UCL
Université catholique de Louvain

Pôle d'imagerie moléculaire, radiothérapie et oncologie
Institut de recherche expérimentale et clinique

Machine Learning group
Institute of ICTEAM

Incremental organ segmentation with machine learning techniques

Application to radiotherapy

Guillaume Bernard

Thesis presented for the Ph.D. degree
in engineering sciences

PhD committee : John A. Lee – Supervisor
Vincent Grégoire
David Sarrut
Jean-Philippe Thiran
Michel Verleysen – Chairman

2014

Résumé

La radiothérapie est une modalité de traitement des cancers qui est utilisée chez plus de cinquante pour cent des patients. Elle peut être considérée comme un problème balistique où une cible (la tumeur) doit être irradiée tout en préservant les organes sains aux alentours. Avant de commencer, un traitement en radiothérapie est planifié sur base d'une image tomographique du patient. Le radiothérapeute dessine sur l'image les contours des organes à risque, de la tumeur et des zones ganglionnaires. La précision de ces contours est cruciale pour administrer un traitement optimal. Toute imprécision risque de diminuer la probabilité de contrôle local de la tumeur ou d'accroître les effets secondaires du traitement. Or, en pratique, le dessin manuel est une opération longue, répétitive et sujette à une certaine variabilité intra- et inter-observateur.

Cette thèse a pour objectif de fournir de nouveaux outils pour faciliter la délimitation des organes à risques lors de la planification du traitement. Pour ce faire, une méthode automatique de segmentation et d'identification des organes à risque est proposée. En automatisant la partie la plus répétitive du dessin des organes, le radiothérapeute peut se concentrer sur les points les plus critiques : la tumeur et les zones ganglionnaires.

Actuellement, les méthodes de segmentation automatique sont encore rarement utilisées. Lorsqu'elles le sont, elles reposent souvent sur des techniques de recalage non-rigide. Cette approche utilise des images déjà annotées par des experts, appelées atlas, et les déforme pour les faire correspondre à l'image du patient à traiter. Après le recalage, les contours de l'image déformée sont transférés vers la nouvelle image. La qualité des contours obtenus dépend directement de celle de la déformation, qui est critique et difficile à contrôler. En effet, le recalage fait intervenir des modèles de régularisation du champ de déformation dont la paramétrisation est complexe, surtout dans les cas inter-patients.

Nous proposons une méthode qui n'est pas basée sur le recalage déformable, mais sur l'apprentissage automatique (*machine learning*). Les images sont d'abord sur-segmentées afin de déterminer des zones homogènes et d'éviter de traiter chaque pixel indépendamment. De plus, nous faisons l'hypothèse que des pixels voisins ayant la même intensité font partie du même organe. Chaque organe est ainsi composé d'une ou plusieurs zones. Pour établir l'appartenance d'une zone à un organe, nous proposons de travailler de manière incrémentale. Les organes sont identifiés les uns après les autres à l'aide de techniques de classification et ceux déjà identifiés participent à l'identification des suivants. Nous montrons que cette approche incrémentale est pertinente et qu'elle permet

d'améliorer les performances de classification. Néanmoins, pour obtenir des résultats optimaux, une bonne séquence d'identification est nécessaire. Deux méthodes sont présentées dans cette thèse pour déterminer une bonne séquence de classification d'organes à partir de quelques images annotées. Nous montrons que ces méthodes conduisent à de bonnes performances d'identification. Nous montrons également que les erreurs produites par l'approche incrémentale peuvent facilement et rapidement être corrigées par un opérateur humain. Enfin, notre méthode étant générique, elle peut être adaptée pour toutes les régions du corps humain.

Summary

Radiotherapy is a cancer treatment modality that is used for more than fifty percent of patients. It can be considered as a ballistic problem where a target (the tumour) must be irradiated while sparing the surrounding healthy organs. Before it starts, a radiotherapy treatment must be planned on a tomographic image of the patient. The radiation oncologist draws on the image the contours of the organs at risk, tumour and nodal areas. The accuracy of these contours is crucial to deliver an optimal treatment. Any inaccuracy might decrease the probability of local tumour control or increase the undesired side effects of treatment. In practice, though, manual drawing is a quite lengthy, repetitive operation, which is moreover subject to a certain intra- and inter-observer variability.

This thesis aims at providing new tools to facilitate the delineation of organs at risk in the treatment planning step. For this purpose, an automatic method of segmentation and identification of organs at risk is proposed. By automating the most repetitive part of organ drawing, the radiation oncologist can concentrate on the most critical points : the tumour and the nodal areas.

Currently, automatic segmentation methods are still seldom used. When they are, they often rely on image registration techniques. That approach uses images that have already been annotated by experts, called atlases, and deforms them so that they match the new image of the patient to be treated. After registration, the contours on the deformed image are transferred to the new image. The quality of the obtained contours depends directly on that of the deformation, which is critical and difficult to check. Registration entails indeed regularisation models of the deformation field, whose parameters are complex to adjust, especially in the inter-patient cases.

We propose a method that involves no deformable registration ; instead, it relies on machine learning (i.e., automated statistical model inference). The images are first over-segmented in order to determine homogeneous areas and to avoid processing each pixel independently. Moreover, we assume that neighbouring pixels with similar intensity are both part of the same organ. Each organ is thus comprised of one or several such areas. To determine the organ each area belongs to, we propose to work in an incremental way. The organs are identified one after the other, thanks to classification techniques ; those that are already identified contribute to identifying the next ones. We show that this incremental is effective and allows improving the organ classification accuracy. However, in order to get optimal results, a good sequence of identification is necessary.

Two methods are presented in this thesis to determine such a good sequence of organ classification, starting from a few annotated images. We show that these methods lead to high performances of identification. We also show that errors made by the incremental approach can be easily and quickly corrected by a human operator. Finally, our method being generic, it can be adapted to all regions of the human body.

Remerciements

C'est la fin. Ces quatre dernières années sont passées à une vitesse incroyable. Avec ces quelques lignes, je voudrais remercier les personnes qui m'ont aidé d'une façon ou d'une autre à réaliser ma thèse. Cette thèse, c'est aussi un peu la vôtre.

Pour commencer, je souhaiterais remercier mon promoteur John Lee. Merci de m'avoir accueilli il y a un peu plus de quatre ans. Je me souviens de cette première visite du labo, lorsqu'on a discuté d'un sujet de thèse et que tu m'as dis que tu ne pouvais pas me promettre des résultats à la fin. Nous y voici. J'espère qu'on a réussi à découvrir une partie des possibilités que le sujet offrait. Merci pour ta disponibilité. Il me suffisait de parler fort depuis mon bureau pour te poser une question et obtenir une réponse immédiate. Merci pour tes connaissances dans de multiples domaines que tels que le machine learning, l'imagerie, la physique, la radiothérapie, les Monty Python et les autres éléments de la culture geek.

Un merci particulier à Michel Verleysen qui a participé à mon encadrement pendant ces années. Merci de m'avoir présenté John lorsque je cherchais un sujet de thèse et de nous avoir ensuite supervisés quand on était (un peu) dans le flou (surtout administratif, parce qu'on gérait quand même). Je garde en tête beaucoup de souvenirs des différentes éditions de l'ESANN à Bruges, nos repas en groupe et nos stress d'organisateurs (même si on stressait beaucoup moins que toi). C'était un plaisir de participer et d'aider à l'organisation de ces conférences.

Je tiens également à remercier Vincent Grégoire qui m'a accueilli au sein de son laboratoire à Woluwé. Comme vous avez parfois pu le constater en passant dans notre bureau, nous, ingénieurs informaticiens, sommes parfois des gens étranges. Quel plaisir de travailler dans votre laboratoire regroupant de multiples disciplines.

Je souhaiterais également remercier David Sarrut et Jean-Philippe Thiran qui, en tant que membres du jury, ont pris le temps de lire et de critiquer mon manuscrit. Merci pour vos questions et remarques pertinentes. Elles ont grandement contribué à l'amélioration du manuscrit.

Je voudrais ensuite remercier Pierre Dupont qui m'a donné l'envie de faire du machine learning lors de mon master et pendant mon mémoire. Merci également d'avoir participé à mon comité d'encadrement. Vos questions et votre rigueur ont grandement contribué à la qualité de ma recherche.

En écrivant ces quelques lignes, je pense aussi à mes collègues du machine

learning group : Adrien, Alexandra, Benoît, Dimitri, Émilie, Jérôme et Samuel. Ce fut un plaisir d'échanger avec vous sur nos problèmes et sur les nouveautés en machine learning. Amusez-vous bien à Bruges l'année prochaine (même si je ne serai pas là).

En quittant Louvain-la-Neuve après mon mémoire pour Woluwé, j'ai découvert une autre partie de l'UCL. Travailler dans un laboratoire multidisciplinaire fut très enrichissant personnellement et professionnellement. J'ai tout de suite été très bien accueilli par Fiona, Marie et Samuel. Merci pour tous ces moments à parler de voyages et de choses de la vie. J'espère avoir été aussi bon que vous pour accueillir Ana et Kevin. Essayez quand même de décorer un peu le bureau une fois que je serai parti. ;) Je voudrais remercier Kevin pour l'instauration des "vendredis bidouille". Merci aux occupants des autres bureaux Anne, Dario, Edmond, Jefferson, Karolin, Sarah, Séverine et Vanesa. J'ai passé de très bons moments avec vous, en faisant des brainstormings dans des domaines où mes compétences sont limitées ou en discutant de tout et de rien pendant les temps de midi. J'ai une pensée pour nos émérites qui sont toujours au travail et qui viennent de temps en temps nous poser des questions (souvent liées à l'informatique).

Je souhaiterais également remercier mes amis geeks de Louvain-la-Neuve : Fanfwè, JayBe, Jey, Karim, Nico, Romain, Sam et Xa. Boire des coups avec vous tout en discutant de nos thèses a toujours été un plaisir. Merci aussi pour tous les emails échangés, bien qu'ils aient parfois affecté mon efficacité.

Je voudrais remercier mes coéquipiers du club de football de Loyers qui m'ont permis de me changer les idées après les journées difficiles. Les troisièmes mi-temps et les fermetures de vestiaire après les entraînements n'ont pas toujours été reposantes.

Merci à toute ma famille qui m'a soutenu et s'est intéressée à mes travaux. J'ai certainement du être barbant par moment, mais je vous remercie de ne pas me l'avoir dit. Merci aussi de m'avoir aidé au début de ma thèse, lorsque je déménageais tous les ans de coloc en coloc.

Je terminerai pas un énorme merci à Caro (parce que "Caroline", ça sonne bizarre dans ma bouche). Merci de m'avoir supporté quand j'étais insupportable. Merci pour tout le temps que tu m'as consacré pour les répétitions des présentations orales, pour la relecture du manuscrit... Tu as géré, mais c'est parce qu'on est quand même des sacrés L. !

Contents

Acronyms

kNN

k nearest neighbours (see 2.2.2) – Classification technique based on the hypothesis that similar instances have to belong to the same class.

BCR

Balanced classification rate (see 3.1.4) – Metric used to correctly measure the classification error without bias when the classes are unbalanced (different number of instances in each class).

CT

Computed tomography – Image modality used to acquire a three-dimensional, anatomical image of a patient with the use of an X-ray source and sensors.

CTV

Clinical target volume – Anatomical concept. Volume of tissue that contains the GTV and/or subclinical microscopic malignant disease, which has to be eliminated.

DNA

Deoxyribonucleic acid – Molecule that encodes the genetic instructions used in the development and functioning of all known living organisms.

GPA

Generalised Proscrutes alignment – Method that applies the Procrustes analysis to superimpose a set of shape (optimally translate, rotate and scale the shapes).

GTV

Gross tumour volume – Anatomical volume. Gross, palpable, or visible/demonstrable extent and location of a malignant growth.

HU

Hounsfield units – Quantitative scale for describing radiodensity. Hounsfield units measure the attenuation of an X-ray source by the matter. The attenuation of water is set at 0 HU, while the air is -1000 HU. In the human body, soft tissues range roughly from -120 HU to 80 HU and bones are beyond 400 HU.

MRI

Magnetic resonance imaging – Medical imaging technique that produces two- or three-dimensional image of the patient anatomy by using the magnetic resonance properties of protons.

OAR

Organs at risk – Normal tissues whose radiation sensitivity may significantly influence treatment planning and/or prescription of the radiation dose.

PCA

Principal component analysis – Statistical procedure that decorrelates a set of observations of possibly correlated variables.

PET

Positron emission tomography – Functional imaging technique that produces a three-dimensional image of the distribution of a radiotracer in the body.

PTV

Planning target volume – Geometrical concept. Defined to select appropriate irradiation parameters, taking into consideration the net effect of all the possible geometrical variations and inaccuracies in order to ensure that the prescribed dose is actually delivered in the CTV.

RF

Random forest (see 2.2.2) – Classification techniques using an ensemble of randomised decision trees grown from a random sampling of the training set.

SAM

Statistical appearance model (see 2.3.2) – Statistical models of the shape and appearance of an object, which is iteratively deformed to fit some instance of this object in a new image.

SSM

Statistical shape model (see 2.3.2) – Statistical models of the shape of an object, which is iteratively deformed to fit some instance of this object in a new image.

SVM

Support vector machines (see 2.2.2) – Classification technique that finds the optimal hyperplane separating the classes (i.e. with the largest margin).

TV

Target volume – Tissues that are to be irradiated with a specified dose.

Chapter 1

Introduction

1.1 Medical Context

Cancer is an uncontrolled growth of a group of cells that can effect healthy cells. In the most developed countries, about one in three males and one in four females will develop cancer before their 75th birthday (Jemal et al. 2011). Even if treatment methods have been already improved for these last years, they have to be enhanced further. Improvement in the different treatment modalities led to an increase of the five-year survival rate in the past decade. Among those treatments, radiotherapy is widely used in current practice, together with chemotherapy and surgery. It has been established that more than 50% patients with cancer should be treated with radiotherapy (Delaney et al. 2005). Radiation oncology aims at eliminating cancerous cells in the tumour and preventing their spread in the patient body. Radiotherapy thus plays a major role in local tumour control, preventing recurrence by controlling locally the growth of the unhealthy cells (van der Kogel and Joiner 2009).

1.2 Radiotherapy

In this section, the principle of the radiation therapy is briefly explained and the different stages involved in the planning of a radiotherapy treatment are detailed. A specific focus is set on the delineation of organs on medical images.

1.2.1 Principle

Radiation oncology relies on the fact that ionizing radiations damage the deoxyribonucleic acid (DNA) of the cells (Ward 1988). DNA contains genetic instructions necessary for the development and functioning of the cell. Cells are naturally programmed to correct damaged DNA up to a certain degree. If the deterioration is too important, cell dies. However, it has been demonstrated that healthy cells recover better than cancerous cells when are exposed to degradation (van der Kogel and Joiner 2009). Radiotherapy exploits the radiobiological difference between healthy and cancerous cells. Within a given area mainly composed of cancerous cells, properly repeated irradiation lead to the death of cancerous cells, while surrounding healthy cells stay alive. In practice, the most common treatment involves irradiating the tumour every day during 5 to 7 weeks.

1.2.2 Flowchart of a radiotherapy treatment

Radiotherapy can be seen as a ballistic problem in which a target has to be be hit while avoiding its surroundings. A trade-off has to be reached to maximise irradiation of the unhealthy cells and preserve the healthy ones. In order to solve this optimisation problem, the radiotherapy treatment is often organised in two phases which are the planning and the delivery. During the planning phase, the images are acquired, the regions of interest are identified and the ballistic problem is solved for the acquired data. The planned treatment is then delivered to the patient.

Once a patient is diagnosed with cancer that requires radiotherapy treatment, the planning phase starts. Planning is divided in five stages involving different specialities. These stages are sequentially executed, each stage using the results obtained from the previous ones (see Figure 1.1).

Imaging. In this stage, an operator acquires a three-dimensional, anatomical image of the patient with a modality called computed tomography (CT). To acquire this image, an X-ray source and several X-ray sensors rotate around the patient. The source and the sensors are placed face to face both sides of the patient. During the rotation, several X-ray images of the patient from different angles are acquired. Those images, also called projections, are used to reconstruct the 3D image of the patient. Each pixel of the CT image measures the attenuation of the X-ray source. This attenuation is expressed in Hounsfield

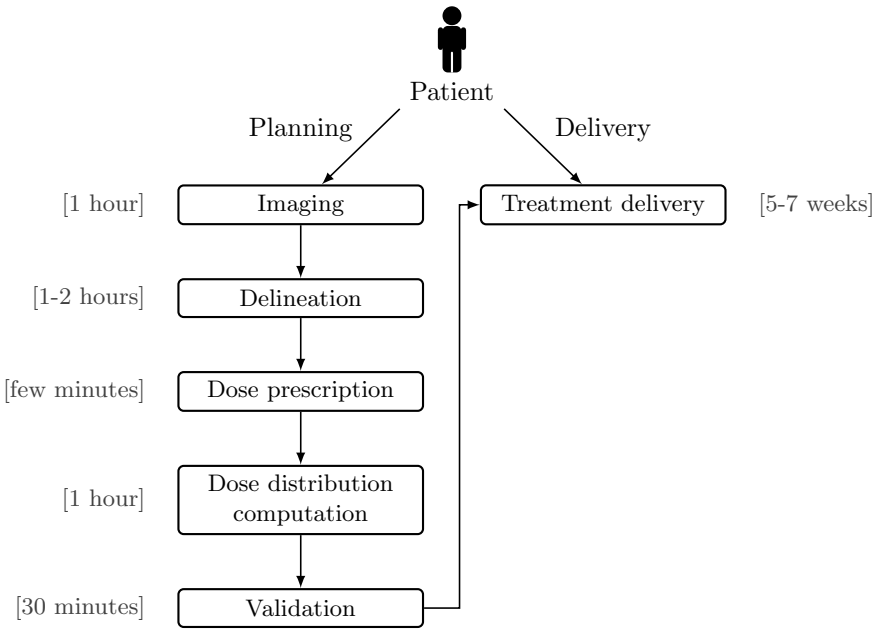


Figure 1.1 – Flowchart of a common radiotherapy treatment.

units (HU). Water has an attenuation of 0 HU, while the air is −1000 HU. In the human body, soft tissues range roughly from −120 HU to 80 HU and bones are beyond 400 HU. In some cases, radiocontrast agents can be injected in the patient to improve the visibility of area of interest. Those contrast media modify the observed values of HU. The CT image is systematically acquired as it gives an estimation of the electronic density of the anatomy, which is required to compute the dose distribution in the patient body. As this modality is affected by a lack of contrast between soft tissues, other images have sometimes to be acquired. Depending of the cancer type, images such as positron emission tomography (PET) and magnetic resonance imaging (MRI) can be prescribed.

Delineation. Based on the acquired images, the physician determines the position of the target volumes (TVs) as well as the position of some specific organs, the so-called organs at risk (OARs). To determine the position of the TVs, the physician defines the borders of regions of interest on the image, also called delineation, that corresponds to the gross tumour volumes (GTVs). It is usually performed by drawing contours on two dimensional (2D) slices

extracted from the 3D CT. The delineated region of interest, is made up of several 2D shapes from different slices of the image. As there are assumptions of microscopic spread of cancerous cells around the tumours, margins are added around the GTV. The new volume, called clinical target volume (CTV), takes into account cancerous cells that may not be seen on the image. A third volume, the planning target volume (PTV), is created as an extension of the CTV and takes into account the uncertainties in planning and treatment delivery. It is a geometric volume designed to ensure that the prescribed radiation dose is correctly delivered to the CTV. The OARs have to be delineated to ensure that they do not receive a higher-than-safe dose. There exists different specifications for each OAR. In some cases, as for the PTV, an extra margin is added around the OARs to take into account the uncertainties. Depending on the localisation of the tumour, the delineation stage can take up to 2 hours.

Dose prescription. During the third stage, the physician evaluates the tumour propagation in the patient body by using staging system such as "tumour–nodes–metastasis" (TNM) and makes the appropriate prescription. The prescription includes, among others, the number of fractions and the dose the tumour has to receive. Those prescriptions should follow the recommendations made by the International Commission on Radiation Units and Measurements (ICRU) (reports ICRU 50, ICRU 62 and ICRU 83).

Dose distribution computation. The delineated images and the prescriptions are then given to the physicist who computes the dose distribution. This computation is the fourth stage of planning. The physicist tries to find the best trade-off between maximising the dose on the PTV and preserving the OARs. The quality of the dose distribution depends on the equipment, the time spent on the problem and the expertise of the physicist. The possibilities of optimisation change according to the equipment. Some radiation therapy machines offer more capabilities than others. As for many optimisation problem, the more time you spend on a problem, the more likely you are to find a better solution. This is why the physicist with high level expertise in guiding the optimisation algorithm may save time and may find a better optimum.

Validation. Once the dose distribution is satisfactory, it is given to the physician for validation. He checks whether all the requirements are respected. If needed, a new dose distribution is computed by the physicist.

In a usual treatment, the patient is planned once and is irradiated several times (e.g. five times a week for five to seven weeks). Each day of treatment, the patient receives a fraction of the planned dose. In the case of adaptive radiotherapy, his/her treatment is replanned during the treatment to avoid wrong dose delivery.

1.2.3 Focus on delineation

Among all stages of planning, TV and OAR delineations are critical steps since they define all the areas of the image to irradiate or avoid. This task is usually done manually by the physician. However, it has been shown that the manual delineation leads to different delineations when they are done by different experts. This is called the inter observer variability. Moreover, it has been shown that an expert delineating the same image several times obtains different delineations. This second type of variability is named intra-observer variability. These variabilities have been reported for both TVs and OARs (Cazzaniga et al. 1998; Tai et al. 1998; Yamamoto et al. 1999; Caldwell et al. 2001; Hurkmans et al. 2001; Pitkänen et al. 2001; Steene et al. 2002; Weiss et al. 2003; Struikmans et al. 2005; Wong et al. 2006; Landis et al. 2007; Petersen et al. 2007; Li et al. 2009).

