

環境3Dモデルに基づくスマートフォン上での実時間隠消現実感

本田 俊博^{*1} 齋藤 英雄^{*1}

Diminished Reality on a Smartphone Based on a 3D Environment Model

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

Abstract – 本論文では、実環境中の物体を視覚的に取り除く技術である隠消現実感をスマートフォン上で実現するシステムについて述べる。事前に環境3次元モデルを作成し、視点生成型学習によって特徴量データベースを構築する。隠消現実感処理を実行する際には、スマートフォン画像とデータベースの特徴量を比較し、対応付け、スマートフォンとモデルの相対位置を算出する。そしてユーザの指定した領域を非表示にしてスマートフォン画像上にレンダリングする。主な処理をサーバで行うことで、実時間処理を可能にした。実験の結果、対象シーンが3次元形状であっても、約8fpsでスマートフォン上に隠消現実感を実現できた。

Keywords : 隠消現実感, 実時間, スマートフォン, RGB-D カメラ, 視点生成型学習

1 はじめに

近年、コンピュータビジョンの分野の中で隠消現実感 (Diminished Reality) と呼ばれる技術が注目されている [1]。拡張現実感 (Augmented Reality) が実環境中に CG 物体を重畳し情報を付加するのに対し、隠消現実感とは実環境中の物体を視覚的に取り除くという点が異なる。隠消現実感を実行する環境には様々な条件が考えられ、どのような条件にも適切に対応できる万能な手法は存在しないと言ってもよく、隠消現実感における個々の研究は特定の条件に焦点を当てている。例えば、ユーザ視点カメラは静的か動的か、除去対象は静的か動的か、ユーザの見たシーン、すなわち除去対象の背後 (隠背景と呼ぶ) は静的か動的か、隠背景は平面か3次元形状か、実時間で動くか動かないかなどといった条件の組み合わせが考えられる。それぞれの条件を明確にした上で、隠消現実感の手法をいくつか述べる。

Zokai らは、離れた位置から撮られた異なる3枚の静止画像を用いて、適切な背景を生成する手法を提案した [2]。この手法は、ユーザ視点カメラ、除去対象、隠背景は静的でなければならない。隠背景は3次元形状でも対応できるが、隠背景を細かく復元するほど処理時間がかかってしまう。また、障害物の境界をユーザが指定する必要がある。

Shen らは、1台のカメラで取得した画像列を用い、除去対象が移動していることを利用して隠背景を重畳する手法を提案した [3]。この手法は、ユーザ視点カメラはパン、チルト、ズームのみ可能で、除去対象は動

的であることが必須である。隠背景は動的でも対応できるが、平面又は平面に近似できるものでなければならない。オフラインでの処理を想定されているため、処理時間については言及されていない。また、最初のフレームで隠背景領域をおおまかに指定しておく必要がある。

我々は以前に3台のスマートフォンを用い、互いに不足した隠背景情報を補完し合うことで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手法を提案した [4]。この手法は、各スマートフォン、除去対象、隠背景が動的でも対応でき、実時間で動作するが、隠背景は平面又は平面に近似できるものでなければならなかった。

上記に対して、本論文では、平面近似できないような3次元形状を持つ隠背景を対象に、ユーザ視点カメラ (スマートフォン) が自由に移動可能な状況で実時間で動作するような隠消現実感を実現する手法を提案する。本システムは除去対象、隠背景共に静的の場合を想定して構成されている。

まず事前に除去対象と隠背景及びその周囲の3次元モデルをRGB-Dカメラを用いて作成する。このモデルを基に視点生成型学習 (Viewpoint Generative Learning) [5] を行い、特徴量のデータベースを構築する。隠消現実感を実行する際にはスマートフォン画像とデータベースの特徴量を比較し、対応付け、スマートフォンとモデルの相対位置を算出する。そしてユーザがスマートフォンをタッチした位置 (除去対象とみなす) に対応するモデル中のメッシュを非表示にし、その奥のメッシュをスマートフォン画像上に重畳することで、隠消現実感を実現する。

