OpenCL based machine learning labeling of biomedical datasets

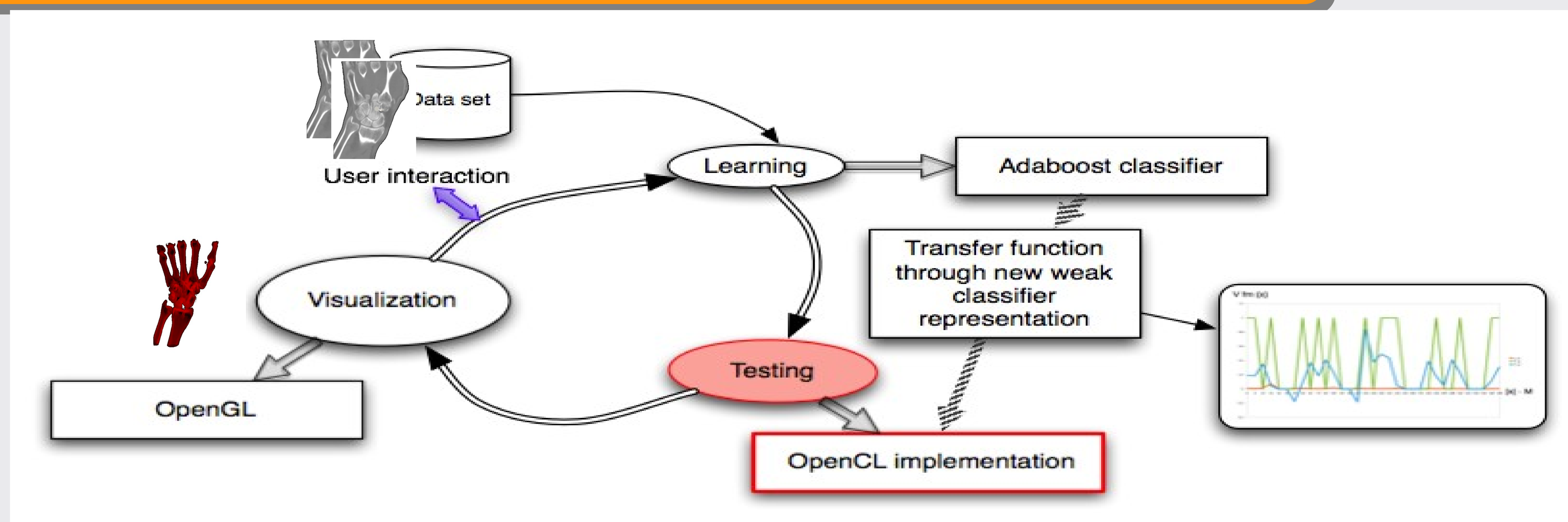
Oscar Amoros1, Sergio Escalera2 and Anna Puig3

1,2,3 Dept. Matemàtica Aplicada i Anàlisi, University of Barcelona, Gran Via 585, 08007, Barcelona, Spain
2 Computer Vision Center, Universitat Autònoma de Barcelona, 08193 Cerdanyola, Spain 3 WAI-Movibio Research Groups,
oamorohu7@alumnes.ub.edu, sescalera@cvc.uab.es, anna@maia.ub.es

Abstract

In medical imaging it becomes imperative to provide an automatic and interactive method to label or to tag different structures contained into input data. In this work, we propose an alternative representation of the Adaboost binary classifier. We use this proposed representation to define a new GPU-based parallelized Adaboost testing stage using OpenCL. We provide numerical experiments based on large available data sets and we compare our results to CPU-based strategies in terms of time and labeling speeds.

1.Overview



·The Adaboost procedure [1] trains the classifiers fm(x) on weighed versions of the training samples, giving higher weights to cases that are currently misclassified. For each fm(x) we just need to compute a threshold value and a polarity to make a binary decision, selecting that one that minimizes the error based on the assigned weights.

