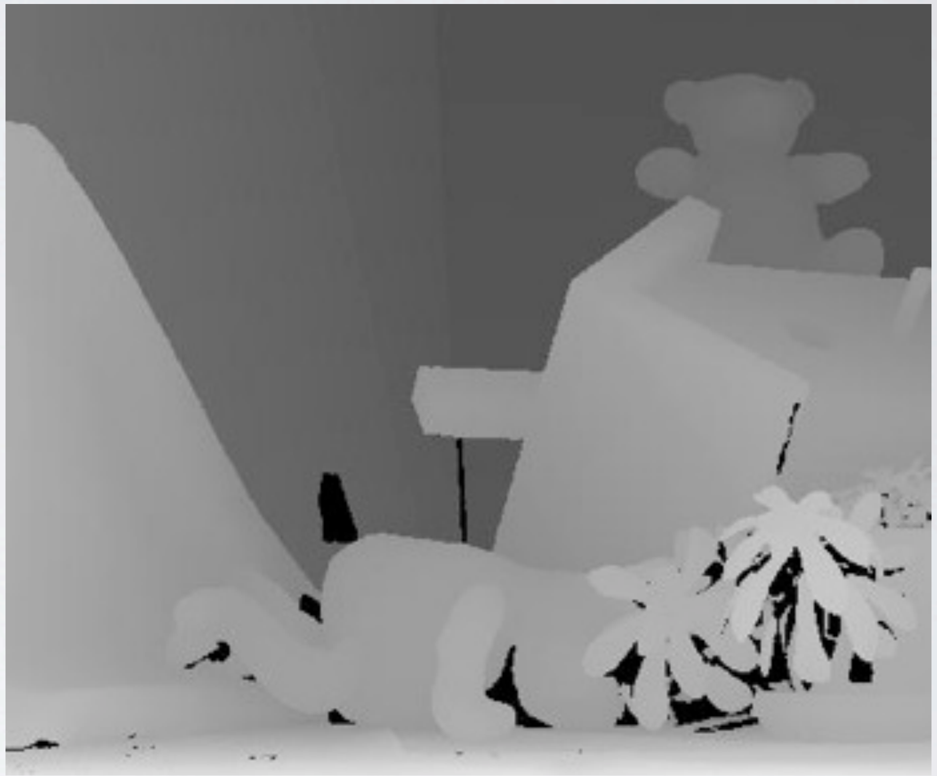
ASSIGNMENT 3

Stereo Correspondence

Nov. 6, 2013

STEREO CORRESPONDENCE



STEPS FOR STEREO CORRESPONDENCE

- Camera calibration
- Dense pixel correspondence (epipolar constraint)
  - Image rectification
  - Disparity estimation
- Depth estimation

STEPS FOR STEREO CORRESPONDENCE

- ~~Camera calibration~~ Assume calibrated camera
- Dense pixel correspondence (epipolar constraint)
    - Image rectification
    - Disparity estimation
- Depth estimation

STEPS FOR STEREO CORRESPONDENCE

- ~~Camera calibration~~
- Dense pixel correspondence (epipolar constraint)
  - ~~Image rectification~~ Assume rectified images
  - Disparity estimation
- Depth estimation

STEPS FOR STEREO CORRESPONDENCE

- ~~Camera calibration~~
- Dense pixel correspondence (epipolar constraint)
    - ~~Image rectification~~
    - Disparity estimation
- ~~Depth estimation~~ Trivial to obtain from disparity

STEPS FOR STEREO CORRESPONDENCE

- ~~Camera calibration~~
- Dense pixel correspondence (epipolar constraint)
  - ~~Image rectification~~
  - Disparity estimation
- ~~Depth estimation~~

DISPARITY ESTIMATION

- Estimate the optimal disparity assignment for each pixel by minimizing the following energy function

$$E(y,x,d) = \sum_{x,y}^{\text{Pixels}} data(y,x,d(y,x)) + \sum_{x,y,nx,ny}^{\text{Pixel neighbors}} smoothness(d(y,x), d(ny,nx))$$

where:

data(y, x, d) = cost of assigning disparity d at pixel (y, x)

smoothness(d1, d2) = cost of assigning disparities d1 and d2 at neighboring pixels

STEPS FOR DISPARITY ESTIMATION

$$E(y,x,d) = \sum_{x,y}^{\text{Pixels}} data(y,x,d(y,x)) + \sum_{x,y,nx,ny}^{\text{Pixel neighbors}} smoothness(d(y,x),d(ny,nx))$$

