

資料

ジェスチャによるカーナビ操作システムの検討

篠村 祐司*・藤原 直樹*・平井 克尚*・大櫃 秀治*・泉 賢二*

1. 目的

現在多くの自動車には車載機器としてカーナビゲーション装置（以下、カーナビ）が搭載されており、経路案内や音楽機能などの利便性から一般的に普及している。一方、運転時のカーナビ操作やカーナビ画面の脇見が関連した事故¹⁾が発生しており、運転時のカーナビ操作の課題解決が求められている。

一般的なカーナビの操作方法として、タッチパネルやステアリングスイッチなどがある。タッチパネルの場合、操作時に直接画面を見て画面に触る必要があるため、脇見時間が発生してしまう。また、カーナビに表示されるコンテンツはカテゴリ毎に階層構造になっていることが多いため、操作の複雑さが長時間操作につながる。ステアリングスイッチの場合、簡易な操作であれば少ないスイッチで済むが、多機能にするほどスイッチ数が増えてしまい、その結果目視でのスイッチ確認操作が必要となる。その他の操作方法としては、カーナビをジェスチャで操作する製品としてパイオニア株式会社製のカーナビ²⁾が挙げられる。この製品では、カーナビ画面下部の赤外線センサ部分に5～15cm程度手を近づけ、手を左右どちらかに振ることでカーナビコンテンツを操作する。タッチパネルやスイッチ等と異なり非接触での直感的な操作が出来る一方、「手を近づける」、「右（または左）へ手を払う」のジェスチャのみ対応しており、ジェスチャに割り当てられるカーナビ機能数は限られる。

そこで本研究では、ジェスチャ入力による非接触操作技術によって多くのカーナビ機能を簡便に操作できることを目的としたカーナビ操作システムの検討を行ったので、その結果について報告する。

2. 方法

2.1 試作システム構成

本研究では、距離画像センサを用いたジェスチャ認識技術と、ステアリングに搭載したスイッチとを組み合わせたカーナビ操作システムを考案し試作を行った。試作システ



図1 試作システム

ムの構成を図1に示す。試作システムは、カーナビコンテンツを表示するカーナビモニター、距離画像センサ、ステアリングのスイッチで構成されており、それぞれPCに接続されている。システムの操作方法としては、距離画像センサの前に手をかざし、ステアリングスイッチを押しながら特定のジェスチャを行うことで、ジェスチャ操作結果に応じたカーナビコンテンツの表示切り替えの操作が可能となる。

システム内部処理としては、距離画像センサから取得した距離画像に対して画像処理およびジェスチャパターン判定を行う「手情報算出」、ステアリングスイッチがONの状態の場合ジェスチャに応じた操作を受け付ける「ジェスチャ操作判定」、ジェスチャ操作結果に応じた「カーナビコンテンツ操作」が行われる（図2）。PC上では画像処理及びスイッチ入力を受け付けるプログラムとカーナビコンテンツ制御プログラムが稼働している。詳細については後述する。なお、本研究における試作開発は県有特許第5750687号³⁾を基に行った。

