Interactive Digital Photomontage

Aseem Agarwala, Mira Dontcheva et al.
University of Washington
SIGGRAPH' 04
Presenter: 林昭炜

Outline

- Segmentation
  - Ineraction Penalty
  - Inertia
- Gradient domain fusion
- Results
- Discussions

Using Graph-cut

- InnerLoop: Given a Label α and Current Label, accept label that has less costs
- OuterLoop: Iterates for all possible labels

Graph-cut

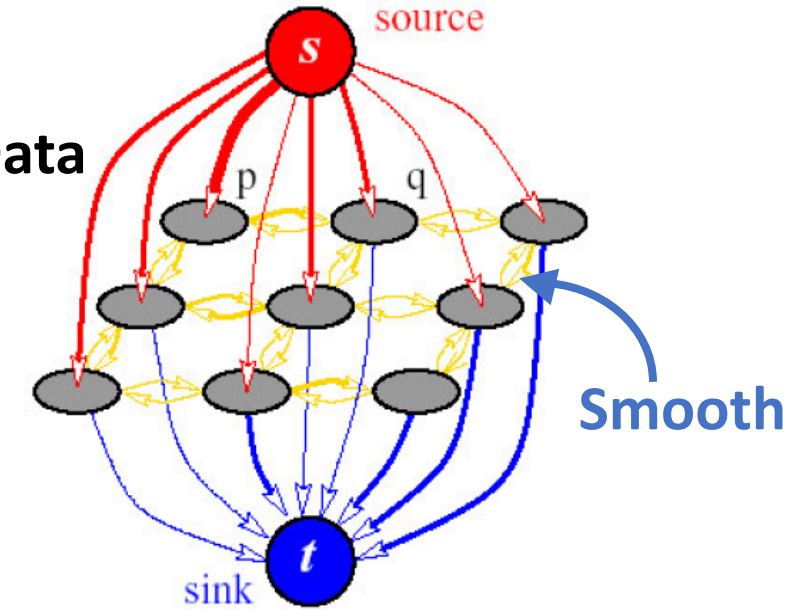
$$C(l) = \sum_{p} C_d\big(p, L(p)\big) + \sum_{p,q} C_i(p, q, L(p), L(\mathrm{q}))$$

Graph-cut

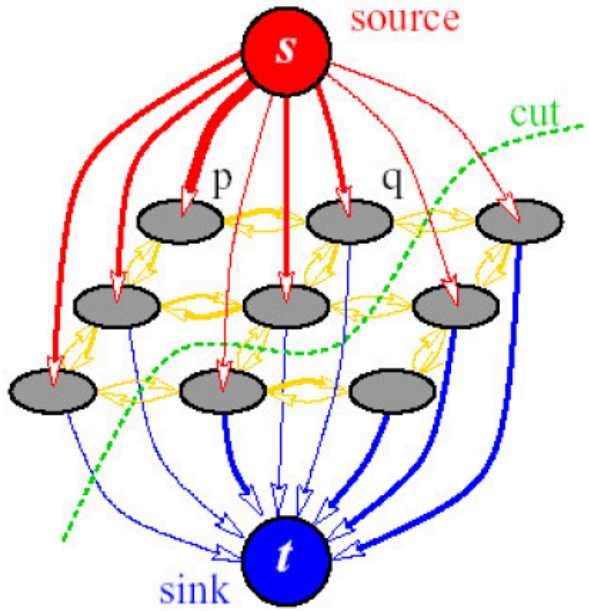
$$C(l) = \sum_{\boldsymbol{p}} \boldsymbol{C_d}\left(\boldsymbol{p}, \boldsymbol{L}(\boldsymbol{p})\right) + \sum_{p,q} C_i(p, q, L(p), L(\mathrm{q}))$$