For the inter-observer variability, the main reason often reported is the lack of guidelines to draw TVs and OARs. Indeed, the quality of the images as well as the human interpretation of the images make the delineation of the regions of interest a complicated task. Experts with different formations are likely to perform different delineations. The insufficient formalisation of the task has led to the creation of guidelines. Guidelines help to standardise the method of delineation, based on anatomical and biological knowledge. The importance of guidelines has been highlighted in the last decade and radiation oncologists are better informed about their importance.

The technology used by the physician can also be a source of problem. Because the physician works on a screen, he can only see one 2D slice of the 3D image at a time. He has to switch from one slice to another to have a partial 3D view of the patient. Moreover, a human is able to differentiate only from 700 to 900 shades of gray simultaneously on a well calibrated medical display system (Kimpe and Tuytschaever 2007). To circumvent the limitations of the human vision, the physician works within a window that is narrower than the total range of gray intensity. Even if he is able to apply intensity level windowing to improve the contrast on a desired area, it is noteworthy that the

physician has limited tools to view the patient's anatomy in 3D. The lack of effective visualisation tools may jeopardise the correct human interpretation of the image.

A last, an explanation of the variability in the delineation can be the length and the repetition of the task. Indeed, the delineation of regions of interest is a very repetitive task. Repeating the same task can be tiresome and leading to mistakes. Improving the guidelines and using several image modalities can reduce the variability but, in the end, the task will remain long and repetitive causing a risk of delineation errors.

There exist alternative methods such as atlases and statistical shape models, in order to automate the delineation (described in Section 2.3). All those methods use delineated images to performed an automatic delineation of new images. Nevertheless, most of those methods can still suffer from a lack of accuracy and are sometimes rather slow. This is why they are rarely applied in practice. Moreover, those methods are not always able to generalise the characteristics of the human body and are very dependent of their parameters and the set of delineated images they used.

1.3 Motivation, hypotheses, and objectives

This thesis aims at developing a new, fast, and automatic delineation technique. We wanted this approach to be at least as consistent and reproducible as physicians. We also want the technique to be enough generic so that only the preprocessing and postprocessing need to be adapted to the delineated region.

In order to develop the technique, some assumptions have been made:

- Nearly every patient of the same sex has the same organs independently to the ethnic backgrounds. There exists some exceptions such as patients with supernumerary body parts and patients with ablated organs.

- The organ position is often the same in the patient body. For example, the heart is on the left and the spinal canal pass within the vertebrae. There exist some exceptions as patients suffering of *situs ambiguus* or *situs inversus*. In this special cases, the organs are not located where they should be.

- The delineation can be learnt from a set of images already delineated. From this set of images, relevant properties about the position, shape and

intensity of the regions of interest can be learnt to be able to automatically delineate images.

- The organ shows, at least locally, some homogeneity in gray-level or texture. Conversely, discontinuities are associated with the boundaries between different organs.

- Some regions of interest are easier to identify than others. By identifying firstly some easy-to-find regions of interest, other regions of interest become easier to determine. By working incrementally, from the simplest to the hardest, the delineation task becomes easier and more accurate.

Based on those assumptions, we suggest a method that learns from a set of images delineated by experts. From the acquired knowledge, the method is capable of iteratively delineating the regions of interest from the simplest to the most complicated. This method can be used with a restricted number of parameters. The offered method is generic and can theoretically be adapted to any disease localisation and for any kind of patient morphology with little effort.

1.4 Organisation of the document

The document is organised in two main chapters. Chapter 2 describes the state of the art in image segmentation. In this chapter, the first section (Section 2.1) defines the terms used in the document. The second section (Section 2.2) investigates the methods commonly used in image segmentation outside the field of radiotherapy. The last section is dedicated to the method currently employed in radiotherapy (Section 2.3). Chapter 3 describes the developed method as well as the results obtained with that method. The details about the experiments can be found in the papers in appendices.

1.5 Contributions

This section provides a short summary of our contributions. Those contributions can be found in the appendices at the end of the document.

Appendix A: Incremental feature computation and classification for image segmentation

The first publication describes the principle and a prototype of the method. In this conference paper, the early method is applied to a small toy example. Starting from the fact that image segmentation problems can theoretically be solved with classification algorithms, it is observed that their use is often limited to features derived from intensities of pixels or patches. Features such as contiguity of two regions cannot be considered without prior knowledge of one of the two class labels. Instead of stacking various classification algorithms, the document describe a first incremental classifier that works in a space where features are progressively evaluated. Experiments on artificial images demonstrate the capabilities of the incremental scheme.

Bernard, Guillaume, Michel Verleysen and John A Lee (2012). 'Incremental feature computation and classification for image segmentation'. In: *20th International Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2012)*, pp. 157–162.

Appendix B: Segmentation with Incremental Classifiers

This second paper proposes an approach to extract and encode the physician's expertise. The method relies on a specific classification method that incrementally extracts information from groups of pixels in the images. The incremental nature of the process allows us to extract features that depend on partial classification results but also convey richer information. In the paper, two methods to guide the incremental classification are proposed. The method is illustrated with experiments on artificial images with similar properties to medical images.

Bernard, Guillaume, Michel Verleysen and John A Lee (2013). 'Segmentation with Incremental Classifiers'. In: *Image Analysis and Processing–ICIAP 2013*. Springer, pp. 81–90.

Appendix C: Organ delineation with watersheds and machine learning

This abstract proposes an alternative to atlases, which is aimed at segmenting and recognizing objects and organs in medical images, using watershed transforms and machine learning techniques. With this method, potential segmentation errors are easier to correct. The method is tested on a synthetic

dataset and briefly compared to the results obtained by a segmentation with an atlas.

Bernard, Guillaume, Michel Verleysen and John A Lee (2014). 'Organ delineation with watersheds and machine learning'. In: *29th Belgian Hospital Physicists Association Annual Meeting (BHPA 2014)*.

Appendix D: Automatic organ at risk delineation with machine learning techniques

This abstract and its supplementary documents presents the first results obtained with 2D images of real patients. We know that the manual delineation of organs at risk on CT images consumes much time. Automatic segmentation methods like atlases partly address this issue. However, atlases depend on deformable registration quality. This work proposes an atlas-like method that relies on machine learning techniques instead of registration. The first results show good performances on real dataset.

Bernard, Guillaume, Michel Verleysen and John A Lee (2014). 'Automatic Organ at Risk Delineation with Machine Learning Techniques'. In: vol. 41. 6. American Association of Physicists in Medicine, pp. 101–101.

Appendix E: Incremental classification of objects in scenes: application to the delineation of images

This article is accepted in Neurocomputing. In machine learning, we know that usual multiclass classification techniques often rely on the availability of all relevant features. In practice, however, this requirement restricts the type of features that can be considered. Features whose value depends on some partial, intermediate classification results, can convey precious information but their nature hinders their use. A typical example is the identification of objects in a scene, where the distance from some yet unclassified object to some other that would already be identified earlier in the process. This paper proposes a generic method that solves classification problems involving such features in an incremental way. It proceeds by decomposing the multi-class problem into a sequence of simpler binary problems. Once a binary classifier gives an object its class tag, all features depending on this object are computed and appended to the list of known features. Experiments with both synthetic and real data, comprised of tomographic images, show that the proposed method is effective.

Bernard, Guillaume, Michel Verleysen and John A Lee (2014). 'Incremental classification of objects in scenes: application to the delineation of images'. In: *Neurocomputing*. Forthcoming.

Chapter 2

Image segmentation, object detection and delineation

This chapter provides an overview of the state of the art in the fields of image segmentation, object detection and object delineation. All those methods are applied to images in various fields inside and outside radiotherapy.

A first section defines the general concepts required to explain the different methods. A second section details generic methods used in image segmentation and object detection. In the last section, automatic methods currently used in radiotherapy are described.

2.1 Definitions

Images from any sources depict objects of interest. These objects are characterised by their position, shape, texture, or other features like their spatial arrangement. Before going through the different segmentation methods, some important concepts are briefly defined hereafter.

Image

An image with n dimensions and m channels is a function in $\mathbb{R}^n$ that associates a value $\mathcal{F}(x)$ with each point x in subspace $G \subset \mathbb{R}^n$, $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}^m : x \to \mathcal{F}(x)$. G is a sampling grid of coordinates in $\mathbb{R}^n$. The coordinates of G define the pixel of the image. $\mathcal{F}(x)$ corresponds to the vector of intensity of the pixel

x in the image. Depending on the acquisition system, the dimension of the intensity can be $m = 1$, for medical images with gray values, or $m = 3$, for images acquired by a camera with red, green, blue channels. Moreover, the dimension of the image can be $n = 2$ (e.g. photography and X-ray image), $n = 3$ (e.g. 3D computed tomography (CT), magnetic resonance imaging (MRI) and photography with time component called video) or $n = 4$ (e.g. functional MRI and 3D CT with time component called respiration correlated CT).

Object and scene

Objects, in a very generic sense, are the main elements depicted in an image. Every pixel of an image is supposed to belong to an object. Objects can be a bike, a ball, the sky or, in medical images, a vertebra, a lung or the air. The same objects can sometimes be found in several images, in a similar arrangement. Typical examples are frames in a video, pairs of stereoscopic images, and photographs taken in burst mode. A scene is a picture of the layout of objects living in a p-dimensional space. The image is a representation of the scene, either as a projection (like a 2D picture taken by a digital still camera) or a full 3D image (like 3D CT in medical imaging).

Boundary, contour line and contour mesh

In order to understand the scene, the objects have to be recognised. The best way to identify them is to find their location and boundaries.

A boundary is the set of points that are the closest to an object. In a 2D image, boundaries are usually approximated by contour lines. A contour line is a set of vertices linked by edges forming a closed path (starting and finishing at the same point). By setting the vertices along the boundary, the boundary is approximated by straight line segments. In a 3D image, boundaries can be approximated by defining contour lines on 2D slices extracted from 3D image. By interpolating between the slices, the 3D representation of the object can be approximated. Another solution is to use a mesh to approximate the boundaries. The contour mesh is a generalisation of the contour line to 3D. A 3D mesh is a set of faces, typically triangular ones, defined by linking vertices with edges. A mesh is closed when every edge belongs to two faces. By setting the vertices and edges along the boundary, the boundary of the object is approximated by straight faces.

The establishment of contours is independent of the sampling of the image.

The vertices are not required to be laid on the sampling grid defined by the pixels.

Mask

An alternative to boundaries is the mask. A mask of dimension n is a function in $\mathbb{R}^n$ that associates a value mask(x) in $M \subset \mathbb{R}$ for each point x in subspace $G \subset \mathbb{R}^n$, $m : \mathbb{R}^n \to \mathbb{R} : x \to \text{mask}(x)$. A mask is usually associated with an image and is defined in the same subspace G. The subset M defines the kind of mask:

- binary, if $M = \{0, 1\}$;
- fuzzy, if $M = [0, 1]$;
- multiple, if $M = \{0, 1, \ldots, k-1\}$.

A binary mask allows defining one objects in the image (e.g. 1 represent the object and 0 its complement or background). A fuzzy mask gives the probability that a pixel belongs to an object. A multiple mask defines C different objects in the image (or $C-1$ objects and their background). The mask is directly related to the image and allows defining objects at the same resolution as the image. The mask identifies all the pixels belonging to an object while the contour lines and meshes only define the boundaries.

Prior knowledge and features

Prior knowledge can be used to characterise the scene in various ways. Firstly, it can be related to the application field. This may be general information about the scenes or the objects such as the technology used to realise the image of the scene. It can also be known information about the objects. For example, when taking a picture with a camera, the sky is rarely green and often in the upper part of the image. Another example, when segmenting scanned text document, the object “background" is white, while the object “foreground" is darker. The second use of prior knowledge is to establish a link between images and known information. In this case, specific prior knowledge is associated with each known image. It can be, for example, the localisation (with contours, meshes or masks) of the objects in each known images. The localisation is often used to extract richer information, called a feature (or descriptor). Quantitative features can be measurements of colour, shape, texture, etc. They can be computed for one pixel or a group of pixels from an object.

Model

A model is a system or representation that tries to imitate some phenomenon (e.g. to predict its outcome, like in meteorology). A meta-model is a formalism allowing describing a model. Meta-parameters are parameters related to a meta-model. By fixing the values of the meta-parameters of a meta-model, a specific model is instantiated. Once they are instantiated, static models are fixed and cannot be automatically tuned. With the static models, the meta-parameters and/or the meta-model integrate general prior knowledge specific to the application field. On the contrary, learning models are able to learn some intrinsic parameters based on prior knowledge. By learning the parameters, they become more specific to their prior knowledge and become hopefully better in solving their problems. The learning models usually used prior knowledge from observable data to find its best parameters.

Training set

Learning models often use training sets to tune their parameters. A training set is composed of an ensemble of pairs of input data (e.g. images) and desired output results (e.g. position of the objects in the image). Depending on the field, the training set can be built by experts or non-expert people (via crowdsourcing projects). According to the people who built the training set, its quality may vary greatly.

Image segmentation, object detection and recognition

Segmentation is an image processing operation that aims at grouping pixels according to predefined criteria. The most simple criterion is a threshold on the intensity in a gray-level image. In this case, each segment of the image corresponds to a group of pixels sharing similar properties. Depending on the method, the segment can be connected or disconnected. A connected segment is a segment composed of one piece. As the segment are defined as being a group of pixels, they can be easily represented by masks.

As its name suggests, object detection aims at determining whether some given objects are present in an image. The object detection method requires the use of learning models and training sets. Indeed, the prior knowledge about complex objects is impossible to encode manually. Beyond mere detection, a more challenging task is to give an approximate position of the objects. The

positions can be given with coordinates or in the form of encompassing boxes, masks, contours or meshes.

If the exact boundaries of the object have to be found, the problem is called object delineation. The result is usually contours or meshes. They have to depict as much as possible the boundaries of the searched object.

2.2 Generic segmentation methods

Image segmentation and object detection have been used in several domains. The methods were adapted and optimised depending on their application field. This section provides an overview of the generic methods used for image segmentation or object detection. In the first subsection, methods using static models are presented. These models are opposed to learning models, as their use of prior knowledge is limited to general information regarding the application field. The second subsection presents methods involving learning models. Each of those methods is personalised thanks to the training set to learn the parameters that give the best results.

2.2.1 Static models

Only prior knowledge from the application field can be used to build or select the model.

Thresholding

Thresholding is the simplest image segmentation method. The underlying assumption is that the input image is composed of an object and its background, which can be separated by using only the intensity of the pixels or some other scalar feature. A threshold value, which is a value corresponding to pixel intensity, can be tuned by the user. All pixels above the threshold belong to the object. All other pixels are part of the background (see example in Figure 2.1). The value of the threshold can be set globally, computed from the whole intensities in the image (Otsu 1975), or set locally, computed for each pixel based on its neighborhood (Pappas 1992; Sauvola and Pietikäinen 2000). The number of objects can be increased by using multiple threshold values.

Local thresholding, also called adaptive thresholding, is useful in text recognition and image filtering (Pappas 1992; Sauvola and Pietikäinen 2000). In medical

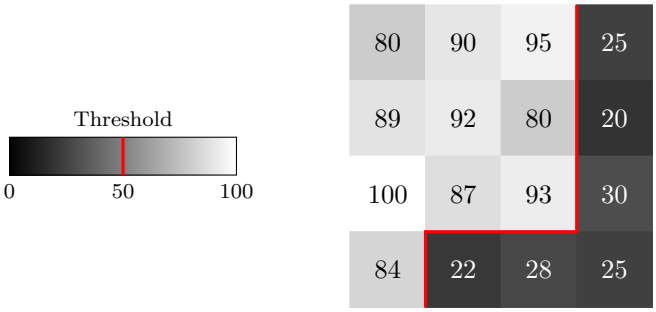


Figure 2.1 – Thresholding on a 4 by 4 pixels image. Left, the scale of gray-level intensities. The threshold value is depicted by the red line. Right, the segmented image with the segmentation in red.

imaging, thresholding can be used to segment positron emission tomography (PET) images in order to delineate target volumes (TVs) (Lee 2010).

This method works well when the foreground and background are segmented, both having different pixel intensity properties. Thresholding does not imply connectedness of the detected areas. The foreground and the background can thus be separated in several groups of pixels disconnected from each other. This issue can be addressed for instance with region growing techniques (Adams and Bischof 1994; Zhu and Yuille 1996).

Graph cuts

The segmentation with graph cut uses graph theory to optimise the segmentation of an image. A graph $\mathcal{G}$ is defined by a finite ensemble $\mathcal{V}$ of vertices and a finite ensemble $\mathcal{E}$ of edges, which are pairs of vertices. If the edge $e = \{u, v\}$ is in $\mathcal{E}$, vertex u and vertex v are adjacent in $\mathcal{G}$. A graph $\mathcal{G}$ is connected if for all pair $\{u, v\} \in \mathcal{V} \times \mathcal{V}$, there exists a path of edges connecting u to v. With graph cuts for image segmentation, the edges are weighted by integers : $\forall e \in \mathcal{E} : W(e) \in \mathbb{N}$.

In the case of image segmentation, the image is converted into a graph where the vertices are the pixels and the edges connect adjacent pixels in the image (Figure 2.2). The image can be represented in different ways by changing the weights of the edges. They can measure the level of similarity or dissimilarity between adjacent pixels. Depending of the algorithm used to segment, some representations can be preferred than others.

In graph theory, a cut is a partition of the vertices of $\mathcal{G}$ in two subsets. Each

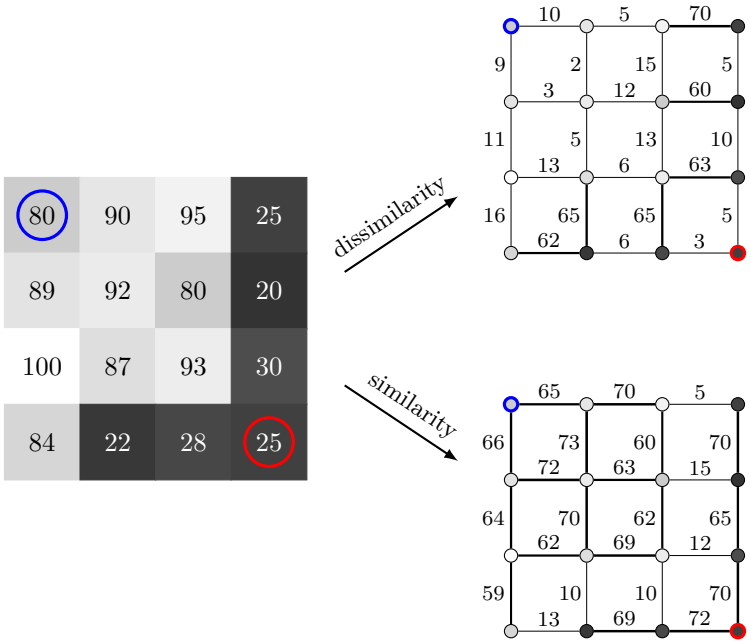


Figure 2.2 – Conversion of an image into a graph of vertices and edges where the weights of the edges measure the dissimilarities (top right) or similarities (bottom right) between the pixel intensities. Blue and red pixels define the seeds for the image segmentation (see text for details).

subset and their related edges define a connected graph. The cut determines a cut-set, the set of edges that goes from one subgraph to the other. In image segmentation, the cut-set define the position of the boundaries.

In order to use the graph cut methods, user interaction is often required. The user has to set one seed per object in the image. Those seeds are used as starting points by the segmentation algorithms. In this section, two popular graph cut methods are presented, the min-cut and optimal spanning forest.

Min-cut. In graph theory, the minimum cut problem consists in finding a cut between two given vertices (seeds), so that the sum of the weights of the edges defining the cut is minimal. The solution to the problem gives a cut of $\mathcal{G}$ where the both connected components contain one seed. There exists specific solvers to find the best solution in a polynomial time (Boykov and Kolmogorov

2004). To apply the solver on the graph representation of the image, the weight of the edges have to measure the similarity between pixels. Each component of the partition corresponds to an object. In order to initialise the solver, two seeds defining the objects need to be provided. A minimalist example of image segmentation with min-cut is shown in Figure 2.3.

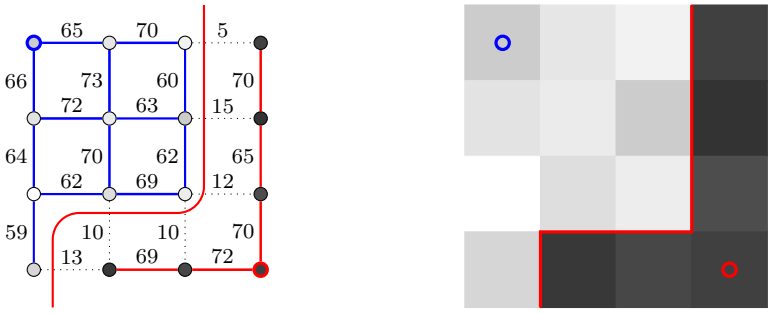


Figure 2.3 – Min-cut segmentation on a 4 by 4 pixels image. The two seeds are the red and blue vertices. Left, the graph representation of the image. The weights of the edges measure the similarities between pixels. Dotted edges represent the set of edges composing the min-cut. The red curved line highlights the cut. Right, the segmented image with the segmentation in red.

To avoid degenerate cuts with only very few pixels or almost all of them, it is possible to use the normalised cuts where the minimisation criterion is the average weight of the edges in the cut instead of the sum (Shi and Malik 2000).

