



Title	ステレオカメラを利用した、動くボール位置計測
Author(s)	Yi, Jun Hyung
Citation	平成30年度学部学生による自主研究奨励事業研究成果報告書. 2019
Version Type	VoR
URL	https://hdl.handle.net/11094/71966
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

平成30年度学部学生による自主研究奨励事業研究成果報告書

ふりがな 氏名	イ ジュンヒョン YiJun Hyung	学部 学科	基礎工学部システム科学科	学年	3 年
ふりがな 共 同 研究者氏名		学部 学科		学年	年
					年
					年
アドバイザー教員 氏名	浦西 友樹	所属	大阪大学サイバーメディアセンター情報メディア教育研究部門		
研究課題名	ステレオカメラを利用して動く物体の3次元位置を計測				
研究成果の概要	ステレオカメラを利用して、画像から動く物体の3次元での位置を計測する。				

1. 自主研究の目的

ステレオカメラを利用して得られたデータより動く物体の3次元での位置を計測し人間の目のようなコンピューターの目を作る。

2. 装置及び実験環境

2.1 装置リスト

Table 1 装置リスト

装置及び道具	数量
Brio ウェップカメラ	2 台
カメラ台	2 台
KinectV2 デップスセンサー	1 個
ノートパソコン	1 台
ボール	2 個

2.2 システム構成

2.2.1 計測システム

Table 2 計測システムの装置

装置及び道具	数量
Brio ウェップカメラ	2 台
カメラ台	2 台
ノートパソコン	1 台
ボール	2 個

計測システムは二つのカメラを利用し 3 m ~ 7 m 距離内にある特定の物体を認識しその距離を算出するシステムである。

2.2.2 評価システム

装置及び道具	数量
KinectV2 デップスセンサー	1 個
ノートパソコン	1 台
ボール	2 個

3. 理論

動く物体の 3 次元計測のために、三つの段階的な過程が必要である。まず画像の中で目的物を探すために物体認識過程、次に探した目的物を計測器からの位置を計測するための位置計測過程、最後に計測した結果が正しい結果を出しているか確認するための正確度評価過程に構成した。

3.1 物体認識過程

物体認識を行うアルゴリズムは多いが今回の自主研究では動きがあるボールを追跡するためもののが動きを判断する Optical Flow を使う。

3.1.1 オプティカルフロー

画像のピクセルに対して時間に従う変化量(速度)を求めることが Optical Flow といふ。

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \quad (1)$$

Optical Flow は(1)の式のように表され、テーラー展開により変形すると(2)の式が得られる。得られた(2)の式を Optical Flow Equation と呼ぶ。

$$\frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t = 0 \quad (2)$$

Optical Flow には色々なアルゴリズムがあるがこの実験では Lucas-Kanade 方法を扱う。

3.1.1.1 Lucas-Kanade 方法

Lucas-Kanade 方法は追跡する点を指定しその点の近い領域から探索するアルゴリズムである。

Lucas-Kanade 方法には三つの基本仮定がある。

- A. Brightness constancy : ある画像のピクセルはフレームが変わっても値が変わらない
- B. Temporal persistence or small movements : 動く物体の速度は早くない
- C. Spatial coherence : 空間上近接しているピクセルは同じ物体である可能性が高く、動きの方向性が一致する。

Lucas-Kanade 方法は演算量が少ない長所があるが物体の速度が早くなつた場合認識できなくなる短所がある。このために改良させたピラミッド Lucas-Kanade 方法は探索領域のスケールを変化していく大きい動きにも対応できる。

3.1.1.2 Iterative Lucas-Kanade 方法

Lucas-Kanade 方法が持っている短所である大きな動きを探索できない問題を解決するためイメージピラミッドを投入する。

もとの 画像を縮小しながら探索する。スケールが変化することで大きな動きの検出しやすくなる。

3.2 位置計測過程

3.2.1 カリブレーションの必要性

3次元の実世界をカメラで撮るとき、3次元点はカメラの位置、方向、レンズとイメージセンサーの位置関係、レンズの曲率により影響され実際の位置関係とは違うデータが得られる。このような誤差を無くすためにステレオカリブレーションよりパラメータを計測し誤差を無くすことが必要である。

3.2.2 ステレオカリブレーションの結果物

3.2.2.1 Essential Matrix

実空間上の点 X が画像 A では X_1 画像 B では X_2 に投影されたと考えると投影された二つの点には次の関係が成立する。

$$X_2^T E X_1 = 0 \quad (3)$$

この時 3×3 行列を Essential Matrix と言う。しかし Essential Matrix はカメラの内部パラメータによる誤差を含んでいないため、Fundamental Matrix を求める。

