

人を見るシステムにおける人物拡大追跡システムの構築

林 豊洋 馬場 功淳 江島 俊朗

九州工業大学

あらまし 我々は、「人を見る」システムに関する研究を進めている。研究プロジェクトはLPS(Looking at People System)と呼ばれ、人とコンピュータとの新たなインタラクションを目指している。LPSの中心となるシステムは、人物検出環境 HeadFinder である。本論文では、HeadFinderの検出する情報を活用したシステムの構築を試みる。具体的には、HeadFinder が検出した頭部を拡大追跡するシステム HeadTracker の構築を行う。手順としてまず第一に、HeadFinder の検出した情報に基づき首振りカメラを制御し、頭部を含む高解像度画像を獲得する。獲得画像は、頭部の一部がフレームアウトを起こす可能性のあるものや、背景を含んだものとなる。従って、獲得画像に対し HeadFinder に相当する処理を行い、獲得画像の精度向上を試みる。これらの手法を実装した HeadTracker は、頭部の高解像度画像の獲得に成功した。

Construction of Person Tracking System on Looking at People System

Toyohiro HAYASHI Naruatsu BABA Toshiaki EJIMA

Kyushu Institute of Technology

Abstract We are researching on the “Looking-at-People” system. A research project is called LPS(Looking at People System), aimed at new interaction between people and computer. The system lead in LPS is person-detection system HeadFinder. In this paper, we try construction of a system utilized HeadFinder. On a concrete target, we construct HeadTracker which tracks and zoom in the head detected by HeadFinder. The first procedure, control PTZ camera and acquisition high-resolution image containing a head. Acquisition image is containing “Head is about to be out of the frame”, “Background is included”. Therefore, we apply HeadFinder to high-resolution image, so as to improve the capability of getting a head image. HeadTracker which implemented these techniques succeeded in acquisition of high-resolution image of a head.

1 はじめに

現在我々は、「人を見る」システムに関する研究を進めている。画像中の人物の有無や向き等の認識のみではなく、人が笑っているのか怒っているのかなどの認識環境の構築を目指す。研究プロジェクトはLPS(Looking At people System)と呼ばれ、人とコンピュータとの新たなインタラクションを可能とするものである。

LPSの中心となるシステムである HeadFinder[1]は、人物の頭部を「実時間、高速、安定」に検出することが可能である。したがって、HeadFinder が検出した情報を活用したシステムを構築することにより、HeadFinder 単体では認識できない情報の獲得が可能となる。

これらの背景より、HeadFinder が検出した人物の頭部を、首振りカメラを制御し拡大追跡するシステムである HeadTracker を構築した。Head-

Tracker を用いることにより、頭部の拡大画像が獲得でき、LPSの目指す人の表情認識等が可能となる。

HeadTracker は、HeadFinder と UDP プロトコルによるデータ通信を行い、検出された頭部データを獲得する。獲得データより、認識した人物の3次元軌跡と移動量を算出する。また、首振りカメラの動作遅延に対応するため、人物の移動量の予測を行い、カメラの制御パラメータを決定する。

これらの手法を用いて実装した HeadTracker は、HeadFinder が検出した頭部の拡大追跡を行うことができるが、頭部の一部がフレームアウトを起こす場合や、背景の除去が行えない問題がある。そこで、獲得した拡大画像に HeadFinder に準じた処理を行う処理を加え、HeadTracker の改良を行う。

本論文は7章からなる。2章では、LPSとその中心である HeadFinder について述べる。3章では HeadTracker の概要について、4章で実装について述べる。5章では HeadTracker の改良点について

述べる。6章で評価と考察を行い,7章でまとめとする。

2 LPS:「人を見る」システム

LPSとは”Looking at People System”の略称であり,「コンピュータが人を見るシステム」をキーワードとして我々が研究開発を進めている人物画像理解プロジェクトである。人物追跡機能に加え,人物の表情やジェスチャなどの情報を統合的に判断してユーザの行動に応じたきめ細かな処理を行う事を目的としている。

LPSのシステム構成を図2に示す。

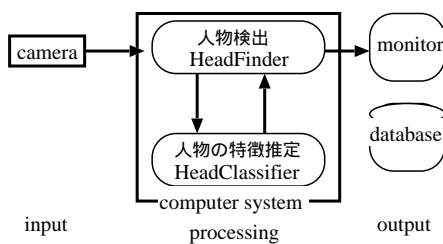


図1:「人を見る」システム構成図

現在,人物検出環境 HeadFinder,頭部の向きや男女を識別する環境 HeadClassifier 等の実装が進んでいる LPS の中心となるシステムは,人物検出環境 HeadFinder である。

