

距離画像センサを用いた3次元指差し方向認識

武田 泰幸[†] 浅野秀胤[‡] 寺林 賢司[†] 梅田 和昇[†]

[†] 中央大学理工学部 〒112-8551 東京都文京区春日 1-13-27

[‡] 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

E-mail: [†] takeda@sensor.mech.chuo-u.ac.jp, {terabayashi, umeda}@mech.chuo-u.ac.jp

E-mail: [‡] hidetsugu_asano@post.pioneer.co.jp

あらまし 近年、操作者のジェスチャを認識して家電製品などを操作するシステムが様々な場面で期待されている。本論文では、ジェスチャ操作システムの一つであるインテリジェントルームにおいて、操作する家電製品の選択を、距離画像センサを用いた3次元ジェスチャ認識によって行うことを提案している。操作者が操作したい家電製品に対して指を差すことで、操作する家電製品の選択を行っている。距離画像センサで取得した画像から指領域を抽出し、指の3次元的角度を求めることで指差し方向を認識する。実験により、指差し方向の認識率を評価している。

キーワード 距離画像センサ, 指差し方向認識

1. 序論

近年、人のジェスチャを用いて家電製品を操作することを目的とした研究が多く行われている。我々は、部屋にカメラを設置し、操作者のジェスチャを認識して家電製品の操作を行うインテリジェントルームを構築している[1]。従来システムでは操作したい家電製品を指差しにより選択をしている。指差し方向の認識は、2台以上のカメラを用いて実現している。しかし、この手法ではカメラの方向に指差しを行った場合に必要な特徴量を得ることができず、正しく指差し方向を認識できないことがある。そこで本研究では、対象物体までの距離情報が取得可能な距離画像センサである Kinect を用いる。このセンサを用いることで、従来システムでは誤認識するシーンでも、正しく指差し方向の認識を行うことが可能なシステムを構築する。本システムに対して、実験を行いその評価を行う。

2. 指領域抽出手法

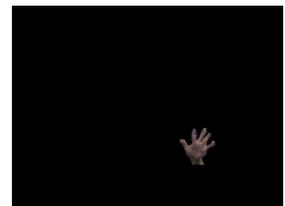
指差し方向を認識するために指領域の抽出を行う。まず、Kinect から取得した画像から手領域を抽出し、さらにその画像から指領域を抽出する。

2.1. 手領域抽出

我々のインテリジェントルームでは、操作者が手振りを行うことで、システムが操作者の特定を行っている。そこで、Kinect でも手振りによって操作者の特定を行う。Kinect では、手振りの検出や、検出した手の中心点を追跡する関数がライブラリとして公開されている[2]。これらを用いることで手領域の抽出を行う。



(a) 入力画像



(b) 手領域画像

図1 手領域抽出例

具体的には、検出した手の中心点からの奥行方向の距離が $\pm 15[\text{cm}]$ の範囲内の点を抽出する。抽出した手領域の例を図1に示す。距離計測する画像領域の大きさは、中心点のカメラからの距離に応じて動的に変化させる。

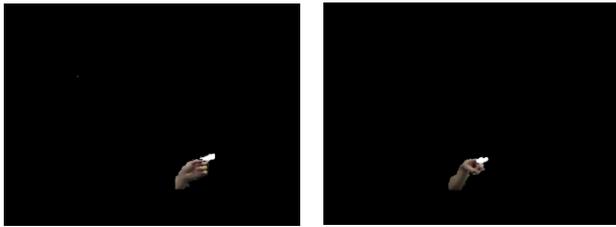
2.2. 指領域抽出

2.2.1. 指領域抽出(1)

抽出した手領域を2値化し、その輪郭線を取得する。取得した輪郭線から指の形状の部分のみを抽出する。抽出した指領域の例を図2(a)に示す。本図の白い領域が本手法で抽出した指領域である。

2.2.2. 指領域抽出(2)

前項で述べた方法だと、指がカメラの光軸方向に向いているときに輪郭線から指の形状を取得できない場合がある。そのようなときには、抽出した手領域の全画素のカメラからの距離を求め、その距離の最小値と最小値+5[cm]の範囲内にある領域を抽出する。抽出した領域が複数存在するときもあるので、抽出した領域を2値化・ラベリングし、領域の大きさが最大の領域を取得することで、指領域を抽出す