^{*1}慶應義塾大学

^{*1}Keio University

2 隠消現実感処理

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1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．また，サーバには事前に取得しておいた環境3次元モデルとその特徴量データベースが保存されている．Kinect (RGB-D カメラ) は環境3次元モデルの取得時のみ使用し，隠消現実感処理実行時は設置する必要はない．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．そしてサーバに保存されているモデルとデータベースを用いて本手法の処理を施すことで，除去対象が除去された出力画像を得ることができる．

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．本システムはモデルとデータベースを作成するオフライン処理と，実際に隠消現実感処理を行うオンライン処理に分かれている．オフライン処理では，まず Kinect を用いて KinectFusion[6] を使用し，テクスチャ付き環境3次元モデルを作成する．その後，このモデルを多視点でレンダリングし，各視点で自然特徴を抽出，クラスタリングする視点生成型学習によって特徴量データベースを作成する．

オンライン処理では，まずスマートフォン画像の自然特徴を抽出し，データベースと特徴点マッチングを行い，スマートフォン画像座標とモデル座標の2D-3D対応を得る．この対応を用いることで，モデル座標系をスマートフォンカメラ座標系に変換する行列 $[R|t]$ を算出できる． $[R|t]$ を用いると，スマートフォン視点でモデルを扱うことができる．

ユーザがスマートフォン上で除去対象をタッチすることにより，除去を行う領域（除去対象領域）が指定される．除去対象領域のモデルのメッシュを非表示にしてスマートフォン視点でモデルをレンダリングすることで，除去対象領域の隠背景を取得できる．この隠背景をスマートフォン画像の除去対象領域上に重ねることで隠消現実感を実現する．さらにこの時，モデルから作成した隠背景と元のスマートフォン入力画像を比較し，除去対象が物理的に除去されたと判断した場合には隠消現実感処理を中止し，入力画像をそのまま出力画像としてスマートフォン画面に表示する．なお，スマートフォン側は入力画像，タッチ座標，タッチ時の圧力の送信と出力画像の受信のみを行い，隠消現実感処理はサーバで行っている．

2.2 環境3次元モデル作成

本システムでは，Point Cloud Library[7] によって提供されている KinectFusion という手法を用いて環

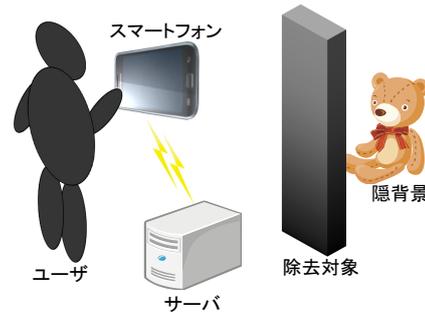


図1 システムの概要
 Fig.1 Overview of the system

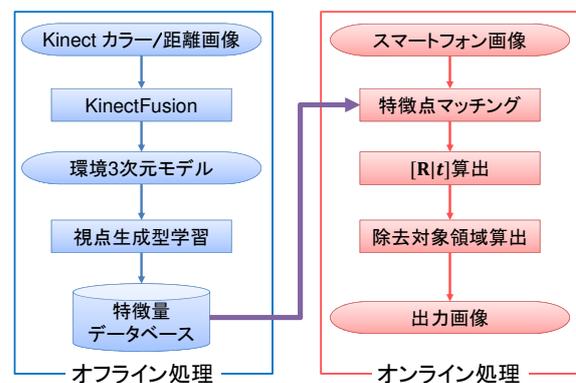


図2 本手法の流れ
 Fig.2 Flow of the proposed method

境3次元モデルを作成している．KinectFusion とは，Kinect で毎フレーム距離画像を取得し，Iterative Closest Point アルゴリズム [8] を用いてレジストレーションを行い，Truncated Signed Distance Function[9] を用いて3次元再構成を行うという手法である．3次元再構成と並行してカラー画像の取得とその時のカメラ位置姿勢情報を保存しておくことで，テクスチャ付きの3次元モデルを作成することができる．