2.1 HeadFinder

HeadFinder は,実時間で動作する人物の頭部を追跡するシステムである。HeadFinder は,「動画中の動く円形は人物の頭部である」という仮定で人物の頭部を検出している。動画像中からの処理対象の抽出にフレーム間差分,頭部の検出に円の Hough 変換を用いることにより頭部を検出する。特徴として,以下の点が挙げられる。

- 処理対象の抽出に,フレーム間差分を用いているため,環境の変化にロバストである
- 人物の頭部は,高速化した Hough 変換で検出するため,特殊なハードウェアなしで高速に動作する。

以上の手法により,HeadFinder は実時間で動作し,安定かつ高い検出能力を持つ。

2.2 獲得情報

HeadFinder が獲得可能な情報を以下に示す。

- 認識した頭部の重心 (x, y)
($0 \leq x \leq xwidth, 0 \leq y \leq yheight$)
- 認識した頭部の半径 r
($0 \leq r \leq 7$)
- 現在,画像中に人物がいるかの有無
($true, false$)

人物の頭部と認識情報との対応は,図2となる。

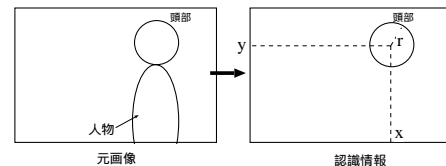


図2:人物の認識情報

3 HeadTracker: 頭部拡大追跡システム

2章で述べた通り,LPS の中心となるシステムである HeadFinder は,頭部検出をロバストかつ実時間で行う事が可能である。従って,HeadFinder の検出した頭部に関する情報を活用したシステムを構築する事により,新たな有用な情報の獲得が期待できる。以上の背景より,頭部拡大追跡システム HeadTracker を構築する。

3.1 概要

HeadTracker は,HeadFinder が検出した頭部を追跡し,頭部がズームアップされた高解像度画像として獲得するシステムである。ズームアップ画像の獲得には,パンチルトズームを自由に制御できる首振りカメラを用いる。なお,首振りカメラは,HeadFinder が用いる固定カメラとは別に用意する。

HeadTracker の処理の流れは以下となる。

1. HeadFinder からデータを受信する
2. 受信データから,人物の3次元軌跡を計算する
3. 3次元軌跡のデータから,首振りカメラの制御データを求める
4. 首振りカメラコントローラが,カメラに制御パラメータを送る

それぞれの実装については,4章で説明する。

3.2 拡大画像

HeadTracker は, HeadFinder が検出した人物を追跡し, 拡大画像の獲得を行うことを目的としている.

図 3 に, HeadFinder への入力画像と, 拡大画像を比較したものを示す.



図 3: 拡大画像との比較

図 3 における拡大画像は, 頭部を高解像度で獲得している. 高解像度の画像が獲得できることにより, 今後 LPS の構築に必要な機能の実装が期待できる.

人物の「向き」を判定することは, HeadFinder の入力画像と同等の解像度で可能である. しかしこの解像度では, 人物の詳細なテクスチャ情報が獲得できないため, LPS の目指している「人が笑っているのか怒っているのか」といった表情の認識や「この人物は誰なのか」といった人物の特定は難しい. 高解像度の画像を利用することにより, 上記の認識が可能になる.

さらに, 獲得した画像は 1 フレームごとに記録することが可能である. 画像を記録するタイミングで, 時間等のインデックス情報を付加することにより, 頭部画像データベースの構築が可能となる. データベースより, 「ある時刻に通過した人」といった検索条件での画像検索が可能となる.

4 HeadTracker の実装

HeadTracker のシステム構成を図 4 に示す.

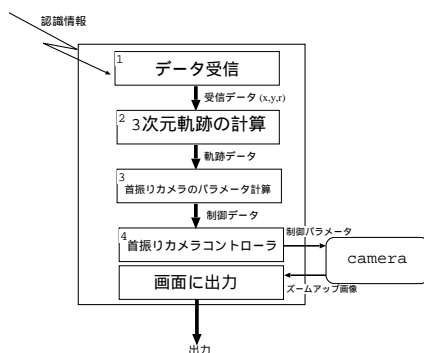


図 4: システム構成図

各モジュールについて, 以降説明する.

