Learning Methods applied to high-resolution CT volume data classification

Maite López-Sánchez, Jesús Cerquides,
Santi Ontañón, Anna Puig, Eloi Puertas
WAI, Volume Visualization and Artificial Intelligence
MAiA Dept., Universitat de Barcelona
{cerquide,maite,santi,anna,eloi}@maia.ub.es
Dani Tost
CREB, Centre de Recerca en Enginyeria Biomèdica
Universitat Politènica de Catalunya, dani@lsi.upc.edu

Abstract

This paper analyzes how to introduce machine learning algorithms into the process of direct volume rendering. A conceptual framework for the optical property function elicitation process is proposed and particularized for the use of attribute-value classifiers. The process is evaluated in terms of accuracy and speed using different off-the-shelf classifiers (Decision Trees, Naïve Bayes, and Simple Logistic). The empirical results confirm the classification of high-resolution time-varying computerised tomography volume data as a challenging problem where an opportunity for further research emerges.

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]. Furthermore, many biomedical studies, such as the study of bone fractures, require the analysis of high-resolution time-varying images, where time-varying images are assumed to share some similarity among them. In these applications, the 3D region is sampled according to a regular 3D grid, by parallel image planes. For example, a typical data set is composed by 512^3 or even 1024^3 samples. The representation of the volume is a voxel model consisting of a set of parallel cubical and face-adjacent cells called *voxels* with a property value at each

voxel vertex and such that the reconstruction of the property inside a voxel can be done by interpolation of the voxel vertices property values. The property is usually *tissue density*, scaled as an *intensity* level between 0 and 2^n, being n the number of bits allocated for the storage of the intensity.

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 the anatomical structures contained in the data, the selection of the visualization preferences and the appearance of the different tissues 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 the different data values. Thus, the OPF at each point is computed as a mapping of its property value to the corresponding optical properties. Obviously, users should assign coherently similar optical properties to data values corresponding to the same regions. Selection of the anatomical regions to be visualized is accomplished indirectly by assigning to zero the optical property of opacity, since totally transparent samples do not contribute to the final image.

The use of transfer functions presents a major advantage: they can be stored as look-up tables, directly indexed by the intensity data values during the visualization, which significantly speeds up rendering. However, their manual definition is complicated even for skilled users. A lot of effort has been put on developing user friendly interfaces that make this definition more intuitive. Nevertheless, it is still an open problem. In addition, it has been proven [3] that using only one-dimensional transfer functions, based on the intensity data levels fails at accurately detecting complex combinations of material boundaries. The use of multidimensional transfer functions based on the first and second derivative of the intensity values brings major refinements to the classification but it makes harder its definition and increases the memory requirements for its storage.

The transfer function between intensity values and optical properties is often 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 it belongs to. SOPAF is a function that assigns to each anatomical structure a set of optical properties.

In this case, the elicitation is a two-step process. During the classification step a labelled voxel model is created that contains a unique identifier of the region to which the voxel belongs. This classified model is then used together with the original voxel model to build a (R, G, B, α) model to be visualized.

In this approach, the classification function

is used to skip non-selected regions, and thus reduces the cost of model traversal. It simplifies the edition of the transfer functions, separating the problem of selection and classification from the optical properties assignment problem. A drawback of this approach is that the usage of an intermediate labelled model increases memory requirements, which are critical during rendering.

On the contrary, the advantage of this approach is that, since the 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. In particular, a first step of segmentation can be applied to separate the regions of interest, followed by a labelling of the segmented regions which results in the assignment of a region identifier per voxel [4]. This strategy cannot be applied if classification is done on-the-fly, during rendering, but it is suitable for a pre-process. Another promising approach is the application of probabilistic classifiers. It was early described in [5], but has nevertheless been little addressed since then. Its major advantage is that it can be automatized for similar datasets. Thus, in the case of time-varying datasets, the classification function generated for one time-frame could be useful for the next one. The work presented in this paper explores this idea.

