FINAL REPORT
FUNDAMENTAL RESEARCH GRANT SCHEME (FRGS)

Laporan Akhir Skim Geran Penyelidikan Fundamental (FRGS)
Pindaan 1/2015

A

RESEARCH TITLE: A Multi-Scale Approach for Estimating Gradient from Volumetric Data

PHASE & YEAR: 1/2013

START DATE : 01 May 2013
END DATE : 30 April 2015
EXTENSION PERIOD (DATE): RMC LEVEL: 30 October 2015
KPM LEVEL:

PROJECT LEADER : Bahari Belaton
I/C / PASSPORT NUMBER: 660823-08-6065

PROJECT MEMBERS: 1. Jamaludin Md. Ali (Retired end of 2015)
(including GRA) 2.

PROJECT ACHIEVEMENT *(Prestasi Projek)*

B

ACHIEVEMENT PERCENTAGE			
Project progress according to milestones achieved up to this period	**0 - 50%**	**51 - 75%**	**76 - 100%**
Percentage (please state #%)		55 - 65%	

RESEARCH OUTPUT		
Number of articles/ manuscripts/ books *(Please attach the First Page of Publication)*	**Indexed Journal**	***Non-Indexed Journal***
Conference Proceeding *(Please attach the First Page of Publication)*	**International**	***National***
Intellectual Property *(Please specify)*	2 Technical Reports	

HUMAN CAPITAL DEVELOPMENT

Human Capital	Number				Others (please specify)
	On-going		Graduated		
Citizen	Malaysian	Non Malaysian	Malaysian	Non Malaysian	Two GRA (on honorarium per given tasks basis) 1. Badr Mohammed Omar Lahasan (PN : 04632773) 2. Chen Junfen (PN: G37225006)
No. PHD STUDENT					
Student Fullname: IC / Passport No: Student ID:					
No. MASTER STUDENT					
Student Fullname: IC / Passport No: Student ID:					
No. UNDERGRADUATE STUDENT					
Student Fullname: IC / Passport No: Student ID:					
Total					

EXPENDITURE *(Perbelanjaan) as Borang K1(RMC)*

C

Budget Approved *(Peruntukan diluluskan)*	:	**RM 78 000.00**
Amount Spent *(Jumlah Perbelanjaan)*	:	**RM 10 242.30**
Balance *(Baki)*	:	**RM 67 757.70**
Percentage of Amount Spent *(Peratusan Belanja)*	:	13.13 %

ADDITIONAL RESEARCH ACTIVITIES THAT CONTRIBUTE TOWARDS DEVELOPING SOFT AND HARD SKILLS (Aktiviti Penyelidikan Sampingan yang menyumbang kepada pembangunan kemahiran insaniah)

D

International

Activity	Date (Month, Year)	Organizer
(e.g : Course/ Seminar/ Symposium/ Conference/ Workshop/ Site Visit)		

National

Activity	Date (Month, Year)	Organizer
(e.g : Course/ Seminar/ Symposium/ Conference/ Workshop/ Site Visit)		

E PROBLEMS / CONSTRAINTS IF ANY *(Masalah/ Kekangan sekiranya ada)*

The main problem is no GRA (local) to work on the project since the beginning of the project (May 2013). We have advertised on MyGrant's system to recruit & hire GRA post for the project. Through my co-supervisor, we have also broaden the recruitment advertisement to School of Mathematic, USM. While at the same time, in every new semester starting in September 2013 intake, I have personally invited new Master in CS students to be part of the project. Unfortunately, despite all these efforts we have not get GRA interested in the project. We have taken a "piecemeal" approach for over a period of time, to only partially address the project's objectives – refers to the past progress reports. We hired on honorarium monthly basis, existing PhD's students to work on the specific tasks of the project. The problem with this approach is no sustainability & continuity of the research work – several of these PhD students have then left, and furthermore, they are not trained with the specific knowledge required in the project. They were mainly employed to implement & deliver specific tasks that we've formulated and modelled as per module (or piece by piece) e.g. implementing specific task/module on Matlab/C for testing purposes. The transition period of hiring new student, once the last one left also quite substantials, often it takes months to get one good (and willing) candidate, case in point was the recent situation where we have not get any student assistant to work on the project from Feb 2015 till the end of the extended project's period – this explain why there was no real progress from Feb 2015 to Oct 2015.

There are other related issues as well, for instance, we have failed to fully devise a comprehensive idea of the research problem in the midst and rush of preparing the proposal form. Upon further research in the specific area of spatial filter design, we have found extensive works in gradient filter design have be done, both in the area of spatial as well as in signal processing field (mainly in EE). Attempts were made to re-focus the work on a few other possibilities, this includes extending the gradient's filter into 3D domain and looking at the aspect of accelerating the gradient filter algorithm using GPU. All these efforts requires fulltime engagement of GRA, which as explained above this was not fully materialized. The other part of the problem is more of a personal issue. For a period of 5 months from April 2014 – August 2014, I have served my sabbatical leave locally and abroad (spent 3 months at NUS). This event of my personal development undoubtedly have affected the progress of the project.

In general, we have achieved 55-65 % progress with the spending of 13% on the budget, the extension period seeked and granted (6 months) was part of the attempt to salvage and to address some if not all of the issues impeding the research project's progress. It is now clear, despite my efforts to complete phase 3 of the project, I may not be able to fulfil the originally stated research objectives, and the expected deliveries as agreed in the original proposal.

F RECOMMENDATION (*Cadangan Penambahbaikan*)

1. It is clear, to fully deliver in a fundamental research project such as FRGS, first we need to secure a PhD/MSc student prior to applying for the grant.
2. Second, with such student secured idea or plan in the problem to be researched is better conceived with prior thematic review of the research relevant works have been done
3. Flexibility should be given to engage Non-Malaysian GRA in FRGS – in the earlier part of my project (circa 2013), we were not allowed to recruit Non-Malaysian GRA, but the rule has been "relaxed" since then.

On personal basis, I will continue researching on this topic with the focus as highlighted in section E above. This is my first failure of completing a research project, and it is a first of it kind in "fundamental" research that I have headed. Certainly I have learnt some very good lessons including those issues that appear to be crucial for FRGS.

G RESEARCH ABSTRACT – Not More Than 200 Words *(Abstrak Penyelidikan – Tidak Melebihi 200 patah perkataan)*

Gradient is mostly used to approximate a quantity called normal vector, which is widely used in techniques such as ray-casting or shading to calculate the shade of a pixel. Central Different Equation (CDE), Forward Difference Equation (FDE) and Backward Difference Equation (BDE) [1] are examples of common finite-difference gradient filters to estimate gradient from discrete data. We observed that gradients estimated these ways are very sensitive to the method employed, and worse if the volumetric data exhibits some patterns (or noise) which is common in CT-scan and MRI, the results we obtained are unpredictable, as small perturbations in the dataset results in significant difference in the results.

In this research we are proposing to study the fundamental properties and structures of finite-differences gradient

	estimators for discrete volumetric data. This will enable us to discover causal relationships within the discrete volumetric data that are prone to uncertainties. We would also explore and carry out empirical research to discover key causal relationships inherent on complex 3D data. Thus the main objective is to address the instability problem in Finite-Difference (FD) method by proposing an alternative multi-scale approach. Extracting shape features from 3D volumetric data directly is crucial for certain critical applications e.g. performing surgery or laser cancer treatment on a patient's brain. With this FRGS grant, we hope to uncover the inherent limitations in existing finite difference methods to estimate gradients (the key component for feature extraction) and proposed a much robust methods to handle uncertainties in discrete volumetic data.
	Date : 28 Jan 2016 *Tarikh* — **Project Leader's Signature:** *Tandatangan Ketua Projek*
H	**COMMENTS, IF ANY/ ENDORSEMENT BY RESEARCH MANAGEMENT CENTER (RMC)** *(Komen, sekiranya ada/ Pengesahan oleh Pusat Pengurusan Penyelidikan)*
	PI ini akan disuarai kutan. **Name:** *Nama:* PROF. DR LEE KEAT TEONG Directo. Research Creativity & Management Office Universiti Sains Malaysia **Signature:** *Tandatangan:* **Date:** *Tarikh:*

6/2/16

Tarikh Cetakan : 28/01/2016

UNIVERSITI SAINS MALAYSIA

JABATAN BENDAHARI

PENYATA PERBELANJAAN SEHINGGA 28 JANUARI 2016

Projek : A MULTI-SCALE APPROACH FOR ESTIMATING GRADIENT FROM VOLUMETRIC DATA.
TEMPOH : 01 MEI 2013 HINGGA 30 APRIL 2015
KETUA PENYELIDIK : PROFESOR MADYA BAHARI BELATON
PENYELIDIK BERSAMA : -
PUSAT PENGAJIAN SAINS KOMPUTER

No. Akaun : 203.PKOMP.6711318.