4.1 HeadFinder からのデータ受信

HeadTracker は第 1 段階として, HeadFinder が獲得した人物のデータを受信する必要がある. 受信部分の実装について述べる.

通信方式

アプリケーション間のデータ通信の方式は様々あるが, HeadTracker において必要な条件は, 以下の点が挙げられる.

1. HeadFinder とは別の計算機で実行可能
2. 通信プログラムが HeadFinder に負担をかけない

1 番目の条件を満たすためには, データの通信に TCP によるソケット通信, UDP によるデータグラム通信を用いれば良い. これらの方式を用いることで, TCP/IP で接続された計算機を用いることが可能になる.

2 番目の条件を満たすために, 今回は UDP を用いた. TCP と比較して, UDP の有利な点を以下に示す.

- TCP と比較して, 通信手順が単純
- クライアントが強制的に終了した場合, サーバに直接影響が出ない

UDP は, パケットの再送などの処理がないため, TCP と比べて通信にかかる処理コストが低い. さらに, クライアントが強制終了した場合, サーバ側の処理が必要ない.

受信データ

HeadFinder が獲得したデータは, 1 フレームごとに送信される. 同様に, HeadTracker は送信されているデータを, 1 フレームごとに受信する.

HeadTracker が受信するデータを表 1 に示す.

変数名	変数の意味
x	頭部の重心 x
y	頭部の重心 y
r	頭部の大きさ (8 段階)
$flush\ flag$	人物を検出しているかの有無

表 1: UDP で受信するデータ

これらのデータを受信することにより, 1 フレームごとに「人物がいるか」, 「どこに頭部があるか」という情報が獲得可能となる.

4.2 3次元軌跡の計算

HeadFinderの検出した情報を受信した後、認識した人物の3次元軌跡を計算する。軌跡を求めることにより、人物がどの方向に動いているかを知ることが可能となる。

HeadFinderの検出データのみを用いた軌跡

人物が動いた方向を知るには、3次元の座標データ (x, y, z) が必要となる。しかし、HeadFinderは単眼カメラを用いたシステムであるため、頭部の重心である (x, y) を計算することはできるものの、奥行き情報 z は検出できない。

そこで、HeadFinderが計算した頭部の大きさ r を、奥行き情報の代替として用いる。 r を用いることにより、以下の相似関係から実際の奥行きである z を求めることが可能となる。

$$z : R = f : r \quad z = fR/r$$

(z :実際の奥行き f :カメラの焦点距離 R :頭部の実際の半径 r :計算した大きさ情報)

ここで、 f は既知である。また、 R の個人差を無視できるものとすれば、 $z \propto r$ となる。以後、人物の座標を、ベクトル $X = (x, y, r)$ と表現する。

r が奥行き情報として用いることが可能であることを検証した実験結果を示す。実験は、

1. カメラと平行に、近い (3.0m), 中間 (4.0m), 遠い (5.0m) を歩く
2. カメラの前で“8の字”に歩く

の2種類を行った結果を図5に示す。

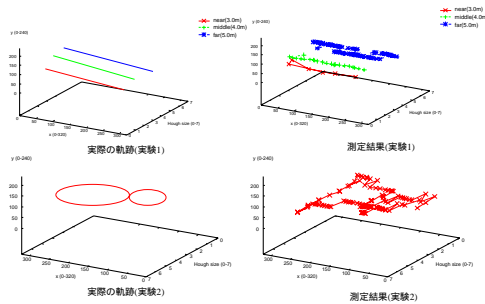


図5: r を奥行き情報として用いる

結果は、 r を用いることにより3次元軌跡を表現できることを示しているといえる。

頭部軌跡の平滑化

軌跡 $\{X_t; t = 0, 1, \dots, n\}$ は、HeadFinderが人物の位置を検出している場合 ($flushflag = 0$), X を用

いて求め、さらに1フレームで人物が動いた変化量と加速度を求める。

ただし、頭部の大きさ r に関しては、値を8段階しか取らないため、HeadFinderから受信した値をそのまま用いた場合、値に“ぶれ”が生じてしまう。値の“ぶれ”が生じる例として、「頭部の大きさが n と $n+1$ の中間」といった場合があげられる。このとき、 r は n または $n+1$ のどちらかに決定される。