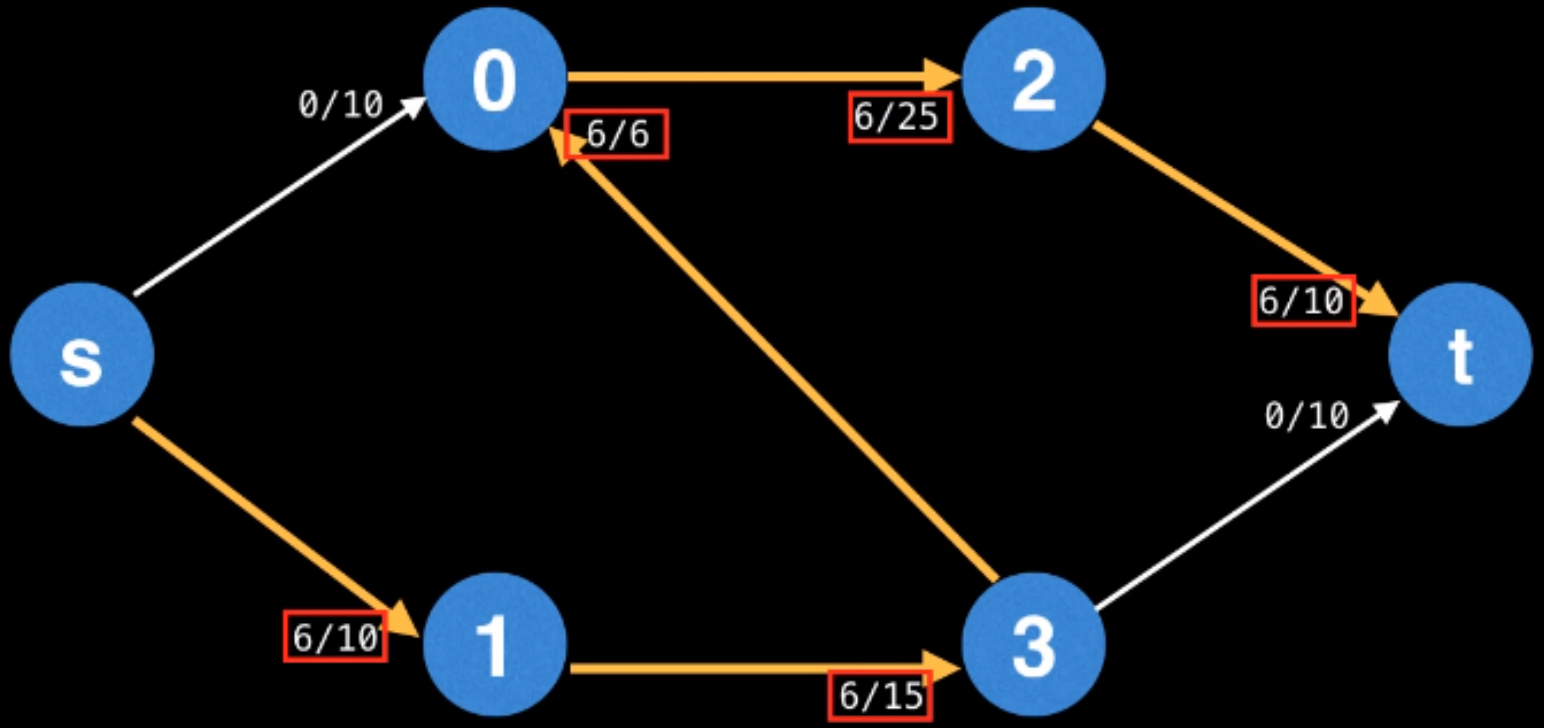
Data Penalty

Data

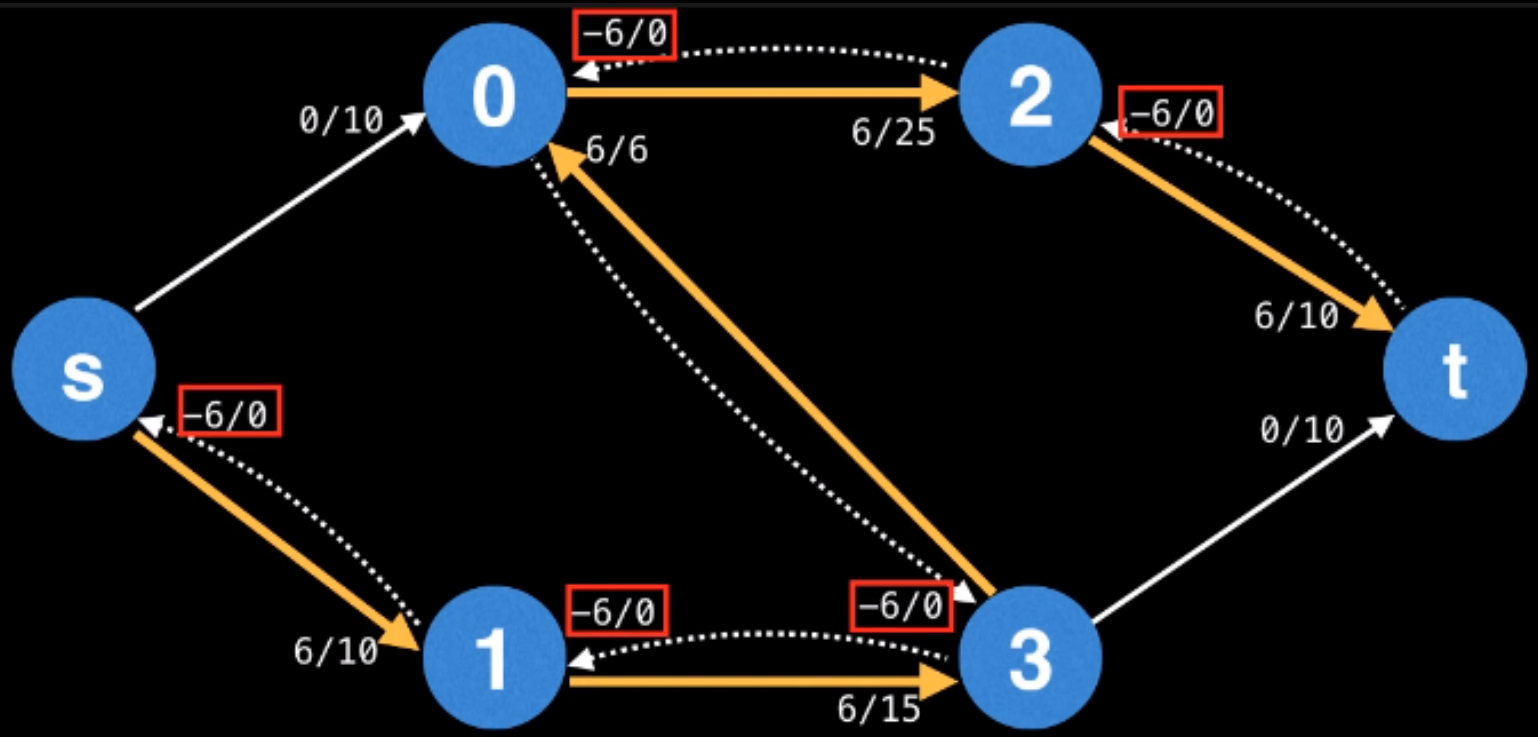


Smooth



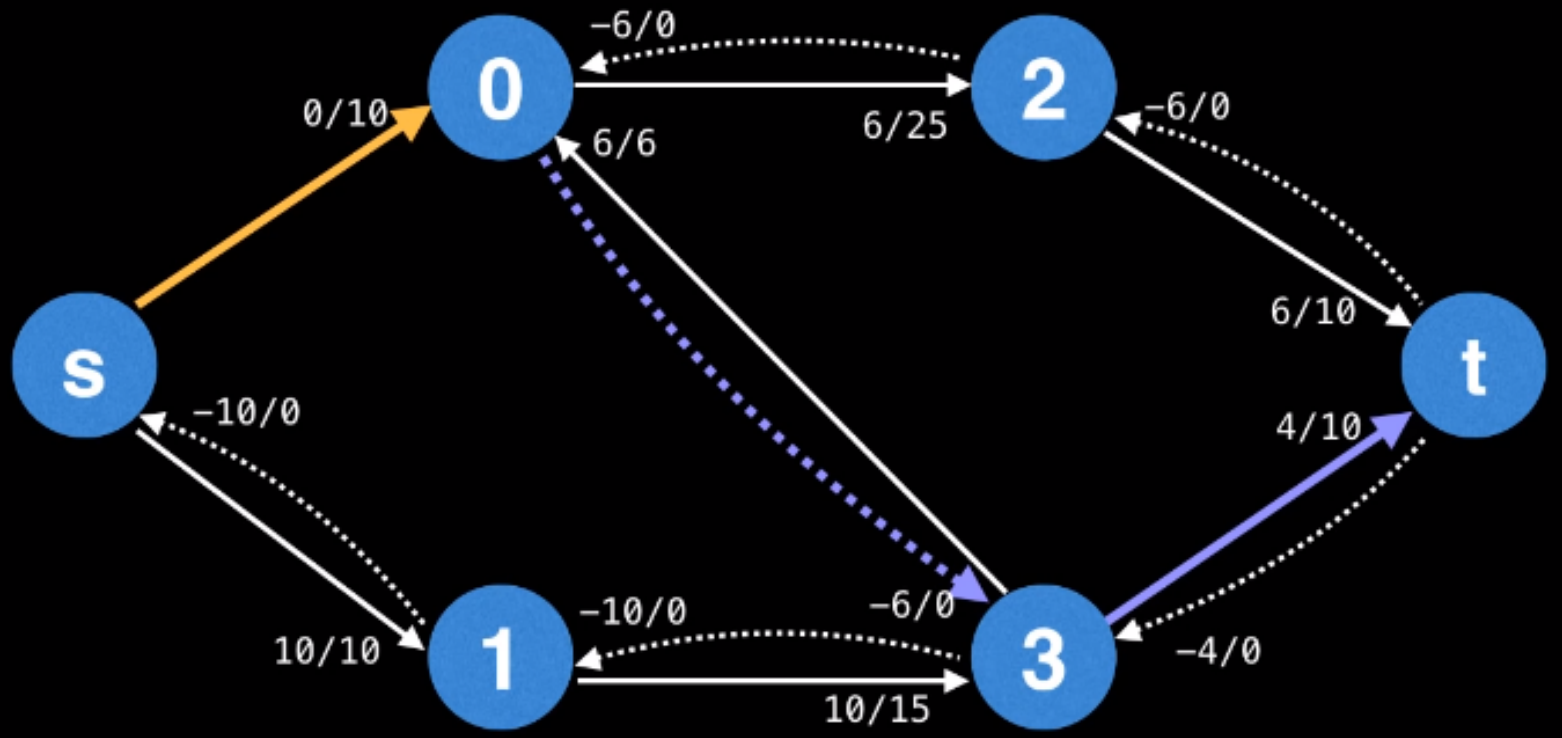


Step1: Find A path



Step1: Find A path

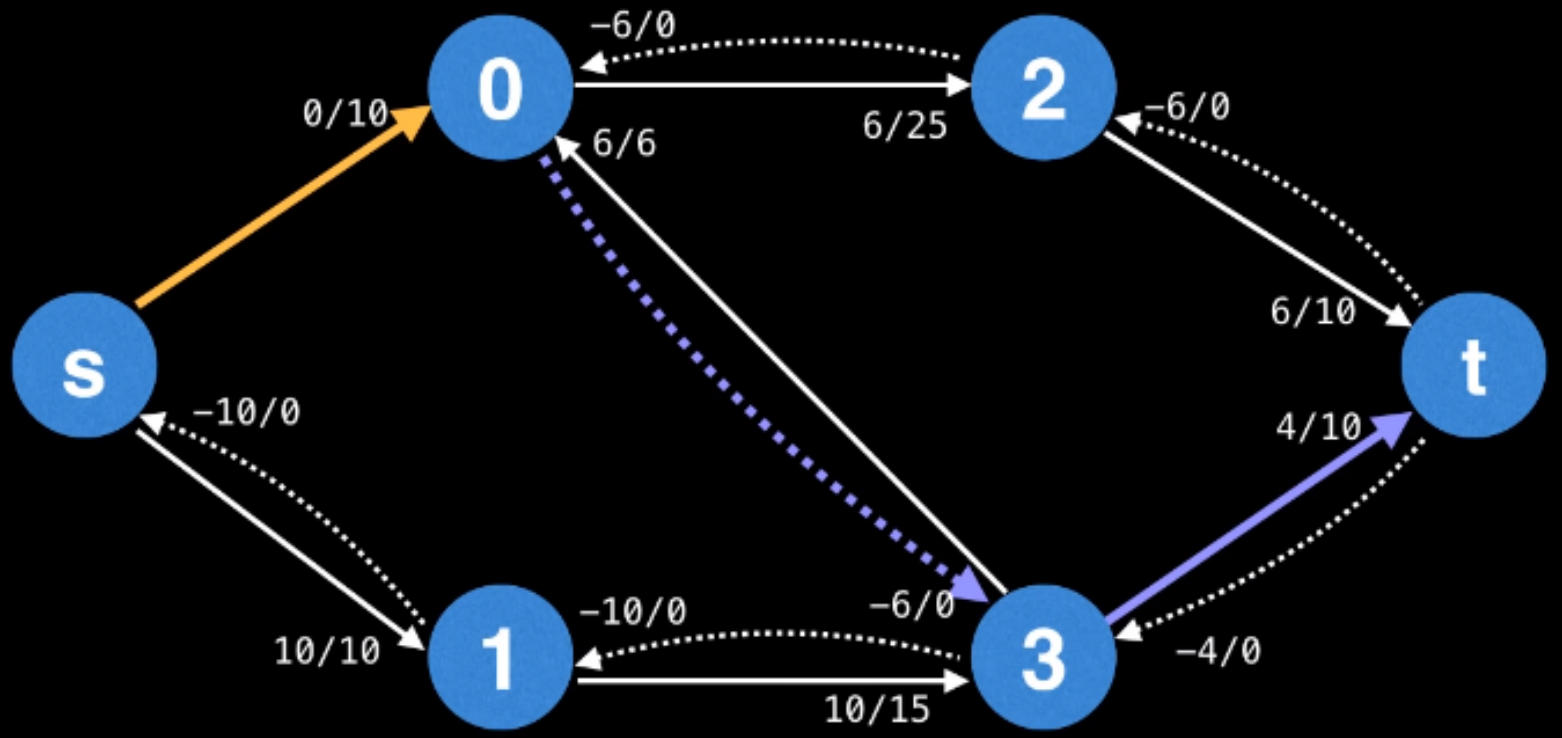