Vot	Nama Vot	Peruntukan Projek	Perbelanjaan Terkumpul Sehingga Thn Lalu	Baki Peruntukan Tahun Lalu	Peruntukan Thn Semasa	Jumlah Peruntukan Thn Semasa	Tanggungan Semasa	Bayaran Thn Semasa	Jum Belanja Thn Semasa	Baki Projek
111	GAJI	48,000.00	0.00	48,000.00	0.00	48,000.00	0.00	0.00	0.00	48,000.00
221	PERJALANAN DAN SARA HIDUP	13,000.00	0.00	13,000.00	0.00	13,000.00	0.00	0.00	0.00	13,000.00
227	BEKALAN DAN BAHAN LAIN	2,000.00	218.00	1,782.00	0.00	1,782.00	0.00	0.00	0.00	1,782.00
228	PENYELENGGARAN & PEMBAIKAN KECIL	4,000.00	0.00	4,000.00	0.00	4,000.00	0.00	0.00	0.00	4,000.00
229	PERKHIDMATAN IKTISAS & HOSPITALITI	7,800.00	7,924.30	-124.30	0.00	-124.30	0.00	0.00	0.00	-124.30
335	HARTA MODAL	3,200.00	0.00	3,200.00	0.00	3,200.00	0.00	0.00	0.00	3,200.00
441	BIASISWA DAN GERAN PELAJARAN	0.00	2,100.00	-2,100.00	0.00	-2,100.00	0.00	0.00	0.00	-2,100.00
	Jumlah	**78,000.00**	**10,242.30**	**67,757.70**	**0.00**	**67,757.70**	**0.00**	**0.00**	**0.00**	**67,757.70**

Penyata ini adalah cetakan komputer tiada tandatangan diperlukan
Penyata ini adalah dianggap tepat jika tiada maklumbalas dalam tempoh masa 14 hari dari tarikh penyata

The Progress Report of Project (in April)

Introduction

Tasks for the month of April 2014 (21 Apr – 30 Apr)

1. Perform 3D Gaussian filtering to estimate gradients on a set of 3D discrete data analytically generated from an ~~implicit~~ function:

a. Identify the functions, you can have more than one functions with different level of complexities (they must be 3D functions) (in 5.1)

b. Generate sample dataset for various resolutions, says start with a very coarse sampling, e.g. 10 x 10 x 10, than vary gradually to higher resolutions (in 5.2)

c. For each of the sample dataset generated above perform Gaussian kernel to estimate gradients from the scalar values :

- ~~Vary the mu and sigma or determine them analytically based from some theory~~ (reason in section4)
- Vary the kernel filter size i.e. the window, to capture the best estimate; typical sizes are 3x3x3, 7x7x7 or 15x15x15, etc.

d. Store or collect the results for further evaluations and analysis (see below) (in 5.3)

2. Calculate analytically from the chosen implicit function the gradient at each point for the various resolutions (see 1. (b) above). This is to establish the "ground-truth" data for analysis & comparison with Gaussian's estimated gradient (in 5.1 & 5.3)

~~3. Perform a 2 nd order derivative estimation for both discrete data (Gaussian) and analytically on the implicit function.~~ (Gradient refers to the first order partial derivatives)

4. Perform the evaluation and analysis from the data collected in 1 and 2. You can do this in various ways, for instance you can evaluate the accuracy of Gaussian's estimated gradients with the ground-truth data via errors analysis, etc. (solution: magnitude of gradient is computed, then MSE measures the CDE and CDE combined Gaussian filter)

5. Prepare a technical report of your findings (similar to the one you've prepared earlier for the Finite Difference theoretical analysis).

In addition, the contents related to the gradient and Gaussian smoothing have been prepared in the first four sections.

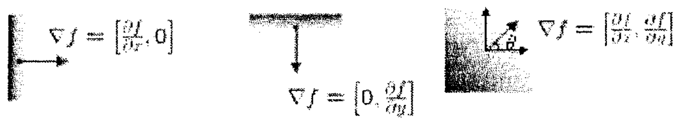
- ✓ section1 introduce the gradient concept
- ✓ Section2 provides the computation algorithms of gradient for 2D discrete data
- ✓ Section3 provides the computation algorithms of gradient for 3D discrete data
- ✓ Section4 introduces centered Gaussian kernel (there is no mu), the algorithm of smoothing data using Gaussian filter

1. Basic concept of gradient

For a two-variables function $f = f(x, y)$, the gradient of function f is defined as

$$\nabla f = \frac{\partial f}{\partial x}\vec{i} + \frac{\partial f}{\partial y}\vec{j} = f'_x\vec{i} + f'_y\vec{j} \qquad (1)$$

Usually, the abbreviated form $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right] = [f'_x, f'_y]$ is employed. Points in gradient direction have the most rapid change



$\nabla f = [\frac{\partial f}{\partial x}, 0]$
$\nabla f = [0, \frac{\partial f}{\partial y}]$
$\nabla f = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}]$

Figure1 Horizontal, vertical and general gradients

The magnitude of gradient is

$$\| \nabla f\| = \sqrt{(f'_x)^2 + (f'_y)^2} \qquad (2)$$

And the gradient direction is given by

$$\theta = tan^{-1}(f'_y/f'_x) \qquad (3)$$

For three-variables function $f = f(x, y, z)$, the gradient of function f is defined as

$$\nabla f = \frac{\partial f}{\partial x}\vec{i} + \frac{\partial f}{\partial y}\vec{j} + \frac{\partial f}{\partial z}\vec{k} = f'_x\vec{i} + f'_y\vec{j} + f'_z\vec{k} \qquad (4)$$

Usually, the abbreviated form $\nabla f = [f'_x, f'_y, f'_z]$ is employed.

The magnitude of gradient is

$$\| \nabla f\| = \sqrt{(f'_x)^2 + (f'_y)^2 + (f'_z)^2} \qquad (5)$$

2. Calculation gradient based on centered difference equation (CDE) for discrete data generated by two-dimensional function

Function $[f_x, f_y] = gradient(f, h_x, h_y)$ returns the numerical gradient of a matrix f. f_x corresponds to $\frac{df}{dx}$, the differences in x (horizontal) direction. f_y corresponds to $\frac{df}{dy}$, the differences in y (vertical) direction. Where $f = f(x, y)$ is a 2D matrix, h_x, h_y are chosen as scalar to represent the spacing between points in each direction. The default value is set as $h_x = h_y = 1$. The calculation formulas of the first order partial derivatives are described as:

$[m, n] = size(f)$

When $j = 1$

$f_x(i, 1) = f(i, 2) - f(i, 1)$

When $j = n$

$f_x(i, n) = f(i, n) - f(i, n - 1)$

When $1 < j < n$

$$f_x(i, j) = \frac{f(i, j + 1) - f(i, j - 1)}{2h_x}$$

The similar to the first order derivative for y

When $i = 1$

$f_y(1, j) = f(2, j) - f(1, j)$

When $i = m$

$f_y(m, j) = f(m, j) - f(m - 1, j)$

When $1 < i < m$

$$f_y(i, j) = \frac{f(i + 1, j) - f(i - 1, j)}{2h_y}$$

Figure2 Calculation formulas of the first order partial derivatives for two-dimensional function

Example1

$$A = \begin{bmatrix} 8 & 9 & 6 & 9 & 2 \\ 5 & 8 & 7 & 0 & 1 \\ 4 & 5 & 4 & 6 & 9 \\ 8 & 1 & 0 & 9 & 5 \end{bmatrix}$$

Using $[A_x, A_y] = gradient(A)$, then we can obtain

$$A_x = \begin{bmatrix} 1 & -1 & 0 & -2 & -7 \\ 3 & 1 & -4 & -3 & 1 \\ 1 & 0 & 0.5 & 2.5 & 3 \\ -7 & -4 & 4 & 2.5 & -4 \end{bmatrix}$$

$$A_y = \begin{bmatrix} -3 & -1 & 1 & -9 & -1 \\ -2 & -2 & -1 & -1.5 & 3.5 \\ 1.5 & -3.5 & -3.5 & 4.5 & 2 \\ 4 & -4 & -4 & 3 & -4 \end{bmatrix}$$

Next, we can draw the gradient of each point, at the same time, mark the original data value in the corresponding position.

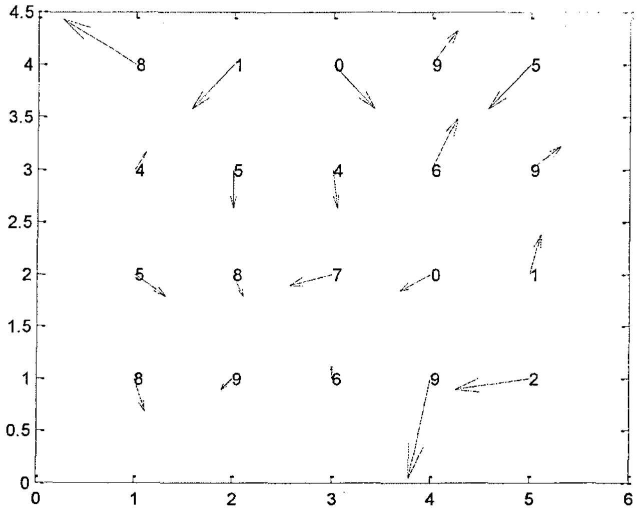


Figure3 Gradient of matrix A

The command of drawing Figure3 is recorded as:

```
>>figure
>>quiver(Ax,Ay)
>>for i=1:5
>>for j=1:4
>>text(i,j,num2str(A(j,i)))
          >>end
>>end
```

In Figure3, the values are marked in the corresponding coordinates, at the same time, the gradient (using a vector) of each point is labelled.

