

# 階層グラフカット でステレオ

画像応用数学特論レポート

平成 27 年 1 月 13 日

医用画像工学研究室

中村優記

開発環境:

OS:Windows 7 Professional

CPU:Intel®Core™ i5-2540M CPU @ 2.60GHz 2.60Ghz

実装メモリ(RAM):4.00GB

開発環境 :

:Microsoft Visual Studio 2013

言語:c++

Opencv2.4.

実験内容 :

2枚の対象画像に対して階層グラフカットを用いたステレオマッチングを行い、視差画像を求める。

アルゴリズム :

1. 対象となる左画像、右画像を読み込む
2. ブロックマッチングを行い、各画素の視差の初期値を決定する
3. 階層ラベルを作成する
4. ラベルの更新が無くなるまで以下を繰り返す
  - (ア) 画素間にノード・エッジの追加された2次元グラフを作成
  - (イ) 各ノード・エッジにコストを設定する
  - (ウ) グラフカットを行い、新しいラベルの総コストを計算する
  - (エ) 新しいラベルの総コストが現在より小さければ、更新する
5. 視差画像を出力する

対象画像:

左画像(元データを 450×399 にリサイズして使用)



右画像:

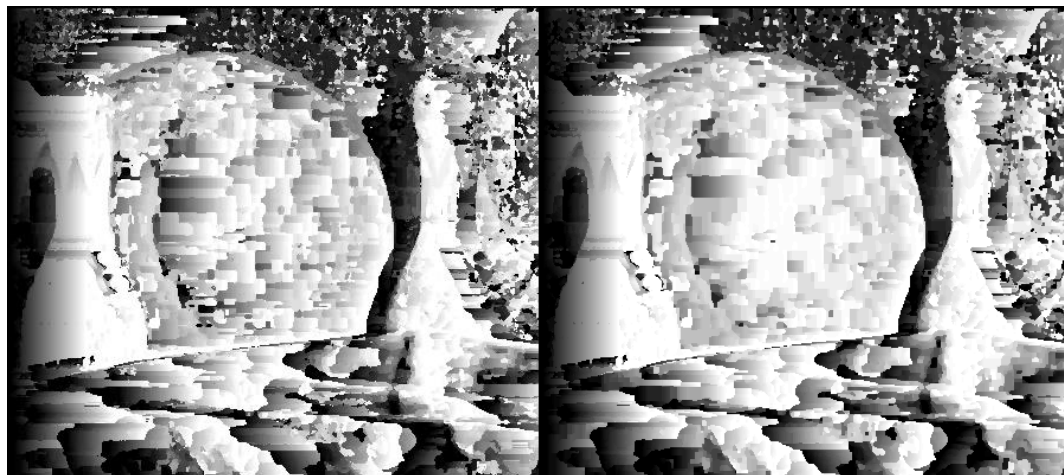


結果:

階層グラフカット、 $\alpha$ 拡張グラフカットを行った結果をそれぞれ示す。

(ウィンドウサイズ:3 で固定、スムーズコスト:1,10,100,1000)

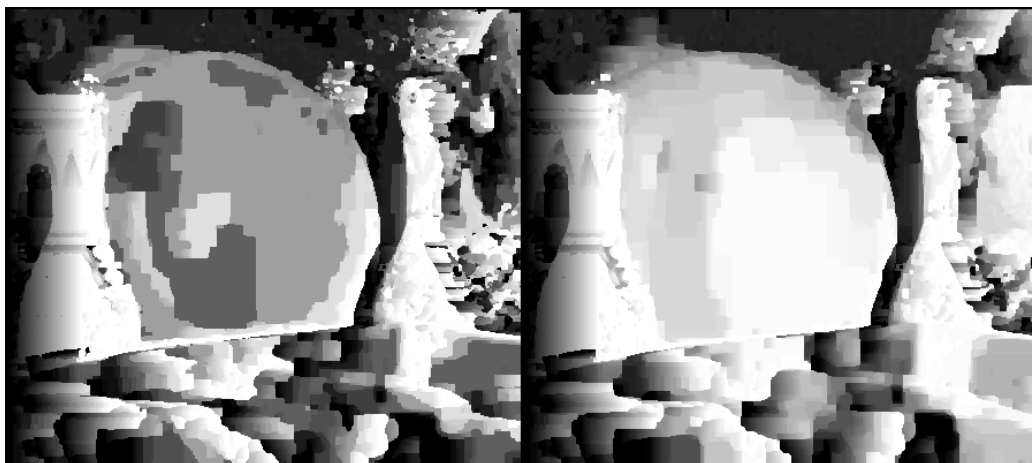
スムーズコスト:1(左が階層グラフカット、右が $\alpha$ 拡張)



スムーズコスト:10



スムーズコスト:100



スムーズコスト:1000



計算時間比較：

スムーズコスト	階層グラフカット	$\alpha$ 拡張グラフカット
1	44.413sec	144.732sec
10	45.893se	203.051sec
100	64.875sec	825.244sec
1000	64.989sec	1384.322sec

考察:

計算時間比較を見てみると、 $\alpha$  拡張グラフカットの計算時間に比べて階層グラフカットでは計算時間を大幅に短縮できていることがわかる。今回は最大視差の大きな画像を使用したため特に計算時間に差が出たのだと思われる。スムーズコストを変化させて結果に関して、スムーズコスト 1,10 はほとんど結果に差はなく、100,1000 に関しては結果に大きく差が生じた、どちらの結果もある程度の視差画像は求められているが部分的におかしな値が出ていることを確認した。

使用方法 :

1. 付属のプロジェクトを Visual Studio (Visual Studio2013 以降のバージョン) で開く
2. Opencv2.4.9 をインストールする

(<http://imaging-solution.net/program/opencv/opencv2-4-dynamic-downlaod-install/>

を参考)

3. プロジェクトのプロパティから Opencv のライブラリの参照設定を行う

(読み込む画像の形式はデフォルトでは jpg

出力形式は png)