この問題を解決するため、 X に関して、過去数フレーム間の平均を取り、スムージング処理を施す。スムージング処理によって、軌跡が滑らかに表現できる。スムージング処理を行った場合と行わない場合の軌跡を図6に示す。

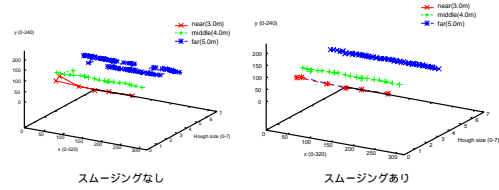


図6: スムージング処理

スムージング処理を施した後、フレーム間の差分と加速度を求める。各データは、表2に示した式で求める。

変数名	変数の意味	式
\bar{X}	X の n フレーム間平均値	$\bar{X} = \frac{1}{n} \sum_{i=0}^{n-1} X_{(t-i)}$
X'	X の t における変化量	$X' = X_t - X_{t-1}$
X''	X の t における加速度	$X'' = X'_t - X'_{t-1}$

表2: 頭部軌跡

4.3 首振りカメラの制御

頭部軌跡の情報に基づき、首振りカメラの制御を行う。

軌跡のフィルタリング

軌跡データから、首振りカメラの制御データへの変換を行う場合、もっとも単純な手法は、以下の手順である。

1. 獲得座標の変化量が $X' = (x', y', r')$ と計算される
2. その座標を基に制御する

この手法の欠点は、HeadFinderから受信したいかなる情報も信用してしまう点にある。 X' が非常に大きい場合は、HeadFinderがノイズの影響等により検出エラーを出し、誤情報を送信している可能性が高い。この変化量を用いた場合、首振りカメラの機械的な動作が追いつかない可能性がある。また、 X' が非常に小さい場合は、平滑化した軌跡では

なく,HeadFinder から受信したそのものの値を用いた方が精度が良い。

したがって,変化量に閾値を設け,軌跡のフィルタリングを行う。表3に制御に用いる値を示す。

条件	制御に用いる値
$T_s < X'_t < T_m$	X_t
$X'_t > T_m$	$X_{t-1} + X'_{t-1} + X''_{t-1}$
$X'_t < T_s$	X_t

表3: 制御に用いる値

首振りカメラの物理遅延への対応

前説で用いた制御データの計算を用いることにより,首振りカメラの制御が可能となる。しかし,制御対象の首振りカメラは物理的なデバイスであるため動作遅延が生じる。首振りカメラに命令を送信し,動作が完了するまで,約100msの動作時間を要する。

HeadFinderの処理速度は,1フレームあたり約50ms(20fps)であるため,動作の遅延は,以下の式で求めることができる。

$$\frac{100ms(\text{首振りカメラの動作時間})}{50ms(\text{HeadFinderの処理速度})} = 2 \text{ フレーム}$$

そこで,前節で求めた制御データに,2フレーム先のデータを予測したものを加える。ここでの予測値は,「1フレーム後の変化量 = 直前フレームからの変化量 + 変化量の加速度」である。

この予測値を2フレーム分考慮することにより,物理的な遅延に対応できる。時刻 t における, n フレーム先のデータを予測した場合,予測値は以下となる。

$$\begin{aligned} \hat{X}_{t+n} &= X_t + \sum_{i=1}^n \hat{V}_{t+i} \Delta t \\ \hat{V}_{t+i} &= \hat{V}_{t+i-1} + \alpha_t \Delta t \\ (\hat{V}_{t+0} &= V_t) \end{aligned}$$

これらの式を,実際のデータに対して用いた式を,以下に示す。

$$\begin{aligned} \hat{X}_{t+2} &= X_t + \hat{V}_{t+1} \Delta t + \hat{V}_{t+2} \Delta t \\ \hat{V}_{t+1} &= X'_t + X''_t \Delta t \\ \hat{V}_{t+2} &= \hat{V}_{t+1} + X'_t \Delta t \end{aligned}$$

4.4 予測データの送信方法

前節の手順で計算された予測データを,首振りカメラに送信する。首振りカメラと計算機はシリアルポートで接続し,データ通信のプロトコルには,SONYの首振りカメラを制御することが可能なVISCAを用いる。

