

CLASSIFICATION OF BIOMEDICAL HIGH-RESOLUTION MICRO-CT IMAGES FOR DIRECT VOLUME RENDERING

Maite López-Sánchez, Jesús Cerquides, David Masip, and Anna Puig
WAI, Volume Visualization and Artificial Intelligence
MAiA Dept., Universitat de Barcelona
Gran Via de les Corts Catalanes, 585
Barcelona, Spain
{maite,cerquide,davidm,anna}@maia.ub.es

ABSTRACT

This paper introduces a machine learning approach into the process of direct volume rendering of biomedical high-resolution 3D images. More concretely, it proposes a learning pipeline process that generates the classification function within the optical property function used for rendering. Briefly, this pipeline starts with a data acquisition and selection task, it is followed by a feature extraction process, to be ended with sequence of supervised learning steps. Learning comprises Gentle Boost and CRF (Conditional Random Fields) classifiers. The process is evaluated in terms of accuracy and overlap metrics so that we can measure how performance increases along the whole pipeline process. Empirical results confirm that, even though the classification of high-resolution computerized tomography volume data poses a challenging problem for single-run classifiers, it can be significantly improved by subsequent learning steps and refinements.

KEYWORDS

Machine Learning, Biomedical 3D Images, Classification, CRF (Conditional Random Fields), GBP (Generalized Belief Propagation).

1 Introduction

Volume rendering has emerged as one of the most active fields in Scientific Visualization. It consists of rendering property values measured at points of a 3D volumetric region. One of the major applications of volume rendering is the visualization of biomedical data captured with 3D imaging devices such as high-resolution Computer Tomographies (CT) [1, 2]. In these applications, the 3D region is sampled according to a regular 3D grid, by parallel image planes. The representation of the volume is a voxel model consisting of a set of parallel cubical and face-adjacent cells called *voxels* with associated property values that usually correspond to *tissue density* scaled as an *intensity* level. A typical data set is composed by 512^3 voxels.

During rendering, the voxel model is traversed. The intensity is computed at a set of 3D positions in the volume called *rendering samples*. Every rendering sample is then shaded according to the lighting conditions and to the

optical properties of the anatomical structure to which they belong. Finally, rendering samples are ordered to compute the final 2D projection.

The definition of the optical properties can be viewed as an elicitation process which extracts user knowledge about anatomical structures and visualization preferences – such as the appearance of different tissue structures–. This elicitation process defines the Optical Property Function (OPF), which is a 3D continuous function defined for all spatial points (x, y, z) contained into the data voxel model to the optical properties, such as emission (R, G, B) and absorption (α) .

The elicitation process is often performed through the user definition of *transfer functions*. These functions directly associate optical properties to data values. Thus, the OPF is computed at each point as a mapping of its property value to the corresponding optical properties. During visualization, transfer functions can be stored as look-up tables, indexed by the intensity data values. This presents the advantage of speeding up rendering significantly. However, it has been proven [3] that only using intensity-based one-dimensional transfer functions fails at accurately detecting complex combinations of material boundaries. This problem becomes harder for rich textures in gray-scale images, where a given gray intensity value does not uniquely correspond to a specific anatomical structure, tissue or material. Multidimensional transfer functions –based on first and second derivatives of the intensity values– tend to improve classification but at the cost of increasing both complexity and memory requirements.

Transfer functions can be broken into two: the Classification Function (CF) and the Structure to Optical Properties Assignment Function (SOPAF). CF is a continuous function which determines, for each point inside the voxel model, the specific anatomical structure, tissue or material it belongs to. On the other hand, SOPAF assigns a set of optical properties to each anatomical structure, tissue or material. Therefore, the elicitation becomes a two-step process. Firstly, during the classification step, a labeled voxel model is created containing, for each voxel, a label of the region it belongs to. Secondly, this classified model is used, together with the original voxel model, to build a (R, G, B, α) model suitable for visualization.

During rendering, this CF is used to skip non-selected regions, thus reducing the cost of model traversal. The usage of an intermediate labeled model increases memory requirements. Nevertheless, since classification is carried on as a preprocess before rendering, it can cope with the usage of more complex and computationally expensive classification methods than transfer functions [4].