STEPS FOR DISPARITY ESTIMATION

$$E(y,x,d) = \overbrace{\sum_{x,y}}^{\text{Pixels}} data(y,x,d(y,x)) + \overbrace{\sum_{x,y,nx,ny}}^{\text{Pixel neighbors}} smoothness(d(y,x),d(ny,nx))$$

compute the data term

STEPS FOR DISPARITY ESTIMATION

$$E(y,x,d) = \overbrace{\sum_{x,y}^{\text{Pixels}} data(y,x,d(y,x))} + \overbrace{\sum_{x,y,nx,ny}^{\text{Pixel neighbors}} smoothness(d(y,x), d(ny,nx))}$$

compute the data term

compute the smoothness term

STEPS FOR DISPARITY ESTIMATION

$$E(y,x,d) = \overbrace{\sum_{x,y}}^{\text{Pixels}} data(y,x,d(y,x)) + \overbrace{\sum_{x,y,nx,ny}}^{\text{Pixel neighbors}} smoothness(d(y,x),d(ny,nx))$$

compute the data term

compute the smoothness term

optimize the energy function

ALGORITHMS TO IMPLEMENT

<table>
<tr><th>compute data term</th><th>compute smoothness term</th><th>optimize energy</th></tr>
<tr><td rowspan="2">L1</td><td rowspan="2">L1</td><td>dynamic programming</td></tr>
<tr><td>graph cut</td></tr>
<tr><td>awesome</td><td>awesome</td><td>awesome</td></tr>
</table>

ALGORITHMS TO IMPLEMENT

Required

compute data term	compute smoothness term	optimize energy
L1	L1	dynamic programming graph cut
awesome	awesome	awesome

ALGORITHMS TO IMPLEMENT

compute data term	compute smoothness term	optimize energy
L1	L1	dynamic programming
		graph cut
awesome	awesome	awesome

Required

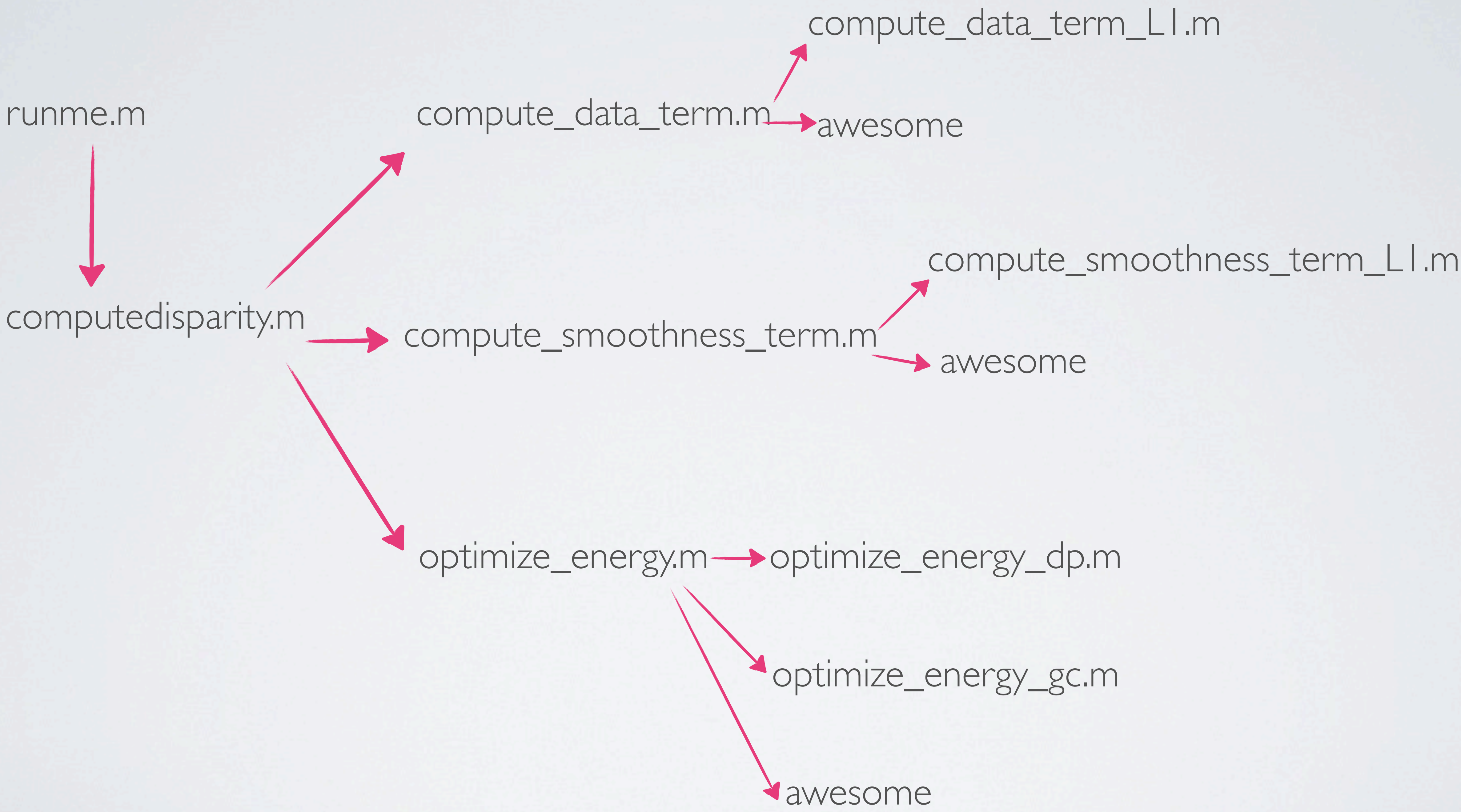
Implement at least one awesome

HOW TO RUN

- runme.m

```
run_configurations = { ...
  % {'test01', 'L1', 'L1', 'graph_cut'}, ...
  % {'test01', 'awesome', 'L1', 'dynamic_programming'}, ...
  % {'test01', 'L1', 'awesome', 'dynamic_programming'}, ...
  % {'test01', 'L1', 'L1', 'awesome'}, ...
  {'all', 'L1', 'L1', 'dynamic_programming'}, ...
  {'all', 'L1', 'L1', 'graph_cut'}
  };
```