VISCAプロトコルによる首振りカメラの動作手順は,以下の2種類である。

1. 動作命令送信後カメラが動作し,動作完了後に終了通知される
2. 動作命令送信後通知が返答され,次の命令まで動作を続ける

1番目の手順は,指定したカメラパラメータに対してカメラが動作する。正確なカメラ制御が可能な反面,カメラが動作している間,プログラムが一時停止を起こす。

一時停止は,カメラの制御部分を割り込みやスレッドとして実装することにより対処する事が可能である。しかし,これらの実装は動作遅延が大きいため,HeadTrackerへの実装は適さない。従って,制御データの送信は2番の手順を採用する。

2番目の手順は,指定した方向と速度に基づいてカメラが動作する。送信通知の返答が1番目の手順より早いため,プログラムは停止しない。ただし,カメラパラメータを直接指定できないため,制御に工夫が必要となる。カメラを指定の座標へ制御する手順は,以下である。

1. 首振りカメラの現在のカメラパラメータを獲得する
2. 移動したい座標との差分を計算する
3. カメラの移動速度を以下の式で決定する

$$speed = maxspeed \times pow\left(\frac{abs(moveval - curval)}{maxmoveval}, c\right)$$
 $maxspeed$: 首振り可能な最大速度
 $maxmoveval$: 移動できる最大量
 $moveval$: 移動したい座標
 $curval$: 現在のカメラパラメータ
4. 計算した速度を首振りカメラへ送信する

5 HeadTrackerの改良

4章で述べた手法を用いることにより,頭部の拡大追跡を行う事が可能である。ここで,より精度の良い拡大画像を獲得するため改良を行う。

5.1 獲得画像の問題点

HeadTrackerが獲得する拡大画像には,以下の問題点がある

1. 追跡頭部の一部がフレームアウトしており,完全な頭部画像ではない
2. 画像中に背景部分が多く含まれる

追跡頭部のフレームアウト

追跡頭部のフレームアウトとは、頭部の左右の一部が獲得画像からはみ出してしまう場合を指す。図7に例を示す。追跡頭部のフレームアウトは、頭部



図 7: 追跡頭部のフレームアウト

が移動したにもかかわらず、検出情報が更新されない場合に生じる。HeadFinder が検出できないゆっくりした速度で頭部が移動した場合や、UDP のデータグラムが Loss を起こし、HeadFinder からの情報を受信できなかった場合、検出情報は更新されない。

HeadTracker は、HeadFinder の検出情報のみを利用しカメラ制御を行い、頭部をズームアップして撮影する。検出情報が更新されない場合、ズームアップを行う座標は変化しない。したがって、頭部の一部がフレームアウトを起こす。

背景の存在

HeadTracker の獲得した拡大画像は、頭部以外に背景が存在する。(図8) 背景が含まれた拡大画像は、3



図 8: 背景の存在する拡大画像

章で述べた「表情認識」「人物同定」を実現する際に用いにくい画像である。可能な限り、背景部分の除去を行う必要がある。

5.2 HeadFinder の拡大画像への適用

前節で述べた問題への対策として、HeadTracker が獲得した拡大画像に対し、HeadFinder の頭部検出と同様の仕組みを組み込む。改良後のシステムブロックを図9に示す。

改良後の HeadTracker は、固定カメラ (camera0) を用いる HeadFinder(HF0) から検出された人物の軌跡と、首振りカメラ (camera1) を用いる HeadFinder(HF1) の検出した情報を利用する。なお、HF1 は拡大画像に対して適用される。

HF0, HF1 の検出情報は controller で処理され、適切な首振りカメラの制御を行う。controller が HF1

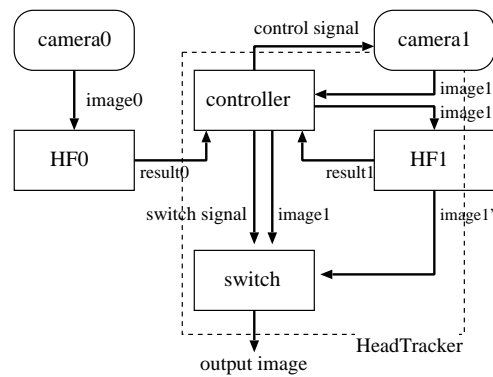


図 9: HeadTracker システムブロック

の検出情報を利用し、首振りカメラの制御を行うことにより、頭部がフレームアウトを起こしそうな場合これを回避する事が可能となる。

さらに、HF1 が頭部を検出した場合、switch が頭部を中心とした拡大画像を選択し、背景の除去された画像が獲得可能となる。

HF0 と HF1 の機能的な相違点は以下である。

- Hough テンプレートの画像サイズに対する比率
- 入力画像の解像度

拡大画像中の頭部は、HF0 への入力画像と比較して大きいと言える。従って、HF1 の Hough テンプレートの画像サイズに対する比率は HF0 と比較して大きい。テンプレートサイズを大きくした場合、頭部以外の物を検出する可能性があるが、頭部拡大画像に関しては、存在する円の最大候補は人物の頭部となる。