3. Calculation gradient based on centered difference equation (CDE) for discrete data generated by three-dimensional function

Function $[f_x, f_y, f_z] = gradient(f, h_x, h_y, h_z)$ returns the numerical gradient of a 3D matrix f. f_z corresponds to $\frac{df}{dz}$, the differences in the z direction. Here, suppose matrix f is $m \times n \times p$ entries which is able to be viewed a piled up 2D matrix to form the thick (altitude). Based on this consideration, the calculation of the first order partial derivatives about x, y, z respectively is generalized from two-variable function.

```
For k=1:p
temp=f(:,:,k);
Compute partial derivatives $f_x^k, f_y^k$ of temp matrix according to the formulas described in Figure2
$f_x(:,:,k) = f_x^k$
$f_y(:,:,k) = f_y^k$
End
To calculate the partial derivative with respect to z
When $k = 1$
$f_z(:,:,1) = f(:,:,2) - f(:,:,1)$
When $k = p$
$f_z(:,:,p) = f(:,:,p) - f(:,:,p-1)$
When $1<k<p$
$f_z(:,:,k) = \frac{f(:,:,k+1) - f(:,:,k-1)}{2h_z}$
```

Figure4 Compute the first order partial derivatives for three-dimensional function

Example2 Give 3D matrix v, the data is depicted as

```
v(:,:,1) =
 -0.0600  -0.2107  -0.5531  -0.9835  -1.4334  -1.8742  -2.2994 -2.7122  -3.1205  -3.5362
  0.1743  -0.0598  -0.5511  -1.1171  -1.6754  -2.2094  -2.7276  -3.2498  -3.8085  -4.4695
  0.2855  -0.0017  -0.5771  -1.2119  -1.8245  -2.4100  -2.9895  -3.6013  -4.3242  -5.4343
  0.1743  -0.0598  -0.5511  -1.1171  -1.6754  -2.2094  -2.7276  -3.2498  -3.8085  -4.4695
 -0.0600  -0.2107  -0.5531  -0.9835  -1.4334  -1.8742  -2.2994  -2.7122  -3.1205  -3.5362

v(:,:,2) =
  0.1743  -0.0598  -0.5511  -1.1171  -1.6754  -2.2094  -2.7276 -3.2498  -3.8085  -4.4695
   0.6298   0.1032  -0.7809  -1.6442  -2.4647  -3.3258  -4.4676  -6.4575  -4.7292  -4.3451
   0.9720   0.0586  -1.1855  -2.3499  -3.7080  -5.6155  -4.1380  -3.8390  -3.7335  -3.6999
   0.6298   0.1032  -0.7809  -1.6442  -2.4647  -3.3258  -4.4676 -6.4575  -4.7292  -4.3451
   0.1743  -0.0598  -0.5511  -1.1171  -1.6754  -2.2094  -2.7276   -3.2498  -3.8085  -4.4695

v(:,:,3) =

  0.2855  -0.0017  -0.5771  -1.2119  -1.8245  -2.4100  -2.9895 -3.6013  -4.3242  -5.4343
   0.9720   0.0586  -1.1855  -2.3499  -3.7080  -5.6155  -4.1380  -3.8390  -3.7335  -3.6999
   1.6094  -0.8755  -1.5261  -1.9169  -2.1972  -2.4159  -2.5953 -2.7473  -2.8792  -2.9957
   0.9720   0.0586  -1.1855  -2.3499  -3.7080  -5.6155  -4.1380  -3.8390  -3.7335  -3.6999
   0.2855  -0.0017  -0.5771  -1.2119  -1.8245  -2.4100  -2.9895   -3.6013  -4.3242  -5.4343

v(:,:,4) =
  0.1743  -0.0598  -0.5511  -1.1171  -1.6754  -2.2094  -2.7276   -3.2498  -3.8085  -4.4695
   0.6298   0.1032  -0.7809  -1.6442  -2.4647  -3.3258  -4.4676   -6.4575  -4.7292  -4.3451
   0.9720   0.0586  -1.1855  -2.3499  -3.7080  -5.6155  -4.1380   -3.8390  -3.7335  -3.6999
   0.6298   0.1032  -0.7809  -1.6442  -2.4647  -3.3258  -4.4676   -6.4575  -4.7292  -4.3451
   0.1743  -0.0598  -0.5511  -1.1171  -1.6754  -2.2094  -2.7276    -3.2498  -3.8085  -4.4695

v(:,:,5) =

  -0.06  -0.2107  -0.5531  -0.9835  -1.4334  -1.8742  -2.2994   -2.7122  -3.1205  -3.5362
   0.1743  -0.0598  -0.5511  -1.1171  -1.6754  -2.2094  -2.7276   -3.2498  -3.8085  -4.4695
   0.2855  -0.0017  -0.5771  -1.2119  -1.8245  -2.4100  -2.9895   -3.6013  -4.3242  -5.4343
   0.1743  -0.0598  -0.5511  -1.1171  -1.6754  -2.2094  -2.7276  -3.2498  -3.8085  -4.4695
  -0.0600  -0.2107  -0.5531  -0.9835  -1.4334  -1.8742  -2.2994   -2.7122  -3.1205  -3.5362
```

****

$[f_x, f_y, f_z] = gradient(f)$, the default spacing of each direction is 1. The derivatives are achieved and listed in the following:

```
dx(:,:,1)=
  -0.1507  -0.2466  -0.3864  -0.4401  -0.4454  -0.433  -0.419 -0.4106  -0.4120  -0.4157
  -0.2341  -0.3627  -0.5287  -0.5622  -0.5461  -0.5261  -0.5202  -0.5405  -0.6099  -0.6610
  -0.2871  -0.4313  -0.6051  -0.6237  -0.5991  -0.5825  -0.5957   -0.6674  -0.9165  -1.1101
  -0.2341  -0.3627  -0.5287  -0.5622  -0.5461  -0.5261  -0.5202  -0.5405  -0.6099  -0.6610
  -0.1507  -0.2466  -0.3864  -0.4401  -0.4454  -0.4330  -0.4190 -0.4106  -0.4120  -0.4157

dx(:,:,2)=
  -0.2341  -0.3627  -0.5287  -0.5622  -0.5461  -0.5261  -0.5202 -0.5405  -0.6099  -0.6610
  -0.5266  -0.7054  -0.8737  -0.8419  -0.8408  -1.0015  -1.5658   -0.1308   1.0562   0.3841
  -0.9133  -1.0787  -1.2043  -1.2613  -1.6328  -0.2150   0.8882    0.2022   0.0696   0.0336
  -0.5266  -0.7054  -0.8737  -0.8419  -0.8408  -1.0015  -1.5658   -0.1308   1.0562   0.3841
  -0.2341  -0.3627  -0.5287  -0.5622  -0.5461  -0.5261  -0.5202   -0.5405  -0.6099  -0.6610

dx(:,:,3)=
  -0.2871  -0.4313  -0.6051  -0.6237  -0.5991  -0.5825  -0.5957   -0.6674  -0.9165  -1.1101
  -0.9133  -1.0787  -1.2043  -1.2613  -1.6328  -0.2150   0.8882    0.2022   0.0696   0.0336
  -2.4849  -1.5677  -0.5207  -0.3356  -0.2495  -0.1990  -0.1657   -0.1420  -0.1242  -0.1165
  -0.9133  -1.0787  -1.2043  -1.2613  -1.6328  -0.2150   0.8882    0.2022   0.0696   0.0336
  -0.2871  -0.4313  -0.6051  -0.6237  -0.5991  -0.5825  -0.5957   -0.6674  -0.9165  -1.1101


dx(:,:,4)=
  -0.2341  -0.3627  -0.5287  -0.5622  -0.5461  -0.5261  -0.5202   -0.5405  -0.6099  -0.6610
  -0.5266  -0.7054  -0.8737  -0.8419  -0.8408  -1.0015  -1.5658   -0.1308   1.0562   0.3841
  -0.9133  -1.0787  -1.2043  -1.2613  -1.6328  -0.2150   0.8882    0.2022   0.0696   0.0336
  -0.5266  -0.7054  -0.8737  -0.8419  -0.8408  -1.0015  -1.5658   -0.1308   1.0562   0.3841
  -0.2341  -0.3627  -0.5287  -0.5622  -0.5461  -0.5261  -0.5202   -0.5405  -0.6099  -0.6610

dx(:,:,5)=
  -0.1507  -0.2466  -0.3864  -0.4401  -0.4454  -0.4330  -0.4190   -0.4106  -0.4120  -0.4157
  -0.2341  -0.3627  -0.5287  -0.5622  -0.5461  -0.5261  -0.5202   -0.5405  -0.6099  -0.6610
  -0.2871  -0.4313  -0.6051  -0.6237  -0.5991  -0.5825  -0.5957   -0.6674  -0.9165  -1.1101
  -0.2341  -0.3627  -0.5287  -0.5622  -0.5461  -0.5261  -0.5202   -0.5405  -0.6099  -0.6610
  -0.1507  -0.2466  -0.3864  -0.4401  -0.4454  -0.4330  -0.4190   -0.4106  -0.4120  -0.4157
```