Many papers in volume rendering literature address classification [6]. Most of them are based on the edition of transfer functions and specially on the design of user friendly interfaces for their specification. Recently, some preliminary work based on learning methods have been published: supervised methods such as bayesian networks [7], neural networks [8] , decision trees [9] and non-supervised methods [10]. Additionally, in [11], clustering-based supervised and non-supervised learning methods are compared for the classification of magnetic resonance images (MRI). However, there is a lack of a systematic comparative study of the application of different learning methods to classification.

In this paper, we address classification as a *data mining* problem. We interpret voxels as objects to classify and their property values, derivatives and positions as the attributes to evaluate. We apply different learning methods to a subset of already classified voxels and after that, in order to test our method, we classify various voxel models. Our goal is threefold: to define an optical property function elicitation process based on machine learning approach; to compare the adequacy of different learning methodologies to classify data; and to determine the size and type of sampling of classified subsets used for learning that are more suitable for a posterior rendering.

2 Incorporating machine learning into the optical property function elicitation process

Machine learning in the most general sense allows a computer program to learn to perform a task by the analysis of previously solved tasks. If we consider the most general setting for the optical property function elicitation process, the task that we would be interested in automating is the transformation of a data voxel model into an optical property function. From a formal point of view, the task has as input a set of pairs, each containing a data voxel model and an optical property function used to visualize it (i.e. a problem and its solution). The output is

a function that maps a data voxel model into its corresponding optical property function. Due to the particular structure of this problem, most of the machine learning techniques in the literature cannot be applied "out-of-the-box". Furthermore, the possibilities for the user to control this process are very limited.

These two problems, unsuitability of common machine learning tools and lack of user control, can be overcome if we decide to apply machine learning to the class based process. The proposed resulting process can be seen in Figure 1. In this case, learning is applied only to the process of classification. Machine learning methods are much easier to apply to this process and the user gains control over the processes of selection and assignment of optical properties to the different classes or materials.

Figure 1 introduces the concept of *classification model*. A classification model contains the information needed to map a data voxel model into a classification function. It can be understood as an actionable compilation of the information residing in the training data provided as input to the process of *learning*.

In order to build a CF for an incoming data voxel model, we use the process of *classification function construction*, where the classification model is applied. Furthermore, sometimes it can be useful to incorporate additional information in the form of a *partial classification function*, which provides information about the classification of a subset of the space. This can be defined by the user by means of the *partial classification* process. For example, this gives us the possibility to ask the user to classify several 2D slices of the data and incorporate this information into the CF construction process. The information provided by the partial CF can also be used to improve or refine the classification model.

2.1 Incorporating Attribute-Value Learning

Several alternatives can be used to implement the learning process. A common assumption underlying many of the algorithms proposed by the machine learning community is the fact that the objects that are going to be classified can be described by a set of attributes, where each attribute can take a set of values. There are a plethora of algorithms embracing this assumption, and they are usually known as attribute-value learning algorithms.

The application of these algorithms turns out to be much easier if we divide the problem of classifying a data voxel model into as many independent problems as voxels contained in the image. This assumption, however, hinders the detection of contiguous regions, introducing discontinuities in the classifications. In order to alleviate this process, usually a *feature extraction* process is used, where for each voxel we can incorporate information about its neighborhood. In Figure 2, we can see the steps when the learning process is adapted to attribute-value algorithms, including a sampling step, where a subset of all the voxels available for learning is selected.

3 Simulations

3.1 Metrics and analyzed algorithms

In order to evaluate the accuracy of the learning methods, we have used manually-guided expert segmentation models as the target classification functions of the data voxel models. From now on, we will call reference model (RM) these

