

複数のスマートフォンカメラの協調利用による 実時間隠消現実感

本田 俊博^{*1}

斎藤 英雄^{*1}

Real-Time Diminished Reality using Multiple Smartphone Cameras

Toshihiro Honda^{*1} and Hideo Saito^{*1}

Abstract – In this paper, we present a system for real-time Diminished Reality using multiple smartphones. In this system, we assume multiple smartphones capture the same scene that is occluded by obstacles. Areas of the obstacles are extracted from each camera image replaced with image of the hidden areas which are captured using different viewpoint camera. In the proposed method, we suppose that the target scene can be approximated as a plane. Therefore, we compute Homography between each camera image by using natural features. Then, obstacle area which is not approximated as a plane can be removed by synthesizing the image warped with the Homography and the user viewpoint image. We can perform real-time processing because we send each camera image to PC which returns obstacle-removed images at every frame. We experimentally demonstrate the effectiveness of the proposed method using three viewpoint images.

Keywords : Diminished Reality, multiple smartphones, real-time, Homography

1 はじめに

近年、コンピュータビジョンの分野の中で拡張現実感 (Augmented Reality) と呼ばれる技術が注目されている [1]. 拡張現実感とは、仮想物体をあたかも実環境に存在するかのように重ね合わせて表示する技術である。この技術を用いることにより、ただ実環境を見ているだけでは得られないような情報を仮想物体の重ね合わせという方法で付加させることができる。現在までに、拡張現実感を利用したシステムが多く研究されている。その例として、紙地図を携帯型カメラで撮影すると対応した3次元モデルを表示することができるシステム [2], 部屋のモデルを作ることによって環境光が入ってくる様子をシミュレーションし視覚化できるシステム [3], 既存のビデオに対し視点変換や透明度変化を行い教師ビデオを作成し、ユーザの作業支援をするシステム [5] などがある。このように、仮想物体を実環境に付加させることでユーザを視覚的に支援することができるので、ナビゲーションシステム、医療・福祉、建築デザイン・都市設計、教育、ゲームなど、幅広い分野に渡って拡張現実感の技術は利用されている。

一方、拡張現実感が仮想物体を実環境に重ね合わせるのに対して、実環境中の物体を取り除く隠消現実感 (Diminished Reality) [4] と呼ばれる技術がある。隠消現実感とは、実環境中の物体 (障害物) を取り除き、そ

れらを適切な対象シーンの画像に置き換えることで、あたかも物体が除去されたかのように見せる技術である。今までこの隠消現実感という技術は、拡張現実感と組み合わせることで、特に建築デザイン・都市設計の分野で非常に有用な技術であった。しかしながら、建築デザイン・都市設計以外でも、日常生活において目の前にある障害物のために、見たい対象シーンが見えないという場面が多々ある。例えば、サッカーや野球などのスポーツ観戦において前の席の観客が障害物となってしまうたり、学校の授業において教師が障害物となり黒板が見えなくなってしまうたり、動物園において檻が障害物となって動物が見づらくなってしまうたり、美術館において前にいる人で見たい絵画が隠れてしまったりなど、様々な状況が考えられる。

建築デザイン・都市設計の分野においては、障害物の位置は固定であり、その背景である対象シーンも変化することはない。しかし、上で挙げたような日常生活で起こりうる場面では、障害物となる人は動いていると考えられる。また、スポーツ観戦においては、見たい対象シーンすなわちフィールド上での試合経過は常に変化している。ユーザの視点についても常に同じ位置から見ているわけではないので、自由に視点移動している。今まで建築デザイン・都市設計の分野で用いられてきた隠消現実感では、障害物は固定、対象シーンは変化なし、視点移動もしない、という制約条件下での手法のため、スポーツ観戦等においてはそのまま

^{*1}慶應義塾大学

^{*1}Keio University

適用することができない．そのため，障害物の移動，対象シーンの変化，ユーザの視点移動に対応した隠消現実感を実現する必要がある．これを実現するためには，従来とは異なる新しいアプローチが必要であると考えられる．

従来行われてきた隠消現実感の手法は，大きく分けて，動画像を用いる手法、複数視点画像を用いる手法、1 視点 1 フレームの画像のみを用いる手法の 3 つに分けることが出来る．

動画像を用いる手法について述べる．Mann ら [6] は，動画像において，ある平面物体をユーザが作成した別の平面物体に置き換える手法を提案した．また，Wang ら [7] は，動画像を motion segmentation algorithm により同じような動きをしている領域に分割し，それぞれにレイヤーを割り当て，あるレイヤー上に他のレイヤーを重畳し取り除くという手法を提案した．他にも，Lepetit ら [8] は，動画像においてユーザが指定した境界線を追跡してオクルージョンを検出し，シーンの物体を取り除く手法を提案した．上で挙げた手法は，ビデオカメラで撮影された動画像の明度の時系列を利用している．これらの手法は，動画像の視運動に基づいて物体の分割と追跡を行うことが可能であるが，処理が複雑で処理時間が遅いという欠点がある．

複数視点画像を用いる手法について述べる．Zokai ら [9] は，画像明度の時系列を利用する代わりに，大きく離れた位置から撮られた異なる 3 枚の静止画像の投影モデルを用いて，適切な背景を生成する手法を提案した．この手法は，複数視点からの画像を用いて，障害物の裏の対象とするシーンを得ることで障害物の除去を行うというものである．しかし，障害物の境界はユーザが指定する必要があるため障害物が動くような動的シーンには適さないという欠点があった．

