Using Futhark for a fast, parallel implementation of forward and back projection in algebraic reconstruction methods - A pre-study

Lærke Pedersen and Mette Bjerg Lindhøj

University of Copenhagen

08/11/2018

SIRT

Solve the problem:

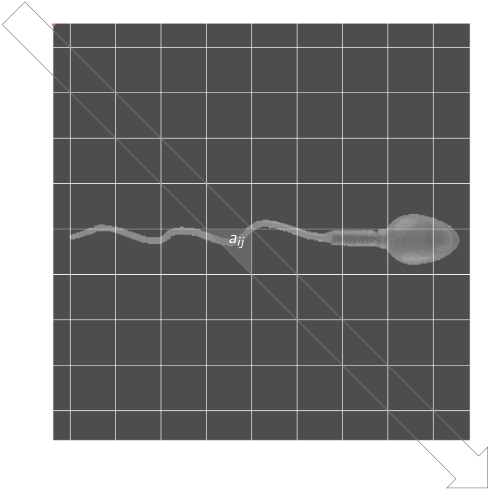
$$\boldsymbol{f}^* = argmin_{\boldsymbol{f}} \|\boldsymbol{p} - \boldsymbol{A}\boldsymbol{f}\| \tag{1}$$

iteratively using this update step:

$$\boldsymbol{f}^n = \boldsymbol{f}^{(n-1)} + \boldsymbol{C}\boldsymbol{A}^T\boldsymbol{R}(\boldsymbol{p} - \boldsymbol{A}\boldsymbol{f}^{(n-1)}), \tag{2}$$

where $\boldsymbol{C}$ and $\boldsymbol{R}$ are the diagonal matrices containing the inverse column and row sums of the system matrix respectively.

The system matrix



The problem is in the size

- Consider reconstructing a single slize of a volume from a detector of size $n \times n$
- The number of rays is n
- The number of angles is $\frac{n \cdot \pi}{2}$
- A typical value for n is 2048
- In semi sparse format the matrix will take up $2 \cdot 4 \cdot 2048 \cdot \lceil \frac{2048 \cdot \pi}{2} \rceil \cdot (2 \cdot 2048 - 1) \approx 216 GB$

System matrix computation

```
for ray = 0; ray < numberofrays; ray++ //parallel
  for pixel = 0; pixel < pixels; pixel++ //parallel
    if ray intersects pixel:
        (p1,p2) = intersectionpoints pixel ray
        A[ray][pixel] = distance p1 p2
```

Figure: $W(r, n) = O(r \cdot n^2)$, $D(r, n) = O(1)$

```
for ray in rays //parallel
  while (isingrid focuspoint) //seq
        pixel = calculatepixel focuspoint ray
        nextpoint = findnextpoint focuspoint ray
        A[ray][pixel] = distance nextpoint focuspoint
        focuspoint = nextpoint
```

Figure: $W(r, n) = O(r \cdot n)$, $D(r, n) = O(n)$

Flattening

Preliminiary results - not good but also we weren't completely done. Also flattening takes up even more space!

Code transformations

- By dumping the compiled code with the –dump command and getting the time spend in different kernels with the -D option we found that most of the time was spend on calculating the system matrix
- Futhark can not merge a map with a loop
- The code had a lot of branching
- We tried to mitigate this by removing as much from the loop as possible, and reorganising branches.

Changing the algorithm

```
for ray in rays //parallel
  for i=-halfsize; i < halfsize; i++ //parallel
        (l1,pixel1) = intersection1 ray i
        (l2,pixel2) = intersection2 ray i
        A[ray][pixel1] = l1
        A[ray][pixel2] = l2
```

Figure: $W(r, n) = O(r \cdot n)$, $D(r, n) = O(1)$

Flattening the new algorithm

Show them how its done ;-)

Hao Gao.
Fast parallel algorithms for the x-ray transform and its adjoint.
Medical physics, 39(23127102):7110–7120, November 2012.

Y. Long, J. A. Fessler, and J. M. Balter.
3d forward and back-projection for x-ray ct using separable footprints.
IEEE Transactions on Medical Imaging, 29(11):1839–1850, Nov 2010.

F. Natterer.
The Mathematics of Computerized Tomography.
Society for Industrial and Applied Mathematics, 2001.

Z. Xue, L. Zhang, and J. Pan.
A new algorithm for calculating the radiological path in ct image reconstruction.
In *Proceedings of 2011 International Conference on Electronic Mechanical Engineering and Information Technology*, volume 9, pages 4527–4530, Aug 2011.