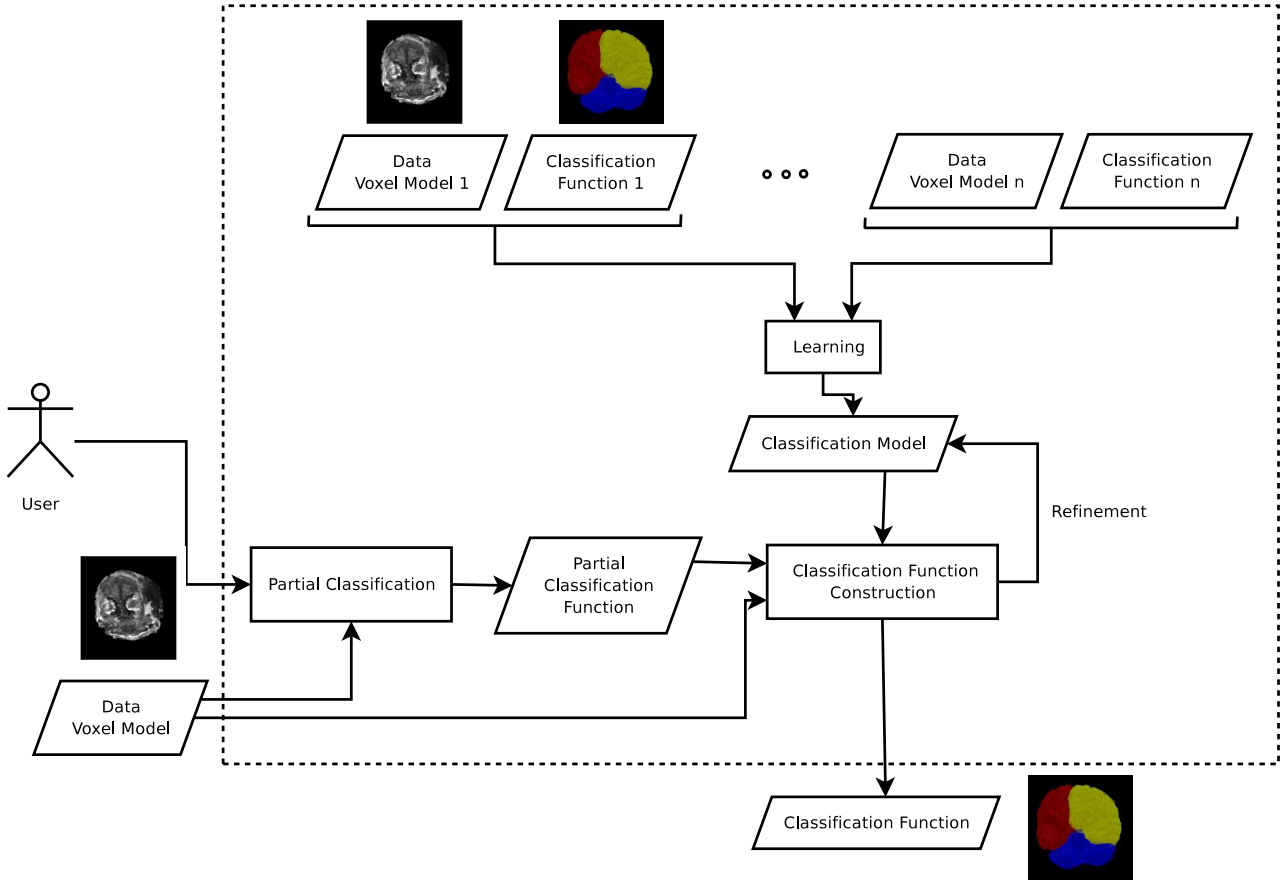

Figure 1: The class based direct volume rendering process incorporating machine learning.

labelled models. Moreover, we have constructed the set of classified instances, SCI, by sampling these reference models. We have used SCI as input for both processes, the partial and the complete classification process. Finally, we have compared the results of the models classified with the learning methods with the corresponding reference models RM.

We have chosen four different sampling methods to construct the set of classified instances from the first time-frame image: random, systematic, slice and stratified sampling. The random sampling selects a subset of data set instances at a random choice. Systematic sampling extract a subset of instances at regular sampling steps starting from a random initial offset. This sampling can be used for scaling our data sets as a 3D geometrical transformation, and it can be interpreted as if our original data had been captured with lower resolution. As its name points out, slice sampling implies to choose a few equally spaced 2D slices of the 3D voxel model. Finally, the stratified sampling method has the aim of generating a sample whose class distribution is as balanced as possible.

