オプティカルフローを用いた移動手振り検出

浅野　秀胤† 　織茂　達也† 　寺林　賢司†† 　太田　睦† 　梅田　和昇†††

† パイオニア株式会社　〒 212–0031 神奈川県川崎市幸区新小倉 1–1
†† 静岡大学　〒 432–8561 静岡県浜松市中区城北 3–5–1
††† 中央大学　〒 112–8551 東京都文京区春日 1–13–27
E-mail: †{hidetsugu_asano,tatsuya_orimo,mutsumi_oota}@post.pioneer.co.jp, ††tera@eng.shizuoka.ac.jp,
†††umeda@mech.chuo-u.ac.jp

あらまし　カメラを用いたジェスチャインタフェースとして，移動しながらの手振りを認識可能な手法を提案する．オプティカルフローによる映像内の動き算出を行い，フローの追跡結果に三次関数を当てはめ，手振りと他の動きを分離する．この残差を周波数解析することにより，移動しながらの手振りを検出する．本手法により，ながら操作など，より自由度の高いジェスチャインタフェースの実現が可能である．

キーワード　画像処理，ジェスチャインタフェース，オプティカルフロー，フーリエ変換

1. はじめに

人間にストレスを与えない自然なヒューマン・マシン・インタフェースとして，様々なジェスチャインタフェースが提案されている [1], [2]．一般に，これらの手法では，ユーザはカメラの前に正対した上で，ジェスチャを行う必要があり，使いやすさに問題があった．これに対し我々は，部屋の中に数台のカメラを設置して操作者の手振りを検出し，家電を操作するシステムを提案している [3], [4]．これにより，ユーザは部屋のどこからでも機器操作を行うことが可能になった．しかし，この手法では静止して手を振り続けることが前提となっており，意図せず手振り位置が変化した場合に検出処理がやり直しになる，歩きながらなどの「ながら操作」ができない，といった課題が残されていた．

本研究では，ユーザが移動しながら行ったジェスチャも検出可能な手法を提案する．オプティカルフローを用いて映像内の動きを追跡し，ユーザの移動などに起因する動きを推定・除去し，残差を周波数解析することにより手振りを検出する．

2. アルゴリズム

本研究の目的は，ユーザがマーカなどを身につけることなく，ユーザの位置や姿勢，カメラの位置を制限せずに移動手振り検出を実現することである．このとき，カメラの設置状況や使用環境が不定であるため，ユーザの写り方を限定できない．このため，人物の姿勢推定を利用した手法の構築は困難である．そこで本手法では姿勢推定などを行わず，映像内のオプティカルフローから直接手振りを検出する．オプティカルフローは，ユーザの手振りとユーザの移動の複合で起きるため，これらを推定し，分離することで移動手振り検出は実現可能である．

本手法は以下の手順で移動手振り検出を行う．

（1） 映像からのオプティカルフロー算出

（2） 胴体や腕の動きに起因するフローの推定と，その成分の除去

（3） 周波数解析による検出

以下にそれぞれの詳細について述べる．

2.1 オプティカルフロー算出

オプティカルフローの求め方は種々提案されているが，本稿では以下の 4 つについて比較検討する．

- Block Matching

入力画像と参照画像で最も類似する箇所を画素毎に探索する．また，この際に近傍画素同士のフローが類似した方向を持つよう，大域最適化を行う．大域最適化手法としては GraphCut [5] を用い，繰り返し処理により最適化する．エネルギー式は (1)〜(3) を用いる．

$$E = \sum_{p} C(p,\nu) + \lambda_{BLOCK} \sum_{p} V(\nu,\nu_N) \tag{1}$$

$$C = \sum_{BLOCK} \sum_{RGB} (|p_{in} - p_{ref}|) \tag{2}$$

$$V = \sum_{N} \min(||\nu - \nu_N||^2, \sigma^2) \tag{3}$$

C はデータ項，V は平滑化項，λ_{BLOCK} はデータ項と平滑化項の重み付け係数である．p は画素値，ν はフロー，N は注目画素の 4 近傍画素を示す．p_{in} と p_{ref} は入力画像と参照画像の画素値を表し，ν_N は注目画素の 4 近傍画素のフロー，σ^2 はフローの二乗誤差の最大値である．データ項は画素毎に RGB 値の絶対差分和を，平滑化項はフローの二乗誤差を用いている．

