

BONE ANISOTROPY MAPPING

Jarunan Panyasantisuk, Joao Rivera, Rajan Gill, Ryan Cherifa

Department of Computer Science
ETH Zürich
Zürich, Switzerland

ABSTRACT

Todo

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. To save computing cost, an FE model of bone is generated from a clinical computer tomography (CT) scanned image with low resolution (1-3mm). However, the bone microstructure details can be obtained only by high resolution peripheral CT with the resolution of 60-82 μm . Therefore, bone anisotropy mapping methodology was required to map the bone microstructure orientation from the high resolution image onto the low resolution image.

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 study and optimize the performance of MIL calculation depending on the extracted region size.

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 resolution 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 divided 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.

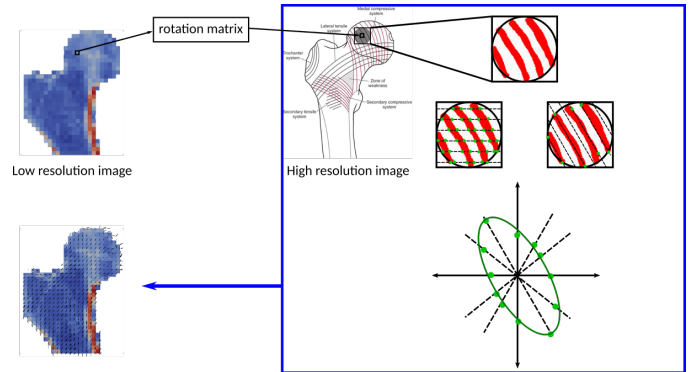


Fig. 1. Methodology

From the basic implementation, MIL calculation, ellipsoid fitting and region extraction consume approximately 80%, 15% and 5 % of the overall computing time, respectively. Therefore, we focused on these three algorithms for the optimization. The run time was measured by using time stamp counter (TSC) and the cost analysis of the three algorithm is shown in Table 1.

Region extraction. This algorithm applies a sphere mask

Table 1. Cost analysis

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

on the high resolution image and copies the extraction region to a separate array. A multiplication was performed for each image voxel.

Mean Intercept Length (MIL) method.

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

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

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. **Todo:** e.g. explaining about strides

Ellipsoid fitting. **Todo**

3. METHODS

Region extraction. Loop unrolling

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

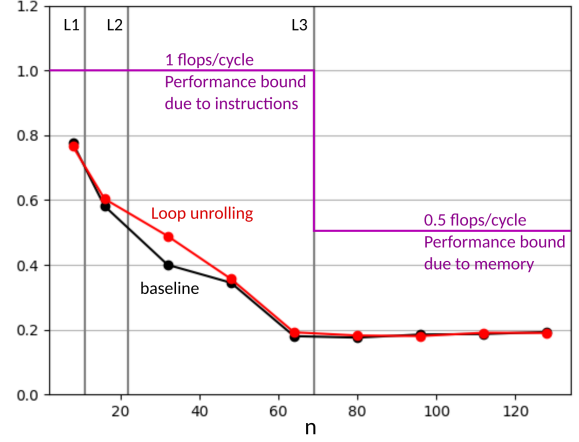
Ellipsoid fitting. moving loop invariant code, SIMD

4. EXPERIMENTAL RESULTS

The experimental results are presenting here according to the algorithms.

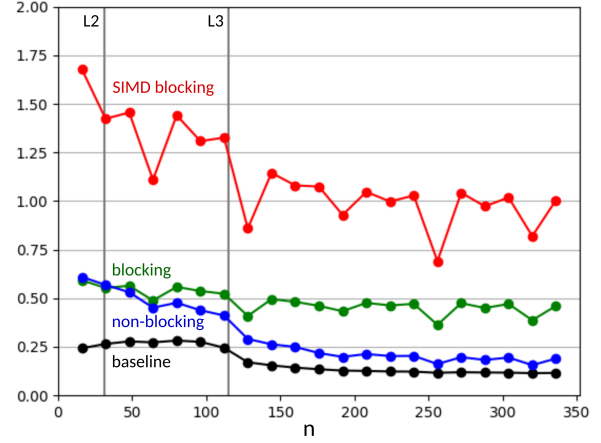
Region extraction. *Experimental setup* The tests were performed on Intel i7 U7600 (Kaby Lake), 2.8 GHz without TurboBoost. The L1, L2 and L3 cache size were 32KB, 256KB, and 8MB, respectively. The highest optimization flag was used without vectorization (`-O3 -fno-vectorize-tree`). The sphere region size were ranged from $n = 8^3$ to 128^3 .

Results As seen in Fig. 2, the loop unrolling method did not improve the performance for the region extraction algorithms.

Performance [flops/cycle]**Fig. 2.** Performance of region extraction

MIL calculation. *Experimental setup*

Results

Performance [flops/cycle]**Fig. 3.** Performance of MIL

Ellipsoid fitting. *Experimental setup*

Results

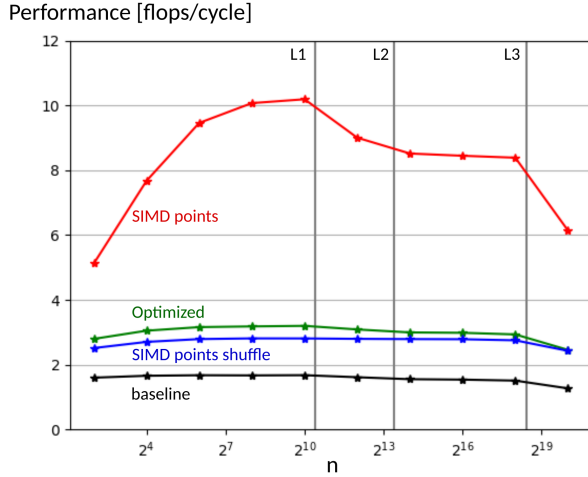


Fig. 4. Performance of ellipsoid

Overall performance.

5. CONCLUSIONS

The bone anisotropy mapping was integrated and optimized. MIL calculation consumes the large percentage of the overall computation time, followed by ellipsoid fitting and region extraction.

The performance of region extraction was bounded by the instructions mix when it was in cache and memory bound when in memory. The loop unrolling did not improve the performance and the compiler already optimized the alias checking in the basic implementation.

MIL calculation

Ellipsoid fitting

6. FURTHER COMMENTS

Here we provide some further tips.

Further general guidelines.