

# カメラ画像とセンサからの頭部位置・姿勢追跡による 指先軌跡の空中描画システム

田中雄規<sup>†</sup> 河野恭之<sup>†</sup>

**概要:** 本研究では、ユーザが自身の指先で空中に残した軌跡が、ユーザの頭部に取り付けたカメラから HMD を通して仮想空間に表示されることで空中に描画するシステムを提案・実装する。頭部装着カメラ画像から指先検出を行いカメラ画像中の指先座標を取得する。カメラ座標系を現実空間の座標系と位置合わせするために、頭部に取り付けた加速度センサ・ジャイロセンサから頭部の姿勢を計測し、時系列で隣接するフレーム間の特徴点を比較することで、頭部の位置推定を行う。位置合わせを行った指先軌跡を、HMD を通してユーザに提示する。

## 1. はじめに

本研究では、指先をポインティングデバイスに用いて空中に文字や絵を描画するシステムを提案・実装した。計算機の小型化・処理速度の向上に伴い、身体に小型計算機を装着するウェアラブルコンピュータの画像処理が現実的になってきた。これによりコンピュータグラフィックスで生成した仮想物体を現実空間に重ね合わせて表示する拡張現実感(Augmented Reality, AR)の研究が盛んになっている。本研究では、落書きと AR の概念を組み合わせる。世界中の観光名所では、訪問した人が残した落書きがある。ローマの観光名所であるコロッセオに落書きした観光客に罰金支払いが命じられるなど、法律に反しても落書きを行う人がある。落書きは反社会的な行為であっても人にとって魅力的な行為だと言える。そこで本研究ではウェアラブルなシステムを作成し、落書き行為を仮想空間上に行うことで周囲を気にせずに場所や空間に絵や文字を描くことを目指す。また将来的には訪れた場所にメモを残すことで、備忘に利用する。図 1 に、本システムが想定する利用イメージを示す。ユーザが、自分がここに居たことを隠れたメッセージとして残そうとしている。一般に AR を実現するためには、カメラの位置・姿勢を推定することが必要である。位置推定方法にはマーカや画像特徴を用いたビジョンベースによる手法と、センサを用いる方法、その 2 つを組み合わせる手法が挙げられる。山本らはセンサを取り付けたペン型のデバイスをポインティングデバイスに用いて任意の空間に文字や絵を描く手法を提案している[1]。ユーザの立ち位置を計測するために靴に位置センサを取り付け、またペンの空間座標を取り込むための位置センサを用いて現実空間と仮想空間の位置合わせを行う。センサを用いた位置合わせはユーザやカメラの位置姿勢を簡単に計測することが可能で計算量が小さい。しかしセンサのみで位置合わせを行うとセンサの誤差が累積するため時間の経過とともに誤差が

多くなってしまふ。中里らは、再帰性反射材を使用することで不可視マーカを作成し、実環境の景観を損なうことなくカメラの位置・姿勢を推定している[2]。マーカによる位置姿勢推定は、画像中のマーカ抽出が容易で計算量が小さく、現実空間における絶対位置がわかるという利点がある。しかし本研究では、ユーザが任意の場所で空中に文字や絵を描画するシステムを目指している。そのため、あらかじめビジュアルマーカを配置することは現実的ではない。R. Cipolla らはあらかじめ撮影しておいた環境画像とその撮影位置・姿勢データベースを利用して、新たに撮影したカメラ画像とデータベースを照合して対応付けることでカメラの位置・姿勢を推定している[3]。この研究では事前に環境の情報が必要であるため、実用的とは言えない。佐藤らはジャイロセンサからカメラ姿勢の計測を行う。ランドマークを検出し、その時のカメラ姿勢情報を付与し、プレートマッチングを行うことにより再び検出したランドマークからジャイロセンサのドリフト誤差を軽減する[4]。



図 1 本システムの利用イメージ

<sup>†</sup> 関西学院大学大学院理工学研究科 人間システム工学専攻  
Department of Human System Interaction, Kwansai Gakuin University