In the present study, we analyze three different attribute-value learning algorithms in order

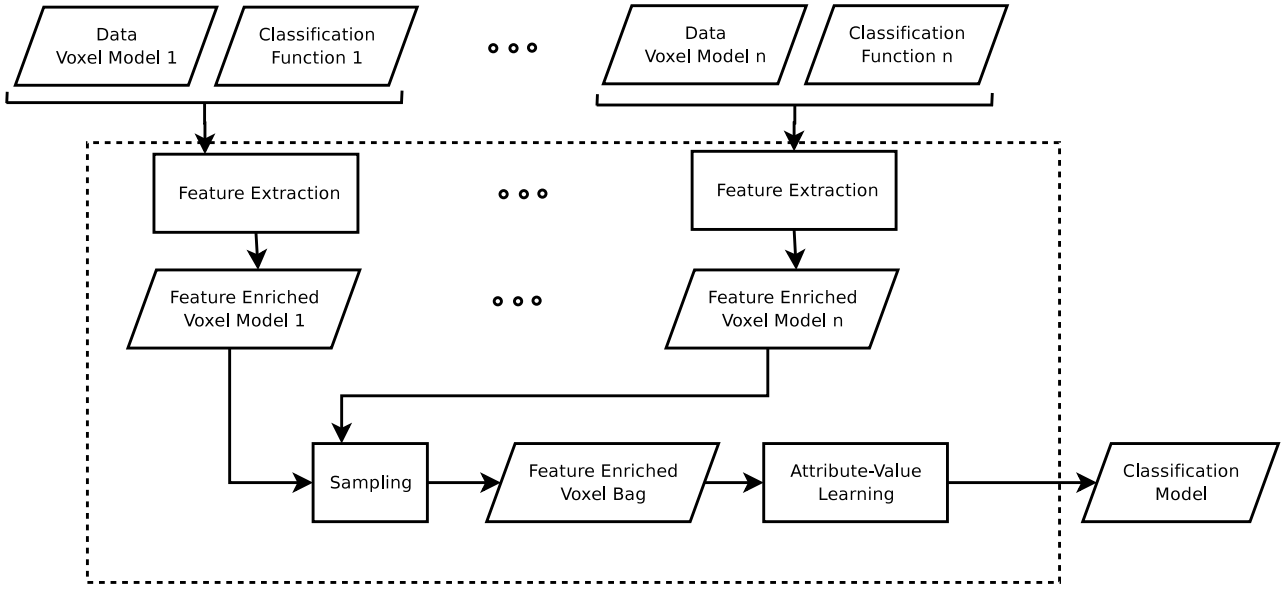

Figure 2: The learning process adapted to use attribute value learning algorithms

to evaluate their performance and their trade-off between time and accuracy in similar volume data. The evaluated algorithms are:

- **Naïve Bayes** [12]: It is a widely known, simple but very robust algorithm that is commonly used as a reference method. It can only capture simple dependencies between the non-class attributes and the class attribute.
- **Decision Trees**: Induction of decision trees [13] is one of the most used off-the-shelf classification algorithms that provides a good balance between accuracy, speed and interpretability of results.
- **Simple Logistic**: Logistic regression models are commonly used by the statistical community. The simple logistic [14] algorithm is an up-to-date representative of this family that uses boosting [15] to calculate regressors.

In the original experimental design we included a fourth algorithm based on neural networks, but backpropagation turned out to be much too expensive in terms of learning time for this size of datasets.

To evaluate the quality of the classification results, we compute the Overlap Metric (OM) for each class. In this manner, we define the Overlap Metric for a class C_v and a learning method A as

$$OM(C_v, A) = \frac{|C_{v_A} \bigcap C_{v_{RM}}|}{|C_{v_A} \bigcup C_{v_{RM}}|}$$

where $C_{v_{RM}}$ stands for the set of instances that are classified as C_v in the Reference Model and C_{v_A} notes the set of instances that are classified as C_v 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 similarly classified voxels.

