MATHEMATICAL FOUNDATIONS OF COMPUTER GRAPHICS AND VISION

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.

? TODO for testing part 2.2

result result image of part 1 and part 2.

> ***periodogram.png*** TODO
>
> ***pcf.png*** TODO

TOOD For running each .m script, check dependencies (especially for ***part1_4.m*** and ***part1_5.m***) and adjust parameters first. Note that **these scripts only work properly in MATLAB R2016b environment.** 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:

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

where δ is the Dirac delta function, $\mathbf{x_i}$ are the locations of the points in a given points distribution and $\mathscr{F}$ 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 algorithm, *Matern, FPO, Dart, Balzer* and for each algorithm, results from 10 different point set were averaged.

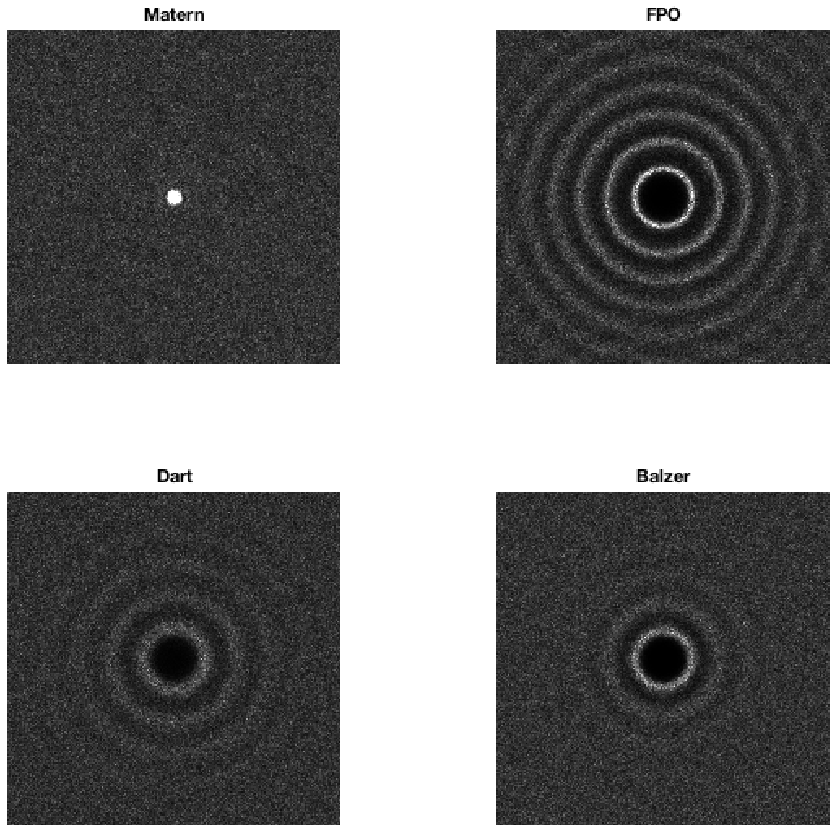
1.1.2 Running

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. 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 the parameters, 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. (TODO REFERENCE) Two methods can be adapted for determining the r_{max} value as follows:

- the method by Lagae and Dutré (TODO REFERENCE):

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

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

- the method by Gamito and Maddock (TODO REFERENCE):

