Exercise 4. Sampling Patterns and Graph Cuts

Dongho Kang

16-948-598

May 7, 2017

MATLAB R2016b version was used for coding and testing:

MathWorks, MATLAB R2016b (9.1.0.441655)
64-bit (maci64)

The ***code*** directory contains the followings:

part1_1.m script .m file for exercise part 1.1.

part1_2.m script .m file for exercise part 1.2.

part2_1.m script .m file for exercise part 2.1.

part2_2.m script .m file for exercise part 2.2.

PART I provided directory for part 1.

PART II directory which contains implementation of part 2 and provided files including skeleton code etc.

img directory which contains images for testing part 2.2

result result image of part 1 and part 2.

For running each .m script, check dependencies (especially for ***part2_1.m*** and ***part2_2.m***) and adjust parameters first. Note that **these scripts only work properly in MATLAB R2016b environment** and **have done in Mac OS 10.11.6.** More details are stated in the *Running* section of each parts.

1 EXERCISE PART 1: ANALYZING SAMPLING PATTERNS

In this exercise, two analysis techniques for sampling distributions was implemented:

1. Periodogram (task 1)
2. Pair Correlation Functions (task 2)

1.1 TASK 1: COMPUTING PERIODOGRAMS OF SAMPLING PATTERNS

1.1.1 DESCRIPTION

The periodogram is computed by **taking the Fourier transform of the impulse process corresponding to the sampling pattern.** It can be estimated by follows:

$$P(w) = \left| \mathcal{F}[\frac{1}{n}\sum_{i=1}^{n}\delta(\mathbf{x}-\mathbf{x_i})] \right|^2 \tag{1.1}$$

where δ is the Dirac delta function, $\mathbf{x_i}$ are the locations of the points in a given points distribution and $\mathcal{F}$ denotes the Fourier transform. As suggested, this was implemented by rasterizing the function $\frac{1}{n}\sum_{i=1}^{n}\delta(\mathbf{x}-\mathbf{x_i})$ and taking the discrete Fourier transform by using MATLAB function **fft2**.

The periodogram was generated for the 4 different sampling algorithms, *Matern, FPO, Dart* and *Balzer*. For each algorithm, results from 10 different dataset were averaged.

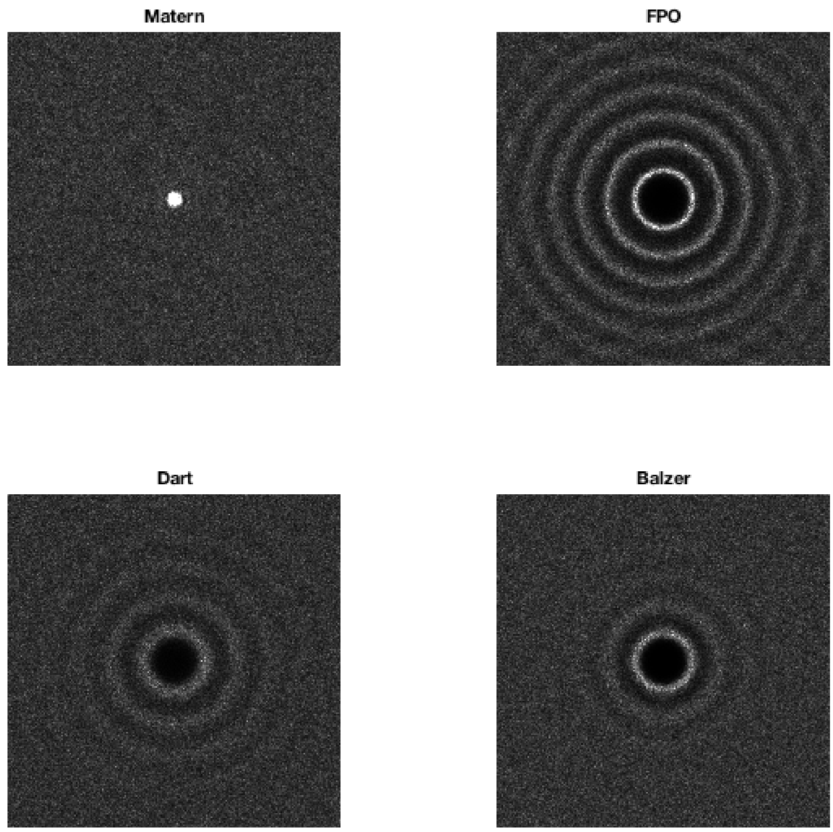
1.1.2 RUNNING

Run the script ***part1_1.m*** after setting the parameter *width, height* for width and height of initial image. The default values are 400 for both *width* and *height*.