The classification algorithm results are evaluated on the basis on their learning time, size

of the learned models, Overlap Metric and classification time for a microCT slice.

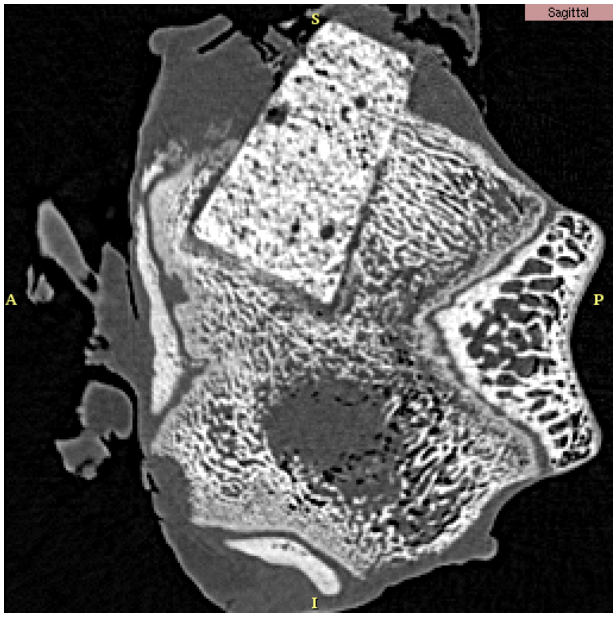

Figure 3: A first week microCT slice. The bioimplant can be indentified as the rectangular area on the left.

3.2 Application domain

In order to compare these different learning algorithms we have applied them to the task of classification function construction from time-varying high-resolution 3D microtomographies (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 properties, they are implanted into rabbit femur bones by means of surgery. Afterwards, its evolution is tracked by taking microCTs at different points in time. Those microCTs are currently visually analized by means of volume rendering techniques. However, the task of identifying biomaterial inside the bone is currently being done manually. Therefore, our main objective is to contribute to the automatization of this task.

Our experiments have been performed with two microCTs, corresponding to the first and fourth weeks respectively (see Figure 3). These data voxel models are do have different dimensions: $486 \times 423 \times 562$ and $498 \times 426 \times 544$ each having an approximate size of 111Mb. Both microCTs have been manually classified into biomaterial and non-biomaterial (which comprises both bone and background) so they can be used for testing different learning methods. The feature extraction process in Figure 2 composes a 5-dimensional attribute vector for each voxel. The attributes used are the classical features reported in literature: 3-D position, intensity value and gradient magnitude. It is worth noticing that no registration process has been performed between both microCTs, and hence, positional information is hardly transferable among them. Finally, the four sampling methods presented before are used to construct our Feature Enriched Voxel Bag (again, see Figure 2). The samples taken contain a 1% of the complete Feature Enriched Voxel Model.

We have compared the three classifiers introduced above for the task of Classification Function completion (that is, for extending a partial classification). Furthermore, we have also compared them for the task of classifying the second data voxel model by using the Classification Model previously obtained. Next section presents an analysis of the results.