- AD Census

このフロー算出は，Mei ら [6] による，ステレオマッ

チングで用いられているコスト算出手法を元にしている．画素値の絶対差分（AD）とセンサス構造 [7] の違い（Census）を手がかりとして用いている．Block Matching と同様に，GraphCut による大域最適化を行う．用いたエネルギー式は (4)〜(7) の通りである．

$$E = \sum_p C(p) + \lambda_{AD\_census} \sum_p V(\nu, \nu_N) \quad (4)$$

$$C = (1 - exp(-\frac{C_{census}}{\lambda_{census}})) + (1 - exp(-\frac{C_{AD}}{\lambda_{AD}})) \quad (5)$$

$$V = \sum_N \min(||\nu - \nu_N||^2, \sigma^2) \quad (6)$$

$$C_{AD} = \frac{1}{3} \sum_{RGB} |p_{in} - p_{ref}| \quad (7)$$

C_{census} は注目画素近傍のセンサス構造のハミング距離を取る．本稿では，注目画素を中心に 9×7 の範囲でセンサス構造を求める．C_{AD} は注目画素の RGB 値の絶対差分和である．また，λ_{AD_census} はデータ項と平滑化項の重み付け係数で，λ_{census} と λ_{AD} は AD と Census の重み付け係数である．

- Moving Gradients

Mahajan ら [8] による中間画像生成手法で用いられているコスト算出手法を元に，RGB 値と勾配の，二乗誤差を重み付け加算したものをデータ項としている ((8)〜(10))．

$$E = \sum_p C(p) + \lambda_{MG} \sum_p V(\nu, \nu_N) \quad (8)$$

$$C = (||\nabla p_{in} - \nabla p_{ref}||^2 + 0.5 \times ||p_{in} - p_{ref}||^2) \quad (9)$$

$$V = \sum_N \min(||\nu - \nu_N||^2, \sigma^2) \quad (10)$$

λ_{MG} はデータ項と平滑化項の重み付け係数で，∇p は画素値の勾配を表す．この手法についても大域最適化を行う．Mahajan らは画像全体の輝度変化に対応するため，近傍画素の画素値の分散を用いてデータ項を正規化し，α 乗している．しかし，ジェスチャインタフェースで用いる場合，画像の急激な変化はあまり起こらず，正規化したことによりベクトルの誤りが増加する可能性があるため，デメリットが大きいと考えられる．そこで，本稿では正規化等を行わず，式 (9) を用いるものとする．

- Classic+NL

Sun ら [9] によるオプティカルフロー算出手法である．輝度値の差分と，周囲のフローとの差分を元に勾配法で最適化を行い，結果を重み付けメディアンにより平滑化している．

2.2 胴体や腕の動きの推定とその成分の除去

移動しているユーザの手振りを検出するためには，手振り動作によるフローと，ユーザの移動などによるフローを分離する必要がある．ユーザが移動している際に，胴体・足・腕それぞれの箇所 (図 1) で発生するフローを，1 秒間追跡した様子を図 2 に示す．この図に見るように，ユーザが移動した時の胴体や手・足など，箇所により発生するフローの特徴は異なっている．移動中の胴体で発生するフローはほぼ線形で，手や足などは三次曲線で近似可能な形状を取り，いずれも 1 秒程度の時間内であればあまり複雑な動きをしていないことが分かる．次に，ユーザが移動しながら手を振っている場合と，静止して手を振っている場合の 2 通り (図 3) について，手振り箇所でフローを追跡した結果を図 4 に示す．静止状態での手振りでは周期的なフローが発生しているのに対し，移動手振りでは人物の移動により発生するフローの上に，手振りによるフローが高周波成分として重畳されていることが分かる．

図 2，4 から，人物の移動などに起因する動きは，1 秒程度で考えれば三次関数での近似で除去可能で，手振りはより高い周波数で現れるため影響を受けないと仮定できる．そこでここでは移動に起因するフローの推定として，最小二乗法による三次関数推定を行う．推定結果の残差を図 5，6 に示す．手振りが行われていない時は誤