1.1.3 RESULT

The Figure 1.1 is averaged periodograms of different sampling algorithms. In order to get a clear result image, the periodogram was scaled by 200.

Figure 1.1: Averaged periodograms for the Matern, FPO, Dart and Balzer algorithms



1.2 Task 2: Computing the pair correlation function of sampling patterns

1.2.1 Description

Another way to analyze point distribution is via point process statistics. The pair correlation function (PCF) is widely accepted as the most informative. This measure $g(\mathbf{x}, \mathbf{y})$ describes the joint probability of having points at locations $\mathbf{x}$, and $\mathbf{y}$ at the same time.

In the isotropic case, PCF is only depends on the distance between the points and can be estimated as follows:

$$\hat{g}(r) = \frac{|V|}{\left|\partial V_d\right| r^{d-1} n^2} \sum_{i \neq j} k_\sigma(r - d(\mathbf{x}_i, \mathbf{x}_j)) \tag{1.2}$$

Here n is the number of samples and $\left|\partial V_d\right|$ denotes the volume of the boundary of a unit sphere in a d dimensional domain. Since it's 2-dimensional case, $d = 2$, $|V| = 1$, $\left|\partial V_d\right| = 2\pi$ and $d(\mathbf{x}_i, \mathbf{x}_j)$ is euclidean distance. The Gaussian kernel was used for $k_\sigma(x) = \frac{1}{\sqrt{\pi}\sigma} e^{-x^2/\sigma^2}$. For σ, as suggested in the manual, $\sigma = 0.25$ was used.

The most tricky part is choosing r_a, r_b and normalizing the data. r_a and r_b are the lower and upper limit of the r values. In order to define these parameters in relative terms, sample points should be normalized by the distance r_{max} defined as the minimum distance between pairs of points for the maximum packing of points in a given volume.[1] Two methods can be adapted for determining the r_{max} value as follows:

- the method by Lagae and Dutré [2]:

$$r_{max} = \sqrt{\frac{1}{2\sqrt{3}N}} \tag{1.3}$$

  where N is the number of samples. For $N = 1024$, $r_{max} = 0.0168$.

- the method by Gamito and Maddock [3]:

$$r_{max} = \sqrt[n]{\frac{\gamma_{n_{max}}}{N} \frac{\Gamma(\frac{n}{2}+1)}{\pi^{n/2}}} \tag{1.4}$$

  where N is the number of samples, Γ denotes Gamma function, $n = 2$, and $\gamma_{n_{max}} = \frac{1}{6}\pi\sqrt{3}$ for 2-dimensional case. In fact, since $\Gamma(2) = 1$ for $n = 2$, r_{max} is exactly same as r_{max} calculated by Lagae and Dutré method.

As normalizing the samples by dividing by r_{max}, $|V|$ also should be divided by r_{max}^2. For the best value of r_a and r_b, $r_a = 0.01\sigma$, $r_b = 5$ were suggested but here $r_a = 2\sigma$, $r_b = 10$ were used because of the numerical issue.

