Fast global interactive volume segmentation with regional supervoxel descriptors

Imanol Luengo[a,b], Mark Basham[b], and Andrew P. French[a]

[a]Computer Vision Laboratory, University of Nottingham, Nottingham, UK
[b]Diamond Light Source Ltd, Harwell Science & Innovation Campus, Didcot, UK

ABSTRACT

In this paper we propose a novel approach towards fast multi-class volume segmentation that exploits supervoxels in order to reduce complexity, time and memory requirements. Current methods for biomedical image segmentation typically require either complex mathematical models with slow convergence, or expensive-to-calculate image features, which makes them non-feasible for large volumes with many objects (tens to hundreds) of different classes, as is typical in modern medical and biological datasets. Recently, graphical models such as Markov Random Fields (MRF) or Conditional Random Fields (CRF) are having a huge impact in different computer vision areas (e.g. image parsing, object detection, object recognition) as they provide global regularization for multiclass problems over an energy minimization framework. These models have yet to find impact in biomedical imaging due to complexities in training and slow inference in 3D images due to the very large number of voxels. Here, we define an interactive segmentation approach over a supervoxel space by first defining novel, robust and fast regional descriptors for supervoxels. Then, a hierarchical segmentation approach is adopted by training Contextual Extremely Random Forests in a user-defined label hierarchy where the classification output of the previous layer is used as additional features to train a new classifier to refine more detailed label information. This hierarchical model yields final class likelihoods for supervoxels which are finally refined by a MRF model for 3D segmentation. Results demonstrate the effectiveness on a challenging cryo-soft X-ray tomography dataset by segmenting cell areas with only a few user scribbles as the input for our algorithm. Further results demonstrate the effectiveness of our method to fully extract different organelles from the cell volume with another few seconds of user interaction.

Keywords: Volume, Interactive segmentation, Random Forest, Contextual Information, Markov Random Fields

1. INTRODUCTION

The amount of data generated by current bio-imaging techniques is currently overtaking our ability to efficiently analyse it. In order to analyse a biomedical volume, expert knowledge is required to visually explore the volume and manually tag different areas/sections of it for measurement. This manual segmentation labour is tedious, error prone and time consuming. In the past years, several computer vision approaches have been adapted to the task of biomedical image segmentation with the aim of reducing the user effort needed to segment medical images. By combining expert knowledge with template based techniques, Active Appearance Models[1] or Atlas based models[2] are able to learn a template of the human body an automatically segment its bones, brain areas or individual organs. These approaches however, require a significant amount of training data, that over the past decade has been collected by several researchers or doctors. When there is a limited amount of training data available, Active Contour Models (*Snakes*,[3,4] or Level Set methods[5]) are able to assist the end user, which only needs to draw an approximate boundary around the desired object and the contour will evolve (expand or shrink) to efficiently match the object's boundary and segment it.

Further author information: (Send correspondence to Imanol Luengo)
Imanol Luengo: imanol.luengo@nottingham.ac.uk
Mark Basham: mark.basham@diamond.ac.uk
Andrew P. French: andrew.p.french@nottingham.ac.uk

Such models are efficient and provide very accurate segmentation of the objects of interest, either by volume registration or contour optimization. However, biological datasets (i.e. TEM, SEM or cryo-TEM datasets) remain very challenging as each dataset corresponds to a different sample, which has its own subtleties in terms of image properties. This sample is usually at the cell-scale with low signal to noise ratio, with potentially tens to hundreds of subcellular features such as organelles which need labelling. Here, as the internal organization of a cell is not likely to match any other one, template models cannot be learnt. Active Contour Models could still provide a sufficient segmentation of different organelles, however, this would involve too much user interaction as a contour would be needed for each of many organelles.

To handle the lack of structure in the target datasets, machine learning approaches have been adopted. Here, using available training data, local voxel descriptors are extracted and a classifier is trained to learn to classify a voxel into different classes. This approach has previously been used to successfully segment mitochondria[6] or cell boundaries and organelles.[7] It is very effective, and only requires a limited amount of previously available training data to automatically segment future datasets. This approach still requires expert interaction to segment some initial datasets, but after sufficient manual segmentation is gathered, the user would no longer be necessary and the learnt model would be sufficient to automatically segment future datasets based on local or global learnt voxel/supervoxel statistics.

Despite all the advantages of the machine learning based models, here they are not sufficient as they require a considerable amount of training data. In some environments, biologists are constantly gathering volumes of different samples. Different samples have different image properties or correspond to a different organism, and thus, the learnt model doesn't necessarily match the target dataset. Moreover, in some cases just a single volume of a single sample is obtained, which is hard to generalize and different to previously seen ones. Despite some approaches trying to solve this problem by manually segmenting part (half or a third) of the dataset and use it to learn a model and segment the other remaining part,[6] this still requires a per-volume tedious and time-consuming user interaction.

Thus, due to the lack of structure in the dataset, and the scarcity of similar training data, a trend in semi-automatic volume segmentation techniques has been growing recently. Semi-automatic segmentation techniques (also termed *interactive techniques*) aim to find a balance between automatic methods and manual segmentation. Here, the user is assisted and by using a short period of user interaction, such approaches learn a model on-the-fly to segment the target dataset. There are different ways of incorporating user knowledge in interactive approaches; the previously mentioned Active Contours Methods is one of them, where the user draws a contour and the algorithm refines it. Other approaches use user scribbles, acting as simple and fast markup on the image,[8] to learn a classification model and propagate class probabilities to the rest of the dataset. However, so far the effort of interactive approaches has mainly focussed on assisting the user to segment a few organelles. In biomedical volumes, due to the large size of the datasets ($1000 \times 1000 \times 500$ voxels), this approach requires too much user interaction to achieve desirable results, and have a computationally expensive training and testing steps.

