Vol.2011-CVIM-177 No.4
2011/5/19

手先の三次元位置推定に基づく複数動画を同時操作可能なジェスチャインタフェース

藤野晴樹[†1] 森武俊[†1] 下坂正倫[†1]
野口博史[†1] 佐藤知正[†1]

本論文では動画群の中から特定のシーンをマニュアルで探す操作を効率的に行えるようにするために複数の動画を同時に操作できるシステムを提案する．システムの操作をユーザの手の動きに着目したジェスチャで行うことを考え，環境設置型のセンサを用いて手先の三次元位置をリアルタイムで推定する．片手の動きの速度に着目したジェスチャデザインにすることで動きと動画の操作を結びつける．また提示部の表示構成をドック型にするなどの工夫により同時・複数人でも使いやすいインタフェースとした．このシステムを用いることによって動画のシーン探索の効率が向上することを実際の使用シーンに則した実験により示す．

Gesture Interface for Simultaneous Operation of Multiple Movies Based on 3D Hand Position Estimation

H.Fujino,[†1] T.Mori,[†1] M.Shimosaka,[†1] H.Noguchi[†1]
and T.Sato[†1]

At video sharing sites, users have tasks to look for specified scenes manually from some videos after searching with keywords from a lot more movies. In this paper, we propose a system which allows users to operate multiple movies at the same time using hand gestures. We use environment-embedded sensors for measurement of the hand position in 3d space to recognize gesture and designed effective gesture and dock styled presentation part to realize simultaneous operation of multiple videos for one or more users. Experimental result shows the effectiveness of the method.

1. はじめに

1.1 研究背景

近年，動画共有サイトの普及などの影響から，個人のアクセスできる動画の数は飛躍的に増加している．ユーザの閲覧できる動画数の増加に伴い，いくつかの問題も生じている．その１つがアクセスできる動画の数の増加に比例して，ユーザが本当に見たいシーンを見つけ出すのが難しくなってしまっていることである．一般的に見たいシーンを探す最初の手順として考えられるのは，そのシーンに関連するキーワードによる検索である．しかしこの検索にも限界がある．その理由として２つ考えられる．１つは言語による絞り込みの際，動画の１シーンごとに情報が付加されているわけではないため，検索の対象となるのは動画のタイトルやタグといった情報のみとなっていることが挙げられる．もう１つの理由としては自分の見たいシーンを言語のみで表現することに限界があるからである．そのためキーワードによる検索では大量の動画の中からある程度の量まで絞り込むことしかできず，その動画群の中から順番に中身を目で見て確認しながら探していかなければならない．このような現状を打開する手法が必要となる．

その１つの方法として，動画の操作手法の改善が考えられる．現在動画を操作するデバイスとして用いられているのは，テレビ等で用いられるリモコンやパーソナルコンピュータ等で用いられるマウス・キーボードなどが一般的である．これらのデバイスは通常複数同時に接続し別々の操作を担うことは想定されていない．そのため同時に操作できるのは動画というメディアに限らずただ１つであることが多い．しかし複数の動画の中から目的の１シーンを見つけるというタスクを考えたとき操作対象が常に１つに限定されてしまうことはシーン探索の効率を下げていると考えられる．なぜならそういった状況では見たいシーンのおおまかなイメージは掴めているため複数の動画を同時に見ながらでも，現在見ているシーンが探しているシーンかどうかを判断するのは難しくなく，人目でできるからである．また動画を複数人で見るといったシーンは一般的によくあることであるが，見たいシーンを探す際には同様に操作方法の問題から，通常同時に別々の動画からシーンを探すことができない．このことから，複数人で同時に別々の動画を操作できれば効率が飛躍的に向上すると考えられる．

†1 東京大学
University of Tokyo

1.2 提案手法

複数動画を同時に操作するためには複数同時入力が可能なインタフェース入力が必要となる．このとき操作できる動画の数と操作するデバイスの数が比例することは，デバイスの管理の手間や使用時に混乱を招く恐れがあり，好ましくない．そこで環境中のセンサを用いて腕の動きを検出し，ハンドジェスチャとして解釈することでインタフェース入力とする操作方法が考えられる．この手法であればそれぞれの腕を別の動画の操作に割り当てることで両手や複数人による複数動画同時操作が可能になると期待できる．この手法には以下のようなメリットがある．

まずユーザへの負担が少ないことである．ハンドジェスチャの認識には環境中に設置したセンサを用いるため，ユーザはジェスチャ認識のためにグローブ等を装着する必要がない．これは操作する動画の数と操作するために必要なデバイスの数が比例関係になくなり，デバイスの管理や操作時の混同を防げるという利点もある．