2.3 視点生成型学習

スマートフォンが自由に移動しても安定に位置姿勢推定を行うため，環境3次元モデルを入力とする視点生成型学習を行う．初めに，モデルを多視点でレンダリングし，GPU-SIFT[10] を用いて各視点における自然特徴点の検出及び SIFT 特徴量の抽出を行う．GPU-SIFT は，頑健な特徴抽出手法である SIFT の処理を，GPU を用いて高速化したものである．本システムではモデルの正面（モデル座標系の原点から Z 軸方向）を基準に，左右は -90° から 90° まで 15° 刻みで，上下は -15° から 90° （真上）まで 15° 刻みでレンダリングしている（図3）．ただし，真上の場合は左右 0° の1回のみレンダリングしているため，合計92視点で特徴抽出を行っている．抽出された特徴量を全て保存するとデータベースが膨大になり，誤対応も増えると考えられるため，以下の手法を用いてデータベースを圧縮する．まず，各視点の特徴点座標をモ

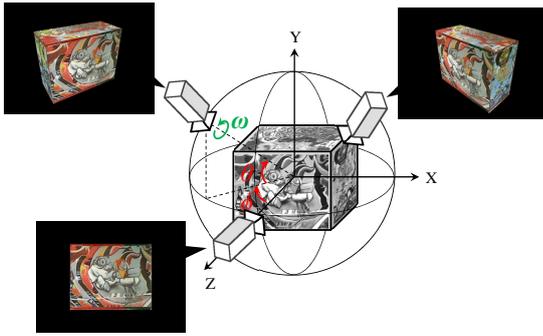


図3 多視点でのモデルのレンダリング [5]
Fig.3 Rendering model from many view-points

デル座標系に逆投影していく。このとき、すでに逆投影された特徴点から一定距離内に逆投影された場合はその特徴点の repeatability スコアを 1 増やし、逆投影された特徴点の特徴量を保存する。つまり、1 つの特徴点に対し、複数の視点での特徴量を保存しておく。一定距離内に登録済みの特徴点がない場合は、新しい特徴点として登録する。repeatability スコアが高いものを stable keypoint と呼ぶ。stable keypoint は多くの視点で観測できる特徴点ということを表す。本システムでは repeatability スコアの高い上位 2000 個の特徴点を stable keypoint として使用している。各 stable keypoint の特徴量を k-means++[11] によってクラスタリングし、特徴量データベースに保存する。本システムでは各 stable keypoint の特徴量を 8 つにクラスタリングしている。レンダリングの間隔、stable keypoint の数、クラスタリングの数は安定してカメラ位置姿勢推定が行えるように実験により決定した。データベースには、stable keypoint のモデル座標系の 3 次元座標と、クラスタリングによって算出された 8 つの特徴量が保存されている。本システムの視点生成型学習では、1 つのモデルしか使用していないため、光沢のある物体のような、視点によって見え方が変わる環境ではうまくカメラ位置姿勢推定ができない可能性がある。

2.4 モデル座標系からスマートフォンカメラ座標系への変換

モデルをスマートフォン視点で扱うために、モデル座標系をスマートフォンカメラ座標系に変換する 3×4 行列 $[R|t]$ が必要となる。 $[R|t]$ は以下の式を用いることで求めることができる。

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = A[R|t] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (1)$$

ただし、 A はスマートフォンカメラの内部パラメータを表す 3×3 行列、 (X, Y, Z) はモデル座標、 (u, v) はスマートフォン画像座標を表す。本システムでは A は既知とする。よって、スマートフォン画像座標とモデル座標の 2D-3D 対応を取得することで、 $[R|t]$ を算出できる。対応点を取得するために、GPU-SIFT を用いて毎フレームスマートフォン画像の SIFT 特徴量を抽出する。そして、データベースに保存された特徴量と距離計算を行う。最も距離が短い特徴点を対応点とする。6 組以上の対応点が見つければ、(1) 式を用いて Direct Linear Transformation 法 [12] により、 $[R|t]$ の最適なパラメータを求める。この際、誤対応が存在しても頑健に $[R|t]$ を算出できるように、RANSAC(Random Sample Consensus)[13] を用いている。この対応点からパラメータを求める処理は、OpenCV[14] の関数を用いている。