·This simple combination of classifiers has demonstrated to reduce the variance error term of the final classifier F (x).

·In Algorithm 2, we show the testing of the final decision function using the Discrete Adaboost algorithm with Decision Stump "weak classifier".

$$F(x) = \sum_1^M c_f f_m(x)$$

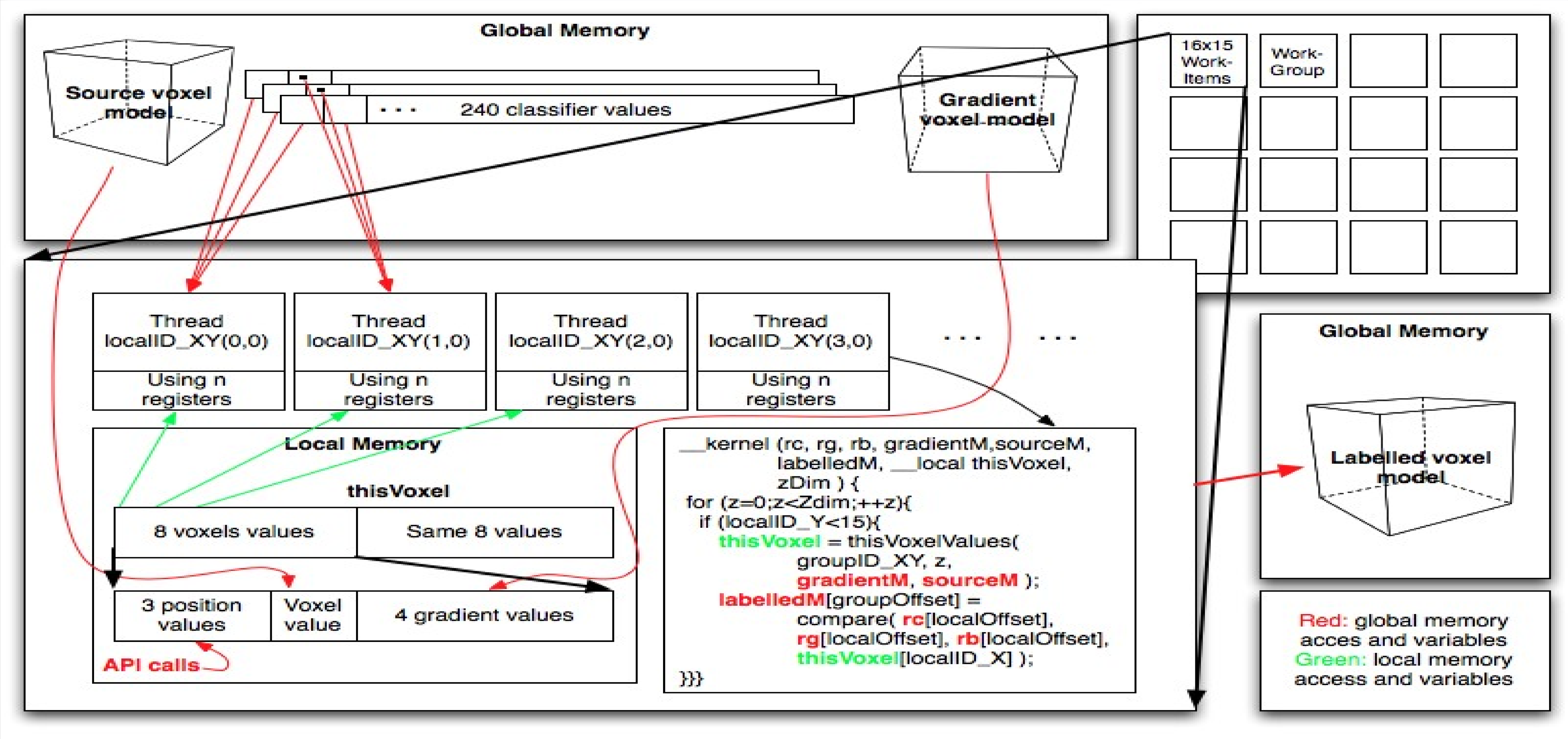
1: Start with weights $w_i = 1/N, i = 1, .., N$.
2: Repeat for $m = 1, 2, .., M$:
(a) Fit the classifier $f_m(x) \in -1, 1$ using weights w_i on the training data.
(b) Compute $\mathrm{err}_m = E_w[1_{(y \neq f_m(x))}]$, $c_m = \log((1 - \mathrm{err}_m)/\mathrm{err}_m)$.
(c) Set $w_i \leftarrow w_i \exp[c_m \cdot 1_{(y_i \neq f_m(x_i))}]$, $i = 1, 2, .., N$, and normalize so that $\sum_i w_i = 1$.
3: Output the classifier $\mathrm{sign}[\sum_{m=1}^M c_m f_m(x)]$.