Although many papers in volume rendering literature address classification [5], most of them are based on the edition of transfer functions and on the design of user friendly interfaces for their specification. Some preliminary work based on learning methods have also been published: supervised methods such as bayesian networks [6], neural networks [7], decision trees [8] and non-supervised methods [9]. Additionally, in [10], clustering-based supervised and non-supervised learning methods are compared for the classification of magnetic resonance images (MRI).

In this paper, we address classification as a *machine learning* problem. We interpret voxels as objects to classify and extracted image features around them as the attributes to evaluate. In previous work ([11]) we evaluated the performance of several well-known attribute-value classifiers for the same problem. Here, we train a pipeline of classifiers with manually classified voxels and test its performance by classifying different ones. Our main goal is thus to define an CF in a way that is suitable for a posterior high-resolution rendering.

The rest of the paper is organized as follows. Next section presents the learning pipeline process, describing the machine learning techniques that we propose to apply for each step in the pipeline. Then, Section 3 characterizes the methodology used in the experiments and presents the obtained results. Finally, Section 4 concludes the paper.

2 Learning Pipeline Process

Generally speaking, machine learning processes involve both a training and a testing phases. Training is the most important and is structured in three main tasks:

- **Data Acquisition and Selection.** Initially, we must select a subset of our acquired data to be used for training purposes.
- **Feature Extraction.** Afterwards, this data must be characterized in a way that includes enough information to learn the classification function (its accuracy strongly depends on the input instance representation).
- **Learning.** Finally, a specific learning method is chosen –or defined– to learn the model and is subsequently applied to classify.

This section describes this training process providing the details of how can it be applied to large datasets such as high-resolution images.

First task, data acquisition and selection, depends on the algorithm learning type. We focus on supervised learning applied to 3D volume data classification, and therefore, training data is defined as a set of voxel-classification pairs, where voxels are provided together with their corresponding classification. Due to our experimental learning settings (see Section 3.1), provided voxels are required to define slices in the 3D image, so that they have an implicit neighborhood relation in two out of the three dimensions.

Second task corresponds to feature extraction. Our previous work ([11]) used classical features reported in the literature such as the 3D voxel position, its intensity value, and a gradient magnitude. Positions were considered because intensity values and gradients are not enough for characterizing complex gray-scale textures. Nevertheless, positions are not appropriate when textures do not correspond to anatomical structures but to materials or tissues whose global position could vary from image to image. Instead, we propose to use Gabor filters, a well established feature extraction method. Briefly, a Gabor filter can be defined as a linear filter whose impulse response is defined by a harmonic function multiplied by a Gaussian function [12]. Psycho-physical observations in textures conclude that histograms of a set of Gabor filters may be sufficient statistics in texture perception, i.e., two textures cannot be told apart in early vision if they share the same histograms of Gabor filters [13, 14]. Furthermore, Gabor filters allow to incorporate neighborhood information for each voxel without depending on global coordinates. Therefore, since voxels within a 3D image should not be considered to be independent from its neighbors, it is appropriated to use them in this feature extraction task.

The definition of the learning algorithm constitutes the main design decision in the learning process. In our context of high resolution 3D images, we assume that a few 2D slices can be manually classified by human experts, so that supervised learning becomes feasible. Among the plethora of supervised learning methods, we propose the use of discriminative models instead of generative models. The main reason is that generative models do have very strict independence assumptions on the data that can not be ensured in our image domain. In fact, we do not propose the use of a single learning method, but a process pipeline that comprises the following steps:

- **Attribute-value classifier.** As an initial learning process we propose the use of a Gentle Boost classifier [15]. Gentle boost is a boosting algorithm ([16]) that performs an ensemble of weak classifiers, typically using regression stumps, to minimize the training error. For our task, Gentle Boost provides good classification rates, but leaves spurious points that do not seem to comply with physical intrinsic locality.
- **Spatial coherence model.** A second learning step is then proposed to deal with classification locality. For this purpose, we use another discriminative model: Conditional Random Fields [17] (CRF, see next sub-

section 2.1), which assumes locality not only in the extracted features but also on the resulting classes. In this manner, a voxel surrounded by voxels of a given class is most likely to belong to this same class. Much like a Markov random field, a CRF is an undirected graphical model in which each vertex represents a random variable whose distribution is to be inferred, and each edge represents a dependency between two random variables. The distribution of each random variable in the graph is conditioned on the input image. Moreover, given this input data, CRF has a single exponential model for the joint probability of the entire mesh of class labels.