1 視点 1 フレームの画像のみを用いる手法について述べる．Herling ら [11] は，障害物の周りの画素値から障害物に隠れた領域の画素値を推定し，障害物を除去したように見せる手法を提案した．このシステムではユーザが最初は大雑把な障害物領域を指定する他には事前学習や複雑な設定が必要なく，カメラも 1 台だけでよく，実時間で動くという特徴がある．しかし，実際に障害物に隠れた領域の画素値を取得しているわけではないので正確には障害物を除去したとは言えず，また障害物の周りは単純なテクスチャでなければならないなどの制約もある．

これらの他にも，隠消現実感という言葉は用いていないが，”物体を透視する”という研究は成されてきた．川上ら [10] は，光学迷彩を用いて対象物を透視する手法を提案した．この手法では，透視したい対象物に再

帰性反射材という特殊な素材を貼付し，事前に，もしくは実時間で撮影した背景映像をプロジェクタで投影することで，対象物の透視を実現するシステムである．しかし，このシステムでは再帰性反射材という特殊な素材が必要であり，またプロジェクタも必要となるので，装置が大掛かりとなりあまり実用的ではない．

上で述べたように，障害物を除去する隠消現実感の研究は従来でも行われてきたが，それらの研究は障害物の位置は固定，対象シーンは時間的に変化なしという制約条件下のものであったり，オフライン処理で行うためユーザの自由な視点移動に対応していなかったり，障害物に隠れた領域の画素値を取得していなかったり，大掛かりな装置が必要であったりという問題がある．

Enomoto ら [12] は，カメラ 3 台と ARTag というライブラリを用いて Homography を算出し，ある 2 台のカメラ画像を別の 1 台のカメラ画像の視点に合わせ，ブレンド処理によって障害物を除去するという手法を提案し，これらの問題に対処した．しかし，ARTag を利用するためには障害物の近くに固定マーカを貼付する必要がある，またそれぞれのカメラを別々の PC につなげて有線ネットワークを通じてデータを共有しているため，ユーザの移動範囲は PC の近辺に限られているという問題がある．

そこで本論文では ARTag の代わりに自然特徴を用いることで固定マーカを使わずに Homography を算出し，複数のユーザそれぞれがスマートフォンを用いることで自由に移動できる隠消現実感を実現する手法を提案する．本手法では 3 台のスマートフォンを用い，全てのスマートフォン上に障害物が除去された画像を表示する．本研究は美術館に展示された絵画や，遠距離にいるステージ上のパフォーマーなど，平面または平面に近似できるシーンを対象とする．対象シーンを平面または平面に近似できるものと仮定することで，自然特徴点同士のマッチングから Homography を算出することができ，固定マーカが不要となる．そして Homography を用いて視点変換した 2 枚の画像と基準画像の画素値の中央値を取ることで障害物を除去する．

2 障害物除去処理

2.1 システムの概要

図 1 にシステムの概略図を示す．本システムは 3 台のスマートフォンと 1 台のサーバにより構成される．サーバには無線 LAN ルータが接続されており，無線通信でスマートフォンとサーバ間の画像の受け渡しを行っている．各ユーザはスマートフォンに付属しているカメラから障害物のあるユーザ視点画像を得る．そして他のユーザのカメラ画像を用いて本手法の処理を

施すことで、対象シーンの前に存在する障害物が除去された出力画像を得ることができる。

本システムが対象とする環境は、ユーザが見たい対象シーンが平面または平面に近似できるものであり、その対象シーンの前に障害物が存在しているというものである。

なお、以後の説明では便宜的に対象シーンの正面ににいるユーザが取得する画像を正面画像、正面ユーザの左右にいるユーザが取得する画像をそれぞれ左画像、右画像とし、正面画像に映っている障害物を除去する場合について述べる。実際には左画像、右画像に対しても同様な処理を行っている。

図2に本システムを実現するための提案手法の流れを示す。まず、正面画像、左画像、右画像の自然特徴を抽出する。特徴点マッチングにより Homography を算出し、左画像と右画像を正面視点に変換した画像を生成する。そして、生成された2枚の画像と正面画像を用いてメディアン処理による画素値算出を行い、障害物が除去された画像を生成する。本手法は常に3台のスマートフォンから画像を取得し続け、処理を行っているため、対象シーンが変化したり、障害物が動いたり、ユーザが動いたりしても対応できる。

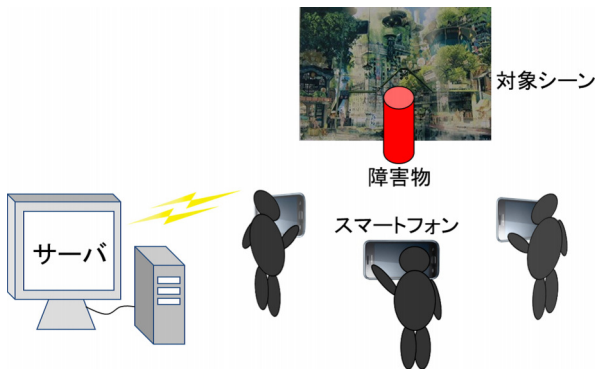


図1 システムの概要
Fig. 1 Overview of the system

2.2 自然特徴抽出

自然特徴を抽出するには、まず自然特徴点を検出し、さらにその点がどのような特徴を持っているかを記述する必要がある。本システムでは自然特徴抽出手法に Leutenegger らの BRISK(Binary Robust Invariant Scalable Keypoints)[13] を用いる。抽出された自然特徴は Homography 算出に使用される。

2.3 自然特徴を用いた Homography 算出手法

Homography の自由度は8であり、また1組の対応点は2個の式を与えるので、4組以上の対応点が必要である。よって、左画像を正面視点から見た画像に変換する Homography を算出するには、左画像と正面画像における対応点を4組以上見つければよい。この