この研究ではランドマークとなる複雑な建造物の検出が必要であり、使用できる場所が限定される。藤井らは、Webカメラをペンに取り付けたデバイスを用いて空中に文字を書くことで文字を認識するシステムを構築している[5]。カメラから得られる動画像により局所画像間での動きベクトルを求める。文字を描画中に得られるベクトル群とあらかじめ用意しておいた辞書コードを比較して、ユーザが空中で書いた文字の認識を行う。しかしウェブカメラを取り付けたペン型デバイスを持つ煩わしさがある。またペンの動き方向のみを推定しているため、ユーザが文字の描画中に筆跡を確認することができない。そこで本研究では、頭部に取り付けたセンサとカメラ画像を用いて指先追跡で得られた指先の軌跡をヘッドマウントディスプレイ(HMD)に重畳表示し高速かつ高精度なウェアラブルな空中描画システムを目指す。

## 2. 提案システム

頭部に装着したカメラ画像から指先の軌跡を求め、画像の特徴を用いた頭部の姿勢推定と、頭部のセンサ情報を用いた頭部の姿勢計測を組み合わせた空間位置合わせにより、HMDを通してユーザが描画した文字や絵を提示する。指先をカメラ画像から検出し追跡することで、カメラ座標系における指先座標群を得る。頭の動きに応じてHMDを通して文字や絵を空中に描画するには、カメラ座標系を現実空間の座標系に変換する必要がある。そのため頭部に取り付けたカメラの画像情報からカメラの姿勢を推定し、頭部の姿勢に用いる。しかし画像情報を用いた姿勢推定は計算量が大きい。そこで頭部の姿勢計測のためのセンサを使用することで、計算量が少なく精度の高い空間位置合わせを行う。得られた頭部の姿勢を用いて、指先の軌跡を現実空間の座標系に位置合わせしHMDを通して、ユーザが描いた文字や絵が空中にあるかのように提示する。図2に提案するシステムの流れを示す。

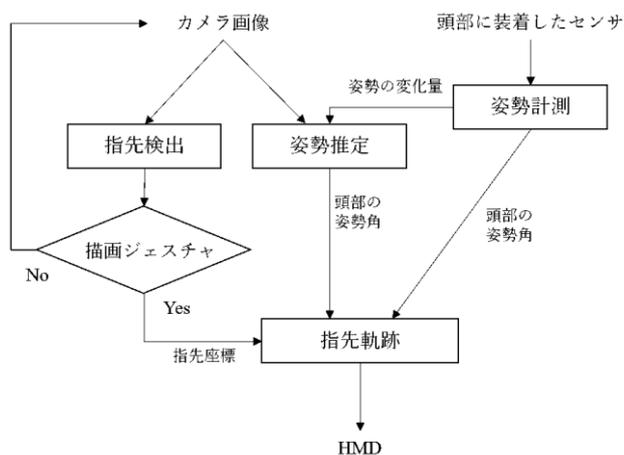


図2 システムの流れ

## 2.1 システムの構成

本研究では、ウェアラブルな文字や絵を現実空間に重畳表示するためには、ユーザの頭部の姿勢推定を行う必要がある。そこで頭部にカメラとセンサ、HMDを取り付けてウェアラブルなシステムを作成する。そのため本システムではHMDとセンサが一体になったOculus VR社[6]のOculus Rift Development Kit 2(DK2)を採用した。ユーザはOculusRiftDK2のディスプレイを通して現実空間を認識する。ユーザの目線位置にカメラを装着することで、ユーザに違和感がない視界を提示し、カメラ画像中に映る指先を画像処理で検出する。ユーザは図3に示すように、頭部にHMD、カメラを装着する。システムを装着したユーザが、自分の人差し指をカメラ画像に映る範囲内で動かすことで、文字を描画することができる。



図3 本システムを利用したユーザ

## 2.2 空間位置合わせ

ユーザが頭部に装着したカメラからHMDを通して指先の軌跡を見るとき、頭部の動きの変化に応じて仮想空間上に表示する指先の軌跡も変化させる必要がある。そのため頭部に取り付けたカメラの姿勢を計測し、カメラ画像中の指先検出で得た座標をカメラ座標系から現実空間の座標系に変換する。カメラの姿勢を得るには、画像処理に基づいた姿勢推定とカメラに加速度センサ、ジャイロセンサなどを取り付けて姿勢を求めるセンサに基づいた姿勢計測がある。画像処理に基づいた姿勢推定は、画像の特徴を抽出しフレーム間で比較を行い対応付けするため、精度は高いが計算量が大きい。一方、センサに基づいた姿勢計測はセンサから得られる値を計算処理するだけであるため計算量は小さいが、センサの誤差により精度が低くなる。そこで本システムではカメラとセンサを組み合わせることで計算量が小さく、またカメラ画像を用いて精度の高いシステムを実現する。

