

モバイル環境で利用可能なライブ画像合成

中村俊介^{†1} 田中正行^{†1} 奥富正敏^{†1}

近年、画像合成処理の分野において、さまざまな画像合成が可能となってきた。しかし、多くの画像合成処理は、画像処理の知識を有しない多くのユーザーまで浸透しているケースは少ないといえる。その主な原因として、画像合成処理を行うためには、煩雑な作業を必要とすることや、コンピュータや画像処理の知識が必要であるためである。本論文では、画像合成処理の知識を必要とせず、誰でも簡単に画像合成処理を使用してもらうための新しい画像合成手法を提案する。

誰でも簡単に画像合成処理を使用するための方法として、最終画像合成結果を確認しながら撮影を行うライブ画像合成撮影を提案する。このライブ画像合成撮影の実現には、画像合成処理を高速に行う必要がある。しかし、従来手法では、ユーザーが前景や背景とする領域を指定し、画像合成したいオブジェクト領域を正確に抽出する必要があり、高速に処理することは困難である。また、この従来の画像合成方法では、ユーザーが合成に使用する領域を指定する必要があり、煩雑であった。そこで、本論文では、ユーザーによるオブジェクト領域抽出を必要とせずに画像合成を行う方法として、背景領域において画像結合を行う画像合成方法を提案する。

本論文では、ライブ画像合成撮影を実現する方法として、モバイル端末上でリアルタイムに画像合成処理を行い、モバイル端末のライブビューを通して、画像合成処理の最終結果をユーザーに提示しながら撮影を行うライブ画像合成システムを提案する。この提案手法を利用する事により、ユーザーは画像合成結果を撮影確認しながら撮影する事が可能となる。

1. はじめに

近年、AgarwalaによるInteractive Digital Photomontage¹⁾やRotherによるAutoCollage²⁾などといった手法が提案され、比較的簡単な作業で画像合成を行うことが可能となってきた。しかし、多くの画像合成処理は、画像処理の知識を有しない多くのユーザーまで広く浸透しているとは言えない。その主な原因として、従来の画像合成処理は、煩雑な作業を必要とするためである。画像編集ソフトや従来手法^{1),2)}を用いることで、煩雑な作業を軽減することが可能であるが、画像編集ソフトを使いこなすための訓練や、ノウハウを要するため、多くのユーザーが利用可能な手法とは言えない。本論文では、画像合成をするために撮影を行い、撮影した画像を使用して画像合成処理を行うことを考える。このような場合、自然な画像合成を行うためには、撮影時から合成結果を想定して撮影を行う必要がある。しかし、従来の画像合成処理では、撮影を行っている段階では、最終的な画像合成結果を直接確認することができないため、合成結果を想定した撮影が困難である。

以上の理由から、従来の画像合成処理は、ユーザーの使いやすさを考慮しているとは言えない。そこで、画像合成処理を、広く使用してもらうために、ユーザーの使いやすさを考慮した画像合成手法を提案する。

本論文では、ユーザーにとって使いやすい画像合成方法について考察を行い、ユーザーの使いやすさを考慮した画像合成を提案し、誰でも簡単に画像合成を行えるシステムの実現を目的とする。

2. 新しい画像合成処理のコンセプトと概要

2.1 ユーザーの要求

このとき、画像合成処理をする際にユーザーが要求することが想定される項目について、以下に簡潔に述べる。

1. 操作が簡単で使いやすい。
2. 最終的な合成画像結果を確認しながら撮影したい。

まず、1について述べる。画像処理にかかわらず、操作が複雑なシステムは、ユーザーに受け入れられにくい。したがって、画像合成処理を行う際も同様に、簡単で分かりやすい操作で、作業が完了することが望まれていると言える。

次に、2について述べる。画像処理に限らず多くの処理においては、最終結果を確認しながら処理する事ができれば、ユーザーにとって有益であるといえる。その一例として、多くのデジタルカメラは、現在のカメラの撮影結果をリアルタイムにディスプレイに表示するライブビュー機能が搭載されており、撮影結果を確認しながら撮影を行うことができる。デジタルカメラが、世代を選ばず多くの人に利用されているのも、ライブビュー機能によって、その場で撮影結果を確認しながら撮影できることが

^{†1} 東京工業大学大学院 理工学研究科
Tokyo Institute of Technology

大きいと考えられる。よって、画像合成処理結果を確認しながら撮影できるシステムも、ユーザーに望まれていると考えられる。以上の項目を考慮しながら、誰でも簡単に画像合成処理を行うことが可能な方法を提案する。