****

```
dy(:,:,1)=
   0.2343   0.1509   0.0020  -0.1336  -0.2420  -0.3351  -0.4282   -0.5376  -0.6880  -0.9333
   0.1727   0.1045  -0.0120  -0.1142  -0.1956  -0.2679  -0.3451   -0.4446  -0.6019  -0.9490
     0        0       0       0       0       0       0        0        0       0
  -0.1727  -0.1045   0.0120   0.1142   0.1956   0.2679   0.3451    0.4446   0.6019   0.9490
  -0.2343  -0.1509  -0.0020   0.1336   0.2420   0.3351   0.4282    0.5376   0.6880   0.9333

dy(:,:,2)=
   0.4555   0.1629  -0.2298  -0.5271  -0.7892  -1.1165  -1.7400   -3.2077  -0.9207   0.1244
   0.3988   0.0592  -0.3172  -0.6164  -1.0163  -1.7031  -0.7052   -0.2946   0.0375   0.3848
     0        0       0       0       0       0       0        0        0       0
  -0.3988  -0.0592   0.3172   0.6164   1.0163   1.7031   0.7052    0.2946  -0.0375  -0.3848
  -0.4555  -0.1629   0.2298   0.5271   0.7892   1.1165   1.7400    3.2077   0.9207  -0.1244

dy(:,:,3)=
   0.6865   0.0603  -0.6084  -1.1380  -1.8835  -3.2055  -1.1485   -0.2377   0.5908   1.7345
   0.6620  -0.4369  -0.4745  -0.3525  -0.1864  -0.0029   0.1971    0.4270   0.7225   1.2193
     0        0       0       0       0       0       0        0        0       0
  -0.6620   0.4369   0.4745   0.3525   0.1864   0.0029  -0.1971  -0.4270  -0.7225  -1.2193
  -0.6865  -0.0603   0.6084   1.1380   1.8835   3.2055   1.1485   0.2377  -0.5908  -1.7345

dy(:,:,4)=
   0.4555   0.1629  -0.2298  -0.5271  -0.7892  -1.1165  -1.7400  -3.2077  -0.9207   0.1244
   0.3988   0.0592  -0.3172  -0.6164  -1.0163  -1.7031  -0.7052  -0.2946   0.0375   0.3848
     0        0        0        0        0        0        0        0        0        0
  -0.3988  -0.0592   0.3172   0.6164   1.0163   1.7031   0.7052   0.2946  -0.0375  -0.3848
  -0.4555  -0.1629   0.2298   0.5271   0.7892   1.1165   1.7400   3.2077   0.9207  -0.1244

dy(:,:,5)=
   0.2343   0.1509   0.0020  -0.1336  -0.2420  -0.3351  -0.4282  -0.5376  -0.6880  -0.9333
   0.1727   0.1045  -0.0120  -0.1142  -0.1956  -0.2679  -0.3451  -0.4446  -0.6019  -0.9490
     0        0        0        0        0        0        0        0        0        0
  -0.1727  -0.1045   0.0120   0.1142   0.1956   0.2679   0.3451   0.4446   0.6019   0.9490
  -0.2343  -0.1509  -0.0020   0.1336   0.2420   0.3351   0.4282   0.5376   0.6880   0.9333
************************************************************

dz(:,:,1)=
   0.2343   0.1509   0.0020  -0.1336  -0.2420  -0.3351  -0.4282  -0.5376  -0.6880  -0.9333
   0.4555   0.1629  -0.2298  -0.5271  -0.7892  -1.1165  -1.7400  -3.2077  -0.9207   0.1244
   0.6865   0.0603  -0.6084  -1.1380  -1.8835  -3.2055  -1.1485  -0.2377   0.5908   1.7345
   0.4555   0.1629  -0.2298  -0.5271  -0.7892  -1.1165  -1.7400  -3.2077  -0.9207   0.1244
   0.2343   0.1509   0.0020  -0.1336  -0.2420  -0.3351  -0.4282  -0.5376  -0.6880  -0.9333


dz(:,:,2)=
   0.1727   0.1045  -0.0120  -0.1142  -0.1956  -0.2679  -0.3451  -0.4446  -0.6019  -0.9490
   0.3988   0.0592  -0.3172  -0.6164  -1.0163  -1.7031  -0.7052  -0.2946   0.0375   0.3848
   0.6620  -0.4369  -0.4745  -0.3525  -0.1864  -0.0029   0.1971   0.4270   0.7225   1.2193
   0.3988   0.0592  -0.3172  -0.6164  -1.0163  -1.7031  -0.7052  -0.2946   0.0375   0.3848
   0.1727   0.1045  -0.0120  -0.1142  -0.1956  -0.2679  -0.3451  -0.4446  -0.6019  -0.9490


dz(:,:,3)=
   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0

dz(:,:,4)=
  -0.1727  -0.1045   0.0120   0.1142   0.1956   0.2679   0.3451   0.4446   0.6019   0.9490
  -0.3988  -0.0592   0.3172   0.6164   1.0163   1.7031   0.7052   0.2946  -0.0375  -0.3848
  -0.6620   0.4369   0.4745   0.3525   0.1864   0.0029  -0.1971  -0.4270  -0.7225  -1.2193
  -0.3988  -0.0592   0.3172   0.6164   1.0163   1.7031   0.7052   0.2946  -0.0375  -0.3848
  -0.1727  -0.1045   0.0120   0.1142   0.1956   0.2679   0.3451   0.4446   0.6019   0.9490


dz(:,:,5)=
  -0.2343  -0.1509  -0.0020   0.1336   0.2420   0.3351   0.4282   0.5376   0.6880   0.9333
  -0.4555  -0.1629   0.2298   0.5271   0.7892   1.1165   1.7400   3.2077   0.9207  -0.1244
  -0.6865  -0.0603   0.6084   1.1380   1.8835   3.2055   1.1485   0.2377  -0.5908  -1.7345
  -0.4555  -0.1629   0.2298   0.5271   0.7892   1.1165   1.7400   3.2077   0.9207  -0.1244
  -0.2343  -0.1509  -0.0020   0.1336   0.2420   0.3351   0.4282   0.5376   0.6880   0.9333
```

Next, for $[f_x, f_y, f_z] = gradient(f, h_x, h_y, h_z)$, change h_x, h_y, h_z value to verify the variations of partial derivatives. Three set scalars are employed

- (1) $h_x = h_y = h_z = 1$ (The experimental results are displayed exhaustively in example2)
- (2) $h_x = 1.1, h_y = 1.5, h_z = 1.5$
- (3) $h_x = h_y = h_z = 0.8$

The comparisons are performed about three partial derivatives respectively shown in Figure5, Figure8 and Figure9. Firstly, a $5 \times 10 \times 5$ matrix is transferred to a column vector, hence the length of this vector is 250. In order to illustrate clearly, a factor 10 is used to enlarge all the derivatives.

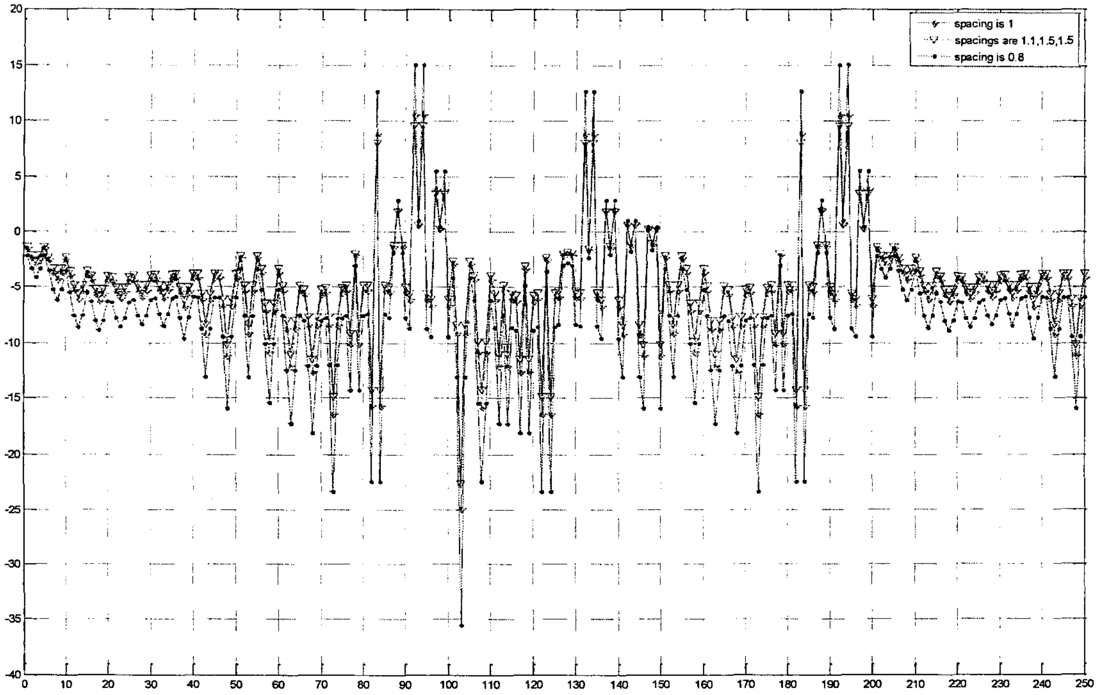


Figure5 First order partial derivatives with respect to x. The blue line

In order to observe clearly the trend of Figure5, two local enlarged figures are illustrated in the following Figure6 and Figure7.