### 2.2.1 カメラ画像を用いた空間位置合わせ

時系列に隣接するカメラ画像間の特徴量を対応付けすることでカメラ姿勢の回転・移動を推定する。本システムでは、カメラをユーザの視線位置に取り付けて、HMDにカメラからの映像を提示する。つまりカメラ画像は頭部の姿勢の変化に依存するため、時間軸に隣接するフレーム間では、画像の背景に重なりがあると想定できる。画像の特徴点を求め、フレーム間で対応付けを行うことにより姿勢の変化量を得る。背景に重なりを持つ画像において、互いの点が現実空間で同じ箇所を指しているか判別するには、特徴的な点を探し出し比較すればよい。カメラから得られた画像に対して、グレースケール化を行い、Herbert Bayらによって提案された Speeded Up Robust Features (SURF) アルゴリズム[7]を用いて特徴量をそれぞれ得る。SURFは、Loweの Scale-Invariant Feature Transform (SIFT) [8]を近似処理に置き換えて高速化したもので、画像中の特徴となる点の抽出し、その特徴点の周辺領域の輝度勾配の方向とスケールを特徴記述するアルゴリズムである。こうして得られた特徴点を画像間で比較することにより、現実空間で同じ箇所を指す点を得る。

K-近傍探索法を高速化した Muja [9]の Fast and Accurate Approximate Nearest Neighbor Search (FLANN 探索法)を用いて、フレーム間での特徴点の対応付けを行う。実際に図4の左図を前フレーム、右図を現フレームと仮定し、対応付けを行う。2枚の画像はカメラの姿勢が右方向に回転したときの時間軸に連続する画像と言える。この2枚の画像間での頭部の姿勢の変化を求めるために、画像の特徴点を対応付けする。図5に、これらの画像に対して、画像間での対応付けを行い2枚の画像を合成した結果を示す。本システムでは2枚の画像を合成することはないが、説明のために合成した画像を示している。また対応付けられた特徴点ペアを明示するため直線を示している。示した直線から明らかに誤った対応があることがわかる。

頭部に取り付けたセンサの情報から明らかな誤対応を除去する。FLANN探索法を用いた特徴点の対応付けは、特徴量空間での距離の近さで対応付けを行う。そのため対応付けした点が画像上では大きく異なる場合がある。このような誤対応は、最終的に得られる変換行列の精度の低下を引き起こす。本研究ではカメラとセンサは頭部に固定した状態であるため、センサから得られる頭部の運動方向を基に、誤対応を検出することが可能である。ジャイロセンサの値を積分することで得られた回転行列を用いて、カメラの運動方向を8方向に分類する。例えば、上を見た時、新たに得られる対応点は前フレームと対応する点に対して、同様に、上方向にあるはずである。得られた対応点群の中で、頭部の運動方向と類似しない対応点を誤対応とみなし除去する。8方向に分類したカメラの運動方向ベクトルと前後のフレームで対応する点を結ぶベクトルのなす角度が



