BONE ANISOTROPY MAPPING

Jarunan Panyasantisuk, Joao Rivera, Rajan Gill, Ryan Cherifa

Department of Computer Science
ETH Zürich
Zürich, Switzerland

The hard page limit is 6 pages in this style. Do not reduce font size or use other tricks to squeeze. This pdf is formatted in the American letter format, so may look a bit strange when printed out.

ABSTRACT

Describe in concise words what you do, why you do it (not necessarily in this order), and the main result. The abstract has to be self-contained and readable for a person in the general area. You should write the abstract last.

1. INTRODUCTION

Bone fabric anisotropy or microstructure orientation was recently included in finite element (FE) models to improve the accuracy in predicting bone stiffness and strength. An FE model of bone is preferrably generated from a low resolution image which saves computing cost. Such a scanned image with a low resolution of 1-3mm can be obtained by widely-available clinical computer tomography (CT). However, the bone microstructure details can be obtained only by high resolution peripheral CT with the resolution of 60-82 μm. Bone anisotropy mapping methodology includes coordinates mapping between a low and a high-resolution images, region extraction, the mean intercept length (MIL) method for quantification of the microstructure orientation, ellipsoid fitting of MIL and eigendecomposition to obtain the major direction of the microstructure.

Motivation. In the recent study, bone anisotropy mapping algorithms are performed for all low resolution image voxels and for all pairs (n=71) of low and high resolution images. This preprocessing step consumes a significant amount of computing time. Moreover, researchers expect larger dataset to create a more general FE models of bone. Therefore, the performance of these algorithms are crucial. Two software packages for image processing which include MIL calculation are Medtool, a commercialized PYTHON package, and BoneJ, an open-source JAVA plugin for ImageJ. The external packages needed to be integrated to the computation pipeline and optimization is not straightforward.

In this paper, we are presenting an integrated and optimized methodology of bone anisotropy mapping. At our best knowledge, this is the first paper to optimize the performance of MIL calculation.

2. BACKGROUND

The methodology is shown in Fig. 1. Coordinates from a low resolution image were mapped to its own high resolution image. Then, a sphere region is extracted and centered at the mapped coordinate in the high reoslution image. Subsequently, the anisotropy of the extracted bone region is quantified by using the mean intercept length (MIL) method which imposed direction vectors on the regions. The mean length of each vector is the sum of the length inside the bone region devided by the number of intercepts which intersect with bone/non-bone transition. The MIL values can be plots as a cloud of points in the direction vector space and an ellipsoid can be fitted to obtain a representative two dimensional tensor, for which three eigenvalues and three eigenvectors are calculated. The eigenvector associated with the minimum eigenvalue is the major direction of that bone region.

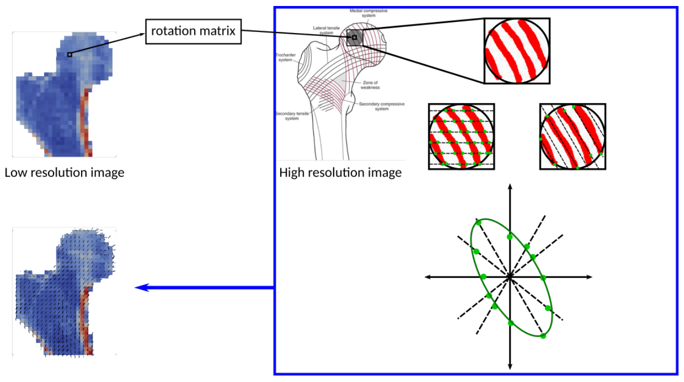

Fig. 1. Methodology

Region extraction. Applying mask on the bone region requires a multiplication for each image voxel.

Table 1. Cost analysis

	flops	read/write [doubles]	Operational intensity [flops/double]
Region extractions	n^3	$3n^3$	1/3
MIL			
Ellipsoid fitting			

Mean Intercept Length (MIL) method.

The mean intercept length of the vector v can be expressed as:

$$MIL(v) = \frac{h}{C(v)} \tag{1}$$

where h is the summation of the length of all direction vectors and $C(v)$ is the number of intercepts of the vector v. The algorithm includes an addition and a comparison.

Ellipsoid fitting. algorithm ...

3. METHODS

Region extraction. Loop unrolling

MIL. Loop unrolling and scalar replacement, blocking, SIMD blocking.

Ellipsoid fitting. moving loop invariant code, SIMD

4. EXPERIMENTAL RESULTS

Here you evaluate your work using experiments. You start again with a very short summary of the section. The typical structure follows.

Experimental setup. Specify the platform (processor, frequency, cache sizes) as well as the compiler, version, and flags used. I strongly recommend that you play with optimization flags and consider also icc for additional potential speedup.

Then explain what input you used and what range of sizes. The idea is to give enough information so the experiments are reproducible by somebody else on his or her code.

Results.