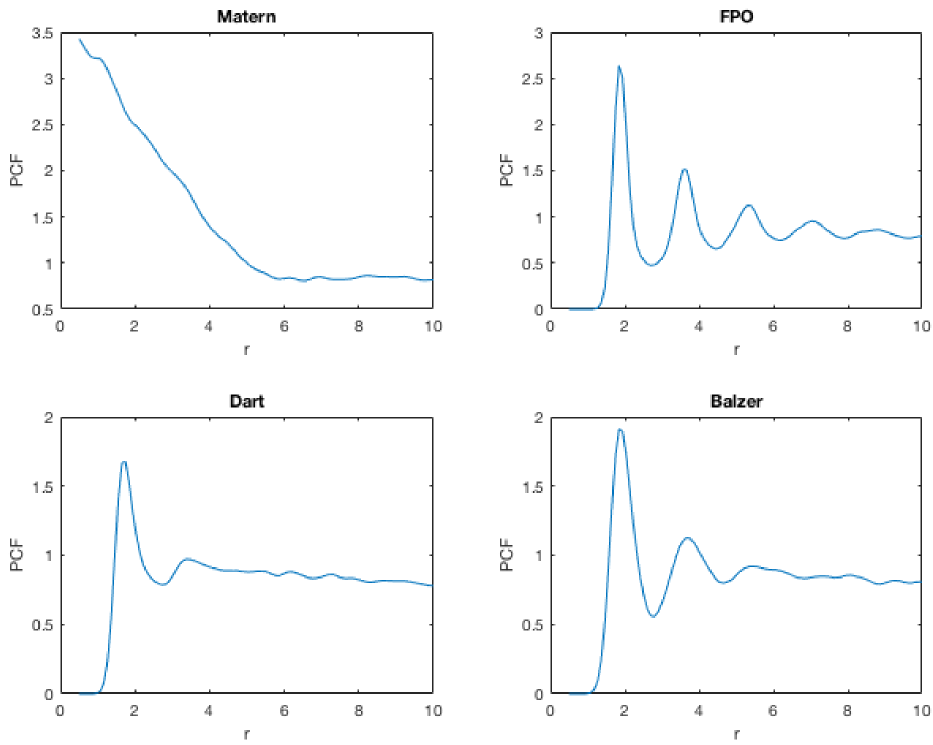
1.2.2 Running

Run the script ***part1_2.m*** after setting the parameter *array_size* for the number of r values. Since every parameter was carefully chosen, do not change any value except *array_size*.

1.2.3 Result

The Figure 1.2 is PCF of different sampling algorithms. The first datasets (***<algorithm>/1.txt***) for each algorithm were used as sample data.

Figure 1.2: PCF for the Matern, FPO, Dart and Balzer algorithms



1.3 Discussion

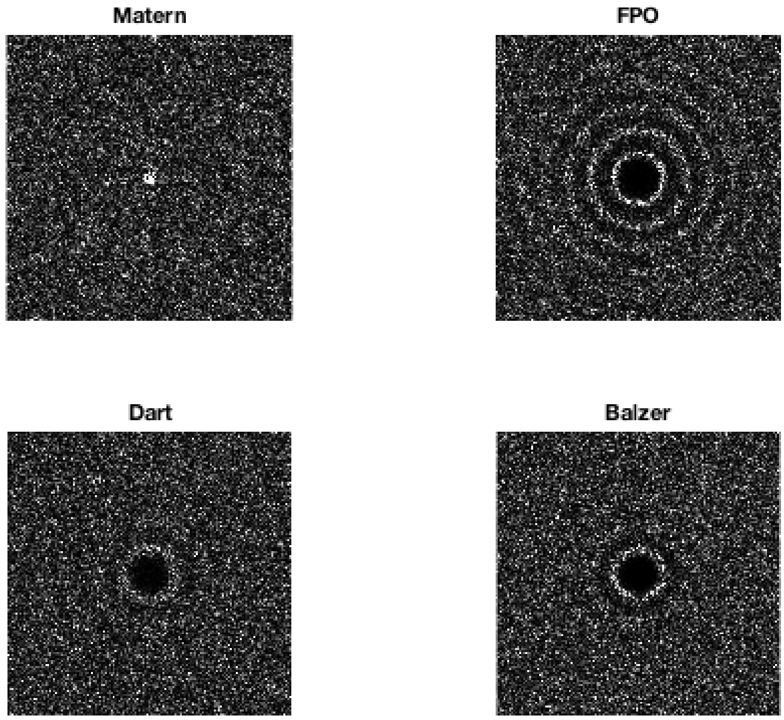
- Why we had to average over multiple point sets for each algorithm when computing the periodograms but not when computing the PCF's?

  Referring to [1], unlike periodograms, **the PCF interpretes directly linked to the distribution of the distances between pairs of points** i.e. the PCF, itself is statistical analysis.

  Looking close into equation (1.2), the PCF includes the kernel $k_\sigma(r - d(\mathbf{x}_i, \mathbf{x}_j))$ term thus, it's already smoothing (or averaging) the distribution of given sample points. That is why the PCF is called **smoothed histogram**.

