ADABOOST GPU-BASED CLASSIFIER FOR DIRECT VOLUME RENDERING

Oscar Amoros[1], Sergio Escalera[2] and Anna Puig[3]

1,2,3 Dept. Matemàtica Aplicada i Anàlisi, University of Barcelona, Gran Via 585, 08007, Barcelona, Spain

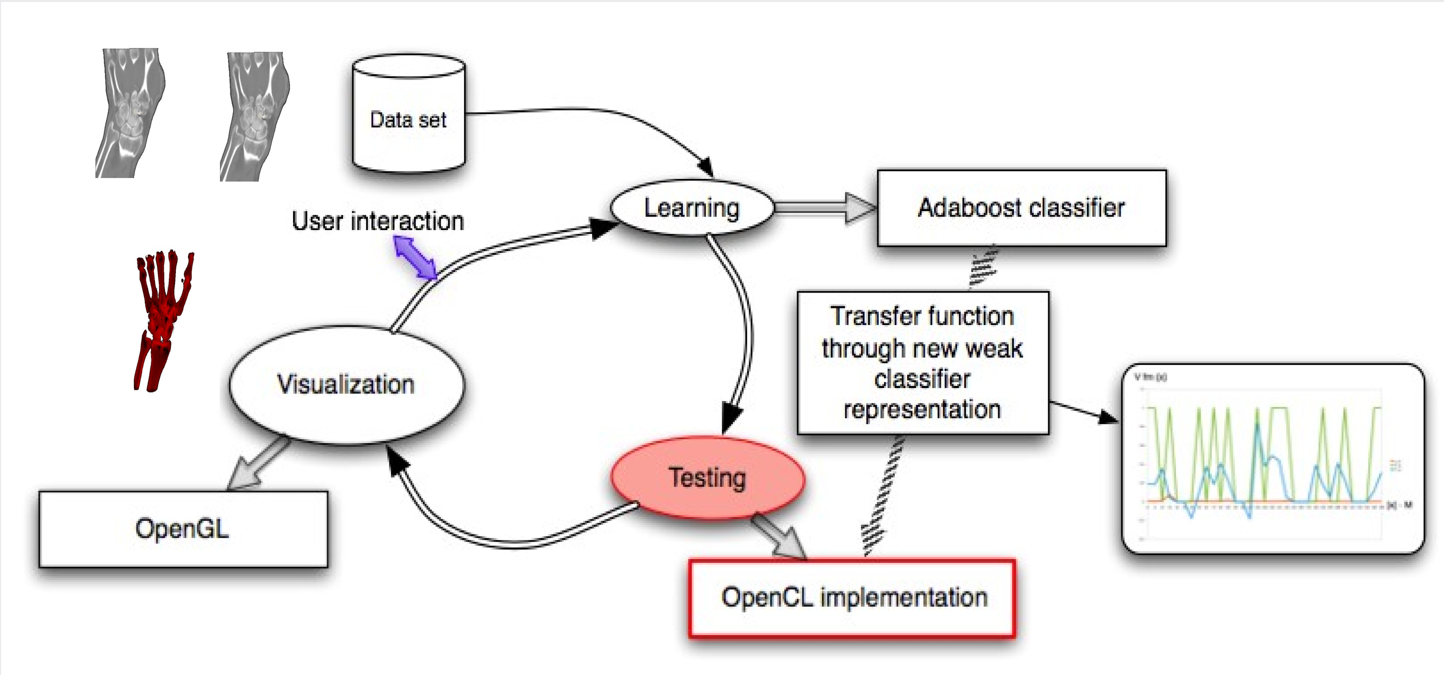
2 Computer Vision Center, Universitat Autònoma de Barcelona, 08193 Cerdanyola, Spain 3 WAI-Movibio Research Groups,

oamorohu7@alumnes.ub.edu, sescalera@cvc.uab.es, anna@maia.ub.es

Abstract

In volume visualization, the voxel visibility and materials are carried out through an interactive editing of Transfer Function.We present a two-level GPU-based labeling method that computes in times of rendering a set of labeled structures using the Adaboost machine learning classifier. In a pre-processing step, Adaboost trains a binary classifier from a pre-labeled dataset and, in each sample, takes into account a set of features. Then, at the testing stage, each weak classifier is independently applied on the features of a set of unlabeled samples. We propose an alternative representation of these classifiers that allow a GPU-based parallelized testing stage embedded into the visualization pipeline.

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).