2.2 従来手法の画像合成とその問題点

従来の画像合成処理に必要な作業について簡単に述べる。まず、画像処理を行う画像を選択し、必要なオブジェクト領域を抽出する。次に、合成したい画像に抽出したオブジェクト領域を張り合わせ、貼り付け先との位置調節を行い、合成画像を保存する。しかし、自然な画像合成を行うには、先に述べた手順だけでは、十分とはいえず、その他にも複数の作業を行う必要がある。例えば、オブジェクト領域を抽出する際には、抽出するオブジェクト領域の輪郭にそって、不要な画素を削除する必要がある。また、オブジェクト領域と背景領域を自然に合成するためには、抽出したオブジェクト領域の明るさ、スケール、回転の変更などの調整を行う必要がある。画像を張り付けた後も、張り付けたオブジェクト領域と背景領域の境界をぼかす処理が必要となるなど、画像合成を行うためにはさまざまな手順を踏む必要があり、操作が煩雑で難しい。

オブジェクト抽出を簡単に行う方法として、Li らが提案する Graph Cut を用いた領域セグメンテーションがある^{7),8),9),10),11),12)}。彼らの手法を用い、ユーザーがオブジェクト領域と背景領域を指定することで、精度よくオブジェクト領域を抽出する事が可能となった。しかし、Graph Cut を用いた領域セグメンテーションは、計算コストの高い最適化処理と大量のメモリを必要とする傾向がある。また、領域セグメンテーションを行う方法は、ユーザーが前景と背景とする領域を指定し、セグメンテーション結果を確認する作業を繰り返す必要があるため、煩雑になってしまう。また、オブジェクト領域を抽出する画像合成では、正確にオブジェクトを抽出する必要がある。これは、人間が画像を鑑賞するときに、人物や物などのオブジェクト領域と、その周辺領域を中心に鑑賞するため、不正確に抽出された場合、強い違和感があるためである。

以上の考察より、モバイル端末で動作可能な画像合成システムを実現するためには、Graph Cut を用いた領域セグメンテーションではなく、計算コストの低い、必要な領域選択を高速に処理が可能な、新しいアルゴリズムが必要である。

2.3 提案手法のコンセプト

ユーザーの要求を満たしながら、誰でも簡単に画像合成を行える方法として、画像合成結果を確認しながら撮影を行い、撮影を行うだけで画像合成処理が完了するライブ画像合成撮影のコンセプトを提案する。撮影時に画像合成結果の確認を可能とするため、モバイル環境での実現を考える。

ここでは、2枚の画像を合成する場合を例に、提案する画像合成処理を説明する。提案手法では、ある境界線を用いて2つの画像の背景領域を左右に分割し、背景領域において画像を結合することで画像合成を行う。本論文では、このような処理を背景



(a) 自然な画像合成結果 (b) 位置ずれのある場合 (c) 明るさ変化のある場合
図1 背景領域結合による画像合成結果

領域結合と呼ぶことにする。背景領域結合には、大きく分けて2つの利点がある。一つは、境界領域が、人間の注目度の高いオブジェクト領域周辺ではなく、人間の注目度の低い背景領域であるため、画像境界における違和感が相対的に軽減されることである。もう一つは、正確なオブジェクト領域の抽出が必要ないため、従来手法の領域セグメンテーションを行う方法と比較し、計算コストを大幅に削減することができることである。

次に、撮影しながら画像合成が行える利点を述べる。背景領域結合による画像合成が自動的に行われ、その結果がライブビューを通して確認できるため、ユーザーはカメラの撮影位置を変更するだけで、画像合成する位置調整することが可能である。この操作は、通常のカメラ撮影とほぼ同様な操作であるため、画像処理に関する知識や特別な訓練を必要としない。

以上の提案するコンセプトは、以下のようにまとめることができる。

1. 背景領域で2つの画像を結合し画像合成を行う（背景領域結合）。
2. 画像合成結果を確認しながら撮影を行う。

このようなコンセプトに基づき、新たな画像合成システムを実現することにより、従来手法の問題を解決することができる。

ところで、単純な直線境界を利用した背景領域結合では多くの場合、自然な画像合成結果は得られない。具体的な例を図1に示す。カメラが三脚等で固定されておりかつ、明るさ変化がない場合は、図1(a)に示すように、単純な直線境界であっても自然な画像合成結果が得られる。一方、図1(b)に示すようにカメラが固定されておらず位置ずれが生じている場合や、図1(c)に示すように明るさ変化が生じている場合は、単純な直線境界では、境界が目立ち違和感のある合成結果が得られる。そのため、背景領域結合により、自然な画像を合成するためには、適切な境界を探索する必要がある。次章では、自然に画像結合をするための境界探索方法について述べる。

