

# 全方位カメラ視差画像による 3D 空間障害物検知システムの開発

Development of a 3D-space Obstacle Detection System using Omnidirectional Parallax Images

福谷武司

Takeshi Fukutani

機械素材研究所 機械・計測制御担当

国や県の政策で生産性革命が推し進められる中、ロボットのものづくり現場での活用が徐々に広がりを見せている。一方、グラフィックカードに搭載されている GPU を用いた高速並列処理演算を活用することにより、2次元カメラから3D空間の立体像を再現できるフォトグラメトリ技術の製造現場活用が注目されつつある。本報告では、これを実現する3D空間障害物検知システム手法のプログラム開発方法と、その動作について述べる。

While the productivity revolution has been promoted by national and prefectural policies, utilization of robots at manufacturing sites has been spreading gradually. Meanwhile, attention is being paid to the application of photogrammetry technology at manufacturing sites. The technology can reproduce stereoscopic images from a 2D camera by utilizing a high-speed parallel processing operation using a graphics processing unit (GPU) mounted on a graphics card. In this paper, a program development method for a 3D-space obstacle detection system technique and its operation are reported.

## 1. はじめに

従来、自動搬送車等の障害物検知システムは、ミリ波、赤外線レーザー、画像等を組み合わせた高コストで大がかりな手法で障害物までの距離をリアルタイムに測定する手法が主流となっている。一方、一般的なデジタルカメラで大量の写真撮影を行い画像のみでその視差を生かして3D空間データを生成するフォトグラメトリと呼ばれる手法も手軽さとコストの安さから広がりを見せている。その理由として、近年 High Performance Computing (HPC) の1手法として、グラフィックカードに搭載された画像表示のためのプロセッサ (GPU) による、高速並列処理演算プログラム手法が普及してきたことがある。これを生かした高速処理、特に行列演算を多用する3D空間処理に優位性があることでこれを生かした手法が開発されてきている。

また、近年、ドライブレコーダの普及にも見られるように、小型の機器で長時間の動画を撮影する、いわゆるアクションカメラと呼ばれる分野のカメラ

が普及してきており、その一種で全方位カメラが登場し、これを生かした広範囲撮影が可能になった。

本報告では、フォトグラメトリ手法を全方位カメラによってより早く障害物の検出を行うツールとして活用する検討をした結果を示す。

## 2. 開発ハードウェアとプログラム開発環境

### 2.1 開発用ハードウェア

3D空間の障害物データ生成を実現するためのハードウェアとして、表1に示す開発環境を構築した。

表1 開発環境構成

PC	CPU メモリ OS	Intel Xeon E5-2690(2.9GHz) DDR3-1333 SDRAM 64GB Windows 7 64bit
グラフィックカード	名称 コア数 クロック メモリ	NVIDIA GeForce GTX 1070 1920 1506MHz(ブースト) 384bit GDDR5 SDRAM 6GB
GPU 開発環境	コンパイラ CUDA	C++コンパイラ(Visual Studio 2015 for Professional 同梱) CUDA 8.0
OS		Windows 7 Professional
全方位カメラ		Kodak PixPro SP360 4k

この中で、使用した全方位カメラの仕様を表2に撮像範囲を図1に示す。

表2 全方位カメラの仕様

撮像素子	タイプ	1/2.3 BSI CMOS
	有効画素数	約1240万画素
レンズ	総画素数	約1276万画素
	焦点距離	0.85mm
	35mm換算	8.2mm
	F値	F2.8
	レンズ構成	7群9枚
撮像範囲	光学ズーム	無し(固定焦点)
	撮影距離範囲	50cm~∞
撮像範囲	光軸周り	360°
	面角	235°

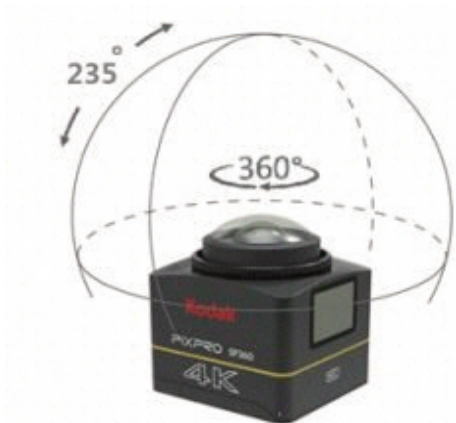


図1 使用した全方位カメラと撮像範囲

## 2.2 プログラム開発環境

全方位カメラを扱うためのプログラム開発環境を構築した。プログラム開発に当たっては、オープンソースであるOpenMVGを利用した。複数の元画像について、同じ位置として認識する特徴点を選定し、さらに対応点を推定することができる。原理としては図2の通りであり、一致する一点をそれぞれの画像上で見つけ出すことが必要となる。これを行うためにプログラムでは図3の手順を行った。