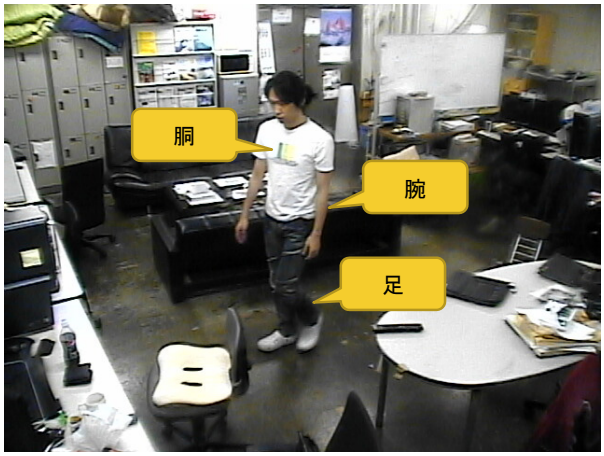


図 1　追跡を行った箇所

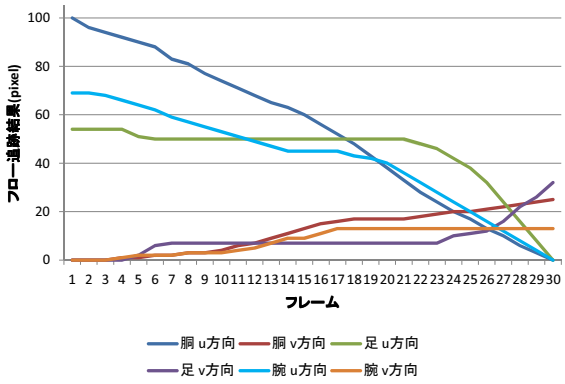


図 2　歩行時の人体に発生するフローの追跡結果．追跡した位置のうち，最小の値を 0 としてプロットした．

(a) 移動しながらの場合　(b) 静止している場合

図 3　手振りの追跡を行った箇所

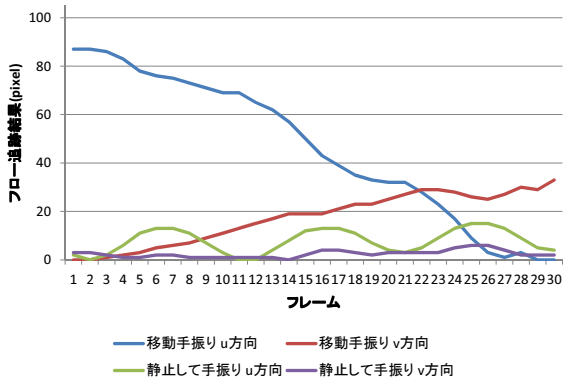


図 4　手振りが行われている箇所のフローの追跡結果．手振りは横 (u) 方向に行われている．

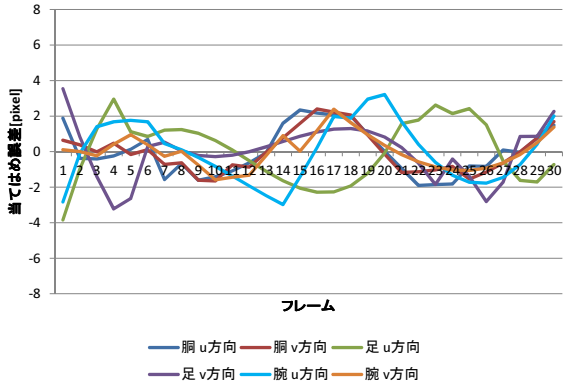


図 5　人体の追跡結果に三次関数を当てはめた残差

差が小さくなる一方で，手振りの周期的なフローは影響を受けない．ここから，移動によるフローと手振りによるフローを分離できていることが分かる．

また，腕の動きの除去についても同様の処理で対応可能である．

2.3　周波数解析，閾値算出，検出

手振りの有無を判断するため，残差を周波数解析する．図 5 の胴体の残差と図 6 の移動手振りの残差の u・v 方向それぞれでフーリエ変換を適用した結果を図 7 に示す．移動手振りが行われている場合については，手振りの周波数に対応した，3Hz にピークが発生している．移動時の胴体については 2Hz に弱いピークが発生しているが，これは推定誤差に起因すると考えられる．推定誤差