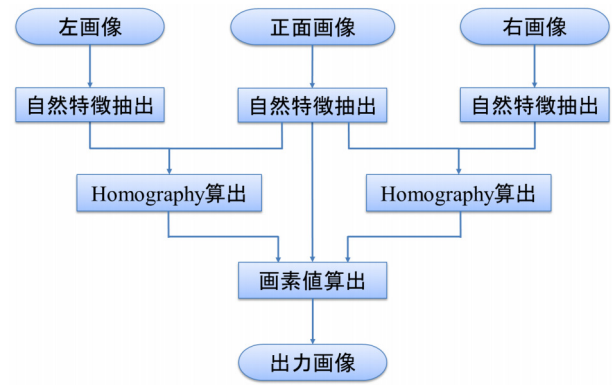


図2 本手法の流れ
Fig. 2 Flow of the proposed method

時、変換される画像(左画像、右画像)のディスクリプタをクエリディスクリプタ、基準となる画像(正面画像)のディスクリプタを訓練ディスクリプタと呼ぶことにする。BRISKではディスクリプタは2値データなので、クエリディスクリプタと訓練ディスクリプタの類似度はハミング距離によって知ることができる。誤対応を減らすため、閾値以上のハミング距離を持つ訓練ディスクリプタしか存在しないクエリディスクリプタは対応付けないようにしている。図3に実際にマッチングが行われた例を示す。丸で囲まれた箇所が特徴点、線の引かれた点同士がマッチングされた対応点の組を表す。

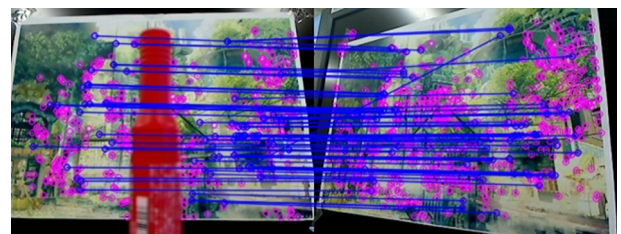


図3 マッチング結果
Fig. 3 Result of matching

対応点の組から Homography を算出する際に、Open Computer Vision Library[14] と呼ばれるライブラリを用いた。まず誤対応が存在してもロバストに Homography を算出できるように、Fischler らの RANSAC(Random Sample Consensus)[15] という手法を用いて、Homography 算出に用いる対応点の組を定める。その対応点の組を用いて DLT(Direct Linear Transformation) 法[16] により Homography を算出する。

誤対応が多すぎる場合には、正しい Homography が算出できないと考えられる。その場合、左画像または右画像は正面視点の画像に変換することが出来ず、2.5で述べるメディアン処理を行っても障害物は除去でき

ない. このような場合は, 正面画像をそのまま出力画像とすることで, 見づらさを軽減している.

左画像を正面視点に変換する場合を例として, Homography が正しいかどうか判別する手法について述べる. Homography が正しい場合, 左画像と正面画像は正しくマッチングされた対応点の組が多く存在すると考えられる. そのため計算により求めた, 左画像を正面視点に変換する Homography H_l を左画像にかけると, 左画像においてマッチングで使われた特徴点が H_l によって移動した座標と, 正面画像における対応する特徴点の座標との距離の差 (再投影誤差) は小さくなる (図4参照). 本システムでは, 再投影誤差が5画素分より大きい対応点の組の数が全体の組の数の4/5より多い場合に Homography は正しくないと判断し, 正面画像をそのまま出力している. 5画素, 4/5 という値は, 正しくない Homography をなるべく用いず, かつ正しい Homography が正しくないと誤判定されることが少なくなるように, 実際の出力画像を見て試行錯誤的に決定した.

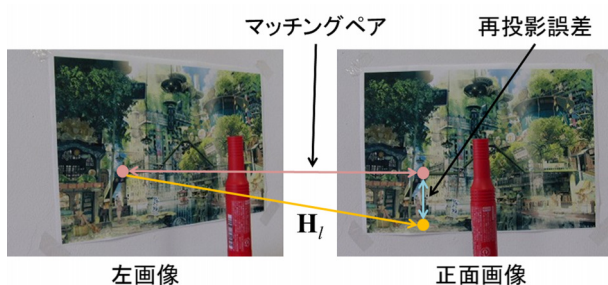


図4 再投影誤差
Fig.4 Reprojection error

左画像, 右画像から正面画像への Homography がどちらも正しいと判断された場合, その Homography を用いて, 図5のように左画像と右画像を正面視点の画像に変換する. なお, H_l , H_r はそれぞれ左画像, 右画像から正面画像への Homography を表す. 以後, これらの変換された左画像と右画像を便宜的に左正面画像, 右正面画像と呼ぶことにする.

2.4 明度値補正

光源の方向やカメラのコントラスト調整などにより, それぞれのカメラから取得する画像の全体的な明度が異なる場合がある. そのため, 出力画像の明度値を算出する前に, 明度値の補正を行う.

左画像の全体的な明度を正面画像の全体的な明度に合わせる場合を例として, 明度値補正手法について述べる. Homography 算出の際に各画像間の対応点を求めるが, 対応が正しければそれぞれの対応点の明度値は一致するはずである. よって, 左画像の対応点の明

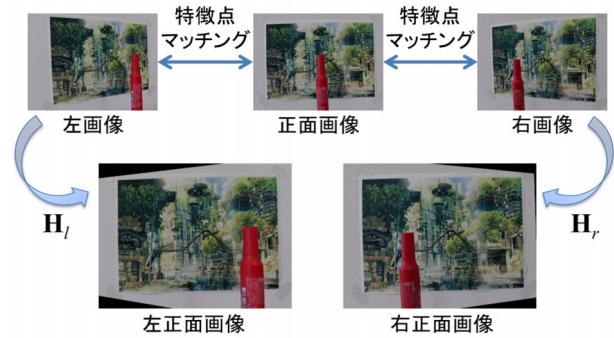


図5 Homography を用いた左右画像の変換
Fig.5 Warping right and left images with Homography

度値を正面画像の対応点の明度値に合わせるような補正を行えばよい. 本システムでは以下のガンマ変換の式を用いて補正を行う.