3. ライブ画像合成システム

3.1 ライブ画像合成処理システム

本論文では、最終的な画像合成結果を確認しながら撮影するライブ画像合成システムを提案する。ライブ画像合成システムでは、気軽に持ち運びが可能なモバイル端末を利用し、ライブビュー機能を通して、ユーザーに画像合成処理結果を提示する。画像合成処理結果を、リアルタイムでライブビューに表示するために、モバイル端末内で高速に画像合成処理を行う必要がある。高速に画像合成処理を行う方法として、2.3で提案した背景領域結合を用いる。

ライブ画像合成システムの概要を図2に示す。ライブ画像合成システムは、大きく2つの処理から構成される。一つ目は、背景領域で画像を自然に結合するための境界を探索するシームレス境界探索である。二つ目は、結合した画像の最適化処理を行うシームレス画像合成である。この2つの処理について、それぞれ述べる。

3.2 シームレス境界探索

本論文では、自然な背景領域結合を行うため、シームレス境界を探索する必要がある。Avidanらが提案するSeam Carving^{3),4),5)}では、画像勾配に基づくコスト関数を設定し、そのコスト関数に基づき、画像理ターゲットングやリスキューリングが行われている。Seam CarvingをVideoフレームに応用したImproved seam carving for video retargeting⁴⁾やDiscontinuous Seam-Carving for Video Retargeting⁶⁾なども提案されている。本論文でも、Seam Carvingと同様に境界の勾配に基づき、コスト関数を設定する。しかしながら、Seam Carvingとは異なり、提案手法では、2つの画像の結合であるため、結合の前後の画像勾配の差に基づく、コスト関数を設定する。

まず、提案手法におけるシームレス境界の概念について述べ、シームレス境界コストを定義する。このシームレス境界コストを最小化する問題は、Graph CutやDP(Dynamic Programming)を用いて最適化する事が可能である。Graph Cutを用いる場合のメリットは、複雑な最適化問題に対応できる柔軟性が高いことである。デメリットとは、計算コストとメモリの使用量が高くなってしまいうことである。DPを用いる場合のメリットとは、計算コストとメモリの使用量を抑え、高速処理が可能であることである。デメリットとは、Graph Cutと比較して、対応できる最適化問題が限定され、柔軟性が低いことである。以上の理由により、本論文では、シームレス境界コストの最小化には、DPを利用することにする。

3.2.1 シームレス境界のコスト関数

画像1と画像2の結合後の境界における画像勾配と、結合前の画像1および画像2の境界における画像勾配が、それぞれ等しい場合、その境界は画像勾配に対してシームレスな境界であると言える。このような画像勾配に対してシームレスな境界を利用して、後述するシームレス画像合成を適用することで、自然に画像を合成すること

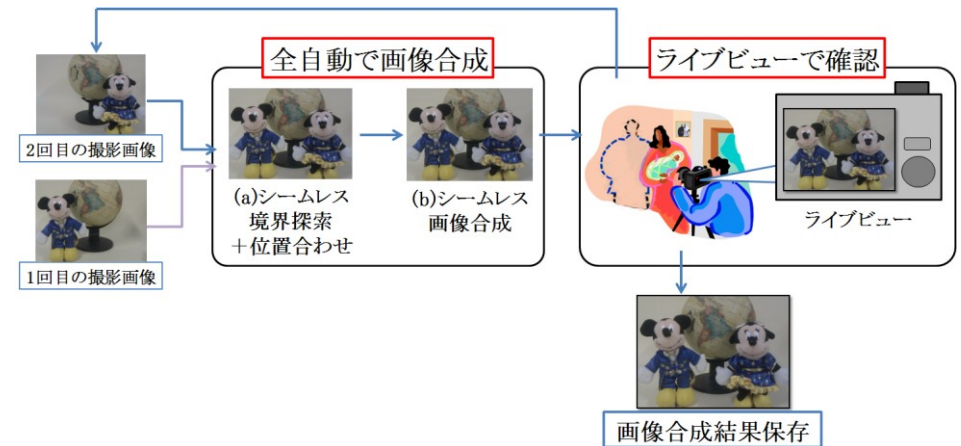


図2 ライブ画像合成システムの構成図

が可能である。以後、画像勾配に対してシームレスな境界を単にシームレス境界と呼ぶことにする。

次に、シームレス境界コストについて、1次元画像を例に説明する。なお、シームレス境界は画素と画素の間に設定されるとする。画像勾配とシームレス境界の関係を図3に示す。画像結合前の画像1と画像2の画素値を、それぞれ I, I' で表す。また、画素位置を (i,j) で表す。まず、画像1の画像勾配と画像結合後の画像勾配を比較する。境界における画像1の画像勾配 $A(i,j)$ は次式のように表される。