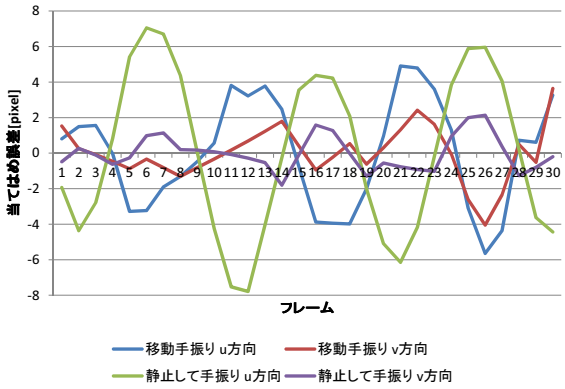


図 6　手振りに三次関数を当てはめた残差

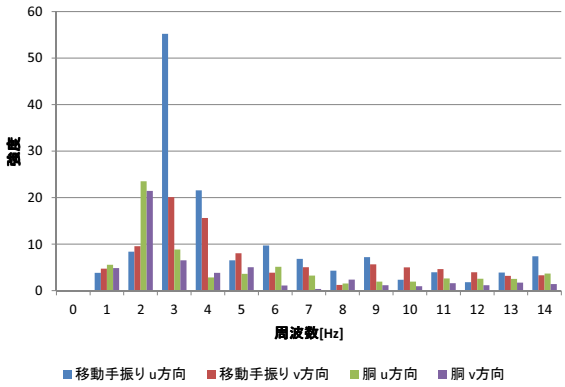


図 7　残差にフーリエ変換を適用した結果

が 2Hz にピークを持つ場合が多いことから，解析対象とする周波数は 3〜6Hz とする．ここで，6Hz は手振りが取り得る速さの上限である．

図 5 の例で，手振り以外の動きは残差が小さくなった．しかし，画面内の移動量が大きい場合は，残差の持つ値 (振幅強度) も大きい値をもつ．このため，検出に振幅強度を用い，単純な閾値処理を行うと，動きが大きい場所を全て検出するおそれがある．そこで，画面内全ての残差の振幅強度を用い，閾値が適応的に変化するよう式 (11)，(12) を用いた．

$$A_{max} = \max_{3\sim 6Hz} A \tag{11}$$

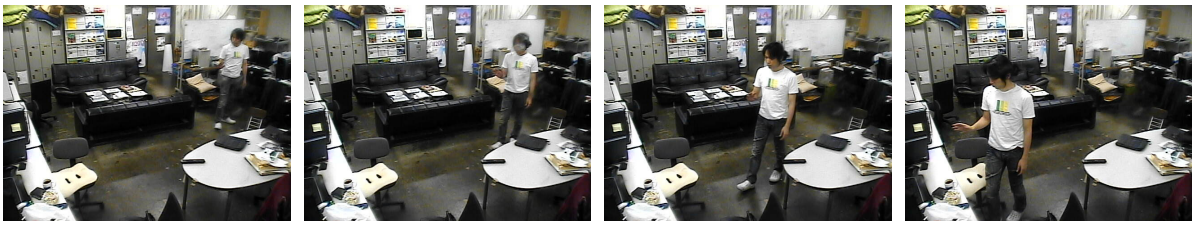
$$th = \bar{A}_{max} + a \times \sigma_{A_{max}} \tag{12}$$

A は該当周波数の振幅強度，$\bar{A}_{max}$ と $\sigma_{A_{max}}$ は A の平均と標準偏差，th は閾値を表す．a は検出する手振りの大きさを調整する係数である．ここで，$A_{max} > th$ かつ $A_{max} > A_{2Hz}$ を満たす画素を手振り箇所として検出する．

