Weekly Report

LIU Honghao

March 11 2021

1 Intro

Inverse discrete fourier transform

$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) \exp\left[2\pi i \left(\frac{xu}{M} + \frac{yv}{N}\right)\right] \tag{1}$$

In the radio astronomical imaging application, the u and v are not integrals. Therefore, there is no M or N, just total number of visibility - K. Given a pair of values - (u,v), there is a corresponding value for visibility. Then the new equation is

$$f(x,y) = \frac{1}{K} \sum_{u} \sum_{v} F(u,v) \exp\left[2\pi i \left(\frac{xu}{max(u)} + \frac{yv}{max(v)}\right)\right] \tag{2}$$

Convert equation 2 into matrix format.

$$\omega(u_j) = \frac{2\pi i u_j}{max(u_j)}, \omega(v_j) = \frac{2\pi i v_j}{max(v_j)}$$

Suppose the number of pixels of images is P.

$$U_p = \begin{bmatrix} \omega(u_0) * p \\ \omega(u_1) * p \\ \omega(u_2) * p \\ \vdots \\ \omega(u_k) * p \end{bmatrix} \qquad V_p = \begin{bmatrix} \omega(v_0) * p \\ \omega(v_1) * p \\ \omega(v_2) * p \\ \vdots \\ \omega(v_k) * p \end{bmatrix} \tag{3}$$

$$\Omega(u) = \begin{bmatrix} U_0 & U_0 & \dots & U_0 & U_1 & U_1 & \dots & U_1 & \dots & U_P \end{bmatrix} \tag{4}$$

$$\Omega(v) = \begin{bmatrix} V_0 & V_1 & \dots & V_P & V_0 & V_1 & \dots & V_P & \dots & V_P \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} f_{(0,0)} \\ f_{(0,1)} \\ f_{(0,2)} \\ \vdots \\ f_{(P,P)} \end{bmatrix} = \begin{bmatrix} F_0 & F_1 & F_2 & \dots & F_k \end{bmatrix} \cdot \exp\left[\Omega(u) + \Omega(v)\right] \tag{6}$$

Number of U_p or V_p is P in $\Omega(u)$ or $\Omega(v)$.

2 Memory issue

The global memory of Tesla V100-SXM2 is 32GB. In 32 bit machine, float and double are 4 bytes and 8 bytes respectively. The sizes double for complex number.
Suppose that the number of visibility is roughly equal to that of pixels.

precision	image size	memory size
float	1024 * 1024	8TB
double	1024 * 1024	16TB
float	512 * 512	2TB
double	512 * 512	16TB

Table 1: Caption

The memory to store $\Omega(u)$ with complex type will exceed the maximum memory.

3 Calculating piece of f

$$U = \begin{bmatrix} U_1 & U_1 & \dots & U_1 \end{bmatrix} \tag{7}$$

$$V = \begin{bmatrix} V_0 & V_1 & \dots & V_P \end{bmatrix} \tag{8}$$

The formula to calculate the ith row of the image which means $f(i,0), f(i,1), f(i,2) \cdots f(i,P)$ is

$$\begin{bmatrix} f_{(i,0)} \\ f_{(i,1)} \\ f_{(i,2)} \\ \vdots \\ f_{(i,P)} \end{bmatrix} = \begin{bmatrix} F_0 & F_1 & F_2 & \dots & F_k \end{bmatrix} \cdot \exp\left[iU + V\right] \tag{9}$$

The memory to store U, V with real type and temp_result with complex type is showed in table 2. The U and V can not be freed because they will be used in next round.

precision	image size	memory size
float	1024 * 1024	16GB
double	1024 * 1024	32GB
float	512 * 512	4GB
double	512 * 512	8GB

Table 2: Caption

To calculate multiple rows if the memory is efficient, eg. 2

$$\begin{bmatrix} f_{(i,0)} \\ f_{(i,1)} \\ f_{(i,2)} \\ \vdots \\ f_{(i+1,P)} \end{bmatrix} = \begin{bmatrix} F_0 & F_1 & F_2 & \dots & F_k \end{bmatrix} \cdot \exp\left[iU * [iI \ \ (i+1)I] + V * [I \ \ I]\right] \tag{10}$$

Unit matrix I can be stored in sparse matrix which takes less memory and can be ignored.

3.1 Update

Just saving V eq. 8 and U_1 eq. 3, because in U all the information are redundant, the calculation of U is meaningless.

4 Questions

4.1 Grid and block setting

Maximum number of threads per multiprocessor: 2048
Maximum number of blocks per multiprocessor: 32
Maximum number of thread per block: 1024
Maximum sizes of each dimension of a block: 1024 x 1024 x 64
Maximum sizes of each dimension of a grid: 2147483647 x 65535 x 65535