$$C_f = 255 \left(\frac{C_l}{255} \right)^{\frac{1}{\gamma}} \quad (1)$$

ただし C_l は左画像の対応点の明度値の平均, C_f は正面画像の対応点の明度値の平均を表す. (1) 式において未知数である γ の値を求め, 左画像をガンマ変換することで, 左画像の全体的な明度を正面画像の全体的な明度に合わせることができる. ガンマ変換では全体的な明度を高く, あるいは低くできるが, ホワイトバランスが異なることによって全体的に赤みがかっていたり, 青みがかっていたりする画像を修正することはできない. しかし, 本システムは3人のユーザが同じ対象シーンを見ているという状況を想定している. よって, 1人は屋内で屋内用のホワイトバランス, 他の2人は屋外で屋外用のホワイトバランス, といったことは考えにくく, 3台のカメラのホワイトバランスは意図的に変えない限り同一のものになると考えられる. また, カメラの撮影範囲に光沢のある物体があり, それぞれのカメラで異なるハイライトが撮影されるなど, 画像の一部の明度が極端に異なってしまうような場合には補正がうまくいかないと考えられる.

図6は実際に明度値の補正を行った結果である. 左画像が正面画像に合わせて全体的に明度が高くなっていることが分かる.

2.5 メディアン処理

障害物の除去にはメディアン処理を用いる. 本手法で用いるメディアン処理とは, 全ての画素について対応する正面画像, 左正面画像, 右正面画像の画素値の中間値を取っていくという処理である. なぜメディアン処理によって障害物が除去できるのかについて以下に述べる.

メディアン処理は, 画像座標系での画素の位置によっ



図6 ガンマ変換による明度値の補正
Fig.6 Correction of pixel values by gamma transform

て4つのケースに分類できる．なお，画像座標 (i, j) における正面画像の画素値を p_f ，左正面画像の画素値を p_l ，右正面画像の画素値を p_r とする．図7に例として4つ目のケースにおける画像を示す．

1つ目のケースが，3枚の画像全てにおいて対応する画素が対象シーンの領域にある場合である．この場合は，すべての画像の画素値が対象シーンのものとなるので，どの画像の画素値を選んでも構わない．実際には光源の方向などにより，3枚の画像の画素値は微妙に異なる場合が多い．例えば p_l が p_f よりも黒みがかっており， p_r が p_f よりも白みがかった場合は，メディアン処理により p_f が選ばれる．

2つ目のケースが，画素の位置が左正面画像において障害物領域に入ってしまう場合である．この場合は， $p_l < p_f \leq p_r, p_l < p_r < p_f, p_f \leq p_r < p_l, p_r < p_f < p_l$ のいずれか，すなわち p_f または p_r が選ばれるが，どちらも対象シーンの画素値である．

3つ目のケースが，画素の位置が右正面画像において障害物領域に入ってしまう場合である．この場合は， $p_r < p_f \leq p_l, p_r < p_l < p_f, p_f \leq p_l < p_r, p_l < p_f < p_r$ のいずれか，すなわち p_f または p_l が選ばれるが，どちらも対象シーンの画素値である．

4つ目のケースが，画素の位置が正面画像において障害物領域に入ってしまう場合である．この場合は， $p_f < p_l \leq p_r, p_f < p_r < p_l, p_l \leq p_r < p_f, p_r < p_l < p_f$ のいずれか，すなわち p_l または p_r が選ばれるが，どちらも対象シーンの画素値である．

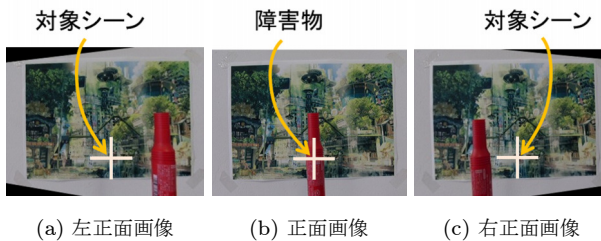


図7 4つ目のケース
Fig.7 Case 4

ただし，メディアン処理を用いても対象シーンの画素値を得られない場合がある．本手法ではRチャンネル

ル，Gチャンネル，Bチャンネルそれぞれについてメディアン処理を行っている．ここで，ある画素の画素値を(Rチャンネルの値,Gチャンネルの値,Bチャンネルの値)と表記する．例えば，正面画像の障害物領域のある画素の画素値が， $(200,101,200)$ ，対応する左正面画像の画素値が $(100,100,100)$ ，対応する右正面画像の画素値が $(102,102,102)$ とする．ただし，対応する左正面画像と右正面画像の画素値は対象シーンのものであるが，光源の方向などにより画素値が一致していないとする．このとき，出力の画素値はメディアン処理により， $(102,101,102)$ となる．R,Bチャンネルは対象シーンから得られるが，Gチャンネルは障害物から得てしまう．しかし，左正面画像，右正面画像の画素値が対象シーンのものであれば，たとえ障害物領域の画素値が選ばれたとしても，その値は左正面画像と右正面画像の画素値の間の値であり，対象シーンの値とほぼ同じになる．

このように，画素の位置によってメディアン処理は4つのケースに分類できるが，どのケースにおいても対象シーンの画素値を取得でき，また対象シーンの画素値を取得できない場合でもそれに近い値を得ることができるので，結果として障害物が除去された画像を生成できるのである．