- **Refinement.** Although CRF is able to remove spurious classified voxels, the certainty of a class assignment is not always as high as desired, specially for those voxels laying on class boundaries. Therefore, our proposal is to refine the resulting classification by focusing only on class boundaries. In this manner, we propose to select those areas where voxels belong to different classes (or do have a weak membership certainty), to afterwards add its assigned classification as an extra feature, and, finally, to repeat the Gentle Boost learning method for these selected voxels. Refinement considers far less input data than the first learning process, so we consider refinement’s computational cost to be affordable regarding the obtained classification improvement (see Section 3).

2.1 Conditional Random Fields

As a conditional (discriminative) model, CRF specifies the probability of possible label sets given an observation set. It allows arbitrary, non-independent features on the observation set X .

Definition Let X be a random variable over the input data set to be labeled, and let Y be a random variable over the corresponding label sets. Let $G = (V, E)$ be a graph such that $Y = (Y_v)_{v \in V}$, so that Y is indexed by the vertices of G . Then (X, Y) is a conditional random field in case, when conditioned on X , the random variables Y_v obey the Markov property with respect to the graph: $p(Y_v | X, Y_w, w \neq v) = p(Y_v | X, Y_w, w \sim v)$, where $w \sim v$ means that w and v are neighbors in G .

In our case, we used a two dimensional grid as the graph that provides structure to the set of labels and observations.

Learning a CRF model [18] requires to determine the parameters $T = (\lambda_1, \lambda_2, \dots; \mu_1, \mu_2, \dots)$ from independently and identically distributed training data $D = \{(x^{(k)}, y^{(k)})\}$ with empirical distribution p over (x, y) . These T parameters correspond to the logistic regression parameters that maximize the overall probability distribution on the graph. In our settings, we propose to consider SG (Stochastic Gradient [19]) descent method to estimate T . Additionally, parameter estimation requires to

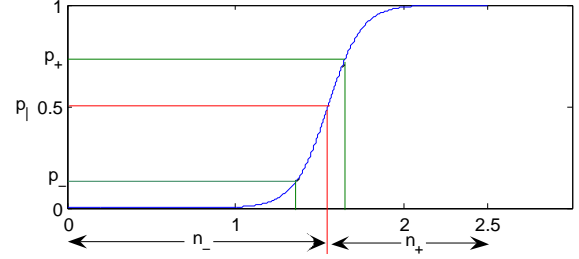


Figure 1. Illustration of boundary selection parameters.

be combined with an inference process. We propose to use the GBP (Generalized Belief Propagation [20], [21]) method for inference, because Belief Propagation fails to converge for two dimensional grids. Henceforth, once the CRF model has been learnt, this same inference method will be applied over it in order to classify (i.e., label) new incoming data.

2.2 Class Boundary Selection for the Refinement Step

Let V be the output of the spatial coherence model, which contains, for each voxel¹, a certainty measure about its classification. Assuming a problem with two classes (+, -), for each voxel we will have the probability of the voxel belonging to class + (which also implies the probability of belonging to class -).

V can be ordered considering an ascending order relation over the probability of belonging to class +⁽²⁾ (i.e., first voxel in V has the lowest probability of being class +). From here, we define two thresholds p_+ , p_- and classify each voxel into one of three sets (see Figure 1):

$$I_+ = \{x \mid p_x \geq p_+\}, I_- = \{x \mid p_x \leq p_-\}, I_B = \{x \mid p_- \leq p_x \leq p_+\}$$

I_+ and I_- correspond to the voxels with higher certainty of belonging to class + and class - respectively, and I_B can be seen as the boundary voxels, for which the classifier was less certain about its class. In fact, I_B provides us with the set of boundary voxels that require further refinement.

