くことができる。

次に、未知数 \mathbf{x} を求める式に変形すると式(5)になる。

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{c} \dots (5)$$

式(5)において、係数行列Aの逆行列と定数ベクトル \mathbf{c} を掛け合わせれば未知数のベクトル \mathbf{x} を求めることができる。しかし変数の数が莫大になると、係数行列Aの逆行列から \mathbf{x} を求める方法は効率がよくない。この場合、ガウス・ザイデル法と呼ばれる反復法を用いて連立一次方程式を解く。以下にガウス・ザイデル法のアルゴリズムを記述する。

まず、n次正方行列Aの上三角行列をU、下三角行列をL、対角行列をDとする。すると $\mathbf{A}=\mathbf{U}+\mathbf{L}+\mathbf{D}$ と書くことができる。そして、これを式(4)のAに代入し、順に式変形すると以下のような式(6)になる。

$$\begin{aligned}(\mathbf{U} + \mathbf{L} + \mathbf{D})\mathbf{x} &= \mathbf{c} \\ \mathbf{D}\mathbf{x} &= \mathbf{c} - (\mathbf{L} + \mathbf{U})\mathbf{x} \\ \mathbf{x} &= \mathbf{D}^{-1}\{\mathbf{c} - (\mathbf{L} + \mathbf{U})\mathbf{x}\} \dots (6)\end{aligned}$$

ここで未知数 \mathbf{x} の初期値を \mathbf{x}_0 とし、k回の繰り返し計算で得られた \mathbf{x} の値を $\mathbf{x}^{(k)}$ とすると、以下の反復法の漸化式が得られる。

$$\mathbf{x}^{(k+1)} = \mathbf{D}^{-1}(\mathbf{c} - \mathbf{L}\mathbf{x}^{(k+1)} - \mathbf{U}\mathbf{x}^{(k)}) \dots (7)$$

式 (7) において解が収束した時、 $\mathbf{x}^{(k+1)}$ と $\mathbf{x}^{(k)}$ は同じ値になる。それは、式 (6) の連立方程式の形に戻ると、未知数 \mathbf{x} の解になっている。

次に詳細な実装の式 (8) について説明する。式 (8) も式 (2) と同様に文献[3][5]に記載されているものである。

$$|N_p|f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^* + \sum_{q \in N_p} v_{pq} \cdots (8)$$

式 (8) において、記号 Ω 、 f^* 、 $\partial\Omega$ 以外は式 (2) で説明したとおりである。 Ω は貼付け先の領域を示しており、 $\partial\Omega$ はその領域の境界部分を示している。 f^* は貼り付け先の境界部分の元々の画素値を示している。 f^* は処理を施す境界部分において、貼り付け先の元の画素値を残すことを意味する。その影響を領域内部に反映させ、境界を目立ちにくくする役割を果たしている。境界条件はマスク画像から、境界部分を判定することにより定めている。

よって、式 (2) を改良した式 (8) をガウス・ザイデル法で解くことにより詳細な実装による画像合成を行うことができる。また、簡易な実装は同様に式 (2) をガウス・ザイデル法で解くことにより実現できる。