$$A(i,j) = (I(i,j) - I(i+1,j)) \quad (1)$$

また、画像1と画像2の画像結合後の画像勾配 $B(i,j)$ は次式のように表される。

$$B(i,j) = (I(i,j) - I'(i+1,j)) \quad (2)$$

$A(i,j)$ と $B(i,j)$ の2つの画像勾配の差分 $d(i,j)$ が小さくなる時、画像を結合した後の画像の境界が自然であると言える。そのため、この境界に対して、次式のようなコストを考える。

$$d(i,j) = (A - B)^2 = (I(i+1,j) - I'(i+1,j))^2 \quad (3)$$

同様に、画像2の画像勾配と画像結合後の画像勾配も同様に考え、式(4)に表す2つの画像勾配の差分 $d'(i,j)$ が小さくなる時、画像を結合した後の画像の境界が自然であると言える。

$$d'(i,j) = (I(i,j) - I'(i,j))^2 \quad (4)$$

そこで、 $d(i,j)$ と $d'(i,j)$ の和を境界におけるシームレス境界コストとして定義する。こ

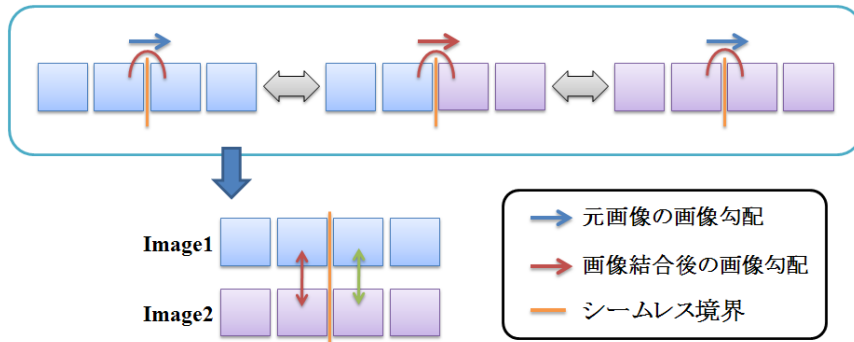


図3 元画像の画像勾配と、画像結合後の画像勾配の比較

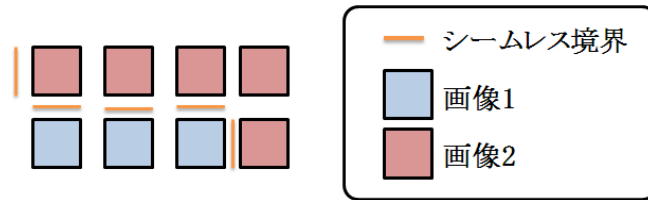


図4 適切でないシームレス境界の移動

のシームレス境界コスト $D(i,j)$ は、式(5)に示すように、結果としてシームレス境界を挟んだ画像の差分を取ることに等しい、

$$D(i,j) = d(i,j) + d'(i,j) = (I(i+1,j) - I'(i+1,j))^2 + (I(i,j) - I'(i,j))^2 \quad (5)$$

この1次元画像のシームレス境界コストを、2次元画像に拡張することで、提案手法におけるシームレス境界コストが得られる。

ところで、シームレス境界の移動量に制限を設けることで、より自然な画像合成が可能となることが Improved seam carving⁴⁾によって知られている。よって、本論文における画像結合も同様に、図4に示すように、横方向に複数ピクセルにわたってシームレス境界が移動しないようにシームレス境界の移動量に制限を設ける。