2.2 距離画像

本研究では、センサで撮像した人の手の動きをカーナビの操作インターフェースとして利用する。カーナビの場合、利用時間帯（日中・夜間）や天候による外光変化が大きい

*ヒューマンインターフェイス技術開発プロジェクトチーム（現 情報・ヒューマンアメンティ科）

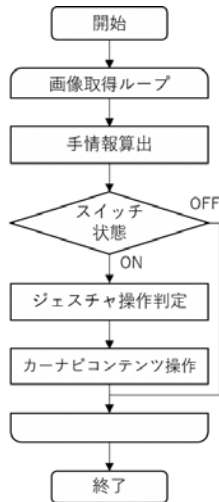


図2 システム内部処理手順



図3 RGB画像（左図）及び距離画像（右図）

ため、環境の影響を受けやすいRGB画像での画像処理は困難である。そこで、RGB画像ではなく他の画像センシング方法として距離画像を採用した。

距離画像とは撮像物体とセンサまでの距離が画素単位で保持される画像形式で、距離画像を生成するために、2点間のカメラの視差から距離を算出するステレオビジョン方式や、照射光と反射光の位相差を距離に変換し画像として取得するToF（Time of Flight）方式等様々な測距方式が提案されている⁴⁾。距離画像は一般的なwebカメラのRGB画像と違い、距離画像センサからの距離情報や物体の大きさが分かるため、実空間での形状認識等の画像処理に適している。ToF方式については、外光影響を受けにくいセンサも近年登場しており、今後主流になると考えられる。

今回車内に設置する点を考慮し、市販されている比較的安価で小型のToF方式の距離画像センサ（Creative Technology社製Creative Senz3D）を使用した。図3に本センサで撮影したRGB画像及び距離画像を示す。本センサではRGB画像及び距離画像の2種類の画像を取得可能だが、本システムでは距離画像を画像処理対象データとして利用し、RGB画像は使用しなかった。

2.3 手情報算出

カーナビ画面上のコンテンツをジェスチャで操作するため、距離画像に対する画像処理によって手の情報（指先位置、重心位置）の算出を行う。

まず、距離画像センサによって、横320ピクセル、縦240ピクセルの距離画像を取得する（図4（a））。次に、距離画

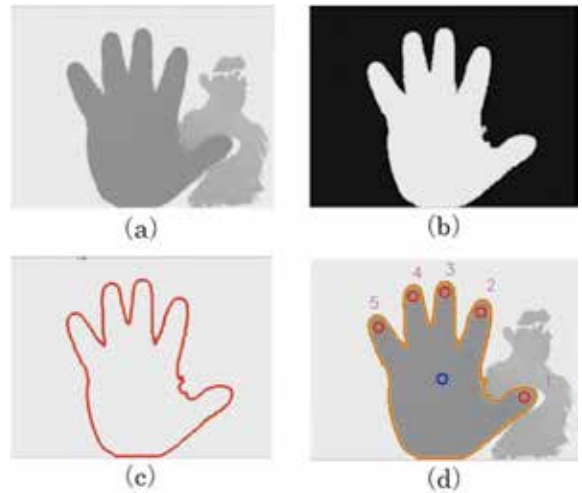


図4 手情報算出手順 [(a) 距離画像 (b) 一定距離部分取得 (c) 輪郭線抽出 (d) 指先及び重心位置算出]

像に対してセンサから一定の距離近接している画素のみ抽出する（図4（b））。画素毎に距離データ（mm単位）を保持しているため、閾値処理によって簡単に検出領域のみのデータを取り出すことが出来る。抽出した領域に対して輪郭線抽出を行う（図4（c））。そして、輪郭線に対する凸形状検出により指先位置を算出する（図4（d））。なお、距離画像センサの場合、物体の輪郭部分は距離データのノイズが多く、正しい距離データが取得できないため、検出した指先の内側付近を指先位置とし、指までの距離を算出する。併せて、手の領域の画素から重心位置（図4（d）の手中心円部分）を算出している。

指先の位置及び重心位置を算出することで、「指の本数」だけでなく、「掌の回転による指先位置の変化」、「指先又は重心位置の変化」を利用したジェスチャパターンによる操作が可能となる。また、距離画像の特徴である、センサからの距離を利用することが出来るため、押し込み等の3次元的な動きをジェスチャパターンとして利用することも可能となる。

2.4 スイッチによるジェスチャ操作判定

図1のように、距離画像センサをカーナビモニタ付近に設置した状態で画像処理を行うと、助手席に手を伸ばす動作や他の車載機器操作など、操作者の意図しない動きをジェスチャ入力操作と誤判定し、カーナビを誤動作させる可能性がある。そこで操作者のジェスチャ入力の誤判定を低減させるために、ステアリングにジェスチャ操作開始用のスイッチを設けた（図1）。操作者はジェスチャ操作を開始するタイミングでスイッチを押下する。そしてスイッチを押下し続けている間のみジェスチャ入力が操作判定に反映される。ジェスチャ認識開始のタイミングがスイッチと連動することで、操作者の意図しない動きによるジェスチャ操作の誤作動が防止できるとともに、ジェスチャ認識結果を判定する時間の短縮に繋がり、その結果、操作時間