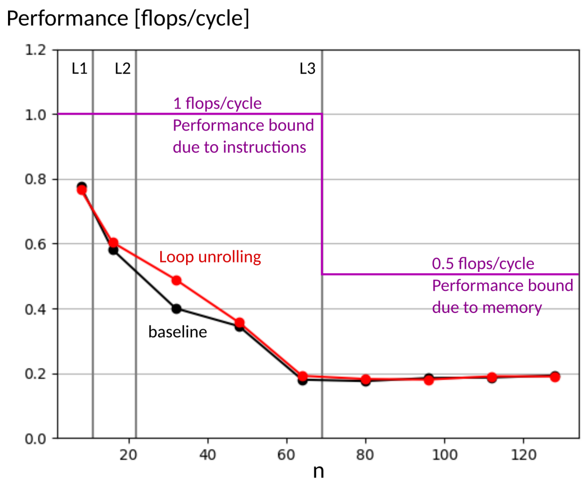

Fig. 2. Performance of region extraction

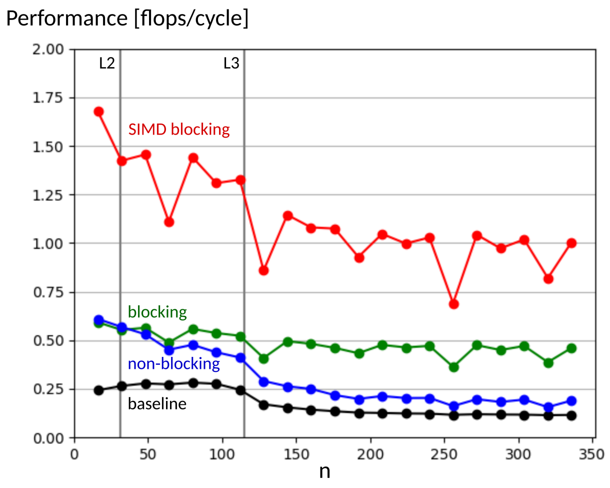

Fig. 3. Performance of MIL

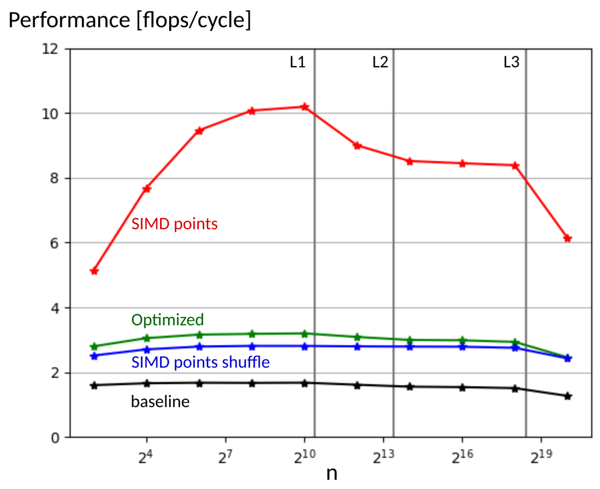

Fig. 4. Performance of ellipsoid

5. CONCLUSIONS

Here you need to briefly summarize what you did and why this is important. *Do not take the abstract* and put it in the past tense. Remember, now the reader has (hopefully) read the paper, so it is a very different situation from the abstract. Try to highlight important results and say the things you really want to get across (e.g., the results show that we are within 2x of the optimal performance ... Even though we only considered the DFT, our optimization techniques should be also applicable ....) You can also formulate next steps if you want. Be brief.

6. FURTHER COMMENTS

Here we provide some further tips.

Further general guidelines.

- For short papers, to save space, I use paragraph titles instead of subsections, as shown in the introduction.
- It is generally a good idea to break sections into such smaller units for readability and since it helps you to (visually) structure the story.
- The above section titles should be adapted to more precisely reflect what you do.
- Each section should be started with a very short summary of what the reader can expect in this section. Nothing more awkward as when the story starts and one does not know what the direction is or the goal.
- Make sure you define every acronym you use, no matter how convinced you are the reader knows it.

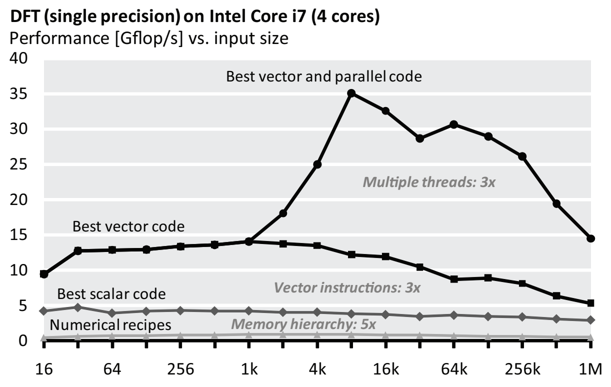

Fig. 5. Performance of four single precision implementations of the discrete Fourier transform. The operations count is roughly the same. *The labels in this plot are too small.*

- Always spell-check before you submit (to me in this case).
- Be picky. When writing a paper you should always strive for very high quality. Many people may read it and the quality makes a big difference. In this class, the quality is part of the grade.
- Books helping you to write better: [**?**] and [**?**].
- Conversion to pdf (latex users only):

  dvips -o conference.ps -t letter -Ppdf -G0 conference.dvi

  and then

  ps2pdf conference.ps

Graphics. For plots that are not images *never* generate (even as intermediate step) jpeg, gif, bmp, tif. Use eps, which means encapsulate postscript, os pdf. This way it is scalable since it is a vector graphic description of your graph. E.g., from Matlab, you can export to eps or pdf.

Here is an example of how to get a plot into latex (Fig. **??**). Note that the text should not be any smaller than shown.