3.2.2 Dynamic Programming を用いた最適化

本論文における DP を用いた計算方法については、画像の上から下に向けて最適化を行うものとする。

3.2.1 で述べたシームレス境界コストを水平方向の境界に発生するコストと考えると、垂直方向のコストも同様に定義され、次式のように表される。

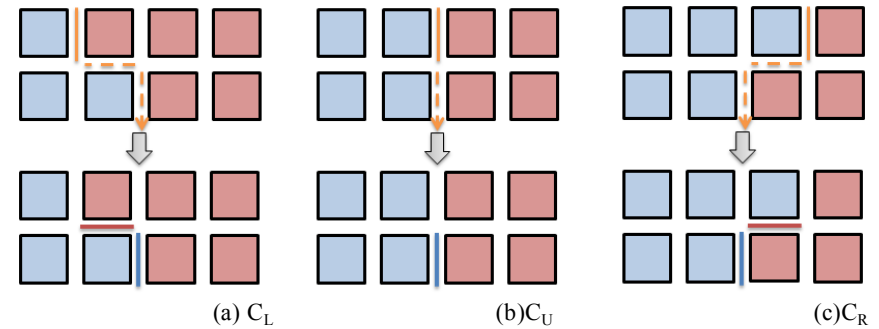


図5 シームレス境界が移動する場合に比較する画像勾配コスト

$$d_v(i,j) = (I(i,j-1) - I'(i,j-1))^2 \quad (6)$$

$$d'_v(i,j) = (I(i,j) - I'(i,j))^2 \quad (7)$$

$$D_v(i,j) = d_v(i,j) + d'_v(i,j) \quad (8)$$

次に、高さ $j-1$ までの、任意の境界位置に対する最小シームレス境界コストと対応する境界が得られているとして、 $M(i,j)$ の位置における境界の最小シームレス境界コストを求めることを考える。水平方向の境界の移動を1画素に制限しているため、可能な境界は図5に示す3通りである。それぞれの場合のコストの増加分を考えると、境界が左に移動する場合、図5(a)の赤線と青線の部分に境界が発生するため、シームレス境界の概念に基づきコスト(a)が発生する。残りも同様にして、境界が下に移動する場合のコストを(b)、境界が右に移動する場合のコストを(c)とする。

$$(a)C_L(i,j) = D(i,j) + D_v(i-1,j) \quad (9)$$

$$(b)C_U(i,j) = D(i,j)$$

$$(c)C_R(i,j) = D(i,j) + D_v(i+1,j)$$

次に、 $M(i,j)$ における局所コストは次式で得られる。

$$M(i,j) = \min \begin{cases} M(i-1,j-1) + C_L(i,j) \\ M(i,j-1) + C_U(i,j) \\ M(i+1,j-1) + C_R(i,j) \end{cases} \quad (10)$$

このように逐次的に、最適化を行うことにより、シームレス境界コストを最小にするシームレス境界が得られる。

3.3 シームレス画像位置合わせ

シームレス境界探索によって、自然に合成される境界を探索する方法を述べた。しかしながら、2つの画像の位置が大きくずれている場合、シームレス境界探索だけでは限界がある。そこで、シームレス境界コストに基づき、位置合わせ処理を行う。本論文では、この位置合わせ処理を、シームレス画像位置合わせと呼ぶことにする。

一般的な位置合わせ処理では、画像間の重なりに対して、画素値の差に基づく SSD (Sum of Squared Difference) を評価関数に用いている。SSD を用いた位置合わせでは、SSD を位置ずれパラメータで微分する事が可能であるため、勾配法に基づく最適化アルゴリズムが用いられている。一方、シームレス画像位置合わせでは、評価関数がシームレス境界コストであり、位置ずれパラメータによる微分は不可能である。

最も単純には、想定する全ての位置ずれ量に対して、それぞれシームレス境界コストを算出することにより、位置合わせすることが可能である。しかしながら、このような全探索では、大きな計算コストが必要である。そこで、本論文では、現在の位置ずれ量と、その周辺の4点に関して、シームレス境界コストを算出し、シームレス境界コストを式(11)の2次関数で、近似することにより、高速に位置合わせ処理を行う。

$$f(x) = ax^2 + bx + cy^2 + dy + e \quad (11)$$

ところで、近似された2次関数が、上に凸な関数となる場合がある。この場合は、図6に示すように、導出した近似関数の微分に基づき、現在の位置ずれ量を更新することにする。

さらに、ライブ画像合成システムにおける位置合わせ処理では、撮影済みの1枚の画像は変化しないものの、撮影中の2枚目の画像は、フレーム毎に常に変化する。そのため、フレーム毎に最大の更新量を設定し、それ以上は更新しないようにした。これは、1フレームであまりにも大きな位置ずれ補正がすると、ユーザビリティが逆に悪化するためである。

3.4 シームレス画像合成

シームレス画像合成では、画像勾配に基づき、画像を合成する。画像境界を目立たなくする手法として、ガウシアンフィルタや、Alpha Blend が知られている。しかしながら、ガウシアンフィルタや、Alpha Blend 等は、エッジやテクスチャなどの画像勾配情報まで失ってしまうため、かえって境界が目立ってしまう事がある。そこで、本論文では Poisson image editing^{13),14),15),16)} を利用する。Poisson image editing は、画像勾配から得られるポアソン方程式を解くことにより、画像合成を実現している。ライブ画像合成システムにおいても、Poisson image editing により、自然に画像が合成されることが期待される。

次に、本論文で用いた Poisson image editing によるシームレス画像合成について述べる。基本的な考えは、従来手法と同じであるが、画像境界において、2枚の画像の画像勾配を平均した後、従来の Poisson image editing を適用している。これは画像境界において、2つの画像の画像勾配が矛盾しているため、その矛盾を解消するためである。提案手法で用いたシームレス画像合成の処理結果例を図7に示す。図7(a)の輝度の単純はめ込みでは明るさ変化のため、画像境界が不自然であるが、図7(b)のシームレス画像合成では自然に画像が合成されていることが確認できる。