また複数動画操作に対する相性が良いこともメリットとして挙げられる．キーボード・マウス・リモコンなどのインタフェース入力は同種のデバイスを用いた複数同時他入力をすることを想定されていないため，一般的には動画を同時操作することができない．複数同時操作が可能な他の方法として，音声入力やタッチパネルを用いる方法が考えられる．しかし音声入力は動画というコンテンツが性質上音声を発するものであるため適していない．またタッチ操作は動画を見ることを考えると手で画面を隠してしまう可能性があり，相性が悪いと考えられる．それらと比較してハンドジェスチャは相性が良いと言える．

腕の動きによるジェスチャを認識するために手先の三次元位置に注目し，本研究ではこの位置を推定しジェスチャの解釈をすることにした．この目的のために利用出来るセンサは複数ある．センサに違いがあっても同様に使えるようなインタフェースにすることは汎用性の観点から考えて重要である．そこで手先位置推定とインタフェース入力への変換部を分離することでそれを可能にした．インタフェースへの入力変換部の有用性を検証すべく，本研究では手先位置推定手法として多視点カメラシステムにより復元されるユーザのボクセルを用いる方法と，距離画像センサである Kinect を用いる方法の 2 つの手法を用意した．

またジェスチャ操作の観点では，速度に着目した簡潔なジェスチャデザインにすることで複数の動画であっても同時に操作しやすくなるようにした．さらに同時操作や複数人操作を想定し，ドック型の動画配置などの，可視性の観点からも適切であると考えられるユーザー提示方法を提案する．

提案するインタフェースのシステム構成は**図 1** のようになる．

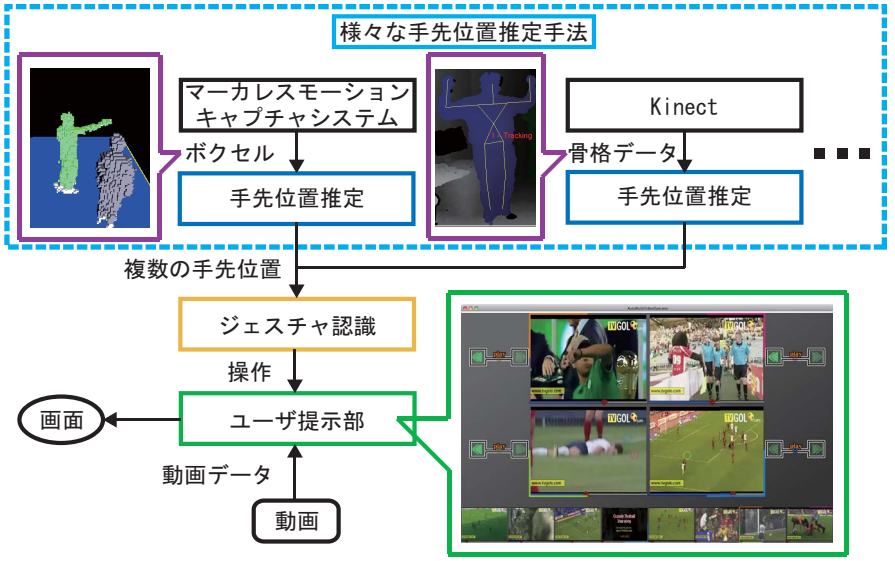

図 1 システム構成
画面中の動画は http://www.youtube.com より
Fig. 1 System configuration
The movies in this figure are from http://www.youtube.com

2. 関連研究

センサの装着なしに人間の行動情報を取得できる研究としては，カメラを用いてその画像から何らかの特徴を抽出し，認識に用いるという手法がある．例としては，Sminchisescu ら[1] や Schüldt ら[2] などの研究が挙げられる．彼らは環境に設置した単眼カメラから得られる 2 次元画像を使って姿勢情報を獲得する．また，Weinland ら[3] は，複数のカメラを使い多視点動画像を得て，そこから 3 次元の人体領域を復元し，姿勢情報として用いている．環境にカメラを固定するのではなく，動くカメラから得た画像を用いて認識するアプローチとして， Yilmaz ら[4] の研究もある．そこでは動くカメラに対応したエピポーラ幾何学を提案し，異なる対象者や異なる環境においても頑健に認識する手法を紹介している．

また，認識方法も多岐にわたり，例えば， Alireza ら[5] は画像から取得した 2 次元の関節角を用いて Motion Exemplar と呼ばれる関節の順序を示したモデルを構築し，Gibbs

Sampling を用いてその順序を推定するという手法を採っている． Niebles ら[6] は，画像フレームから Feature Layer と Part Layer と呼ばれる 2 層の階層構造を経て特徴を抽出し，主成分分析による特徴量圧縮の後， SVM によって認識を行っている．