The problem can be extended to more than two seeds but in that case, the problem becomes non-deterministic polynomial-time hard (NP-hard) (Boykov and Kolmogorov 2004). NP-Hard problems have a non-polynomial complexity and are difficult to solve when the number of pixel increases.

Optimal spanning forest. In graph theory, a cycle is a path of edges starting and ending in the same vertex. A spanning tree of $\mathcal{G}$ is a connected subgraph of $\mathcal{G}$ without any cycle, in which all the vertices of $\mathcal{G}$ are included. A spanning forest is an ensemble of disjoint spanning tree covering a partition of a graph. With the min-cut segmentation, the maximum/minimum spanning forest (MSF) is a well-known solution to build a segmentation from several seeds. In the optimal spanning forest problem, each seed defines the starting point of a spanning tree. Each of the spanning trees contains strictly one seed. The maximum (respectively minimum) spanning forest problem consists in finding

the spanning forest for which the sum of the edges in the forest is maximum (resp. minimum). Depending on the representation of the image, the maximum or the minimal spanning forest have to be computed. When using a weighted graph representation with similarities (resp. dissimilarity), the maximum (resp. minimum) spanning forest needs to be computed. The advantage over the min-cut segmentation is that the number of seeds can be larger than two without increasing the computational complexity.

The MSF solver finds the best spanning forest for a given graph so that each spanning tree contains one seed and the sum of the edges in the spanning forest is maximum/minimum. An example of segmentation with a MSF solver is provided in Figure 2.4.

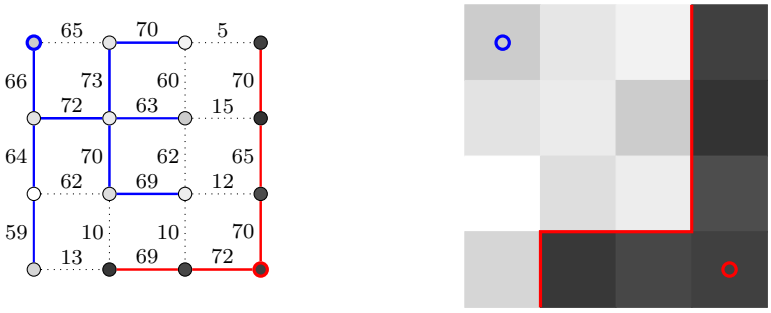


Figure 2.4 – Maximum spanning tree segmentation on 4 by 4 pixels image. The two seeds are the red and blue vertices. Left, the graph representation of the image. The weights of the edges measure the similarities between pixels. The plain edges belong to a spanning tree, dotted edges do not belong to any spanning forest.

The two presented methods (min-cut and optimal spanning forest) request user interaction in order to set the seeds in the image. More advanced methods can be used to improve the segmentation based on seeds such as the graph cut on n-dimensional images proposed by Boykov and Funka-Lea (2006).

Graph-based segmentation methods that do not require any seed also exist (Felzenszwalb and Huttenlocher 2004). In those methods, the number of regions is optimised during the computation. At the beginning of the process, the image is over-segmented (each vertex represented a region), then the elements of the segmentation are progressively merged to minimise an energy function. The energy function defines criteria to control element merging.

Methods with graphs are sensitive to noise. Indeed, only the similarity/dis-

similarity between neighbour pixels is taken into account. If the noise in the pixel intensities is important, the measure can be distorted.

Watershed segmentation

Watershed segmentation was first proposed by Beucher and Lantuéjoul (1979). The idea is to use the magnitude of the gradient of the image as a topological relief. A flooding is simulated starting from all local minima. If two catchment basins become connected, a dam is built and the flooding continues. The process stops when all pixels are either under water or belong to a dam. The dams between the catchment basins define the segmentation of the image (Beucher and Meyer 1992; Meyer 1994). Figure 2.5 shows the principle of the watershed on a one dimentional image. The watershed segmentation can easily be generalised to images of dimension n.

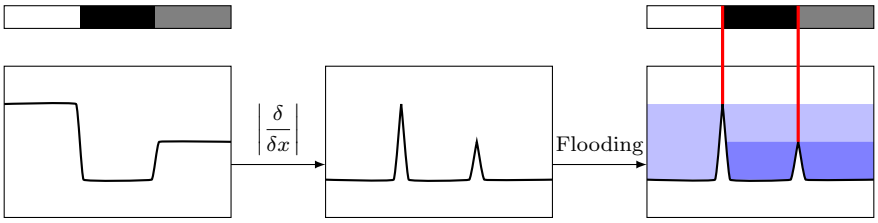


Figure 2.5 – Watershed on 1D image. Top, the images before (left) and after (right) the watershed segmentation. Bottom left, profile of intensity level of the pixel along the image. Bottom center, profile of the magnitude of the gradient of the intensity. Bottom right, profile of the flooding of the gradient and growth of the dam.

The watersheds can also be determined by simulating water dropping on the gradient magnitude image. The water follows the steepest descent until the bottom of the catchment basin. All points crossed during the descent belong to the same basin. By repeating the process with each point of the gradient, all pixels get assigned to catchment basin. The watershed lines can be drawn between the catchment basins and follow the dams built in the flooding simulation. The watershed can also be applied to the graph representation of the image. In this case, the gradient is no longer used (Cousty et al. 2009). The watershed method with graph representation is very powerful, it allows having a algorithm with linear complexity to solve the problem. This means that the computational time linearly increases with the number of pixels. The method

is therefore usable on large images.

The watershed method is very sensitive to noise. Indeed, each local minimum of the gradient magnitude is already a tiny catchment basin. When working with the gradient, a merging post-processing method can be used to reduce the number of the catchment basins (Bleau and Leon 2000). In order to set in advance the number of catchment basins in the image, Cousty et al. (2009) propose to filter the graph representation of the image. The filtering method (Najman and Couprie 2006) allows filtering the graph so that the catchment basins are merge based on their area or volume. For a catchment basin, the area is the number of pixels, while the volume takes into account both the area and the depth of the basin in each pixel.

As for the graph cut methods, seeds can be used in the watershed method. Seeds introduce local minima, while the other local minima are filled and flattened in order to disappear. By doing this, each catchment basin has one and only one seed. As demonstrated by Allène et al. (2007), there exist strong relations between min-cut, optimal spanning forest and watershed. They showed that under some conditions (e.g. seeds, meta-parameters, representations), all these methods give the same results.

2.2.2 Learning models

In the previous subsection, methods using limited amount of prior knowledge were presented. This subsection focuses on methods using learning models for image segmentation, object recognition and detection. Those methods require a training set in order to adjust the parameters of the model. Before presenting the methods dedicated to image processing, general classification methods are described.

Classifier

Let $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_N, y_N)\}$ denote a training set for classification. It is composed of N observed data, where $\boldsymbol{x}_i \in X$ is the ith instance represented by a feature vector of dimension M, and $y_i \in Y$ is its label. A classifier is a function $\hat{f}$ that estimates the function $f : X \to Y$, where X is the feature space in which each feature is a dimension and Y is the output space. Each instance $\boldsymbol{x}_i$ can be seen as a point in the feature space X. For example, if all features are defined in $\mathbb{R}$, the feature space is $\mathbb{R}^M$ and $\boldsymbol{x}_i$ can be seen as a point in $\mathbb{R}^M$. If the number of label is equal to two ($|Y| = 2$), the classification problem is

binary. If there exist more than two labels, it is called a multiclass classification problem. All the instances from the feature space having the same label define a class.

Classifiers should be used carefully as they are subject to overfitting. Overfitting occurs when function $\hat{f}$ is so specific to the training set that $\hat{f}$ is not capable to correctly classify unseen instances. In other words, function $\hat{f}$ is not able to correctly estimate the function f. Real data is often polluted by some noise. When a classifier overfits, it somehow learns both the useful signal and noise that affects it. Overfitting should be avoided as the classifier needs to be able to correctly classified any data from the feature space.

Classifiers are generic, they can thus be trained on nearly any kind of data: pixels, weather measurements, financial data, genomic data, etc. Some classifiers are briefly presented in this document.

***k* nearest neighbours.** The k nearest neighbours (kNN) classifier relies on the hypothesis that instances with the same labels are similar in the feature space X (Cover and Hart 1967). The similarity or dissimilarity of a query instance $\boldsymbol{x}_\mathrm{q}$ with all instances in the training set are computed in order to determine its label. The labels of the k most similar instances (k nearest neighbours) are used in a majority vote process where the most frequent label is given to x_q.

A few meta-parameters can be tuned to modify the classifier behaviour:

- The number k of neighbours can be changed to modify the size of considered neighbourhood. If k is too small, the method is very sensitive to noise. If k is too big, the risk to take into account distant and thus possibly unrelated neighbours exists. This can lead to wrong classification.
- The similarity measurement between instances can be adapted. In addition to the traditional Euclidean distance, we can cite the Manhattan and Chebychev distances, and Kendall's rank correlation. Some metrics are better suited to some feature representations or problems.

The kNN algorithm is very simple but classifying a new instance requires to measure its similarity to all instances in the training set. If the training set is big, the computation time can become important for each classification query. Algorithms allowing filtering of the instances by grouping them address this issue (Wilson and Martinez 2000; Kubat and Cooperson 2001; Brighton and Mellish 2002).

Decision tree. A tree is a graph where there exists only one path (sequence of edges) to travel from one vertex to another. A decision tree $\mathcal{T}$ is a tree defined by a finite ensemble of vertices $\mathcal{V}$ and edges $\mathcal{E}$ where the edges are oriented. If the edge $e = (u, v)$ is defined in $\mathcal{E}$, there is an edge from vertex u to vertex v in $\mathcal{T}$, v is a child of u and u is the parent of v. The vertices having children are called nodes, while the vertices without child are called leaves. In a decision tree, there always exists a root node r that does not have any parent. Figure 2.6 represents an example of decision tree built from real data.

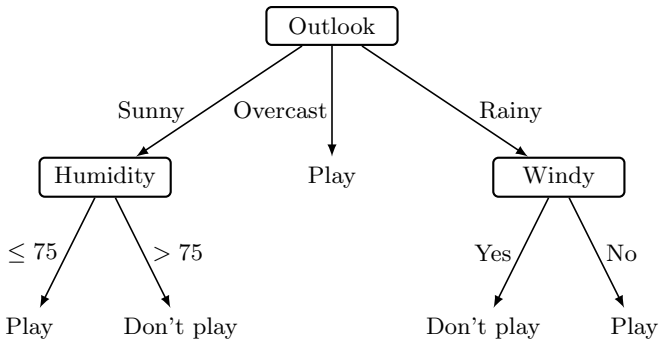


Figure 2.6 – Decision tree describing if a golf game should be played according to the weather conditions (Quinlan 1993). The rectangles are the nodes where one feature is tested. The contents of the node represent the name of the tested features. The edges represent the different solutions to the test. The leaves are the terminal vertices They give the label, in this example, if the game should be played or not.

In a decision tree, each node tests one entry of the feature vector associated with instance $\boldsymbol{x}_{\mathrm{q}}$. The test can be binary with two possible outcomes (e.g. one threshold on the value of a continuous feature), or multiple with more than two outcomes (e.g. several thresholds on the value of a continuous feature or several possible values for a non continuous feature). Each possible outcome results in an edge leading to a vertex. The vertex can be a child node or a leaf. For each leaf, a label is associated (see example in Figure 2.6). When classifying a query instance $\boldsymbol{x}_{\mathrm{q}}$, the features are tested from node to node starting from the root until reaching a leaf. The label associated with the leaf is given to the query instance.

The decision tree is learnt from the training set. All the instances are used to evaluate the feature tested at the level of r. The feature that discriminates

the most different labels is chosen to be tested in the node. The instances are separated into different groups according to the value of their tested feature. If a group contains only instances with the same label, a leaf with this label is created as a child. If there exist different labels in a group, a subtree is grown as a child.

Several meta-parameters can be tuned to modify the classifier behaviour:

- The metric used to evaluate the most discriminant feature can be changed. The most used metrics are the Gini index (Gini 1921) and the information gain (Kullback and Leibler 1951).
- The minimum number of instances in a group to grow a subtree can be changed in order to avoid overfitting. Indeed, by growing the subtree with only a few instances, the classifier may become too specific to the training data. When a subtree is stopped, a leaf is created and the most frequent label present in the group of instance is associated with the leaf.
- As for the number of instances in a group, the maximum depth of the tree can also be limited to avoid overfitting.
- The tree can be pruned after being completely grown. Pruning consists in converting the unproductive subtree into leaf. The unproductive subtrees are usually evaluated with an additional dataset (instances and corresponding labels), called validation set, that is not used for the training of the decision tree.

Unlike kNN, classifying a new instance with a decision tree is fast. Most of the computation is done when learning the decision tree. Decision trees are subject to overfitting and should be used carefully. Due to their construction, decision trees can be easily visualised. Nowadays, there are often replaced with ensembles of decision trees such as random forests.

Well-known algorithms to build decision trees from training set are ID3 (Quinlan 1986), CART (Breiman et al. 1984) and C4.5 (Quinlan 1993).

Random Forest. A random forest (RF) is an ensemble of randomised decision trees grown from a random subsample of the training set (Ho 1998; Breiman 2001). A randomised decision tree is a decision tree where a random subset of features is evaluated to find the one that discriminates the most different labels in each node during the growth of the subtree. The subset is chosen randomly for each node. For each randomised decision tree composing the RF, N instances are randomly selected from the training set. Each instance can be

present more than once. The use of two randomisations allows reducing the risk of overfitting. To determine the label of a new instance $\boldsymbol{x}_{\mathrm{q}}$, the instance is classified in each decision tree and the most frequent label is given.

Several meta-parameters can be tuned to modify the classifier behaviour:

- The number of trees in the forest can be changed. Adding trees improve the accuracy of the method but increase its computation time.
- The size of the subset of features evaluated at each node can be modified.
- All meta-parameters that could be changed for decision trees can also be tuned in the RF.

Unlike the decision trees, the RF can difficultly be displayed or visualised in practice. The classification of a new instance with a random forest remains easy. As all the decision trees composing the forest are independent, the classification by a RF can be parallelised, each tree is evaluate at the same time, to improve the speed of classification.

Linear classifier. Linear classifiers aim at solving binary classification problems ($Y = \{-1, 1\}$) with real feature vectors ($X = \{\mathbb{R}^M\}$) where the classes are linearly separable. Two classes are linearly separable if there exists a M-1-dimentional hyperplane that makes the boundary between the two classes in the feature space. In the two-dimensional case illustrated in Figure 2.7, the hyperplane is the solid line separating the blue and green dots.

A linear classifier aims at finding a hyperplane separating the two classes. If the two classes are linearly separable, it is indeed possible to find such a hyperplane. An instance of one side of the hyperplane belongs to one class, if not, it belongs to the other class. Once the hyperplane is known, the label of a query instance $\boldsymbol{x}_{\mathrm{q}}$ can be determined by the decision rule

$$y_{\mathrm{q}} = \operatorname{sign}\left(\boldsymbol{w} \cdot \boldsymbol{x}_{\mathrm{q}} + b\right) , \tag{2.1}$$

where $\boldsymbol{w}$ is the normal vector to the hyperplane, b is the bias term and sign is a function that returns -1 (respectively 1) if the number is negative (resp. positive).

Most of the time, there exists an infinity of hyperplanes between two linearly separable classes. This is why it is preferable to use support vector machines (SVM) which try to find the best hyperplane among all admissible ones.

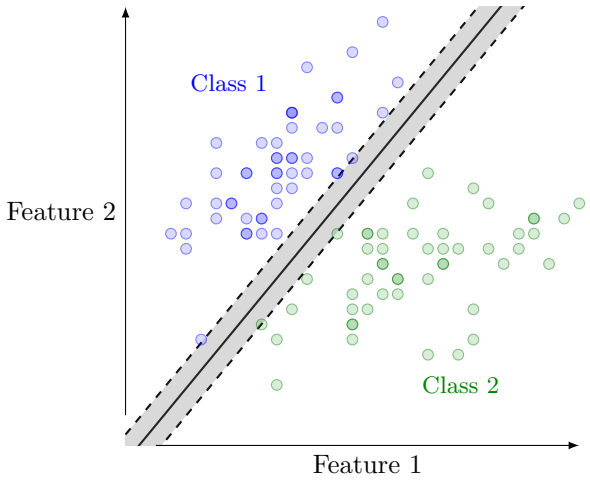


Figure 2.7 – Linear separation of two classes by a straight line. The separating line is the plain line. The margins that are delimited by dashed lines show the area between the two classes without instance. Support vector machine aims at maximising the margins.

Support vector machines. The SVM can be seen as a generalisation of the linear classifier. SVM aim at finding the hyperplane that maximise the margin between the hyperplane and the instances (see Figure 2.7) (Cortes and Vapnik 1995). Finding the hyperplane that maximises the margins can be solved with a quadratic programming method. The details about the resolution of the optimisation problem are beyond the scope of this document.

The SVM also introduce the concept of soft margin. It allows having a limited class mixture on each side of the hyperplane when the instances are not linearly separable (Cortes and Vapnik 1995). Margin softness can be adjusted by changing a meta-parameter before the optimisation.

The SVM have the ability to project the instances in another space, called feature space[1], before learning the best hyperplane. By projecting the instances in the feature space, the method can find complex, nonlinear class boundaries such as curved surfaces. The transformation is defined by a kernel function κ redefining the dot product between instances, namely, $\phi(\boldsymbol{x}_\mathrm{i}) \cdot \phi(\boldsymbol{x}_\mathrm{j}) = \kappa(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_\mathrm{j})$ where ϕ is the implicit mapping from the input space to feature space (Boser et al. 1992). By using a kernel, the SVM are able to identify admissible boundaries

[1] In a technical meaning, different from the one used elsewhere in the document.

of linearly and nonlinearly separable classes. Well-known kernel functions are:

- Polynomial: $\kappa(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_\mathrm{j}) = (a\boldsymbol{x}_\mathrm{i} \cdot \boldsymbol{x}_\mathrm{j} + b)^d$,
  where a, b, and d are meta-parameters.
- Gaussian: $\kappa(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_\mathrm{j}) = e^{-\gamma * |\boldsymbol{x}_\mathrm{i} - \boldsymbol{x}_\mathrm{j}|^2}$,
  where γ is a meta-parameter.
- Sigmoidal: $\kappa(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_\mathrm{j}) = \tanh(a\boldsymbol{x}_\mathrm{i} \cdot \boldsymbol{x}_\mathrm{j} + b)$,
  where a and b are meta-parameters.

All those kernel functions require the user to define some meta-parameters before solving the associated optimisation problem.

Linear classifiers and SVM can be adapted to deal with multiclass problem, where $|Y| = C$ and $C > 2$, by building several classifiers. This can be done by learning one-versus-one classifiers. In this case, $\frac{C(C-1)}{2}$ classifiers are learnt to identify one class against another. Once a new instance $\boldsymbol{x}_\mathrm{q}$ is classified, it receives the label of the most frequently given class. Another solution is to learn C one-versus-all classifiers. Each binary classifier is learnt to identify one given class (identify with label $y = 1$) against all others (identified with label $y = -1$). For a new instance $\boldsymbol{x}_\mathrm{q}$, the classifier with the strongest confidence, that is, yielding the highest value of $\boldsymbol{w} \cdot \boldsymbol{x}_\mathrm{q} + b$ (see equation 2.1), gives the label.

The SVM are powerful but sometimes difficult to parametrise. Indeed, the selection of the best margin softness and optimal kernel meta-parameters remains complicated. Meta-parameters are often defined by using a cross-validation method, where a validation set is used to determine the best combination of meta-parameters.

Object detection

Methods relying on classifiers have been used for object detection. As a reminder, object detection aims at finding the position of an object in a image. It is used in modern digital still cameras to detect and track faces. Most of the time, the object is shown encompassed in a simple box. Fields of application are face and human body detection (Osuna et al. 1997; Mohan et al. 2001; Enzweiler and Gavrila 2009).

A first approach is to use the pixel intensities to detect the position of the object. Rectangles containing the object are extracted from training images. All rectangles are rescaled and the pixel intensities are used as features to train the classifier. When a query image is given, a window is slid on the image. Pixels

in the window are used as features for a query. The classifier processes the pixel values to determine whether an object of interest is present in the rectangle.

In the past decades, more and more complex features were designed to extract richer information from the images. The aim is to be able to extract richer information that should allow for a better classification. Instead of using all pixel intensities, Viola and Jones (2001) propose to combine them in richer features. They suggest the use of block filters to build the features. Among other well-known feature construction methods, we can cite the Haar wavelet (Papageorgiou and Poggio 2000; Lienhart and Maydt 2002), scale-invariant feature transformation (SIFT) (Lowe 1999; Ke and Sukthankar 2004), speeded up robust features (SURF) (Bay et al. 2006) and the histograms of oriented gradients (HOG) (Dalal and Triggs 2005).

Tracking systems often need to work fast. Due to this requirement, a trade-off has been realised between speed and accuracy. Only the position of the object is given by a box. There is no information about the boundaries of the object.

Segmentation and recognition

Methods depending on classifier have been used to label pixels of complex images. One application field is the segmentation of airborne and satellite images. Those images have to be automatically segmented in different zones to study climate changes, deforestation, or urban expansion. Typical areas are forests, urban area, roads, land, and water. Image data are acquired by instruments detecting several electromagnetic spectra. Each pixel of the image is therefore characterised by several intensities, one per acquired spectrum. By using the different spectra as features, the pixels can be classified with classifiers such as decision tree, RF or SVM (Friedl and Brodley 1997; Brown et al. 2000; Mahesh Pal and Mather 2003; Song and Civco 2004; M Pal 2005; Gislason et al. 2006; Mountrakis et al. 2011).