図4 時間軸で隣接した画像

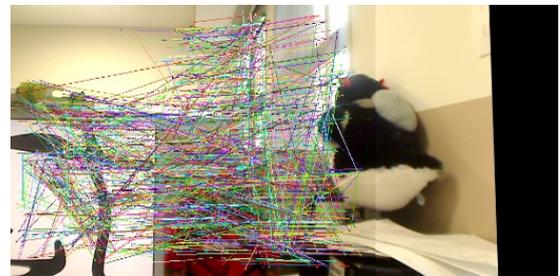


図5 対応付けして得られた合成画像

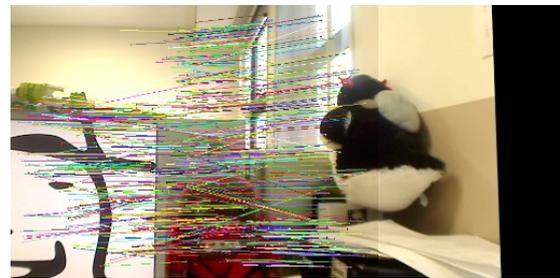


図6 誤対応を除去して得られた合成画像

25度以下の時、類似すると判定した。実際に、図5の2枚の画像に対して誤対応の除去を行い、新たに得られた対応点を用いて合成した結果を図6に示す。図5に比べて対応する特徴点が減少していることがわかる。2枚の画像間ではカメラの姿勢が右方向へ移動しているため、新たなフレームで得られる対応点も前フレームに対して右方向に存在する。対応付けされた特徴点間の長さや頭部の回転量を比較することで、さらに誤対応を除去が可能ではある。しかしカメラ画像中に映る物体の奥行き情報がわからないため、遠くの物体と近くの物体ではフレーム間で得られる特徴点間の長さが異なる。そのため本研究では、対応する特徴点の向きだけを利用する。本システムでは、ポインティングデバイスを指先としているため、カメラ画像中に手の領域が含まれることが多く、カメラ画像中に占める割合も大きい。そのため手領域のエッジ部分で特徴点を持ち対応付けられることが考えられる。この手法は、実装が容易で本システム特有の誤対応に対しても有効である。誤対応を除去して得られた対応点からフレーム間の射影変換行列を求める。この行列は前フレームのある点が見フレームの対応する点に移動したときの変換行列とみなせる。ゆえに本

研究では、この行列の逆行列を指先の軌跡に毎フレーム掛けることで、カメラ座標系を現実空間の座標系に対応付けしている。

## 2.2.2 センサを用いた頭部姿勢追跡

2.2.1 項で述べた画像特徴を用いた空間位置合わせは各フレームで、画素の比較を行うため計算量が大きくリアルタイムで使用することは難しい。そこでセンサを使用することにより、小さい計算量で頭部の姿勢計測を行う。指先の軌跡を空中に描画する際、カメラ画像中の指先座標はHMDの姿勢の変化に応じて変化する。頭部に取り付けた加速度センサ、ジャイロセンサ、地磁気センサの情報から姿勢角を求めることで、姿勢変位の行列を取得する。ジャイロセンサは角速度を検出するため、時間積分を行うことで頭部の回転角度を計測する。頭部の回転運動などに対してロバストであるが、センサ値の積分により姿勢角度を求めているため、センサ値に含まれるわずかな誤差も蓄積してしまう。そこで蓄積した誤差を補正するために、加速度センサと地磁気センサを用いる。しかし加速度センサは頭のブレなど、並進運動が加わると誤差が大きくなるため正しい姿勢を得ることができない。そこで本研究では、相補的にセンサから得られる姿勢行列を利用することでロバストな姿勢計測を行う。図7に座標系を示す。ピッチ角、ヨー角、ロール角をそれぞれ $\theta_p$ 、 $\theta_y$ 、 $\theta_r$ とする。ジャイロセンサから得られる値をそれぞれの軸の方向に $\omega_x$ 、 $\omega_y$ 、 $\omega_z$ とするとき、ジャイロセンサから得られる姿勢角の変化量はオイラー角速度を用いて式1で表せる。また加速度センサから得られる値を $\mathbf{a}$ とすると、頭部の姿勢の傾斜は回転行列と重力加速度を用いて式2で表すことができる。これにより $\theta_p$ 、 $\theta_r$ は式3となる。 $\theta_y$ は重力加速度と直交するため、加速度センサからは得られない。そのため加速度センサで得られた $\theta_p$ 、 $\theta_r$ と地磁気センサの値を用いて算出する。加速度から得られた $\theta_p$ 、 $\theta_r$ を用いて、地磁気の座標系をx軸、

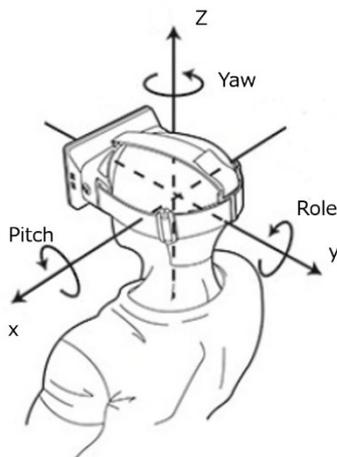


図7 頭部の座標系

y軸について回転すると、ヨー角について回転した地磁気センサの値が得られる。はじめに得られた値を原点とし、現在の値と差をとることで $\theta_y$ は計測できる。ジャイロセンサはドリフト誤差が発生するため、低周波成分に弱い。また加速度センサは頭のブレなどの高周波成分に弱いため、それぞれにハイパスフィルタとローパスフィルタをかけて足して合わせることでロバストな頭部姿勢計測を行う。得られた頭部姿勢のフレーム毎の変化量を、カメラ画像中の指先の軌跡に用いることで、現実空間の座標系に対応付けしている。