システムに意思を伝達するためのジェスチャ認識やジェスチャインタフェースの研究も数多く存在する．Coldefy ら[7] はリモートで映像の共有とジェスチャによる操作が可能なツールについて研究しており，Zigelbaum ら[8] は反射材を埋め込んだグローブを用いてハンドジェスチャを認識し，擬似 3D 空間に並べた動画像の位置・大きさ・操作を行うインタフェースの研究を行っている．またナチュラルインタフェースの研究として Kinemote プロジェクトがあり，これは Kinect センサを利用して手の動きを認識し，PC のキーボードやマウスにジェスチャをマッピングすることでデバイスなしに従来の PC 操作をすることを目的としている．

3. 手先位置推定手法

1.2 節で触れたとおり，手先位置を推定する方法として本研究では 2 つの手法を用意した．以下でそれぞれの手法と，推定した手先位置をインタフェース入力に変換する方法について述べる．

3.1 多視点カメラ環境を用いた方法

1 つは多視点カメラシステムであるマーカレスモーションキャプチャシステム[9] を用いた推定手法である．以下でマーカレスモーションキャプチャシステムによるボクセル復元の概要とボクセルからの手先位置推定手法について説明する．

3.1.1 ボクセル復元

マーカレスモーションキャプチャシステムは知的居住空間内における居住者の位置・姿勢推定を目的としているため，一般的な居住空間での利用を想定しており，家具などが配置された部屋の天井を囲うように配置された 8 台のカメラを用いる多視点カメラシステムである．このシステムでは，知的居住空間内に存在するユーザの人体領域に相当するボクセルを復元することができる．各カメラ画像に対して背景差分法を用いてシルエット画像を作成し，8 枚のシルエット画像から視体積交差法を用いという手順でこれを行う．その様子を**図 2** に示す．

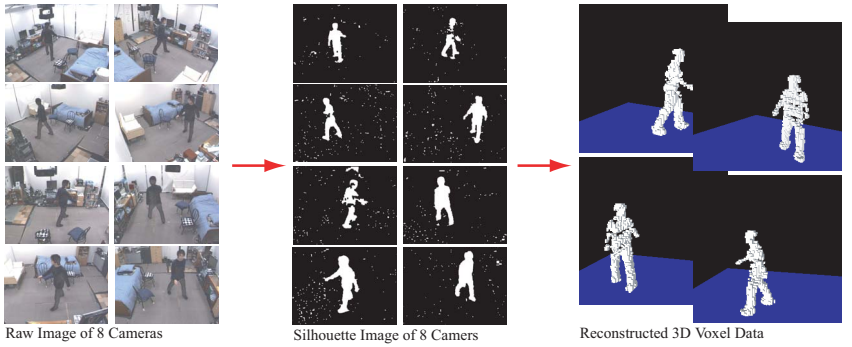

図 2 ボクセル復元の例
Fig. 2 Sample of voxel reconstruction

3.1.2 腕のクラスタリング

復元されたボクセルから手先位置を推定するために以下のような仮定を用いる．

(1) 操作の際，手はインタフェースとして出力している画面の方向に突き出している
(2) 肩の位置は重心から画面方向に対して方位空間において垂直な位置にある
(3) 手先は腕領域内で肩から一番遠い位置にある
(4) 操作時に上半身は床に対して垂直になっている

これらの仮定を用いて手先の位置を以下のような方法で推定する．

まず，仮定 1 に基づき操作時に手は重心の近くにないと考えられるので，対象空間内にいる人物それぞれの方位空間において重心から一定の範囲内にあるボクセルを除去する．本研究ではボクセルを除外する範囲を重心から方位方向に半径 20 cm の領域とした．また，重心よりも低い位置にあるボクセルに関しても手先位置推定に必要ないと考えられるので，同様に除外した．

残ったボクセルは想定する操作姿勢において肩から手先の部分に相当する箇所のみになると考えられる．このボクセル群を 2 つにクラスタリングすることでそれぞれの腕の領域に分ける．本研究では 2 種類のラベリングを用いてこれを行った．

1 つはボクセル間の空間的な連結性を考慮した連結型ボリュームラベリングである．まず除外されなかったボクセル群に対して 6 近傍ラベリングを行い，対象空間内にいる人物につき 10 から 15 程度の塊に分ける．これらの塊の重心の位置に基づいて右手を構成するボクセルと左手を構成するボクセルの 2 つのクラスにクラスタリングする．クラスタリング手法としては k-means 法を用いた．この手法を用いると腕同士が近づいたときに誤ったラベルを振ってしまうことがある．

上記の問題に対処するためにボクセル同士の連結性を考慮せず，各ボクセルを独立的に扱

う探索型ボリュームラベリングを併用した．この手法では，ラベリングとクラスタリングを別のフェーズとして実行するのではなく，1 つのフェーズに落とし込む．ボクセルごとに，1 つ前のフレームのラベリング結果との最小距離を探索する．ここでいう距離とは，それぞれのボリューム全体を代表する重心座標との距離ではなく，ボリュームを構成する要素との距離を指す．最小距離の値が一定値を下回る場合，ボリュームに付与されていた ID を新たにボクセルに付与する．最小距離が一定値以上の時には腕が突然現れることは考えにくく，なんらかの原因で生じたノイズの可能性が高いのでラベルは付与しない．このようにすることで，それぞれのボクセルを独立にラベリングすることができ，ボクセル同士の連結性の影響を軽減することができる．