3.　実　　験

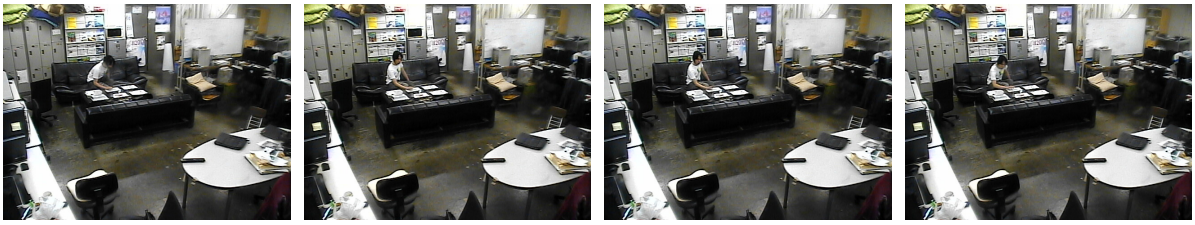
3.1　テストシーケンス

実験にあたり，以下の 4 種類のテストシーケンスを作



(a) シーケンス A:右から左に手を振りながら移動する

(b) シーケンス B:静止して手を振っている



(c) シーケンス C:静止して指振りを行っている

(d) シーケンス D:移動のみしている

図 8　実験に用いた映像シーケンス

成した．シーケンスは VGA サイズを 30fps で撮影したもので，それぞれ 150 フレーム用いている．

シーケンス A:　ユーザがカメラから 2〜3m の位置を，右から左に手を振りながら移動する．

シーケンス B:　ユーザがカメラから 4m の位置で静止して，手を振っている．

シーケンス C:　ユーザがカメラから 4m の位置で静止して，指振りを行っている．

シーケンス D:　ユーザがカメラから 3〜4m の位置を移動する．

それぞれのシーケンスを図 8 に示す．

3.2　オプティカルフロー算出手法比較

2.1 節で述べたオプティカルフロー算出手法について精度比較を行った．シーケンス D の 26 フレームと 27 フレーム (図 9) についてそれぞれの手法を適用し，人手で作成した正解データと比較した．この時用いたパラメータは表 1 の通りである．またすべての手法において，処理の高速化のために，ガウシアンピラミッドを用いて階層的なフロー算出を行った [10]．繰り返しによる最適化はそれぞれ 100 回である．処理には Intel®Core ™i7 975・メモリ 6GB，NVIDIA®GeForce®GTX 580 の PC を用い，フローの計算は GPU で行った．フローの算出結果を図 10，処理時間を表 2，誤差を表 3，4 に示す．いずれの手法も良好な結果であったが，Block Matching と Moving Gradients では人物の胴体のフローで誤差がやや大きくなっている．また，Block Matching と AD Census ではフリッカの影響で誤ったフローが発生している．AD Census と Classic+NL は，全体に細かいフローがノイズとして発生したため，平均終点誤差が大きくなっている．また，Classic+NL は処理時間が非常に大きかった．

これらの結果から，以降の実験では，平均終点誤差が小さく処理量が軽い Moving Gradients を用いて評価を行う．

3.3　検 出 処 理

Moving Gradients を用いたオプティカルフロー算出結果を用い，式 (12) の a を 15 として提案手法による検出処理を行った．従来手法として，筆者らの指振りの検出手法 [11] との比較も行なっている．検出結果を表 5 に