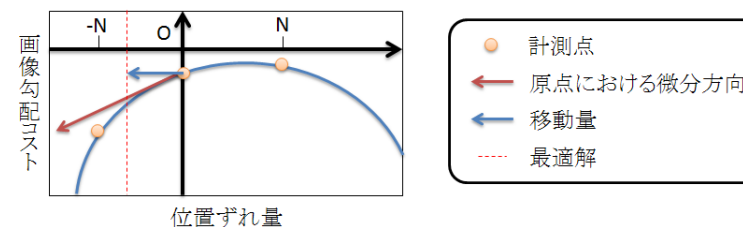
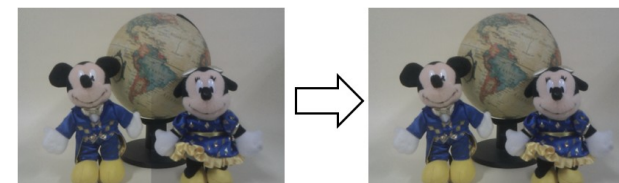


図6 微分を用いたエネルギー最小化方向の推定による最適化



(a) 輝度の単純はめ込み (b) シームレス画像合成結果

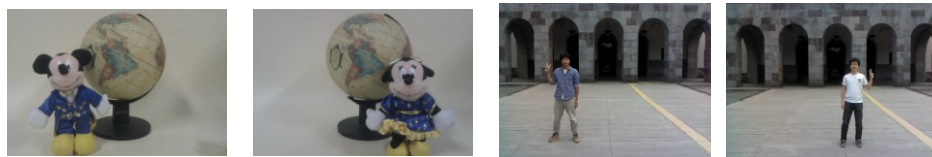
図7 シームレス画像合成結果

4. 実験結果

まず、シームレス境界探索とシームレス画像位置合わせについて、シミュレーション実験を行う。次に、ライブ画像合成システムを用いて、画像合成撮影した結果を示し、ライブ画像合成システムの有効性について検証する。実験を行うために、Android OS を搭載したモバイル端末¹⁷⁾に、ライブ画像合成システムを実装した。ライブ画像合成システムを用いた画像合成処理のユーザビリティテストを行い、提案手法と従来手法において、画像合成が完了するまでの時間を比較する。また、それぞれの手法で作成した画像の画質について評価する。

4.1 シームレス境界探索の実験

提案手法を適用し、480 x 320 の画像サイズの画像を2枚使用し、シームレス境界探索による画像結合の効果を検証した。シームレス境界探索の探索範囲は、 $x = 240 \sim 340$ とする。図8左に、実験に用いた2枚の画像を示す。各手法による結果を図9に示す。図9における画像については、図中上段に画像合成結果、下段に画像境界を赤線で強調した画像を示す。(a)は提案手法である DP によるシームレス境界探索による合成結果である。(b)は従来手法の矩形領域によるオブジェクト選択後、人物の輪郭にそって不要な画素を削除した合成結果である。(b)では、地球儀や、影の部分で色の変化が発生していることが確認できる。(c)は直線の境界によって分割した合成結果である。(c)



シームレス境界探索

シームレス画像位置合わせ

図 8 シミュレーション入力画像

は合成した境界は目立たないが、人物の明るさが違うために、人物が浮いてしまっていることが確認できる。(a)の提案手法では、地球儀や影の部分での色の変化が、他の手法に比べて少ないことが確認できる。

4.2 シームレス画像位置合わせの実験

本節では、シームレス画像位置合わせによる位置合わせの効果を確認する。実際のライブ画像合成システムでは、撮影しながら位置合わせが行われる。そのため、撮影済み画像は変化しないものの、もう一方の入力画像はフレーム毎に変化する。しかしながら、ここでは、撮影済み画像および入力画像ともに、変化しないものとして、シームレス画像位置合わせの効果を確認する。シームレス境界探索範囲は $x=390\sim 490$ とした。また、現在の位置ずれ量と、現在の位置ずれから上下左右それぞれ、5画素の位置ずれに対するシームレス境界コストを計算し、式(11)の2次関数で近似した。さらに、1フレーム当たりの最大更新量を5画素として、シームレス境界コストが収束するまで、繰り返し処理を行った。図8右に、実験に用いた2枚の画像を示す。図10にシームレス画像位置合わせの効果を示す。シームレス画像位置合わせを用いない場合では、背景のアーチ部分において、画像のずれが目立っていることが確認できる。提案のシームレス画像位置合わせにより、アーチ部分の画像のずれが目立たなくなっていることが確認できる。

