

提出日：2013/01/15

学籍番号：[REDACTED]

氏名：河野歩実

e-mail：kawano☆ime.info.hiroshima-cu.ac.jp(☆→@)

画像応用数学特論 レポート

[課題]視点が異なる2つ画像において、階層グラフカットによるステレオマッチングを行い、視差画像を出力として考察を行う。

実行環境

OS	CentOS release 5.5 (Windows 7 Professional 上で VMware Player により実行)
コンパイラ	g++ ver.4.1.2
使用言語	C++
使用ライブラリ	OpenCV 1.1 maxflow(http://pub.ist.ac.at/~vnk/software.html)
CPU	Intel(R)Core(TM)i5-2540M CPU @ 2.60GHz(仮想マシンへは1コア)
メモリ	4GB(仮想マシンへは1GB)

実行方法

ソースファイル一式を DL

↓

コンパイル

(階層グラフカットの場合) `g++ -ggdb `pkg-config --cflags opencv` `pkg-config --libs opencv` graph.cpp maxflow.cpp graph.h block.h hierarchical.cpp -o hierarchical`

(α 拡張の場合) `g++ -ggdb `pkg-config --cflags opencv` `pkg-config --libs opencv`
graph.cpp maxflow.cpp graph.h block.h alpha_stereo.cpp -o alpha_stereo`

↓

実行

(階層グラフカットの場合) `./hierarchical 左画像 右画像 出力画像 最大視差`

(α 拡張の場合) `./alpha_stereo 左画像 右画像 出力画像 最大視差`

実装したアルゴリズム

・変数宣言
・for p=全てのピクセル
・fp=初期値
・end for
・A={{0},...,{3,...,64}}
・E=とても大きな値
・for ループ=0~とても大きな値
・success=0
・for i=0~11
・グラフの初期化
・全てのノードの追加
・for p=全てのピクセル
・A[i]のうち、beta_p に最も近い値を alpha_p に設定
・ノードのソース側に D(beta_p)、D(alpha_p)を設定
・end for
・for(p,q)=全ての隣接点
・A[i]のうち、beta_p に最も近い値を alpha_p に設定
・A[i]のうち、beta_q に最も近い値を alpha_q に設定
・もし、 $V(\beta_p, \beta_q) \leq V(\alpha_p, \alpha_q)$ の場合
・ノード a からノード p へのエッジの重みに 10000 を
設定
・ノード a からノード q へのエッジの重みに 10000 を
設定
・ノード p からノード a へのエッジの重みに、
$V(\alpha_p, \beta_q) - V(\beta_p, \beta_q)$ か0のうち大きいほうを設定
・ノード q からノード a へのエッジの重みに、
$V(\beta_p, \alpha_q) - V(\beta_p, \beta_q)$ か0のうち大きいほうを設定
・もし、 $V(\beta_p, \beta_q) > V(\alpha_p, \alpha_q)$ の場合
・ノード p からノード a へのエッジの重みに 10000 を
設定
・ノード q からノード a へのエッジの重みに 10000 を
設定
・ノード a からノード q へのエッジの重みに、

$V(\alpha_p, \beta, q) - V(\alpha_p, \alpha_q)$ が0のうち大きいほうを設定
 ・ノード a からノード p へのエッジの重みに、
 $V(\beta_p, \alpha, q) - V(\alpha_p, \alpha_q)$ が0のうち大きいほうを設定
 ・ end for
 ・最大流・最小カットアルゴリズムの適用
 ・ E' =求めたラベルで計算した総コスト関数
 ・ $E' < E$ なら、現在のラベルを求めたラベルにし、 $E = E'$ にし、success
 =1にする
 ・グラフの消去
 ・ end for
 ・もし success == 0 ならループを脱出
 ・ end for