Good classification is sometimes not achievable with pixel intensities only. In those cases, richer features can be computed with Haar wavelet, SIFT, SURF or HOG methods, such as mentioned above.

2.3 Segmentation methods in radiotherapy

In the planning step of a radiotherapy treatment, the target volumes (TVs) and organs at risk (OARs) have to be delineated. The delineation of the

volumes is later used to optimise the radiation dose distribution. Accurate delineation is mandatory to deliver the best treatment for the patient. Even if the delineation task is mainly done manually nowadays, methods exist to perform this task automatically. For those methods, prior knowledge is usually provided by a training set containing images and their associated contours. In this section, several delineation techniques used in radiotherapy are studied. The first method is the atlas, which is the most used one. The model involved in this method makes limited use of prior knowledge and can be considered as static (see Section 2.2.1). The second studied strategy is the statistical model in which the shape and sometimes the appearance of the organs are learnt. The third considered technique is the machine learning technique based on pixel intensity. Finally, a possible combination between those techniques is discussed. For each of those methods, a description is given as well as the pros and cons.

2.3.1 Atlas

Obvious anatomical similarities between individuals exist. For most people, the heart is placed slightly on the left of the chest, the skull encapsulates the brain, etc. These similarities are even more pronounced when the compared individuals are of the same sex. Starting from this observation, a reasonable hypothesis can be made that the body of someone can be deformed somehow into the body of someone else.

If a delineated image, called atlas, can be deformed to maximise its similarity with a query image, the deformed contours can then be applied on this image (Rohlfing, Brandt, Menzel, Russakoff et al. 2005) (see Figure 2.8). The key point of the atlas method is to find the best deformation, through a process called image registration. However, this problem is ill-posed as several deformations often exist to transform an image into another. Some of them, however, are more 'realistic'. The choice of the meta-parameters of the registration method can thus critically influence the result.

Before detailing the different kinds of existing atlases, the next subsection briefly describes image registration.

Image registration

Lets us consider two images: a fixed one (F) and a moving one (M) (see Figure 2.9). Those images can have different modalities and sizes. Let $\mathcal{M}$ be a metric such that $\mathcal{M}(I_1, I_2)$ measures the dissimilarity between images I_1 and

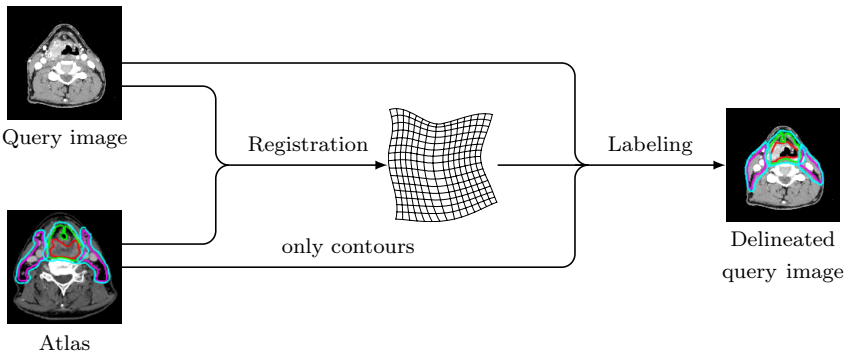


Figure 2.8 – General overview of the atlas method. The atlas is deformed to the query image to obtain the deformation field, which is then applied to the contours. The contours are used with the query image.

I_2. Finally, $\mathcal{T}$ is a transformation computed with a registration method. The deformed image D is obtained by applying transformation $\mathcal{T}$ on the moving image so that $D = \mathcal{T} \circ M$. As $\mathcal{T}$ is continuous and the image is a discrete sampling of a scene, the use of an interpolator is requested. The registration algorithm iteratively updates $\mathcal{T}$ such that $\mathcal{M}(F, \mathcal{T} \circ M)$ becomes minimal. However, the transformation may become unrealistic in a physical or anatomical point of view. This is why a regularisation term $\mathcal{R}(\mathcal{T})$ is added to the optimisation process to maintain the transformation plausible. In the end, the optimiser identifies $\tilde{\mathcal{T}}$ such that

$$\tilde{\mathcal{T}} = \arg\min_{\mathcal{T}} \mathcal{M}(F, \mathcal{T} \circ M) + \mathcal{R}(\mathcal{T})\,. \tag{2.2}$$

In this fully automatic process, four elements can be tuned : the interpolator, the metric, the optimiser and the transformation model.

Linear, trilinear, or cubic interpolation provide different trade-offs between simplicity, computation speed, and smoothness. The result provided by the interpolator is used in the metric in order to evaluate the similarity.

The choice of the metric is typically driven by the modalities of the fixed and moving images. If the images have the same modalities, the intensities of each pixel should be comparable in both images, the sum of squared differences can be used. This is the simplest metric in which the differences of intensities are compared pixel by pixel and summed to measure the dissimilarity between

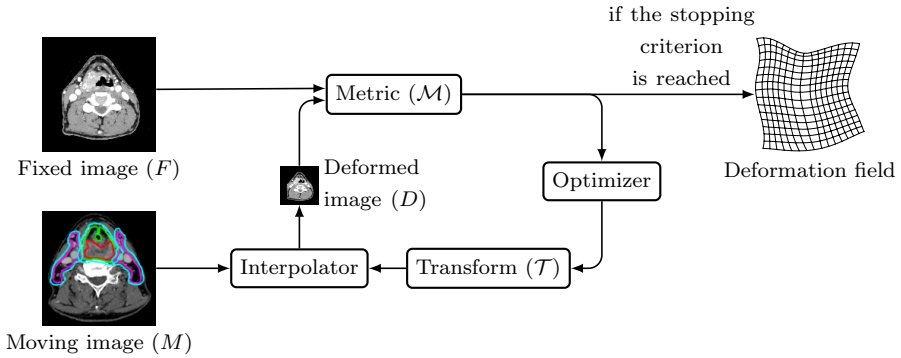


Figure 2.9 – Registration of two images. For more details, refer to the text.

the images. If a linear dependence exists between the intensities of the images, a correlation measure can be used as a metric. Finally, if intensities are mapped nonlinearly between images, then mutual information is the metric of choice. This typically happens when the images are from different modalities (e.g. by registering a CT image with an MRI). Stemming from information theory, mutual information measures the statistical dependence between two images. If images are mutually dependent, there exists a strong relationship between their pixel intensities.

The choice of transformation $\mathcal{T}$ is less straightforward and more related to the geometry of the depicted objects and their mechanical properties in the images. In the case of atlas-based registration, rigid or affine transformations do not suffice. Non-rigid transformations are needed to capture complex body deformation. These are typically implemented with vector fields, often called deformation fields, sharing the same sampling grid as the fixed image. The non-rigid transformation methods can be split in two categories, which are mainly determined by the way the deformation field is parametrised and regularised.

Parametric deformable transformations. This first approach is to allow a local deformation in only a limited number of predefined control points in the images, usually much lower than the number of pixels. Between this control points, the deformation field is interpolated, with a specific model. Among those, the most common ones are the radial basis function (Fornefett et al. 2001), the thin-plate splines (Rohr et al. 1996), the B-spline (Rueckert, Sonoda et al. 1999) and the mesh-based linear elastic finite

element (Ferrant et al. 2000). In the case of meshes, the determination of the position of the control points can be difficult. They are usually placed where there is the more information in the images such as the edges and the high contrast zones. Nevertheless, the description of the transformation in low contrast zone, such as soft tissues, remains a difficult task. Registration consists in optimising the parameters of all the local deformations in the control points. In parametric approaches, prior knowledge is hard-coded in the model, which restricts the set of possible deformations and thereby regularises the registration problem. Hence, the regularisation term in (2.2) can be dropped.

Non-parametric transformations. In this second approach, there is no model associated with the deformation field. Each pixel has its own, unconstrained deformation vector, which can be optimised independently of its neighbours. Nevertheless, this freedom can lead to unrealistic deformation fields. In order to avoid such useless solution, the vector field must be regularised. This is typically achieved by smoothing the deformation field, e.g. with a Gaussian filter. The advantage of the non-parametric approach is the possibility to separate metric minimisation and regularisation. Prior knowledge is here implicitly encoded in the regularisation term. Regularisation typically enforces the smoothness of the deformation field. Those approaches are very sensitive to the regularisation model as well as their meta-parameters. A strong regularisation will lead to small local differences in the field of deformation, which can give poor registration results. Having a small amount regularisation will lead to unrealistic deformation. The most common registration method involving non-parametric transformations is the Demons (Thirion 1998). Another example is the Morphons (Knutsson and Andersson 2005).

A multi-scale approach is often used to increase speed, accuracy, and robustness (Beauchemin and Barron 1995). At the beginning, the images are registered at low resolution, that is, after blurring and downsampling (see Figure 2.10). Next, image resolution is partly restored and a new registration is initiated. This process is repeated until reaching the original resolution. The deformation field computed is used as an initialisation field for the following scale. The complete registration is therefore performed from coarse to fine. This method allows finding bigger displacement that could not be estimated because of the local optimum created by the noise in the full resolution image.

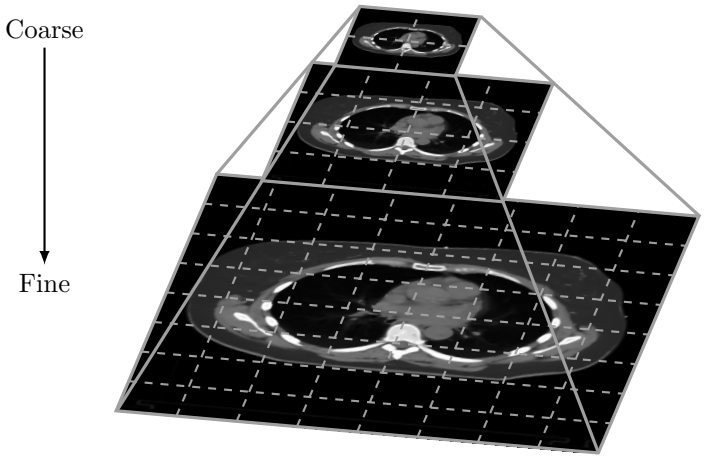


Figure 2.10 – Multiscale registration. The registration is often realised from low to high resolution. The resolution increases until obtain the full resolution.

Determining an appropriate regularisation of non-rigid methods remains a difficult task. The human body can indeed undergo very complex modifications. Some tissues can appear, disappear, or change their composition, which cannot be rendered with spatial deformation.

Registration methods are the central parts of the delineation by atlas. The atlas can be used in different ways in order to give the best delineations.

Different atlas combinations

Due to the limitations of registration, especially in their regularisation, it can be useful to use more than one atlas. Several approaches, the most common ones, are presented hereafter.

Single atlas (Figure 2.11) This case is the most basic and involves only one registration with a single atlas. The main drawback with this method is potential dissimilarity between the atlas and the query image, which may be large. If the image of the atlas differs too much from the query image (because of the gender, weight, etc.), none of the possible deformations can explain the difference between the images. However, this method is useful when the query image and atlas are from the same patient.

Nowadays, the CT scanners are able to acquire 4D images. Those images

are a series of traditional tomographic 3D images that capture motion in time. This allows the organs to be imaged at different phases of breathing. The manual delineation of the organ on each phase can be very long. One example of successful application of the single atlas consists in delineating manually the organs of interest on a single phase, which is then used as an atlas to automatically delineate the same organs in the other phases.

Works using single atlas delineation are Rueckert, Sonoda et al. (1999), Rohlfing, Brandt, Menzel and Maurer Jr (2004) and Han et al. (2008).

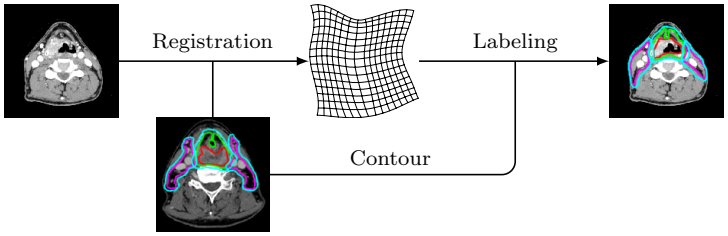


Figure 2.11 – Single atlas. This method uses only one atlas. The atlas is registered to the query image and the contours of the atlas are then used with the query image.

Multiple atlas (Figure 2.12) In this case, several atlases, forming a training set, are used to delineate organs in the query image. In practice, the same approach as in the single atlas is used with each atlas in the training set. Among all provided delineations, it is necessary to use a decision rule in order to get a single result. The easiest decision rule is a majority vote. For each pixel, a vote is given by each delineation. Those votes are then counted, and the most frequent label is assigned to the pixel. More complex iterative methods taking into account the degree of agreements have been developed recently. Those methods improve the quality of the delineation compared to the majority vote rule. The most popular method is "Simultaneous truth and performance level estimation" (STAPLE) (Warfield et al. 2004), which is designed to combine the delineations of a small number of experts. A recent method called "Selective and Iterative Method for Performance Level Estimation" (SIMPLE) (Langerak et al. 2010) has been designed to be able to detect and discard the bad performing segmentation during the process of label-fusion. With multiple atlases, label fusion is mandatory but this can lead to disconnected objects (i.e. in several non contiguous parts). However, these are undesirable as they are often unrealistic.

The separation of the objects in two or more delineated volumes can be caused by missing or supernumerary pixels in the volume. In addition to label fusion, the computation time of the registration increases every time an atlas is added in the training set.

Works using multiple atlas delineation are Dawant et al. (1999), Rohlfing, Brandt, Menzel and Maurer Jr (2004), Han et al. (2008), Klein et al. (2008), Aljabar et al. (2009) and Sabuncu et al. (2010).

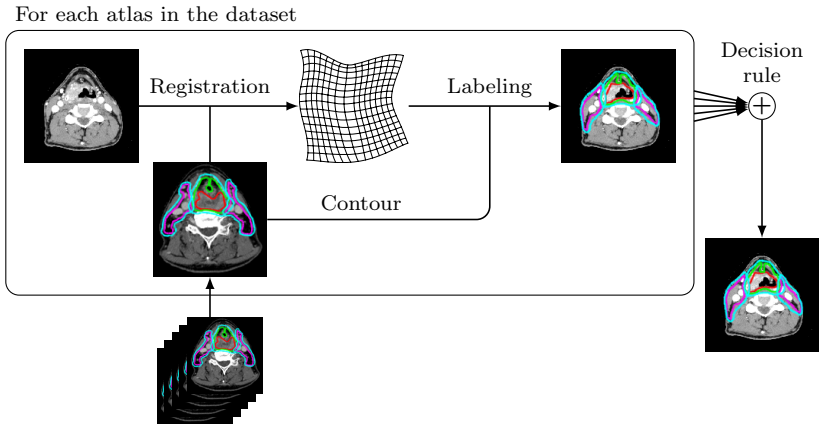


Figure 2.12 – Multiple atlases. This method repeats the single atlas method for each atlas in the training set. A decision rule is applied with all the results to give the final delineation.

Similarity-based atlas (Figure 2.13) Like in the multiple atlas approach, a set of atlases is used to obtain the delineation. The underlying assumption here is that one atlas in the training set will register better than the other ones with the query image. It would be preferable to use it in order to obtain the best delineation. Nevertheless, as the delineation of the query image is unknown, a heuristic has to be used to find that best atlas. Two criteria can evaluate the quality of each atlas. The first one relies on the registration metric, the second one relies on the magnitudes of the deformation. Those criteria can be computed after registration and are minimal for the best atlas. When the choice is based on the magnitudes of the deformation, the average or maximum deformation can be used as a criterion. Compared to the multiple atlas approach, no decision rule is needed but the computation time remains high.

Works using similarity-based atlas delineation are Rohlfing, Brandt, Menzel and Maurer Jr (2004), Commowick and Malandain (2007), Aljabar et al. (2009), Commowick, Warfield et al. (2009) and Jia et al. (2010).

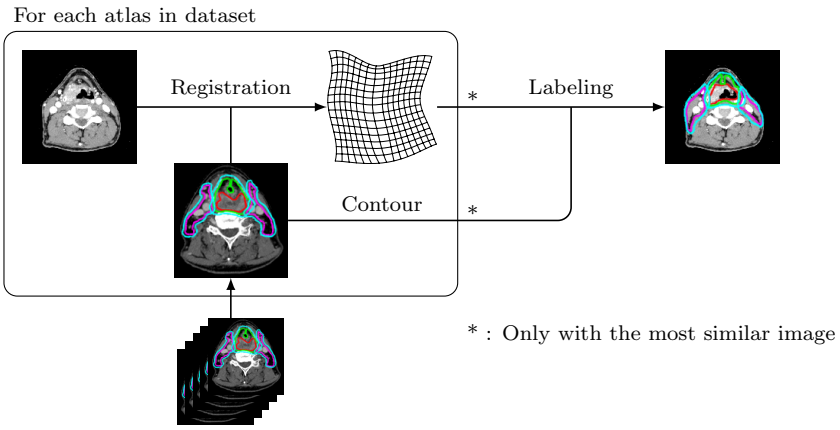


Figure 2.13 – Similarity atlas. This method tries to find the most similar image, after registration, in the training set to performed the delineaton.

Average or mean atlas (Figure 2.14) In this approach, all atlases in the training set are registered to a reference one, leading to an average atlas. The query image is registered to latter. The main advantage is that the average atlas can be computed beforehand. Once it is determined, the use of the average atlas is equivalent as the use of the single atlas method.

The choice of reference atlas can influence the quality of the average one. Indeed, the reference atlas must be representative of the others in the training set and avoid the biggest/thinnest or the tallest/smallest individual. In order to mitigate this problem, Rohlfing, Brandt, Menzel, Russakoff et al. (2005) propose to use an iterative method that avoids the use of a reference atlas. Another solution is to use the most similar atlas to the query image as the reference for building the average atlas. However, in this latter case, the average atlas cannot be computed beforehand. In any case, the merging of all atlases may cause some unrealistic blurring in the average atlas. The border of some anatomical parts of the body can be poorly defined and jeopardise registration quality. When registering the atlases with the reference atlas, all the delineations need to be merged. As for the multiple atlases, a label fusion method, such as STAPLE and

SIMPLE, should be used. Once again, there is a risk of disconnected contour when using those methods.

Works using average atlas delineation are Qazi et al. (2011), Rohlfing, Brandt, Menzel and Maurer Jr (2004), Blezek and Miller (2007) and Commowick, Warfield et al. (2009).

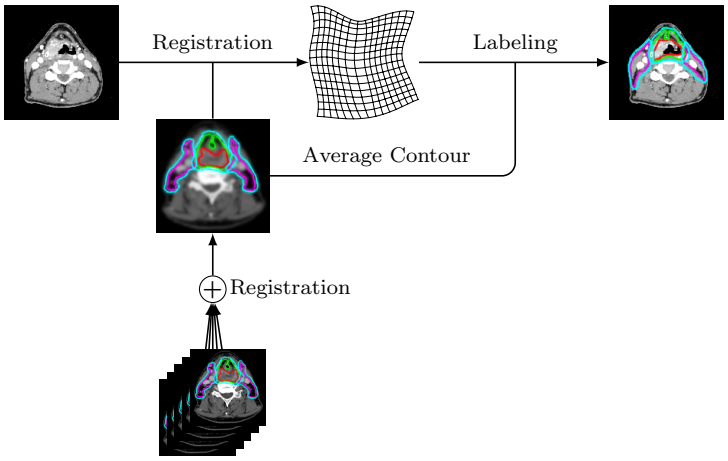


Figure 2.14 – Average/Mean atlas. This method registers all the atlases in the training set. The average atlas is then registered with the query image to give the delineation.

Pros and cons

Almost all atlas methods entail image registration. Image registration has been well studied for years and atlases can rely on this strong expertise to automatically delineate query images. The registration principle is easy to comprehend intuitively. Moreover, atlases are relatively easy to build, since any segmented image is eligible as a possible atlas. The construction of an atlas does not require much additional work by the physician. As a particular case, non-rigid registration methods are very useful when the query image and the atlas are acquired from the same patient. As briefly explained above, the most simple example of atlas consists in registrating a segmented phase of a 4D CT scan with all other phases of the respiratory cycle. In recent years, the deformation of organs in the human body has been further studied and has led to improvements in the regularisation schemes used in non-rigid registration.

Nevertheless, regularisation remains difficult and an open field of research. Too weak a regularisation naturally decrease the value of the registration metric but the apparently better solution can be unrealistic. On the contrary, when regularisation is too strong, deformations can be overly smoothed and no longer able to capture complex situations, like cavity occlusions, sliding organs, or other discontinuities. This issue, already observed for intra-patient registration, gets even worse in the inter-patient case. The definition of acceptable deformations between two potentially very different patients can be extremely difficult. In the end, inappropriate regularisation may lead to poor registration and hence inaccurate delineation of organs in the query image. The usefulness of the atlas might then be lower than expected.

The use of more than one atlas improves segmentation quality. The multiple, average, or similarity-based atlases can involve images of several patients, hence making the method more representative of the population and more robust when processing query images that are possible outliers. In all cases, the key point is to determine and use an atlas that is not too different from the query image, in order to avoid pushing registration to its limits. Similarity-based atlases are those that come the closest towards this objective. The choice of the best atlas among all others is derived from an indirect measurement (the registration metric that measures the dissimilarity between the fixed and deformed images, or the magnitude of deformation). With the multiple or average atlases, several candidate segmentations have to be merged at the end of the process. Label-fusion methods can be used for this purpose, with the risk that those methods can generate organs with disconnected pieces, which is often hardly plausible from an anatomical point of view.

