Manual of Kirchhoff depth migration on GPU by overlapping ray tracing and imaging kernel

Contents

1.Introduction

This code is designed to run the Kirchhoff depth migration on GPU, in that the ray tracing and imaging kernel can be ran overlapped to save more time, you can download this code and use it after a little adjustment to computer environment.

2.Computer operation system and hardware

(1). The user needs a computer with GPU that supports CUDA with computer ability >3.0

(2).This code run on Linux operation system, the original codes were compiled on Centos 6.4, so we highly recommend that the user run this code on the same level system or some higher version. The CUDA version is CUDA 9.0 and the Seismic Unix version is 4.3.

3. How to compile the code

(1) down load, the code and examples, unzip it with

```
tar –zxvf kpsdm.tgz
```

(2) go to the src directory, change some include and library detail in Makefile for the CUDA and Seismic Unix, then run: make in the command line.

(3). Go to the examples director , run the code by

```
./cudakpsdm par=mar.par
```

(4). The imaging result is marmig.0, this is a sub file, you can display by :

```
suximage < marmig.0 perc=99
```

4. Parameters introduction

(1) .The user can put all the parameters in a *.par file, like outflag=1. The necessary parameters are as follows:

Table 1. the parameters of this program

Number	Parameters	Introduction of this parameters
1	Infile	su format seismic data for imaging
2	project	directory of this imaging project
3	status name	tell the status when running
4	inner	number of traces in the infile
5	slineXY	where the subline is : 1 in x direction, 2 in y direction
6	outflag	1: output stack 2. Output gather
7	weightflag	if with amplitude preserve :1 yes 0 : no
8	mute	if with mute ,please write 0
9	vpickf	velocity file ,format introduced below
10	TTTdir	where the ray path files in
11	TTToldnew	if ray tracing with new velocity,1: no 2 : yes

12	gridfile	geometry of this project ,introduced below
13	vsubmin	min velocity subline
14	vsubmax	max velocity subline
15	vsubinc	interval of subline velocity
14	vcrsmin	min velocity crsline
15	vcrsmax	max velocity crsline
16	vcrsinc	interval of crsline velocity
17	submin,sbumax,subinc	imaging region of subline
18	crsmin,crsmax,crsinc	imaging region of crsline
19	off=	output offset bin if outflag=2
20	maxoff	maximum offset in su infile
21	daperture , xaperture	provide the aperture of imaging process
22	maxapertue	the maximum aperture
23	nz	imaging point in depth
24	dz	interval of imaging point in depth
25	fz	first imaging point in depth
26	device_list	the GPU number of running the GPU code
27	verbose	show information when running
28	maxmem	maximum memory of used GPU

(2) The velocity format

The velocity format should be like this:

```
HANDVEL    cdpnmber     subline    crsline
  5(depth)    3000(velocity)    10    32000
HANDVEL    cdpnmber2     subline2    crsline2
  5(depth)    3200(velocity)    10    32000
```

(3) The Geometry format(take example of 2D marmousi)

1	1	minimum subline of project
2	10	maximum subline of project
3	1	increased of subline
4	1	minimum of crstline
5	737	maximum of crsline
6	1	increased of crsline
7	0	x coordinate of minimum subline and crsline points
8	0	y coordinate of minimum subline and crsline points

9	0	subline increasment in x direction
10	12.5	subline increasment in ydirection
11	12.5	crsline increasment in x direction
12	0	crsline increase in direction

5. Run the examples

The examples are to test the code by Marmousi model, you can run the code after compiling the source code. Goto the examples directory and run :

```
 ./ cudakpsdm par=mar.par
```

After finished, you can display by :

```
suximage < marmig.0 perc=99
```

you can change the parameters to further test of this code.