  Besides, the periodogram without averaging is not a statistical analysis. Figure 1.3 is the periodogram which created with one dataset points. Since itself is not statistical, it's hard to find a pattern of each algorithm. Thus in order to find a pattern of certain sampling algorithm, periodogram should be averaged over a number of datasets.

Figure 1.3: The given graphical model for task 1.



- Is the PCF are sufficient to describe the provided point patterns as they are only one dimensional, while the periodograms are two dimensional? Do the periodograms contain more information for the provided patters?

  The [1] only considered stationary and isotropic case i.e. translation and rotation invariant. Under the assumption, **the joint probability** $g(\mathbf{x},\mathbf{y})$ **of having points at location x and y at the same time only depends on the distance between the points.** Thus, though the PCF is only one dimensional, it can describe the point patterns.

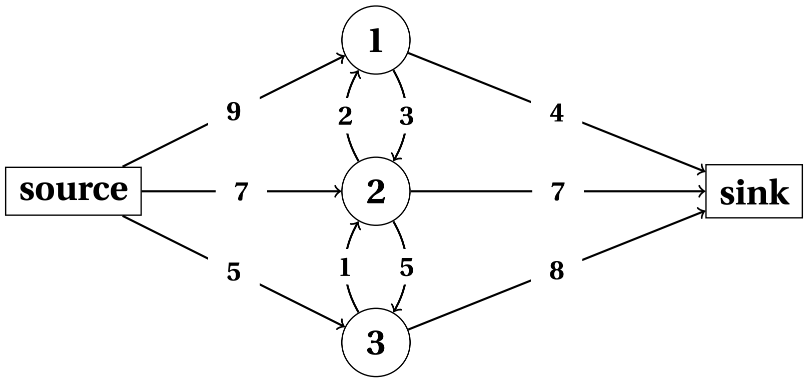
  However, the PCF is spatial domain analysis while the periodogram is frequency domain analysis. **Therefore periodogram has more information about certain patterns' properties in frequency domain.**

2 EXERCISE PART 2: INTERACTIVE SEGMENTATION WITH GRAPH CUT

2.1 TASK 1: HANDLING MAX FLOW

2.1.1 DESCRIPTION

Figure 2.1: The given graphical model for task 1.



The max flow problem for Figure 2.1 was solved by the algorithm by Boykov and Kolmogorov (BK algorithm). It was implemented using *BK library*. The comparison between the result by the algorithm and the result by hand is discussed in the section 2.1.4.

For unary costs (edges to source and sink) and pairwise costs (edges between nodes), following values were set as input:

Table 2.1: Unary costs for assigning label by BK algorithm

label	node 1	node 2	node 3
source (1)	4	7	8
sink (2)	9	7	5

The cost for assigning source(label 1) to node is defined as the cost of cutting the edge between the node and the sink terminal. For sink(label 2), it's the cost of cutting the edge between the node and the source terminal.

Table 2.2: Pairwise costs between nodes. From row to column.

	node 1	node 2	node 3
node 1	.	3	.
node 2	2	.	5
node 3	.	1	.

2.1.2 RUNNING

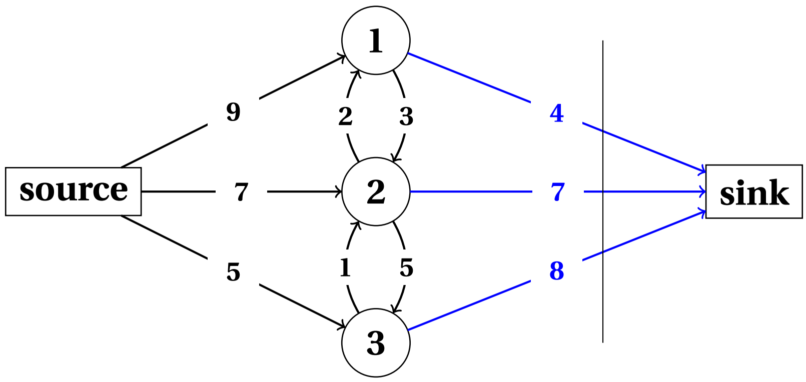
Since third party library *BK library* was used, check if the library was built properly. All library files including ***bin*** directory which contains the compiled binaries, should be placed in the ***PART II/GraphCut*** directory.

Run the script ***part2_1.m***. The script invoke subscript ***PART II/task1.m***, the implementation of computing max flow of Figure 2.1.