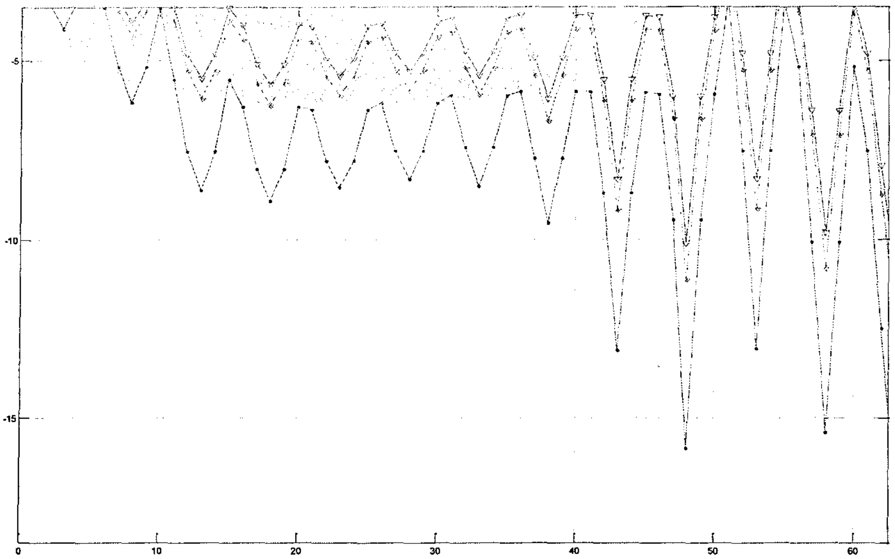


Figure6 First part of the enlarged Figure5

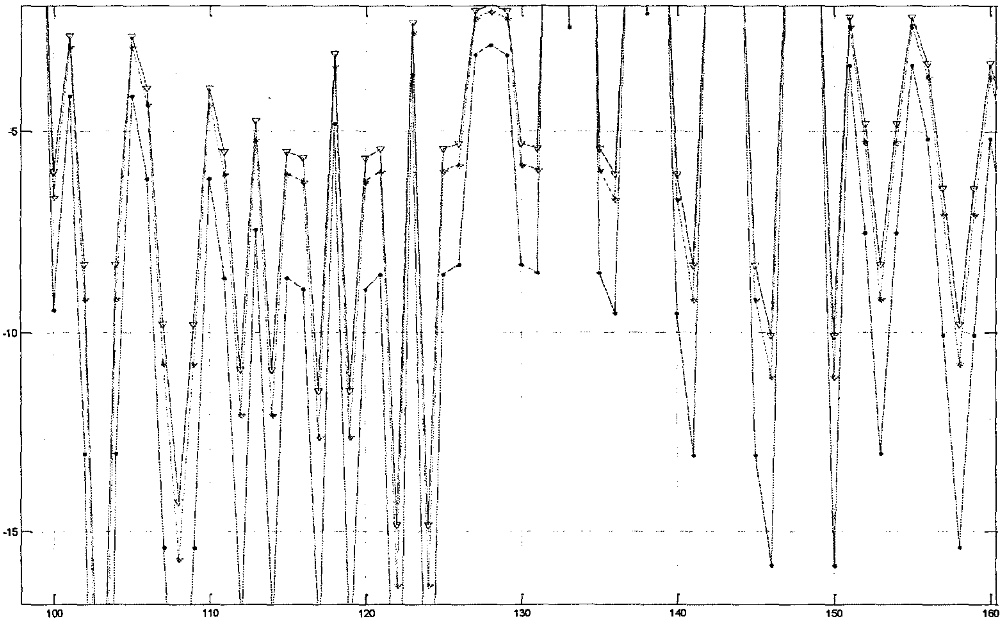


Figure7 Middel part of the enlarged Figure5

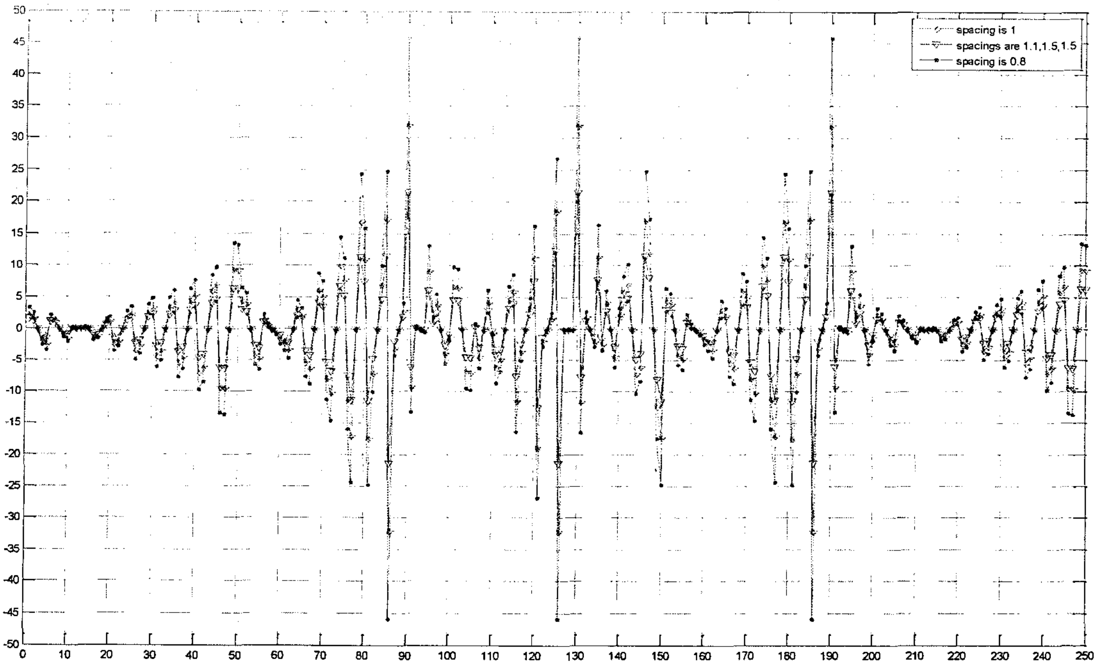


Figure8 First order partial derivatives with respect to y for three cases

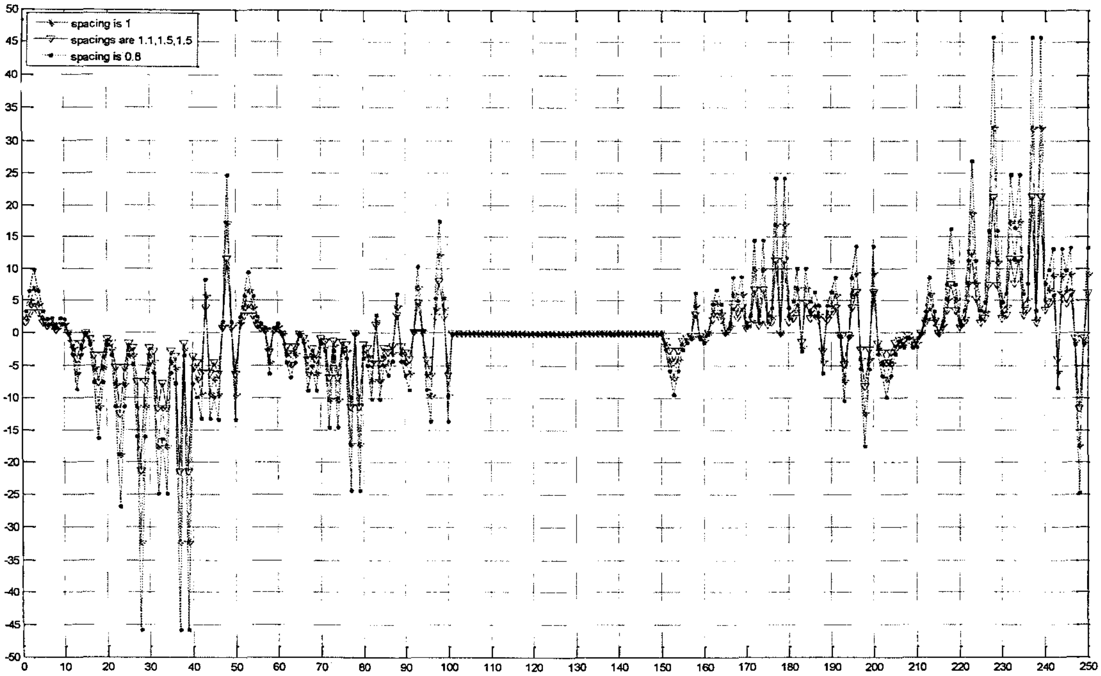


Figure9 First order partial derivatives with respect to z

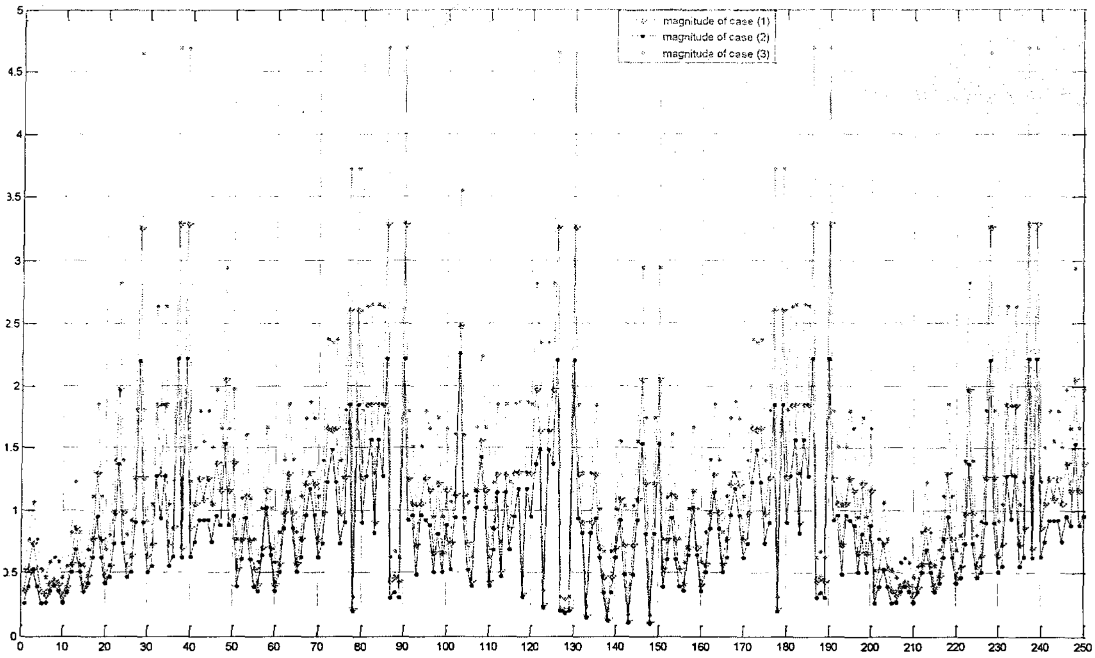


Figure10 Magnitudes of three cases

Observation carefully Figure5, Figure8 and Figure9, we can easily found

- The variation amplitude of the case of (2) is the smallest among three cases. That is to say, when$h_x = 1.1, h_y = 1.5, h_z = 1.5$, the first order partial derivatives are the most accurate.
- Furthermore, we calculate the magnitude of gradient for these three cases (shown in Figure10). The means and standard variances are listed in Table1

Table1. The mean and standard variance of three magnitudes

	(1)	(2)	(3)
Mean	1.092	0.8338	1.56
std	0.7061	0.4904	1.0087

4. Gaussian kernel filtering

- 1D Gaussian kernel: $f(x) = \frac{1}{\sqrt{2\pi\sigma}} \cdot exp(-\frac{x^2}{2\sigma^2})$
- 2D Gaussian kernel: $f(x,y) = \frac{1}{2\pi\sigma} \cdot exp(-\frac{x^2+y^2}{2\sigma^2})$
- 3D Gaussian kernel: $f(x,y,z) = \frac{1}{(\sqrt{2\pi\sigma})^3} \cdot exp(-\frac{x^2+y^2+z^2}{2\sigma^2})$

There are two solutions for 3D Gaussian filtering to smooth the given 3D data include

- smooth3
- fspecial3 generates a 3D Gaussian filter, and then imfilter.

Next, we will introduce clearly these two ways

① smooth3 function is used to smooth 3D data

w = smooth3(v, method, size, arg)

Inputs:

- 3D data set v
- Method option can be either of the filters 'gaussian' or 'box' (default) and determines the convolution kernel
- Size option sets the size of the convolution kernel (default is [3, 3, 3]). If size is a scalar, the size is interpreted as [size, size, size].
- Arg option sets an attribute of the convolution kernel. When method is 'gaussian', arg is the standard deviation (default is .65).

Outputs:

- The smoothed data is returned in W

② fspecial3 creates a 3-dimensional filter

h = fspecial3('gaussian',size)