3. 3 ポアソン画像合成の適用例

ポアソン画像合成の適用例を2つ示す。1つは中間顔画像の生成、もう1つは傷がついた画像の修復である。

3. 3. 1 中間顔画像の生成

ポアソン画像合成を用いて、2画像から中間画像を生成した例を図5に示す。

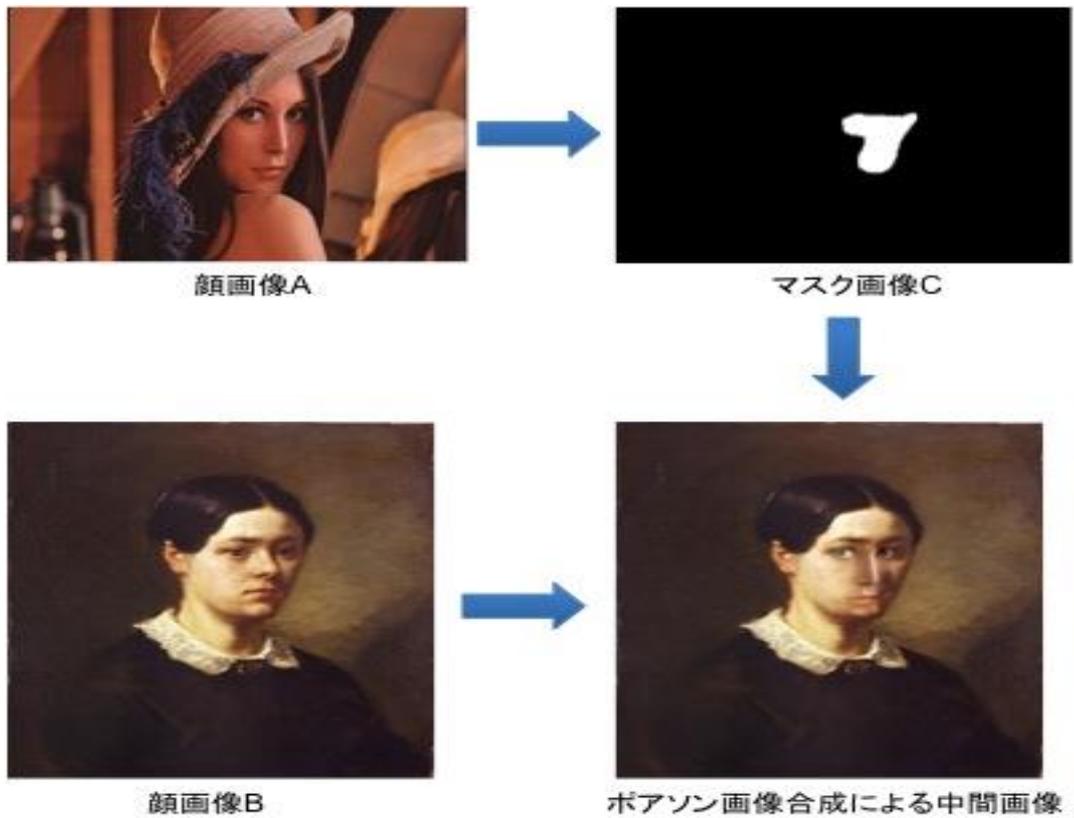


図5. ポアソン画像合成による中間画像の生成

ポアソン画像合成で、顔画像 A の顔部分の勾配を利用して顔画像 B の顔部分の画素値を計算すると、境界部分が目立ちにくい自然な中間画像を生成することができた。

3. 3. 2 傷がついた画像の修復

ポアソン画像合成を用いて、画像に生じた傷の修復を行った例を図6に示す。

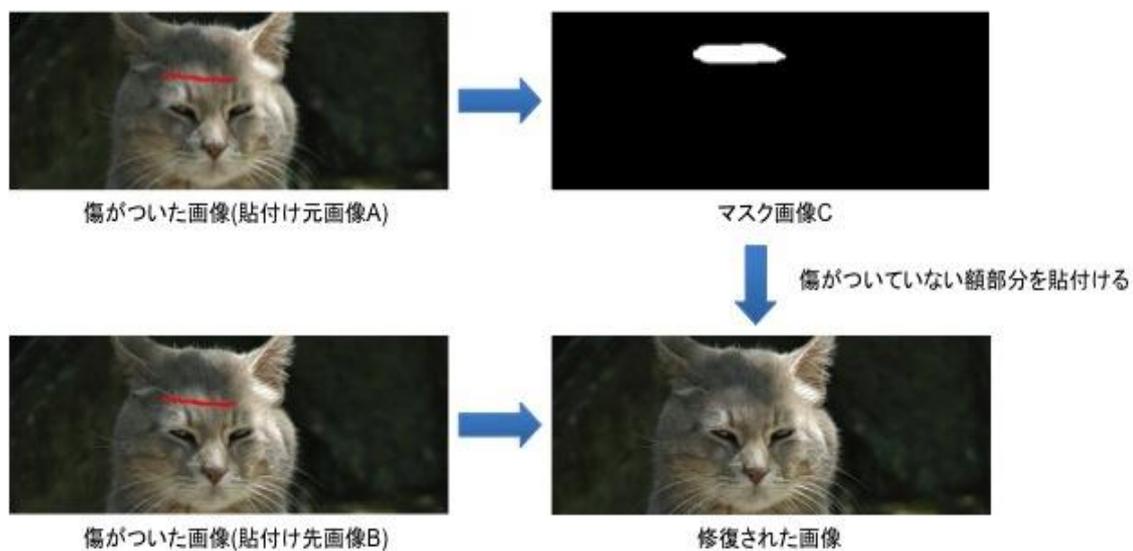


図6. ポアソン画像合成による傷がついた画像の修復

まず、傷がついた同じ画像を2枚用意し、貼付け元画像 A、貼付け先画像 B とする。次に、傷がついていない部分で、傷がついている部分に画像パターンの近い領域をマスク画像 C で指定して切り出す。そして、その傷がついていない領域の領域の勾配を利用して傷がある部分の画素値を計算することにより、修復された画像を生成することができる。

4. 簡易な実装と詳細な実装の比較

4章では、簡易な実装と詳細な実装で合成された画像を比較することで、両者の違いを明らかにした。

4. 1 簡易な実装と詳細な実装の適用例と比較

簡易な実装と詳細な実装をサンプル画像に適用した。図7に結果画像を示す。図7から、どちらの実装による結果画像も視覚的には差がないように見える。



簡易な実装を適用した結果画像



詳細な実装を適用した結果画像

図7. 自然画像に対して簡易な実装と詳細な実装を適用した結果

このように自然画像を用いた合成では、簡易な実装と詳細な実装の差を視覚的に確認することができなかった。そこで、ペイントツールで作成した人工的な画像(貼付け元画像)を用いて、境界部分において差が出るかどうかを調べる比較実験を行った。