Thresholds p_+ and p_- are defined, in our case, as biased α -quantiles around p_1 , being p_1 equal to 0.5. Notice that p_1 provides us with a partition of V that separates the set of voxels having a probability of belonging to class + higher than $p_1 = 0.5$ from those voxels whose probability is lower than that.

Additionally, since the cardinality of I_B fixes the amount of voxels in the image to be refined, we need to parameterize it so we gain control over the computational resources to devote. Thus, we define α as the percentage (between 0 and 1) of voxels to be refined, and then, we define thresholds p_+ and p_- to be dependent on α so that we

¹Voxels here are treated as 2D pixels. We keep using this notation for coherence purposes only.

²The overall process presented here would be equivalent for class -

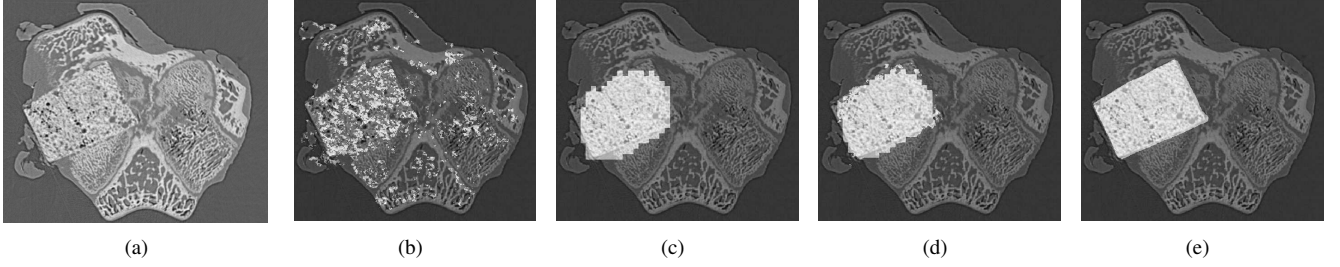


Figure 2. Consecutive classification results for a single 2D slice (shown in a): b) Gentle Boost; c) CRF; d) refinement; and e) manual (reference). Classification binary images are displayed combined with the original a) in order to facilitate comparison.

can guarantee that $|I_B| = \alpha \cdot |V|$. In our experiments, α has been set to 0.1.

Therefore, we define

$$p_+ = \{p_x \mid |\{y \mid p_y < p_x\}| = n_- + \alpha \cdot n_+\}$$

$$p_- = \{p_x \mid |\{y \mid p_y < p_x\}| = n_- - \alpha \cdot n_-\}$$

where $n_+ = |\{x \mid p_x \geq p\}|$, $n_- = N - n_+$, and $N = |V|$.

Defining the boundary interval I_B in V in such a way allows us guarantee that the class distribution will be maintained because the number of voxels moved to I_B from the initial class partitions are proportional to the obtained distribution.

As an implementation remark, it is worth to mention that output V of the spatial coherence model contains three components for each voxel: the label of the class it belongs to (l_V) together with the confidences about each class ($conf_+$, $conf_-$). As expected, the label value corresponds to the class having the highest confidence. When these two confidence values sum 1, it means that they correspond to probabilities and that the learning process converged properly.

$$V = \{(l_V, conf_+, conf_-) \text{ where} \\ l_V \in \{+1, -1\} \text{ and} \\ conf_+, conf_- \in [0, 1] \text{ and} \\ conf_+ + conf_- = 1 \text{ and} \\ l_V = +1 \text{ if } conf_+ > conf_-\}$$

Since convergence is a strong condition to impose to the proposed algorithms, we handle $conf_+ + conf_- \neq 1$ cases by forcing confidence values to comply with it. In order to do it, we assign: $conf_+ = \frac{conf_+ + (1 - conf_-)}{2}$ and $conf_- = 1 - conf_+$.