Because of all these issues, the atlases still suffer from a slow adoption by the physicians. Moreover, it is also noteworthy that atlases make rather limited use of prior knowledge, which is contained, for one part, in the organ contours drawn by the experts and, for the rest, in the parametrisation and regularisation of non-rigid registration. Therefore atlases can be considered to involve only static models and cannot adapt dynamically by using images available in a training set. For instance, optimal registration meta-parameters are likely to change for each considered body part (chest, pelvis, head, etc.). Hence, some tuning is required for each new atlas.

The contours in the atlases contain prior knowledge about organs depicted in the image but they are only deformed and most of the information conveyed by the contours remains implicit (e.g. shape, appearance) and likely underexploited.

Statistical models are an alternative that address this issue make more explicit use of such prior information to help image segmentation. Unlike atlases, the images are not registered but the shapes and, sometimes, the appearance of the organ, are learnt in order to be found in a query image.

2.3.2 Statistical models

In the statistical models, the shape or the appearance of an object is learnt from a training set. The learning model is only able to identify one object at a time. Statistical models are based on the assumption that the shape of an object can be represented by a mean shape that can vary following different modes (Heimann and Meinzer 2009). The variation modes describe possibles deformations of the mean shape. In practice, the mean shape and the variation modes can be computed from the training set.

In this subsection, two methods of statistical models are described. The first one is the statistical shape model (SSM) that only takes into account the shape (Cootes, Taylor et al. 1995). The second one, the statistical appearance model (SAM), is an extension of the SSM that also takes into account the intensities and textures in the shape (Cootes et al. 2001). Before developing the methods, let us describe how the shapes are represented.

Representation of the shapes

In the training set, the representation of the object can be encoded in various formats (see 2.1 for details):

- binary mask
- fuzzy mask
- contour meshes

It is easily possible to convert from one format to another but sometimes information loss is inevitable.

Generally, those structures are converted into a landmark representation where the shape is reduced to a few points. A landmark is not necessarily a salient point but it has to represent the same point in each object instance. The representation of a 3D shape $\boldsymbol{x}$ composed of M landmarks can be expressed as:

$$\boldsymbol{x} = (x_1, y_1, z_1, \ldots, x_M, y_M, z_M)^\intercal \ ,$$

where x_i, y_i, z_i are respectively the coordinates in the axis x, y, z for the ith landmark.

Some additionnal information about connectivity between landmarks can be stored to obtain a mesh representation that will improve the result provided by the search algorithm. It is possible to use other kinds of representation such as medial models (Pizer et al. 2003) but their use is more anecdotal. The SSM with landmarks are sometimes called Point Distribution Models (PDM) (Heimann and Meinzer 2009).

Having the same dense landmark representation for each shape of the training set is difficult. It is not feasible to make the identification manually in 3D as it requires too much time and suffers from a lack of reproducibility. Several automatic shape correspondence techniques address this issue:

Mesh-to-mesh registration (Besl and McKay 1992; Rangarajan et al. 1997; Subsol et al. 1998). When the objects are represented by meshes, they have to be registered in order to have common landmarks. Algorithm allowing surface matching are used to register the meshes. The meshes can have different number of vertices. The algorithm delivers a transformation from a surface to another. The landmarks are defined in the reference surface and are propagated to all the meshes.

Mesh-to-volume registration (Shen et al. 2001). When working with medical images, the objects are often represented by several contour lines in different slices of the images. Those contour lines define a volume of pixels. The volumes from the different datasets can be registered to a global mesh where the landmarks are defined. After registration, the landmarks can be propagated from the mesh to the images.

Volume-to-volume registration (Frangi et al. 2001; Rueckert, Frangi et al. 2001). Instead of registering the mesh to each images, an atlas containing the mesh can be registered to each image of the dataset. The deformation field is applied to the mesh to obtain the landmarks for each image.

Parametrisation-to-parametrisation registration (Kelemen et al. 1999). Parametrisation is a bijective mapping function between a mesh and a common base domain. The common base domain is usually a very simple element. For 2D images, it is usually a circle. Nevertheless, the problem becomes more complex with 3D images and depends on the topology of the object. If the mesh describing the object is closed without self-intersection, it can be transformed into a sphere. The landscapes are defined on the common base domain and reported to the meshes.

Population-based optimisation (Kotcheff and Taylor 1998; Davies et al. 2002). As for the parametrisation-to-parametrisation registration, a mapping function between the meshes and the common base domain is used. In this case, the mapping function is modified to optimise a cost function describing the quality of the model.

In radiotherapy, due to the representation of the object boundaries, the mesh-to-volume and volume-to-volume are the most used techniques to define the landmarks.

Once the landmarks are well-defined on each image of the training set, the shapes are defined. By definition, shapes are invariant in translation, rotation, and scaling. Hence, they are aligned in a common coordinate system to become invariant to each other. To performed the alignment, a generalised Proscrutes alignment (GPA) is often realised (Gower 1975; Goodall 1991). This allows finding the translation, rotation, and scaling for each shape in order to be well aligned. Once all the shapes are aligned, the model can be learnt.

The learning of the model

The general idea is to extract a mean shape and a number of modes of variation from a set of images. The goal is to be able to represent any shape variation by the mean shape combined with the weighted mode. In practice, a principal component analysis (PCA) is usually used. Due to their representation, extraction of the mean shape $\bar{\boldsymbol{x}}$ is straightforward:

$$\bar{\boldsymbol{x}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i \,,$$

where N is the number of shapes in the training set, $\boldsymbol{x}_i$ is the shape representation of the ith shape. Based on all representations, the covariance matrix $\boldsymbol{S}$ can be computed, using,

$$\boldsymbol{S} = \frac{1}{N-1} \sum_{i=1}^{N} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}) (\boldsymbol{x}_i - \bar{\boldsymbol{x}})^{\intercal} \,.$$

The eigendecomposition of $\boldsymbol{S}$ gives the principal modes $\boldsymbol{\phi}_j$ (eigenvector) and their associated variances λ_j. Because of the rank of the matrix, the total number of modes (m) is at most $\max((N-1), 3M)$. We assume that the eigenvectors and eigenvalues are ordered so that $\lambda_j > \lambda_{j+1}$. Any shape can be

represented by a combination of the different modes,

$$\boldsymbol{x} = \bar{\boldsymbol{x}} + \sum_{j=1}^{c} b_j \boldsymbol{\phi}_j \,, \tag{2.3}$$

where b_j are the weights for each mode. Usually, c is chosen so that the ratio between $\sum_{j=1}^{c} \lambda_j$ and $\sum_{j=1}^{m} \lambda_j$ is in the range $0.9 - 0.98$. Choosing c is a trade-off between preserving enough possible variations and avoiding variations resulting from noise in the shapes. In general, a ratio below 0.9 does not keep enough mode to describe all the possible variation in the shape. On the other hand, a ratio above 0.98 keeps too many modes whose some of them derive from the noise in the shapes.

Equation 2.3 can be rewritten in matrix form,

$$\boldsymbol{x} = \bar{\boldsymbol{x}} + \boldsymbol{\Phi}\boldsymbol{b} \,,$$

where $\boldsymbol{\Phi} = (\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_c)$ and $\boldsymbol{b} = (b_1, b_2, \ldots, b_c)^\intercal$. The parameters in $\boldsymbol{b}$ can be determined by

$$\boldsymbol{b} = \boldsymbol{\Phi}^\intercal \left(\boldsymbol{x} - \bar{\boldsymbol{x}}\right) \,.$$

In order to have plausible shapes, the weights $\boldsymbol{b}$ need to be constrained. A local constraint is a solution to force each b_j to stay within 3 standard deviations of the mean,

$$-3\sqrt{\lambda_j} < b_j < 3\sqrt{\lambda_j} \,.$$

The constraint on $\boldsymbol{b}$ can also be set more globally by forcing it to stay in an hyperellipsoid,

$$\sum_{j=1}^{c} \frac{b_j^2}{\lambda_j^2} \leq E \,,$$

where E is a constant that can be chosen from a χ^2 distribution.

In the case of SAM, not only the shape but also the intensity and texture of the object are taken into account. The first step consists in creating a SSM. All shapes of the training set and their contents (pixel intensities) are deformed to fit the mean shape. Information about the texture is extracted from the shapes. The intensities form a texture vector $\boldsymbol{g}$ for each shape. The texture vector for a mean shape of L pixels is expressed by

$$\boldsymbol{g} = (g_1, g_2, \ldots, g_L)^\intercal \,,$$

where g_i is the gray value intensity of the ith pixel. Normalisation of $\boldsymbol{g}$ is usually performed so that its mean is zero and its variance is one.

As for the shape, a mean and modes of variation can be computed for the texture with PCA (Cootes et al. 2001). This leads to the following system

$$\begin{cases} \boldsymbol{x} = \bar{\boldsymbol{x}} + \boldsymbol{\Phi}_{\mathrm{s}} \boldsymbol{b}_{\mathrm{s}} \\ \boldsymbol{g} = \bar{\boldsymbol{g}} + \boldsymbol{\Phi}_{\mathrm{g}} \boldsymbol{b}_{\mathrm{g}} \end{cases},$$

where $\boldsymbol{\Phi}_{\mathrm{s}}$ (respectively $\boldsymbol{\Phi}_{\mathrm{g}}$) is the matrix of modes for the shape (resp. the texture) and $\boldsymbol{b}_{\mathrm{s}}$ (resp. $\boldsymbol{b}_{\mathrm{g}}$) is the vector of weight for each mode of shape variation (resp. texture variation). The system can be rewritten in order to have only one vector of parameters. At the same time, a weight diagonal matrix $\boldsymbol{W}_{\mathrm{s}}$ is added to take into account the difference of units between the shape and the grey models,

$$\boldsymbol{b} = \begin{pmatrix} \boldsymbol{W}_{\mathrm{s}} \boldsymbol{b}_{\mathrm{s}} \\ \boldsymbol{b}_{\mathrm{g}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{W}_{\mathrm{s}} \boldsymbol{\Phi}_{\mathrm{s}}^{\intercal} (\boldsymbol{x} - \bar{\boldsymbol{x}}) \\ \boldsymbol{\Phi}_{\mathrm{g}}^{\intercal} (\boldsymbol{g} - \bar{\boldsymbol{g}}) \end{pmatrix}.$$

A PCA can be computed on the parameter vector $\boldsymbol{b}$ to obtain a vector of appearance parameters $\boldsymbol{a}$ controlling both the shape and texture. This gives

$$\boldsymbol{b} = \boldsymbol{\Phi}_{\mathrm{a}} \boldsymbol{a},$$

where $\boldsymbol{\Phi}_{\mathrm{a}}$ are the eigenvectors obtained with PCA. The system can be completely rewritten as a function of $\boldsymbol{a}$,

$$\begin{cases} \boldsymbol{x} = \bar{\boldsymbol{x}} + \boldsymbol{\Phi}_{\mathrm{s}} \boldsymbol{W}_{\mathrm{s}}^{-1} \boldsymbol{\Phi}_{\mathrm{a}s} \boldsymbol{a} \\ \boldsymbol{g} = \bar{\boldsymbol{g}} + \boldsymbol{\Phi}_{\mathrm{g}} \boldsymbol{\Phi}_{\mathrm{a}g} \boldsymbol{a} \end{cases},$$

where $\begin{pmatrix} \boldsymbol{\Phi}_{\mathrm{a}s} \\ \boldsymbol{\Phi}_{\mathrm{a}g} \end{pmatrix} = \boldsymbol{\Phi}_{\mathrm{a}}$.

By grouping the matrices, the system becomes:

$$\begin{cases} \boldsymbol{x} = \bar{\boldsymbol{x}} + \boldsymbol{Q}_s \boldsymbol{a} \\ \boldsymbol{g} = \bar{\boldsymbol{g}} + \boldsymbol{Q}_g \boldsymbol{a} \end{cases}, \tag{2.4}$$

where $\boldsymbol{Q}_s = \boldsymbol{\Phi}_{\mathrm{s}} \boldsymbol{W}_{\mathrm{s}}^{-1} \boldsymbol{\Phi}_{\mathrm{a}s}$ and $\boldsymbol{Q}_g = \boldsymbol{\Phi}_{\mathrm{g}} \boldsymbol{\Phi}_{\mathrm{a}g}$. The $\boldsymbol{Q}_s$ and $\boldsymbol{Q}_g$ matrices describe the modes of variation extracted from the training set. Learning the model consists in computing those two matrices.

The SSM and SAM are generative models. Indeed, it is possible to generate

shapes (respectively appearances) by initialising the values of the vector $\boldsymbol{b}$ (resp. $\boldsymbol{a}$). Even if a generative model is not necessary in radiotherapy, this property can be used to verify that the model has correctly learnt the shape and textures.

The search of the object

Once the model is learnt, a new shape can be searched in a query image. Like in the previous subsection, the method is first explained for the SSM and subsequently, refined to the SAM.

As already stated above, the shape is invariant to global scaling, translation and rotation. A spacial transformation is mandatory to project the shape in the image space. The transformation allows translation, rotation, and scaling of the shape. The instance of the model in the image is defined by a spatial transformation T and a shape with parameters $\boldsymbol{b}$,

$$\mathbf{x} = T\left(\bar{\boldsymbol{x}} + \boldsymbol{\Phi}\boldsymbol{b}\right) .$$

There are several methods to initialise the parameters of the transformation. One of them consists in asking a human operator to provide an approximative position of the object. Fully automatic solutions also exist. Among them, let us cite the affine registration based on pixel intensities (Fripp et al. 2005) and the use of histogram information to estimate the position of the shape (Soler et al. 2001). Another solution is to conduct a global search (Hill, Cootes et al. 1992; Hill and Taylor 1992; Pitiot et al. 2002; Stegmann et al. 2001).

Once the approximate position of the shape is known, optimal displacements are evaluated for each landmark. Usually, the optimal displacement $d\mathbf{x}_\mathrm{p}$ is searched orthogonally to the shape. All landmarks are moved perpendicularly to the shape until they encounter a variation of the pixel values. The parameters of the transformation T are optimised by a Procrustes analysis between $\mathbf{x}$ and $\mathbf{x} + d\mathbf{x}_\mathrm{p}$. The shape has to be updated to reduce the residual displacement $d\mathbf{x}_\mathrm{s}$. It is transformed into parameter variation

$$d\boldsymbol{b} = \Phi^\intercal T^{-1}\left(d\mathbf{x}_\mathrm{s}\right) ,$$

where T^{-1} is the inverse of T. As the transformation is only composed of translation, rotation, and scaling, the computation of the inverse is straightforward. When the parameters are updated, the shape is modified and the positions of the landmarks are therefore revised. By incrementally computing the local optimal

displacements of the landmarks and updating the parameters, the shape is fit to the object. Those two steps are repeated until satisfying a given criterion (e.g. when the landmark displacements or parameter displacements are small enough).

In the case of appearance models, the search of the object is different as it also takes into account the textures inside the shape. In the model space, the shapes are set invariant to translation, rotation, and scaling. Similarly, normalisation of the textures makes them invariant too. To be able to use an instance of the model in the image, a transformation T must be applied. This transformation has parameters that can be joined with the parameters of the models. They can thus be optimised together. Those joint parameters are denoted $\boldsymbol{p}$.

The current texture in the image space $\mathbf{g}_{\text{im}}$ is converted by using the inverse transformation T^{-1} to be compatible with the model space. The texture in the model space $\boldsymbol{g}_{\text{m}}$ is obtained from (2.4) with the current parameter $\boldsymbol{a}$. The vector of residual $\boldsymbol{r}$ is computed by comparing the two textures,

$$\boldsymbol{r}(\boldsymbol{p}) = T^{-1}(\mathbf{g}_{\text{im}}) - \boldsymbol{g}_{\text{m}} \,,$$

where $\boldsymbol{r}(\boldsymbol{p})$ is the vector of residuals when using parameters $\boldsymbol{p}$. The error can be evaluated from the residuals by computing the squared norm of $\boldsymbol{r}$. The hypothesis is made that there exists a constant relationship, described by matrix $\boldsymbol{R}$, between the texture residual $\boldsymbol{r}(\boldsymbol{p})$ and the parameter updates $d\boldsymbol{p}$. This relationship can be expressed as

$$d\boldsymbol{p} = -\boldsymbol{R}\boldsymbol{r}(\boldsymbol{p}) \,.$$

Matrix $\boldsymbol{R}$ can be computed in different ways from the training set. Its definition is crucial as it guides the search of the appearance (Cootes et al. 1998; Cootes et al. 2001; Donner et al. 2006).

As for the atlas, the search of the shape is sensitive to noise in the image. To reduce the risk of falling into a local minima during the optimisation of the parameters, the search is carried out iteratively at different scales, from coarse to fine (see Figure 2.10 in Subsection 2.3.1).

Pros and cons

The SSM and SAM require to have well aligned shapes from the training set. The alignment defines the landmarks that are used to learn the models. It

is impossible to determine if, after the alignment of the shape in each image, the landmarks represent exactly the same point on the boundary of the object. However, as they are generative models, SSM and SAM allow verifying their quality before using them.

If the number of images used to build the model is low, there is a non-negligible risk to overfit the shape. The overfitting occurs when the model is too specific to the training set and is not able to fit new acceptable shapes. The model usually overfits because the sampling of the possible shapes is too low. The model also overfits when it tends to learn some of the noise in the shapes. In this case, the model loses its robustness when it delineates query images.

When using SSM, any kind of modalities can be used in the training set and the query image. Indeed, only the shapes of the contours are learnt and searched for. However, the boundaries of the object have to be visible in the query image. Low contrast near the object boundaries in the query image can degrade the quality of the delineation. Moreover, the landmarks have to be well defined so they can be identified in the query image.

For both SSM and SAM, the search of the shape requires an initialisation. The position of the shape has to be provided manually or automatically. The automatic methods can be slow, especially when they work with 3D images. If the initial position is too distant (rotation, translation, and scale) from the searched object, this can lead to poor object identification.

As for the atlases, building a SSM or SAM does not required extra work for the physician. Indeed, existing images and delineations can be used to determine landmarks. The atlases can be used with only one image (simple atlas) or several images (multiple, similarity-based, and average) in the training set. This is not the case for the statistical models that require as many images as possible in the training set to correctly learn the shape of the object. Unlike the atlases, the statistical models use a learning model. The mean shapes and modes defining the model are indeed learnt from the training set. The same process can be used to train models that recognise different objects in different patient body parts. Nevertheless, before learning the model, the landmarks have to be defined. Because their definition remains very dependent of the object of interest, this first step can be considered as a static part of the model. This complex static step makes the statistical models more difficult to generalise.

2.3.3 Machine learning

Methods using machine learning for medical image delineation are very similar to those presented in subsection 2.2.2. They use primarily pixel intensities to segment the image. In addition, richer features are sometimes built in order to improve classification.

Classifiers are used for brain tumour delineation in Zhang et al. (2004). They proposed to use two features per pixel corresponding to their gray-level in MRIs with and without contrast. By using two different kernels with a SVM classifier, they are able to segment the brain tumour. Geremia et al. (2011) suggest the use of RF to segment brain lesions. Instead of using the pixel intensities, richer features are constructed based on the neighbourhood of the pixel in different MRIs. A RF is then learnt and used with those features.

Healthy organs can also be delineated with classifiers. An automatic liver delineation method is discribed by Luo et al. (2009). Texture features are computed for each pixel by taking into account the neighbouring pixels. The pixels are subsequently classified by a SVM classifier. As suggested by Criminisi et al. (2009), multiple organ localisation can be achieved with a RF classifier. Rich features are computed from the pixel neighbourhood. The features are used by a RF classifier to provide the probability of each pixel to belong to a given organ.

The use of machine learning methods remains anecdotal compared to atlas and statistical model methods. It remains difficult to define rich features from medical images. Moreover, compared to images acquired with digital still cameras, medical images contain little texture information and noise or artefacts are stronger.

2.3.4 Method combinations

Combinations of atlases, statistical models, machine learning and other image segmentation methods exists. Most of the time, the result of a method is used as an initialisation point for other methods. Method combinations aim at compensating the weaknesses of each method used alone.

In Qazi et al. (2011), they suggest the use of an average atlas to initialise a statistical model. The average atlas allows approximating the position of organs. By using a statistical model, the boundaries are adjusted to give a better delineation. In Fortunati et al. (2013), a multiple atlas is used to produce a probability map of the location of organs in the head and neck area. Intensity

histogram information is extracted from multiple atlases and used with the probability map to build a graph representation of the model with optimised weighted edges. A min-cut method is then applied on the graph representation to obtain the delineation of the organ in the image. By using the intensity histogram information, their model becomes less static and can adapt with a training set.

The combination of different methods naturally increases the number of different meta-parameters. Moreover, multiple methods also require more computation time.

Current methods use a small amount of prior knowledge to build the model. With the atlases, all the knowledge is static and the model cannot be automatically adapted to the training set. The SSM/SAM methods build the model from the dataset, but the search of the shape in the image uses nearly no prior knowledge from the training set. The use of machine learning methods for image delineation remains anecdotal due to their difficulty to extract rich features. In the following chapter, a new method based on machine learning is proposed. Most of its parameters are adjusted automatically with a training set. The method has the ability to dynamically evaluate rich features during the classification process.

Chapter 3

Incremental image delineation

In this chapter, we propose an alternative to atlases and statistical models. It has been observed that atlases are static models, and cannot adapt dynamically their parameters by using the images from the training set. The parametrisation and the regularisation of the registration remain difficult and require adaptations to be applicable to all parts of the human body. Unlike the atlases, the statistical shape model (SSM) and statistical appearance model (SAM) are learning models. They use the delineations of the object from the training set to learn properties about their shape and texture. By learning the shapes and their variation modes, they are theoretically adaptable to any part of the human body. However, the variability of the human anatomy is quite complex and the shapes of the organs remain in some cases difficult to be learnt robustly. Moreover, the search of the object is realised without using prior information about its position.