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

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

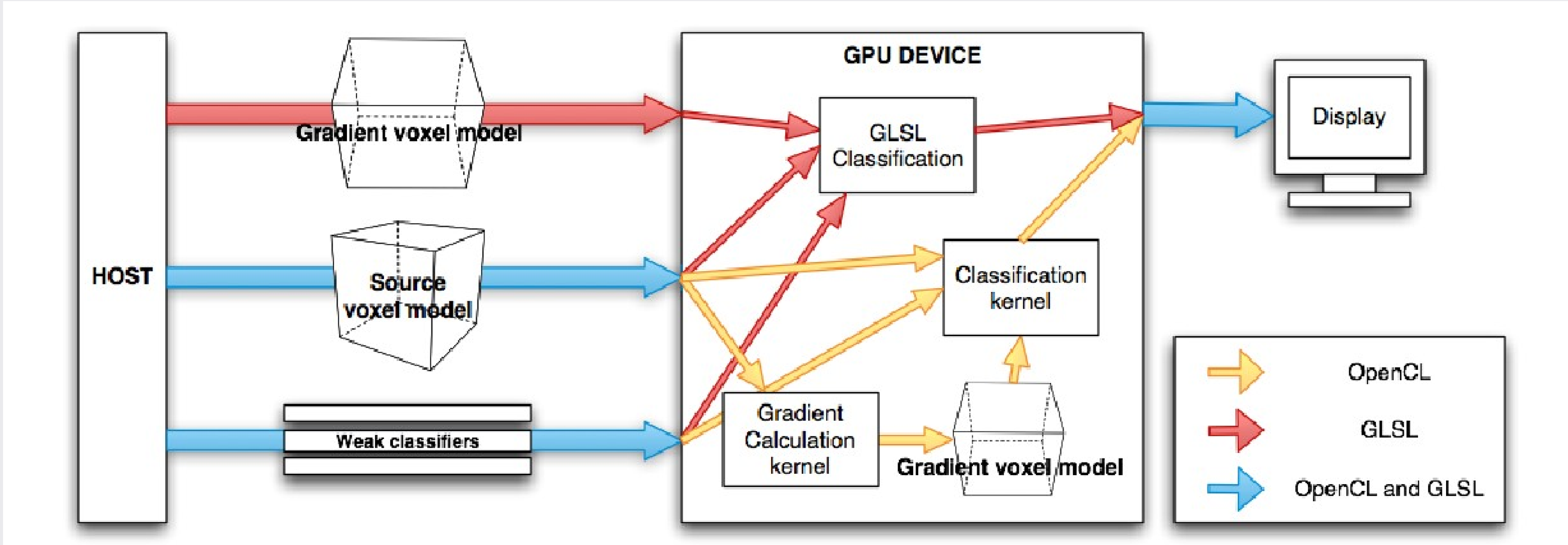
1: Start with weights $w_i = 1/N, i = 1,..,N$.
2: Repeat for $m = 1, 2, .., M$:
(a) Fit the classifier $f_m(x) \in -1, 1$ using weights w_i on the training data.
(b) Compute $\mathrm{err}_m = E_w[1_{(y \neq f m(x))}]$, $c_m = \log((1 - \mathrm{err}_m)/\mathrm{err}_m)$.
(c) Set $w_i \leftarrow w_i \exp[c_m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

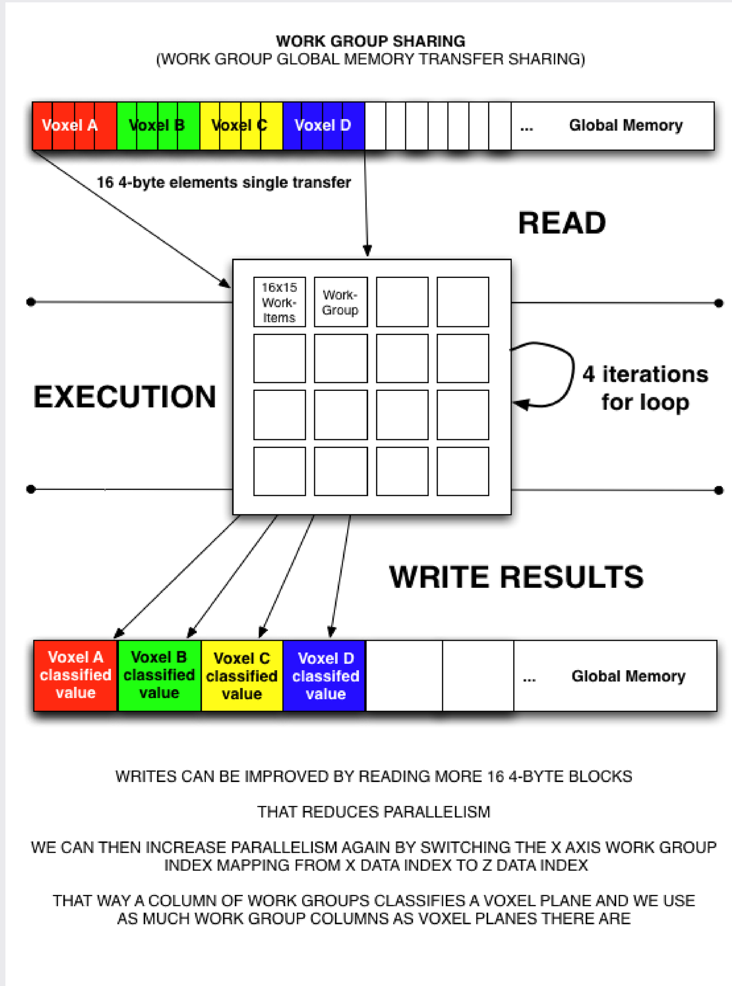
Algorithm 2: Discrete Adaboost testing algorithm.