3 Results

3.1 Application domain

As previously stated, this paper focuses on the task of classification function construction from high-resolution 3D micro-tomographies (microCTs). These microCTs were taken by our group at the European Synchrotron Radiation Facility (ESRF) located at Grenoble (France) within the framework of a research project whose main objective is evaluating the quality of different biomaterials for bone reconstruction. In order to evaluate each biomaterial prop-

erties, they are implanted into rabbit femur bones by means of surgery. Afterwards, its evolution is tracked by taking microCTs, which are currently visually analyzed by means of volume rendering techniques. However, the task of identifying biomaterial inside the bone is currently being done manually. Henceforth, our main objective is to contribute to the automatization of this task.

Our experiments have been performed with a microCT that corresponds to a rabbit femur with a one-week-old bioimplant (see Figure 2 a) for a 2D slice). The dimensions of this data voxel model are $486 \times 423 \times 562$, having an approximate size of 111Mb. This microCT has been manually classified into biomaterial and non-biomaterial (which comprises both bone and background).

3.2 Metrics

To evaluate the quality of the classification results, we compute, in addition to the accuracy, the Overlap Metric (OM) for each class. We define the Overlap Metric for a class C_i and a learning method A as

$$OM(C_i, A) = \frac{|C_{i_A} \cap C_{i_{RM}}|}{|C_{i_A} \cup C_{i_{RM}}|}$$

where $C_{i_{RM}}$ stands for the set of instances that are classified as C_i in the Reference Model and C_{i_A} notes the set of instances that are classified as C_i by the learning method A . For each class, this metric approaches a value of 1 for results that are very similar and it is near 0 when they share no equally classified voxels.

3.3 Learning Process and Testing Results

In order to test the proposed learning pipeline process, we consider the micro-CT image described above as the input voxel model and its manual annotation as the classification label reference model. More concretely, we have performed the proposed learning process by taking one 2D slice as training data so that, we can afterwards test the classification with nine other 2D slices (i.e. using them as testing datasets). This experiment has been repeated and averaged over 10 different slices, so that our results correspond to a 10-fold cross validation. Considering the learn-

ing process tasks detailed in section 2, this selection of the training slice is done for the data acquisition and selection task. Thus, our training data are voxel-classification pairs, where voxels belong to a single 2D slice and are provided together with their manual classification.

Second task in the process –that is, feature extraction– is afterwards performed by applying Gabor Filters with 6 different orientations and 4 scales. As a consequence, this results in 24 different features per voxel. Furthermore, we need features to belong to the Real domain so we take the module of Gabor complex numbers. Let F_V and L_V be the features and the labels corresponding to the training slice.

Along the third task, we concatenate the learning of three different classifiers. Firstly, we use both F_V and L_V to train the Gentle Boost classifier along 100 iterations. Once the classifier is obtained, we use it to classify F_V so that we obtain new labels \hat{L}_V . Secondly, we use both L_V and \hat{L}_V to train the CRF classifier, so that \hat{L}_V can be afterwards classified. As a result, we obtain not only a new \tilde{L}_V but an associated confidence \tilde{Conf}_V . Finally, the refinement step will use this \tilde{Conf}_V to select the voxel features that require further treatment, so that they can be used (together with their corresponding labels in L_V) to train the third classifier. As an implementation detail, just mention that CRF process requires a downsizing of the data, so that refinement needs to take this into account and to resize accordingly.

Once all three classifiers have been trained, we can use a second slice for testing purposes. Analogously to the training process, we extract the Gabor features F_T . Thus, we use the first Gentle Boost classifier to classify F_T into \hat{L}_T labels, which are the input for the CRF classifier. This classification results in \tilde{L}_T and \tilde{Conf}_T so that refinement uses \tilde{Conf}_T to select boundary voxels from the original F_T . Selected voxels are then classified –by means of the third classifier– so that resulting labels can substitute the corresponding labels in \tilde{L}_T . Henceforth, the new \tilde{L}_T becomes the final classification that can be compared with the manual L_T .