3.2.2.2 Fundamental Matrix

カメラ内部パラメータ行列を K とする。この時 Fundamental Matrix と Essential Matrix 間に次の関係式が成立する。

$$E = K_1^T F K_2 \quad (4)$$

OpenCV のステレオカリブレーション関数では Fundamental Matrix を求めることができる。

3.2.3 ステレオビジョンの目的

二つのカメラで得られた画像の中で同じ点を探し、その差を計算することより二つのカメラがある軸から対象点までの最短距離を求めることができる。

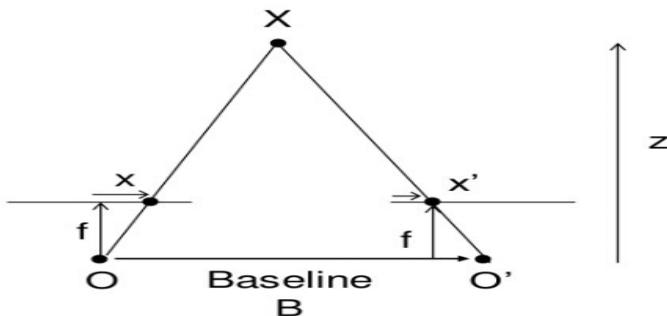


Figure 1 Disparity to Depth

幾何的な関係より Disparity と Depth の間に次の式が成立する。

$$\text{Disparity} = x - x' = \frac{B f}{z} \quad (5)$$

しかし二つの画像で同じ点を探すこと(マッチング)はすごく大きい演算を必要にする。演算量が大きくなるとライブで動く物体の距離を出すのは難しいので計算量を減らす必要がある。

3.2.3.1 エピポラ制限条件 Epipolar constraint

計算量を減らすため数学的な制限条件を使う。

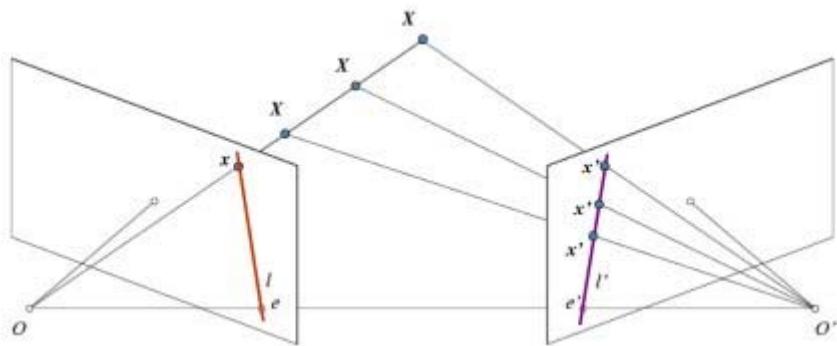


Figure 2 Epipolar Constraint

ある点 X に対して O を原点にするカメラ A と O' を原点にするカメラ B を考える。

カメラ A で観測した点 X を考える時、スクリーン上にある x と実空間の X の間の線 xX はカメラ A で観測した点 X に見えることができる集合である。この場合 X に対応するカメラ B のスクリーン上の x' が作る直線ができる。

画像全体に対して同じ点を探す必要はなくこの直線に同じ点を探すことで演算量を減らすことができる。

3.3 正確度評価過程

3.3.1 TOF デップスセンサーを利用した距離測定

TOF は Time of Flight の略字で、光が反射され戻ってくる時間を測定することで距離を計算する。

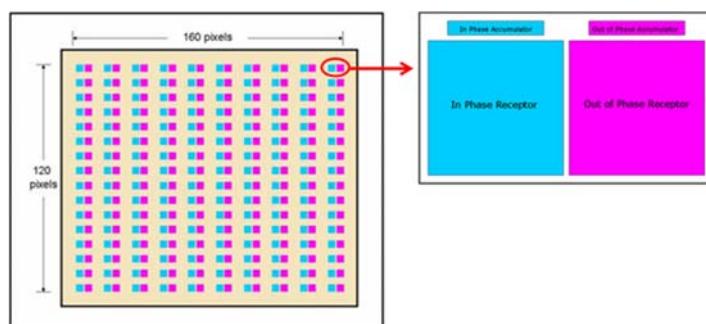


Figure 3 TOF Camera

TOF カメラのセンサーは二つの Recepter に構成されている。センサーの周辺部には LED を設置し周期的に点滅する。LED が光ってる時を In Phase、光っていない時を Out Phase と呼ぶ。

In Phase Recepter は LED が光っているとき、In Phase の時だけ活性化し Out Phase Recepter はその反対である。

活性化される時間差により累積される光の量を比較し距離を判断する。しかし強い太陽光の環境では使えない短所も持っている。

4. 結果

時間不足により各理論のサンプルコードを実行しその結果を確認すること以上は進めなかった。

5. 参考文献

詳解 OpenCV	Adrian kaehler・Gary Bradski / オライリー出版
OpenCV 公式サイト	https://opencv.org/
Time-of-Flight Camera An	Larry Li / Texas Instruments

Introduction	
--------------	--