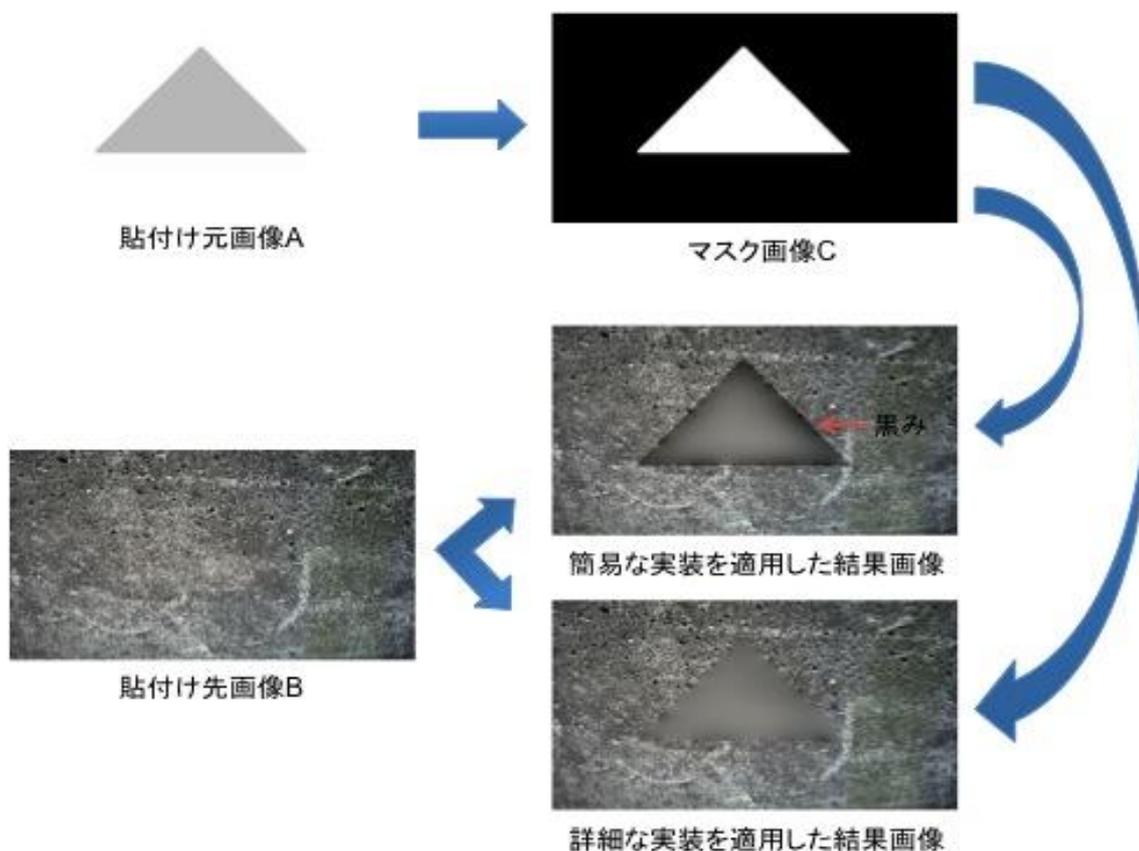


図8. 人工的な画像を使って簡易な実装と詳細な実装を適用した結果

図8の結果画像で簡易な実装と詳細な実装を比較すると、貼り付けた三角形の境界部分に明確な差が見られる。簡易な実装を適用した結果画像では、貼り付けた三角形の境界部分に黒みがかかっており境界が不連続な結果画像になっている。詳細な実装を適用した結果画像では、三角形の境界部分に黒みがかかっておらず、境界部分の画素値が連続した結果画像になっている。簡易な実装では、境界条件を考慮せず、貼付け元画像の勾配を使って画素値を計算する。従って、境界部分の勾配、すなわち画素値の段差がそのまま合成画像に現れたものと考えられる。一方、詳細な実装では境界部分には貼付け先画像の元々の画素値を用いるようにすることで、段差が緩和された画像合成が行われたと考えられる。このように、簡易な実装と詳細な実装の違いは境界部分がより目立ちにくくなる処理を施すか、しないかということが再確認された。その結果、簡易な実装では図8のように境界部分が目立つ場合がある。図7のような顔画像の合成では、簡易な実装と詳細な実装に、ほとんど差が生じない。

5. ポアソン画像合成と単純画像合成の加重平均による画像合成

この章では、ある利用場面におけるポアソン画像合成と単純画像合成の比較を行う。そこでの両手法の特徴を調べ、互いの長所を生かせる画像合成手法を提案して、効果を実証する。