Algorithm 1: Discrete Adaboost training algorithm.

1: Given a test sample x
2: $F(x) = 0$
3: Repeat for $m = 1, 2, .., M$:
(a) $F(x) = F(x) + c_m(P_m \cdot x^m < P_m \cdot T_m)$;
4: Output $\mathrm{sign}(F(x))$

Algorithm 2: Discrete Adaboost testing algorithm.

2. OpenCL implementation



- The eight features considered at each sample by our binary classifier are: the spatial location (x, y, z), the sampled value (v), and its associated gradient value (gx, gy, gz, |g|).
- Our binary classifier, for each feature, we have a total of N possible Cm values, with N = 3 ·M .
- We create a matrix of Work-Groups that covers the x and y size of the dataset fitted into GPU global memory, whereas the component z is computed in a inner loop at each kernel. Each WorkGroup classifies one voxel. Inside each Workgroup, we define N 8 threads (or WorkItems). Each thread computes a single operation with the 3 channels or weights of the weak classifier. These N 8 values will be reduced at the end of the execution and compared to a reduced addition. The final label at each voxel is directly computed by this comparison.

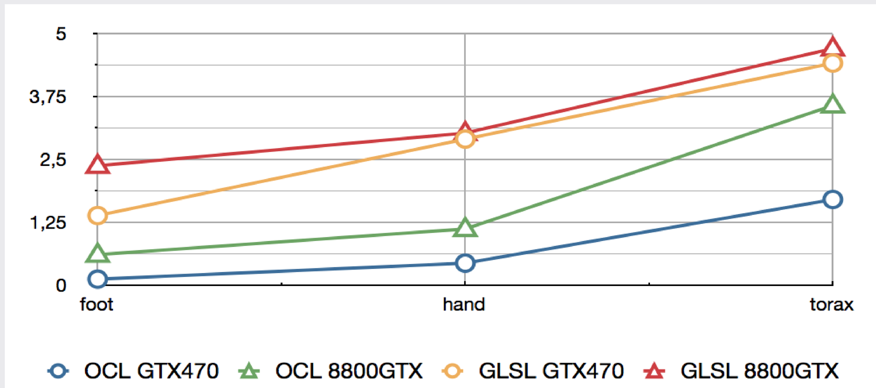
3. Simulations and Results

Dataset	Size	Mathlab	CPU-based	OpenMP	GLSL	OpenCL
Foot	128x128x128	18.32s	9.63s	8s	1.32s	0.1256s
Hand	244x124x257	67.29s	26s	20s.	2.86s	0.1653s
Thorax	400x400x400	114.28s	33.76s	25s	4.41s	1.9253s

Table 1. Testing step times in seconds of the different datasets with the five implementations. GLSL and OpenCL times has been obtained using the GTX470 graphic card.