$$\begin{aligned}\dot{\theta}_y &= (-\omega_x \sin \theta_p + \omega_z \cos \theta_r) / \cos \theta_p \\ \dot{\theta}_p &= \omega_x \cos \theta_r + \omega_z \sin \theta_r \\ \dot{\theta}_r &= -\dot{\theta}_y \sin \theta_p + \omega_y\end{aligned}\quad (1)$$

$$\begin{pmatrix} 0 \\ -g \\ 0 \end{pmatrix} = \mathbf{R}_z \mathbf{R}_x \mathbf{R}_y \mathbf{a}\quad (2)$$

$$\begin{aligned}\theta_p &= \tan^{-1}(-a_y / \sqrt{a_x^2 + a_z^2}) \\ \theta_r &= \tan^{-1}(-a_x / a_z)\end{aligned}\quad (3)$$

## 2.2.3 組み合わせ手法

2.2.1 項と 2.2.2 項で求めたそれぞれの頭部姿勢推定手法の長所を組み合わせる手法を提案する。画像を用いた頭部姿勢推定による射影変換行列は頭部姿勢の回転・移動に強い。しかし計算量が大きい問題がある。さらに隣接するフレーム間で同じ背景を含まない場合は射影変換行列を求めることができない。本システムでは、これらの問題点を解決するためセンサから得られる頭部姿勢の変化量に注目する。変化量が大きいとき、つまり隣接するフレーム間で背景に重なりが少ない場合は画像を用いた位置合わせから得られる姿勢行列の代わりセンサから計測した回転角の変化量を用いる。本システムで使用するカメラの画角が76度であるため、画像の縦横比と隣接する画像間の背景の重なり度合を考慮して変化量が大きい時のしきい値は34度とした。またセンサから求めた頭部姿勢の変化量が小さい時、つまりフレーム間で背景に変化が少ないときは画像を用いた位置合わせを行わず、センサを用いた位置合わせを行い、変位が大きくなった時それまでのセンサにより代用していた位置合わせを画像による位置合わせで補正する。

## 2.3 指先軌跡の空中描画

カメラ画像中の指先を検出し、時系列に検出を行うことで指先の軌跡を得る。得られた軌跡を3章の位置合わせを用いて、ユーザにHMDを通して指先で描いた文字や絵を提示する方法について述べる。

### 2.3.1 指先軌跡の取得

指先をポインティングデバイスとして用いるため、頭部に装着したカメラ画像から指先を検出する。まずカメラ画像中の肌色領域を抽出するため、カメラから得た RGB 画像を HSV 画像に変換し、Hue(色相)に注目する。人の肌色の領域について二値化する。頭部に装着したカメラからユーザの指先までの距離は腕の長さ程度であるため、ユーザ自身の手の領域はカメラ画像における肌色の物体領域で最も大きくなると仮定し、抽出した肌色領域画像の中で最大の領域を手の領域候補とする。次に得られた手領域候補がポインティングデバイスであるか判別する。本研究では、ユーザが指を使って空中に描く動作は、人差し指を上向きにして指すジェスチャが自然と考え、図 8 に示すジェスチャを想定する。これにより手領域候補の左上端に位置する点を検出すべき指先である可能性が高い。指先候補点を中心として時計回りに探索して見つけた肌色領域を新たな中心点として、円探索を繰り返す。このとき中心と新たな点を結ぶベクトルを保持しておき、新たに得られたベクトルとの位置関係が、指先に沿う境界と手の甲に沿う境界の位置関係に類似する時、ジェスチャと判定しポインティングデバイスであるべきユーザの指先とする。



図 8 人差し指ジェスチャ

指先を用いて絵を描画するため検出した指先を追跡し、指先の軌跡を取得する必要がある。時系列上で連続する画像において、対象物体のフレーム間での移動距離は大きくないと考え、前フレームの指先位置の座標を基準にして、探索する範囲を限定することで、探索範囲内で検出した指先を軌跡とする。探索窓は誤検出を防ぎ、探索数の減少により処理速度向上にも役立つ。指先検出の欠落を考慮して、2 フレーム続けて探索窓範囲内でジェスチャが見つからない場合は、指先の軌跡の終点とみなし、新たな指先軌跡の始点を検索する。