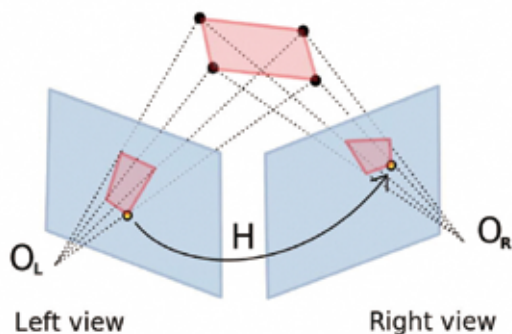


図2 画像上の点を3次元の空間上の点に返還する原理図  
(出典: OpenMVG web サイト  
<http://imagine.enpc.fr/~moulonp/openMVG/>)

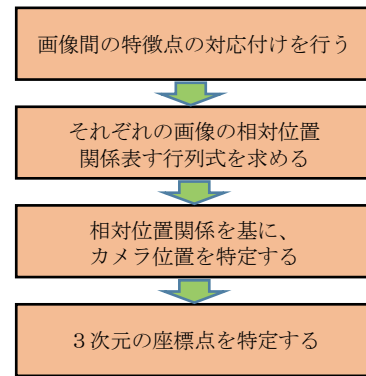


図3 3次元点を特定するまでの手順

図4は全方位カメラの2画像の場合3次元点測定までの過程である。左列の2つの画像から、中列のように特徴点選定を行い(黄色の点)、続いて右列の対応点推定を行ったものである。これにより図5のような点群データが得られる。室内写真に点群データを重ね書きすると図6のようになり、帯鋸、部品棚、天井配管などが形状として表れている。