5. 1 ポアソン画像合成と単純画像合成の比較

ポアソン画像合成と単純画像合成を同じサンプル画像に適用し、結果画像の比較を行う。図9に適用した例を示す。

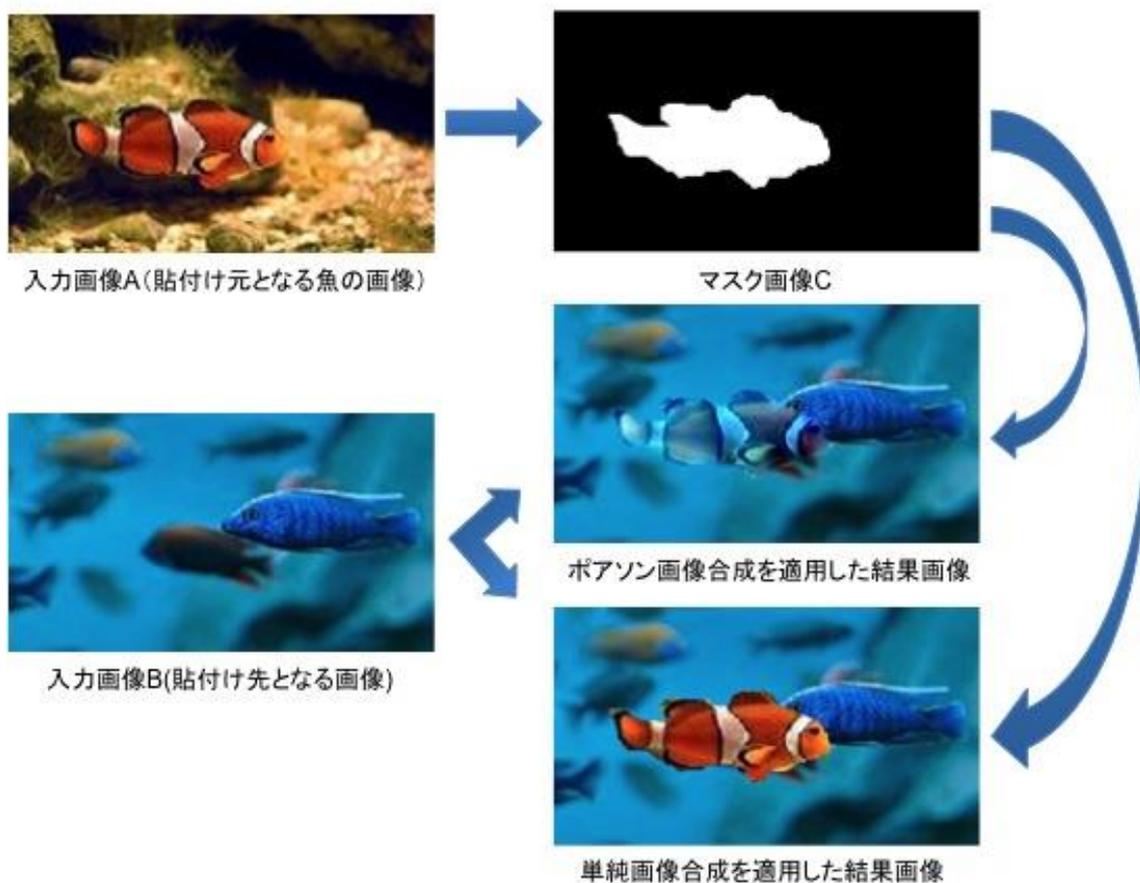


図9. ポアソン画像合成と単純画像合成の比較

図9においてポアソン画像合成を適用した結果画像では、入力画像 A から切り出した魚の色味が、入力画像 B の背景色の影響を受けすぎている。それに対して、単純画像合成を適用した結果画像では、背景色の影響を全く受けず、魚の色味があまりにも鮮明になっている。この場合は、どちらの方法も最適な結果画像を生成していない。

ポアソン画像合成では、貼り付け元画像の物体と貼り付け先画像の背景の色味の差が大きいと不自然な画像合成になる。ポアソン画像合成は、貼付け元画像の色味を合成することには適していない。しかし、3. 3節の適用例からわかるように、2枚の顔画像の中間顔画像を生成することや、画像に生じた傷の修復はうまく行える。2画像間の似た特徴の画像合成や傷の修復には適していると考えられる。これに対して、単純画像合成は貼り付け元画像の画素値をそのまま貼り付け先画像の画素値として利用するため、色味が多少違っていても、ある程度は違和感の無い自然な合成ができる。そのため、色味の合成には適していると考えられる。しかし、2. 1節の単純画像合成の適用例から、2画像間の似た特徴の画像合成には適していないと考えられる。