表1 カーナビコンテンツ（ミュージック）とジェスチャ操作手順

機能	概要	画面	ジェスチャ操作手順
曲送り	アルバム内曲リストの曲送り操作		① 上方方向指差しをした状態でスイッチを押す ② ①をした後、スイッチを押したまま右方向に指を傾ける
曲戻し	アルバム内曲リストの曲戻し操作		① 上方方向指差しをした状態でスイッチを押す ② ①をした後、スイッチを押したまま左方向に指を傾ける
音量アップ	音量を上げる		① 右方向指差しをした状態でスイッチを押す ② ①をした後、スイッチを押したまま上方向に指を傾ける
音量ダウン	音量を下げる		① 右方向指差しをした状態でスイッチを押す ② ①をした後、スイッチを押したまま下方向に指を傾ける
停止	再生状態から曲を停止		① バー形状の状態でスイッチを押す
再生	停止状態から曲を再生		① バー形状の状態でスイッチを押す

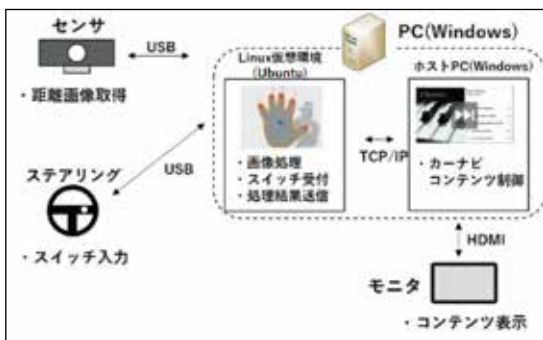


図5 検証環境（PC）

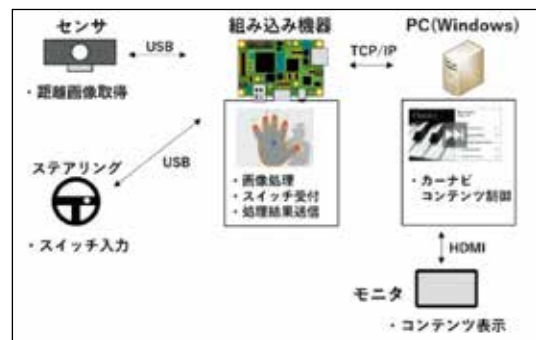


図6 検証環境（組み込み機器）

表2 検証環境の主な仕様

	PC 上の Linux 仮想環境 (Ubuntu)	R-Car Starter Kit Pro M3	Armadillo-840
CPU	Intel Core i7-3630QM (ホスト PC)	ARM Cortex-A57	ARM Cortex-A9
コア数	1 (仮想環境割り当て)	2	1
コアクロック	2.4GHz (ホスト PC)	1.7GHz	800MHz
RAM	4GB (仮想環境割り当て)	2GB	1GB

の短縮を実現することが可能となる。

2.5 カーナビコンテンツの構成と操作

試作システムにおけるカーナビコンテンツ構成について説明する。今回の試作システムでは、実際のカーナビではなく、PC上で動作する疑似的なカーナビコンテンツを作成した。コンテンツカテゴリとして、「ミュージック」、「動画」、「ラジオ」、「地図」、「電話応答」に分かれている。表1にミュージックでのジェスチャ操作手順を示す。各操作に対して簡易で直感的なジェスチャが割り当てられており、ステアリングのスイッチと連動してジェスチャ操作によるコンテンツ機能切り替えが可能となる。また、各カテゴリで異なるジェスチャを割り当てるとジェスチャパターンを覚えきれなくなってしまう可能性があるため、類似の機能操作については同じジェスチャを割り当てている。例えば、ミュージックの曲送りとラジオの局送りは同じジェスチャとしている。これにより、少ないジェスチャでの操作が可能となる。