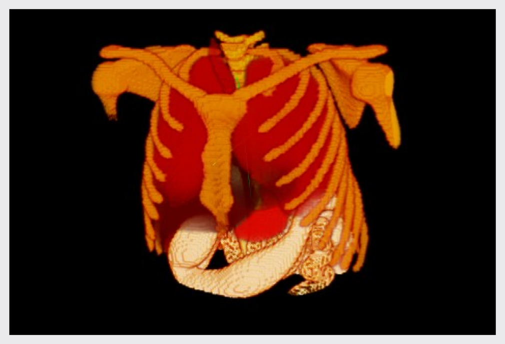
- We measure the performance in terms of the mean execution time from 500 code runs on the same hardware.

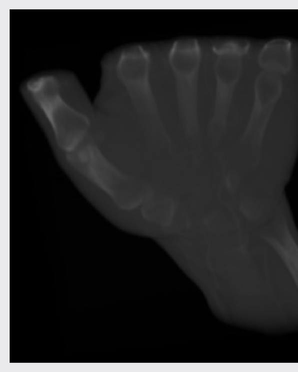
- In Table 1, we show the averaged times of the five implementations with the different sized datasets. Our proposed OpenCL-based optimization has a speed up of 89.91x over a C++ CPU-based algorithm and a speed up of 8.01x over the GLSL GPU-based algorithm.



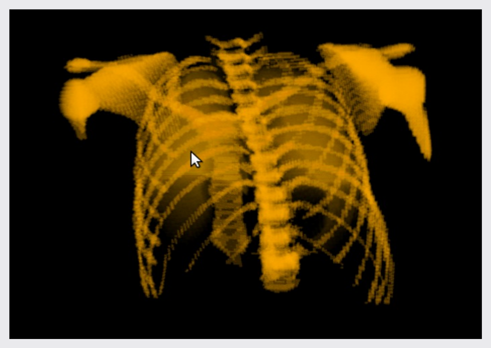
At hardware level there are tree main differences between the 8800GTX and GTX470 cards:
1. The number of Compute Units (CU), 16 for the 8800GTX in contrast to 14 for the GTX470,
2. The number of Processing Elements (PE) 8 for the 8800GTX and 32 for the GTX470,
3. The L1 and L2 caches only present in the Fermi architecture (GTX470).

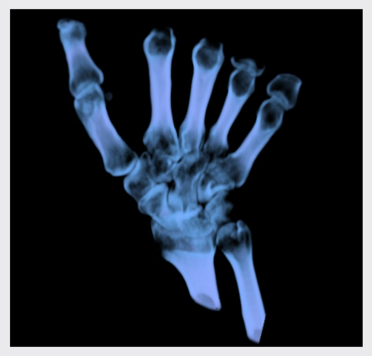
Datasets without classification





Classified datasets for bone visualization





4. Conclusions and future work

- Our representation of the weak-classifiers guarantees the equivalence to the classical approach.
-Our breakthrough is a proved optimization of the labeling process involved in several biomedical applications. This optimization increases the interactively of these processes, and also can be easily integrated in the clinical routine.
- Even with the minimum global memory use, we achieve a good improvement in performance, so we expect real time testing when integrating it with OpenGL visualization, and implementing the global memory optimizations.
- Future work includes optimizations, Multiclass testing, GPU learning stage, interactive inteface and testing more maintainable and portable language ***StarSs*** *[2].*

Bibliography:

[1] Yoav Freund, Robert E. Schapire. "A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting", 1995.

[2] http://www.prace-project.eu/documents/08_starss_jl.pdf

ACKNOWLEDGMENTS
This work has been partially funded by the project CICYT TIN2008-02903, by the research centers CREB of the UPC and the IBEC and under the grant SGR-2009-362 of the Generalitat de Catalunya.
This work has also been supported in part by projects TIN2009-14404-C02, CONSOLIDER INGENIO CSD 2007-00018, and the CASE department of Barcelona Supercomputing Center.