4.3 ライブ画像合成システムに要する処理時間

本節では、提案手法を用いてツーショット画像の合成撮影に要する時間と、従来手法で、ある被験者が画像編集ソフトを用いて画像合成に要する時間を検証した。比較する時間は、撮影時間と、処理に要する時間である。従来手法については、計算機に画像を取り込む時間は含めないものとする。また、従来手法では、Photoshopを使用したことのある被験者Aと、使用したことのない被験者Bに、Photoshopを用いて画像合成方法を指定せずに編集してもらった。Photoshopを用いて編集を行う画像は、提案手法によって撮影した画像を使用せず、別途撮影を行った。また、提案手法とPhotoshopを用いた従来手法のそれぞれの場合において、操作の練習を1時間設けた。実験では、図11左に示すように、撮影を交代しながら一人ずつ撮影を行った。



(a-1)Graph Cut/ DP

(b-1)人物の手動抽出

(c-1)中央で画像結合

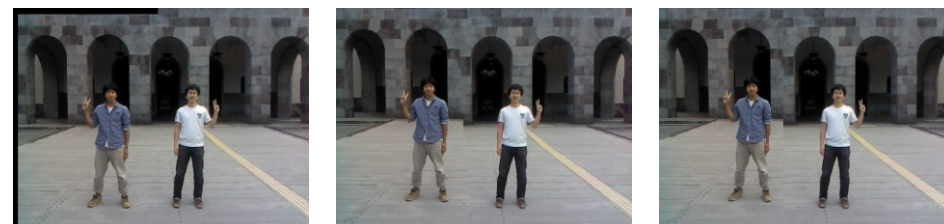


(a-2)境界画像

(b-2)境界画像

(c-2)境界画像

図 9 シームレス境界探と従来手法による画像合成結果



(a)シームレス画像
位置合わせ画像

(b) シームレス境界による
画像結合

(c)単純境界による
画像結合

図 10 シームレス位置合わせ結果

実験で使用したシーンを図11に示す。図11のシーン1は東京工業大学の記念碑の前で撮影した。図11のシーン2は東京工業大学図書館前で撮影した。各手法における画像合成結果を図12、図13に示す。提案手法と従来手法に要した時間を表1に示す。

被験者Aによる手動画像合成では、人物を含めた背景領域を選択し、リサイズして背景画像と位置合わせを行った。その後、消しゴムツールを用いて不要な画素の削除や、インペインティングツールを用いて、画像の境界を修正した。被験者Bによる手動画像合成では、人物を選択ツールで抽出し画像合成を行った。人物の初期抽出後に、消しゴムツールを用いて、不要な画素の削除を行い正確な人物の抽出を行った。最後



図 11 実験風景と実験で使ったシーン

に、ぼかしツールや、インペインティングツールを用いて、抽出した人物の境界を修正した。

提案手法を用いた場合では、3.4 で述べたシームレス画像位置合わせの補助の有無にかかわらず、約 1 分程度で合成画像の撮影が完了した。従来手法を用いた撮影でも同様に、1 分前後で撮影が完了した。従来手法における撮影では、画像合成処理を想定して撮影を行うため、通常の撮影と比較すると、撮影時に多少の時間を要する。そのため、提案手法による画像合成を含めた撮影時間と、従来手法による撮影結果はほぼ同じ撮影時間を要する結果となった。手動画像合成方法では、Photoshop の経験者である被験者 A は、10 分程度で納得のいく画像が完成した。また、Photoshop 非経験者である被験者 B も、およそ 10 分程度で納得のいく画像が完成した。

次に、それぞれの手法による合成結果について検証する。被験者 A による画像合成結果については、背景領域で画像結合しているため、人物の周辺は自然に合成されていることが確認できる。また、背景領域を結合している境界についても、自然に合成されていることが確認できる。被験者 B による画像合成結果については、画像右側の人物を抽出して合成を行っているため、股下にできる影の領域など、境界があいまいな領域や、白い柱との髪の毛の境界のように、抽出した領域と背景の明るさに差がある領域等で、不自然となっている事が確認できる。提案手法は、背景領域で画像結合しているため、人物が自然に合成されていることが確認できる。しかし、図 13(a)のように直線が多いシーンでは、位置合わせ処理が十分でない場合に、一部の領域で不自然な画像結合となってしまうことが確認できた。これは、人間が、輪郭部分に着目するため、本来直線であるべき線が、直線となっていないために、不自然さを強く感じるためだと考えられる。よって、直線を多く含むシーンでは、高速に精度の高い位置合わせを行うことで、自然な画像合成が可能になることが期待できることが分かった。