しかし，障害物が横に長く，2枚以上の画像で障害物領域が重なってしまうと，中間の画素値は障害物のもになってしまう．つまり，障害物領域が重なっている範囲は障害物が残ってしまう．障害物が残ってしまう場合は，図8のように，ユーザがスマートフォン上の障害物領域をタッチすると，他のユーザに Homography によって変換された対応する領域が表示され，その領域に対象シーンが映るようにスマートフォンを動かしてもらい，といった対処方法が考えられる．



図8 タッチ操作による障害物領域の表示
Fig.8 Displaying an obstacle area by touching

2.6 出力画像の改善

左正面画像，右正面画像を作成する際，元画像に映っていない範囲はプログラムによって補間される．本システムでは，図9のように，元画像に映っていない範囲は黒で補間されている．この範囲は障害物領域と同じような影響があるため，他の画像の障害物領域

もしくは黒で補間された領域と重なってしまうと、対象シーンの画素値を得ることができない。本システムでは黒で補間された領域を正面画像の画素値で上書きすることで、出力がより自然になるよう改善している。

具体的な手法について述べる。左画像を左正面画像に変換する Homography を H_l 、右画像を右正面画像に変換する Homography を H_r とする。メディアン処理によって作成された画像に H_l^{-1} 、すなわち H_l の逆行列をかけると、それぞれの画素が左画像のどの座標に対応するか分かる。この時、対応座標が左画像の範囲外であった場合、その画素は左正面画像において黒で補間された領域であると判断できるので、正面画像の画素値で上書きする。例えば左画像が 320×240 画素の場合、メディアン処理によって作成された画像に H_l^{-1} をかけて得た左画像における対応座標を (x, y) とすると、 $x < 0$ または $x > 319$ または $y < 0$ または $y > 239$ の場合に正面画像の画素値で上書きする。同じように H_r^{-1} を用いて右正面画像の黒で補間された領域も正面画像の画素値で上書きする。

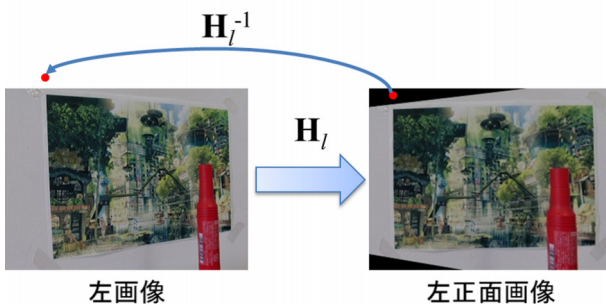


図9 左画像と左正面画像
Fig.9 Left and left-front images

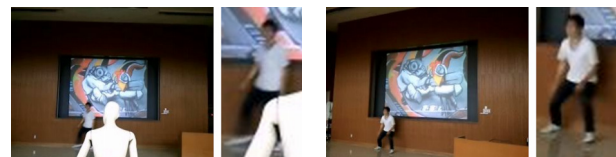
3 実時間処理

本章では実時間で処理を行うための手法について述べる。スマートフォンで取得された画像は一旦サーバに送られる。サーバでは自然特徴抽出や Homography 算出、画素値算出といった障害物除去処理が実行される。そして障害物が除去された画像がスマートフォンに送り返され、スマートフォン上には障害物が除去されたユーザ視点の画像が表示されるのである。

あらかじめ 352×288 画素の正面画像、左画像、右画像を用意しておき、同じアルゴリズムでスマートフォン上 (OS:Android 2.3.3, CPU:Qualcomm MSM6290 1GHz, RAM:512MB) とサーバ上 (OS:Windows 7 32bit, CPU:Intel Core i7-2600 3.40GHz, RAM:3.49GB, 実装環境:Microsoft Visual C++ 2010) それぞれで障害物除去処理を行った結果、スマートフォン上では 1830msec かかったのに対し、サーバ上では 60msec と、サーバ

の方が処理速度が約 30 倍となった。よってサーバ上で障害物除去処理を行うことは実時間で処理を行うことに関して有効であると言える。

また、今回使用したサーバの CPU は 4 コア 8 スレッドのマルチ CPU であるので、サーバ上では 1 つのスマートフォンに対して画像の送受信を行うスレッドを 1 つずつ (3 台のスマートフォンを使う場合は計 3 スレッド) 作成し、障害物除去処理を行うスレッドを別に 1 つ作成してマルチスレッド処理を行うことで、処理速度の向上を図っている。ただし、各スマートフォン及び各スレッドの同期処理は行っていない。そのため、スマートフォンで取得する画像が激しく変化する場合、図 10 のように、同フレームにおいて使用される、各スマートフォンから取得した画像の間に時間的なずれが生じてしまう。



(a) 正面画像 (b) 右画像

図 10 同フレームにおける正面画像と右画像
Fig.10 Front and right images at the same frame

4 実験及び考察

本章では、以下の 3 つの項目についての実験の結果及び考察を述べる。

実験 1 対象シーンが平面の場合に障害物が除去できるか

実験 2 対象シーンが立体の場合にどれだけ離れれば平面に近似できるか

実験 3 自然特徴抽出手法によって結果がどのように変わるか

すべての実験において、スマートフォンで取得する画像のサイズは 320×240 画素である。使用したスマートフォンのスペックは、OS:Android 2.3.3, CPU:Qualcomm MSM6290 1GHz, RAM:512MB である。使用したサーバのスペックは、OS:Windows 7 64bit, CPU:Intel Core i7-2860QM 2.50GHz, RAM:16GB, 実装環境:Microsoft Visual C++ 2010 である。

4.1 実験 1：対象が平面の場合の障害物除去実験

実験 1 では、対象シーンが平面の場合に、本手法を用いて障害物が除去できるかどうかを確認する。対象シーンは長方形の紙に描かれた絵であり、対象の約 20cm 前に紙製の黒い筒を置いた。スマートフォンカメラのホワイトバランスは「蛍光灯の下」モードである。