Step2: Add Residual Edge



Step1: Find A path

Step2: Add Residual Edge

Step3: Repeat Step1, Considering new Reidual Edge

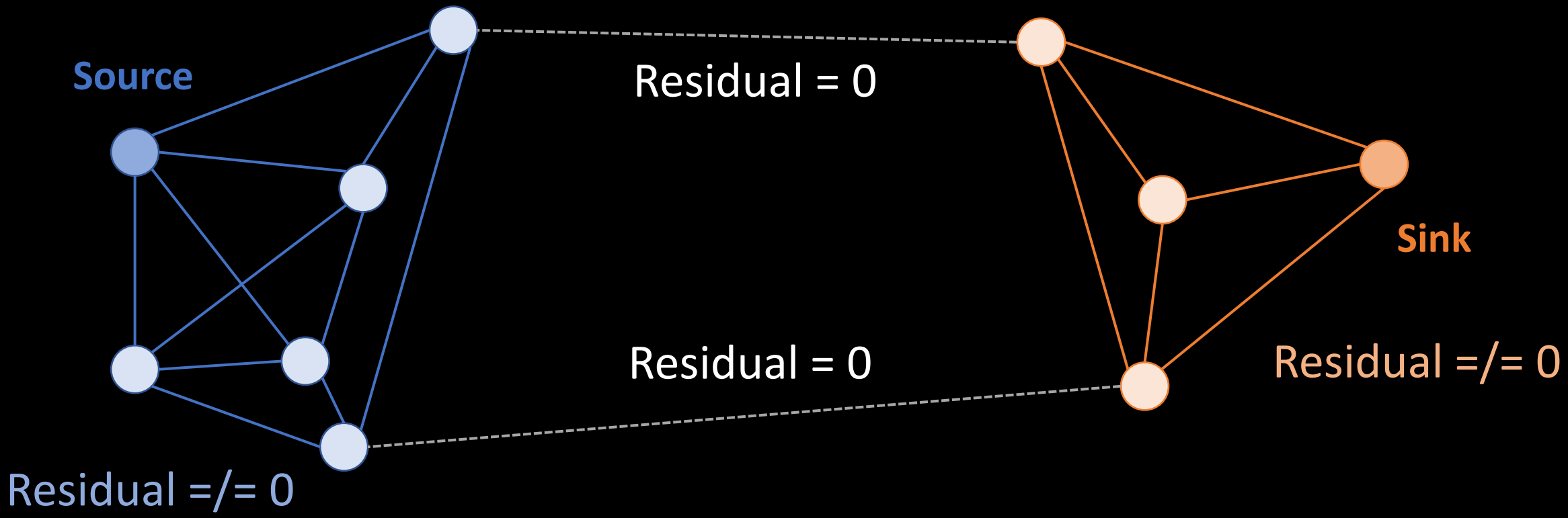


Step1: Find A path **(DFS/BFS)**

Step2: Add Residual Edge