この手法の処理の流れを以下に定式化する．ある時刻 t におけるボクセルデータを $\boldsymbol{v}(t)$ と定義する．ボリュームラベリングの結果，I 番目のボクセルには以下の属性 $\{\boldsymbol{v}(t)\}_I$ が付与される．

$$\{\boldsymbol{v}(t)\}_I = \begin{cases} -1 & : \text{voxel consists of left arm} \\ 0 & : \text{voxel doesn}'\text{t exist} \\ 1 & : \text{voxel consists of right arm} \end{cases} \tag{1}$$

また，I 番目と $\tilde{I}$ 番目のボクセル間の距離を測る関数を $D_v(I,\tilde{I})$ と定義する．この関数は I 番目のボクセルを開始位置として，6 近傍をたどっていき，$\tilde{I}$ 番目のボクセルに到達するまでの経路の最短値（city-block 距離，マンハッタン距離）を距離として返す．このような経路ベースの距離を定義すると，一度探索した経路の探索結果を蓄えておくことにより，同一の経路を再度探索する手間を省くことができ，効率的な探索が可能となる．

属性 $\{\boldsymbol{v}(t)\}_I$ は前フレームのラベリング結果 $\boldsymbol{v}(t-1)$ に基づき導出される．まず，$\{\boldsymbol{v}(t-1)\}_{\acute{I}} > 0$ を満たすボクセルの中で，$D_v(I,\acute{I})$ の値を最小とする $\acute{I}$ 番目のボクセルを探索する．次に，その時の $D_v(I,\acute{I})$ の値が有効探索範囲 T_v 内であれば，$\{\boldsymbol{v}(t)\}_I = \{\boldsymbol{v}(t-1)\}_{\acute{I}}$ と更新する．まとめると，以下の式に従って $\{\boldsymbol{v}(t)\}_I$ を更新する．

$$\{\boldsymbol{v}(t)\}_I = \begin{cases} 0 & \text{if voxel doesn}'\text{t exist or } D_v(I,\acute{I}) > T_v \\ \{\boldsymbol{v}(t-1)\}_{\acute{I}} & \text{if } D_v(I,\acute{I}) \leq T_v \end{cases} \tag{2}$$

$$\acute{I} = \underset{\tilde{I} \in \{\boldsymbol{v}(t-1)\}_{\tilde{I}} > 0}{\arg\min} \; D_v(I,\tilde{I}) \tag{3}$$

このラベリング手法は連結型ボリュームラベリングよりも高速に処理可能である．具体的には連結型の処理時間が 1 人の人物あたり 2.5ms 程度であるのに対し，探索型では 1.2ms 程度で処理可能であった．認識のリアルタイム性を考えても探索型ボリュームラベリングが有用であると言える．

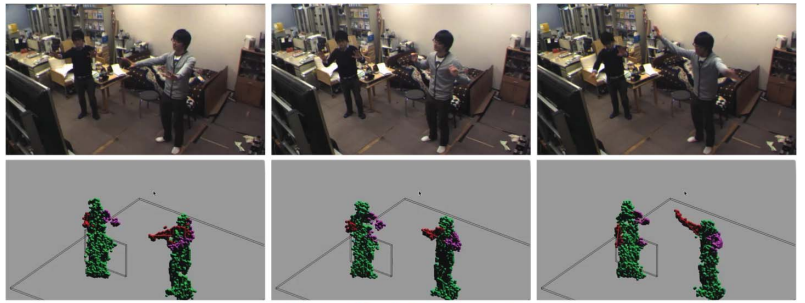

図 3　クラスタリング例
Fig. 3　Sample of classtering arms

なお対象空間内に人が入ってきた時や，片方の腕を構成するボクセル数が明らかに少ないなど，ラベリング結果が不自然だと判断した場合に連結型のボリュームラベリングを行い，それ以外のケースでは探索型ボリュームラベリングを行う．実行結果は**図 3** のようになる．

3.1.3　手先位置の推定

上記のようにしてクラスタリングされた腕に対して，仮定 2 に基づき，肩の位置を推定する．具体的にはユーザに提示するテレビの位置はセンシング領域内で既知とし，ユーザは操作時にテレビの方向に向いているという仮定を置くことで，ユーザの肩の位置は上半身の重心からテレビを結ぶ直線に対し垂直方向にあると考えられる．そこでユーザの重心の位置よりも低い位置にあるボクセルを除外したボクセル群の重心を求め，それを上半身の重心とする．このボクセルとテレビの位置を結ぶ直線に垂直な方向に一定距離進んだ位置にあるボクセルを両肩の位置と仮定する．仮定 3 を踏まえると手先は肩から一番遠い位置にあると考えることができる．このことから左右の肩に対応するボクセル群の中から一番遠い位置にあるボクセルをそれぞれの腕の手先とする．

