処理内容：二つの画像の視差の計算（ステレオ計算）

アルゴリズム

2つの画像ファイルを読み込む

階層グラフカットに使うラベル配列Labelを定義（以下現在のラベルをβ，βにいちばん近いラベルをαとする）

二つの配列にそれぞれの画像ファイルのRGBの情報を保存（左側から見た画像を画像L、右側から見た画像を画像Rとする）

現在のラベルを保存する配列imagedataのために画像R，Lと同じ大きさのメモリを用意する

Eのループを開始

iのループを開始

縦の画素数をy横の画素数をxとすると，x\*y + (x-1)\*y + x\*(y-1)個のノードを定義

（画素に対応するノード＋横の画素間のノード＋縦の画素間のノード）

全てのノードとソース，シンクを辺で結ぶ

画素に対応するノードp,q間を，間のノードaを介して辺で結ぶ

（これを図で表すと以下のようになる）

最大流・最小カットアルゴリズムの適用

E’=求まったラベルで計算した総コスト関数

E’<Eなら，

全てのノードを調べ，最大流・最小カットアルゴリズムの適用でソース側に分けられたなら

そのノードの現在のラベルを求まったラベルαにする，

E=E’とし，succsee=1にする

グラフの消去

　　iの値を+1する

　iが決まった値に達したらループを終了

もしsuccess==0ならループを脱出する

Eが決まった値に達したらループを終了

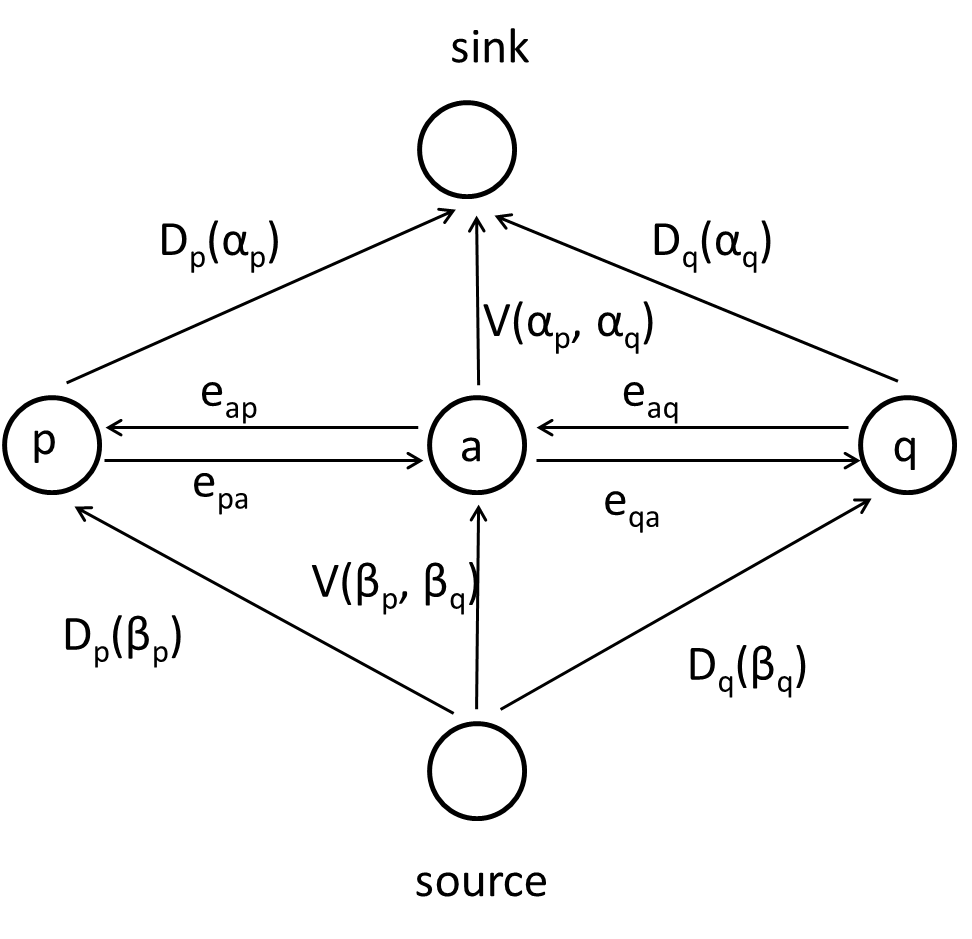
結果を出力する

ノードとソース，シンクをつなぐ辺のコストであるデータコストとノード間の辺のコストであるスムーズコストは以下のように設定した

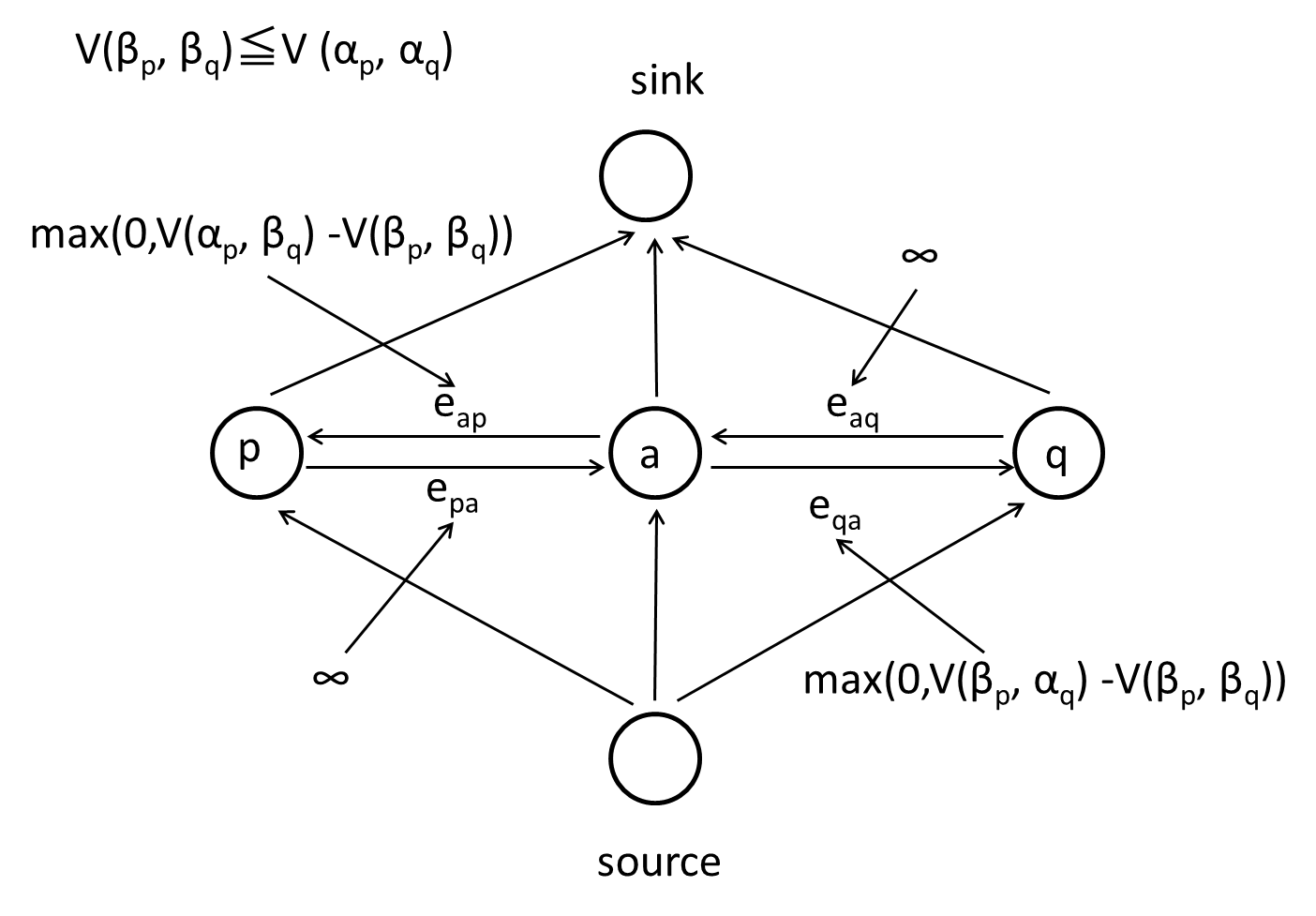
データコスト 

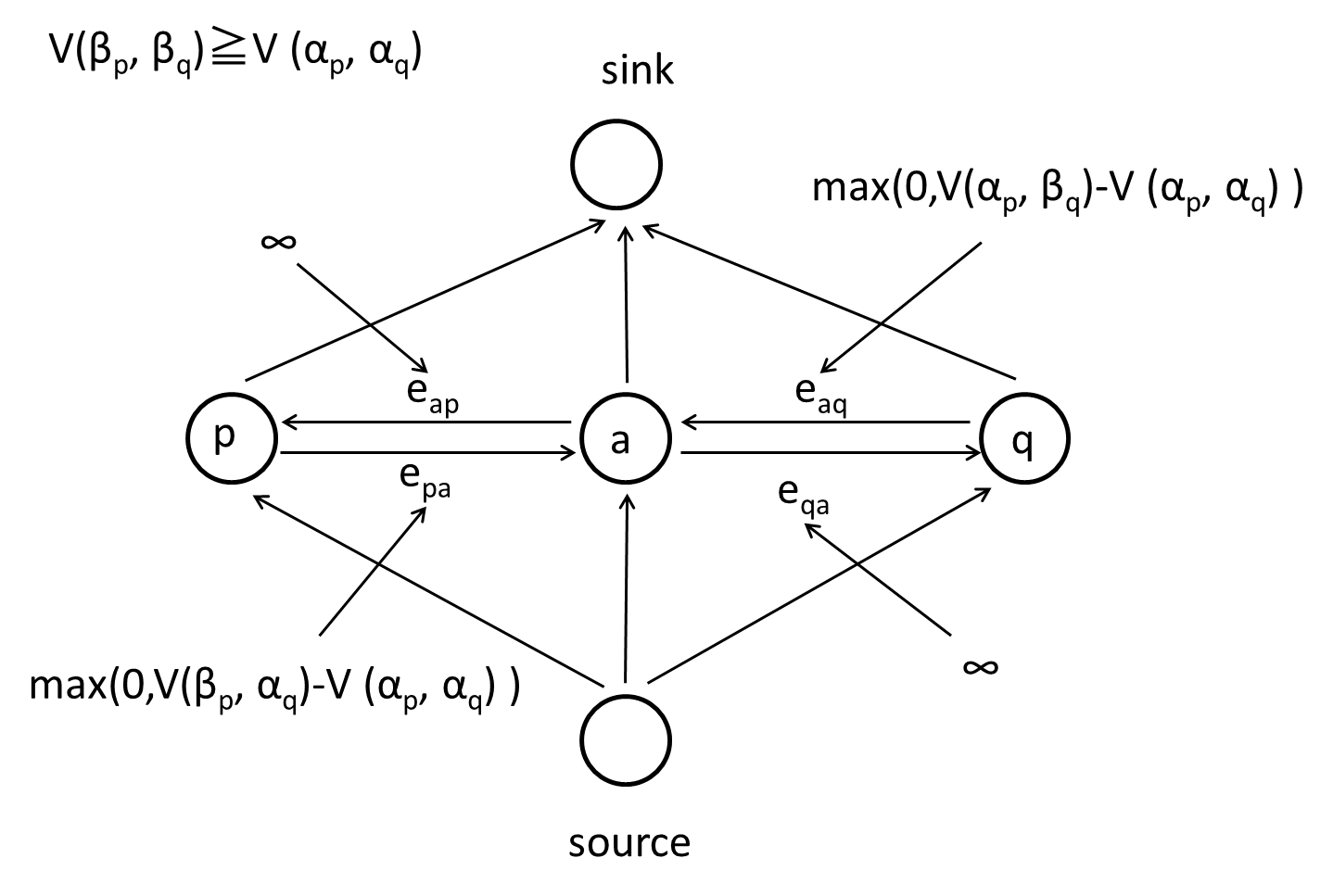
スムーズコスト  cは定数

ノードとエッジの図



スムーズコストeの値は2つの場合に分けられる





使用するOS：Windows7

開発環境：Visual Studio 2008

プログラミング言語：C言語

CPU：Intel® Xeon® CPU X5460 @ 3.16GHz 3.16GHz

メモリ：4.00 GB

画像形式：ppm

コンパイル方法：Visual Studio 2008の同プロジェクトのフォルダ内にソースファイル,graph.cpp.maxflow.cpp,block.h,graph.h,instancesを保存．

Visual Studio 2008のソリューションエクスプローラ内のソースファイルフォルダにgraph.cpp.maxflow.cppをヘッダーファイルフォルダにblock.h,graph.h,instancesを追加

その後メニューバーのビルドからソリューションのビルドを選択．

実行方法：メニューバーのデバッグからデバッグ無しで開始を選択．

ライブラリのインストール方法：web上から圧縮ファイルをダウンロード後展開，中に含まれているファイルをVisual Studio 2008のプロジェクトフォルダ内にコピーする．

計算に使用した画像

左の視点からの画像



右の視点からの画像



階層グラフカットでの出力結果



α拡張との比較

α拡張の結果



かかった時間

α拡張37.48秒 階層グラフカット25.86秒

精度α拡張の結果の方が若干輪郭がはっきりしているように見える．α拡張より階層グラフカットの方が全体的にノイズの少ない滑らかな画像となっているように見える．