Here we present an interactive segmentation pipeline that can be used to segment any kind of volumetric data with an arbitrary number of classes and number of objects of interest with the same user effort. Our method uses a supervoxel oversegmentation[9] to represent the volume reducing the complexity of the following algorithms by several orders of magnitude compared with working over raw voxels. Regional features are extracted from each of these supervoxels and, using the user scribbles as a ground truth and the only input for our method, a hierarchy of labels is created, and an Extremely Random Forest (ERF)[10] is trained for each of the hierarchy levels, using the output of the previous level as contextual information for the next level classifier. By using the probabilistic output of the ERF hierarchy as the unary potentials, the final segmentation is then refined by a Markov Random Field to yield the final result. The main objectives of our pipeline are:

- **Assist the user during the segmentation:** incorporates user knowledge in the form of scribbles to hierarchically segment a given volume.
- **Reduce the amount of user interaction:** uses supervoxels in order to reduce the number of scribbles needed by the user.

- **Offer a fluid segmentation experience:** uses supervoxels to reduce computational and time complexity, as well as GPU computing units to parallelize almost all steps of the pipeline to offer a near real-time experience to the user.

The main contributions of our work can be summarized as follows:

1. **GPU Implementation of SLIC Supervoxels:** a 3D variant of SLIC superpixels,[9] inspired by g-SLIC,[11] which allow us to extract supervoxels from a $1000 \times 1000 \times 200$ volume in around 10-15 seconds.
2. **3D texture features based on Sigma Set:**[12] simple yet robust and efficient features using Sigma Set descriptors, which are covariance-like descriptors that reside in a Euclidean space. These features are very efficient to compute and are also parallelized in the GPU.
3. **Use of contextual information to refine classification:** the user is asked to draw hierarchically more informative scribbles to allow us to partition the image and obtain better estimates with Random Forest, as training a Random Forest using the contextual information of previous levels of the hierarchy reduces the bias towards common classes and the overall variance by spatially separating the classes. This effect is similar to the widely studied auto-context[13] adapted to an interactive segmentation problem. We also include contextual information as a hierarchical label cost in the MRF refinement.

The rest of the paper will be organized as follows: Section 2 will briefly discuss relevant and similar work. Section 3 will briefly describe how to adapt SLIC Supervoxels to oversegment X-ray volumes in the GPU. Section 4 will explain the generation of Textural Sigma Set features for biomedical texture segmentation. Section 5.1 will detail the main contribution of our paper, the hierarchical labels and ERF training and its effect in the classification. Section 5.2 will show how we infer the optimal segmentation from the ERF likelihoods using the label hierarchy as label costs. And, finally, section 6 will show quantitative and qualitative results of our approach compared to a baseline.

2. RELATED WORK ON 3D SEGMENTATION

There have been many approaches trying to segment biomedical images using machine learning based approaches powered by ensemble methods (such as (Extremely)Random Forests or boosting). Ensemble methods consist of a set of weak classifiers trained on different subsets of data or with different weighted samples. Each of the classifiers is not by itself very robust, but, as they have limited information, each classifier tends to learn different discriminative rules. By combining them in the prediction step, the average response of multiple weak classifiers has been proven to give state-of-the-art performance. Additionally, ensemble methods have many appealing properties: they are fast to train, data doesn't need to be normalized and their natural feature selection process allows for efficient training on datasets with large numbers of features and samples (as in the case of a large 3D volume).

The most similar approaches to ours include *ilastik*,[8] which proposes an interactive segmentation pipeline by using Random Forests with Gaussian filter bank features to segment 3D volumes to segment 3D volumes, and the work from Lucchi et al.[6] which introduced supervoxel based mitochondria segmentation with learned shape features. However, *ilastik* operates at pixel level and requires a lot of training data (manual interaction) to be able to segment a challenging volume, while the approach of[6] is very specific. Here we propose an interactive segmentation pipeline on the supervoxel space by performing hierarchical classification to add contextual information.

Adding contextual information to the training and testing process has been gaining a lot of focus recently. The general idea of incorporating contextual information to the estimation of the class-probabilites of a pixel seems quite appealing since standard classification techniques don't exploit the underlying 3D neighbouring structure of voxels. In 2009 Tu and Bai[13] proposed an auto-context algorithm to train hierarchically probabilistic boosting trees (PBT)[14] using the output of one classifier as an input for the next one with interesting results for Brain MRI volume segmentations resulting in increasingly more accurate probability estimates in further layers of the

hierarchy. Later in 2011 A. Montillo et al.[15] proposed Entagled Decision Forests (EDT) to segment CT images. EDT forest also add contextual information to the training stage by adding class-probabilities of neighbouring voxels in a previous node of the decision trees as additional features for the voxels.

The works above, however, learn the contextual information during training step. Here we propose a similar approach to add contextual information to the interactive segmentation by adding a 2-step segmentation. The first step requires quick user scribbles to infer probabilistic volume areas that are used as contextual information for the second step supervoxel classification.

3. SUPERVOXEL OVERSEGMENTATION

Superpixels consist of a group of neighbouring pixels in a given image that share some properties, such as texture or color. Each of the pixel of the image belong to exactly one superpixel, and by adopting the superpixel representation of an image, the complexity of a problem can be reduced two or three orders of magnitude. A sample image with $500 \times 500 = 250000$ pixels can be encoded with just ≈ 800 superpixels, which has a dramatic effect on the speed of any subsequent algorithms, which can act at the superpixel, rather than pixel, level. Superpixels, in order to allow an efficient representation of the image, must share a certain characteristic: every strong boundary of the image should lay along superpixel boundaries, or in other words, all the pixels of a superpixel must lay in a single higher lever object of the image.