- The default size is [5, 5, 5]. If size is a scalar then h is a 3D cubic filter of dimension size^3.
- Returns a centered Gaussian low pass filter of size
- Standard deviations defined as $\frac{size}{4 \cdot \sqrt{2\log 2}}$ so that FWHM equals half filter size.

  Such a FWHM-dependent standard deviation yields a congruous Gaussian shape.

5. Simulation experiments

5.1 Two three-dimensional functions

Firstly, we choose arbitrary two functions to generate the analytical data

①. $f(x,y,z) = x^2 + y^2 + z^2$, $x \in [0,3]$, $y \in [0,3]$ and $z \in [0,3]$

According to the definition of the first order partial derivatives, then we achieve:

$$\frac{\partial f}{\partial x} = f'_x = 2x \qquad (6.1)$$

$$\frac{\partial f}{\partial y} = f'_y = 2y \qquad (6.2)$$

$$\frac{\partial f}{\partial z} = f'_z = 2z \qquad (6.3)$$

② $g(x,y,z) = z \cdot \exp(-x^2 - y^2)$, $x \in [0,3]$, $y \in [0,3]$ and $z \in [0,3]$

$$\frac{\partial g}{\partial x} = g'_x = -2xz \cdot \exp(-x^2 - y^2) = -2x \cdot g(x,y,z) \qquad (7.1)$$

$$\frac{\partial g}{\partial y} = g'_y = -2yz \cdot \exp(-x^2 - y^2) = -2y \cdot g(x,y,z) \qquad (7.2)$$

$$\frac{\partial g}{\partial z} = g'_z = \exp(-x^2 - y^2) = \frac{1}{z} \cdot g(x,y,z) \qquad (7.3)$$

5.2 3D analytical data

In this section, we generate synthetic data based on the given functions and the domains of three independent variables x, y, z. The data descriptions are listed in the following table2.

Table2. Data generated by the given functions

	Function	Size	Description
Data1	$f(x,y,z) = x^2 + y^2 + z^2$	11×11×11	The spacing of x, y, z respective is the same 0.3
Data2	$f(x,y,z) = x^2 + y^2 + z^2$	51×51×51	The spacing of x, y, z respective is the same 0.06
Data3	$g(x,y,z) = z \cdot \exp(-x^2 - y^2)$	11×11×11	The spacing of x, y, z respective is the same 0.3
Data4	$g(x,y,z) = z \cdot \exp(-x^2 - y^2)$	51×51×51	The spacing of x, y, z respective is the same 0.06
Data5	$f(x,y,z) = x^2 + y^2 + z^2$	51×51×51	Add Gaussian noise into data2
Data6	$g(x,y,z) = z \cdot \exp(-x^2 - y^2)$	51×51×51	Add Gaussian noise into data4

Plot the isosurfaces of data2, data4 and their noisy data to indicate the difference between clean data and noisy data.