5. 結論

本論文では、最終的な画像合成処理結果を確認しながら撮影を行う、ライブ画像合成撮影を提案した。ライブ画像合成撮影を実現する方法として、モバイル端末上でリ



(a)ライブ画像合成 (b) 被験者 A による手動画像合成 (c) 被験者 B による手動画像合成

図 12 シーン 1 の画像合成結果



(a)ライブ画像合成 (b) 被験者 A による手動画像合成 (c) 被験者 B による手動画像合成

図 13 シーン 2 の画像合成結果

アルタイムに画像合成処理を行い、モバイル端末のライブビューを通して、画像合成処理の最終結果をユーザーに提示しながら撮影を行うライブ画像合成システムを提案した。

ライブ画像合成システムは、シームレス境界探索、シームレス画像合成の 2 つの処

撮影場所	被験者	提案手法		従来手法	
		位置合わせ補助		撮影	画像合成編集
		なし	あり		
東工大モニュメント前	A	55 秒	47 秒	1 分 20 秒	10 分 13 秒
	B	50 秒	55 秒	1 分 15 秒	11 分 4 秒
東工大図書館前	A	48 秒	38 秒	32 秒	6 分 10 秒
	B	42 秒	45 秒	37 秒	7 分 12 秒

表 1 各手法による画像合成に要した時間

理から構成される。シームレス境界探索では、背景領域で画像結合する境界探索問題に簡略化することで、DP によるモバイル端末での高速処理が可能となった。

シームレス画像合成では、シームレス境界探索で求めた境界の両端に発生する輝度勾配を、Poisson image editing を応用することで、特徴を保存したまま画像最適化が可能な事を確認した。

また、より自然な画像合成を行うために、シームレス境界探索のコストを評価関数に用いた位置合わせ手法を提案した。提案する位置合わせでは、一般的に用いられる特徴点のマッチングによる位置合わせや、画像の差分を用いた位置合わせ手法ではなく、画像合成結果を考慮した位置合わせ処理を選択した。

提案手法であるライブ画像合成システムを用いて、従来手法との比較実験を行なった。その結果、少ない時間で、従来手法と同程度の画像合成処理を行えることを確認した。

参考文献

- 1) A. Agarwala, And M. Dontcheva, “Interactive Digital Photomontage.” SIGGRAPH 2004.
- 2) C. Rother, L. Bordeaux, Y. Hamadi, And, A. Blake, “AutoCollage”, SIGGRAPH '2006.
- 3) S. Avidan, And A. Shamir, “Seam carving for content-aware image resizing.” SIGGRAPH 2007.
- 4) S. Avidan, And A. Shamir, “Improved seam carving for video retargeting.” SIGGRAPH 2008.
- 5) M. Rubinstein, And A. Shamir, “Multi-operator Media Retargeting.” SIGGRAPH '2009.
- 6) M. Grundmann, And V. Kwatra, “Discontinuous Seam-Carving for Video Retargeting.” CVPR 2010.
- 7) Y. Li, J. Sun, C.-K. Tang and H.-Y. Shum: “Lazy snapping”, SIGGRAPH '2004.
- 8) C.Rother, And V. Kolmogorov, “GrabCut” — Interactive Foreground Extraction using Iterated”, SIGGRAPH 2004.
- 9) 永橋知行, “グラフカットによる領域セグメンテーションに関する研究”, 中部大学大学院
- 10) 石川博, “グラフカット” 2007 年 3 月 CVIM 研究会チュートリアル. 名古屋市立大学修士論文, 2007.
- 11) A. Blake, And C. Rother, “IFast Approximate Energy Minimization via Graph Cuts” Pattern Analysis and Machine Intelligence, IEEE Transactions on.

- 12) Y. Boykov, O. Veksler, R. Zabih, “Interactive Image Segmentation using an adaptive GMMRF model.” COMPUTER VISION - ECCV 2004.
- 13) P. Pérez, And M. Gangnet. And A. Blake., “Poisson image editing” SIGGRAPH '2003.
- 14) S. Jeschke, D. Cline, And P. Wonka, “A GPU Laplacian solver for diffusion curves and Poisson image editing” SIGGRAPH Asia 2009.
- 15) 神尾亮, 田中正行, 奥富正敏 “勾配情報と輝度情報に基づくシームレス画像合成”, 第 16 回画像センシングシンポジウム (SSII2010) 講演論文集, IS2-03-1-7, 2010
- 16) 神尾亮, 田中正行, 奥富正敏 “勾配情報に基づく画像合成のためのポアソン方程式安定化”, 情報処理学会研究報告[コンピュータビジョンとイメージメディア], Vol.2010-CVIM-172, pp.1-7, 2010
- 17) 東芝 “スレート型タブレット REGZA Tablet AT300”
http://dynabook.com/pc/catalog/regza_t/110420at300/