A range of superpixel generation methods exisit. Simple Linear Iterative Clustering (SLIC)[9] is one of the most common. It is a superpixel oversegmentation algorithm that splits a given image into superpixels by performing a k-means-based local clustering of the pixels in the 5-D $\{labxy\}$ space. The algorithm take as an input the approximate desired number of superpixels K and a compactness factor m. Then, for an image of N pixels, the approximate size of each superpixel becomes N/K and the image is split into a grid of uniformly distributed superpixels of size $S \times S$ with $S = \sqrt{N/K}$. Then, an iterative local k-means is applied by searching for every super pixel in the grid the pixels that belong to it in a $2S \times 2S$ window according to a custom distance function that controls the compactness of the superpixels D_{sp}. Each pixel is assigned to the cluster with smallest distance. For a given pixel $i \in N$ and a given superpixel centre $k \in K$, the distance from the pixel i to the superpixel k would be defined as $D_{sp}(i,k)$:

$$\begin{aligned} d_{lab}(i,k) &= \sqrt{(l_i - l_k)^2 + (a_i - a_k)^2 + (b_i - b_k)^2}, \\ d_{xy}(i,k) &= \sqrt{(x_i - x_l)^2 + (y_i - y_k)^2}, \\ D_{sp}(i,k) &= d_{lab} + \frac{m}{S} d_{xy}, \end{aligned} \tag{1}$$

where m is the compactness factor, $\{l, a, b\}$ is the L*a*b color space and $\{x, y\}$ corresponds to the x and y position of the pixel in the image and the cluster centre. As the k-means algorithm, a 2-step expectation and maximization algorithm optimizes the $\{l, a, b, x, y\}$ feature of the cluster centres and each pixel is finally assigned to its nearest cluster after $T = 5$ iterations.

Later, g-SLIC[11] introduced a GPU version of SLIC that replaces the search in $2S \times 2S$ windows around each superpixel centre by a lookup table that contains the $M = 9$ nearest cluster centres for each pixel. Thus, instead of performing K sequential searches for every superpixel, M searches are made for every pixel in parallel. The computational cost would be much higher in a sequential CPU, however, by running it in a GPU with hundreds of cores the performance is greatly improved.

To apply SLIC to a volumetric X-ray dataset we sightly change the formulation of[9] and.[11] First, to handle the anisotropicity of some X-Ray datasets, we replace the desired number of superpixels K with the desired average size of the supervoxels $\mathbf{S} = \{S_x, S_y, S_z\}$ in each dimension. This allows us to make for example less deep superpixels (i.e. $\mathbf{S} = \{S_x, S_y, S_z\} = 10 \times 10 \times 3$ supervoxels) if X and Y are the isotropic axes and Z suffers anisotropicity due external factors like limited data or limited available angle in CT-reconstruction. The total number of supervoxels is extracted by the relation between their expected size and the total number of voxels N in the image $K = N/(S_x \times S_y \times S_z)$. By initially sampling K supervoxels of shape $\mathbf{S}$ we already encourage superpixels of customizable shapes in the image. However, to enforce that constrain over the optimization

scheme, we introduce the standard spacing factors $\mathbf{d} = \{d_x, d_y, d_z\}$ to weight spatial distance from a voxel to a superpixel centroid along the different axes. Finally, the 5-D $\{l, a, b, x, y\}$ feature vector is replaced with a 4-D $\{I, x, y, z\}$ vector where I is the intensity of a voxel in a X-ray volume. Equation 1 can then be re-written as

$$\begin{aligned} d_I(i,k) &= \sqrt{(I_i - I_k)^2}, \\ d_{xyz}(i,k) &= \sqrt{\big(d_x(x_i - x_l)\big)^2 + \big(d_y(y_i - y_k)\big)^2 + \big(d_z(z_y - z_k)\big)^2}, \\ D_{sp}(i,k) &= d_I + \frac{m}{|S|} d_{xyz}. \end{aligned} \tag{2}$$

The spacing factors can be set to existing sampling units (in mm, μm or nm) or empirically defined by the user. Note that by setting $\mathbf{d} = \mathbf{1}$ the same spatial distance function d_{xy} is obtained (its 3-D extension).

Last, in order to parallelize the algorithm in the GPU, we use a similar approach to g-SLIC, and parallelize over the voxels by finding at each Expectation-step (of the local k-means) the nearest 26 supervoxel centers (26-connected in 3D) for each voxel and assign to each one its nearest one according to equation 2.

4. TEXTURAL SIGMA SET FEATURE EXTRACTION

Once supervoxels are constructed from the volume, we represent each supervoxel by means of simple first and second order statistics. Typical approaches include using the mean intensity of the supervoxel, or the histogram of intensities within each supervoxel. More advanced features include SIFT or Histogram of Oriented Gradients (HOG), which have been proven to be extremely robust for object detection, but lose performance when the objects of interest have a lot of variability in shape and size.

Here, we use a simple yet robust and efficient Sigma Set descriptor,[12] based on regional covariances that has been used before for saliency estimation and object detection.[16] With $\theta(\mathcal{V}, x, y, z)$ a mapping function that extracts d-dimensional feature $\mathbf{f_i}$ for each voxel i in a volume $\mathcal{V}$ (such as position, intensity, derivatives or other local textural cues). Then, a region $\mathcal{R}$ inside the feature space θ can be described by means of a $d \times d$ covariance matrix

$$\mathbf{C}_\mathcal{R} = \mathbf{F}_\mathcal{R} \mathbf{F}_\mathcal{R}^T, \tag{3}$$

where $\mathbf{F}_\mathcal{R} = [\hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2 \ldots, \hat{\mathbf{f}}_n]$ denotes the feature matrix of the centered d-dimensional feature vectors $\hat{\mathbf{f}}_i = 1/\sqrt{N}(\mathbf{f}_i - \boldsymbol{\mu})$ of all the voxels $i \in \mathcal{R}$. As noted in[12] a non-singular covariance matrix (or symmetric positive definite matrices) lie in a nonlinear Riemannian manifold and not in an Euclidean space. Thus, in order to perform classification or evaluate differences between descriptors, these need to be projected onto a linear space. To achieve this, the Sigma Set creates a set of points S that satisfies $\mathbf{C}_S \simeq \mathbf{C}_\mathcal{R}$ in terms of first and second order statistics and define efficient Euclidean distance transformations.