Step3: Repeat Step1, Considering new Reidual Edge

Graph-cut: Min-cut

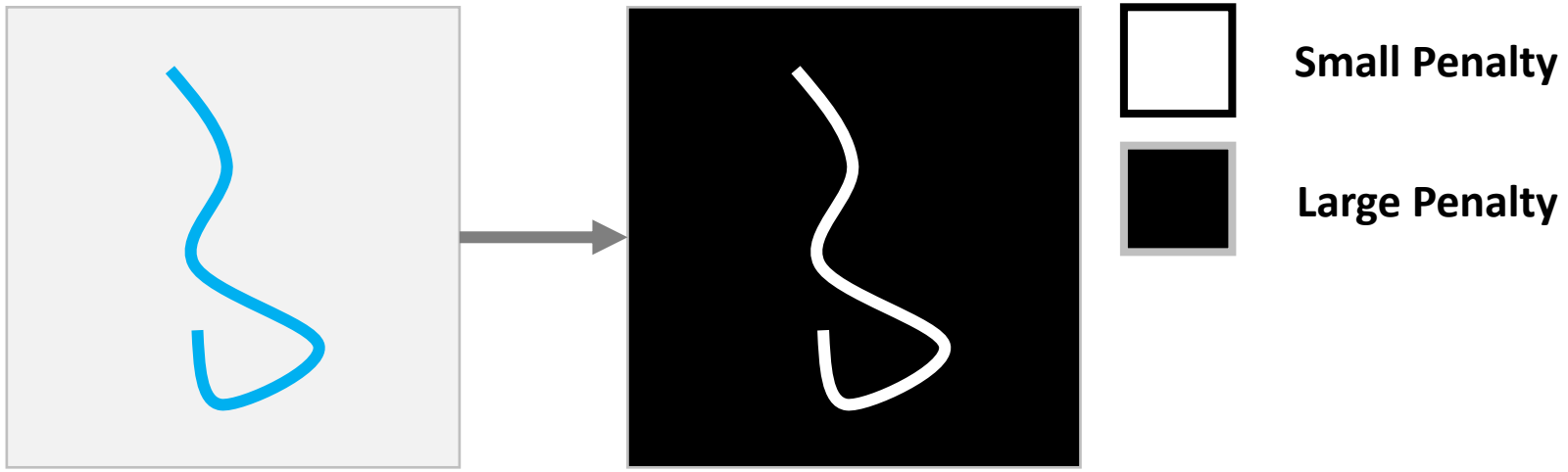


Interaction Penalty

$$C(l) = \sum_{p} \boldsymbol{C_d(p, L(p))} + \sum_{p,q} C_i(p, q, L(p), L(\mathrm{q}))$$