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

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

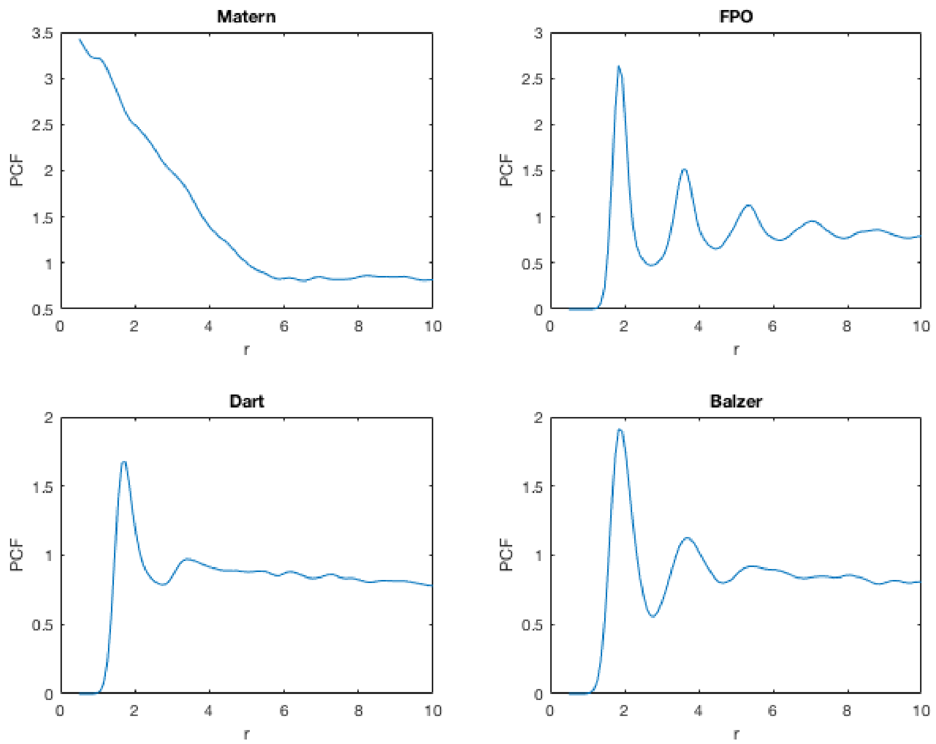
1.2.2 Running

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.

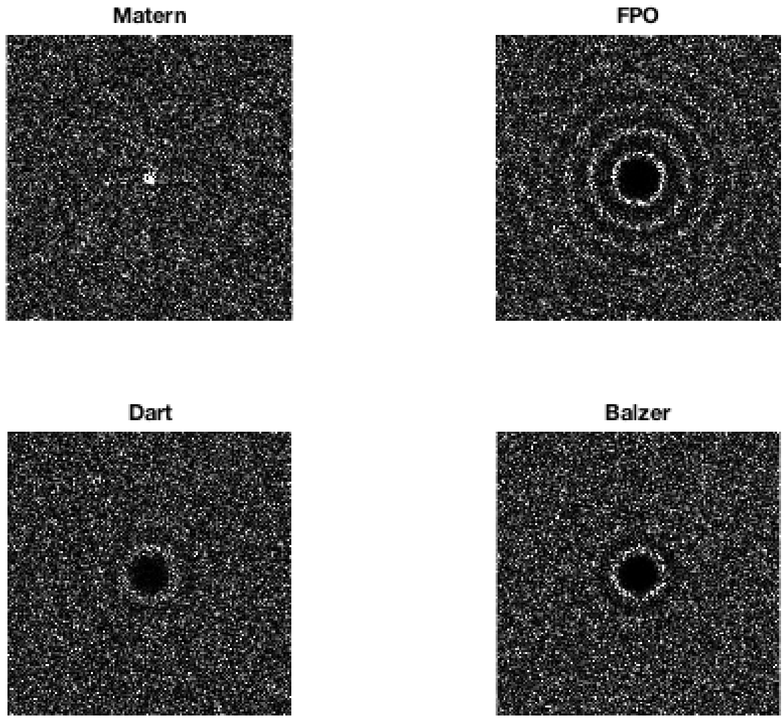
Figure 1.2: PCF for the Matern, FPO, Dart and Balzer algorithms



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?

Figure 1.3: The given graphical model for task 1.



TODO

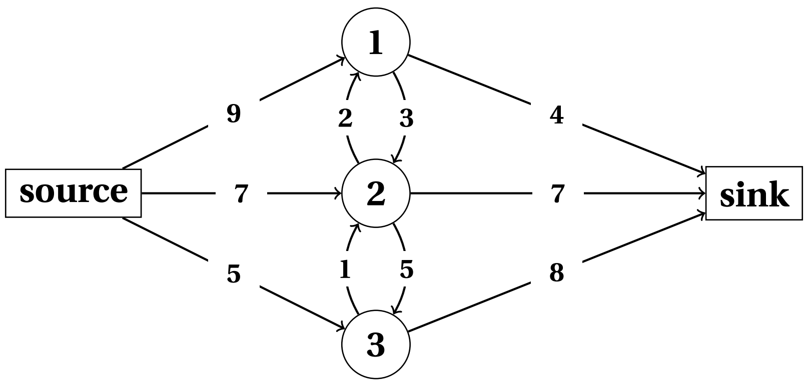
- Is the PCF are sufficient to describe the provided point patterns as they are only one dimensional, while the periodograms are two dimensional? Do the periodograms contain more information for the provided patters?