For this purpose, the Cholesky decomposition of the covariance $\mathbf{C} = \mathbf{C}_\mathcal{R}$ is extracted $\mathbf{C} = \mathbf{L}\mathbf{L}^T$, where $\mathbf{L}$ is a lower triangular matrix, and multiplied by a scalar that as[12] suggests is set to $\sqrt{d}$, yielding $\hat{\mathbf{L}} = \sqrt{d} \times \mathbf{L}$. Thus, our first regional descriptor for a supervoxel k is constructed as:

$$\boldsymbol{\phi}_k = \{\hat{\mathbf{L}}_1, \ldots, \hat{\mathbf{L}}_d, (-\hat{\mathbf{L}}_1), \ldots, (-\hat{\mathbf{L}}_d)\}, \tag{4}$$

where $\hat{\mathbf{L}}_i$ is the i-th column of $\hat{\mathbf{L}}$. This descriptor has the same second order statistics as $\mathbf{C}_\mathcal{R}$, and by adding the mean vectors, the same 1st orders statistics can also be recovered as

$$\boldsymbol{\phi}_k^+ = \{(\hat{\mathbf{L}}_1 + \boldsymbol{\mu}), \ldots, (\hat{\mathbf{L}}_d + \boldsymbol{\mu}), (-\hat{\mathbf{L}}_1 + \boldsymbol{\mu}), \ldots, (-\hat{\mathbf{L}}_d + \boldsymbol{\mu})\}, \tag{5}$$

which yields the final sigma set descriptor. We will later see (in section 5.1.1), that adding the mean $\boldsymbol{\mu}$ vector to the descriptor is helpful in most situations, but has to be used with caution as samples will be biased towards the mean descriptor.

In this work, we use a *7*-dimensional texture descriptor and a *10*-dimensional feature vector to describe the position and texture of a voxel $i \in \mathcal{V}$

$$\mathbf{f}_1 = \{I, I_x, I_y, I_z, I_{xx}, I_{yy}, I_{zz}\}, \tag{6}$$

$$\mathbf{f}_2 = \{x, y, z, I, I_x, I_y, I_z, I_{xx}, I_{yy}, I_{zz}\}, \tag{7}$$

where x, y and z indicate the absolute position (in range $[0, 1]$), I indicates the intensity of the voxel and subscripts indicate directional derivatives. The final Sigma Set descriptors are then constructed from the covariance matrix of all the voxels belonging to each of the supervoxels.

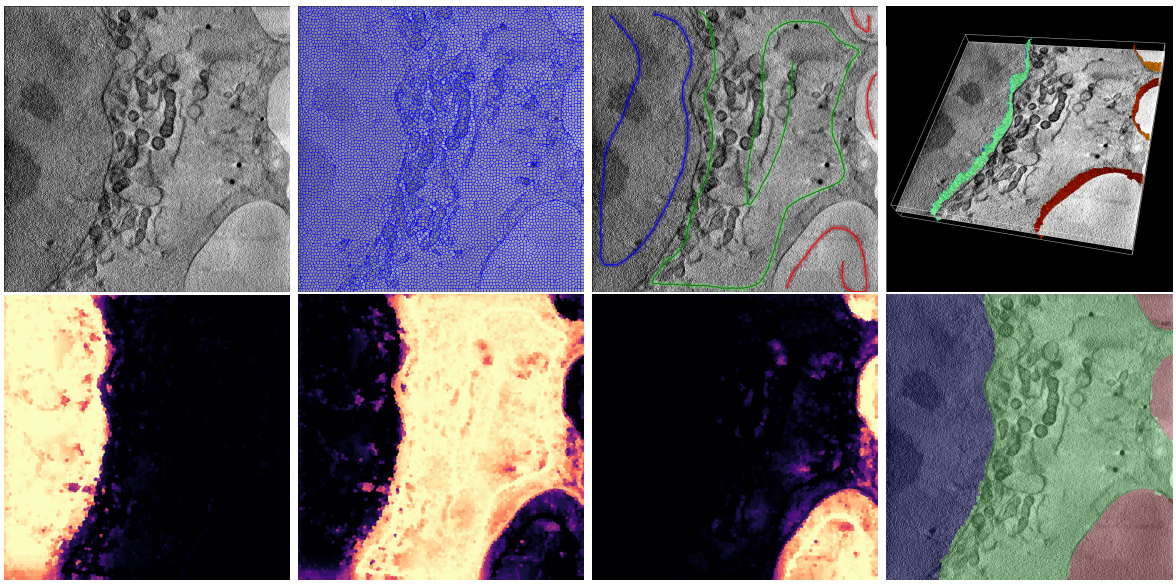

Figure 1. Overview of the algorithm. [**Top row**] user experience, from left to right: central 2D slice of target volume, supervoxel segmentation, user drawn scribbles (blue: nucleus, green: endoplasm/organelles, red: ice), final 3D segmentation showing boundaries between segmented areas. [**Bottom row**] intermediate class likelihoods (nucleus, endoplasm/organelles, ice) given by the ERF and final segmentation with the MRF model. (Note for visualization purposes, only central 2D slices are shown, but processes happen in 3D data)