これらの考察から、「ポアソン画像合成と単純画像合成の適用後に加重平均すれば、境界部分が目立たず色味も自然な画像合成を行える」という手法を提案する。

5. 2 改良ポアソン画像合成

提案手法を図10に示す。この改良されたポアソン画像合成では、まず、ポアソン画像合成と単純画像合成を行って2つの画像を得る。次いで、これら2つの画像を加重平均することで、それらを任意の割合で合成した中間画像を合成する。実際に色味が変化するのはマスク画像によって切り出した部分だけである。

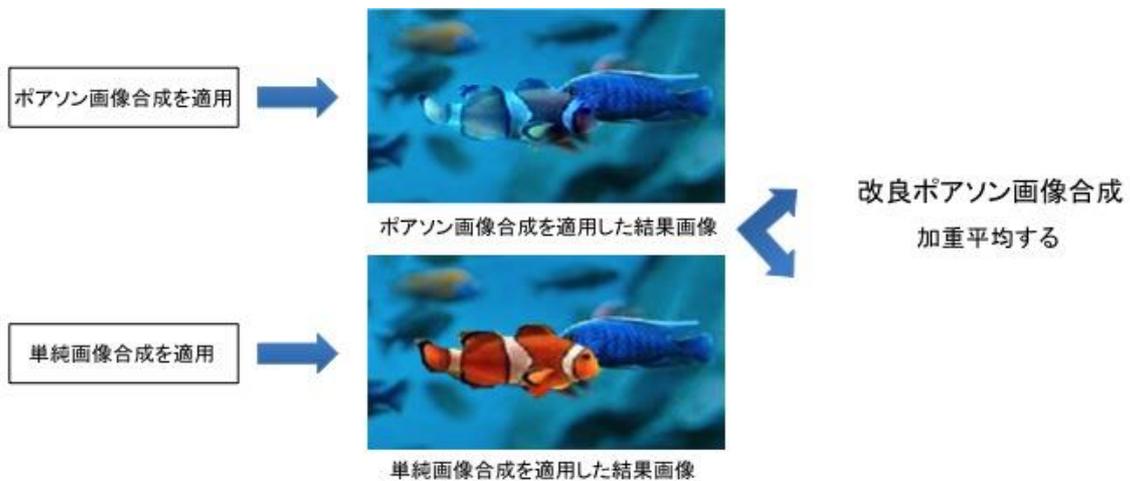


図 10. 提案手法の説明図

5. 3 実験例

5. 1 節で使用した画像を用いて、改良ポアソン画像合成の効果を検証する。図 11 に加重平均の割合を変えた 6 枚の画像を示す。

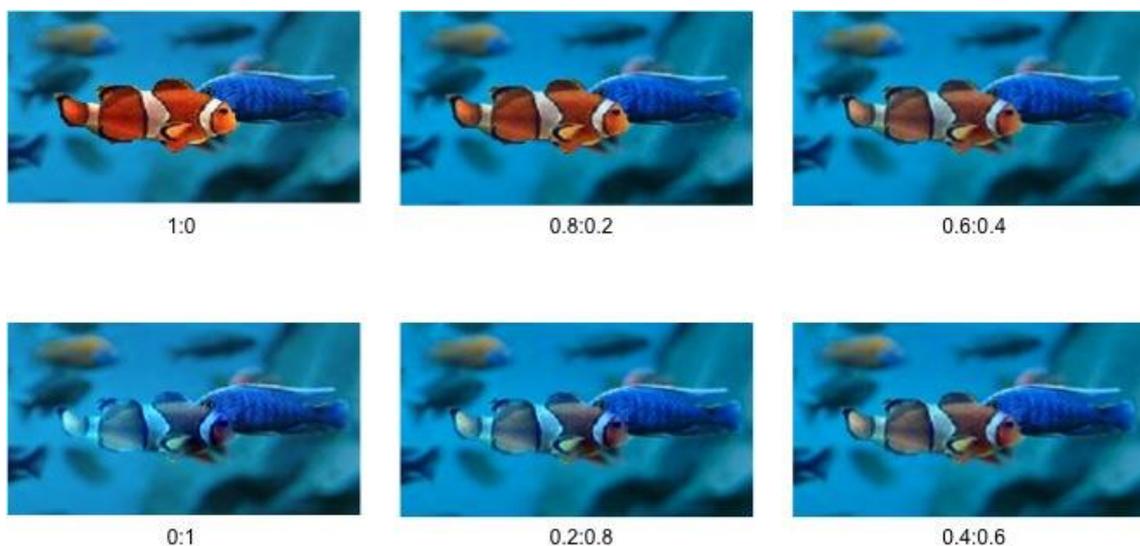


図 11. 加重平均による画像合成を適用した結果

図 11 で、加重平均の割合が 1:0 の時は単純画像合成のみを適用した結果画像に、0:1 の時はポアソン画像合成のみを適用した結果画像になっている。この中

では、0.8:0.2 あるいは 0.6:0.4 で加重平均したものが、最も自然に見える。

6. おわりに

本研究では、まず、ポアソン画像合成と単純画像合成の仕組みを調べた。次に、ポアソン画像合成をより深く理解するために画像合成のプログラムを開発し、いくつかのサンプル画像に適用した。その結果、効果を確認することができた。また、簡易な実装と詳細な実装の違いを調べた。その結果から、自然画像では違いは見られないが、人工的な画像では境界部分において差が出るのがわかった。最後に、ポアソン画像合成と単純画像合成の利用場面の比較を行った。ポアソン画像合成は、2つの顔画像の中間画像の生成や傷の修復といった2画像間の似た特徴の画像合成や修復には適していることがわかった。これに対して単純画像合成では、物体を単純に背景画像に貼り付ける用途に適していることがわかった。この考察から、両手法で得た2枚の画像を加重平均する改良ポアソン画像合成を提案した。この手法は、加重平均の割合を変えることにより、境界部分が目立ちにくく色味の違和感もあまりない画像合成が実現できることがわかった。

