画像応用数学特論　レポート課題　最終

2012/01/23 改め 2012/01/25

画像メディア工学

[redacted] 甲斐寛規

kai@cv.info.hiroshima-cu.ac.jp

以下，2 つの課題の速度や精度の比較を行う．

1. α拡張を用いたステレオマッチング
2. 階層グラフカットを用いたステレオマッチング

＜アルゴリズム＞

・α拡張

1 左，右画像読み込み
2 グラフのノード数、エッジ数を求める
3 出力画像の初期化
4 グラフのノードの数とエッジの数を求める
5 初期値 0 としてラベルを初期化する
6 E = とても大きな値
7 for 0～とても大きな値
    7.1 success = 0
    7.2 for i =0～最大視差
        7.2.1 グラフを初期化する
        7.2.2 ノードの追加を行う
        7.2.3 for p=すべてのピクセル
            7.2.3.1 ノードの、SOURCE 側に D(fp)、SINK 側に D(α)を設定
            7.2.3.2 ノード p とノード a のエッジの重み(双方向)に V(fp, α)を設定
            7.2.3.3 ノード q とノード a のエッジの重み(双方向)に V(α, fq)を設定
        7.2.4 end for
        7.2.5 最大流・最小カットアルゴリズムの適用
        7.2.6 E' = 求まったラベルで計算した総コスト関数
        7.2.7 E' < E なら、現在のラベルを求まったラベルにし、E = E'にし、success = 1 にする
        7.2.8 グラフの消去
    7.3 end for
    7.4 もし success == 0 ならループ脱出
8 end for
9 0-255 へ濃度階調変換を行う

・階層グラフカット

1 左，右画像読み込み

2 グラフのノード数、エッジ数を求める

3 出力画像の初期化

4 初期値 0 としてラベルを初期化する

5 階層構造の配列 A を計算する

6 E = とても大きな値

7 for 0～とても大きな値

7.1 success = 0

7.2 for i =0～階層構造の配列の数

7.2.1 グラフを初期化する

7.2.2 ノードの追加を行う

7.2.2.1 for p=すべてのピクセル A[i]のうち、βp に最も近い値をαp とする

7.2.2.2 ノードの、SOURCE 側に D(βp)、SINK 側に D(αp)を設定

7.3 end for

7.4 for(p,q) = すべての隣接点

7.4.1 A[i]のうち，βp に最も近い値をαp に設定

7.4.2 A[i]のうち，βq に最も近い値をαq に設定

7.4.3 ノード a の，SOURCE 側に V(βp，βq), SINK 側に V(αp，αq)を設定

7.4.4 もし、V(βp，βq) <= V(αp，αq)の場合

7.4.4.1 ノード a からノード p のエッジの重みに 10000 を設定

7.4.4.2 ノード p からノード a のエッジの重みに, V(αp，βq) - V(βp，βq)か 0 のうち大きいほうを設定

7.4.4.3 ノード a からノード q のエッジの重みに 10000 を設定

7.4.4.4 ノード q からノード a のエッジの重みに, V(βp，αq) - V(βp，βq)か 0 のうち大きいほうを設定

7.4.5 もし、V(βp，βq) > V(αp，αq)の場合

7.4.5.1 ノード p からノード a のエッジの重みに 10000 を設定

7.4.5.2 ノード a からノード p のエッジの重みに, V(αp，βq) - V(βp，βq)か 0 のうち大きいほうを設定

7.4.5.3 ノード q からノード a のエッジの重みに 10000 を設定

7.4.5.4 ノード a からノード q のエッジの重みに, V(βp，αq) - V(βp，βq)か 0 のうち大きいほうを設定

7.4.6 end for

7.4.7 最大流・最小カットアルゴリズムの適用

7.4.8 E' = 求まったラベルで計算した総コスト関数

7.4.9　E' < E なら、現在のラベルを求まったラベルにし、E = E'にし、success = 1 にする

7.5　グラフの消去　end for

7.6　もし success == 0 ならループ脱出

8　# end for

9　# 0-255 へ濃度階調変換を行う

＜データコスト＞

SAD　：　$\mathrm{D}(f_p) = |f_p - I_p|$

＜スムーズコスト＞

$$\mathrm{V}(a,b) = \mathrm{c}|a-b| \qquad \mathrm{c}=1$$

用いた画像を図 1(a)，(b)に示す．これらを入力とし，得られたα拡張を用いたステレオマッチングの結果を図 2(a)に，階層グラフカットを用いたステレオマッチングの結果を図 2(b)に示す．

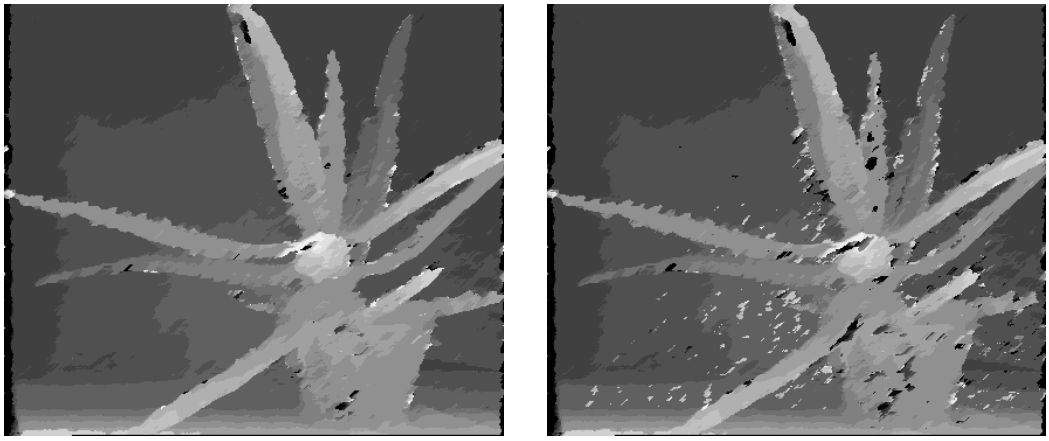
(a) 左　　(b)　右

図 1：入力画像

＜計算時間＞

・α拡張を用いたステレオマッチング　：　12.42 [s]