3.2　Kinect を用いた方法

もう 1 つの方法は Kinect センサを用いた距離画像ベース骨格推定手法である．自然なインタラクションを利用したアプリケーションを育成するための開発者組織である OpenNI が公開している NITE ライブラリを用いて骨格推定を行う．推定の様子を**図 4** に示す．

Vol.2011-CVIM-177 No.4
2011/5/19

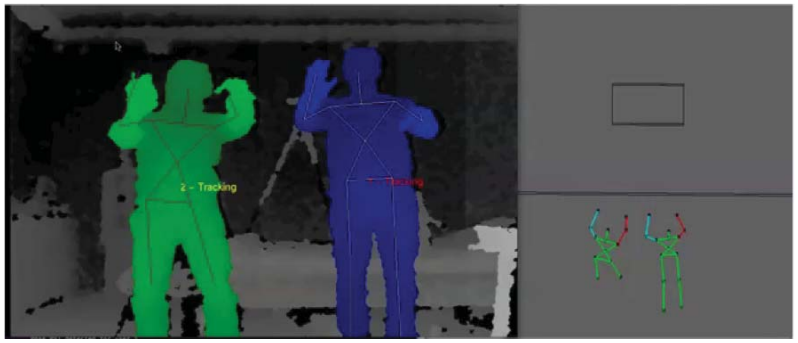

図 4 Kinect を用いた骨格推定
Fig. 4 Estimated Skelton using Kinect

前述の多視点カメラによる推定と比較すると，センシング領域が狭く，1 つの方向からの情報のみを用いた推定しかできないというデメリットがあるが，リアルタイム性の面などで優れている．

3.3 手先位置情報からのインタフェース入力変換

得られたユーザの相対的な手先位置情報とユーザにインタフェースを提示している画面の位置情報を用いてユーザの立ち位置に関わらず同様のジェスチャが同様のインタフェース入力となるようにした．それぞれの手先位置情報を提示部の画面に対応した 2 次元情報と手先と肩の距離情報が含まれる形式に変換する．インタフェース入力への変換イメージは**図 5** のようになる．

また複数同時入力するにあたって通信時のロスが少なくなるよう，タンジブルインタフェースやマルチタッチインタフェース等で用いられる Tuio[10] プロトコルを改変して用いた．このプロトコル規約に従い，センシング領域内にある手の数の文だけ**表 1** に示すデータを用意し，これをインタフェースの入力情報として一括して送信する．

4. ユーザ提示部

4.1 ユーザ提示部の要求仕様

本インタフェースはキーワード検索等を用いてある程度絞りこまれた動画群，もしくは同一動画内を時間軸でいくつかに分割した動画群の中からユーザの見たいシーンを探し出すことを目的としている．このようなインタフェースには以下のような要求が考えられる．

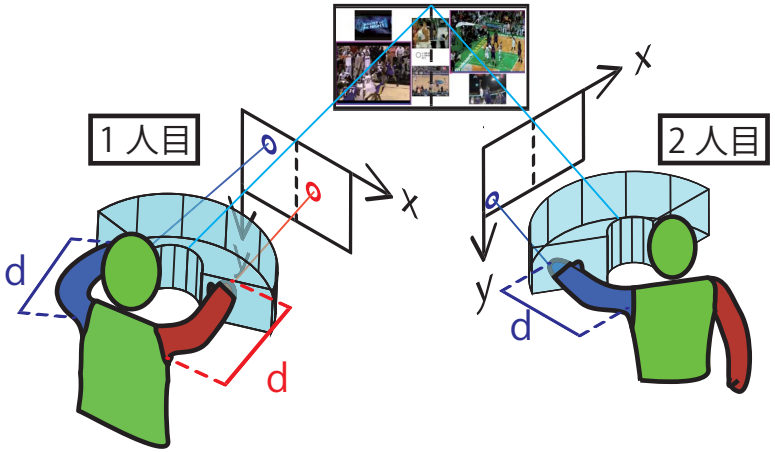

図 5 手先位置からインタフェース入力への変換のイメージ
Fig. 5 Image of converting hand position into interface input

表 1 変換後のインタフェース入力内容
Table 1 Information consists of interface input

変数名	内容
i	手の ID
x,y	各方向ごとのセンシング領域における手先の位置
X,Y	各方向ごとの手先の速度
m	各方向ごとの手先の加速度
d	肩と手先の距離