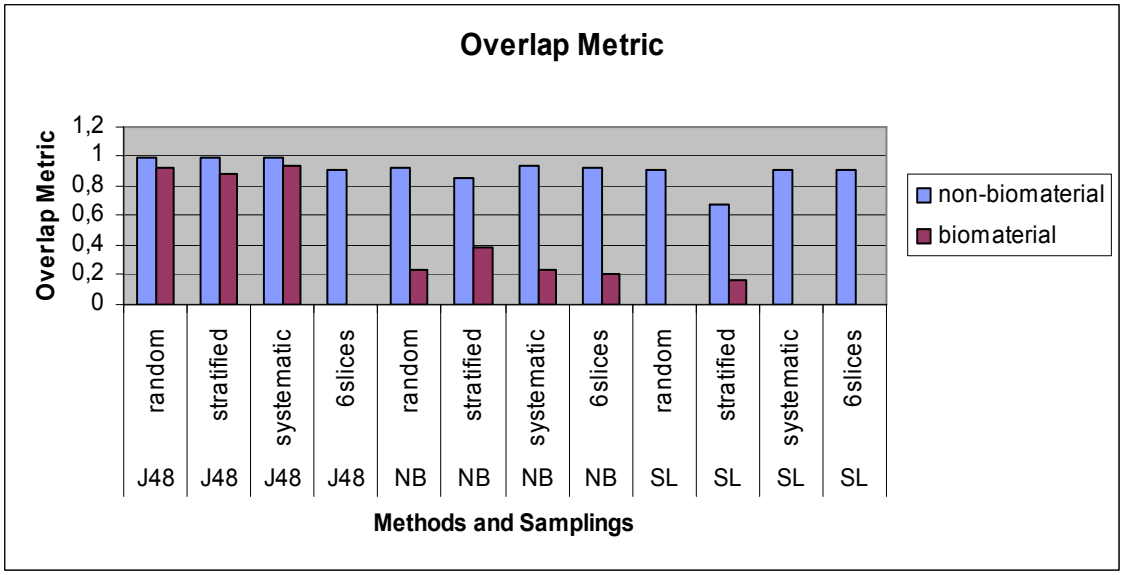

Figure 4: Per class, Overlap Metric for each learning and sampling method over the first week image.

Sampling	NB	J48	SL
random	2.9Kb	329Kb	170Mb
stratified	2.9Kb	375Kb	170Mb
systematic	2.9Kb	380Kb	170Mb
slice	2.9Kb	180Kb	182Mb

Table 1: Classification Model size for each learning and sampling method

3.3 Experimental results

When comparing different classifiers, the first measure we analyze is the size of the Classification Model they generate. Our results can be seen in Table 1: NB builds the smaller model; J48 still keeps a reasonable model size; and SL generates much larger models.

3.3.1 Overlap Metric analysis

Figure 4 shows the Overlap Metric for Decision Trees (J48), Naïve Bayes (NB) and Simple Logistic (SL) when classifying 50 complete consecutive slices from the first week microCT. For all methods, the Overlap Metric for non-biomaterial outperforms the one for biomaterial and this outperformance is especially significant for SL and NB methods. This difference can be explained by the characteristics of the classes: Non-biomaterial corresponds to a non-anatomical region, it concentrates most of the samples and covers a wide range of property values. On the contrary, biomaterial class has a smaller number of instances, making it harder to learn. We observe in the figure that J48 has an overall higher Overlap Metric. On the other hand, Naïve Bayes and Simple Logistic behave in a similar way. However, we must note that Simple Logistic fails to find biomaterial