また、Hough 変換の計算コストは、テンプレートサイズの大きさに比例する。拡大画像に対して Hough 変換を行うため、テンプレートサイズは大きく設定することになる。ここで、HF1 が HF0 と同様の画像サイズに対して Hough 変換を行うと、計算量が膨大になる。そこで、HF1 が用いる入力画像は、半分の大きさに縮小したものをを用いる。頭部拡大画像に対して画像の縮小を行うため、円形の物体が潰れるといった問題は生じない。

5.2.1 controller を用いたカメラ制御

controller には、HF0, HF1 が検出した頭部情報が送信される。受信する情報は、以下の 4 パターンとなる。

1. HF0, HF1 両方が人物を検出していない
2. HF0 が人物を検出している
3. HF1 が人物を検出している

4. HF0, HF1 の両方が人物を検出している

2,4 のパターンに該当する場合, 人物が移動していると考えられるため, 改良前の HeadTracker と同様の拡大追跡を行う。

3 のパターンに該当する場合, フレームアウトを起こす可能性がある。したがって, HF1 が検出した頭部の座標が画面の端に近い場合, 中心に頭部が来るようにカメラを制御し, 補正を行う。



図 10: カメラの配置

5.2.2 switch を用いた獲得画像の選択

switch を用いて, 可能であれば拡大画像中の背景領域を除去する。

switch には, 拡大画像 image1 と, 背景が除去可能を示す switch signal が送信される。switch signal は HF1 が人物を検出している時に除去可能を示す状態となる。この場合, HF1 が検出した頭部の中心座標と大きさを用いて, 頭部付近の画像のみを抽出し, 出力する。

6 評価, 考察

4 章の手順で実装した HeadTracker の性能, 5 章の手順で行った改良に対する評価と考察を行う。

6.1 実験装置

実験は表 4 に示した環境の計算機システムで行う。HeadFinder, HeadTracker 双方を同一の計算機上で動作させる。表 4 の環境において, HeadFinder

計算機	AT 互換機
OS	FreeBSD-4.3
CPU	PentiumIII@733MHz-SMP
HF0 取り込み画像サイズ	300 × 200
HF1 取り込み画像サイズ	150 × 100

表 4: 動作環境

のフレームレートは約 20FPS, HeadTracker のフレームレートは 10FPS と実時間での動作を実現している。なお, HeadTracker のフレームレートは, 首振りカメラとのデータ通信速度 (9600bps) に依存するため, 現在の実装では 10FPS が上限となる。

また, HeadFinder で利用する固定カメラと HeadTracker が利用する首振りカメラの配置は, キャリブレーションが最小限で済む配置とする。図 10 に, 実際用いたカメラを示す。図 10 に示したカメラ配置とすることにより, キャリブレーションは撮影される中心点の調整のみで完了する。なお, 固定カメラには Canon 社の VC-C4, 首振りカメラは SONY 社の EVI-G20 を用いた。

6.2 評価方法

評価は, 以下の 2 つについて行う。

1. HeadTracker の性能
2. HeadTracker 改良の効果

HeadTracker の性能評価として, 以下の 2 つのシーンでの頭部追跡性能と, 予測の効果を測定する。

会話シーン 人が会話している状況を撮影 (カメラの前を低速で移動している状況)

室内環境 室内で, 人が歩いている状況を撮影 (カメラの前を比較的高速で移動している状況)

予測効果 予測するフレーム数を 0 から 3 まで変化させ撮影 (カメラの前を低速で移動)

評価基準は, 以下の式で追跡成功率として定義する。

$$\text{追跡成功率} = \frac{\text{HeadTracker の頭部撮影枚数}}{\text{HeadFinder(HF0) の頭部撮影枚数}}$$