This chapter aims at describing a method that relies on learning models, which are able to detect the position of any object in the image. This method makes the hypothesis that some objects are easier to identify in the image than others. By finding those objects, some richer information is used to improve the identification of harder-to-find objects. From the simplest to the hardest, all objects can be delineated by working incrementally.

Section 3.1 gives an overview of the proposed delineation method as well as a description of the datasets and metrics used to validate the method. Section 3.2 focuses on the pre-processing of the image in order to be usable with the

incremental method. Section 3.3 explained how richer information is extracted all along the incremental process. The details about the incremental method are provided in the Section 3.4, while the establishment of the sequence of identification is detailed in Section 3.5. Section 3.6 lists the different parameters of the method. Finally, Section 3.7 gives the advantages and disadvantages of using our automatic delineation method.

3.1 General overview

The proposed method tries to mimic the work of the physicians, from the vision of the image by the eyes to the recognition of objects by the brain. Human vision can be described in a few simplified steps. Photoreceptor cells located in the retina are excited by photons. The resulting signal goes to different layers of neurons before reaching the optic nerve fibres. By passing through the layers, the signal is progressively converted from raw signal to simple shape such as movements and edges (Hubel 1995). By spotting the gradient crests, the optic nerve reduces the amount of information that needs to be processed by the brain. Indeed, areas in-between crests can be treated as being homogeneous.

An untrained person without knowledge of human anatomy would only see different homogeneous areas of different intensities on medical images. Physicians working in radiation therapy are able to recognise different organs or objects. During their training, the physicians study the human anatomy, look at images with delineated objects, and delineate objects on new images under the control of more experimented physicians. Throughout the formation, their brain builds a model to be able to recognise objects based on human vision and acquired knowledge.

Once the physicians have finished their training, they have acquired enough information to be able to delineate medical images. Our hypothesis is that they make the delineation incrementally. Their optic nerves detect the edges and thereby, the homogeneous areas. Their brain firstly identifies easy-to-recognise areas based on their gray-level and positions in the image (e.g. the patient body and surrounding objects like the couch). As they know the position of the first identified area, they are then able to pinpoint the localisation of other objects (e.g. by using their relative position to the identified area). Incrementally, more and more objects are identified. Most of the areas in the image are implicitly recognised but only the objects corresponding to target volumes (TVs) and organs at risk (OARs) are really delineated and recorded on the image. It should

only take a few seconds before a physician recognises most of the homogeneous areas of a 2D image.

To resume, few important stages can be identified over the process. Firstly, the edges and homogeneous areas are identified by the eyes. Secondly, the physicians take anatomical lessons, train with delineated images and the help of experts. Finally, the physicians can incrementally delineate objects of new images by recognising some areas implicitly that helps the recognition of other ones. Each of those stages is dependant of the previous ones.

The proposed method tries to delineate objects in images in the same way as the physicians. For each of the stages, a method is suggested to mimic their behaviour.

3.1.1 Identification of homogeneous areas

The identification of homogeneous areas is common to the learning and the delineation step. At this stage, the eyes detect edges, and by extension, the homogeneous areas. To mimic this operation, we chose to use the watershed algorithm. As presented in Subsection 2.2.1, the watershed algorithm works on the gradient magnitude of an image to segment it in homogeneous areas or volumes. Gradient magnitude allows detecting edges mostly like the eyes can do it. Homogeneous areas or volumes are called superpixels as they represent a group of similar neighbouring pixels. All pixels belonging to an homogeneous area can be considered as being part of the same object. Working with superpixels rather than pixels significantly reduces the amount of data to process. By using the watershed algorithm, the behaviour of the eye is imitated: raw data (corresponding to pixels) are converted into simple homogeneous areas (corresponding to superpixels).

3.1.2 Delineation of a new image

We make the hypothesis that a model has been built (in the physician's brain or by a computer) to delineate objects of an image. This model is able to identify the belonging of homogeneous areas to the organs in an image depending on acquired information. It is thus used to delineate the objects of a new image. At the beginning, the model allows identifying some homogeneous areas based, for example, on their positions in the image and their gray-level intensities. By identifying areas, richer information can be acquired such as the relative position to an object. This information can be used to identify more complex

areas. Eventually, all the objects/organs in the image can be recognized by repeating those tasks.

During the delineation step, two kinds of information can be distinguished. The first one, which is called intrinsic information, is acquired when looking at the image. It is, for example, the position and colour of areas. This information is directly related to homogeneous areas. It is static and does not change during the delineation. The second kind of information is contextual and is acquired during the recognition of the object. Indeed, once an object is identified, positions of homogeneous areas, relatively to this object, are established by the brain. This information is richer then the first one, and can be modified and refined throughout the identification of the objects. We call this second information, extrinsic information.

Our method tries to mimic the physicians by using the intrinsic and extrinsic information (see Figure 3.1). Let us make the hypothesis that a model has already be learnt. Like the eye, the watershed algorithm segments the image in homogeneous areas, the superpixels. For each of those superpixels, intrinsic information can be acquired. Information is organised into features attached to each superpixel. Intrinsic features can be the position of the homogeneous area in the image or the mean intensity of the area. The feature extraction is the step allowing to acquire values for features. A value is thus obtained for each feature of each superpixel. The model can use these values to identify some superpixels as being part of an object. This is the labelling step. Based on the identified superpixels, extrinsic information can be extracted. The extrinsic features are related to the newly identified object (e.g. the distance to the newly identified object). This richer information allows the model to identify other objects. By repeating the steps of feature extraction and labelling with the model, the method is capable of delineating objects in the entire image.

3.1.3 Learning the model

During the training period, physicians study the human anatomy and learn to identify objects in medical images and how to read them. Step by step, they learn to recognise the most difficult objects. We think that they learn a hierarchy in the process of object identification. Indeed, physicians know that some objects are easier to identify than other ones. They also know that once the easiest object is identified, the others become easier to find. In other words, it is likely that a sequential work allows identifying more easily the difficult objects.

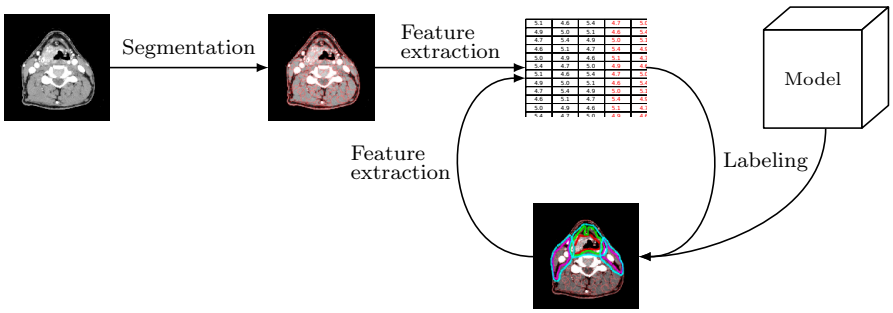


Figure 3.1 – The delineation. The delineation is performed with the help of a learnt model. Homogeneous areas in the image are identified by the watershed algorithm (i.e. during the segmentation step). Intrinsic features are extracted from each superpixels. A first object is identified (labelling) based on the model previously learnt. Extrinsic features related to the newly identified object are extracted and used for the next labelling. Once all the regions have been identified, a final delineation is performed based on all the extracted features.

The proposed method tries to mimic the hypothetical behaviour of the physicians (see Figure 3.2). To train our model, images delineated by experts have to be provided. The images are segmented into superpixels by a watershed algorithm to imitate the eyes behaviour. Each delineated object is make up of at least one superpixel. As all the objects are already delineated, both intrinsic and extrinsic features can be extracted for each superpixel. Based on all those features, a sequence of identification is learnt. This sequence corresponds, for the physicians, to the learning that some objects are easier to identify than others. By working sequentially, the problem is split into subproblems. For each subproblem, a submodel is learnt in order to solve it. Submodels are simple learning models, which are built to identify one specific object based on the available features at the corresponding step of the sequence. All submodels and the sequence are learnt from the training set. All together, they form the model.

3.1.4 Dataset and metric

The suggested method was applied to different datasets (see Figure 3.3). The datasets are made up of images and masks (i.e. the true labels) defining the position of each object. The method was firstly applied to the synthetic caterpillar dataset (Bernard et al. 2012). This dataset is very simple but

Figure 3.2 – The learning. The model is learnt from images with delineated objects. Homogeneous areas of the image are identified by the watershed algorithm (i.e. during the segmentation step). Features are extracted for each superpixel. The model, which is formed of the sequence and submodels, is learnt from the features.

required the use of advanced method to be solved. Afterwards, the method was tested on the synthetic circle dataset (Bernard et al. 2013; Bernard et al. 2014b; Bernard et al. 2014c). This dataset was built to be more realistic and integrate typical medical image noise. Finally, the method was tested on a patient dataset (Bernard et al. 2014a; Bernard et al. 2014b). In this chapter, we only present the results obtained with the patient dataset, which is more challenging than synthetic images. In addition to 2D patient images, preliminary results obtained with 3D images are also presented. More details about the synthetic caterpillar and synthetic circle dataset can be found in the corresponding papers (Bernard et al. 2012; Bernard et al. 2013; Bernard et al. 2014b; Bernard et al. 2014c).

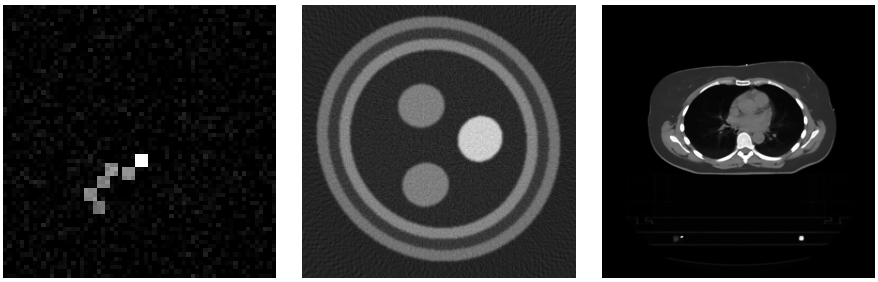


Figure 3.3 – Datasets used to validate the method. Left, an image of the synthetic caterpillar dataset. Middle, an image of the synthetic circle dataset. Right, an image of the patient dataset.

Patient dataset

To create a dataset with patient images, forty-nine 3D computed tomography (CT) scans were collected in the radiotherapy service of the Saint-Luc university hospital (Brussels, Belgium). All these images were part of the routine protocol for female patients treated by radiotherapy after breast cancer surgery. All images were acquired with varying slice thicknesses (2, 3, or 5 mm). In each tomographic acquisition, an axial slice was selected and extracted at the level of the seventh thoracic vertebra. Each slice contains 512^2 square pixels with edge length equal to 1.074 mm. The luminance of each pixel ranges from -2048 to 2048 Hounsfield units (HU) and indicates the electronic density of the depicted material. On each image, 23 objects were identified. They are shown, reported and commented in Figure 3.4 and Table 3.1.

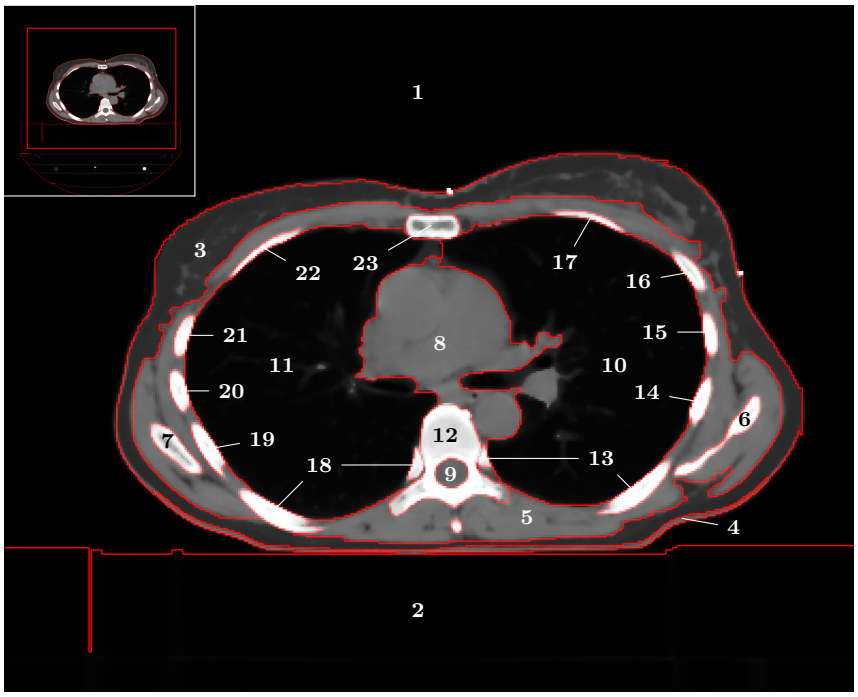


Figure 3.4 – An image from the patient dataset with all objects. The objects are described in Table 3.1. The image was cropped in order to improve its readability. The cropped area is represented on the upper left corner.

1. Out of field and air[a]	13. Left rib 7[h]
2. Treatment table[b]	14. Left rib 6
3. Fat and breast[c]	15. Left rib 5
4. Back skin[d]	16. Left rib 4
5. Muscles[e]	17. Left rib 3
6. Left scapula	18. Right rib 7[h]
7. Right scapula	19. Right rib 6
8. Mediastinum[f]	20. Right rib 5
9. Spinal canal	21. Right rib 4
10. Left lung	22. Right rib 3
11. Right Lung	23. Sternum
12. Vertebra[g]	

Table 3.1 – Images from patient dataset. List of objects labelled in Figure 3.4.
[a]'Out of field and air' includes the padded corners outside the field of view (-2048 HU), as well as air surrounding the patient in the cylindrical field of view of the scanner (-1000 HU).
[b]The low electronic density of the table aims to minimise X-ray attenuation.
[c]If clips were used in breast cancer surgery, they are considered as belonging to the breast.
[d]Due to weak contrast with fat, only part of the skin could be identified in the patients' back.
[e]All muscles are gathered in the same object. Any small area with lower electronic density between two muscles is considered to belong to the muscles.
[f]The mediastinum includes all organs between the lungs, namely, the heart, arteries, veins, and oesophagus.
[g]The vertebra can make up of one or several parts, depending on its orientation.
[h]The left and right seventh ribs are sometimes split in two parts, depending on their orientation, and one of these parts often lies near the vertebra (see Figure 3.4).

A smaller dataset of 3D images has been constructed from the same set of patient images (chest images). For each images, only the slices between the 4th vertebra and the 12th vertebra were kept. This leads to 3D images encompassing almost the lungs. At the time of writing, this dataset contains 7 images and has only been tested to evaluate the usability of the watershed algorithm and the feature extraction on 3D images. A total of 43 objects has been identified on those images.

Metrics

To avoid overfitting, the method was trained on part of the dataset, and then tested on the remaining images. The results were compared to the true labels to validate the method.

When working with images, the number of pixels associated with an object may vary a lot in patient images, depending on the nature of the object. To correctly measure the error of the delineation without bias, it is recommended to use the balanced classification rate (BCR) (Helleputte and Dupont 2009). The BCR is defined as

$$\mathrm{BCR} = \frac{1}{C} \sum_{i=1}^{C} \frac{|T_{Y_i}|}{|Y_i|} \,,$$

where C denotes the number of objects, $|Y_i|$ is the number of pixels that should be associated with the object Y_i, and $|T_{Y_i}|$ is the number of pixels correctly labelled in object Y_i. In the case of multiple memberships, the pixel classified in n objects counts for $\frac{1}{n}$ in $|T_{Y_i}|$ if its true label is Y_i. The BCR ranges between 0 to 1, where 1 corresponds to a perfect classification. A random classification will produce a BCR around $\frac{1}{C}$.

In addition to the BCR, the stability of the sequence was analysed. In order to assess this variation, the Levenshtein distance (Levenshtein 1966), also called edit distance, is used. This distance measures the minimal number of insertions, deletions and substitutions to transform one sequence into another. The distances for all pairs of sequences that are obtained with the different training sets are computed. The bigger the average distance, the less stable the method is.

In the following sections, the different stages of the method are provided together with related results. More details about the experiments and complete results are gathered in the publications (see in appendices Bernard et al. (2012), Bernard et al. (2013), Bernard et al. (2014a), Bernard et al. (2014b) and Bernard et al. (2014c)).

3.2 Segmentation in homogeneous areas

The first step of the delineation is the segmentation of the image in homogeneous areas. During this stage, the method tries to mimic the eyes by segmenting the image in homogeneous superpixels. In this section, the automatic segmentation method is described and a contrast enhancement process is proposed to improve

its quality.

3.2.1 Automatic segmentation

In our case, the purpose of the automatic segmentation is to group neighbouring pixels of similar intensity in a common superpixel. We assume that all pixels in one superpixel belong to the same object. To make this over-segmentation possible, we chose to use the watershed algorithm proposed by Cousty et al. (2009). The principle of the watershed is explained in Section 2.2.1. As the watershed segmentation does not rely on prior knowledge, it can be computed even if the objects are unknown. Nevertheless, the method is very sensitive to noise and its use on an image without pre-processing leads to unusable results. As shown in Figure 3.5, the number of superpixels is huge without filtering, and their mean size is approximately equal to 5 pixels (4 pixels if the padded corners outside of the field of view are not taken into account). So small superpixels are useless. This is why the graph representation of the image has to be filtered before applying the watershed algorithm. By using the method proposed by Cousty et al. (2009), the number of superpixels can be determined beforehand. A comparison of the segmentations with and without the use of a filter is shown in Figure 3.5.

This first step of our delineation process is made by a watershed algorithm. As previously explained, this method is completely static. Only a small amount of prior knowledge is used to perform the segmentation in superpixels. In order to improve the segmentation, we suggest to increase the contrast of the image by using prior knowledge.

3.2.2 Contrast enhancement to improve the segmentation

The contrast in real images with full window of intensity is poor (see Figure 3.5). Indeed, it is almost impossible to distinguish different soft tissues such as fat and muscles. However, the separation is essential in order to be able to segment them in different superpixels. The segmentation result depends directly on the pixel intensity differences. Small differences in a zone lead to fewer superpixels in that zone. By enhancing the contrast in an area of interest, the pixel intensity differences increase and the number of superpixels after filtering increases as well.

When physicians have to interpret an image with low contrast issue, they enhance it manually, depending on the object that they want to delineate. This

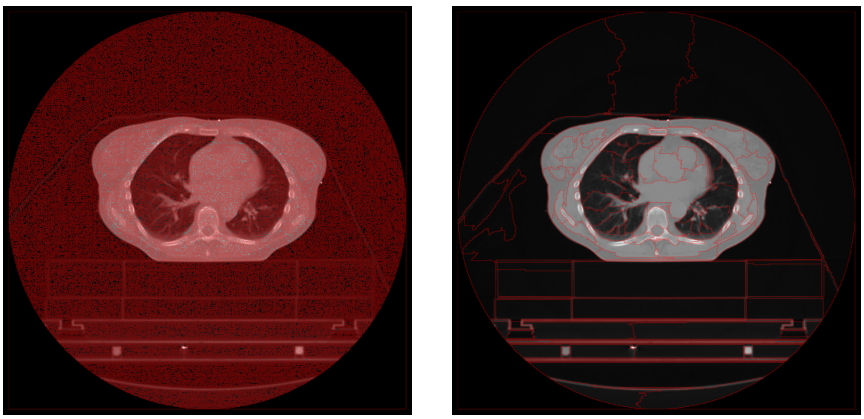


Figure 3.5 – Watershed segmentation applied to the same image. The red lines depict the border of the superpixels. On the left, the result without filter on the graph representation of the image (48972 superpixels). On the right, the result with filter on the graph representation of the image before applying the watershed algorithm (200 superpixels).

allows them to better discern subtle gray-level variations in the image. The contrast is usually increased by windowing the pixel intensities. Physicians can change their windowing during the delineation of objects.

With our method, the segmentation is performed only once by the watershed algorithm. The repetition of windowing is therefore impossible to realise. We suggest to optimise globally the intensities of pixels to improve the quality of the image segmentation. A better segmentation can be performed by increasing the contrast where borders have to be found, and decreasing it where borders are not requested. The contrast can be enhanced in all areas of interest by realising a piecewise linear transformation of the intensities. Figure 3.6 shows an example of transformation for images of patient dataset. After the conversion of the HU, soft tissues span more than 70% of the range of values, while they only take 14% without conversion. It can be thus seen that the soft tissues take a bigger range in the converted intensities. As the differences between pixel intensities in soft tissues are bigger, the gradient magnitude is bigger, leading to more superpixels in those areas. The opposite situation can be observed for the low-density areas (e.g. 'Out of field', 'Air', 'Table', and 'Lungs') that take 12% after conversion, while they take nearly 60% (25% if the 'Out of field' is removed) in the initial intensities. Figure 3.7 shows the result of segmentation

with and without using the intensity transformation. Superpixels get smaller and more numerous in soft tissues, while their numbers decrease outside the patient. It can also be seen that the borders of the bones are sharper and more easily detected.