各スマートフォンから取得した画像及び出力画像を図 11 に示す。

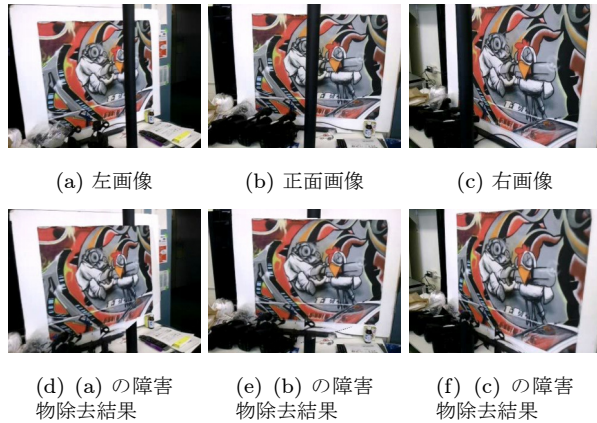


図 11 障害物除去実験の結果

Fig. 11 Result of the experiment to remove an obstacle

図 11 に示されるように、対象シーンが平面の場合、本手法を用いることで、各スマートフォンから取得した画像中の障害物を除去できることが分かる。処理速度は処理全体で約 5-6fps であったので、ある画像を取得してからスマートフォンに出力画像を表示するまでに約 150-200msec かかっている。よって約 150-200msec の間に対象シーンや障害物が大きく変化すると、実環境の状況とスマートフォン上の画像が時間的に少しずつれていると感じてしまうと考えられる。ただし対象シーンや障害物の変化が緩やかであれば、実環境の状況とスマートフォン上の画像のずれは少なく、実時間性を持っていると言える。

処理時間の大半は通信に要している。本手法ではブロッキング通信を行っている。ブロッキング通信とは、データの送受信を行う際に、送受信の完了を待ってから他の処理を開始する通信方法である。本手法では、スマートフォンが画像を送信する処理に到達してからサーバが画像を受信する処理に到達するまでの間と、サーバが画像を送信する処理に到達してからスマートフォンが画像を受信する処理に到達するまでの間でブロッキングが行われている。スマートフォンが画像を送信する処理は、サーバが画像を受信する処理の後に到達する場合が多く、スマートフォンが画像を受信する処理は、サーバが画像を送信する処理の前に到達する場合が多い。つまりスマートフォン側から見た場合、スマートフォンが画像を送信する処理に到達してからサーバが画像を受信する処理に到達するまでの間のブロッキングが発生することは少なく、サーバが画像を送信する処理に到達してからスマートフォンが画像を受信する処理に到達するまでの間のブロッキングが発生することは多い。このときのブロッキング時間は約

100msec 前後である。よってこのブロッキングを短くすることが出来れば処理速度の向上が望める。なお、本システムでは無線 LAN 規格に IEEE 802.11n を使用しており、320×240 画素の画像であれば送受信自体は数 msec しかかからないので、無線通信の速度自体を速くしても処理時間の向上はあまり望めない。ただし、画像のサイズが大きくなればそれだけ送受信にかかる時間もかかるので、無線通信の速度も考慮に入れる必要があると考えられる。

4.2 実験 2: 対象シーンが立体の場合に平面と近似できる距離の調査

実験 2 では、対象シーンを立体とし、ユーザを少しずつ対象シーンから離していき、どの程度離れば対象シーンを平面と近似できるかを確かめる。図 12 に実験環境を示す。

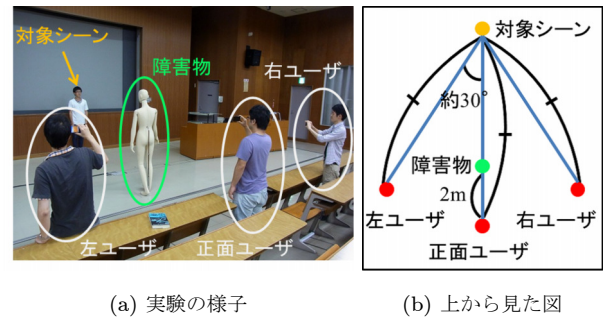


図 12 実験環境

Fig. 12 Experimental environment

対象シーンを平面と近似できる距離は、対象シーンの奥行き深さによって異なると考えられる。本実験では、対象シーンを人(奥行き約 20cm)とした。障害物はウレタンを布で覆い作られたマネキンである。障害物は正面ユーザから 2m の位置に配置する。正面ユーザと対象シーンを結んだ直線及び左ユーザと対象シーンを結んだ直線、正面ユーザと対象シーンを結んだ直線及び右ユーザと対象シーンを結んだ直線それぞれ、なす角度が約 30° となり、かつ正面ユーザと対象シーン間の距離と同じ距離だけ対象シーンから離れた位置に左右ユーザを配置する。対象シーンの後ろのスクリーンに、特徴のとりやすい絵を表示し、Homography が算出されやすいようにしている。スマートフォンカメラのホワイトバランスは「蛍光灯の下」モードである。

各ユーザが対象シーンから 5m, 10m, 15m 離れた位置での正面画像と出力画像を図 13 に示す。

図 13 に示されるように、5m, 10m 離れた位置では対象シーンが 2 重に見える。これは、対象シーンが平面と近似できておらず、左正面画像、右正面画像を作る際に、対象シーンの位置が正面から見た対象シーンの位置からずれてしまっているためと考えられる。15m 離れた位置では対象シーンが 2 重になっていない



図 13 対象シーンと正面ユーザ間の距離を変えた結果

Fig.13 Result of changing the distance between the target scene and the front user