2.5 除去対象領域算出

どこを除去対象とみなし、隠背景を重畳するか決める必要がある。本システムでは、ユーザがスマートフォンをタッチした場所を除去対象としている。まず、何も除去していない状態のモデルを OpenGL[15] 及び 2.4 節で述べた $[R|t]$ を用いてスマートフォン視点でレンダリングする。OpenGL は、z バッファ(カメラと物体までの距離を記憶するメモリ領域)を用いてモデルを描画しており、レンダリングされた画像のある座標 (u, v) を指定すると、対応するモデルの z 座標 d を取得できる。 u, v, d 及びスマートフォンカメラの内部パラメータを用いることで、 (u, v) をスマートフォンカメラ座標系の 3 次元座標に変換することができる。スマートフォンカメラ座標系原点から、ユーザがタッチした画像座標に対応するスマートフォンカメラ座標系の 3 次元点へ向かう視線ベクトルを算出する。本システムで扱うモデルは三角メッシュで構成されており、交差判定法 [16] を用いて各メッシュと視線ベクトルの交差判定を行う。交差判定法は 3 次元空間において、視点からある方向に向かう視線ベクトルと三角形が交差するかどうかを判定し、交差した場合は視点から交点までの距離を算出することができる。交点が 2 つ以上あった場合に、最も手前の交点から距離 d_{th} 以内のメッシュを除去対象とみなし、非表示にする。 d_{th} はユーザがスマートフォンをタッチした時の圧力によって、0 cm ~ 10 cm まで変化する。つまり、弱くタッチすれば細かく除去対象を指定でき、強くタッチすれば広い範囲を除去対象として指定できる。交点が 1 つの場合は、隠背景が存在しないと考えられるので、除去対象領域算出は行わない。

2.6 隠背景の重畳

除去対象領域算出後，除去対象を除去したモデルをレンダリングした画像 I_1 と，除去対象のメッシュのみからなるモデルをレンダリングした画像 I_2 を作成する．OpenGL でレンダリングされた画像の画素は，RGB チャンネルとアルファチャンネルからなる．アルファチャンネルはモデルがレンダリングされていなければ 0，レンダリングされていれば 0 以外の値が入っている．よって， I_2 のアルファチャンネルが 0 以外の画素（除去対象領域）に対応するスマートフォン入力画像の画素に，同じく対応する I_1 の画素値を上書きしていけば，除去対象領域に隠背景を重畳した出力画像を得ることができる．

3 スマートフォンとサーバ間の画像送受信

図 4 にスマートフォンとサーバの処理の流れを示す．サーバでは，スマートフォンからの画像受信及びスマートフォンへの画像送信を行うスレッドと，隠消現実感処理を行うスレッドが存在し，それぞれ独立に処理が行われている．サーバの画像送受信スレッドでは，まず画像を受信する関数が呼ばれ，待機状態に入る．スマートフォン側で画像を送信する関数が呼ばれると，画像を受信し，入力画像を更新する．このとき，サーバの隠消現実感処理スレッドが処理中の場合，更新された入力画像を用いるのは次フレームということになる．サーバ側の画像を送信する関数が呼ばれ，かつスマートフォン側の画像を受信する関数が呼ばれていれば，出力画像を送信する．このとき，サーバの隠消現実感処理スレッドが処理中の場合，送信されるのは前フレームの出力画像ということになる．スマートフォンとサーバのそれぞれにおいて，画像を送受信する関数が呼ばれてから実際に送受信が完了するまでの時間の 10 フレーム平均を計測した．スマートフォンのスペックは OS:Android 2.3.3, CPU:MSM8255 1GHz, RAM:512MB, 実装環境:Eclipse 3.7, サーバのスペックは OS:Windows 7 64bit, CPU:Intel Core i7-2860QM 2.50GHz, RAM:16.0GB, GPU: GeForce GTX 560M, 実装環境:Microsoft Visual C++ 2010 である．スマートフォン画像のサイズは 320×240 画素である．結果を表 1 に示す．また，画像データを受け取り始めてから受け取り終わるまでにかかった時間は，スマートフォンが 14.2ms，サーバが 13.8ms であった．よって，図 4 のように，サーバとスマートフォン両方も受信の際に待機時間が発生していることが分かる．これは，本システムではブロッキング通信を行っており，一方の受信関数が呼ばれた時，もう一方の送信関数が呼ばれるまで処理が停止するためである．つまり，本システムではサーバの受信関数はスマートフォンの