5. INTERACTIVE CLASSIFICATION USING CONTEXTUAL LAYERS

Our hierarchical segmentation approach is based on Extremely Random Forests,[10] an ensemble of T Decision Trees. In Extremely Random Forests, the input is formed by a training samples $\mathbf{X}_{N\times K} = \boldsymbol{X}_1, \boldsymbol{X}_i \ldots, \boldsymbol{X}_N$ where $\boldsymbol{X}_i = \boldsymbol{\phi}_i^+$ and thus, N is the number of samples and K the number of features per sample. During the training step, each tree receives the full set $\mathbf{X}$ and, since the optimal tree training is NP-hard, trees are grown in a greedy manner. The root node selects $k = \sqrt{K}$ random split tests $\mathcal{S}$. A split test contains a randomly selected feature index j (from $\boldsymbol{\phi}$) and randomly sampled threshold t, which can be parameterised as $s = \{j, t\}$. Out of the k split criterions sampled, the best one is selected according to the information gain of the Gini criterion,[17] yielding the optimal split $s^* = \{j^*, t^*\} = max_{s\in\mathcal{S}} \quad G(\mathbf{X}, s)$ (where G stands for information gain). The node splits into a left and right node, and all the samples are split in two sets $\boldsymbol{X}^l = \{X_i \,|\, \forall i \in N\,,\, X_{i,j^*} \leq t^*\}$ and $\boldsymbol{X}^r = \{X_i \,|\, \forall i \in N\,,\, X_{i,j^*} > t^*\}$, which are passed to the left and the right node respectively. The process is repeated in the binary tree structure until a stopping criterion is reached, such as the information gain for the optimal split being lower than a threshold at a given node or a maximum depth is achieved. During testing, each test sample is fed to each of the T trees and routed through the root node to a leaf node (according to the optimal split tests learn during the training phase). The leaf node then, contains a class probability distribution of the nodes that during training reached that same node. The probability distributions returned by each of the trees are then added and normalized to give the final class-probability estimate for each of the samples.

Each trained tree is unique, as each of them selects the split criterion (both features and thresholds) randomly, and thus, despite that each individual tree grown greedily are not by themselves particularly robust, it is shown in[17] that an ensemble of T randomly trained trees achieves state-of-the-art classification rates and helps reduce the generalization error.

5.1 Hirarchical Contextual ERF

We use the Extremely Random Forests for interactive supervoxel classification in the following way. First, the user is asked to define a 2 layer hierarchy of labels with semantic information in a tree structure. The first layer should contain information about areas of the image the user is interested in, such as a *sky*, *vertical* and *horizontal* for a street image,[18] or *nucleus* and *endoplasm* for a more related 3D volumetric image. The second layer of labels aims to focus on the biology-specific labels appearing in each of the regions, such as *mitochondria* or other *vesicles* in the endoplasm, or *nucleoli* inside the nucleus. For each of the layers a random forest is trained and the probabilistic output of the first layer is used as features in the second layer, similar to the auto-context Random Forests,[13],[15] just with different training data in each layer. By doing so, it reduces the variability in the training of the second layer, as different classes with similar appearance in different regions of the image (as is usual in X-ray data) are spatially separated by the contextual information of the first layer, and thus, becomes easier for the classifier to discriminate between them.

5.1.1 First layer classification

In practice, the user is asked to draw a few scribbles using different colors for each of the labels. Those supervoxels marked by a scribble are tagged as ground truth and a standard Extremely Random Forest is trained with $T = 50$ trees and $max\_depth = 30$. For training and testing the $\mathbf{\Phi}^+ = \{\boldsymbol{\phi}_1^+, \boldsymbol{\phi}_k^+, \ldots, \boldsymbol{\phi}_n^+\}$ features from equation 5 formed by $\mathbf{f}_2$ (equation 7) are used. Note that these features include the absolute position of the scribbles x, y, z, and thus, the centroid coordinates of the training supervoxels are part of the training set. In other words, supervoxels of the volume will be biased towards their nearest scribble in the Euclidean space, but other factors such as volume intensity and first and second order derivatives will also play a role in the classification. In practice, this classification will be similar to calculating geodesic distances[19] from the scribbles to the volume, but with much lower computational cost. Since the desired output is a an approximate probability estimate of different image areas, these features provide an excellent compromise between complexity and robustness. Figure 1 shows the power of using a few annotations to spatially separate the cell in regions.

5.1.2 Second layer classification

In this second layer of the hierarchy, the user is again asked to draw some more specific scribbles, to tag different organs/organelles within the image (such us mitochondria/endoplasm or nucleus/nucleoli in our later example) and those supervoxels are marked as ground truth. This time, $\mathbf{f}_1$ features are chosen to represent the texture of voxels, and a modified *contextual* Extremely Random Forest with $T = 100$ trees is trained with $max\_depth = 50$.

This time each training sample i is represented by its feature vector $\boldsymbol{\phi}_i$ and the output class-probabilities of the previously trained spatial classifier (in layer 1) $\mathbf{p}_i = \{p(X_i = l) \,|\, \forall l \in \mathcal{L})\}$ with $\mathcal{L}$ the set of user defined labels. Each node then selects $t = k + d = \sqrt{K} + \sqrt{|\mathcal{L}|}$ splits tests, sampling k features from $\boldsymbol{\phi}_i$ and d features from the spatial probabilities $\mathbf{p}_i$. Thus, each node is able to choose between an appearance feature (textural feature) or a spatial density map. Classes with similar appearance will be spatially separated, whereas appearance will be useful to separate supervoxels with different textural features.

Figure 2 shows a diagram of the hierarchical interactive classification approach.

5.2 MRF-based segmentation refinement

To refine labels, contextual information is taken into account. The final global volume labelling problem is formulated as a minimization of a standard MRF energy function defined over the graph of supervoxel neighbours with labels $\mathbf{c} = \{c_i\}$:

$$E(\mathbf{c}) = \sum_{s_i \in SV} E_{data}(s_i, c_i) + \lambda \sum_{(s_i, s_j) \in \mathcal{N}} E_{smooth}(c_i, c_j). \tag{8}$$

Here, the data fidelity term (or unary potential) E_{data} is defined by the negative log likelihood of the output of the ERF, while the pairwise potential is defined as

$$E_{smooth}(c_i, c_j) = W_{i,j}\, exp(\sigma ||\mathbf{f}_{1_i} - \mathbf{f}_{1_j}||_2)\ \mathbf{C}[c_i, c_j] \tag{9}$$