また、スイッチを一定時間内に連続して押すことでカテゴリ切り替えを行う機能を実装した。これにより、例えばラジオと地図のカテゴリを容易に切り替えることが可能となる。

2.6 システム試作による検証環境の設定

本研究では試作当初、全てのプログラムをWindows上で動作する仕様としていたが、車載機器でのLinuxシェアが高まっている点、最終的な目標が車載向け組み込み機器である点、実際の環境では画像処理部分とカーナビ部分はハードウェアが別構成になる点から、「①Linux系OSでの動作」、「②組み込み機器での動作」の二点を重要視し、PC版、および2機種の組み込み機器版の計3種類の試作を行った。

PC版の構成は図5となる。センサ及びスイッチを、PCと接続し、PC上のLinux仮想環境 (Ubuntu) で画像処理及びスイッチの入力操作プログラムを管理する。同PCのホストOS (Windows) 上でカーナビコンテンツを動作させ、入力操作プログラムからの画像処理結果とスイッチ操作結果をTCP/IP通信にて受信し、モニターにコンテンツを表示する。

組み込み機器版については、前述のLinux仮想環境部分を組み込み機器に置き換え、Linux組み込み機器（画像処理及びスイッチ）とPC（カーナビコンテンツ）で機能を分離して構築している（図6）。組み込み機器は、R-Car Starter Kit Pro M3(ルネサスエレクトロニクス株式会社)、Armadillo-840（株式会社アットマークテクノ）の2機種を使用した。仮想環境及び組み込み機器の主な仕様は表2のとおりである。

3. 結果

今回検証実験として、①PC、②R-Car、③Armadilloにおける、カーナビコンテンツ側でのジェスチャ操作結果データ受信状況をジェスチャポーズ毎のFPS（frame per second）にて比較検証した。実験条件としては、距離画像センサからは、約60FPSでの画像取得を行っており、画像処理や通信にかかる負荷を含めた遅延が受信状況に反映される。測定条件としてstone（ゲー形状）、1～5本（指の本数）の6種類のポーズについて、各ポーズ1000フレームの平均FPSを算出した。また、手の位置としてセンサから20cm、30cm、40cmの位置にて測定した。図7に各距離における手検出例を示す。

図8、9、10に各検証環境におけるFPS測定結果を示す。測定結果から今回利用した指位置算出の画像処理手法が指の本数が増えること（凸部分が増えること）や輪郭線が長くなること（手の位置がセンサに近いこと）で処理時間がかかり、FPSが下がる傾向があることが分かった。R-CarについてはPCと同程度のFPSを出力しており、R-Carに比べてCPU処理性能が低いArmadilloの場合でも最低17.7FPS程度のフレーム速度を取得している。

スイッチを押下してから数フレーム同じジェスチャをすることでジェスチャ操作入力開始されたと判断する本試作システムでは、例えば17FPS（1フレーム当たり約58.8msec）の状況下で5フレームをジェスチャ操作判定の閾値にした場合、約294msecでジェスチャ操作の判断ができるため、組み込み機器の利用が十分可能であると考えられる。

本研究では、ジェスチャにおけるカーナビコンテンツ操作システムについて試作及び検証を行い、システム動作として十分な処理速度であることが確認された。今後の課題としては、ジェスチャパターン認識精度の検証が必要と考えられる。また、現在車載機器はCAN（Controller Area Network）での機器接続が主流であるため、実車へのテストにはCANによる検証が必要と考えられる。

文 献

- 1) 田久保宣晃, 木平真, 小島幸夫. カーナビゲーション装置の関連した事故の分析. 人間工学. 2001, 37, Supplement, p.466-467.
- 2) パイオニア株式会社. ユーザーズガイド (WEB版) AVCI-MRZ099W /MRZ099/MRZ077/MRZ066. <http://pioneer.jp/support/manual/navi/13raku/?section=8>.

