画像応用数学特論　最終レポート

２０１６年　１月　１９日 [redacted] 嘉戸琢真

１．　処理内容

- α拡張と，階層グラフカットアルゴリズムにより，ステレオマッチングを行う

２．　画像のフォーマット

- png 形式(横 1390 画素，縦 1110 画素)

３．　環境

- OS：Mac OS X (ver10.9.4)
- 開発環境：Xcode(ver5.1.1)
- プログラミング言語：C++
- プロセッサ：3GHz Intel Core i7
- メモリ：8GB 1600 MHz DDR3
- ライブラリ：OpenCV

４．　コストについて

- データコストは SAD を使用
- スムーズコストは $V(f_p,f_q)=c|f_p-f_q|$
  - ✧　（いくつか試した結果から　c=5,閾値は 8 として設定）

５．アルゴリズム

視差画像を読み込む
グラフのノード，エッジ数を計算
E=big number
for i = 0 to big number
　　for p = all pixels
　　　　ノードを追加，ソースとシンクにデータコストを設定
　　end for
　　for(p,q) = all adjacent pixels
　　　　ノードを追加
　　　　配列 A[i]中で，βp に一番近い値をαp とする
　　　　配列 A[i]中で，βq に一番近い値をαq とする
　　　　ソースとシンクにスムーズコストを設定
　　　　隣接するノードとのスムーズコストを設定
　　end for
　　max-flow　を使用
　　if flow < E
　　　　E = flow
　　　　ラベルを現在のラベルに置き換え
　　グラフを消去
end for
結果を出力

２０１６年　１月　１９日 [redacted] 嘉戸琢真

６．結果

- 以下に，α拡張と，階層グラフカットアルゴリズムによってステレオマッチングした画像とその結果を示す.

視差画像　左

視差画像右

α拡張

階層グラフカット

- 処理時間は以下のようになった.

α拡張	2786.77[s]
階層グラフカット	1353.79[s]

７．考察

- α拡張と階層グラフカットの結果は，互いに良好であることが見て取れる．若干階層グラフカットの結果が劣っているが，データコスト，スムーズコストのパラメータを細かく調整すれば，さらに結果が良くなると考えられる.