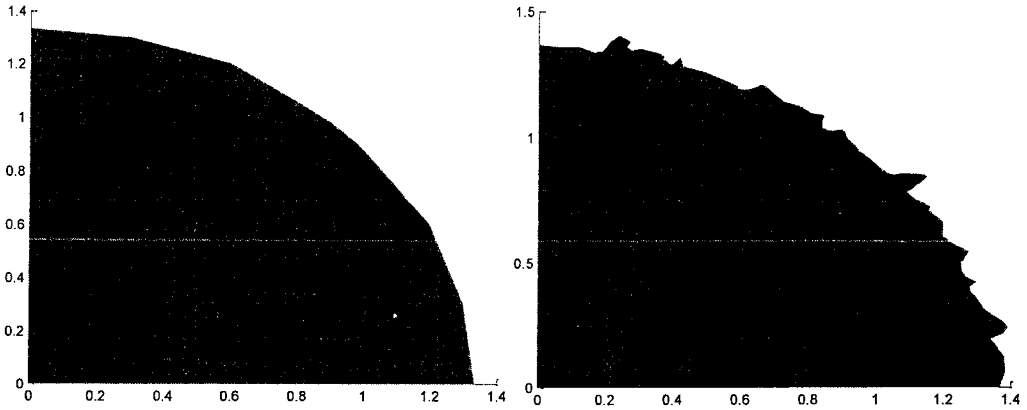


Figure11 the isosurface of data2 and noisydata2

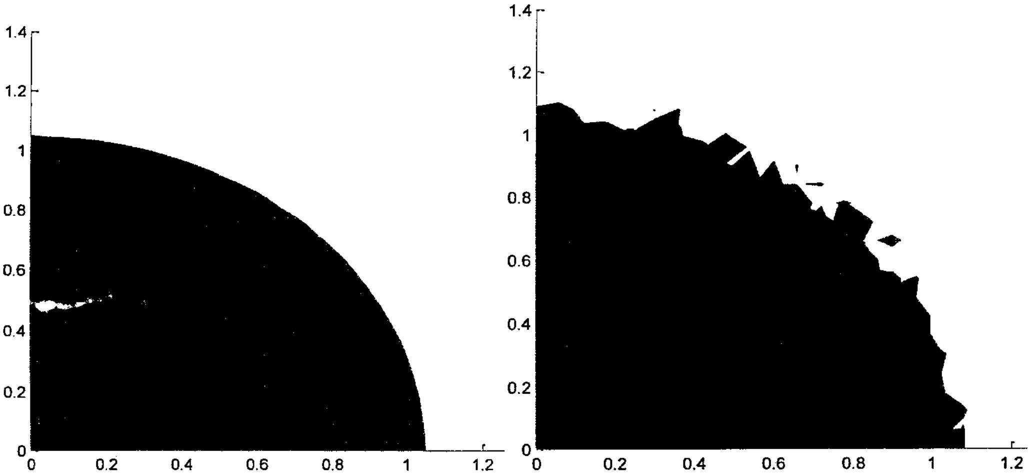


Figure12 the isosurface of data4 and noisydata4

5.3 The measurement of gradient

For a data, there are three methods to calculate the gradient denoting as

method1: The partial derivatives formulas (6) or (7);

method2: The centered difference formula described in Figure2 and Figure4;

method3: The data is smoothed by Gaussian filter, and then method2 is employed to calculate the gradient.

The magnitude of gradient is computed and to be compared. As we know, the gradients of method1 are the most precise, so they are considered as the ground-truth gradients and magnitudes. The magnitudes of gradients of the remainder two methods are expected to compare with ground-truth magnitude. The measurement is the mean squared error between the ground-truth magnitudes and the magnitudes

$$Err = \frac{1}{N}\sum_{i=1}^{N}(mag_i - truth_i)^2 \tag{8}$$

where $N = size(mag, 1) * size(mag, 2) * size(mag, 3)$

The exhaustive experimental results are stored in *functional data.mat* file. The mean squared errors are displayed in Table3 and Table4.

Table3 Three errors on the data generated by the first function

	CDE	Gaussian(size=[3,3,3])	Gaussian(size=[5,5,5])
Data1	0.01739	0.5059	0.5289
Data2	1.3843e-04	0.0663	0.0694
Data5	1.5868	0.1949	0.1878

Table4 Three errors on the data generated by the second function

	CDE	Gaussian(size=[3,3,3])	Gaussian(size=[5,5,5])
Data3	0.0062	0.0071	0.0074
Data4	9.0421e-04	6.1611e-04	6.1343e-04
Data6	4.0676	0.3291	0.3038

Conclusions:

- method2 (CDE is used individually) is superior to method3 (CDE based on Gaussian filter) for the clean data (data1,data2,data3 and data4)
- method3 can provide better results than method2 for the noisy data (data5 and data6)
- Gaussian filter can supply the slightly smaller MSE when window size is smaller. i.e., when size is 3, the smoothing is better than the smoothing of the case where size is 5 on clean data
- The opposite phenomenon is observed on noisy data.

Tasks for the month of May 2014 (first 2 weeks)

The above exploratory tasks are the initial stage for designing a multi-scale Gaussian-based spatial kernel filter. It should give you some idea about the sensitivity of the parameters involved and more generally the performance issues of Gaussian's filter. Based on these findings, we can do further fine-tuning to calibrate and test the filter against real discrete 3D data. So your task now are :

1. Perform Gaussian gradient (1st order) and curvature (2nd order) estimations against temperature dataset – it is a small simulated data. Get some help from Pan about the coding either for Matlab or (if necessary) for OpenGL (I think it is not necessary at this stage)
2. Evaluate and analyze the results, you can actually compare your findings against the results Pan's gathered for his thesis (not using Gaussian).
3. Prepare a technical report to report your findings and conclusion.

** This experiment is sort of a simple exposure of Gaussian to "real" dataset, where noise or errors are negligible. We should expect for Gaussian to perform quite well for this data – in fact I'm anticipating for Gaussian to address some of the limitations in Pan's work.

... potential tasks for the remaining 2 weeks of May 2014

At this stage, we should have a fairly good knowledge (verified by our experiment rather than based on theory) on how well Gaussian estimate gradient and curvature from discrete data. We are then at a position to decide what to do next ...

1. If it performance is reasonably good, we can proceed with Gaussian multi-scale spatial filter to apply it to real data set as generated from CT or MRI scan of a human skull. Note that in this case there are a fair number of noises in the dataset. Theoretically, Gaussian filter may not cope with these outliers very well, but we may have to proof it experimentally.
2. Or if its performance is not encouraging (even for the simple temperature data), we may have to think for other alternatives, including considering Bilateral kernel or the filter framework as proposed by Moller – based on taylor series expansion on some polynomial function. We have to decide this later.

Experiment1, some experiments are conducted on 3D temperature data:

① The description of the data is

Dataset structured_points

Dimensions: 18 ×18 ×10

Origin: {0, 0, 0}

Spacing: {1, 1, 1}

Point_data: 3240

② Filtering the temperature data with a 3D Gaussian filter

w = smooth3(v, method, size, arg), in here,

a) Change the window size from 3 to 9 (3, 5, 7, 9);

b) Vary the sigma from 0.7 to 2 (0.7, 1, 1.3, 1.5, 1.7, 2).

③ Compute the gradient by $[f_x, f_y, f_z] = gradient(f, h_x, h_y, h_z)$, $h_x = h_y = h_z = 1$.

The magnitude of gradient is $\| \nabla f \| = \sqrt{(f_x')^2 + (f_y')^2 + (f_z')^2}$. Some magnitudes are shown in the following two figures.

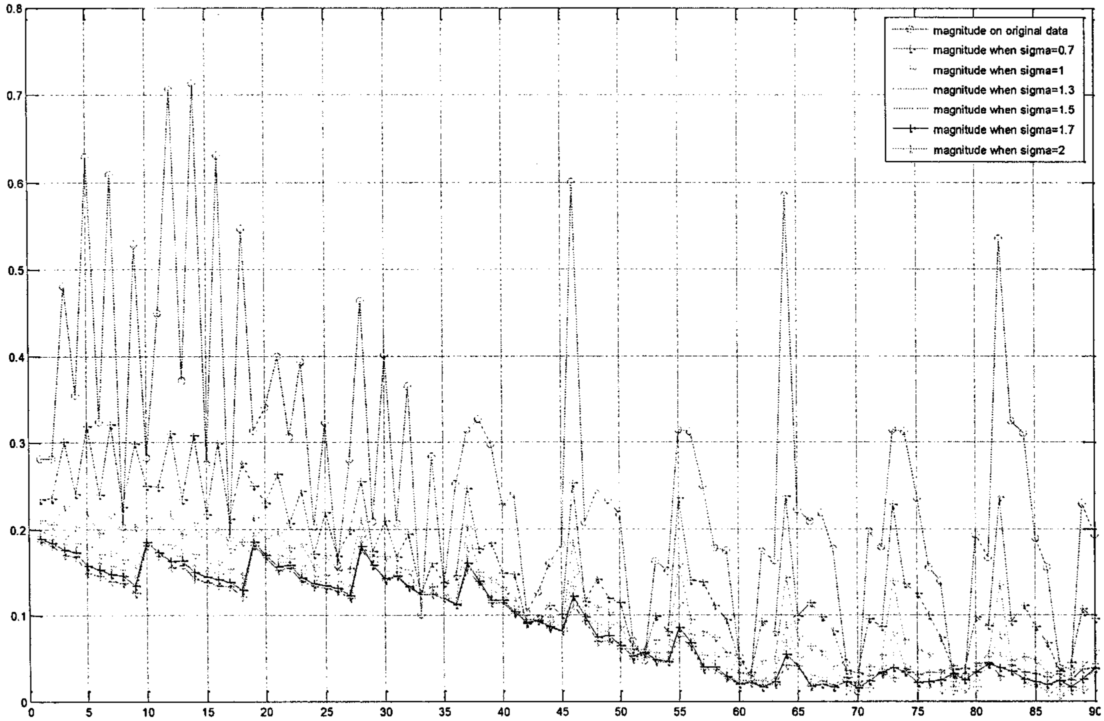


Fig1. The parts of magnitudes when window size=5.