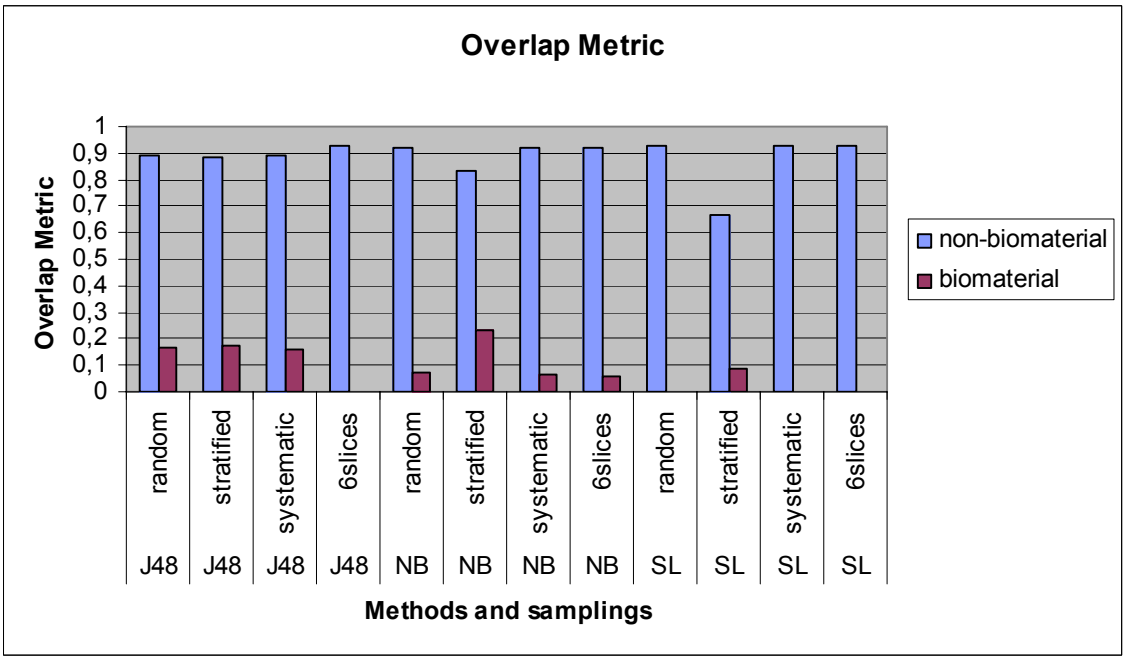

Figure 5: Per class, Overlap Metric for each learning and sampling method over the fourth week image.

for every sampling method except for stratified sampling.

Regarding sampling methods, the response of every learning algorithm is different: while J48 best results are obtained by using systematic sampling, NB and SL obtain a more balanced Overlap Metric with stratified sampling. In this manner, systematic sampling increases the Overlap Metric for the non-biomaterial class at the cost of decreasing the one for biomaterial. On the other hand, the reason for J48 low performance with slice sampling is that the classifier overfits to the provided sample and hence it becomes unable to generalize to other slices.

Figure 5 shows the Overlap Metric when trying to classify the fourth week microCT using the model learnt from a sample of the first week microCT. As would be expected, the difference between images causes an overall Overlap Metric reduction with respect to the results in Figure 4. This is mainly due to the previously stated fact that no registration between both microCTs has been performed. It should be noted that the reduction in Overlap Metric is most relevant for the biomaterial class. Additionally, whilst in Figure 4 the best performing combination was provided by J48 with systematic sampling, here, no combination overperforms the rest.

In Figure 5 we can also observe that J48 is very sensitive to image changes, somehow corroborating our hypothesis that it tends to overfit the input data. On the other hand, NB and SL methods, specially under stratified sampling, seem to be more able to adapt their

classifications to a different microCT. In fact, the behaviour of NB and SL with respect to the different sampling methods is analogous to the one in Figure 4. Overall, both figures show a poor performance of the algorithms with the slice sampling method.

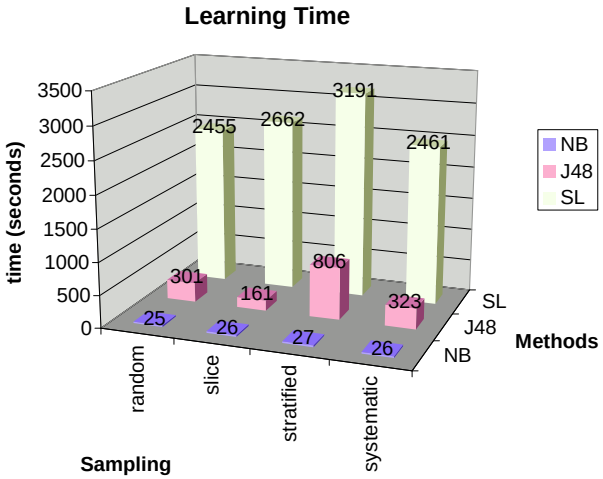

Figure 6: Learning times for each learning algorithm and sampling method

3.3.2 Time analysis

Regarding learning time, it can be seen in Figure 6 that NB clearly outperforms SL and J48. However, it should also be noted that J48 has a significantly larger Overlap Metric than NB, so they provide us with a trade-off between learning time and accuracy. Specially noticeable is the large learning time of SL, which together with its low Overlap Metric, allows us to discard this method for the task in the future.

With respect to the sampling methods, NB learning time is homogeneous, which is not the case for J48 and SL. For those two classifiers, stratified sampling significantly increases the learning time.