- 3) 島根県. カーナビ用ジェスチャ入力装置. 特許5750687号. 2015-7-22.
- 4) デジタル画像処理編集委員会. デジタル画像処理[改訂新版]. 画像情報教育振興協会. 2015, p.405-406.

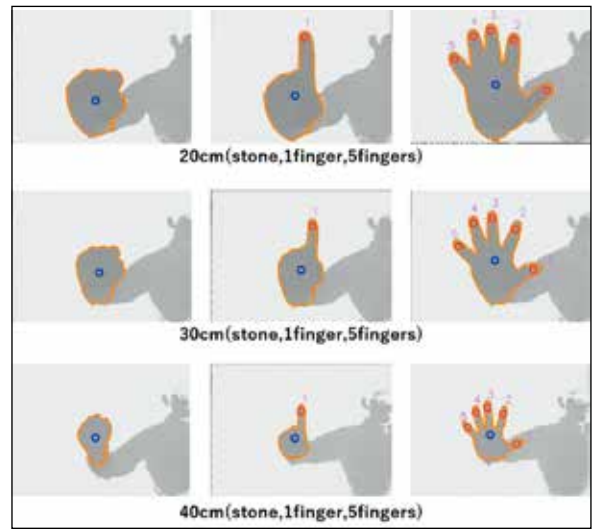


図7 各距離における手検出例

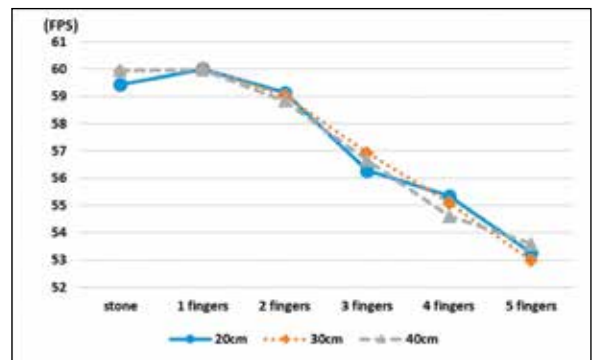


図8 FPS測定結果 (PC)

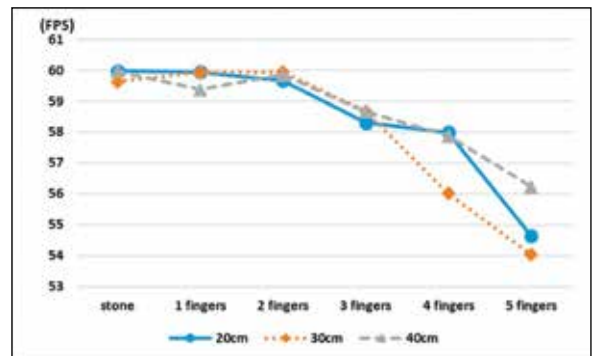


図9 FPS測定結果 (R-Car)

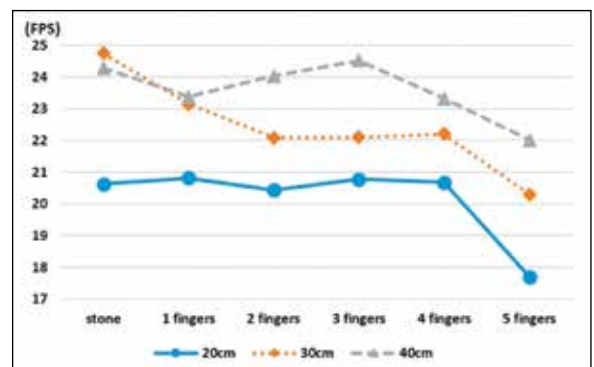


図10 FPS測定結果 (Armadillo)