送信関数の前に呼ばれ，サーバの送信関数はスマートフォンの受信関数の後に呼ばれることがわかる．

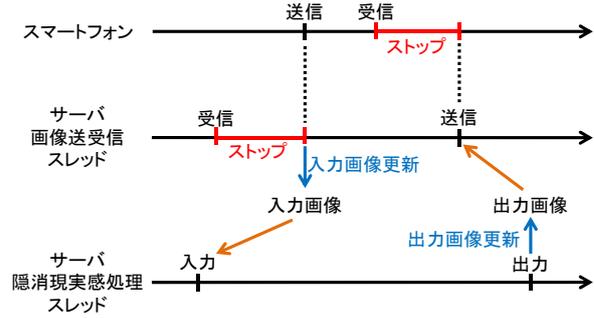


図 4 スマートフォンとサーバの処理の流れ
 Fig. 4 Flow of the processing of the smartphone and the server

表 1 スマートフォンとサーバそれぞれにおける画像送受信にかかった時間 [ms]
 Table 1 Processing time of sending and receiving an image for the smartphone and the server

	受信	送信
スマートフォン	36.3	<1
サーバ	99.1	<1

4 実験及び考察

出力画像と実際の隠背景との類似度，隠消現実感処理を行える範囲，主な処理にかかる時間を調べる．図 5 に実験環境を示す．

隠背景はぬいぐるみ等が置いてあり，複雑な 3 次元形状を有する机の上，除去対象は紙製の箱とした．スマートフォンは除去対象の方向を向くように固定しておき，除去対象がある場合と除去対象を取り除いた場合の 2 種類の環境を撮影する．図 5(b) のように，除去対象を中心とした半径 40 cm, 80 cm, 120 cm の同心円上で 15° 刻みにスマートフォンを動かしていき，除去対象が画面中央に映るように向きを固定し，隠消現実感処理を行える範囲を調べる．本実験では，各スマートフォン位置で条件を同じにするため，タッチによる除去対象領域算出は行わず，除去対象である箱を包含する直方体を除去対象領域としてあらかじめ設定しておく．類似度は，除去対象領域における，出力画像と除去対象を物理的に取り除いた状態の画像（正解画像と呼ぶ）との正規化相互相関 R_{znc} を用いる．

$$R_{znc} = \frac{\sum_{i=1}^{NUM} ((I_i - \bar{I})(T_i - \bar{T}))}{\sqrt{\sum_{i=1}^{NUM} (I_i - \bar{I})^2 \times \sum_{i=1}^{NUM} (T_i - \bar{T})^2}} \quad (2)$$