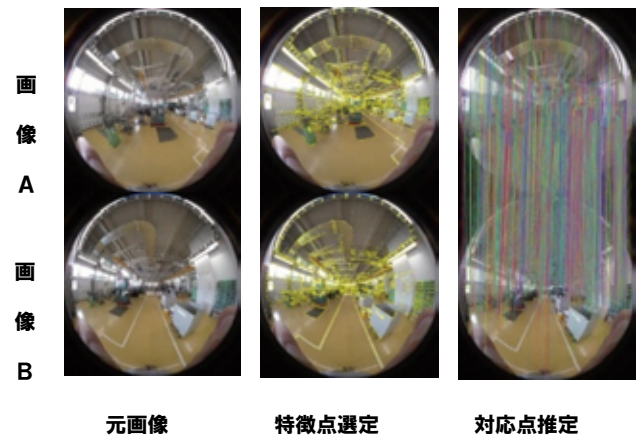


図4 特徴点選定および対応点推定

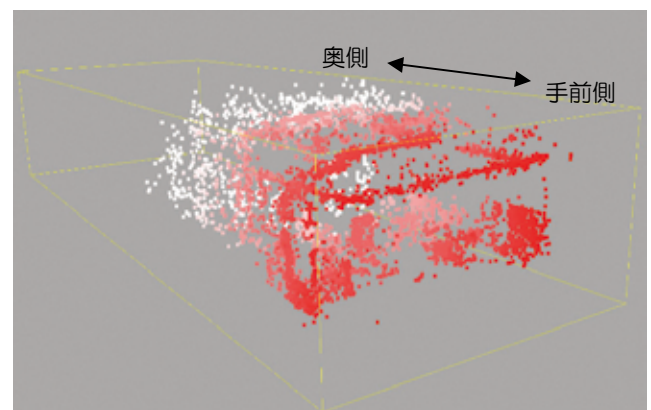


図5 取得した点群データ

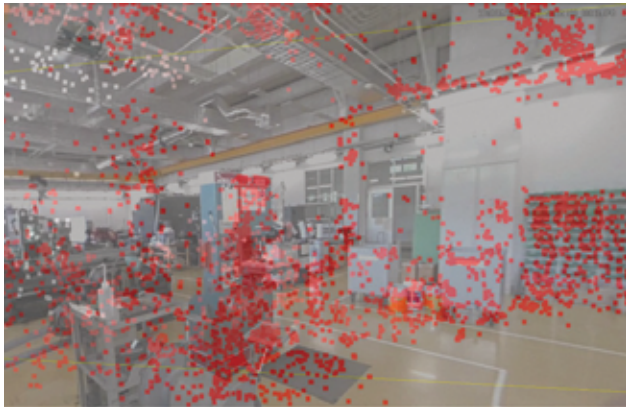


図6 室内写真に点群データを重ね書きした様子

### 3. 3D空間障害物のキャプチャ及び評価

#### 3.1 空間線を描画するための基本プログラム

点群生成状態を評価するにあたり、その指標として、

- ・生成した点の数
- ・点群データの適正配置

の2つについて調べた。

生成した点の数は、点群処理ソフトによりカウントした。図7は点群生成に用いた2画像の類似度と生成点数の関係をグラフ化したものである。

2つの画像の類似度にはほぼ比例して、生成点数が増加していることがわかる。中央の通路の障害物の（あり／なし）を見る。点の生成数は、2つの画像同士の類似の度合い（類似度）とほぼ比例する。

点群データの適正配置については次の方法で評価を試みた。今回の実験場所である機械工作室は中央に通路があり、その幅の実測値は 2.0m である。この通路にノイズによる点の侵入の（あり／なし）によって、点群が良好に生成された指標となる。

実験の結果、類似度 50% の場合ノイズデータが入り、通路幅寸法は 0.7m にとどまった。類似度 80% の場合は 2.0m と実測値と等しくなり、適正な開口部寸法を確保できた。同様にテストした結果、2つの画像の類似度が概ね 80%~95% のときに良好な点群が得られた。95%以上となると、視差を生じさせることができないため、点の数は増えるものの形状を作ることができない。

これらの実験により、全方位カメラ画像に適した

特徴点探索手法と、障害物検知に利用できる 3D 空間データ生成が可能であることが確認された。

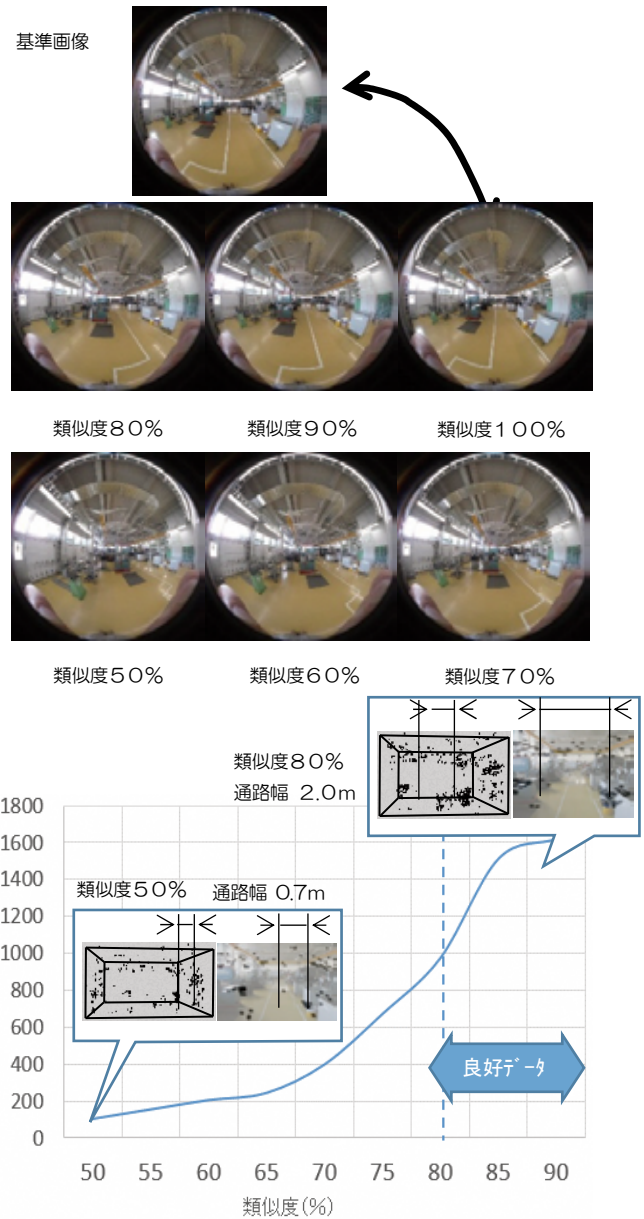


図7 類似度と生成点数の関係

#### 3.2 プログラムの応用事例

3.1 の手法を使えば障害物検知用の高速化は実現できたとも言えるが、生産ラインの3次元データ化などの高密度データ取得の高速化にもニーズがあるため、それにも取り組んだ。