- 動画の片手操作
  動画を片手で操作する必要がある．シーン探索のために必要な操作を片手のみで切り替えることができ，かつ容易なジェスチャを用いて実現することが求められる．
- 同時操作可能なレイアウト
  同時操作をするにあたってそれぞれの操作対象と操作対象の候補となる動画群は同時に見れる必要がある．なぜならば動画群全体の構成を把握しながらシーン探索をする方が効率的であるからである．また二人で操作している場合相手がどの動画を操作している

かを把握できることも効率的なシーン探索に必要不可欠である．
また 1 人が 2 つの動画を操作することを想定しているので操作対象となる 2 つの動画は近接していることが，同時に内容を把握し，操作するうえで望ましいと考えられる．

- 操作対象の選択・切り替えの必要性
  動画群の中から操作対象の選択・切り替えをしながら目的のシーンを探索していく必要がある．そのため動画選択モードと動画操作モードのようなモード切り替えによる画面構成の変化などがあってはならない．また選択・切り替えのためのジェスチャが動画の片手操作に用いられるジェスチャと干渉しないよう注意する必要がある．

これらを踏まえて以下で動画の片手操作，全体のレイアウトと操作対象の選択・切り替え操作について言及する．

4.2 動画の片手操作

動画を片手で操作するためのレイアウトは動画の横に現在の動画の操作状況を示すコントローラ，動画の下にシークバーがあるような構成にし，外観は**図 6** のようになった．シーン探索のために必要な操作と考えられる再生・一時停止・早送り・巻き戻しの切り替えを横方向に手先を素早く振るジェスチャで行うようにした．手先の位置の情報を用いるのではなく，動きの速度に着目した操作としたため，手先位置推定の誤差にある程度対応し，ジェスチャインタフェースで常に問題となるジェスチャの開始の指定ができないという問題に対処した．また早送りと巻き戻しの速度を手先と肩の距離情報である d を用いてユーザが制御できるようにした．これによって容易かつ効果的な操作の実現ができた．

操作の簡略化，および直感性を向上させたことで，操作への慣れに応じてコントローラを見なくとも操作できるようになることが期待でき，1 人が 2 つの動画を操作することを前提としている本インタフェースに適していると言える．

4.3 操作画面のレイアウト

操作画面のレイアウトは**図 7** のようになった．ドック型のレイアウトをしており，操作対象の候補となる動画群は底に並べて表示され，ユーザはそこから操作対象を選択する．x 情報に応じた対象にフォーカスを当て，手先を振り上げるジェスチャを用いて選択をする．フォーカスを当てている対象には，入力ごとに異なる固有のマークが付けられており，複数ある操作対象のどれがマッピングされているかが容易にわかるようになっている．このマークは選択操作中だけでなく，動画操作中にも表示される．反対に選択解除動作には手先を振り下げるジェスチャを用いる．

同一ユーザの操作する動画は横に並べられ，ユーザごとに操作する動画の配置される高さが変化する．動画に付属するそれぞれのコントローラは操作の直感性や同時操作の妨げにならないようそれぞれ両端に配置されるようにした．

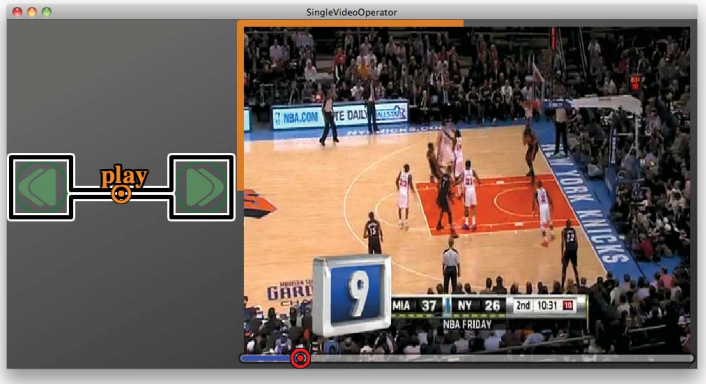

図 6 動画の片手操作モデル
画面中の動画は http://www.nba.com より
Fig. 6 Model for single video operation
The movie in this figure is from http://www.nba.com

また本インタフェースの使用は絞りこまれた動画のどこにユーザが探しているシーンが含まれているのかはわからないことを想定しているので，自動的に見ていない動画を選択する機能も盛り込んだ．動画操作状態から操作対象の選択解除をする際に所定の位置で解除ジェスチャを行うことで自動的にチェックされていない動画が操作対象として選択される．また一度見た動画をもう一度見直すことも十分考えられるので，操作状態から手先を上方に振ると一度見た動画の履歴を辿り自動でその動画を操作対象として選択するようにした．

5. 評 価 実 験

本研究で構築したインタフェースの評価をするため，以下の 2 つの検証実験を行った．

5.1 片手操作の操作性検証実験

1 つ目の実験は一般的な動画操作方法と本研究で提案する片手での動画操作方法との操作性の比較，および手先の三次元位置推定手法間での操作性の比較を目的とした．具体的には一般的な動画プレイヤーをリモコンで操作した場合と，本研究で提案する片手での動画操作に対して手先位置推定手法として多視点カメラシステムを用いた場合と Kinect を用いた