2 Exercise part 2: Image Deformation Using Moving Least Squares

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 by using *BK library*. The comparison between the result by the algorithm and 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.
(TODO TABLE - unary cost)
(TODO TABLE - pairwise cost)

2.1.2 Running

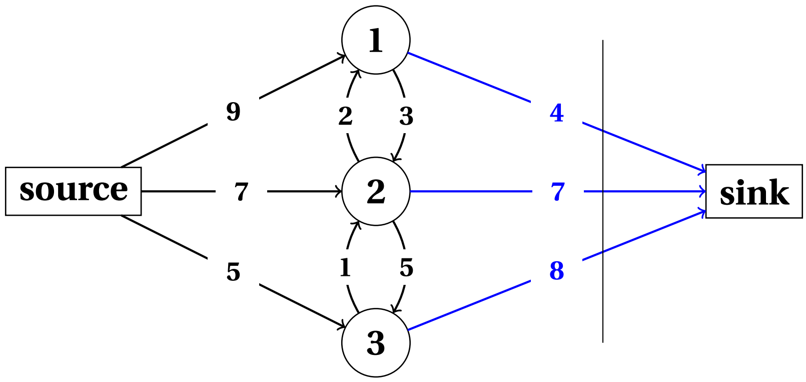
Since third party library *BK library* was used, check if the library was built properly. All library files including ***bin*** directory which contains the compiled binaries, should be placed in the ***PART II/GraphCut*** directory.

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.

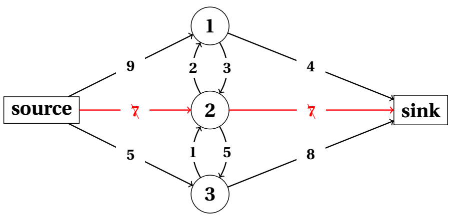


- max flow (or min cut energy) = 19
- label 1 was assigned to all nodes i.e. nodes are connected with source.

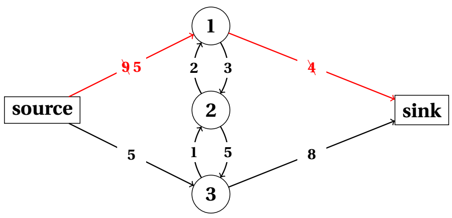
2.1.4 Discussion

Steps for computing max flow by hand is as follows:

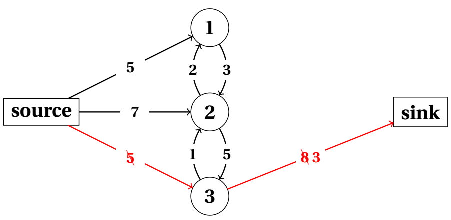
Figure 2.3: The given graphical model for task 1.



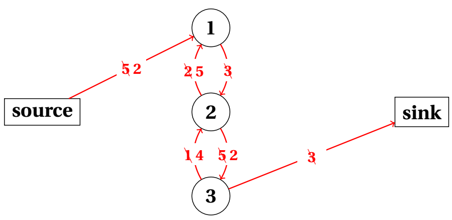
(a) flow = 7



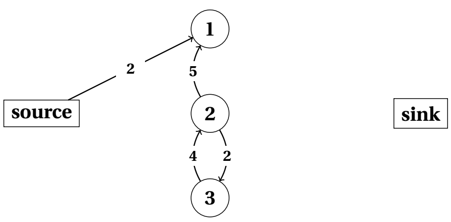
(b) flow = 11



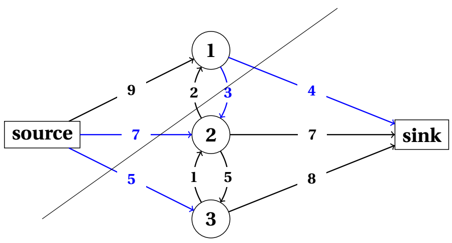
(c) flow = 16



(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 cost of cutting is 19 in both cases.

2.2 Task 2: Interactive Segmentation

2.2.1 Description

1. color histogram
2. getting unary cost
3. getting pairwise cost
4. building and solving a graph. segmentation
5. changing background using the obtained segmentation

Color histogram

Unary cost

Pairwise cost

Graph and segmentation

Changing background

2.2.2 Running

2.2.3 Result

2.2.4 Discussion