

# 「仮想窓」用高精度カメラシステムの開発

電気通信大学 情報工学科 成見研究室 0811123 Senarath C.M.

## 1 はじめに

「仮想窓」とは、ディスプレイがあたかも窓かのように見えるように画像を表示するシステムである。多視点で撮影した背景画像から観察者の視点にあわせ、適切な画像を抽出し、画面上に描画する。昨年の段階で、背景となる多視点画像を収集するため自動撮影機が開発されており、ディスプレイの領域と同様な範囲で視点を上下や左右にずらしながら背景を撮影することができる。また、観察者の視点位置を検出するためマーカーとウェブカメラを使用している。このシステムは観察者の視点を追従し、適切な背景を画面上に描画することに成功しているが、マーカー認識が悪く背景画像の読み込みが低速である。本研究では仮想窓システムの機能を向上させるために高精度カメラシステムを開発することが目的である。

## 2 研究方法

仮想窓システムは観察者の視点の位置を検出するカメラシステムと多視点の背景画像を撮影するカメラシステムの 2 つがある。それぞれのカメラシステムの以下に述べる欠点を改善する。

### 2.1 視点位置検出カメラシステム

昨年のシステムでは 3D メガネの上に付けられたマーカーをウェブカメラと ARToolKit を用いて認識させることによって位置検出を実現している。しかし、マーカーの認識効率が悪く、ユーザーはマーカーが認識されるようにメガネの方向や位置を常に配慮しなければならない。本研究では、視点位置の検出を行うために、人体の動きを認識できる Xbox 用コントローラ Kinect を導入し、マーカーを使う必要がなく、より高精度に視点位置検出することを目指す。

### 2.2 多視点画像撮影用カメラシステム

仮想窓の背景を撮影する自動撮影機では、デジタルカメラの自動撮影に用いているソフトウェアである RemoteCapDC とカメラの位置を制御している

LEGO Mindstorm が完全には同期していないため、撮影中に撮影サイクルがずれていくことがある。また、カメラの上下の位置を変えるための三脚のハンドルをモーターによって上下に移動させると、慣性のため過回転し撮影間隔が変わる。さらに、ディスプレイ一枚に相当する視点画像を撮影するために約 4 時間が必要である。これらの欠点を改良するために自動撮影機を改善するか新たなカメラシステムを構築する必要がある。

## 3 これまでの実験結果

視点位置を検出するカメラシステムについて、観察者の視点位置を検出する以下 2 つ方法を実験した。

### 3.1 人物の目の識別による視点位置検出方法

人物の目を識別し、目までの距離を取得することによって視点位置を計算した。目の識別に OpenCV ライブラリであらかじめ用意されてある、HaarCascade 識別器を使用した。人物の目の認識による視点位置検出の全体像を図 1 に示す。