2.1.3 RESULT

The result of the max flow by BK algorithm as follows:

Figure 2.2: The given graphical model for task 1.

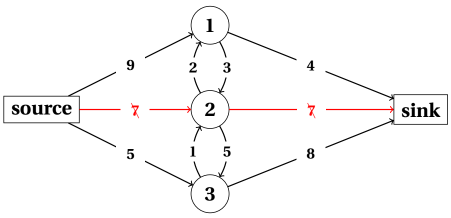


- max flow (or min cut energy) = 19
- label 1 was assigned to all nodes i.e. nodes are connected with source.

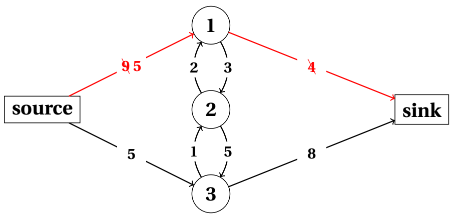
2.1.4 Discussion

Steps for computing max flow by hand is as follows:

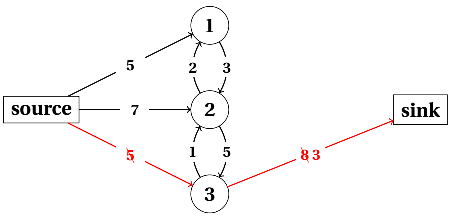
Figure 2.3: The given graphical model for task 1.



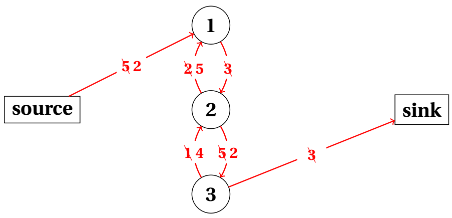
(a) flow = 7



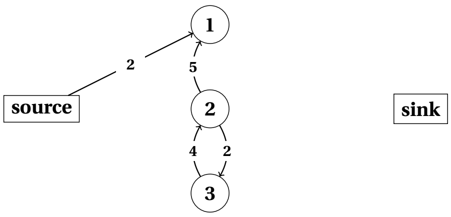
(b) flow = 11



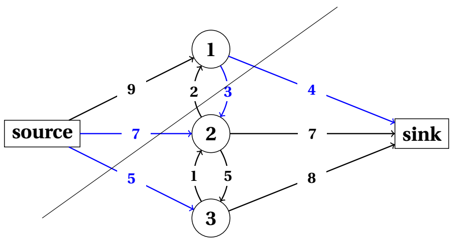
(c) flow = 16



(d) flow = 19



(e) terminated with flow = 19



(f) graph cut with cost = 19

- max flow (or min cut energy) = 19
- label 1 was assigned to node 1. Thus, node 1 is connected with source.
- label 2 was assigned to node 2 and node 3. Thus, node 2 and node 3 are connected with sink.

The result above is same in max flow with the result by BK algorithm but different in labelling. Both labelling is correct because costs of cutting are 19 in both cases.

2.2 Task 2: Interactive Segmentation

2.2.1 Description

An interactive segmentation algorithm was implemented. The steps are as follows:

1. Build a color histogram.
2. Get unary cost.
3. Get pairwise cost.
4. Build and solve a graph. Produce segmentation images.
5. Change background using the obtained segmentation.

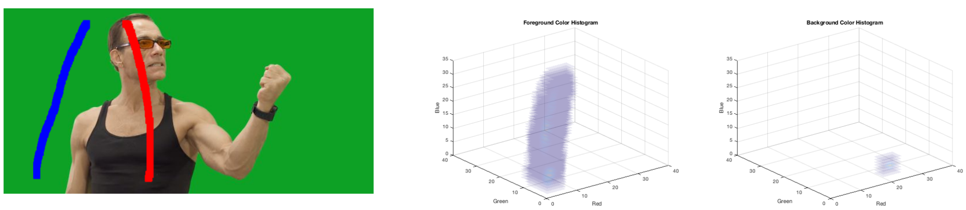
The unary and pairwise costs was defined as the definition on [4].