Table 1 shows how classification performance is increased along the process. Accuracy is raised from 0.889 with the Gentle Boost classifier up to 0.972 when the overall pipeline process is performed. Nevertheless, biomaterial and non-biomaterial classes in this domain are highly unbalanced – their proportion is 1 to 8– and therefore, high accuracy results hide the fact that biomaterial classification is in fact much more difficult than labeling non-biomaterial voxels. Henceforth, we also consider the Overlap Metric given in 3.2, which is more informative than average accuracy, since error information are relative to each class. In fact, Gentle Boost only classifies correctly a proportion of 0.236 of biomaterial voxels. The CRF classifier greatly increases this figure, which reaches a 0.757. Finally, refinement still improves boundary voxels and, therefore, an overall proportion of 0.786 biomaterial voxels are correctly classified. As Figure 2 d) shows, the significance of last refinement can be visually appreciated in rendering. On the

other hand, as it can be seen in Table 1, the Overlap Metric of non-biomaterial class also improves so that an overall 0.970 is reached.

Metric	Gentle Boost	+CRF	+Refinement
Accuracy	0.889±0.001	0.967±0.004	0.972±0.004
OM_+	0.885±0.002	0.964±0.013	0.970±0.013
OM_-	0.236±0.034	0.757±0.064	0.794±0.068

Table 1. Learning performance metric averages and 95% confidence intervals. OM_+ and OM_- correspond to the Overlap Metric for non-biomaterial and biomaterial classes respectively.

As described in Section 1, the aim of this paper is to introduce machine learning techniques into the process of direct volume rendering. Therefore, visualization results are key to validate our approach. Figure 3 shows a direct volume rendering of 50 consecutive central slices. For this example, learning has been performed having as training data the 2D slice obtained from fixing z coordinate to 250. On the other hand, testing data corresponds to slices $z = 225$ to $z = 275$.

4 Conclusions

To summarize, this paper proposes a machine learning pipeline process to be introduced into the direct volume rendering of biomedical high-resolution 3D images. This pipeline process concatenates three different classifiers. Initially, an attribute-value Gentle Boost classifier provides a rather good classification whose spurious labels are removed by the application of a spatial coherence model (i.e., the CRF classifier). And finally, those voxels in the class boundaries are refined by the third classifier. This pipeline process has proven to greatly improve both the accuracy and the overlap metrics, specially for the classification of unbalanced classes.

5 Acknowledgements

This work was partially funded by the Spanish Science and Technology Ministry MAT2005-07244-C03-03 project.

References

- [1] M. J. Paulus, S. S. Gleason, S. J. Kennel, P. R. Hunsicker, and D. K. Johnson, “High resolution x-ray computed tomography: An emerging tool for small animal cancer research,” *Neoplasia*, no. 1–2, pp. 62–70, 2000.
- [2] A. Thompson, J. Llacer, L. Finman, E. Hughes, J. Otis, S. Wilson, and H. Zeman, “Computed tomography using synchrotron radiation,” Tech. Rep. LBL-

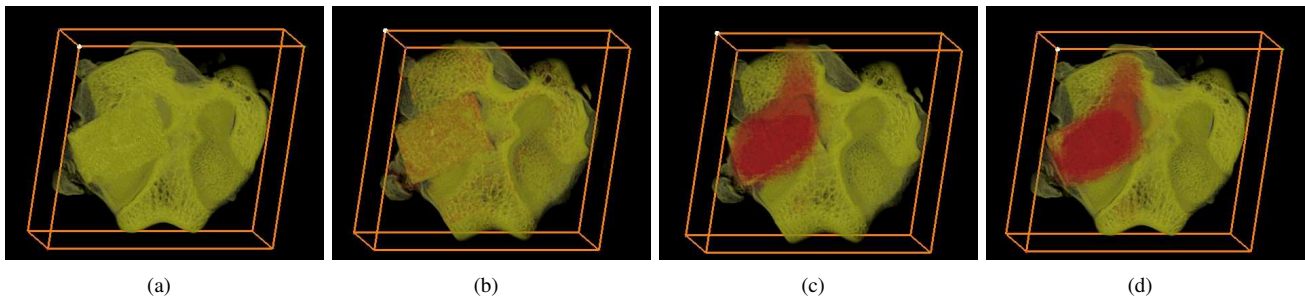


Figure 3. 3D Visualization of classification results over 50 consecutive slices: a) original; b)Gentle; c) CRF; and d) refinement.

