

画像応用数学特論

12月11日課題レポート

課題 α 拡張, 階層グラフカットでステレオマッチング

・環境

- OS : Windows7 Professional 32bit
- CPU : Intel(R) Core(TM) i5-2540M CPU @ 2.60GHz
- メモリ : 4.00GB
- 開発環境 : Microsoft Visual C++ 2010 Express
- プログラミング言語 : C/C++
- ライブラリ : OpenCV 2.4.9, MAXFLOW

・アルゴリズム

対応する2枚のステレオペアの画像ファイルを読み込み, α 拡張, 階層グラフカットのそれぞれを用いてステレオマッチングを行い, 視差を画像ファイルに出力する.

入力 : 対応するステレオペア画像(png形式) 2枚

出力 : 視差画像

使用したコスト関数は以下のとおりである.

・データコスト

SSD

$$D(f_{x,y}) = \sum (I_l(x,y) - (I_r(x - f_{x,y}, y)))^2$$

・スムーズコスト

$$V(f_p, f_q) = c|f_p - f_q|$$

この時 c は定数であり, エッジの強度により変化する.

また α 拡張, 階層グラフカットともに初期値は0とした.

α 拡張, 階層グラフカットのアルゴリズムを以下に示す.

```
•入力画像の読み込み
•初期値の設定
•E=とても大きな値
•for ループ=0~とても大きな値
    •success=0
    •for  $\alpha$  =0~DISPMAX
        •グラフの初期化
        •for 全てのピクセル
            •ノードの追加
            •ソースとシンクとのデータコストの設定
        •end for
        •for 全ての隣接点
            •ノードの追加
            •ソースとシンクとのスムーズコストの設定
            •隣接する画素のノードとのスムーズコストの設定
        •end for
        •最大流・最小カットアルゴリズムの適用
        •flow = 求まったラベルで計算した総コスト関数
        •flow < E のとき
            •E = flow
            •現在のラベル求まったラベルにする
            •success == 1
        •グラフの消去
    •end for
    •success = 0 ならループを脱出する
•end for
•視差画像の出力
```

図 1 : α 拡張のアルゴリズム

- 入力画像の読み込み
- 初期値の設定
- E =とても大きな値
- ラベル A の設定
- 出力画像 β の設定
- for ループ=0~とても大きな値
 - success=0
 - for $i=0 \sim |A|$
 - グラフの初期化
 - for 全てのピクセル
 - ノードの追加
 - $A[i]$ のうち, β_p に最も近い値を a_p に設定
 - ソースとシンクとのデータコストの設定
 - end for
 - for 全ての隣接点
 - ノードの追加
 - $A[i]$ のうち, β_p に最も近い値を a_p に設定
 - $A[i]$ のうち, β_q に最も近い値を a_q に設定
 - ソースとシンクとのスムーズコストの設定
 - 隣接する画素のノードとのスムーズコストの設定
 - end for
 - 最大流・最小カットアルゴリズムの適用
 - flow = 求まったラベルで計算した総コスト関数
 - flow < E のとき
 - $E = \text{flow}$
 - 現在のラベル求まったラベルにする
 - success == 1
 - グラフの消去
 - end for
 - success = 0 ならループを脱出する
- end for
- 視差画像の出力

図 2 : 階層グラフカットのアルゴリズム

・実行結果

ウィンドウサイズ W を 1, 3, 5, 9 にした際の α 拡張と階層グラフカットの結果を以下に示す. 図 3, 図 4 は入力画像である. また α 拡張と階層グラフカットの実行時間を比較したグラフを図 9 に示す.