### 2.3.2 描画ジェスチャ

人がペンを使って紙に文字を描く際に、ペンを紙につける動作と離す動作を行うことで、複数の軌跡を描くことができる。本システムでも空中に文字や絵を提示したいため、一筆書きではない複数のストロークで絵を描画することを

実現する。そのため特定のジェスチャを使ってユーザが描画しているかを区別し空中に描画する。本システムでは、指先を用いて空中に軌跡を描画する際に図 9 に示すように人差し指と親指を上げ、“レ”の字を左右反転にしたジェスチャを適用する。この描画ジェスチャは、描画開始時には親指を上げ、描画終了時には親指を下ろすという動作だけで実現できるため、ポインティングデバイスである人差し指がつかれて動くといった影響が少ないという利点がある。



図 9 描画ジェスチャ

### 2.3.3 軌跡の描画

描画中に取得した指先の位置座標データを現実空間の座標系に変換する。フレーム毎に得られた指先座標に、2.2 節の手法で得られた頭部姿勢の変化量を用いて現実空間の座標系に対応する座標を得る。頭部姿勢の変化がある度、軌跡に適用し座標を更新することで、カメラ座標系にお

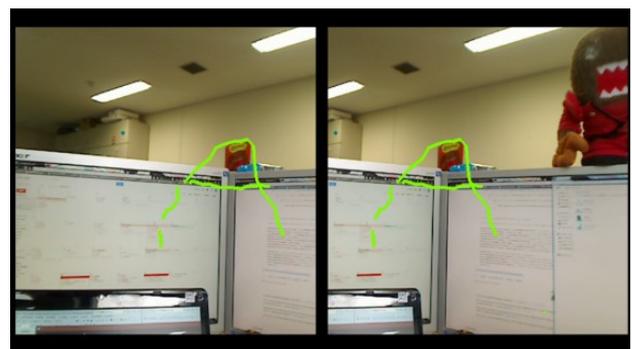


図 10 正面の視点からの描いた軌跡



図 11 異なる視点からの軌跡

る指先座標を現実空間に対応付ける。こうして得られた指先の軌跡を HMD でユーザに提示する。本システムで使う OculusRiftDK2 は、左目と右目にそれぞれディスプレイを持つがカメラは単眼である。そのため実装では、カメラ画像を分割することで左目用と右目用のディスプレイに対応する。実際に本システムを使用して空中に描いた軌跡を重畳表示した画面を図 10 に示す。また図 11 に頭部を動かして、異なった視点から重畳表示した画面を示す。

### 3. 実験

画像による位置合わせは計算量が多い。一方、センサによる位置合わせは精度が落ちるが、計算量が小さい。本研究では、これらの欠点を解決するために組み合わせ手法を 2.2.3 項で述べた。本章では、作成したシステムと画像による位置合わせ、センサによる位置合わせの精度、計算量を計測、比較し有用性を評価する。

#### 3.1 計算量と精度の比較実験

実験では、実装したシステムを使用し精度と計算量を測定する。実験内容について説明する。あらかじめ現実環境に円マーカを貼り、カメラ画像中にも同様に固定点を描画する。固定点と現実環境の円マーカが重なった状態から運動を開始する。このとき固定点の座標に 3 つの手法を用いて位置合わせを行い 3 つの評価点を得る。得られた 3 つの評価点に頭部の姿勢変化に応じて、それぞれ位置合わせを逐次行い 3 つの評価点の座標を更新する。運動を行う過程で、カメラ画像中に描画した固定点を円マーカに重ねる。円マーカと点が重なった時、3 つの評価点の座標を記録しておく、この 3 つの評価点と固定点の距離を計測することで精度評価を行う。実験で行う運動は以下に示す。

- I. 静止状態
- II. 上下回転運動
- III. 左右回転運動

本システムで提案する組み合わせ手法では、計算量を小さくするため頭部の姿勢の変化が少ない場合、計算量の小さいセンサを利用する。実験では、提案手法の有用性を確かめるため、左右運動における頭部の回転速度を変えて計算量を測定する。計算量の測定は、全体のシステムの中で、位置合わせにかかる処理の時間を計測する。指先検出やその他の処理は含まない。