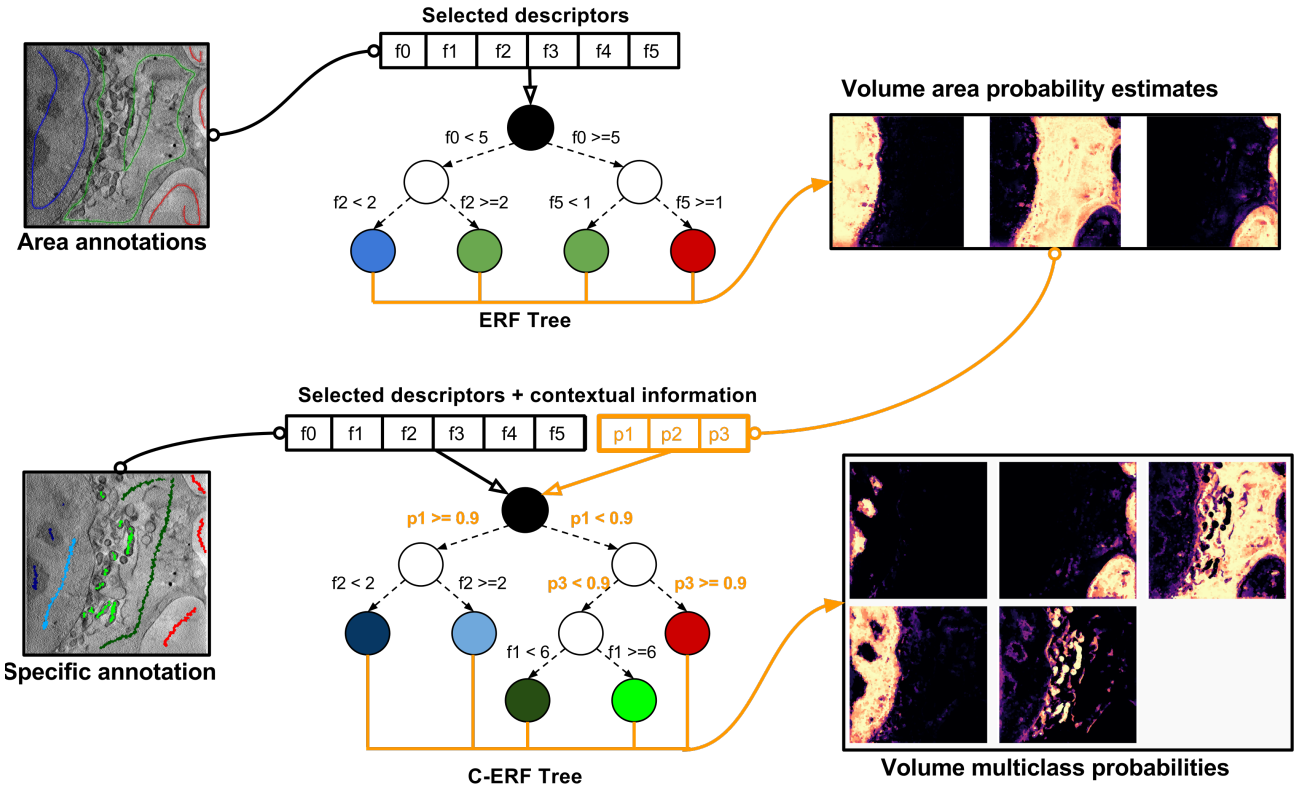

Figure 2. Overview of the hierarchical classification. The predictions of volume areas are used as features for the 2nd layer's contextual ERF (C-ERF), which at each node chooses the best split criterion among the features or probabilities.

which is a standard Gaussian similarity kernel with a label based cost matrix $\mathbf{C}$ and weighted by $W_{i,j}$, a weighting factor depending on the amount of boundary pixels that share supervoxels s_i and s_j, defined as

$$W_{i,j} = \frac{min(t, b_{i,j})}{t} \tag{10}$$

where $\mathbf{b} = \{b_{i,j}\}$ are the boundaries between supervoxels s_i and s_j, $t < \hat{\mathbf{b}}$ is a truncation parameter to avoid extremely large boundaries to unbalance the weighting, and $\hat{\mathbf{b}} = max(\mathbf{b})$ is the maximum amount of boundaries shared between two supervoxels. Therefore, $W_{i,j}$ is then a weighting factor in the range $[0, 1]$.

Finally, we define the label cost matrix $\mathbf{C}$ using the hierarchical label information (instead of using the standard Potts model). Here, we empirically proposed label costs $c1$, $c2$ and $c3$ with increasing cost for our hierarchical labels. Figure 3 shows how the 5×5 label cost matrix is weighted. Being $0 < c1 < c2 \ll c3$, costs are empirically set to $c1 = 1$, $c2 = 5$ and $c3 = 100$. Figure 3 shows an example setup of the matrix cost $\mathbf{C}$.

The MRF is then in a form that can be efficiently approximated by QPBO with α-expansion.[20]

6. EXPERIMENTAL EVALUATIONS

We empirically test our segmentation pipeline over 3 different $946 \times 946 \times 200$ cryo-SXT datasets.[21] These datasets contain the following classes of interest: nucleus, nucleoli, endoplasm, mitochondria and ice. For their segmentation, labels are separated in 2 stages, for the first one, nucleus, endoplasm and ice regions are used to draw contextual information (as shown in Figure 1). For the second stage, the user is asked to annotate mitochondria, nucleoli and other nucleus, endoplasm and ice supervoxels to train a ERF to learn how to discriminate between them, using both appearance features and contextual information from their position. Figures 1 and 4 show a qualitative evaluation of our approach, showing 3D volume renderings of obtained segmentations.

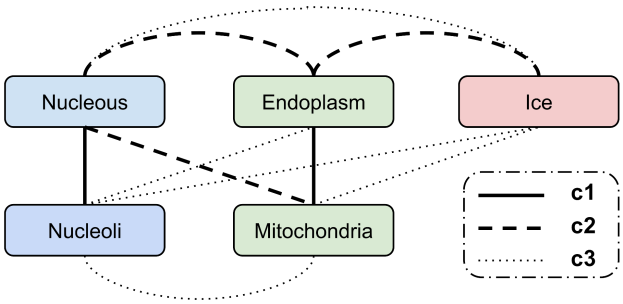

Figure 3. Creating the label cost matrix $\mathbf{C}$ from label structure. $0 < c_1 < c_2 \ll c_3$.