16458;CONF-830910-12, Lawrence Berkeley Lab. Stanford University (CA), 1983.

- [3] J. Kniss, G. Kindlmann, and C. Hansen, "Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets," in *Proceedings of the conference on Visualization 2001*, pp. 255–262, IEEE Press., 2001.
- [4] M. Hadwiger, C. Berger, and H. Hauser, "High-quality two-level volume rendering of segmented data sets on consumer graphics hardware," in *Visualization 2003*, pp. 40–45, IEEE-CSP, 2003.
- [5] H. Pfister, B. Lorensen, C. Baja, G. Kindlmann, W. Shroeder, L. Avila, K. Raghuram, R. Machiraju, and J. Lee, "The transfer function bake-off," *IEEE Computer Graphics & Applications*, vol. 21, no. 3, pp. 16–22, 2001.
- [6] K. V. Leemput, F. Maes, D. Vandermeulen, and P. Suetens, "Automated model-based tissue classification of mr images of the brain," *IEEE Transactions on Medical Imaging*, vol. 18(10), pp. 897–908, 1999.
- [7] F. Tzeng, E. Lum, and K. Ma, "A novel interface for higher dimensional classification of volume data," in *Visualization 2003*, pp. 16–23, IEEE-CSP, 2003.
- [8] M. Ferré, A. Puig, and D. Tost, "A fast hierarchical traversal strategy for multimodal visualization," *Visualization and Data Analysis 2004*, pp. 1–8, 2004.
- [9] F.-Y. Tzeng and K.-L. Ma, "A cluster-space visual interface for arbitrary dimensional classification of volume data," in *EG- IEEE TCVG Symposium on Visualization 2004*, O. Deussen, C. Hansen, D.A. Keim, D. Saupe Eds., 2004.
- [10] G. Gerig, J. Martin, R. Kikinis, O. Kubler, M. Shenton, and F. Jolesz, "Unsupervised tissue type segmentation of 3-d dual-echo mr head data.," *Image and Vision Computing*, vol. 10, no. 6, pp. 349–36, 1992.
- [11] M. López-Sánchez, J. Cerquides, S. Ontañón, A. Puig, E. Puertas, and D. Tost, "Learning methods applied to high-resolution ct volume data classification," in *proceedings of the The Ninth International Conference on Computer Graphics and Artificial Intelligence (3iA2006)*, pp. 103–116, 2006.
- [12] Jain, A. Farrokhnia, and F. Michigan, "Unsupervised texture segmentation using gabor filters," *Systems, Man and Cybernetics*, pp. 14–19, 1990.
- [13] C. Chubb and M. Landy, *Orthogonal Distribution Analysis: A New Approach to the Study of Texture Perception*. ed Cambridge, MIT Press, 1991.
- [14] S.-C. Zhu, "Statistical modeling and conceptualization of visual patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 1–22, 2003.
- [15] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *International Conference on Machine Learning*, pp. 148–156, 1996.
- [16] "The boosting (and related ensemble learning methods) website." <http://www.boosting.org/>.
- [17] C. Sutton and A. McCallum, *An Introduction to Conditional Random Fields for Relational Learning in Introduction to Statistical Relational Learning*. MIT Press, 2006.
- [18] S. Vishwanathan, N. Schraudolph, M. Schmidt, and K. Murphy, "Accelerated training of conditional random fields with stochastic meta-descent," in *International Conference on Machine Learning (ICML'06)*, p. to appear, 2006.
- [19] J. C. Spall, *Introduction to Stochastic Search and Optimization*. Wiley, 2003.
- [20] P. Pakzad and V. Anantharam, "Estimation and marginalization using kikuchi approximation methods," 2003.
- [21] J. Yedidia, W. Freeman, and Y. Weiss, "Constructing free energy approximations and generalized belief propagation algorithms."