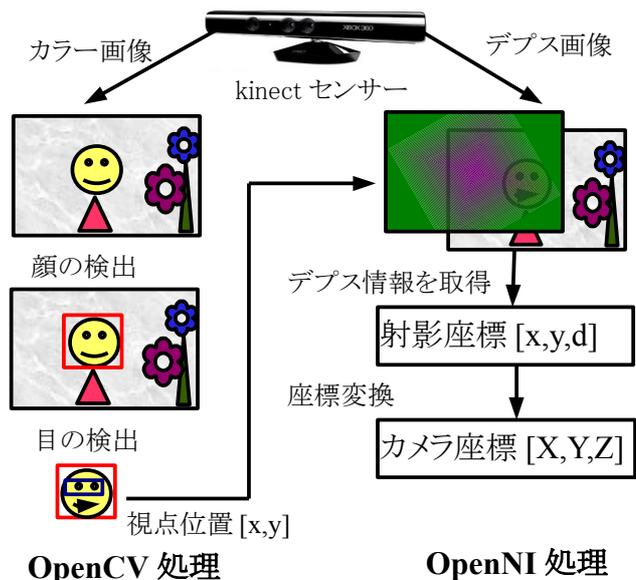


図 1 目の識別による視点位置検出

Kinect により生成されるカラー画像を識別器にかけ、目の位置 $[x,y]$ を元にデプス情報からカメラから目までの奥行き $d$ を求めた。射影座標 $[x,y,d]$ を逆射影変換を行うことによって、カメラの光学中心が原点となるカメラ座標系での位置 $[X,Y,Z]$ を求めた。上記の方法でひとつのフレームの処理を行うために1.5秒程度がかかった。また、3Dメガネを着用した場合は用意されている識別器で認識率がさらに低下し、仮想窓システムに応用することが難しい。

### 3.2 骨格追従による視点位置検出方法

OpenNI ライブラリと NITE ミドルウェアを使うことでカラーカメラ、深度カメラなどのデバイスの情報を元にユーザーやジェスチャの検出が可能になっている。Kinect で同時に6人までユーザーを検出することができ、そのうち2人のアクティブなユーザーの骨格約20関節を検出することができる。頭部の中心の関節の3次元位置を取得し、適切な補正を行うと視点位置を計算することが可能である。Kinect で骨格追従可能な関節を図2に示す。

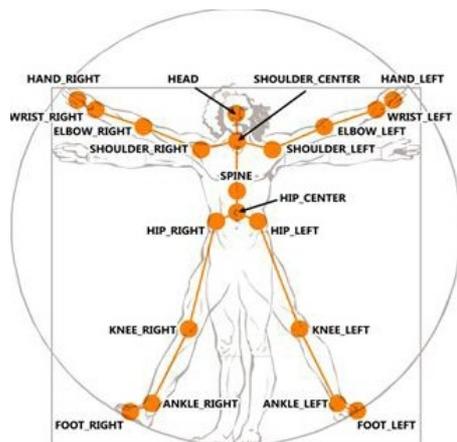


図2 骨格追従可能な関節

骨格追従方法で約0.05秒の速さで視点を取得することができた。姿勢や顔の向きに関わらず骨格追従ができるため、視点位置の検出に特に配慮する必要がなかった。ただし、OpenNI でキャリブレーションを開始させるため最初にガッツポーズをしなければならない。一旦キャリブレーションに成功したら、ユーザー Id を元にスケルトン情報を得ることが可能である。計算速度が速く、マーカなど着用する必要がないので仮想窓システムに応用可能であると考えられる。図3は骨格追従によって視点位置を検出している様子を示している。視点位置は実行画面の右上に黄色い文字で表示している。

## 4. まとめと今後の課題

「仮想窓」システムのカメラ部分を改良するためにまず視点位置検出の精度と速度の向上を目指した。Kinect を使った骨格追従によって0.05秒程度の速さで視点位置を求めることができた。今後の課題として以下の2点がある。

### 4.1 Kinect for Windows SDK の導入

本研究を開始した時点で、簡単に整えられる Kinect の開発環境として、OpenNI ライブラリと NITE ミドルウェアしか存在しなかったが最近では Microsoft 社が SDK を無償で提供している。Kinect for Windows SDK は Kinect と提供者が同じであり、以下のような利点があるため、将来的に導入したいと考えている。

- 標準の Windows 機能を経由して Kinect センサーの情報が取得可能
- OpenNI では、骨格情報得る際にガッツポーズが必要であったが、SDK では不要
- センサーのチルト角度が制御可能

### 4.2 背景画像撮影用のカメラシステムの改善

現時点では、背景画像撮影用のカメラシステムの開発に着手していない。大型ディスプレイ上に仮想窓を実現させるために、高速かつ高精度で多視点画像を撮影できる装置を開発しなければならない。

## 5 参考文献

1. KINECT センサープログラミング 中村薫
2. <http://www.xbox.com/ja-JP/kinect>
3. <http://www.openni.org/>
4. Kinect for Windows SDK beta Programming Guide



図3 骨格追従による視点位置検出