Color histogram A color histogram of dimension 32 × 32 × 32 was built by a given data which user assigned as foreground or background. Each pixel of image goes to a bin which corresponds to its R, G, B values. For proper generalization, resolution for R, G, B set to 32. Thus, index of bin for a pixel with $I = (r, g, b)$ determined as follows:

$$\begin{aligned} i_1 &= Q(r, 8) + 1 \\ i_2 &= Q(g, 8) + 1 \\ i_3 &= Q(b, 8) + 1 \end{aligned} \tag{2.1}$$

where $Q(a, b)$ is quotient of dividend a and divisor b and $\mathbf{i} = (i_1, i_2, i_3)$ is a index of a histogram bin. In order to using the color histogram as probability distribution, it was smoothed and normalized.

Figure 2.5: Color histogram of JCVD.jpg



(a) JCVD.jpg (b) foreground (c) background

UNARY COST Using color histogram, unary cost for each pixel was calculated:

Table 2.3: Definition of unary cost

edge	label	weight	for
$\{p,S\}$	$\{p,T\}$	$\lambda \cdot R_p(\text{"bkg"})$	$p \in \mathcal{P}, p \notin \mathcal{O} \cup \mathcal{B}$
		K	$p \in \mathcal{O}$
		0	$p \in \mathcal{B}$
$\{p,T\}$	$\{p,S\}$	$\lambda \cdot R_p(\text{"obj"})$	$p \in \mathcal{P}, p \notin \mathcal{O} \cup \mathcal{B}$
		0	$p \in \mathcal{O}$
		K	$p \in \mathcal{B}$

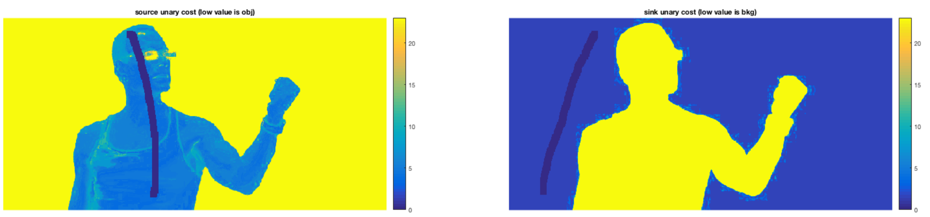
$\mathcal{P}$ is a set of pixels. The subsets $\mathcal{O} \subset \mathcal{P}$ and $\mathcal{O} \subset \mathcal{P}$ denote the subsets of pixels marked as "object" and "background" and $\mathcal{O} \cap \mathcal{O} = \emptyset$.

Regional penalites $R_p(\cdot)$ is defined as negative log-likelihoods:

$$\begin{aligned} R_p(\text{"obj"}) &= -\ln \Pr(I_p|\mathcal{O}) \\ R_p(\text{"bkg"}) &= -\ln \Pr(I_p|\mathcal{B}) \end{aligned} \tag{2.2}$$

where $\Pr(I_p|\mathcal{O})$ and $\Pr(I_p|\mathcal{O})$ are intensity distributions from the color histogram. For K, $K = \infty$ was used. As defining unary cost in the above manner, Figure 2.7 was obtained. (a) is cost of assigning "obj" label to each pixel node (thus the cutting edge cost between node and sink) and (b) is cost of assigning "bkg" label to each pixel node. For each color map, blue indicates low cost and yellow indicates high cost.

Figure 2.7: The unary cost of JCVD.png ($\lambda = 1.0$)



(a) source unary cost (blue is obj)　　(b) sink unary cost (blue is bkg)

PAIRWISE COST Pairwise cost between a pixel and its neighboring 8 pixels was defined as follows:

Table 2.4: Definition of pairwise cost

edge	weight	for
$\{p,q\}$	$B_{\{p,q\}}$	$\{p,q\} \in \mathcal{N}$

where an *ad-hoc* function was used as the boundary penalty $B_{\{p,q\}}$:

$$B_{\{p,q\}} \propto exp\Big(-\frac{(I_p - I_q)^2}{2\sigma^2}\Big) \cdot \frac{1}{dist(p,q)} \tag{2.3}$$

Here, $\sigma = 5$ and the euclidean distance was used for $dist(p,q)$.

GRAPH AND SEGMENTATION Using BK library, label for each pixel and cost of graph cut was obtained. Figure 2.9 shows the result of graph cut with $lambda = 1.0$ which was applied to JCVD.png. Yellow pixels are assigned to "bkg" label and blues are assigned to "obj".