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

コストについて

・データコスト

・スムーズコスト

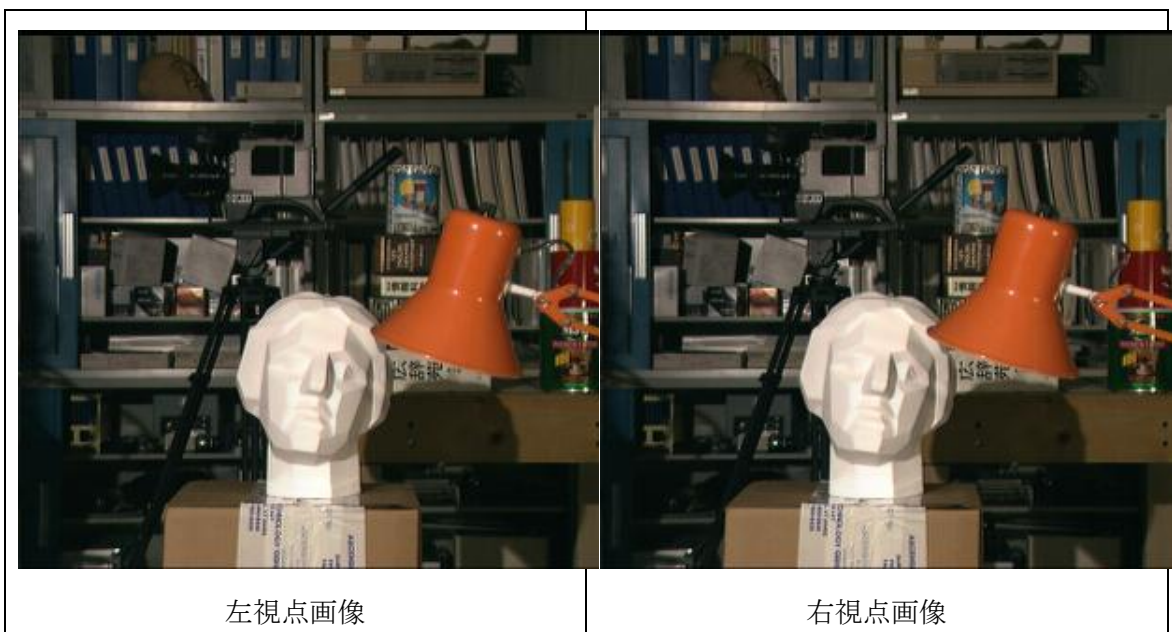
$$S(L_p, L_q) = c |L_p - L_q|$$

実行結果

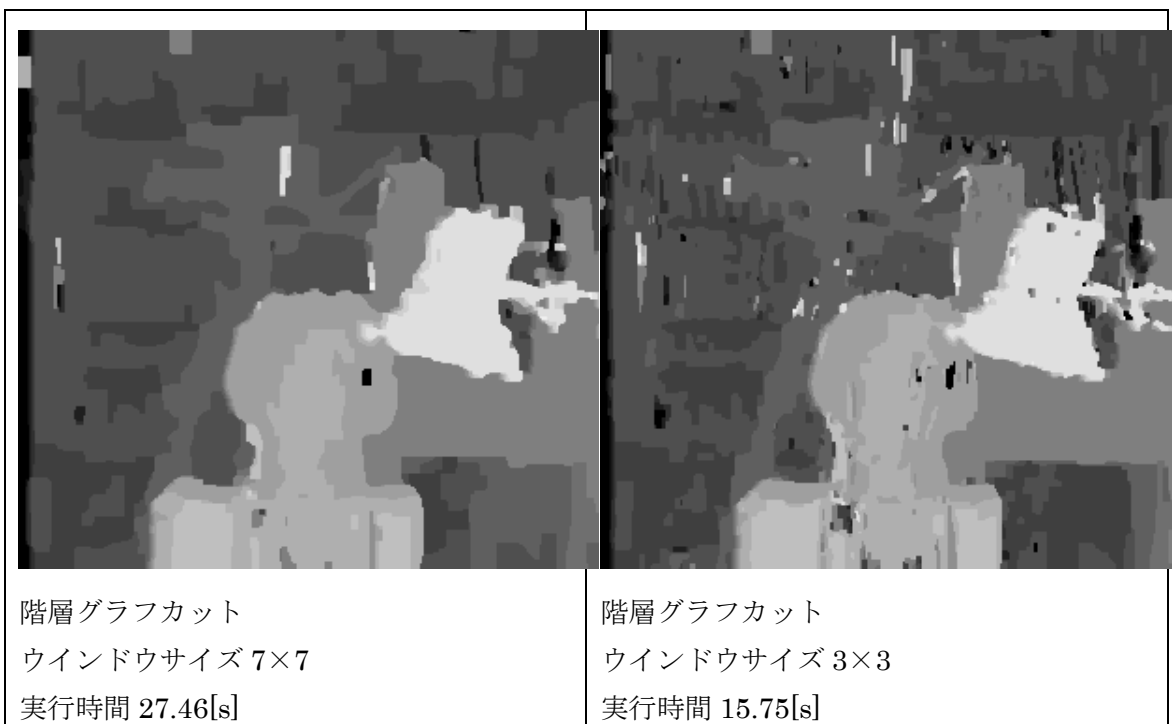
以下の2枚のステレオ画像を用いて階層グラフカットと α 拡張を用いたグラフカットによるステレオマッチングの結果を以下に示す。