Data Penalty

Designated Image:

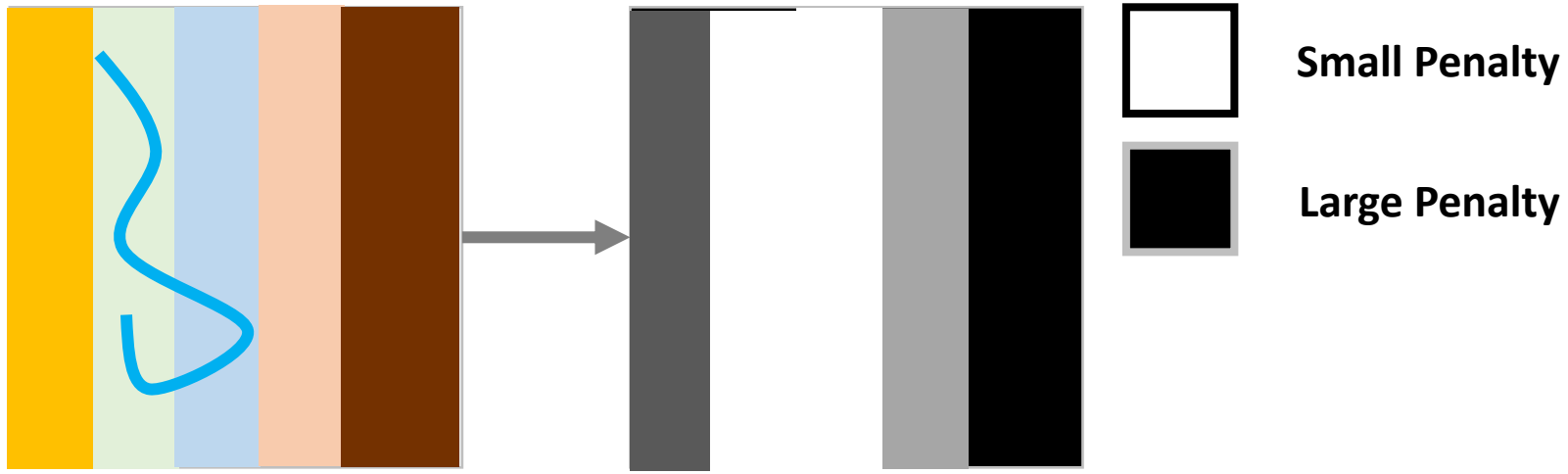


Interaction Penalty

$$C(l) = \sum_{\boldsymbol{p}} \boldsymbol{C_d}\left(\boldsymbol{p}, \boldsymbol{L(p)}\right) + \sum_{p,q} C_i(p, q, L(p), L(\mathrm{q}))$$

Data Penalty

Designated Color:
(Least/Max Simliarity)

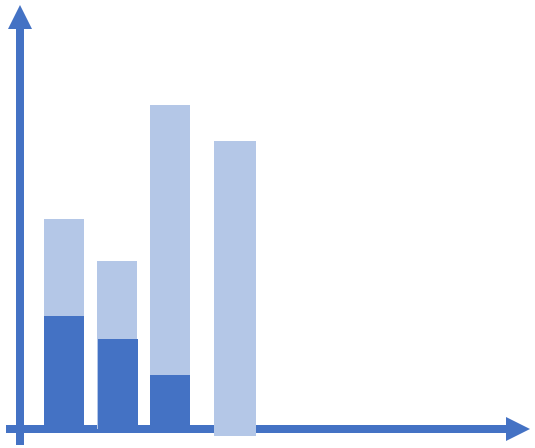


Interaction Penalty

$$C(l) = \sum_{\boldsymbol{p}} \boldsymbol{C_d}(\boldsymbol{p}, \boldsymbol{L}(\boldsymbol{p})) + \sum_{p,q} C_i(p, q, L(p), L(\mathrm{q}))$$

Data Penalty

Designated Color
Luminance
Likelihood



Interaction Penalty

$$C(l) = \sum_{\boldsymbol{p}} \boldsymbol{C_d}\left(\boldsymbol{p}, \boldsymbol{L(p)}\right) + \sum_{p,q} C_i(p, q, L(p), L(\mathrm{q}))$$

Data Penalty

Designated Color
Luminace
Likelihood
Eraser
…

Interaction Penalty

$$C(l) = \sum_{p} C_d\big(p, L(p)\big) + \sum_{\boldsymbol{p,q}} \boldsymbol{C_i(p, q, L(p), L(q))}$$

Seam Penalty

Match Colors: $\left|\left|\boldsymbol{S_{L(p)}(p) - S_{L(q)}(p)}\right|\right| + \left|\left|\boldsymbol{S_{L(p)}(q) - S_{L(q)}(p)}\right|\right|$

Match Gradients

Match Colors & Gradients

Match Colors & Edges

Interaction Penalty

$$C(l) = \sum_{p} C_d(p, L(p)) + \sum_{p,q} \boldsymbol{C_i(p, q, L(p), L(q))}$$

Seam Penalty

Match Colors: $\left|\left|\boldsymbol{S_{L(p)}(p) - S_{L(q)}(p)}\right|\right| + \left|\left|\boldsymbol{S_{L(p)}(q) - S_{L(q)}(p)}\right|\right|$

Match Gradients

Match Colors & Gradients

Match Colors & Edges

Interaction Penalty

$$C(l) = \sum_{p} C_d\big(p, L(p)\big) + \sum_{\boldsymbol{p,q}} \boldsymbol{C_i(p, q, L(p), L(q))}$$

Seam Penalty

Match Colors: $\left|\left|\boldsymbol{S_{L(p)}(p) - S_{L(q)}(p)}\right|\right| + \left|\left|\boldsymbol{S_{L(p)}(q) - S_{L(q)}(p)}\right|\right|$