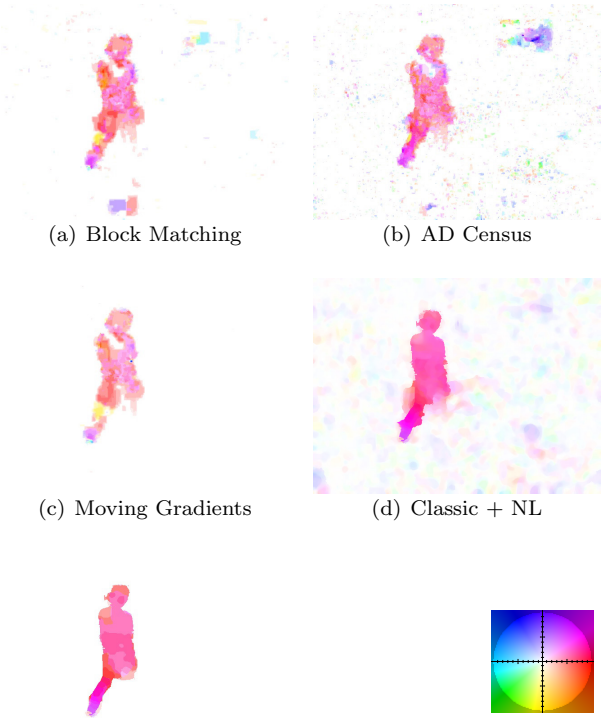
図 9　フロー生成テスト画像

表 1　実験で用いたパラメータ

λ_{BLOCK}	20
λ_{AD_census}	0.15
λ_{AD}	10
λ_{census}	30
λ_{MG}	200
σ^2	25

表 2　フロー算出に要した時間

手法	処理時間 (ms)
Block Matching	38.8
AD Census	57.2
Moving Gradients	28.6
Classic + NL	896.8



(a) Block Matching　(b) AD Census

(c) Moving Gradients　(d) Classic + NL

(e) 正解データ

図 10　フロー生成テスト画像

示す．シーケンス D はネガティブサンプルのため，この表には含まれていない．提案手法により移動しながらの手振りを精度良く検出できている．シーケンス A ではユーザが歩きながら手振りを行っているため，従来手法では全く検出できていないが，提案手法では精度良く

表 3　フローの終点誤差評価

手法	誤差 1pixel 以上の割合 (%)	平均誤差 (pixel)
Block Matching	4.54	0.25
AD Census	6.56	0.30
Moving Gradients	3.69	0.21
Classic + NL	2.85	0.38

表 4　フローの方向誤差評価

手法	誤差 2.5° 以上の割合 (%)	平均角度誤差 (°)
Block Matching	1.46	0.57
AD Census	3.13	3.14
Moving Gradients	1.26	0.49
Classic + NL	7.23	1.02

表 5　検出処理結果 (%)

	従来手法		提案手法	
	再現率	適合率	再現率	適合率
シーケンス A	0.0	0.0	80.0	78.7
シーケンス B	94.7	78.3	100.0	79.8
シーケンス C	84.2	100.0	100.0	66.3

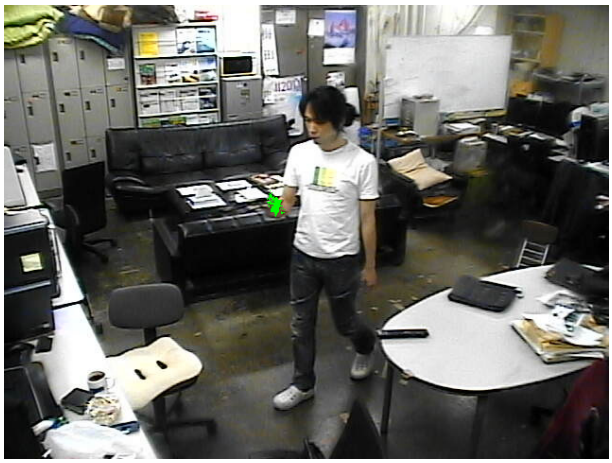
検出できている．シーケンス B は静止しながらの手振りであるが，従来手法より若干の改善が見られる．これは，ユーザがカメラに向けて手を振っているため，正確なフローの算出が可能であったためと考えられる．また，シーケンス C ではどちらの手法も良好な結果であるが，提案手法では誤検出が比較的多く発生している．シーケンス C では手が小さく写り，為される動きも微小であるため，式 (12) による閾値が非常に小さくなったためである．シーケンス D における誤検出の発生数は従来手法が 24 フレーム，提案手法が 44 フレームであった．従来手法と比較すると，やや誤検出が多くなっている．

図 11 に検出結果の例を示す．手振りが行われている領域を正しく検出できている．また，誤検出は足先などに多く発生していることが分かる．これは，足先などは発生するフローが大きいため，残差が大きくなりやすいことが原因と考えられる．