(a) 指が横方向を向いている場合 (b) 指がカメラの方を向いている場合

図 2 指領域抽出例

る。抽出した指領域の例を図 2(b)に示す。本図の白い領域が本手法で抽出した指領域である。

3. 指差し方向認識手法

抽出した指領域から特徴量を取得し、指差し方向の認識を行う。

3.1. 特徴量の取得

指差し方向を認識するために用いる特徴量として、指先と指の付け根部の点の 3 次元座標の値を用いる。指先の点と指の付け根部の点は、指領域の u 座標の最小値と、その最小値を取る画素の v 座標の平均値と、 u 座標の最大値と、その最大値を取る画素の v 座標の平均値をそれぞれ取得し、その 2 点のカメラからの距離を取得することで求める。2.1 節で求めた手振りの中心点と、求めた 2 点のユークリッド距離をそれぞれ計算し、距離が小さい方の点を指の付け根部の点、もう一方の点を指先の点とする。

3.2. 指の角度計算

指先の点と指の付け根部の点の 2 点間の角度を計算する。この 2 点の 3 次元座標 (u,v,d) の値は u,v は [pixel] で、 d は [mm] で取得している。この 2 点の 3 次元座標をカメラ座標空間に変換し、2 点間の角度を計算する。カメラ座標空間における 3 次元座標を (X,Y,Z) とするとその変換式は式(1)で求められる。

$$(X,Y,Z)^t = d \left(\frac{u-c_u}{f}, \frac{v-c_v}{f}, 1 \right)^t \cdot \cdot \cdot (1)$$

ここで C_u, C_v は画像中心座標の u,v 座標、 f はカメラの焦点距離である。これらのカメラパラメータの値を表 1 に示す[3]。式(1)より指先の点と指の付け根部の 3 次元座標 (X,Y,Z) を求めて、この 2 点間の角度を計算する。ただし、カメラに向かって指差しをしている場合は、指の付け根部が検出できないため、正しく角度計算ができない。この場合は、指領域の重心の 3 次元座標と 2.1 節での手振りの中心点の 3 次元座標を用いて角度計算を行う。指の向きがカメラの方向に向いているかの判定は、指領域の面積の大きさで判定している。

表 1 カメラパラメータ [pixel]

C_u	313.7
C_v	259.0
f	526.4

4. 認識実験

構築した指差し方向認識システムの有用性を検証するため実験を行った。Kinect の主な仕様は、画像サイズ:VGA、計測範囲:0.5~10.0[m]である。画像処理ソフトは OpenCV, PrimeSense OpenNI を用いた。実験はカメラから 1.0, 2.0, 3.0[m]離れた場所でカメラに正対して行い、指差しの角度はカメラの方向に対し、 u 軸方向に $0[^\circ], 30[^\circ], 45[^\circ], 60[^\circ], 90[^\circ]$ とし、各角度で 100 回ずつ実験を行った。得られた角度の平均を表 2 に、標準偏差を表 3 に示す。1[m], 2[m]では、ほぼ正しく指差し方向が認識されていることが分かった。しかし、3[m]だとどの角度でも標準偏差が大きくなった。これは、指領域が十分に取得できず、正面への指差しと誤認識してしまったためと、距離計測の誤差が大きくなったためだと思われる。

表 2 角度平均 $[^\circ]$

	0 $[^\circ]$	30 $[^\circ]$	45 $[^\circ]$	60 $[^\circ]$	90 $[^\circ]$
1.0[m]	6.3	38.6	47.1	62.3	89.4
2.0[m]	21.7	38.6	46.5	56.9	87.3
3.0[m]	28.9	68.3	57.2	77.2	83.8

表 3 標準偏差 $[^\circ]$

	0 $[^\circ]$	30 $[^\circ]$	45 $[^\circ]$	60 $[^\circ]$	90 $[^\circ]$
1.0[m]	4.1	9.2	4.6	1.9	1.0
2.0[m]	22.1	9.8	10.6	9.5	2.4
3.0[m]	20.1	42.7	28.0	33.0	25.1

5. 結論

本研究では、指差し方向認識システムを構築し、実験により認識率の評価を行い有用性を検証した。今後は、指差し角度が小さいときでも誤差が小さくなる手法の考案をしていく。

6. 参考文献

- [1] 入江耕太, 若村直弘, 梅田和昇: “ジェスチャ認識に基づくインテリジェントルームの構築”, 日本機械学会論文集 C 編, Vol.73, No.725, pp.258-265, 2007.
- [2] 中村薫: “KINECT センサープログラミング”, 秀和システム, 2011.
- [3] Tercel::Diary: Kinect プチハック (エピソード 3) <http://tercel-sakuragaoka.blogspot.com/2011/04/kinect3.html>