

# Kinect を用いたプライバシー保護に 配慮した防犯システムの開発

畔柳 拓実

指導教員 大野 亙 准教授

## 1. はじめに

監視カメラ需要は過去四年間増加傾向にあり、2019 年には約 500 億円に達すると見込まれている。2020 年の東京五輪、西欧では相次ぐテロリズムにより店舗や商業施設、鉄道などでのより高度なシステムへの移行が考えられている。

このような防犯システムの問題点として、高度なシステムを持ち合わせながらも実際に利用されない点が挙げられる。これは、人工知能によって人の行先や感情、興味が判別されるようになるかもしれないという監視社会への抵抗があるためである。個人の保護とユビキタス社会の推進にとってはプライバシーの保護が重要な課題となる。

経済産業省が公表しているガイドライン Q&A によれば、「防犯カメラの撮影により得られる容姿の映像により、特定の個人を識別することが可能な場合には、原則として個人情報の利用目的を本人に通知又は公表しなければなりません。」とある。人物の顔や服装が隠され個人が特定できない状態では個人情報の保護に関する法律を違反しているとはいえず、責任問題を問われないと考えた。

本研究では Kinect を用いることで、人物の領域を認識し黒く塗りつぶして表示する。予め危険動作を設定しておくことで、不審者を検出した時のみ表示するシステムの開発を行った。

## 2. システム概要

本システムは 3 つの動作部で構成される。映像の元データであるストリームの取得を行うカメラ、データ処理を行うデバイス、そして危険動作学習ファイルを保存しておくローカルサーバーである。図 1 にシステムの概略を示す。

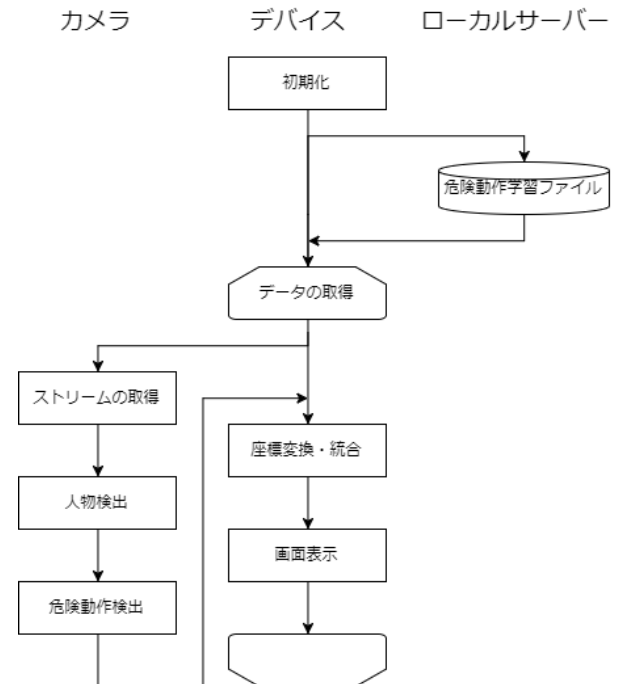


図 1 システム概略図

システム内部には C# を、UI デザインには Xaml を用いた。Kinect v2 は C++ でも開発が可能だが、参考文献の多さから C# を選択した。Xaml とは Microsoft が開発した XML ベースのマークアップ言語である。アプリケーションのビジュアルプレゼンテーションの背後にある言語だ。

Windows 用の Kinect 開発ツールである Kinect for Windows SDK v2、そしてジェスチャー認識のための機械学習ツールである Visual Gesture Builder (以降 VBG) を用いた。

## 3. 研究成果

### 3.1 座標変換

Kinect v2 から取得できるストリームのデータは Color, Depth, Infrared, BodyIndex と Body の 6 つあり、それぞれ座標系が異なる。座標系の特性の違いや各ストリームから得られる情報について表 1 Kinect 座標系特性に示す。

表 1 Kinect 座標系特性

座標系	概要	ストリーム
<b>Color</b>	1920*1080 2 次元座標 15fps 暗所	Color-RGBa-色画像
<b>Depth</b>	512*424 2 次元座標 30fps 固定	Depth-深度 Infrared-赤外線 BodyIndex-人物領域
<b>Camera</b>	メートル 3 次元座標	Body-骨格

Color 座標系、もしくは Depth 座標系に統一して出力する方法が一般的である。高解像度を出力しようとする时使用する PC では動作処理が遅かったため、すべての取得データを Depth 座標系に統合した。

### 3.2 人検出の判定と表示

取り出した Depth データを 8bit に変換し、2 次元座標上にそれぞれの値を代入した。さらに、BodyIndex データを取り出し人物領域を判定し Depth データに上書きした。そして人物の領域以外をカラー画像で出力することができた。

### 3.3 危険動作種類の判定と表示

VGB によって、映像サンプルを分析しジェスチャーデータベースファイルを作成した。大量のサンプルを用意し、データベースをアップデートすることで危険動作認識の精度を上げた。

学習させる動作は Discrete（離散的）と Continuous（連続的）の 2 種類に分けられ、それぞれ AdaBoost と Random Forest という別々のアルゴリズムで学習を行う。例えば、離散的動作である「監視カメラを見つめる」動作を観測した時、DiscreteGestureResults プロパティから値を読み込みその値が正である場合、「見つめる」危険行動を検出したことが分かる表示を行った。

### 3.4 観測中の人数を判定する機能の実装

BodyFrameSource.BodyCount プロパティを用いて Kinect v2 が観測している人数を取得した。人物が観測可能かどうかを判定する Body.IsTracked プロパティを前述の測定した人数分取得し、合計人数を出力した。

## 4. 動作

実際の動作画面を以下の図に示す。通常時には人物をシルエットとして表示し、人物を判別した時にシルエットで表示した（図 2）。あらかじめ学習させたデータファイルをもとに危険行動判定状態に入った。危険行動が検出されれば、シルエット表示が解除され、不審者の姿がカラー画像で判読できた（図 3）。さらに、その動作の種類を判別することができた。



図 2 人物検出時動作画面



図 3 危険動作検出時動作画面

画面上に人物が存在しても人が正面を向いていない場合には人物を検出したと認識されにくい問題がある。これは Kinect の至らない点である。Kinect のプレイヤー姿勢推定の技術は大規模な機械学習で成り立っているため、この問題を解決するためには、人物推定の確からしさの値を個人で操作する必要がある。

## 5. まとめ

本研究の目的の動作を行うプログラムの作成ができた。機械学習による不審者検出と、Kinect を用いた不特定人物保護によって利用者が安心して使える防犯システムの根幹は実装できた。

今後は、危険動作検出時の録画機能や、外出の際でも状況が分かるようなライブ配信機能など付加機能を付けていくことで本防犯システムが社会で使われやすくなっていくと思われる。

## 6. 参考文献

- [1] 中村薫, 杉浦司, 高田智浩, 上田智章: KINECT for Windows SDK プログラミング, 秀和システム, 2015