Match Gradients: $\left\|\boldsymbol{\nabla S_{L(p)}(p) - \nabla S_{L(q)}(p)}\right\| + \left\|\boldsymbol{\nabla S_{L(p)}(q) - \nabla S_{L(q)}(q)}\right\|$

Match Colors & Gradients:

Match Colors & Edges

Interaction Penalty

$$C(l) = \sum_{p} C_d\big(p, L(p)\big) + \sum_{\boldsymbol{p,q}} \boldsymbol{C_i(p, q, L(p), L(q))}$$

Seam Penalty

Match Colors: $\left|\left|\boldsymbol{S_{L(p)}(p) - S_{L(q)}(p)}\right|\right| + \left|\left|\boldsymbol{S_{L(p)}(q) - S_{L(q)}(p)}\right|\right|$

Match Gradients: $\left\|\boldsymbol{\nabla S_{L(p)}(p) - \nabla S_{L(q)}(p)}\right\| + \left\|\boldsymbol{\nabla S_{L(p)}(q) - \nabla S_{L(q)}(q)}\right\|$

Match Colors & Gradients: Match Gradients + Match Colors

Match Colors & Edges :

Interaction Penalty

$$C(l) = \sum_{p} C_d\big(p, L(p)\big) + \sum_{\boldsymbol{p,q}} \boldsymbol{C_i(p, q, L(p), L(q))}$$

Seam Penalty

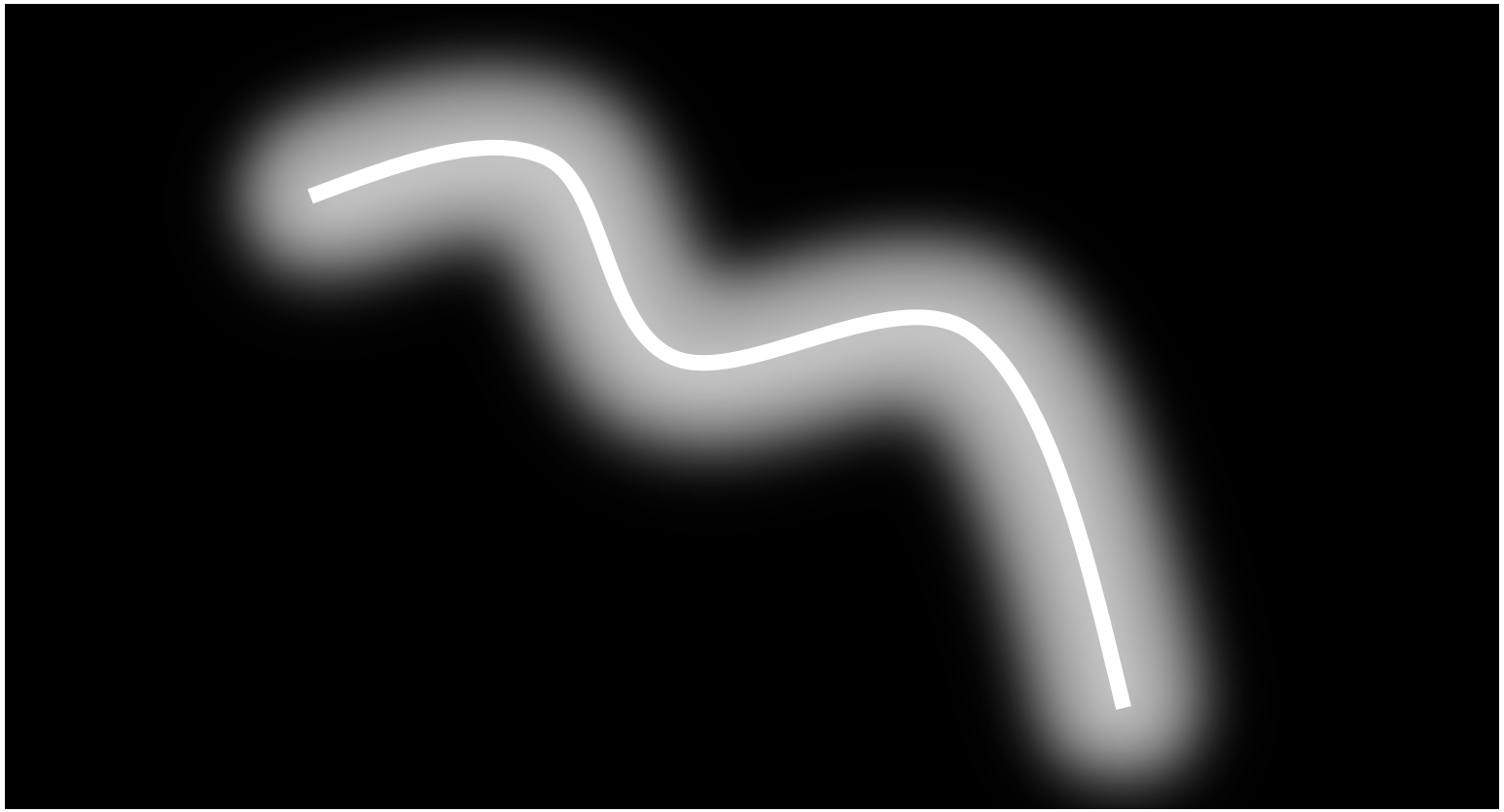
Match Colors: $\left|\left|\boldsymbol{S_{L(p)}(p) - S_{L(q)}(p)}\right|\right| + \left|\left|\boldsymbol{S_{L(p)}(q) - S_{L(q)}(p)}\right|\right|$

Match Gradients: $\left\|\boldsymbol{\nabla S_{L(p)}(p) - \nabla S_{L(q)}(p)}\right\| + \left\|\boldsymbol{\nabla S_{L(p)}(q) - \nabla S_{L(q)}(q)}\right\|$

Match Colors & Gradients: Match Gradients + Match Colors

Match Colors & Edges: Match Colors / Scalar Edge Potential

Spatial Constaint: Inertia



Gradient Domain Fushion

Rational:

There are artifacts in boundries even graph cuts cannot resolve

Gradient Domain Fushion

Rational:

There are artifacts in boundaries even graph cuts cannot resolve

Solution (Neumann Boundaries):

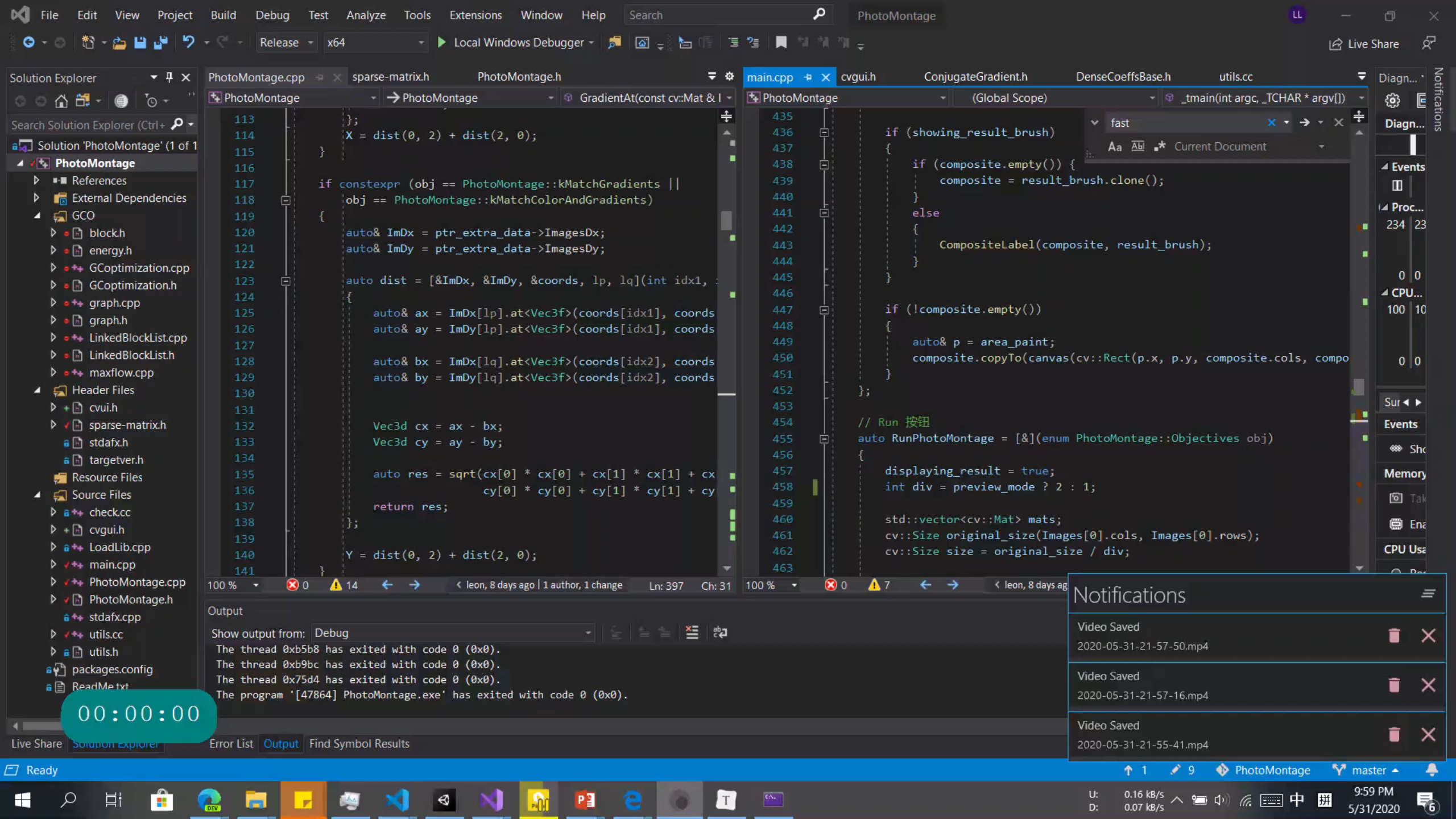
$$v_{x+1,y} - v_{x,y} = \nabla I_x(x,y)$$
$$v_{x,y+1} - v_{x,y} = \nabla I_y(x,y)$$

Gradient Domain Fushion

Solution (Neumann Boundries):

$$v_{x+1,y} - v_{x,y} = \nabla I_x(x,y)$$
$$v_{x,y+1} - v_{x,y} = \nabla I_y(x,y)$$

$$\begin{bmatrix} -1 & 1 & 0 & \cdots & \cdots \\ -1 & 0 & 1 & & \\ & \vdots & & \ddots & \vdots \\ & \cdots & & \cdots & \end{bmatrix} \begin{matrix} I(0,1) \\ I(0,2) \\ \vdots \\ I(1,1) \\ I(1,2) \\ \vdots \\ I(n_x, \mathrm{n}_y) \end{matrix} = \begin{matrix} Ix(0,1) \\ Iy(0,1) \\ \vdots \end{matrix}$$