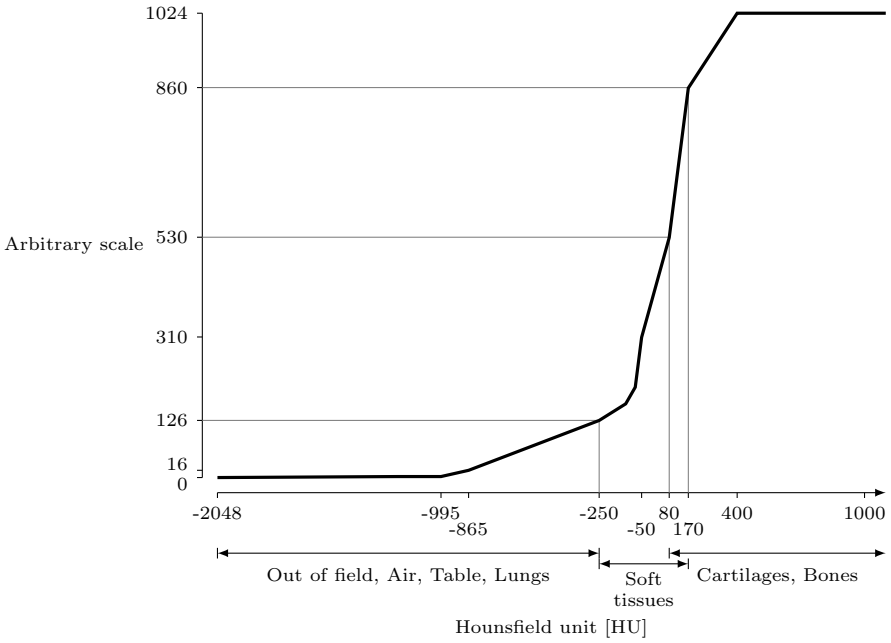


Figure 3.6 – Conversion of the Hounsfield units (HU) to improve the delineation. On the horizontal axis, HU present in the original image. On the vertical axis, the converted values. The critical part for the delineation (i.e. soft tissues) take a bigger part in the convert histogram. It is noteworthy that the function slightly grows between -2048 HU and -995 HU, while it remains constant after 400 HU. The reported values were manually obtained from several images.

The conversion has also been obtained for the 3D dataset. The obtained curve is slightly different as the objects to segment are slightly different between the 2D and 3D dataset.

The optimisation of the intensities in the image integrates prior knowledge before the segmentation by the watershed algorithm. This allows the segmentation to be more specific to the field of application and to improve its accuracy in the critical areas. Currently, this optimisation has to be manually done. We

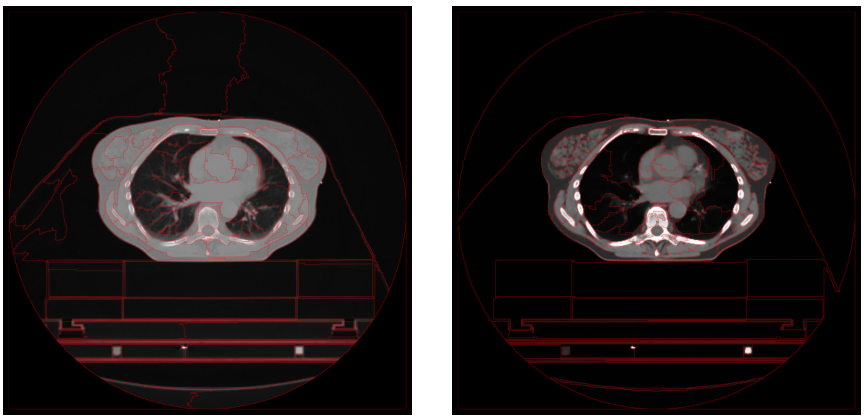


Figure 3.7 – Watershed applied to an image with and without optimisation of the intensity. The red lines depict the borders of the superpixels. On the left, the result of the segmentation (200 superpixels) without optimisation of the intensity. On the right, the result of the segmentation (200 superpixels) with optimisation of the intensity. Much more segmentation can be observed in the soft tissues when optimising the intensity. The border of the bones are also more easily detected.

discuss the feasibility of its automation in Chapter 4.

Other filters, which are not described in this document, can also be used as a pre-processing step to the segmentation. Those filters aim at reducing the statistical noise in the image that is inherent to the image acquisition process. It is advisable to use filters that preserve the edges. Indeed, the boundaries have to remain sharp to produce a good segmentation. The total variation filters are powerful for this task because they reinforce the uniformity in the image (Chambolle 2004). They were used in Bernard et al. (2014a) and Bernard et al. (2014b).

Segmentation of the image into superpixels gives the starting point of the process. If a border between two different objects cannot be detected at this step, they will most certainly be present in the same superpixel. The delineation will not be accurate as a superpixel can belong to only one object. In order to obtain the best segmentation (i.e. each superpixel contains pixels from one, and only one, object), the important tasks are to filter the image (e.g. the total variation filters), its graph representation (i.e. component tree filtering before using the watershed algorithm), and to optimise its intensity (i.e. the contrast

enhancement).

Once the image is segmented, features are extracted for each superpixel. The way they are extracted is critical as their values are used to determine the membership of a superpixel in an object.

3.3 Extraction of rich information

After having segmented the image, feature have to be defined and extracted for each superpixel in order to allow a valid delineation of objects.

Let us define $\mathcal{F}$, the set of features that can be extracted from a superpixel in an image. Two kinds of features can be defined: the intrinsic ones, in $\mathcal{F}_{\text{int}}$, and the extrinsic ones, in $\mathcal{F}_{\text{ext}}$. The intrinsic features are relative to a superpixel and can be computed without external information. As we have shown on the synthetic caterpillar dataset in Bernard et al. (2012), the intrinsic features are often not sufficient to perform an accurate identification of the object. Full list of intrinsic features used with the patient dataset is resume in Table 3.2.

Extrinsic features are also extracted to improve the delineation. Those features do not only rely on the superpixels but they are also related to a known object. By definition, the values of those features can only be computed if some superpixels are identified as being a part of an object. The complete list of extrinsic features used with the patient dataset is shown in Table 3.2. Table 3.3 presents the quantity of intrinsic and extrinsic features for each dataset used in our publications as well as for the 3D dataset. For the 3D dataset, some features related to the z axis were added to the features already defined in Table 3.2. Even, if a small number of extrinsic features is defined, due to the number of objects, they take an important part of the feature set.

When delineating objects from a new image, only the values of the intrinsic features can be computed just after the image segmentation. Those values are always known and never changed. On the contrary, values of the extrinsic features are initially unknown and can change. All the values of the extrinsic features can only be known if the positions of all objects are known. This means that at the beginning of the delineation, none of these values is known and the set of known features $\mathcal{F}_{\text{kn}}$ is equal to $\mathcal{F}_{\text{int}}$. At each iteration, an object is identified or updated. When a new object is recognised, the related features are added to $\mathcal{F}_{\text{kn}}$ and their values can then be computed. When an object is updated, the values of their extrinsic features are updated as well. At the end of the incremental process, the set of known features is full, so that

Intrinsic features	Extrinsic features
Average intensity	*For each object:*
Surface	Contiguity with the object
Contiguity with the border	*Along each axis (x and y):*
Along each axis (x and y):	Distance from centre to centre
Centre position	
Minimum pixel coordinate	
Maximum pixel coordinate	
Size	

Table 3.2 – Intrinsic and extrinsic features used in the patient dataset. Altogether, 11 intrinsic and 69 extrinsic features are extracted.

Dataset	#obj.	$\#\mathcal{F}_{\text{int}}$	$\#\mathcal{F}_{\text{ext}}$/#obj.	$\#\mathcal{F}$
Caterpillars (Bernard et al. 2012)	6	1	1	7
Circles (Bernard et al. 2013)	8	6	3	30
Circles (Bernard et al. 2014b)	8	9	3	33
Patients 2D (Bernard et al. 2014b)	23	11	3	80
Patients 3D (preliminary work)	43	15	4	186

Table 3.3 – Number of intrinsic and extrinsic features used in each dataset. #obj is the number of objects in the image. $\#\mathcal{F}_{\text{int}}$ is the number of intrinsic features. $\#\mathcal{F}_{\text{ext}}$ is the number of extrinsic features. $\#\mathcal{F}$ is the total number of features. As it can be seen, circles dataset was used with different number of extrinsic features in the publications.

$\mathcal{F}_{\text{kn}} = \mathcal{F}_{\text{int}} \cup \mathcal{F}_{\text{ext}}$.

The time required to compute the extrinsic features has to remain low. Indeed, they need to be computed several times during the incremental process. The time complexity is usually used to assess the speed of a method. It theoretically measures the speed of a process as a function of the entries. In our different contributions, all extrinsic features are calculable simultaneously with a time complexity equal to $\mathcal{O}(NC)$, where N is the number of superpixels in the image, and C the number of objects. In this latter case, if the number of superpixels or objects is doubled, the time required for calculation of the features only double. Nevertheless, if both the number of superpixels and objects are doubled, the time required is multiplied by four. Our defined features can easily be used with a large number of objects or superpixels as the time complexity of their computation is linearly dependant of the objects and superpixels.

Once the features and their values are extracted from the images, they are normalised so that each feature has a mean of zero and a standard deviation of one. The normalisation is learnt and performed on the features of the training set. The parameters of the transformation are kept to be used on the values of the features of the test set once they are extracted.

In the application to patient images (Bernard et al. 2014b), the importance of the extrinsic features to obtain a good delineation has been studied. The performance in delineation with a multiclass classifier was compared between classifiers using only intrinsic features (called blind) and classifiers with hypothetical knowledge of the extrinsic features in addition to the intrinsic ones (called oracle). The oracle classifiers are unrealistic as they assume that both intrinsic and extrinsic features are known without error from the beginning. This can only occur when all intermediary steps of the delineation do not make misclassification errors. Table 3.4 from Bernard et al. (2014b) shows that the oracle classifiers perform significantly better than the blind ones with all the tested classifiers (random forest (RF) and support vector machines (SVM)). If it was possible to extract the features without doing any error, the BCR would be nearly 100%. This means that the chosen features are enough to perform a good delineation.

		BCR(%)
RF	blind	84.15 ± 8.87^{a}
	oracle	98.56 ± 2.09^{b}
SVM	blind	85.17 ± 9.37^{a}
	oracle	98.49 ± 1.82^{b}

Table 3.4 – Result of the BCR obtained on the patient dataset. Measure of the BCR acquired with the blind classifiers, when using only intrinsic features, and the oracle classifiers, when using intrinsic and extrinsic features with random forest (RF) and support vector machines (SVM) classifiers. To established the difference between methods, a modified t-test is performed as suggested by Nadeau and Bengio (2003). Values of BCR with different letters are significantly different with a confident level of 0.95 (p-value $< 10^{-3}$).

It is possible to measure the BCR distribution as they can be computed individually on each image. Figure 3.8 shows the result of a kernel density estimator for the distribution of the BCR with the RF classifier. Most of the images obtain a BCR larger than 95%, while all the extrinsic features are well known. In contrast, only few delineations with only intrinsic features reach 95%

BCR.

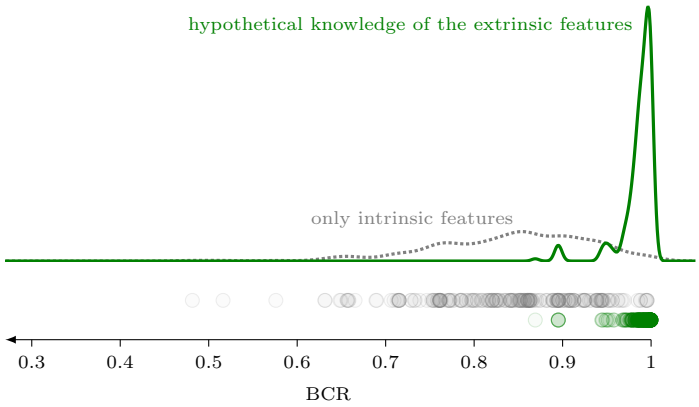


Figure 3.8 – Kernel density estimation of the BCR distribution obtained individually with the random forest (RF) classifier. Having a hypothetical perfect knowledge of the extrinsic features improves the distribution of the BCR compared to the only intrinsic features.

As it has been shown in this section, the features extracted from the image hold an important part in the delineation quality. The feature extraction gives us all the prior knowledge necessary for the learning and the delineation. This prior knowledge is directly extracted from the training set. The model that is learnt from the prior knowledge becomes specific to the training set. Changing the training set may change the prior knowledge and gives a different model. By doing so, our method can be adapted to any part of human body.

Segmentation in superpixels and feature extraction reduce the amount of data to be processed. Indeed, as shown in Table 3.5, the memory space required to record all features is nearly 10 times smaller than the size of the 2D images themselves. The difference is even bigger with 3D images, for which the ratio increases up to 20 in favour of features. Beside, features, additional information has to be extracted to compute some extrinsic features. For example, the contiguity of a superpixel to an object is evaluated with the help of an adjacency matrix. As the latter is sparse, its size is negligible ($\approx$ 40 kB for the 3D images). (Notice that such additional information is not taken into account in Table 3.5.)

We have studied the possibility to use more complex features such as the fuzzy spatial relation proposed by Bloch (1999). Nevertheless, their time

Dataset	Image size	Feature size	Compression rate (%)
Patients 2D	512 kB	62.5 kB	89.79
Patients 3D	[18.5 – 32.5] MB	1.06 MB	[94.27 – 96.74]

Table 3.5 – Estimation of the compression rate obtained by using feature extraction. The number of pixels in 3D images varies from one patient to another because of the patient anatomy and the slice thickness. This leads to different image sizes. Minimum and maximum values are given for image size and compression rate. The 2D patient dataset is segmented in 200 superpixels. The 3D patient dataset is segmented in 1500 superpixels. The size of the image assumes a bit depth of 16 whereas features are encoded in 32 bits floating point numbers.

complexity makes them unusable in practice. Moreover, they often require to keep the whole the image in memory to perform the necessary computations, while we have built features that can be learnt from pre-extracted information.

The next section describes how the features can be used to perform an incremental classification.

3.4 Incremental identification of the objects

We have seen that the extrinsic features are useful to improve the quality of the segmentation. In this section, a description of their use during the learning and delineation steps is provided.

3.4.1 Learning

Let us make the hypothesis that sequence of classification S and training set D are known. Sequence S gives the order in which the identification of objects has to be done, from the simplest to the hardest (this part is further developed in Section 3.5). Training set D is made up of feature values extracted for each superpixel from images that contain the object delineation. As all the delineations of objects are known, both intrinsic and extrinsic features are extracted. The training set is usually described by a matrix where the rows represent the superpixels and the columns, the features.

An object has to be recognised at each step of the sequence of classification. Binary classifiers are therefore learnt from the data to identify each object. Those binary classifiers work as one-versus-all classifiers, they try to find one

object against all the others among all the superpixels. Algorithm 1 outlines the different steps to learn the classifiers.

Algorithm 1

Require: S, D

1: $\mathcal{F}_{\text{kn}} = \mathcal{F}_{\text{int}}$ ▷ At the beginning, only the intrinsic feature can be known. Extrinsic features are progressively added at line 7.
2: $\mathcal{M}$ is an empty model.
3: Add S in $\mathcal{M}$
4: **for all** $o_\ell \in S$ **do** ▷ o_ℓ is the object to be identified.
5: $\mathcal{C}_\ell \leftarrow \text{Train}(D(\mathcal{F}_{\text{kn}}), o_\ell)$ ▷ $D(\mathcal{F}_{\text{kn}})$ is the training set D restricted to $\mathcal{F}_{\text{kn}}$. $\mathcal{C}_\ell$ is a binary classifier identifying superpixels of o_ℓ.
6: Add $\mathcal{C}_\ell$ in $\mathcal{M}$
7: Add extrinsic features related to o_ℓ in $\mathcal{F}_{\text{kn}}$
8: $\mathcal{C}'_\ell \leftarrow \text{Train}(D(\mathcal{F}_{\text{kn}}), o_\ell)$ ▷ Train a new classifier with the updated $\mathcal{F}_{\text{kn}}$.
9: Add $\mathcal{C}'_\ell$ in $\mathcal{M}$
10: **end for**
11: $\mathcal{C} \leftarrow \text{Train}(D)$ ▷ Train a multiclass classifier from the complete data set.
12: Add $\mathcal{C}$ in $\mathcal{M}$

It is noteworthy that two binary classifiers, called submodels, are learnt for each object. The first one only considers features available when the object has to be detected ($\mathcal{F}_{\text{kn}}$ at line 5). Afterwards, features related to object o_ℓ are added in $\mathcal{F}_{\text{kn}}$ (line 7). Therefore, the second submodel, learnt at line 8, takes into account extrinsic features related to the object to identify. This classifier is used during the delineation step to propagate the results obtained with the first classifier. For example, in the dataset composed of patient images, if only one superpixel of the lung is detected with the first classifier, we hope that the second classifier will identify, if it is relevant, the neighbouring superpixels as being a part of the lung. The identification is then propagated by repetitively using this second classifier.

Finally, a multiclass classifier is learnt at the very end of the learning (line 11). This terminal classifier is required to resolve possible conflicts. Indeed, some superpixels could have been identified as belonging to no or more than one object. The use of multiclass classifier to finish the delineation step ensures that each superpixel belongs to one, and only one, object. At late learning step, the model $\mathcal{M}$ contains the sequence S, two binary classifiers per object and one

multiclass classifier.

The method is generic, so it can be used with any kind of classifier as long as it is capable to deal with binary and multiclass classification problem. At each step, it is also possible to evaluate several classifiers and select the best one. In the presented results, the meta-parameters of SVM and RF are optimised at each step of the incremental classification.

The time required to learn the model can be long, depending on the used classifier. The optimisation of the meta-parameters of classifiers at each step increases even more the learning time. This should not be a problem in practice as the model is learnt beforehand only once.

3.4.2 Delineating

When delineating the object in a new image, the values of the extrinsic features are completely unknown at the beginning. The position of objects is indeed unresolved. Algorithm 2 outlines the different steps to use the model in order to delineate a new image. All objects are identified sequentially following the order provided by the model. Each object is identified several times to ensure that all the superpixels belonging to it have been found. Indeed, after its first identification (line 4), the object forces the extraction of extrinsic features about it (line 6). This information is then used to refine the identification (line 9). This step is repeated until reaching stability, when the label assigned to the superpixels does not change. To ensure the termination of the loop, a criterion fixing the number of repetitions can be added at line 8 of Algorithm 2.

It is noteworthy that each superpixel can be identified as belonging to several objects. Indeed, at each identification (lines 4 and 9), all superpixels are tested by the binary classifier.

Once all objects are known, the multiclass classifier is used over all data to provide the final membership of each superpixel. The result of multiclass classifier gives the final delineation of the image. Indeed, a mask defining an object can be created by grouping all superpixels belonging to it. If no error is made during the application of the incremental method, all the features are perfectly known and the final classifier provides the same result as the oracle classifier.

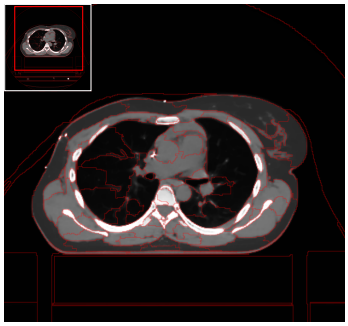
The time required to delineate the objects of a new image depends on the used classifier. When SVM or RF are used, most of the computations is performed during the learning. The identification of all superpixels only takes a few seconds.

Algorithm 2

Require: $\mathcal{M}$, D_t

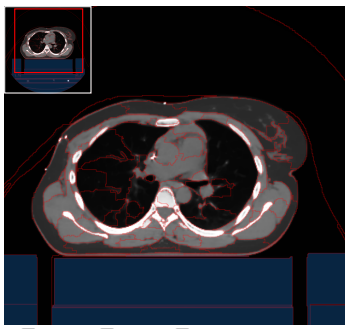
1: $\mathcal{F}_{\text{kn}} = \mathcal{F}_{\text{int}}$ ▷ At the beginning, only the intrinsic features can be known. Extrinsic features are progressively added at line 6.
2: $\mathcal{L} = \emptyset$ ▷ $\mathcal{L}$ is the set describing which superpixel belongs to which object.
3: **for all** $o_\ell \in \mathcal{M}(S)$ **do** ▷ o_ℓ is the object to be identified.
4: $\{s_i\} \leftarrow \mathcal{M}(D_t, \mathcal{F}_{\text{kn}}, o_\ell)$ ▷ The model $\mathcal{M}$ is used to identify superpixels belonging to o_ℓ knowing $\mathcal{F}_{\text{kn}}$.
5: For all superpixels in $\{s_i\}$, assign label o_ℓ in $\mathcal{L}$
6: Add extrinsic features related to o_ℓ in $\mathcal{F}_{\text{kn}}$
7: Compute values of the extrinsic features related to o_ℓ in D_t
8: **while** $\mathcal{L}$ is changing **do** ▷ Iterate until the labels do not change.
9: $\{s_i\} \leftarrow \mathcal{M}(D_t, \mathcal{F}_{\text{kn}}, o_\ell)$ ▷ The model $\mathcal{M}$ is used to identify superpixels belonging to o_ℓ knowing $\mathcal{F}_{\text{kn}}$.
10: Update $\mathcal{L}$ given the last classification
11: Update values of the extrinsic features related to o_ℓ in D_t
12: **end while**
13: **end for**
14: $\mathcal{L} = \mathcal{M}(D_t)$ ▷ Get the final label for each superpixel.

In the following pages, an example of incremental organ recognition is presented step by step with an image from the patient dataset.



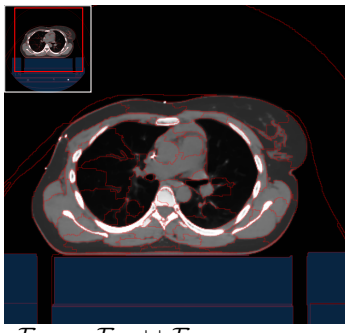
$\mathcal{F}_{\text{kn}} = \mathcal{F}_{\text{int}}$
$\#\mathcal{F}_{\text{kn}} = 11$
Next identification: o_1

At the beginning, only the intrinsic features are known, that is namely, $\mathcal{F}_{\text{kn}} = \mathcal{F}_{\text{int}}$. None of the superpixels has a label. Based on the sequence of classification present in the model, the first object to be identified is known. It is object o_1 corresponding to the table.