Vol.2011-CVIM-177 No.4
2011/5/19

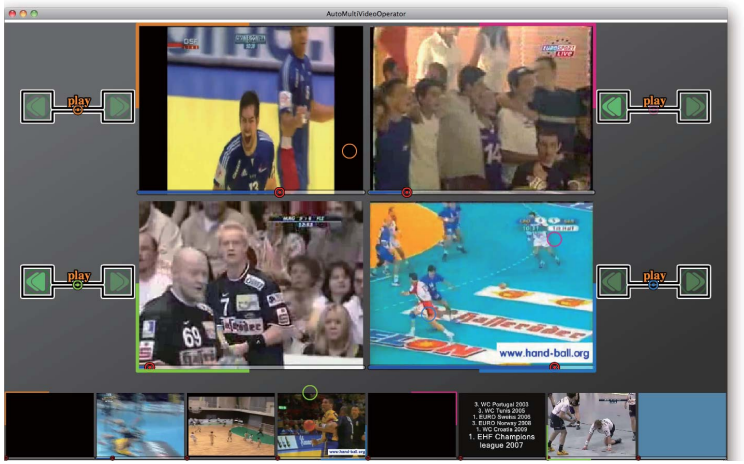

図 7 操作画面のレイアウト
画面中の動画は http://www.youtube.com より
Fig. 7 Layout of proposed interface
The movies in this figure are from http://www.youtube.com

場合で，同じタスクを処理するのにかかる時間を計測した．ここでタスクとは，5 分程度の短い動画の中に含まれる 10 秒から 20 秒程度のシーンを事前に提示しておき，指定した操作方法を用いて動画内から該当シーンを見つけてもらうというものである．実験に用いた動画は一貫した同じテーマがありながら背景などが異なる複数のシーンで構成されており，それぞれのシーンは他のシーンと比較すると大きな違いがあるため，一目で目的のシーンとわかるようにした．

実験は動画を 9 つ用意し，被験者 6 人に対して動画ごとに別の操作方法で同じシーンの探索をしてもらった．つまり 1 つの動画に対して 2 人ずつ同じ操作方法でタスクをこなしてもらったことになる．それぞれのタスク処理時間の平均を操作方法ごとに分類した結果は **表 2** のようになった．

上記の結果から操作方法ごとの処理時間を平均タスク処理時間で割ることによって相対タスク処理効率を算出した．操作方法 i で動画 j を操作した場合の平均タスク処理時間を r_{ij} とすると，相対タスク処理効率は式 4 のように表現される．算出した結果を **表 3** にまとめる．

表 2 平均タスク処理時間（秒）
Table 2 Average of processing time (sec)

操作方法	動画 1	動画 2	動画 3	動画 4	動画 5	動画 6	動画 7	動画 8	動画 9
リモコン	70	27	76	73	36	97	49	84	24
ボクセル	130	83	50	108	77	63	44	27	52
Kinect	140	150	18	30	94	69	31	84	38

$$E_i = 1/\sum_j \{r_{ij}/\sum_i r_{ij}\} \tag{4}$$

表 3 処理効率
Table 3 Efficiency

操作方法	リモコン	ボクセル	Kinect
相対タスク処理効率	1.06	0.95	0.99

この結果は操作方法によるシーン探索への影響が ±6% 程度しかなかったことを示している．これは 5 分程度の動画からのシーン探索であれば ±3 秒程度の影響である．つまり本研究で提案するハンドジェスチャによる動画操作方法は動画のシーン探索に用いるという観点において，一般的な動画操作方法と比較しても問題なく使える方法であることがわかった．また手先位置推定手法が多視点カメラを用いるのものと Kinect を用いるものとで比較しても，結果に差異がほとんどなく，速度に着目したジェスチャデザインによってそれぞれの推定手法間の差異が吸収されていることがわかった．

5.2 両手及び複数人操作の効率比較実験

2 つ目の実験では，本研究で提案する動画の同時操作が動画群からのシーン検索効率に与える影響を検証した．具体的には同一のタスクに対して操作方法が片手のみ，両手，二人同時に両手でと変わったときのタスク完了までの処理時間と動画操作数を計測した．タスクとは事前提示した 20 秒程度のシーンを，それぞれの長さが 1 分程度の 8 つの動画が含まれるセットの中から探してもらうというものである．どの動画に指定されたシーンが含まれているのかについては被験者は知らないこととする．探すシーンが含まれている候補となる 8 つの動画に関しては全く関連のないものではなく，それぞれの動画は一貫したテーマがありながら一目で区別のできるような複数のシーンで構成されており，現在見ているシーンが探しているシーンかそうでないかは見てわかるようになっている．実験では動画セットを 6 つ用意し，被験者 6 人に対して動画セットごとにそれぞれ別の操作方法で同じシーンを探