File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search
PhotoMontage
Release
x64
Local Windows Debugger
Live Share
Solution Explorer
Search Solution Explorer (Ctrl+
Solution 'PhotoMontage' (1 of 1
PhotoMontage
References
External Dependencies
GCO
block.h
energy.h
GCoptimization.cpp
GCoptimization.h
graph.cpp
graph.h
LinkedBlockList.cpp
LinkedBlockList.h
maxflow.cpp
Header Files
cvui.h
sparse-matrix.h
stdafx.h
targetver.h
Resource Files
Source Files
check.cc
cvgui.h
LoadLib.cpp
main.cpp
PhotoMontage.cpp
PhotoMontage.h
stdafx.cpp
utils.cc
utils.h
packages.config
ReadMe.txt
PhotoMontage.cpp
sparse-matrix.h
PhotoMontage.h
PhotoMontage
PhotoMontage
GradientAt(const cv::Mat & I
113 };
114 X = dist(0, 2) + dist(2, 0);
115 }
116
117 if constexpr (obj == PhotoMontage::kMatchGradients ||
118 obj == PhotoMontage::kMatchColorAndGradients)
119 {
120 auto& ImDx = ptr_extra_data->ImagesDx;
121 auto& ImDy = ptr_extra_data->ImagesDy;
122
123 auto dist = [&ImDx, &ImDy, &coords, lp, lq](int idx1,
124 {
125 auto& ax = ImDx[lp].at<Vec3f>(coords[idx1], coords
126 auto& ay = ImDy[lp].at<Vec3f>(coords[idx1], coords
127
128 auto& bx = ImDx[lq].at<Vec3f>(coords[idx2], coords
129 auto& by = ImDy[lq].at<Vec3f>(coords[idx2], coords
130
131
132 Vec3d cx = ax - bx;
133 Vec3d cy = ay - by;
134
135 auto res = sqrt(cx[0] * cx[0] + cx[1] * cx[1] + cx
136 cy[0] * cy[0] + cy[1] * cy[1] + cy
137 return res;
138 };
139
140 Y = dist(0, 2) + dist(2, 0);
141 }
100 %
0
14
leon, 8 days ago | 1 author, 1 change
Ln: 397
Ch: 31
main.cpp
cvgui.h
ConjugateGradient.h
DenseCoeffsBase.h
utils.cc
PhotoMontage
(Global Scope)
_tmain(int argc, _TCHAR * argv[])
435
436 if (showing_result_brush)
437 {
438 if (composite.empty()) {
439 composite = result_brush.clone();
440 }
441 else
442 {
443 CompositeLabel(composite, result_brush);
444 }
445 }
446
447 if (!composite.empty())
448 {
449 auto& p = area_paint;
450 composite.copyTo(canvas(cv::Rect(p.x, p.y, composite.cols, compo
451 }
452 };
453
454 // Run 按钮
455 auto RunPhotoMontage = [&](enum PhotoMontage::Objectives obj)
456 {
457 displaying_result = true;
458 int div = preview_mode ? 2 : 1;
459
460 std::vector<cv::Mat> mats;
461 cv::Size original_size(Images[0].cols, Images[0].rows);
462 cv::Size size = original_size / div;
463
fast
Aa
Current Document
100 %
0
7
leon, 8 days ag
Diagn...
Events
Proc...
234
0
CPU...
100
0
Events
Memory
CPU Usa
Output
Show output from: Debug
The thread 0xb5b8 has exited with code 0 (0x0).
The thread 0xb9bc has exited with code 0 (0x0).
The thread 0x75d4 has exited with code 0 (0x0).
The program '[47864] PhotoMontage.exe' has exited with code 0 (0x0).
00:00:00
Live Share
Solution Explorer
Error List
Output
Find Symbol Results
Notifications
Video Saved
2020-05-31-21-57-50.mp4
Video Saved
2020-05-31-21-57-16.mp4
Video Saved
2020-05-31-21-55-41.mp4
Ready
1
9
PhotoMontage
master
U: 0.16 kB/s
D: 0.07 kB/s
9:59 PM
5/31/2020

Result & Analysis

Match Gradients

Result & Analysis

Match Gradients

Match Colors

Result & Analysis

Match Gradients

Match Colors

Result & Analysis

Match Gradients

Match Colors

Result & Analysis

Gradients are better than colors:

- Robust to illumation changes
- Better capturing locality
- Better maintaining shapes of objects

Result & Analysis

Match Gradients

Match Colors

Result & Analysis

1000 Iterations

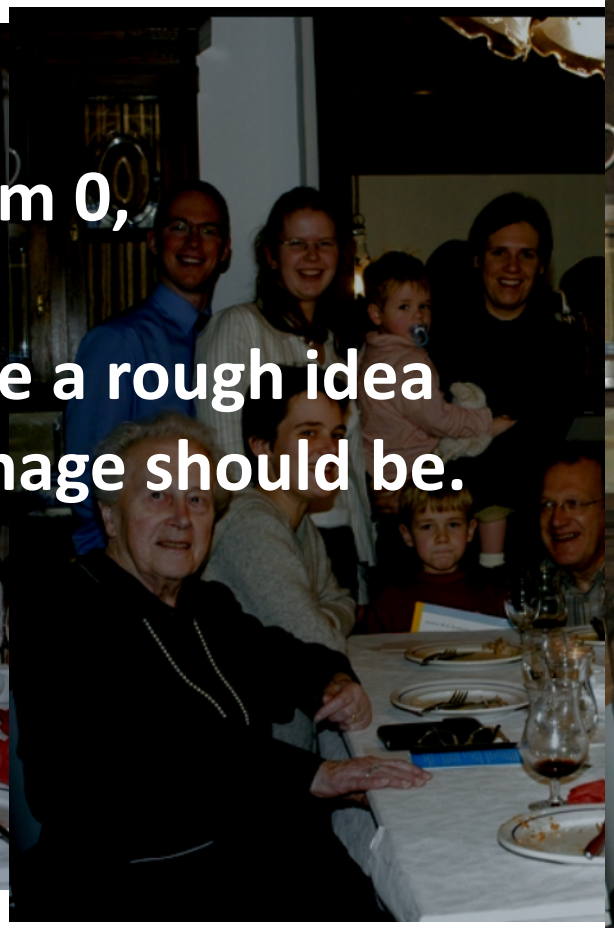
2000 Iterations

3000 Iterations

4000 Iterations

Result & Analysis

1000 Iterations



2000 Iterations

3000 Iterations

4000 Iterations

Result & Analysis

3000 Iterations, Old methods

50 Iterations, New methods

Result & Analysis

Match	Times (Iterations) Old Method	Times (Iterations) New Method
Match Colors	44738ms (3000)	3143ms (50)
Match Gradients	55201ms (3000)	15111ms (50)
Match Colors & Edges	45186ms (3000)	3349ms (50)

Discussions & Future Works

- This technique be futher applied to incremental changes in response to user input.
- Segmentation is now the bottle neck of performance.

Questions 

Thank you

For your attention