図 3 : 左画像



図 4 : 左画像



(a) : α 拡張



(b) : 階層グラフカット

図 5 : ウィンドウサイズ 1



(a) : α 拡張

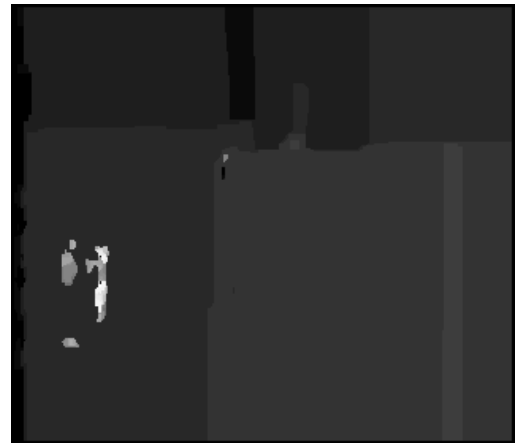


(b) : 階層グラフカット

図6 : ウィンドウサイズ3



(a) : α 拡張



(b) : 階層グラフカット

図7 : ウィンドウサイズ5



(a) : α 拡張



(b) : 階層グラフカット

図8 : ウィンドウサイズ9

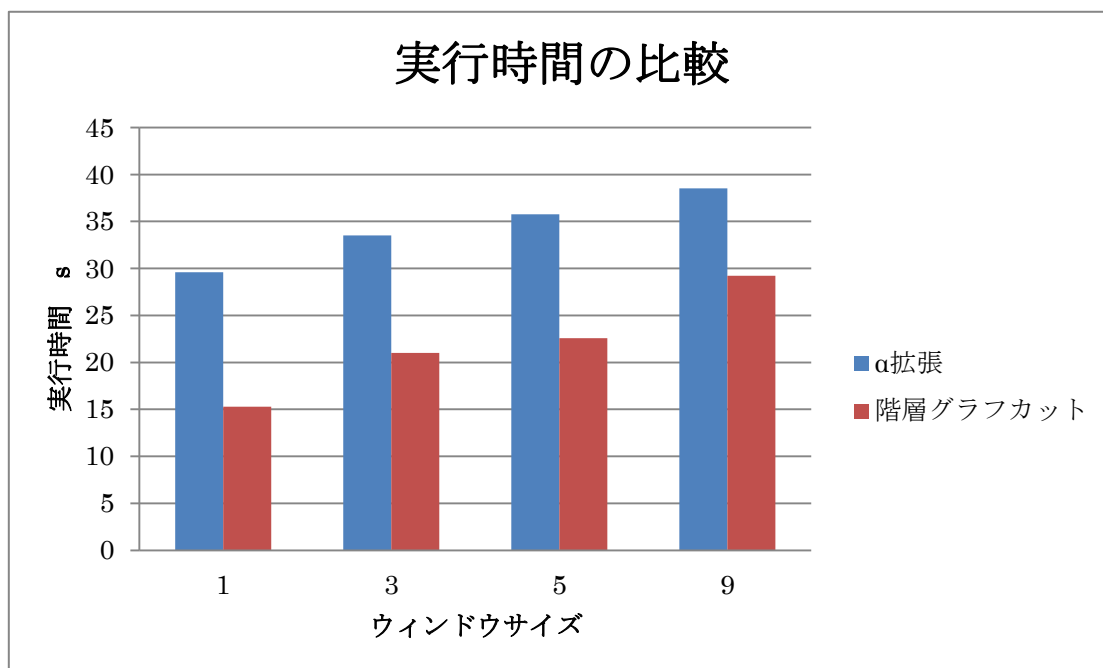


図 9 : α 拡張と階層グラウカットの実行時間の比較

・考察

実行結果より α 拡張, 階層グラフカットともに視差を出力できていることがわかるが, ノイズの乗っている部分もある. これはデータコストやスムーズコストの設定の仕方により, ノイズが乗ってしまったのではないかと考えられる. また初期値も今回は全て 0 に設定しており, 初期値をステレオマッチングの結果を用いるなどの工夫を加えることでさらに良い結果になるのではないかと思った.

また α 拡張と階層グラフカットの結果を比較すると階層グラフカットの方がノイズが多くみられる. これは, 階層グラフカットは視差を近似する形で求めており, 正しい値が正確に求まっていないためであると考えられる. しかし図 9 より実行時間を比較してみると, 階層グラフカットの方が全てのウィンドウサイズで早く, 実行スピードに関しては階層グラフカットの方がよい.

以上のことから処理速度に関しては, 階層グラフカットの方が良いが, 精度に関しては改良の余地が必要である. またウィンドウサイズを大きくしても精度が上がっていないことからウィンドウサイズは小さいほうが良いことがわかった.