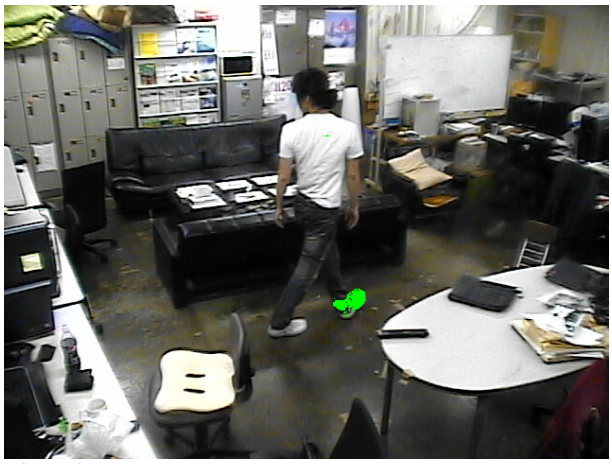
4. ま と め

本研究では，移動しながらでも手振り検出を行うことが可能な手法の提案を行った．これにより，手振りを行う際に静止していなければならない，という従来の制限を取り除くことが可能になった．人物の姿勢推定による，人物の周囲への機器操作割り当てなどの手法 [12] に，本手法を適用することにより，ユーザはいつでも・どんな姿勢でも機器操作を行うことができるようになる．

今後の課題として，手振り検出の閾値算出法の改良が挙げられる．本稿では画面全体のピーク強度を用いることにより，手振りが小さな時でも，ユーザが歩いている時でも高精度な手振り検出を実現した．しかし，画面内




(a) 正しく検出された例

(b) 誤検出が発生した例

図 11　手振り検出結果．緑の領域が横方向，青が縦方向，赤が縦横両方向の手振りが検出されたことを示す．

にユーザが複数存在し，静止状態と移動状態が混在するような場合，本手法での対応は難しい．このため，ユーザが複数であっても対応可能な閾値算出手法が必要である．

文　　献

[1] S. M. Dominguez, T. Keaton, and A. H. Sayed, A Robust Finger Tracking Method for Multimodal Wearable Computer Interfacing, IEEE Trans. on Multimedia, Vol. 8, No. 5, pp. 956–972, Oct. 2006.

[2] X. Liu, K. Fujimura, “Hand gesture recognition using depth data,” IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, Korea, pp. 529–534, May 2004.

[3] 入江耕太，梅田和昇，“濃淡値の時系列変化を利用した画像からの手振り検出，” 日本ロボット学会誌，vol.21, no.8，pp.923–931, 2003.

[4] 若村直弘，鈴木健一郎，入江耕太，梅田和昇，“インテリジェントルームの構築–直感的なジェスチャを用いた家電製品の操作–，” 画像の認識・理解シンポジウム (MIRU2005)，IS3–108，pp. 1074–1081，Jul. 2005.

[5] Y. Boykov, V. Kolmogorov, An Experimental Comparison of Min–Cut/Max–Flow Algorithms for Energy Minimization in Vision, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 26, No. 9, pp. 1124–1137, Sept. 2004.

[6] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang, “On Building an Accurate Stereo Matching System on Graphics Hardware,” 1st IEEE Workshop on GPU in Computer Vision Applications (In conjunction with ICCV 2011), Barcelona, Spain, Nov. 2011

[7] R. Zabih, J. Woodfill, “Non–parametric local transforms for computing visual correspondence,” European Conference on Computer Vision (ECCV), pp. 151–158, Stockholm, Sweden, May 1994.

[8] D. Mahajan, F. C. Huang, W. Matusik, R. Ramamoorthi, P. Belhumeur, Moving Gradients: A Path–Based Method for Plausible Image Interpolation, ACM Transactions on Graphics, Vol. 28, Issue 3, Aug. 2009.

[9] D. Sun, S. Roth, and M. J. Black, “Secrets of optical flow estimation and their principles,” Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2432–2439, San Francisco, USA, Jun. 2010.

[10] T. Lindeberg, Scale–space theory: A basic tool for analysing structures at different scales, Journal of Applied Statistics, vol. 21, no. 2, pp. 224–270, 1994.

[11] 浅野秀胤，織茂達也，高橋真人，寺林賢司，太田睦，梅田和昇，“フーリエ変換を用いた小さな手振りの検出，” ビジョン技術の実利用ワークショップ (ViEW2010)，pp. 264–267，Dec. 2010.

[12] 永易武，浅野秀胤，寺林賢司，梅田和昇，“操作者に固

定された相対座標における指振りを用いた簡便な家電操作システムの構築，” 画像センシングシンポジウム (SSII2012)，Jun. 2012.