しかし、マスク画像の作り方によっても結果画像に影響する問題点があるため、今後の課題は適したマスク画像を作成することである。そして、将来的にはどのような適用例であっても自然な画像合成が実現できるようにしたい。

謝辞

本研究を進めるにあたり、丁寧で熱心なご指導を頂きましたコンピュータ理工学部蚊野研究室 蚊野浩 教授に深く感謝致します。

参考文献

- [1] 情報処理振興事業協会(IPA), 京都高度技術研究所:擬人化音声対話エージェント, **Face Fit**.
- [2] 浅井 文男, “フォトレタッチソフトを利用した衛生画像処理,” 工学・工業教育研究講演会講演論文集, pp.163-166, 2000 年.

- [3] 山崎 俊彦, “100 行で書く画像処理最先端 勾配ベースの画像編集 Poisson Image Editing,” 映像情報メディア学会誌, Vol. 64, No.5, pp.729-737, 2010 年.
- [4] 遠藤 雅守, “Poisson 方程式シミュレーション,” 東海大学理学部物理学科, <http://teamcoil.sp.u-tokai.ac.jp/classes/EM1/Poisson/index.html>
- [5] Patrick Perez, Michel Gangnet, Andrew Blake, “Poisson image editing,” ACM SIGGRAPH 2003 Papers, pp. 313-318, 2003.
- [6] ビジュアル情報処理編集委員会, “ビジュアル情報処理 CG・画像入門,” 財団法人 画像情報教育振興協会(CG-ARTS 協会), pp72, 2004 年.

付録 1 . PoissonDlg.cpp(ポアソン画像編集による画像合成を行うプログラム)

付録 2 . WeightedAverageDlg.cpp(加重平均を行うプログラム)