#### 3.2 結果と考察

##### I. 静止状態

図 12 に、静止状態での各手法を用いた精度の結果を示す。縦軸は pixel、横軸はフレーム数である。実験で用いた

円マーカと固定点の大きさはおよそ 12pixel であるため、画像を用いた手法では位置合わせによるズレはほとんどない。一方、組み合わせ手法とセンサを用いた手法では時間とともに誤差が大きくなっている。これはセンサに含まれる誤差のためだと考えられる。組み合わせ手法では、計算量を減らすため姿勢角の変化が少ないとき画像を用いた位置合わせを行わず、センサを用いた位置合わせを使用する。そのため図のようにセンサを用いた手法と同じ誤差の変化が見られる。しかし 50 フレーム付近で頭部の姿勢が動いたため、画像を用いた手法を使用している。組み合わせ手法では静止状態での、頭部の姿勢計測が難しい、しかし画像を用いた手法では、精度が高い。そこで頭部姿勢の変化量が小さい状態が続けば、画像を用いた手法で空間位置合わせすることで改善する。

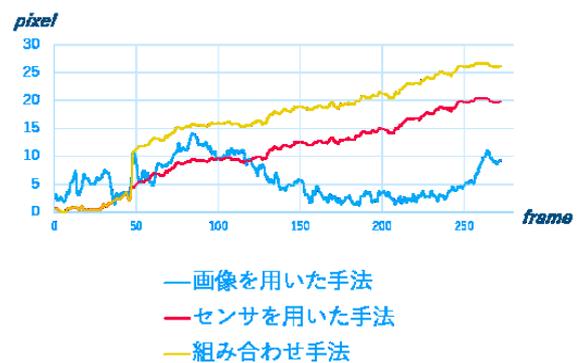


図 12 静止状態における精度

##### II. 上下回転運動

図 13 に、頭を上げ下げする運動について各手法を用いた精度の結果を示す。上下運動は全体で 325 フレームである。センサを用いた手法は運動 I、II と同様に誤差が蓄積されている。特にグラフの横軸 40 の間で誤差が増加している。これは横軸の 40 と 41 の間で 200 フレーム分の運動を行っているからである。画像を用いた手法は誤差の最大

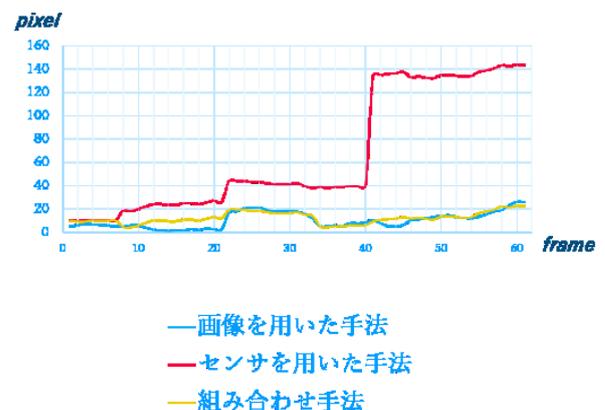


図 13 上下回転運動における精度

値が 26pixel, 平均値が 11pixel, 組み合わせ手法では最大値が 23pixel, 平均値が 13pixel であった. 組み合わせ手法が画像を用いた手法より最大値が低くなっている. また平均値がほぼ等しい. これはおそらく実験で行った上下運動の回転量が大きかったため, 画像の位置合わせを用いたと考えられる.

### III. 左右回転運動

図 14 に, 頭部を左右に回転する運動について各手法を用いた精度の結果を示す. 図中に示されたグラフは, カメラ画像中の固定点と円マーカが重なった時のみを表示している. 実際には 259 フレームの間, 左右運動を行っている. 画像を用いた手法は誤差の最大値が 13pixel であり, 精度は十分に高いと言える. センサを用いた手法では, 誤差が蓄積されたためフレーム数の増加とともに誤差が大きくなっている. 組み合わせ手法は, 誤差の最大値が 48pixel, 平均値が 19pixels であった. 特に 30~35 フレームで組み合わせ手法を用いた位置合わせの精度が悪くなっている. これは左右回転運動の際, 回転量が小さかったためと考えられる. そのためセンサを用いた手法を使用し, 誤差の値が大きくなった. これも I と同様, 連続して姿勢角の変化量が小さければ, 画像を用いた位置合わせを用いることで解決できると考える.