改良後の HeadTracker の評価として, 以下の効果が現れているかを測定する

- 頭部がフレームアウトを起こす事象において, 適正なカメラ制御を行う
- 拡大画像に対し, 背景を除去する

6.3 実験結果

6.3.1 HeadTracker 性能

会話シーン

会話シーンは, カメラの前で会話をしている人に, 実験とは告げずに撮影した。表 5 に, 会話シーンを撮影した場合の追跡成功率を示す。

HeadTracker 撮影成功枚数	416/460 枚
追跡成功率	90.4%

表 5: 会話シーン

室内環境

研究室室内を、実験とは告げずに撮影を行った。表 6 に、室内環境を撮影した場合の追跡成功率を示す。

HeadTracker 撮影成功枚数	2237/2416 枚
追跡成功率	92.6%

表 6: 室内環境

予測の効果

図 11 に予測フレーム数ごとの追跡成功率を示す。図の横軸は予測フレーム数、縦軸は追跡成功率である。

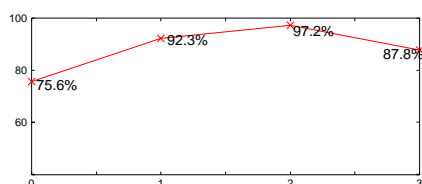


図 11: 予測の考慮

6.3.2 改良の効果

フレームアウトへの対応

改良後の HeadTracker において、HF1 のみが人物を検出する速度で頭部を動かし、頭部が含まれた画像数を表 7 に示す。

撮影枚数	360 枚
頭部撮影成功枚数	345 枚

表 7: フレームアウトへの対応

背景の除去

switch と HF1 の検出情報を用いて背景の除去を行った出力画像を図 12 に示す。

6.4 考察

6.4.1 HeadTracker の追跡成功率

HeadTracker は、HeadFinder の検出した人物を高い割合で追跡しているといえる。追跡が成功した頭部は、HeadFinder が獲得した頭部の画像と比較して、高解像度である。図 13 に、HeadFinder が獲得した頭部画像と、HeadTracker が獲得した頭部画像を示す。

また、予測の効果を検証した実験は、検出した人物の移動速度と加速度を考慮した予測が有効であることを示している。予測が、カメラの動作遅延を吸収したといえる。実験では、予測を 2 フレーム分を行った場合が、最高の成功率であった。これは、理論的に求めた遅延フレーム数と等しい。



図 12: 背景の除去



HeadFinderの認識画像 HeadTrackerの認識画像

図 13: 高解像度画像の獲得

6.4.2 HeadTracker 改良の効果

HeadTracker は、拡大頭部がフレームアウトを起こす、画像に背景が含まれる等の問題点があるため、拡大画像に対し HeadFinder の処理を行う手法を加え、改良を行った。

拡大画像に適用する HeadFinder(HF1) は、入力画像の解像度を小さくする事により、計算コストを低減している。HF1 は、HF0 では人物を検出できない事象において効果を現す。具体的には、人物が低速な運動を始めた時にこのような事象となる。

実験結果より、controller が適切に首振りカメラを制御し、頭部のフレームアウトを低減しているといえる。しかし、HF1 が人物の肩を頭部と誤認識する場合は若干 (360 フレーム中 15 フレーム) 見受けられた。

また、switch が HF1 の検出情報を利用し背景の除去を行い、頭部のみの画像を獲得している。背景の除去された画像は、図 12 に示した通りである。

7 まとめ

本研究では、頭部拡大追跡システム HeadTracker の構築を行った。HeadTracker は首振りカメラを用いて、HeadFinder の検出した人物の頭部を追跡することに成功した。また、追跡頭部のフレームアウトや、背景が画像中に含まれる問題は、拡大画像に対して HeadFinder の処理を適用することにより改善された。なお、現在 HeadFinder は複数人の人物検出が可能となっている。複数台の首振りカメラを用いて複数人の頭部を拡大追跡への対応を行えば、さらに活用範囲は広がると考えられる。

参考文献

- [1] 馬場功淳, 大橋健, 乃乃司, 松尾英明, 江島俊朗 “HeadFinder: 単眼視動画像を用いた複数人追跡” 画像センシングシンポジウム, SSII2001, pp.363–368, 2001
- [2] 林 豊洋, 馬場功淳, 江島俊朗 “HeadFinder を用いた頭部の 3 次元軌跡推定とその可視化” 電気関係学会九州支部連合大会, p.1128, 2000