2. GPU-based implementation

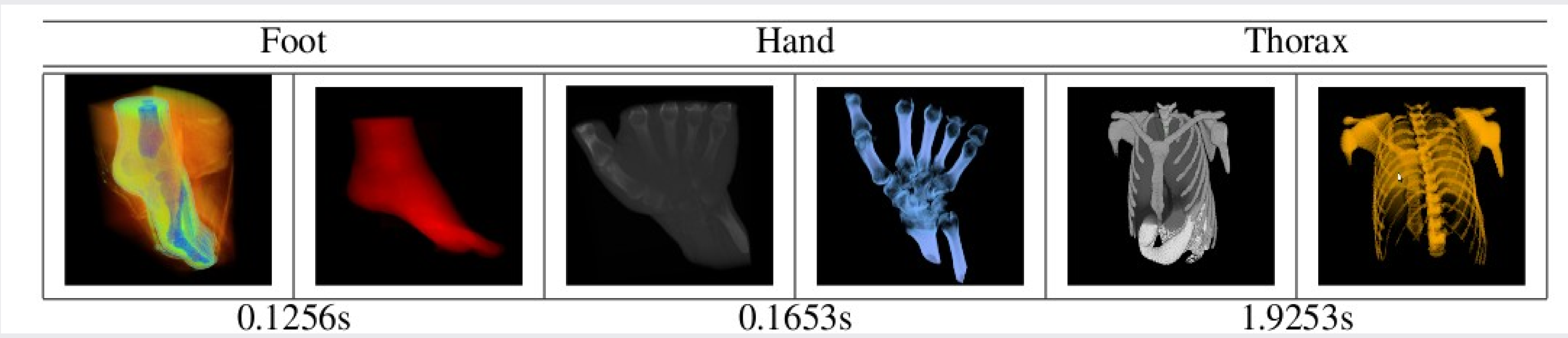


OpenCL kernels substitutes GLSL shaders, and offers a deeper hardware control that translates in a much faster execution. Thanks to OpenCL-OpenGL interoperability we can exploit the OpenCL potential in the classification-visualization process.

Introducing Work Group Sharing. In order to ensure code scalability for future GPU devices and ensuring maximum global memory bandwidth at the same time, we deeply parallelized the classification step using up to 240 Work Items in a Work Group to classify one voxel, and used that Work Group four times to classify 4 voxels, and read all the data for the 4 in one memory transaction.



3. Simulations and Results



Dataset	Size	Features	Weak classifiers	Accuracy	Learning	Testing (GPU)
Foot	128x128x128	Bones and Soft tissue	1	99.95%	2.3s	0.0461s
Foot	128x128x128	Finger's bone	8	99.89%	11.45s	0.1567s
Foot	128x128x128	Ankle's muscle	7	99.21%	10.01s	0.1611s
Thorax	400x400x400	Vertebra and Column	3	99.01	3.2s	0.7157s
Thorax	400x400x400	Bone and lungs	30	84.15%	33.14s	1.9253s
Thorax	400x400x400	Bone and liver	30	78.28%	32.8s	1.9154s
Hand	244x124x257	Bone	1	100%	2.8s	0.1653s

Dataset	Size	Matlab	CPU	OMP	GLSL	OpenCL
Foot	128x128x128	18.32s	9.63s	8s	1.32s	0.12s
Hand	244x124x257	67.29s	26s	20s.	2.86s	0.16s
Thorax	400x400x400	114.28s	33.76s	25s	4.41s	1.92s

4. Conclusions and future work

- We presented an alternative approach in medical classification that allows a new representation of the Adaboost binary classifier.
- We also defined a new GPU-based parallelized Adaboost testing stage using a OpenCL implementation integrated to the rendering pipeline.
- We used state-of-the-art features for training and testing different datasets. The numerical experiments based on large available data sets and the performed comparisons with CPU-implementations show promising results.

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.