高速特徴点探索アルゴリズム (RANSAC) を利用した 3D 空間データ生成システムを構築し、これにより図8のような3次元データの立体像を作成した。

また、動画像によるデータ生成は、画像取得時間

の短縮に有効な他、軌跡データとしての活用が可能  
なことがわかった。これを用いて図9のような3次  
元軌跡データを生成した。

参考までに、これらの図にはカメラ位置に撮影し  
た元の2次元画像を重ね合わせて示した。

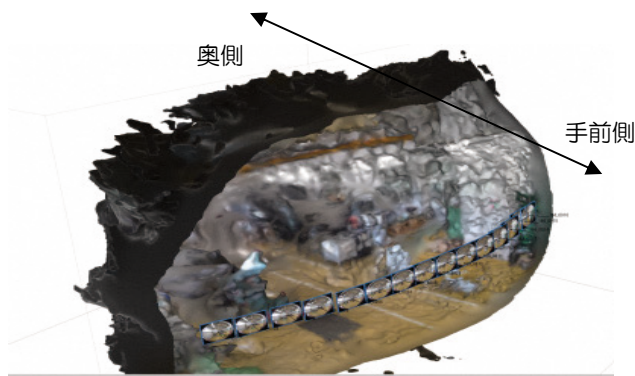


図8 空間データ生成システムによる3次元データ

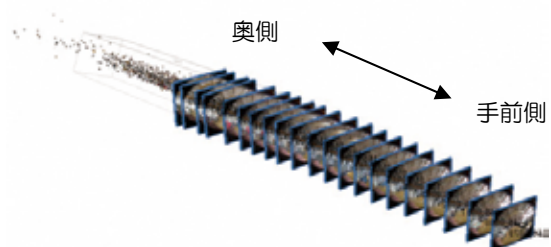


図9 動画画像から進行方向に生成した軌跡データ

図6のデータに対して、キャリブレーションを行  
い、補正したデータを部屋の上方から見た形状を図  
10に示す。初期状態のデータは左右の壁が奥行き方  
向に行くにつれて狭まっていた。

これをカメラパラメータ (f 値、歪値等) の変換  
により平行なデータを生成し、これにより部屋の実  
際の寸法に近いデータを生成した。

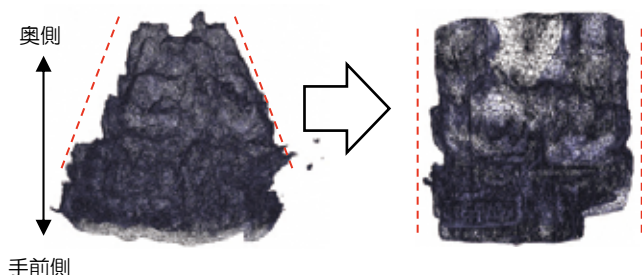


図10 部屋データのキャリブレーションによる補正  
(左:補正前、右:補正後)  
(データは部屋のデータを上面から見たもの)

## 4. 考察

このように、3D障害物データを生成できるよう  
になったが、現場に投入するにはまだいくつか課題  
がある。それは次のようなものである。

### (1) 演算処理速度

現状では演算処理速度がまだ足りておらず、  
障害物検知のためのプロセスには約 15~20 秒  
かかっている。今後ハードウェアの発展ととも  
に徐々に解決されていくと考えられる。

### (2) 撮影対象物による差がある

今回のような加工機や什器が多数配置され  
た現場は比較的特徴点が検出しやすくなって  
いた。しかし、同じ機械を一定間隔で規則的に  
並べるような現場等、特徴点が検出しにくい現  
場も存在する。

これまで培った3Dセンサ技術との併用も  
含め、何らかの補助的手段を用いることで解決  
は可能であると考えられる。

## 5. おわりに

演算量の多い3次元空間処理に関して、全方位カ  
メラによる撮影と高速並列処理演算を活用し、3D  
障害物を生成できることがわかった。ロボット分野  
は、AI、IoTと相まって大きな発展を見せてい  
る分野であり、より利便性の高い製品もリリースさ  
れ続けていることから、その動きを注視し、県内企  
業に役立つ技術として展開させ普及を図りたいと考  
えている。

## 参考文献

- 1) CUDA 高速 GPU プログラミング入門、小山田耕  
二監修、岡田賢治著、秀和システム
- 2) OpenMVG  
<http://github.com/openMVG/openMVG>