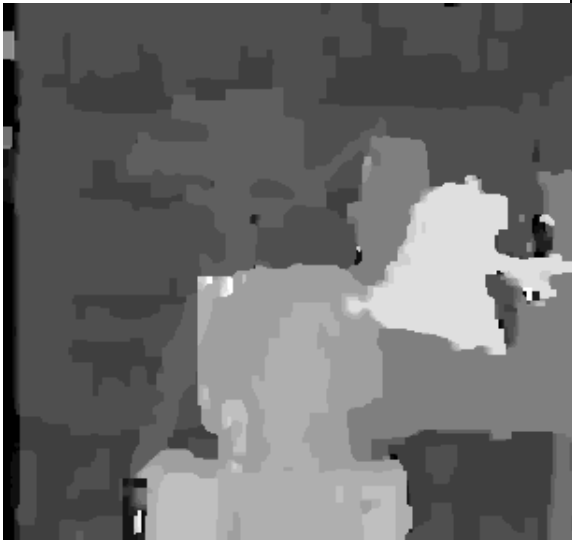

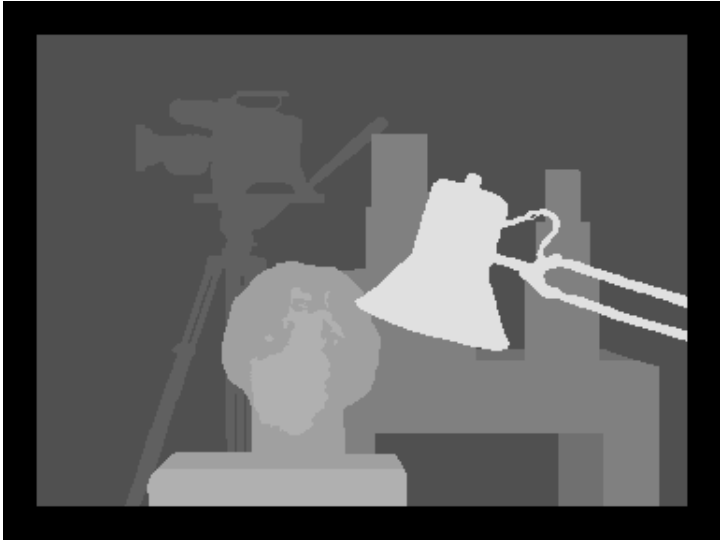
データコストに用いるウィンドウサイズを変更して実行時間と精度に関する考察を行う。

入力画像



出力画像



	
<p>α 拡張を用いたグラフカット ウインドウサイズ 7×7 実行時間 136.84[s]</p>	<p>α 拡張を用いたグラフカット ウインドウサイズ 3×3 実行時間[s]</p>
<div style="text-align: center;">  <p>真値</p> </div>	

考察

階層グラフカットと α 拡張によるステレオマッチングを比較すると、実行時間に関しては階層グラフカットの方が圧倒的に早い結果となった。ラベルを階層的に振り分けることでループ回数が減ったためであると考えられる。精度に関しては見た目の評価ではあるが、真値と比較すると α 拡張を用いたグラフカットの方がやや高い。ウインドウサイズの変更

による結果の違いでは、 3×3 の方よりも 7×7 の方が滑らかになることが分かった。しかし、これ以上ウインドウサイズを大きくしてしまうと実行時間に影響が出ると考えられる。