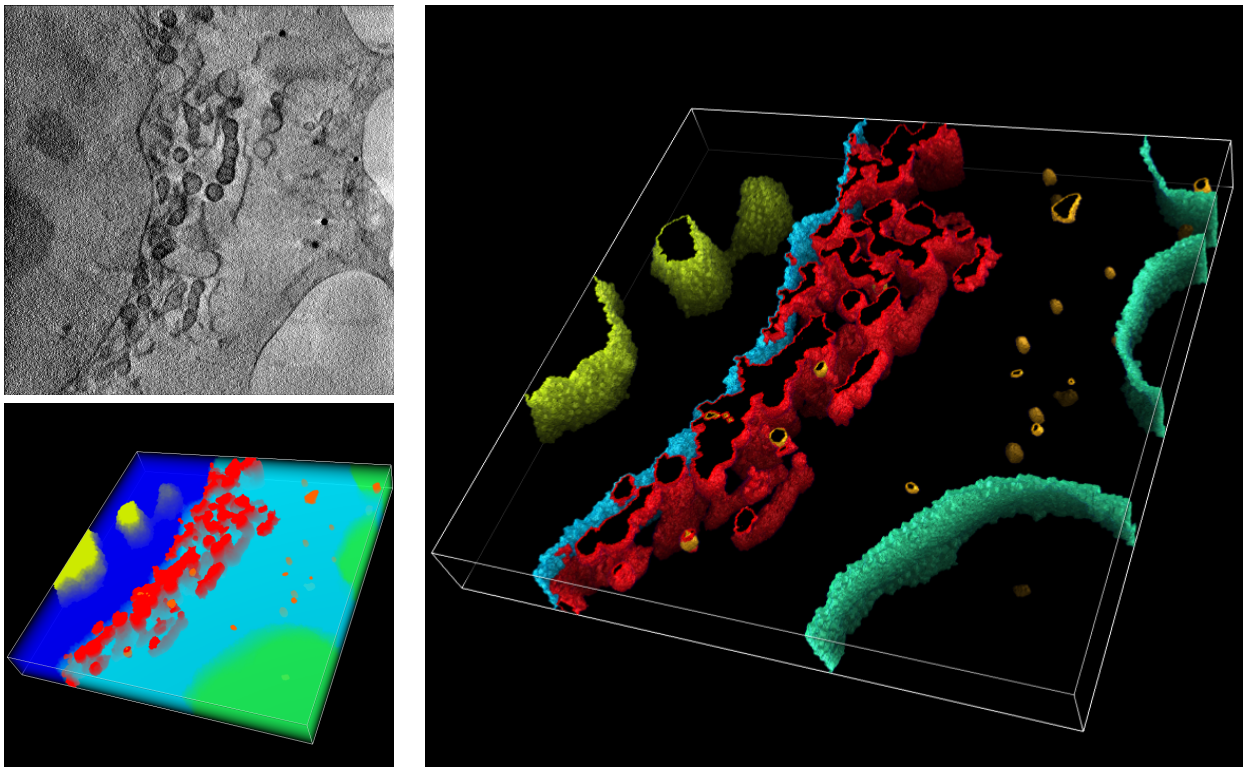

Figure 4. Further cell segmentation with contextual information and MRF-based refinement. The green region from figure 1 was segmented again, allowing for resolution of even smaller, harder-to-distinguish objects (mitochondria: red, nucleoli: yellow and feducials: orange). Feducials where previously extracted by following a basic intensity thresholding approach (as they completely absorb X-rays and can easily be detected as spherical outliers).

For validation purposes, we have manually segmented 10 uniformly separated slices (1 every 20 slices) to demonstrate the segmentation accuracy (in terms of the per-supervoxel label) of our algorithm compared to a baseline RF trained in the whole data, as shown in table 1. We trained an ERF using different splits of the manually segmented 10 slices as ground truth data. The splits are formed of randomly selected 1%, 10%, 30%, 50%, 70% and 90% of the data for training and the rest for testing respectively with the aim of detecting the amount of ground truth needed (using textural sigma set features) to segment a volume. To perform the split, we used a Stratified Split, where the percentages of data for training and testing are taken on a per-class basis, meaning that 1% represents 1% of samples for each of the classes are used for training, and the rest for testing. Table 1 shows the result, indicating that overall around 30% of the data (which is the 30% of 10 slices and can be obtained with a couple of minutes of user scribbles) are enough to get a good classification for most of the volume. The table contains results for a standard ERF with sigma set features in the first section, while the second sections shows results of our contextual ERF assuming the output of the first layer of the hierarchy

is known (the separation between nucleus, endoplasm and ice is known). Thus, it can be seen that adding contextual (position) information to the second layer results in an performance increase as with only 10% of training data already achieves overall better results than using 50% of the data without the contextual split.

	Overall	Mitochondria	Endoplasm	Nucleus	Nucleoli	Ice
1%	0.8704	0.5655	0.9349	0.9597	0.3626	0.8760
10%	0.9319	0.6108	0.9630	0.9594	0.8200	0.9404
30%	0.9513	0.7070	0.9714	0.9662	0.9029	0.9707
50%	0.9561	0.7352	0.9759	0.9676	0.9080	0.9779
70%	0.9589	0.7632	0.9750	0.9716	0.9233	0.9810
90%	0.9638	0.8096	0.9799	0.9705	0.9150	0.9893
	Assuming contextual information is known					
1%	0.9569	0.6316	0.9709	0.9593	0.8530	0.9885
10%	0.9601	0.7314	0.9702	0.9767	0.8756	0.9911

Table 1. Textural SigmaSet feature performance over different classes with different training splits. 10 slices has been manually labelled for each of the datasets. Left column indicates split percentages in a per-class train-test split, meaning the percentage of training samples from each class used for training (while the remaining is uses for testing). Note that the datasets are unbalanced and 1% of nucleus is not the same amount of training data than 1% of mitochondria. Overall score shows per-volume test scores.