・階層グラフカットを用いたステレオマッチング　：　13.93 [s]



(a) α拡張　　(b) 階層グラフカット

図 2：出力結果

<考察>

出力結果を見てわかるように，α拡張を用いたステレオマッチングの方が精度が高いとわかる．また，計算時間についても，今回においてはα拡張を用いている方が早い．講義では，階層グラフカットの方が計算時間が早いと聞いていたので意外な結果が出た．

これは，どちらも最大視差を 16 にしているため，この程度では計算時間に影響を及ぼさないのではないかと考える．

＊使用ライブラリ

MAXFLOW　http://pub.ist.ac.at/~vnk/software.html

OpenCV　http://opencv.jp/download

＊開発環境

OS：CentOS5 (VMWare)

CPU：Intel(R) Core(TM) 2 Duo

コンパイラ：g++　4.1.2 20080704 (Red Hat 4.1.2-51)

プログラミング言語：C++言語

＊コンパイル方法

階層グラフカット：cv graph.cpp maxflow.cpp graph.h block.h main(kai_kai).cpp -o ~~

α拡張：cv graph.cpp maxflow.cpp graph.h block.h main(kai_a).cpp -o ~~

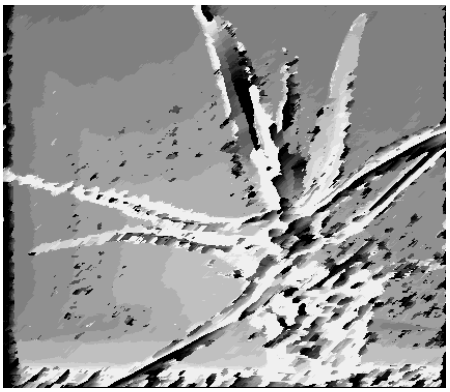
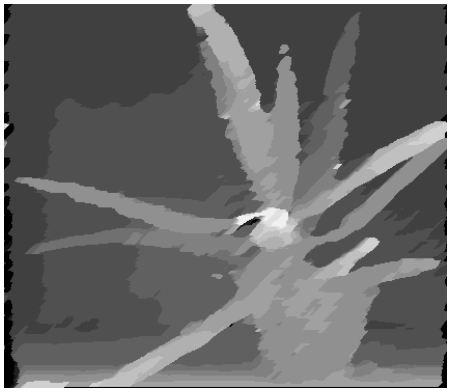
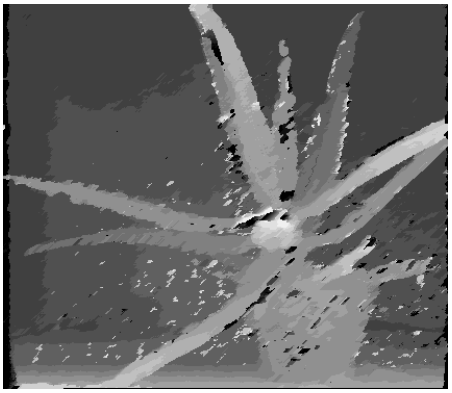
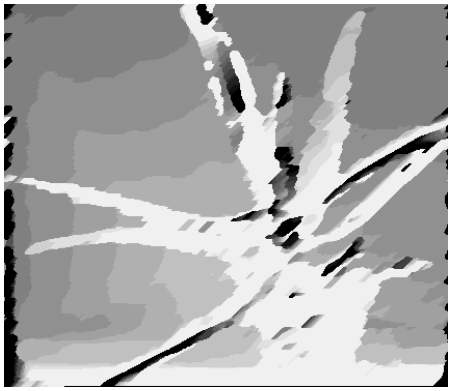
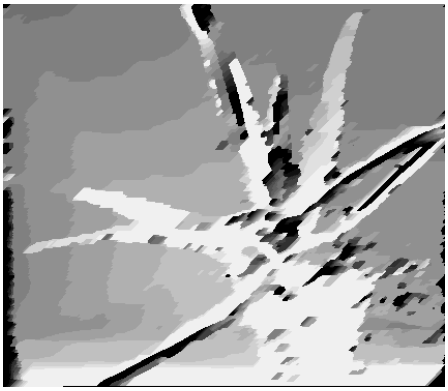
＊実行方法

./~~ [左画像] [右画像] [出力画像]　　（だいたいの画像形式に対応）

＊その他の出力画像

http://vision.middlebury.edu/stereo/data/scenes2006/ThirdSize/zip-7views/

以上からダウンロードした画像での視差画像である.

α拡張	階層グラフカット
	
view0.png -- view2.png	
16.86 [s]	17.01 [s]
	
view1.png -- view2.png	
12.54 [s]	19.91 [s]
	
view1.png -- view3.png	
16.90 [s]	19.88 [s]