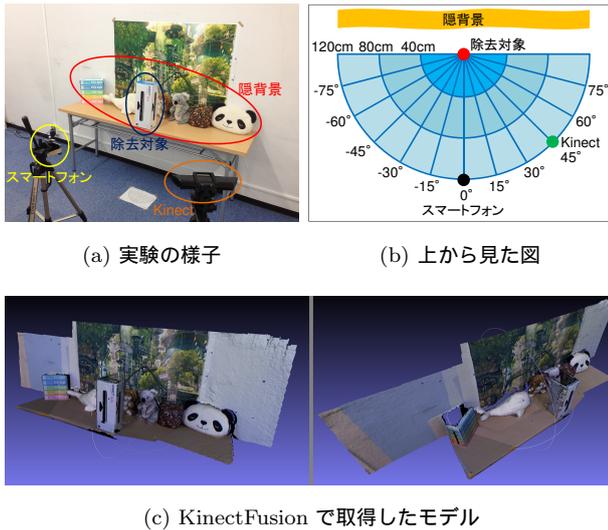


図 5 実験環境
Fig. 5 Experimental environment

ただし、 I_i, T_i はそれぞれ出力画像、正解画像の除去対象領域における画素値、 \bar{I}, \bar{T} はそれぞれ出力画像、正解画像の除去対象領域全体の画素値の平均値、 NUM は除去対象領域の画素数を表す。また $-1 < R_{zncc} < 1$ であり、1 に近づくほど、出力画像と正解画像の類似度が高いことを表す。本システムは光学的整合性に関する処理は行っておらず、幾何学的整合性を重視しているため、全体的な明るさの変動があっても値が大きく変わらない R_{zncc} を類似度として採用する。サーバとスマートフォンのスペックは 3 節で述べたものと同じである。スマートフォン画像のサイズは 320×240 画素である。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。

図 6 に入力画像と出力画像及び正解画像を示す。入力画像は図 5(b) において、上段が $0^\circ, 80\text{cm}$ の位置から、下段が $-30^\circ, 80\text{cm}$ の位置から取得したものである。また、図 7 に R_{zncc} を示す。ある 10 フレームの出力画像において、8 フレーム以上隠消現実感処理が成功したとみなせる場合のみ、成功したフレームにおける R_{zncc} の平均値をプロットしている。寸分変わらずに隠背景を重畳することは非常に難しいため、多少重畳位置がずれていても違和感のない場合は隠消現実感が成功したとしている。具体的な失敗例を図 8 に示す。

図 6 より、隠背景が 3 次元形状でも、スマートフォンの位置に合わせて隠背景を除去対象上に重畳できていることが分かる。除去対象の下部が除去できていないのは、隠背景のモデルが取得できていないためである。

図 7 より、実験を行った全ての位置で隠消現実感処理が成功している。これは、視点生成型学習によって様々な視点での特徴を保持しているためと考えられる。



図 6 入力画像と出力画像及び正解画像
Fig. 6 Input image, output image and the ground truth

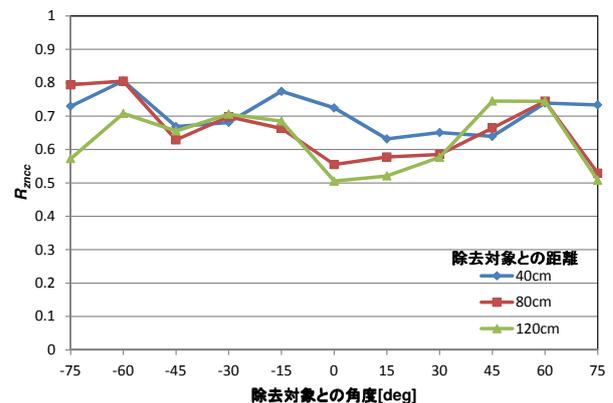


図 7 出力画像と正解画像との正規化相互相関
Fig. 7 Zero-mean normalized cross-correlation between the output image and the ground truth

また、本実験において除去対象との角度が大きい場合に R_{zncc} が高くなるのは、隠背景の一部である壁面が均一な白色であり、位置がずれていても正解画像との画素値の差がほぼないためと考えられる。

表 2 に主な処理にかかった時間の 10 フレーム平均を示す。サーバの隠消現実感処理スレッドの 1 サイクルは約 8fps、画像送受信スレッドの 1 サイクルは約 10fps、スマートフォン画面の更新は約 10fps であった。また、視点生成型学習にかかった時間は約 20 秒であった。スマートフォンカメラで取得した画像が隠消現実感処理を経てスマートフォン画面に表示されるまでの時間（レイテンシ）は、サーバがスマートフォン画像を受信する時間 13.8ms、隠消現実感処理を行う時間 115ms、スマートフォンが出力画像を受信する時間 14.2ms、スマートフォン画面に画像を表示する時間 21ms の合計約 164ms となる。ただし、実際には画像送受信スレッドと隠消現実感処理スレッドは別々



図 8 隠消現実感処理失敗例
Fig. 8 Failure examples

に動いており，入力画像更新直後にその画像を使った隠消現実感処理が始まる，あるいは出力画像更新直後にその画像をスマートフォンに送信するとは限らず，レイテンシはより大きくなると考えられる．