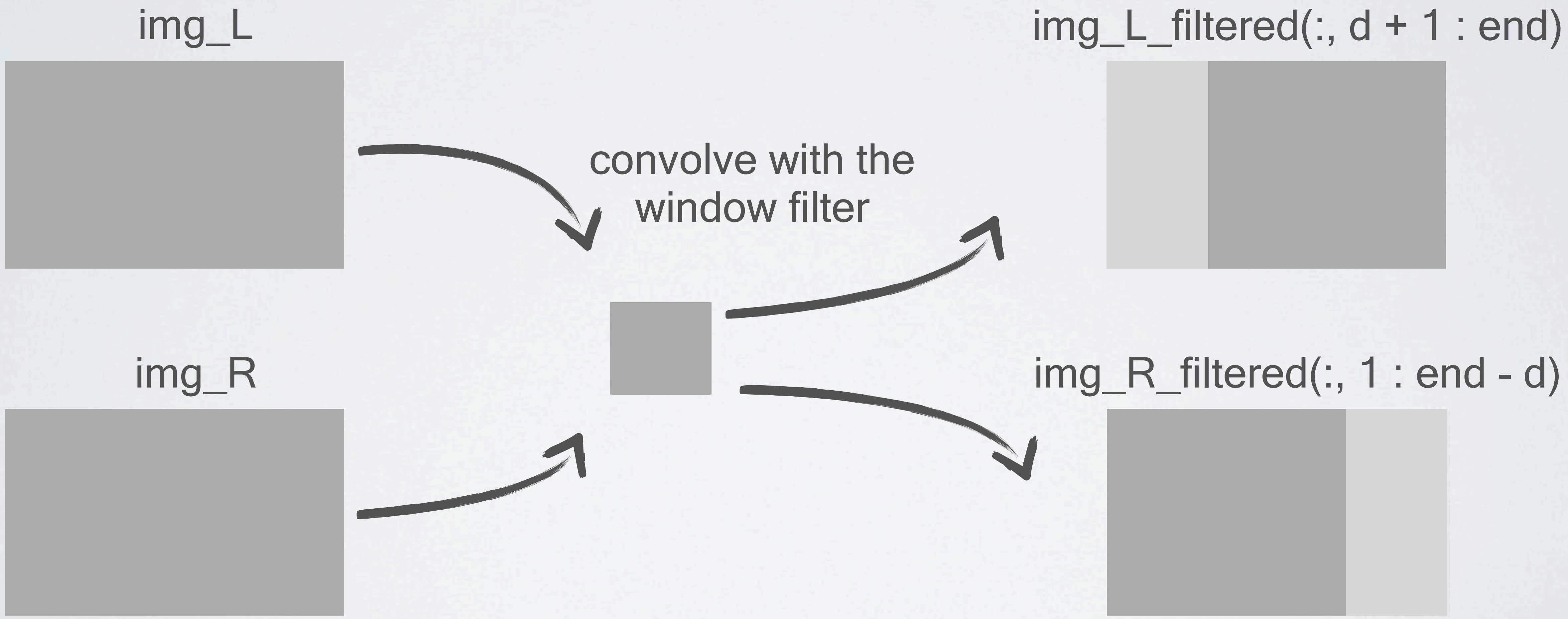
HOW TO RUN


runme.m
computedisparity.m
compute_data_term.m
compute_data_term_L1.m
awesome
compute_smoothness_term.m
compute_smoothness_term_L1.m
awesome
optimize_energy.m
optimize_energy_dp.m
optimize_energy_gc.m
awesome

TIPS AND SUGGESTIONS

COMPUTE DATA TERM

$$data(y,x,d) = \lambda \cdot \min\left(\frac{\sum_{(y,x)\in\text{window}} |I_L(y,x) - I_R(y,x-d)|}{\text{window size}}, \tau\right)$$



PARAMETER SETTING

- maximum_data_term_value: ~10
- maximum_smoothness_term_value: ~1.7
- max_disparity = ~60
- lambda/data_term_weight: ~0.04

GRAPH CUT

- GCMex
  - link: http://vision.ucla.edu/~brian/gcmex.html
- [labels, energy, energyafter] = GCMex(class, unary, pairwise, labelcost, flag);
- You might want to use diag() in MATLAB to create the adjacency matrix

START EARLY & GOOD LUCK!