Figure 7 shows the time spent on classifying a microCT slice. J48 performs better than both SL and NB. However, the difference is much larger with respect to NB than with respect to SL.

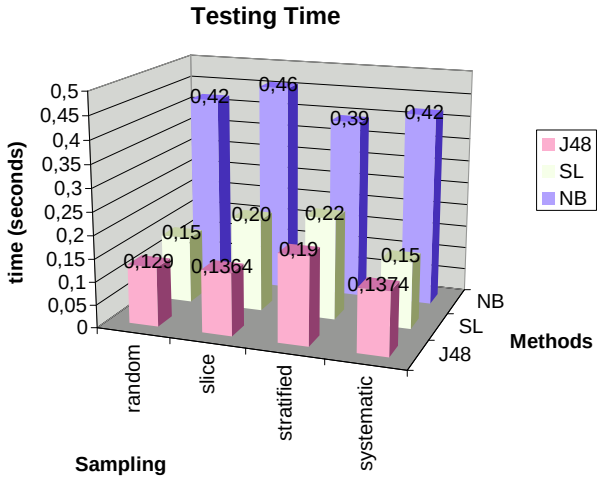

Figure 7: Time to classify a 486×423 voxels microCT slice for each learning algorithm and sampling method

4 Conclusions

In this paper we have analyzed how to introduce machine learning algorithms into the process of direct volume rendering. We have developed a conceptual framework for the optical property function elicitation process and have particularized it for the use of attribute-value classifiers. We have tested the suitability of this process through the experimental evaluation in terms of accuracy and speed of three different learning algorithms.

Our results regarding efficiency point out that the process can be considered feasible when implemented as a preprocessing step in the visualization pipeline.

From the point of view of accuracy, it should be noted that average accuracy is not providing all the information needed. The Overlap Metric is more informative since error information relative to each class highlights which classes are more prone to error. Under this perspective, our classification results are still not usable for real life applications. However, it should be noted that all the evaluated learning methods are general purpose classifiers. Better results should be expected if specific algorithms are designed. At this point we have identified two different features that could be helpful in order to provide a better solution to our problem. The first one is the ability to identify textures, because individual voxel property values have proven insufficient for a correct classification, even including neighborhood information in the form of gradient values. The second one is connectivity, since biomaterial has been implanted as a single piece. In addition, a richer learning set is expected to increase classification performance. Evaluating the relevance of the different attributes provided to the learning algorithms arises as future work.

Automating the process of classification has confirmed itself as a challenging problem, requiring the accurate use of all sources of data available. In this sense, the proposed process can benefit from integrating the information provided by both previously classified voxel models and the partial CFs. The former could be embedded in the system whilst the obtention of the latter requires the design of specific user interfaces which should be highly intuitive and usable.

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-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] R. Drebin, L. Carpenter, and P. Hanrahan, "Volume rendering," *ACM Computer Graphics*, vol. 22, pp. 65–74, August 1988.

[6] H. Pfister, B. Lorensen, C. Baja, G. Kindlmann, W. Shroeder, L. Avila, K. Raghu, R. Machiraju, and J. Lee, "The transfer function bake-off," *IEEE Computer Graphics & Applications*, vol. 21, no. 3, pp. 16–22, 2001.

[7] K. V. Leemput, F. Maes, D. Vandermeulen, and P. Suetens, "Automated model-based tissue classication of mr images of the brain," *IEEE Transactions on Medical Imaging*, vol. 18(10), pp. 897–908,

1999.

[8] 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.

[9] M. Ferré, A. Puig, and D. Tost, "A fast hierachical traversal strategy for multimodal visualization," *Visualization and Data Analysis 2004*, pp. 1–8, 2004.

[10] 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.

[11] 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.

[12] P. Domingos and M. Pazzani, "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss," *Machine Learning*, vol. 29, pp. 103–130, 1997.

[13] J. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1992.

[14] N. Landwehr, M. Hall, and E. Frank, "Logistic model trees.," in *ECML*, pp. 241–252, 2003.

[15] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *European Conference on Computational Learning Theory*, pp. 23–37, 1995.