い。よって本実験の環境では、奥行きが約 20cm の対象シーンに対し、各ユーザが対象シーンから 15m 程度離れていれば、対象シーンを平面と近似できる。ただし、正面ユーザと対象シーンを結んだ直線及び左右ユーザと対象シーンを結んだ直線がなす角度が大きくなる(図 12(b)の「約 30°」と書いてある箇所が 60°などとなる)と、左ユーザは対象シーンの左側面の画素値を、右ユーザは対象シーンの右側面の画素値を取得するため、正面ユーザが取得する出力画像中の対象シーンは正面から見たものと大きく異なってしまうと考えられる。

4.3 実験 3: 自然特徴抽出手法の比較

実験 3 では、自然特徴抽出手法によって本手法の結果がどのように変わるかを調べた。本手法では、Homography を精度よく算出できれば障害物が除去できる。つまり、Homography を算出する過程が障害物除去の精度を左右する最も重要な部分である。Homography 算出に最も影響を及ぼすのが自然特徴抽出手法なので、自然特徴抽出手法を比較し、BRISK が本手法に適しているかを調べる。

比較対象には、現在存在する自然特徴抽出手法の中で最も頑健なもの 1 つと言われている SIFT(Scale-Invariant Feature Transform)[17] と、SIFT を高速化させた SURF(Speeded Up Robust Features)[18] を選んだ。実験 2 と同じ環境で、対象シーンと正面ユーザ間の距離が 4m, 10m, 15m の場合に、各ユーザの入力画像のある 100 フレームを抜き出し、BRISK, SURF,

SIFT それぞれを適用し出力画像を作成する。同じ条件で比較するため、それぞれの特徴抽出手法において、クエリディスクリプタと訓練ディスクリプタの類似度の高い上位 50 組の対応点を用いて Homography を算出している。作成した出力画像を見て、障害物の裏側の画素値が取得できていれば成功、取得できていなければ失敗とし、成功の割合を計算した。具体的な成功例、失敗例を図 14 に示す。



図 14 成功例と失敗例
Fig.14 Examples of success and failure

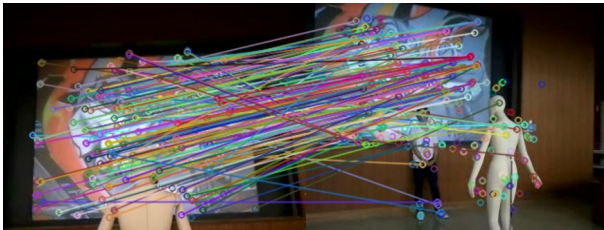
本実験では、対象シーンである人が 2 重になっても、スクリーンに表示された絵が正しく表示されていれば Homography が正しく算出されていると考え、成功としている。Homography が算出できず障害物が消せなかった場合や、算出した Homography のパラメータの値が正しくないために、障害物の一部が対象シーン領域に残ったり対象シーンが歪んでしまった場合を失敗としている。また、正面画像の特徴点を検出し、特徴量を記述した時の処理時間も計測した。

各自然特徴抽出手法に対する実験結果を表 1 に示す。また、対象シーンと正面ユーザ間距離が 4m, 15m の場合、SURF 特徴量を用いたマッチング結果を図 15 に示す。

表 1 より、対象シーンとユーザ間の距離が離れるほど、Homography 算出成功率は下がることが分かる。図 15 に示されるように、スクリーン上に表示された絵同士でしか正しいマッチングが行われていない。また、距離が離れるほど画像に占めるスクリーンの割合が小さくなってしまふ。結果として、マッチング全体に対する正しいマッチングの数が少なくなってしまう、Homography の算出に失敗することが多くなってしまうと考えられる。対象シーンから 4m, 10m, 15m の位置における、画像に占めるスクリーンの割合を算出すると、それぞれ約 46.1%, 9.6%, 4.3% となった。BRISK を用いた場合、4m の位置ではほぼ全てのフ

表 1 各自然特徴抽出手法における Homography 算出成功率と特徴抽出処理時間
Table 1 Success rate of computing Homography and processing time of extracting natural features for each method

対象-ユーザ間距離	BRISK		SURF		SIFT	
	成功率	処理時間	成功率	処理時間	成功率	処理時間
4m	97%	2.9ms	100%	48.9ms	100%	168.7ms
10m	75%	3.0ms	99%	35.6ms	98%	99.4ms
15m	63%	2.7ms	88%	33.4ms	89%	84.9ms



(a) 対象シーンから 4m の位置のマッチング結果



(b) 対象シーンから 15m の位置のマッチング結果

図 15 SURF 特徴量を用いたマッチング結果
Fig.15 Result of matching with SURF features

フレームで Homography の算出に成功している。よって、対象シーンを平面と近似したときに、その近似した平面において、正しいマッチングの取れる領域が画像全体の約半分を占めれば、BRISK を用いて精度よく正しい Homography を算出することができる。正しいマッチングの取れる領域が画像全体の 10% 以下のときに BRISK を用いると、全体の 1/4 のフレームは Homography の算出に失敗してしまう。このような状況で精度よく正しい Homography を算出するには、BRISK の代わりに SURF や SIFT を使う必要がある。ただし、処理時間は SURF、SIFT それぞれ BRISK の約 12~17 倍、32~58 倍と、大幅に遅くなってしまふ。例えば、対象シーンから 10m の位置では、1 枚の画像の特徴抽出に対して、SURF は BRISK よりも処理時間が 32.6ms 多くかかる。出力画像の更新を行うには、3 台のスマートフォンから得られた 3 枚の画像の特徴抽出を行うので、SURF を用いると出力画像の更新 1 回につき、特徴抽出処理にかかる時間が約 100ms だけ BRISK よりも多くかかってしまうことになる。よって、SURF 及び SIFT を用いると、本システムにリア