Last, in table 2 we provide timings of different stages of our algorithm using a standard i5 processor with a K40 Tesla GPU (for CUDA implementations). As can be seen, after supervoxels and features are computed, the pipeline provides a near real-time experience for the user, as the classification requires around a second to predict the class of all the supervoxels of the volume, as individual trees of the ERF are trained in parallel.

1. Supervoxel oversegmentation	∼10 sec
2. Descriptor Extraction	∼3 sec
3. User interaction (scribbles)	-
4. Train+predict ERF	1 sec
5. Refine segmentation with MRF	∼5 sec

Table 2. Computational times for a $1000 \times 1000 \times 200$ volume. Note that the user can loop steps $3-5$ hierarchically.

7. CONCLUSIONS

We have shown that using supervoxels for segmentation of a challenging X-ray dataset with computationally efficient Sigma Set features can quickly achieve good classification accuracy. Additionally, a 2-stage training approach by incorporating predictions of spatial probabilities from quick sketch annotations enhances the classification accuracy of the final segmentation, with a similar effect to the auto-context RF approach. And last, we propose a weighting for the supervoxels in the MRF refinement based in their shared boundaries and adding a label cost based on the user knowledge (the label hierarchy) to improve the quality of the MRF-based label refinement.

8. ACKNOWLEDGES

We gratefully acknowledge Diamond Light Source for jointly funding Imanol Luengo under PhD STU0079 and the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research. The authors also thank Dr. Kyle Dent and Dr. E Duke (Diamond Light Source) for collecting the cryo soft X-ray tomography data and Dr. Minoo Razi and Dr. Lucy Collinson at The Francis Crick Institute formerly the EM Unit, London Research Institute, Cancer Research UK for the provision of the samples (DOI: 10.1111/jmi.12139).[21]

REFERENCES

[1] Mitchell, S. C., Bosch, J. G., Lelieveldt, B. P., Van der Geest, R. J., Reiber, J. H., and Sonka, M., "3-d active appearance models: segmentation of cardiac mr and ultrasound images," *Medical Imaging, IEEE Transactions on* **21**(9), 1167–1178 (2002).

[2] Cabezas, M., Oliver, A., Lladó, X., Freixenet, J., and Cuadra, M. B., "A review of atlas-based segmentation for magnetic resonance brain images," *Computer methods and programs in biomedicine* **104**(3), e158–e177 (2011).

[3] Cohen, L. D. and Cohen, I., "Finite-element methods for active contour models and balloons for 2-d and 3-d images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **15**(11), 1131–1147 (1993).

[4] Gao, Y., Kikinis, R., Bouix, S., Shenton, M., and Tannenbaum, A., "A 3d interactive multi-object segmentation tool using local robust statistics driven active contours," *Medical image analysis* **16**(6), 1216–1227 (2012).

[5] Li, C., Xu, C., Gui, C., and Fox, M. D., "Distance regularized level set evolution and its application to image segmentation," *Image Processing, IEEE Transactions on* **19**(12), 3243–3254 (2010).

[6] Lucchi, A., Smith, K., Achanta, R., Knott, G., and Fua, P., "Supervoxel-based segmentation of mitochondria in em image stacks with learned shape features," *Medical Imaging, IEEE Transactions on* **31**(2), 474–486 (2012).

[7] Vazquez-Reina, A., Gelbart, M., Huang, D., Lichtman, J., Miller, E., and Pfister, H., "Segmentation fusion for connectomics," in [*Computer Vision (ICCV), 2011 IEEE International Conference on*], 177–184, IEEE (2011).

[8] Sommer, C., Straehle, C., Koethe, U., Hamprecht, F., et al., "ilastik: Interactive learning and segmentation toolkit," in [*Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*], 230–233, IEEE (2011).

[9] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Susstrunk, S., "Slic superpixels compared to state-of-the-art superpixel methods," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **34**(11), 2274–2282 (2012).

[10] Geurts, P., Ernst, D., and Wehenkel, L., "Extremely randomized trees," *Machine learning* **63**(1), 3–42 (2006).

[11] Ren, C. Y. and Reid, I., "gslic: a real-time implementation of slic superpixel segmentation," *University of Oxford, Department of Engineering, Technical Report* (2011).

[12] Hong, X., Chang, H., Shan, S., Chen, X., and Gao, W., "Sigma set: A small second order statistical region descriptor," in [*Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*], 1802–1809, IEEE (2009).

[13] Tu, Z. and Bai, X., "Auto-context and its application to high-level vision tasks and 3d brain image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **32**(10), 1744–1757 (2010).

[14] Tu, Z., "Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering," in [*Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*], **2**, 1589–1596, IEEE (2005).

[15] Montillo, A., Shotton, J., Winn, J., Iglesias, J. E., Metaxas, D., and Criminisi, A., "Entangled decision forests and their application for semantic segmentation of ct images," in [*Information Processing in Medical Imaging*], 184–196, Springer (2011).

[16] Kocak, A., Cizmeciler, K., Erdem, A., and Erdem, E., "Top down saliency estimation via superpixel-based discriminative dictionaries," in [*Proceedings of the British Machine Vision Conference. BMVA Press*], (2014).

[17] Breiman, L., "Random forests," *Machine learning* **45**(1), 5–32 (2001).

[18] Tighe, J. and Lazebnik, S., "Superparsing: scalable nonparametric image parsing with superpixels," in [*Computer Vision–ECCV 2010*], 352–365, Springer (2010).

[19] Kontschieder, P., Kohli, P., Shotton, J., and Criminisi, A., "Geof: Geodesic forests for learning coupled predictors," in [*Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*], 65–72, IEEE (2013).

[20] Boykov, Y., Veksler, O., and Zabih, R., "Fast approximate energy minimization via graph cuts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **23**(11), 1222–1239 (2001).

[21] Duke, E., Dent, K., Razi, M., and Collinson, L. M., "Biological applications of cryo-soft x-ray tomography," *Journal of Microscopy* **255**(2), 65–70 (2014).