Figure 2.9: The label obtained by a graph cut. ($\lambda = 1.0$)

CHANGING BACKGROUND As image segmented into two sets, object("obj") set and background("bkg") set, background pixels can be replaced with a new background image.

Figure 2.10: Background change for JCVD.png ($\lambda = 1.0$)

2.2.2 Running

Since third party library *BK library* was used, check if the library was built properly. All library files including ***bin*** directory which contains the compiled binaries, should be placed in the ***PART II/GraphCut*** directory.

Run the script ***part2_2.m***. The script invoke subscript ***PART II/interactiveGraphCut.m*** which is for GUI windows. Follow the steps below:

1. Open an image file.
2. To change the background, open background image file. (can be skipped)
3. Indicate foreground and background by red/blue marker. Pre-generated scribbles can be loaded for ***JCVD.jpg*** and ***bat.jpg***.
4. Enter the value of λ.
5. Press the segment button.

2.2.3 Result

Image segmentation was tried with different values of the unary cost parameter λ.

Figure 2.11: Image segmentation of JCVD.jpg with different λ

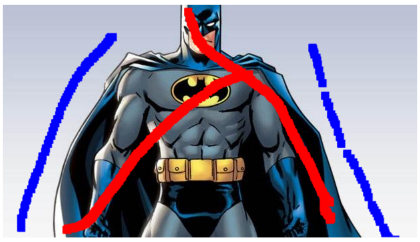
(a) JCVD.jpg

(b) $\lambda = 1.0$

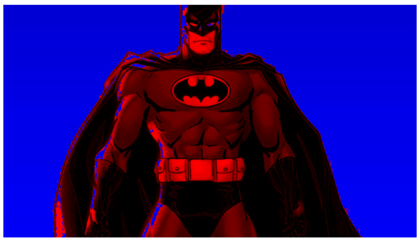
(c) $\lambda = 10^{-4}$

(d) $\lambda = 10^{-8}$

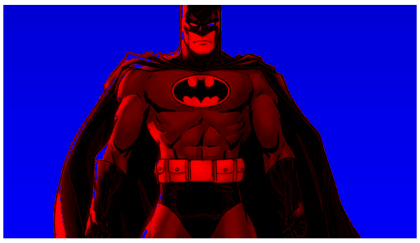
Figure 2.13: Image segmentation of JCVD.png with different λ



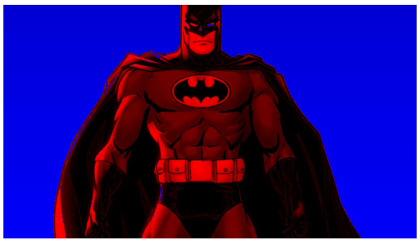
(a) batman.jpg



(b) $\lambda = 1.0$ (the best)



(c) $\lambda = 0.002$



(d) $\lambda = 10^{-6}$ (the best)

The result of changing background is as follows:

Figure 2.15: Background change with $\lambda = 1$

(a) Van Damme at ETH

(b) Dark knight at ETH

Finally, image segmentation was applied to an image which was carefully chosen:

Figure 2.17: Image segmentation of JCVD.png with different λ

(a) upton.jpg (b) upton.png ($\lambda = 1$) (c) upton at ETH

2.2.4 Discussion

- λ determines influence of color histogram. As λ increases, color histogram affects to segmentation more. Besides, as λ decreases, image gradient relatively affects more.
  - For JCVD.jpg, as λ decreases, the large gradient between JCVD's shirt and his left arm leads to assigning his left arm to the background.
  - For batman.jpg, as λ increases, some parts which have similar color with the background belonged to his body was assigned to the background.
- The best value of λ depends on the original image, especially the color difference between background and foreground. Thus, λ should be tuned carefully to get a clear segmentation result.

References

[1] A. C. Öztireli and M. Gross, "Analysis and synthesis of point distributions based on pair correlation," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 170, 2012.

[2] A. Lagae and P. Dutré, "A comparison of methods for generating poisson disk distributions," in *Computer Graphics Forum*, vol. 27, pp. 114–129, Wiley Online Library, 2008.

[3] M. N. Gamito and S. C. Maddock, "Accurate multidimensional poisson-disk sampling," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 1, p. 8, 2009.

[4] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1, pp. 105–112, IEEE, 2001.