表 2 主な処理にかかった時間 [ms]
Table 2 Main processing time

処理内容	時間
スマートフォン側特徴抽出	23.7
特徴点マッチング	23.9
$[R t]$ 算出	47.1
除去対象領域算出	10.1
隠背景の重畳	10.2
除去対象が動いたかの判定	<1

5 おわりに

本論文では，隠背景が3次元形状であっても，RGB-Dカメラによって環境3次元モデルを取得することで，スマートフォン上で除去対象を除去し隠背景を重畳する隠消現実感を実現する手法を提案した．主な処理をサーバで行うことで，約8fpsで処理が可能となり，実時間性を持たせることができた．

今後の課題としては，隠背景が動的の場合に隠消現実感を実現するということが挙げられる．例えば，2台以上のRGB-Dカメラを用いて高速に鮮明な環境3次元モデルを作成することができれば，隠背景が動的でも，鮮明に隠背景を重畳することが可能であると考えられる．また，本システムでは除去対象領域はカメラ座標系において固定されているため，除去対象を動かしてしまうと除去されなくなるが，セグメンテーションの技術を用いることで，除去対象を動かしても除去対象領域を追従させ除去し続けることができると考えられる．さらにセグメンテーションを行うことで，除去対象領域算出時に一定の範囲ではなく，除去対象のみを除去対象領域として設定することも可能になると考えられる．

謝辞

本研究の一部は，科学研究費基盤研究(S)24220004の補助により行われた．

参考文献

- [1] 森 尚平，一刈 良介，柴田 史久，木村 朝子，田村 秀行，“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.239–250，2011．
- [2] S. Zokai, J. Esteve, Y. Genc and N. Navab, “Multiview paraperspective projection model for diminished reality”, In Proceedings of the 2nd International Symposium on Mixed and Augmented Reality, pp.217–226, 2003.
- [3] Y. Shen, F. Lu, X. Cao and H. Foroosh, “Video Completion for Perspective Camera Under Constrained Motion,” In Proceedings of the 18th International Conference on Pattern Recognition, vol.3, pp.63–66, 2006.
- [4] 本田 俊博，斎藤 英雄，“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”，日本バーチャルリアリティ学会論文誌，vol.17，no.3，pp.181–190，2012.
- [5] D. Thachasongtham, T. Yoshida, F. de Sorbier and H. Saito, “3D object pose estimation using viewpoint generative learning,” In Proceedings of the 18th Scandinavian Conference on Image Analysis, pp.512–521, 2013.
- [6] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, “KinectFusion: Real-time dense surface mapping and tracking,” In Proceedings of the 10th International Symposium on Mixed and Augmented Reality, pp.127–136, 2011.
- [7] R. B. Rusu and S. Cousins, “3D is here: Point cloud library (PCL),” In International Conference on Robotics and Automation, pp.1–4, 2011.
- [8] P. Besl and N. McKay, “A method for registration of 3D shapes,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.14, no.2, pp.239–256, 1992.
- [9] B. Curless and M. Levoy, “A volumetric method for building complex models from range images,” In Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, pp.303–312, 1996.
- [10] S. N. Sinha, J. M. Frahm, M. Pollefeys and Y. Genc, “GPU-based video feature tracking and matching,” In Workshop on Edge Computing Using New Commodity Architectures, vol.278, pp.4321, 2006.
- [11] D. Arthur and S. Vassilvitskii, “k-means++: The advantages of careful seeding,” In Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, pp.1027–1035, 2007.
- [12] R. Hartley and A. Zisserman, “Multiple view geometry in computer vision 2nd edition,” Cambridge University Press, pp.88–93, 2003.
- [13] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”, Communications of the ACM, vol.24, no.6, pp.381–395, 1981.
- [14] Open Computer Vision Library, <http://sourceforge.net/projects/opencvlibrary/>
- [15] Open Graphics Library, <http://www.opengl.org/>
- [16] T. Möller and B. Trumbore, “Fast, minimum storage ray-triangle intersection,” Journal of graphics tools, vol.2, no.1, pp.21–28, 1997.