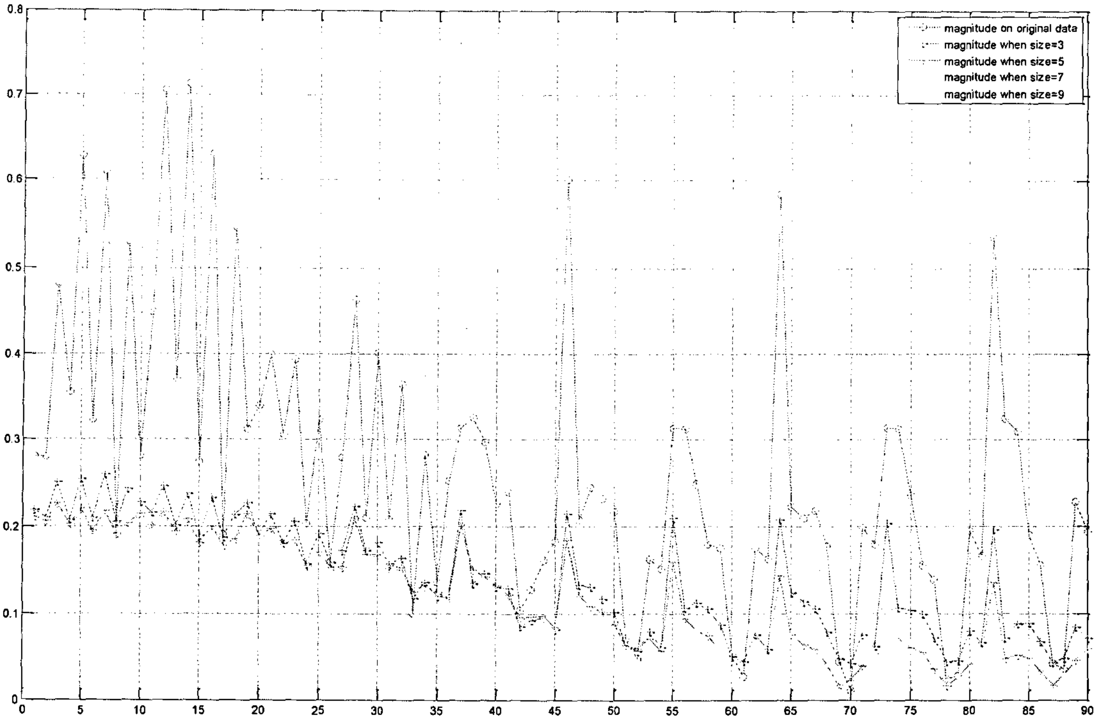


Fig2. The parts of magnitudes when sigma=1.

Observations from Fig.1 and Fig.2 :

- In Fig.1, when window size is fixed, the magnitudes of gradients on the Gasussian filtered temperature data (colourful lines with cross marks) are smaller than those on original temperature data (red line with circle marks).
- In Fig.1, for the Gasussian filtered temperature data, the larger the sigma in Gaussian filter, the smaller magnitudes of gradients.
- In Fig.2, when the sigma in Gaussian filter is fixed as 1, the magnitudes of gradients on the Gasussian filtered temperature data (colourful lines with cross marks) are smaller than those on original temperature data (red line with circle marks).
- In Fig.2, for the Gasussian filtered temperature data, window size in Gaussian filter are 5, 7 and 9, the magnitudes of gradients have slightly variation.
- The magnitudes of gradient are stable without the sharply peak-valley-variation when sigma becomes larger. That is to say, the proper sigma value should be taken.

From statistics viewpoint, the exhaustive mean magnitudes are displayed in table1

Table1. The mean magnitudes on the Gaussian filtered temperature data. Note that the mean magnitudes on the temperature data is 0.2422

	Size=3	Size=5	Size=7	Size=9
Sigma=0.7	0.1274	0.1220	0.1220	0.1220
Sigma=1	0.1053	0.0810	0.0796	0.0796
Sigma=1.3	0.0983	0.0608	0.0549	0.0545
Sigma=1.5	0.0960	0.0548	0.0454	0.0442
Sigma=1.7	0.0945	0.0520	0.0394	0.0373
Sigma=2	0.0932	0.0503	0.0341	0.0309

- Table 1 indicates the mean magnitudes decrease with the sigma increase ;
- When sigma is 0.7, the mean magnitudes keep the same when window size is larger than 5;
- The mean magnitudes decrease with the window size increase;
- Due to the data size limitation, there are no apparent difference between the mean magnitudes of size=7 and those of size=9.

Experiment 2, some experiments are conducted on 3D flow data:

① First create a 3D flow data and add some noise:

```
>>[x,y,z,V] = flow(60);
>>noisyV = V + 0.1*randn(size(V));
```

② Filter the flow data with a 3D Gaussian filter using fspecial3 and imfilter:

```
>>h = fspecial3('gaussian',9);
>>filteredV = imfilter(noisyV,h);
```

And display the result:

Noisy data

Filtered data

③ Compute the gradients and magnitudes

a) The gradients and magnitudes of clean data V as ground-truth;

b) The gradients and magnitudes of noisy data noisyV;

c) The gradients and magnitudes of filtered data filtered;

Table2. The mean magnitudes on the Gaussian filtered temperature data. Note that the mean magnitudes on the temperature data is 0.2714

	Size=5	Size=9	Size=13	Size=19	Size=25
Sigma=0.65	0.1952	0.1952	0.1952	0.1952	0.1952
Sigma=1	0.1736	0.1724	0.1724	0.1724	0.1724
Sigma=1.5	0.1614	0.1532	0.1529	0.1529	0.1529
Sigma=1.9	0.1575	0.1430	0.1411	0.1410	0.1410
Sigma=2.4	0.1551	0.1350	0.1297	0.1289	0.1289
Sigma=3	0.1537	0.1297	0.1203	0.1171	0.1169
Sigma=3.4	0.1531	0.1276	0.1161	0.1108	0.1102

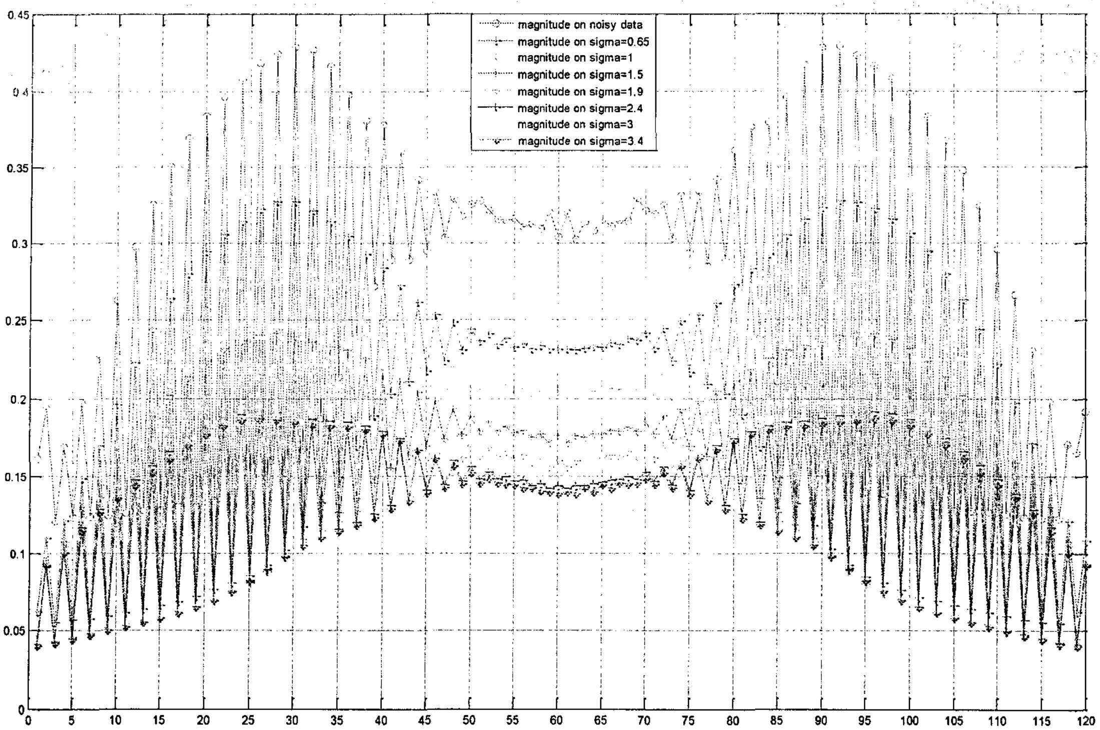


Fig3. The parts of magnitudes

The data in Fig.3 are obtained via the following

```
for i=1:120
v7(i)=sum(Vmag29((i-1)*b+1:i*b))/b;
end
```

Observations in this experiment are totally similar with the conclusion of last experiment.

Conclusion:

- Gaussian filter has a positive influence for computing gradient by smoothing discrete data
- Especially, it is efficient for noisy data

Some thoughts:

1. Gaussian filter is necessary for really generated discrete data. The fact is that can smooth efficiently the noisy data. Experimental results have been proven this conclusion. However, pay attention to the proper window size and sigma values to avoid the over-fitting smoothing.

2. The measurement used to distinguish the difference of two set gradients from different two methods. Here, magnitude of gradient is merely measurement. Under statistical sense, mean magnitudes are used to compare the experimental results.

3. Some details about data should be known, for example, the spacing of data will influence greatly the accuracy of gradient. $[f_x, f_y, f_z] = gradient(f, h_x, h_y, h_z)$