ルタイム性を持たせることは難しいと考えられる。

5 おわりに

本論文では、対象シーンが平面と仮定できる場合に、自然特徴点マッチングから Homography を算出しメディア処理を施すことで、固定マーカの貼付などの事前準備を必要とせずに隠消現実感を行う手法を提案した。画像を取得するデバイスにスマートフォンを使用し、障害物除去処理を行うサーバと無線通信で画像の受け渡しを行うことで、実時間性を持たせつつ、ユーザの自由な移動を可能にした。

今後の課題としては、より多くのスマートフォンを用いた場合の本手法の最適化ということが挙げられる。本研究では 3 台のスマートフォンを用いる場合に最適化されたアルゴリズムを使用しているが、より多くのスマートフォンを用いることで障害物に隠れた領域の画素値を取得できる可能性が増える。つまり、障害物をより安定に除去できることが期待される。ただし扱うスマートフォンの数が増えるほど処理速度が落ちると考えられ、またどのスマートフォンのカメラで取得した画像を用いればよいか判断するアルゴリズムも必要となる。また、Microsoft 社が開発した Kinect と呼ばれる、RGB 画像とデプス画像を同時に取得できるカメラを用いて対象シーンを 3 次元的に復元することで、対象シーンが平面と仮定できない場合でも隠消現実感を実現できるように処理を改良する予定である。

謝辞

本研究におけるスマートフォンとサーバ間の通信の実装に、慶應義塾大学大学院理工学研究科修士課程（現在、ドワンゴ（株））井上卓也氏の協力を得た。

参考文献

- [1] Van Krevelen, D. and Poelman, R. (2010), "A survey of augmented reality technologies, applications and limitations", International Journal of Virtual Reality, vol.9, pp. 1-20.
- [2] Martedi, S. and Saito, H. (2011), "Augmented Fly-through using Shared Geographical Data", International Conference on Artificial Reality and Telexistence, pp. 47-52.

(2012 年 2 月 27 日)

- [3] Sheng, Y. Yapo, T. Young, C. and Cutler, B. (2011), "A spatially augmented reality sketching interface for architectural daylighting design", IEEE Transactions on Visualization and Computer Graphics, vol.17, pp. 38–50.
- [4] 森, 一刈, 柴田, 木村, 田村 (2011), "隠消現実感の技術的枠組と諸問題～現実世界に実在する物体を視覚的に隠蔽・消去・透視する技術について～", 日本バーチャルリアリティ学会論文誌, Vol.16, No.2, pp. 239–250
- [5] Goto, M. Uematsu, Y. Saito, H. Senda, S. and Ike-tani, A. (2010), "Task support system by displaying instructional video onto AR workspace", International Symposium on Mixed and Augmented Reality, pp. 83–90.
- [6] Mann, S. and Fung, J. (2001), "Videoorbits on eye tap devices for deliberately diminished reality or altering the visual perception of rigid planar patches of a real world scene", International Symposium on Mixed Reality, pp. 48–55.
- [7] Wang, J. and Adelson, E. (1994), "Representing moving images with layers", IEEE Transactions on Image Processing, vol.3, pp. 625–638.
- [8] Lepetit, V. and Berger, M. (2000), "A semi-automatic method for resolving occlusion in augmented reality", Computer Vision and Pattern Recognition, vol.2, pp. 225–230.
- [9] Zokai, S. Esteve, J. Genc, Y. and Navab, N. (2003), "Multiview paraperspective projection model for diminished reality", International Symposium on Mixed and Augmented Reality, pp. 217–226.
- [10] 川上, 稲見, 柳田, 前田, 館 (1998), "現実感融合の研究 (第 2 報) - Reality Fusion における光学迷彩技術の提案と実装-", 日本バーチャルリアリティ学会第 3 回大会論文集, pp. 285–286.
- [11] Herling, J. and Broll, W. (2010), "Advanced self-contained object removal for realizing real-time Diminished Reality in unconstrained environments", International Symposium on Mixed and Augmented Reality, pp. 207–212.
- [12] Enomoto, A. and Saito, H. (2007), "Diminished Reality using Multiple Handheld Cameras", Asian Conference on Computer Vision Workshop on Multi-dimensional and Multi-view Image Processing, pp. 130–135.
- [13] Leutenegger, S. Chli, M. and Siegwart, R. (2011), "BRISK: Binary Robust Invariant Scalable Keypoints", International Conference on Computer Vision, pp. 2548–2555.
- [14] Open Computer Vision Library, <http://sourceforge.net/projects/opencvlibrary/>
- [15] Fischler, M.A. and Bolles, R.C. (1981), "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", Communications of the ACM, vol.24, pp. 381–395.
- [16] Hartley R. and Zisserman A. (2003), "Multiple view geometry in computer vision 2nd edition", Cambridge University Press, pp. 88–93.
- [17] Lowe, D. (1999), "Object Recognition from Local Scale-Invariant Features", International Conference on Computer Vision, pp. 1150–1157.
- [18] Bay, H. Tuytelaars, T. and Van Gool, L. (2006), "SURF: Speeded Up Robust Features", European Conference on Computer Vision, pp. 404–417.

[著者紹介]

本田 俊博



2012 年慶應義塾大学理工学部情報工学科卒。現在、同大学大学院修士課程在学中。多視点カメラの協調利用による隠消現実感表示システムの研究に従事。

斎藤 英雄 (正会員)



1987 年慶應義塾大学理工学部電気工学科卒業。1992 年同大学院理工学研究科博士課程電気工学専攻修了。同年同大学理工学部助手、2006 年より同大学理工学部情報工学科教授。1997 年から 99 年までカーネギーメロン大学ロボティクス研究所訪問研究員。コンピュータビジョンや、それを仮想現実感や 3 次元映像メディア処理などに応用する研究に従事。