索してもらった．これにより動画セット 1 つにつき片手での操作 2 回，両手での操作 2 回，二人組での操作 1 回分のデータが集まった．

集めたデータから一分あたりの動画操作数を算出し，動画セット別に操作方法ごとの平均を算出した．またこの数値に対して操作方法ごとの平均をとることで操作方法別に平均化することで操作方法に対する効率を算出した．結果は**表 4** のようになった．

表 4 一分あたりの動画操作数
Table 4 Number of videos available to operate per minute

操作方法	動画群 1	動画群 2	動画群 3	動画群 4	動画群 5	動画群 6	平均
片手	4.1	5.3	4.7	4.6	6.4	7.4	5.4
両手	15.8	13.4	8.0	10.1	11.6	19.1	13.0
2 人（両手）	11.4	10.0	17.8	11.7	20.9	32.3	17.3

これは両手，二人とインタフェース入力の数が増えるに従って，タスクの処理効率がよくなっていることを示しており，本研究で提案するインタフェースが動画のシーン探索に有用であるということを示している．

6. 結　　論

本研究では，マニュアルでのシーン探索を効率的に行うことを目的とした，複数動画を同時に操作することが可能なインタフェースとして手先の三次元位置に注目したハンドジェスチャを用いるインタフェースを提案した．ジェスチャを認識するために環境設置型のセンサを用いて手先の三次元位置を推定した．この時，用いることのできるセンサは多数考えられるが，センシング方法が異なる場合でも同様に使えるインタフェースにすることも目的の 1 つとした．そのため本研究では多視点カメラを用いた手法と距離画像カメラである Kinect を用いた手法を用意した．1 人のユーザが複数の動画を同時に操作することを想定しているため，基本的な動画操作であると考えられる再生・一時停止・早送り・巻き戻しの切り替えを，手先の速度・肩との距離情報を用いるようにジェスチャを設計し，ある程度直感的かつ容易な操作を実現した．また，複数人で同時操作に対応できるようなユーザ提示部のレイアウトを構築した．

そしてこのインタフェースを評価するために 2 つの実験を行った．まず，1 つの動画を一般的な動画操作方法であるリモコンと片手で操作した場合での操作効率を比較し，片手の操作でもリモコンと同程度に操作できることを確認した．また，ジェスチャーマッピングの工夫によりセンシング方法に依らず，同様に操作できることも確認した．もう 1 つの実験で同時に操作する動画数によってシーン探索効率がどうなるかを検証し，複数の動画を同時に操作しながらシーン探索を行うことによって，効率的な探索が行えることを確認した．

参 考 文 献

1) Sminchisescu, C., Kanaujia, A. and Metaxas, D.: Conditional models for contextual human motion recognition, *Computer Vision and Image Understanding*, Vol.104, pp.210–220 (2006).
2) Schüldt, C., Laptev, I. and Caputo, B.: Recognizing Human Actions: A Local SVM Approach, *Proceedings of the 17th International Conference on Pattern Recognition*, Vol.3, pp.32–36 (2004).
3) Weinland, D., Ronfard, R. and Boyer, E.: Free Viewpoint Action Recognition using Motion History Volumes, *Computer Vision and Image Understanding*, Vol.103, pp.249–257 (2006).
4) Yilmaz, A. and Shah, M.: Recognizing Human Actions in Videos Acquired by Uncalibrated Moving Cameras, *Proceedings of the 10th IEEE International Conference on Computer Vision*, Vol.1, pp.150–157 (2005).
5) Fathi, A. and Mori, G.: Human Pose Estimation using Motion Exemplars, *Proceedings of the 11th IEEE International Conference on Computer Vision*, pp.1–8 (2007).
6) N., J.C. and Fei-Fei, L.: A Hierarchical Model of Shape and Appearance for Human Action Classification, *Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.1–8 (2007).
7) Coldefy, F. and Louis-dit Picard, S.: Digitable: an interactive multiuser table for collocated and remote collaboration enabling remote gesture visualization, *2007 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 1–8 (2007).
8) Zigelbaum, J., Browning, A., Leithinger, D., Bau, O. and Ishii, H.: g-stalt: a chirocentric, spatiotemporal, and telekinetic gestural interface, *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, ACM, pp.261–264 (2010).
9) Sagawa, Y., Shimosaka, M., Mori, T. and Sato, T.: Fast Online Human Pose Estimation via 3D Voxel Data, *Proc. of IROS2007*, pp.1034–1040.
10) Kaltenbrunner, M., Bovermann, T., Bencina, R. and Costanza, E.: TUIO - A Protocol for Table Based Tangible User Interfaces, *Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation (GW 2005)*, Vannes, France (2005).