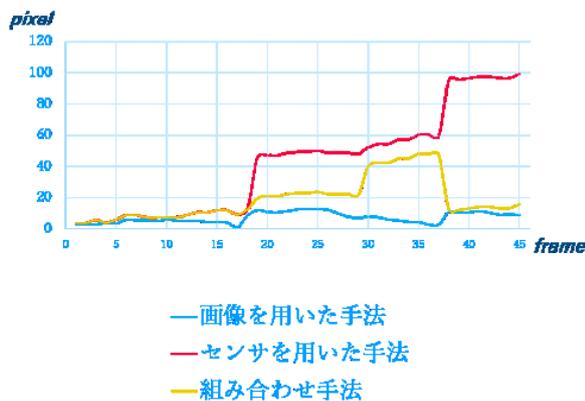


図 14 左右回転運動における精度

表 1 に, 異なる回転速度で頭部を左右に回転運動した時の計算量について表す. 回転速度が遅い時, つまり回転量が小さいとき, 組み合わせ手法の計算量は 12.51fps となり, 画像を用いた手法よりも計算量は小さいと言える. また, 回転速度が速い時も 9.23fps となった. 回転運動が遅いときに比べ, 画像を用いた手法を行う比率が高くなったため回転運動の速度が遅い時に比べて, 計算量が大きい. しかし画像を用いた手法より計算量は軽い.

表 1 運動速度の違いによる計算量 単位: fps

	センサ	画像	組み合わせ
遅い	3339.38	5.06	12.51
速い	2985.78	5.00	9.23

### 4. おわりに

本研究では, ユーザが人差し指と親指を上げるジェスチャをポインティングデバイスに用いて, 空中に文字や絵を描画するシステムを提案・実装した. 実際には, センサを内蔵した Oculus Rift DK2 とカメラを用いることでシステムを作成した. ユーザはこれらのデバイスをつけて, HMD に表示されるカメラ画像を通して周りの環境を認知する. ユーザが HMD から自分の指先を捉えることで, 画像処理により指先を検出する. 検出した指先座標を現実空間の座標系に対応付けするために, カメラ画像による頭部姿勢推定とセンサによる頭部姿勢計測を組み合わせた. 組み合わせ手法とカメラを用いた手法, センサを用いた手法について精度と計算量の比較実験を行った. どの運動においてもセンサを用いた姿勢計測より精度が高い. 計算量については, カメラを用いた姿勢推定より計算量が小さく, 1 秒間に 10 フレーム程度の処理が可能である.

### 参考文献

- 1) 山本吉伸, 椎尾一郎, “空気ペン -空間への描画による情報共有-”, 第 59 回情報処理学会全国大会講演論文集, Vol.4, pp.39-40, 1999.
- 2) 中里祐介, 神原誠之, 横矢直和. “不可視マーカを用いた位置・姿勢推定のための環境構築とユーザ位置・姿勢推定システム”, 日本バーチャルリアリティ学会論文誌 13(2), pp257-266, 2008.
- 3) R. Cipolla, D. Robertson and B. Tordoff: “Image-based localization”, VSMM, pp.22-29, 2004.
- 4) 佐藤清秀, 穴吹まほろ, 山本裕之, 田村秀行. “屋外装着型複合現実感のためのハイブリッド位置合わせ手法”, 日本バーチャルリアリティ学会論文誌, Vol. 7, pp.129-137, 2002.
- 5) 藤井祐介, 竹沢恵, 真田博文, 渡辺一央. “空中手書き文字入力システムの構築に関する一考察”, モバイルコンピューティングとユビキタス通信研究会研究報告, Vol 6, 2009-MBL-50, pp.1-4, 2009.
- 6) Oculus VR Inc. 2015. <https://developer.oculus.com> (2015 年 2 月 6 日 アクセス)
- 7) H.Bay et al.: "Speed-Up Robust Features (SURF)", Computer Vision and Image Understanding, Vol. 110, No.3, pp.346-359, 2008.
- 8) David G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2, pp. 91-110, 2004.
- 9) M. Muja and D. G. Lowe. "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration". Proc. Interational Conference on Computer Vision Theory and Application(VISSAPP'09), 2009.