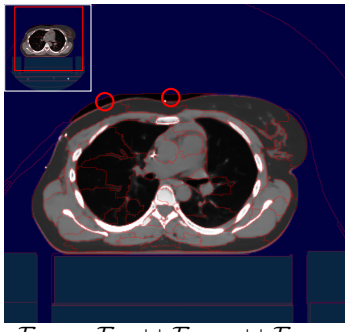
$\mathcal{F}_{\text{kn}} = \mathcal{F}_{\text{int}} \cup \mathcal{F}_{\text{ext}\,o_1}$
$\#\mathcal{F}_{\text{kn}} = 14$
Next identification: o_1

The table (object o_1) is identified based on the values of the 11 intrinsic features. After the identification, extrinsic features related to o_1 are extracted for all the superpixels, including those that already have a label. The set of known features $\mathcal{F}_{\text{kn}}$ is now composed of 14 features. The next step will be to identify object o_1 using the 14 features.



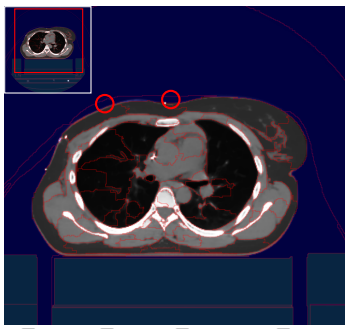
$\mathcal{F}_{\text{kn}} = \mathcal{F}_{\text{int}} \cup \mathcal{F}_{\text{ext}\,o_1}$
$\#\mathcal{F}_{\text{kn}} = 14$
Next identification: o_0

In this example, the second identification does not change the current segmentation. The identification of o_1 is therefore finished. By looking at the sequence of classification, we can determine that object o_0 will be identified in the following step.



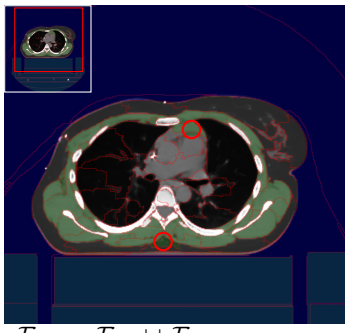
$\mathcal{F}_{\text{kn}} = \mathcal{F}_{\text{int}} \cup \mathcal{F}_{\text{ext}\,o_1} \cup \mathcal{F}_{\text{ext}\,o_0}$
$\#\mathcal{F}_{\text{kn}} = 17$
Next identification: o_0

The second object to be identified is the air (object o_0). Once the identification is done, the features related to o_0 can be extracted. Therefore, the total number of feature is 17. One superpixel (highlighted by the red circles) is not identified as being part of the air as it should be. As for the table, a second identification is performed based on the known features.



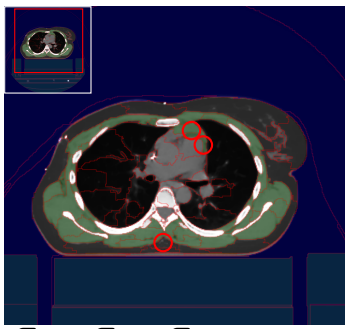
$\mathcal{F}_{\text{kn}} = \mathcal{F}_{\text{int}} \cup \mathcal{F}_{\text{ext}_{o_1}} \cup \mathcal{F}_{\text{ext}_{o_0}}$
$\#\mathcal{F}_{\text{kn}} = 17$
Next identification: o_4

The identification of the table with all the features allows to identify the missing superpixel (highlighted by the red circles). As the segmentation changed, the values of features $\mathcal{F}_{\text{ext}_{o_0}}$ are updated and the identification is repeated. Once the segmentation is stabilised, the next object (object o_4) can be identified with the 17 known features.



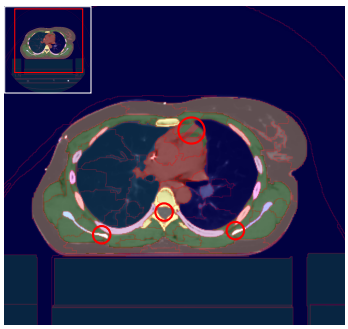
$\mathcal{F}_{\text{kn}} = \mathcal{F}_{\text{int}} \cup \mathcal{F}_{\text{ext}_{o_1,o_0,o_4}}$
$\#\mathcal{F}_{\text{kn}} = 20$
Next identification: o_4

The third object to be identified is muscle (object o_4). Extrinsic features related to o_4 are extracted. This increases the total number of features to 20. We can observe that some superpixels (highlighted by the red circles) are wrongly classified as being muscle.



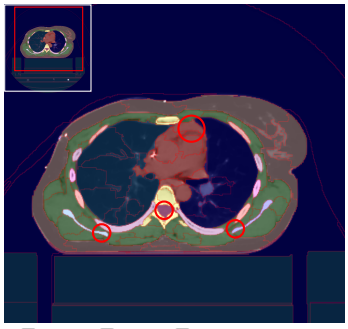
$\mathcal{F}_{\text{kn}} = \mathcal{F}_{\text{int}} \cup \mathcal{F}_{\text{ext}_{o_1,o_0,o_4}}$
$\#\mathcal{F}_{\text{kn}} = 20$
Next identification: o_2

Muscle is identified using the 20 features. The wrong identification at the bottom of the image is corrected during this step but the other errors propagate in the mediastinum. Steps involving identification of an object and extraction of related features are repeated until all objects are identified.



$\mathcal{F}_{\text{kn}} = \mathcal{F}_{\text{int}} \cup \mathcal{F}_{\text{ext}}$
$\#\mathcal{F}_{\text{kn}} = 80$
Next identification: all o_i

At the end of the incremental identification, all objects have been labeled in the images. Moreover, all features are extracted for each superpixel in the image. Nevertheless, some errors can still occur. Some superpixels have zero or several labels (highlighted by the red circles). The final multiclass identification, involving all features will give the final class label of each superpixel.



$\mathcal{F}_{\text{kn}} = \mathcal{F}_{\text{int}} \cup \mathcal{F}_{\text{ext}}$
$\#\mathcal{F}_{\text{kn}} = 80$

The multiclass classifier is used to give the final delineation of the image. The errors identified in the previous step get corrected in this final step.

In this example, we have observed that some errors can be corrected immediately in each step of the incremental procedure. Nevertheless, it sometimes happens that undetected errors propagate to the next steps.

3.4.3 Error propagation

If an error is made during the identification of an object, this error can be propagated among the other objects. Indeed, an erroneous identification of a superpixel leads to incorrect values during feature extraction. Those values will disturb the binary classifier used in the following steps of identification.

The error propagation phenomenon has been observed when using our method on the patient dataset. Figure 3.8 presents the result of a kernel density estimator for the distribution of the BCR with the RF classifier. The results are shown before and after the use of the multiclass classifier and are compared

to results with an oracle. By definition, the oracle classifier does not make any mistake when extracting features. It can be observed that the incremental method does not performed as well as the oracle. This is excursively because of the error made during the incremental process. Nevertheless, the use of the multiclass classifier tends to improve the BCR.

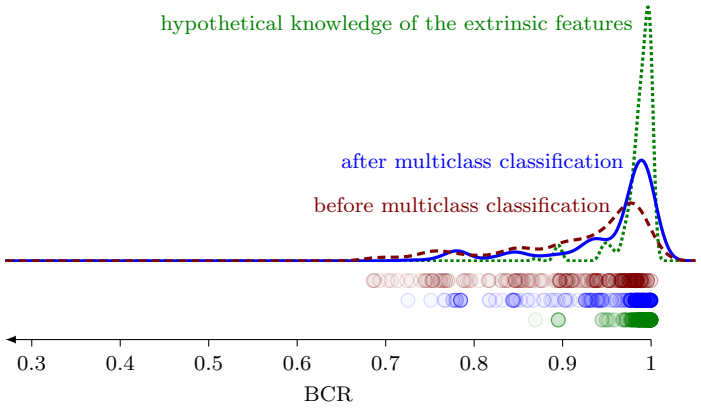


Figure 3.9 – Kernel density estimation (KDE) of the distribution of the BCR obtained individually with the RF classifier. The dashed red curve indicates the KDE before the use of the multiclass classifier. The plain blue curve shows the KDE of the final delineation, after the multiclass classification. The green dotted curve indicates the KDE when no error are made during the incremental process (oracle classifier).

The error propagation problem can be observed on Figure 3.10. Each entry of the confusion matrix is represented with a big pixel whose gray level actually corresponds to the fourth root of the considered value. This nonlinear transformation strongly increases the visual contrast among the really tiny off-diagonal entries. Doing so reveals some error propagation issues. It can happen that the serie of numbered ribs is shifted by one position. Such mistakes partly stem from ambiguous organ delineation in the training set. For instance, the last rib (7th) is sometimes split in two parts, one of them being contiguous to the vertebra. Figure 3.4 illustrates this phenomenon (see labels 13 and 18). The risk is then high that the piece of rib close to the vertebra gets merged with it. Similarly, the classification method can difficultly determine whether the 7th rib tag has to be assigned twice. Mistakes at this stage then propagate

to the other ribs and explain the shift.

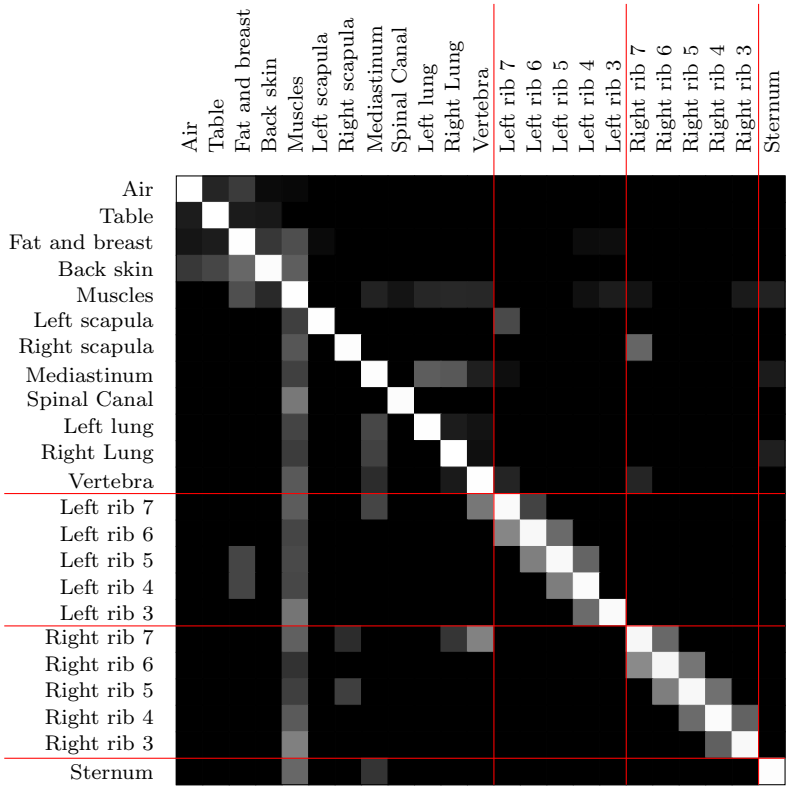


Figure 3.10 – Representation of the confusion matrix with the patient dataset using RF. Each row was normalized so that it adds up to one. To improve visual contrast among the off-diagonal entries, the fourth root was applied after normalisation. The series of numbered ribs are sometimes shifted by one position, the rib next to the vertebra being then merged with the latter.

The erroneous identification of superpixels makes the feature values inconsistent with the prior knowledge acquired from the training set, and thus degrades the quality of the delineation. Nevertheless, too many error propagation cause easy-to-see wrong segmentation. Image with lots of errors are easy to detect for a medical operator. We discuss the feasibility of an automatic error detection system as a perspective in Chapter 4.

3.5 Determination of the classification sequence

During the learning, a classification sequence S has to determine the order in which the objects are identified during the incremental process and, therefore, the order in which the extrinsic features are computed. Each object has to be presented once in the sequence. Each time an object is identified, there exists a risk of misclassification leading to wrong values of extrinsic features. The number of erroneous values should be minimised to ensure a good delineation.

Let us define $S^\star$, an optimal sequence that gives the minimum number of errors at the end of the incremental process. Such a sequence cannot be determined in practice. It would be indeed required to test all the possible sequences to be sure to find $S^\star$. The number of possible sequences in an image containing of C objects is equal to $C!$. For the dataset made up of images from patients ($C = 23$), this represents more than 10^{22} sequences, which is totally unaffordable in practice. The time complexity of solving this problem is $\mathcal{O}(M! \times g(D))$, where $g(D)$ is the time complexity of building a model and using it with the dataset D. In this section, two methods are proposed to approach $S^\star$. Those methods are also compared to random sequences to show their usefulness.

3.5.1 Greedy cross-validation

The first method tries to approach $S^\star$ with a greedy algorithm. Rather than minimise the final error, all the intermediate errors are minimised incrementally. The intuition is that, by minimising the error at each step, the final error should be minimised as well. At each step, the object that can be determined with the minimal error from the current set of known feature $\mathcal{F}_{\text{kn}}$ is identified. The sequence of classification S is obtained by going through Algorithm 3.

To evaluate the error committed with each object, a cross-validation is performed (lines 4 to 10). A cross-validation involves splitting the training set in Q independent subsets. A model is learnt from $Q - 1$ subsets and evaluated on the remaining one (lines 6 to 8). To perform a complete cross-validation, the evaluation is realised on the Q different subsets for each object. In the first step, all the C objects need to be tested by cross-validation. Each time an object is added to the sequence, it does not need to be tested again. The time complexity of this greedy method is $\mathcal{O}(Q \times C^2 \times g(D))$, where Q is the number of subsets tested in the cross-validation.

The obtained sequence S depends of the classifier used. Indeed, some objects

Algorithm 3

Require: S is an empty sequence.
1: $\mathcal{F}_{\text{kn}} = \mathcal{F}_{\text{int}}$
2: Separate dataset D into Q groups named G_i with $i \in 1, ..., Q$.
3: **while** all the objects are not in S **do**
4: **for all** objects o_c not present in S **do**
5: **for all** groups G_i **do**
6: Use $D \setminus G_i$ as a training set.
7: Build a binary classifier that identifies o_c.
8: Measure the classifier performance on the validation set G_i.
9: **end for**
10: **end for**
11: Compute $o_\ell = \max_{o_c} p_{o_c}$, where p_{o_c} is the mean performance for the binary classifier identifying the object o_c.
12: Add o_ℓ into S.
13: Add the features related to o_ℓ in $\mathcal{F}_{\text{kn}}$.
14: **end while**

are more easily identified by some classifiers than others.

This method has the advantage to use a greedy algorithm to approach the best solution. However, the time complexity remains high and can make this method unusable when the images contain a lot of objects.

3.5.2 Direct nearest neighbours

Unlike 'greedy cross-validation', 'direct nearest neighbours' does not involve a classifier. Indeed, the sequence is directly established from the observed values of the features. The usefulness of a feature to identify a given object depends on its marginal distribution. Given a single feature, if an object does not significantly overlap the others, this feature may be considered as useful to identify that object. This method computes a ranking that assesses the overlap of objects if all superpixels were characterised by only one feature.

Let $\mathcal{N}_f^K(s_i)$ denote the set of the K nearest neighbours of superpixels s_i in the subspace of feature f alone (f can be an intrinsic or extrinsic feature) of a dataset D. Let $\mathcal{S}_{o_c}$ be the set of pieces belonging to object o_c. The usefulness of a certain feature to classify a superpixel inside or outside the object o_c can be measured as the proportion of superpixels belonging to object o_c among the K nearest neighbours (regarding only the considered feature) of each superpixel

of the object o_c. It can be written as

$$u_{fc} = \frac{1}{K|\mathcal{S}_{o_c}|} \sum_{s_i \in \mathcal{S}_{o_c}} \left| \left\{ s_j \text{ s.t. } s_j \in \mathcal{N}_f^K(s_i) \text{ and } s_j \text{ belongs to } o_c \right\} \right| ,$$

where $|A|$ denotes the cardinality of set A. The value of u_{fc} can range from 0 to 1. The value u_{fc} indicates how much object o_c stands apart from other objects along the axis of feature f. The bigger u_{fc}, the more feature f can discriminate object o_c. The value of u_{fc} can be computed efficiently (see Bernard et al. (2014b)).

Once all the values u_{fc} are computed, the sequence of classification is established. The object that can be discriminated the most, is added in the sequence. This can be done by going through Algorithm 4.

Algorithm 4

Require: $\mathcal{F}_{\text{kn}} = \mathcal{F}_{\text{int}}$

1: **while** all the objects are not in S **do**
2: $o_\ell \leftarrow \arg\max_c u_{fc}$ with $f \in \mathcal{F}_{\text{kn}}$
3: $u_{\bullet\ell} \leftarrow 0$
4: Add o_ℓ into S
5: Add the features related to o_ℓ in $\mathcal{F}_{\text{kn}}$.
6: **end while**

'Direct nearest neighbours' identifies the most distinguishable object along each feature and takes the best one knowing the current $\mathcal{F}_{\text{kn}}$. The time complexity of the determination of the sequence is $\mathcal{O}(MP^2K\ln(P)\ln(K))$, where M is the number of features, P the number of superpixel and K the number of neighbours tested in the method (see Bernard et al. (2014b) for more details). This is rather low compared to 'greedy cross-validation' as it does not require the use of a classifier. However, the only use of marginal distribution does not take into account the possibility of combinations between features. By doing so, it can miss an easy-to-identify object.

3.5.3 Validation of the sequence methods

To validate the two methods determining the sequence of classification, these one are compared to random sequences and the non incremental methods, oracle and blind.

The first comparison concerns the BCR at the end of the incremental

process. The sequence of classification is determined and the superpixels are identified following the method described at Section 3.4. For 'greedy cross-validation', the same classification method is used to establish the sequence and identify the superpixels. As it is shown in Table 3.6, there exists no significant difference between random sequences and our methods ('direct nearest neighbours' and 'greedy cross-validation'). Our hypothesis is that there are several sequences that may perform nearly as well as $S^\star$. When selecting randomly the sequence of classification, the obtained sequence is sometimes as good as $S^\star$, and sometimes terribly wrong. This creates a high variability in the quality of results produced with the incremental classification using random sequence. The different incremental methods are therefore difficult to differentiate from the random method. Nevertheless, despite this non significant difference, the mean BCR obtained with the two proposed methods remains higher than the random sequences and suffers from less variability for both 'direct nearest neighbours' and 'greedy cross-validation'. It is noteworthy that each of the incremental methods with RF is significantly better than the blind one. Furthermore, they are not significantly different from the 'oracle'.

As we work with images, the results can also be visually compared. In Figure 3.11, the blind classification with RF is compared with the incremental classification with greedy cross-validation and RF on 3 images. The blind method makes a lot of mistakes of various types. The spinal canal is often misclassified as part of the mediastinum or muscle. As blind classification does not use features like the relative position, confusion between the scapula and the ribs is unavoidable. In one case, part of the mediastinum is labelled as being the sternum. Moreover, ribs are often misclassified. Such mistakes are much less frequent with incremental classification.

In addition to the analysis of the BCR, the stability of the sequences can be studied. A stable sequence facilitates its interpretation and its use in the future. To estimate the stability, the Levenshtein distance between sequences is measured (see Section 3.1.4). The Levenshtein distance determines the number of requested modifications (insertions, deletions or substitutions) to pass from one sequence to another. The bigger the average distance, the less stable the method is. Table 3.7 reports the mean Levenshtein distances between the different sequences obtained with one method on several runs. It is noteworthy that, compared to the random sequence, the two proposed methods, 'direct nearest neighbours' and 'greedy cross-validation', reduce the distance and therefore, improve the stability. Moreover, 'direct nearest neighbours' is

Classifier	Classification technique	BCR (%)
RF	Random sequences	93.18 ± 8.64^{a}
	Direct nearest neighbours	93.41 ± 8.49^{a}
	Greedy cross-validation	95.22 ± 6.40^{a}
	Blind (non-incremental)	84.15 ± 8.87^{b}
	Oracle (non-incremental)	98.56 ± 2.09^{a}
SVM	Random sequences	78.55 ± 19.54^{c}
	Direct nearest neighbours	85.03 ± 12.37^{c}
	Greedy cross-validation	83.98 ± 16.01^{c}
	Blind (non-incremental)	85.17 ± 9.37^{c}
	Oracle (non-incremental)	98.49 ± 1.82^{d}

Table 3.6 – BCR of the delineation with the patient dataset. Incremental delineation is performed with two classification methods (RF and SVM). Three methods, 'random sequences', 'direct nearest neighbours´, and 'greedy cross-validation', are used to construct the sequence of classification. Non-incremental methods, 'blind', and 'oracle', are also used. 'Blind' performs a multiclass classification based on the intrinsic features. 'Oracle' performs a multiclass classification based on the hypothetical knowledge of the intrinsic and extrinsic features. To established the differences between methods with the same classifier, a modified t-test is used as suggest by Nadeau and Bengio (2003). For a given classifier, values of BCR with different letters are significantly different with a confident level of 0.95 (p-value $< 4.10^{-2}$).

slightly more stable than 'greedy cross-validation' as it only depends on the training set. Nevertheless, the mean Levenshtein distances remain high, more than nine modifications on average, which tends to prove that there exist several sequences closed to $S^{\star}$.

Sequence	Classifier	Levenshtein Distance
Random sequences	—	21.16 ± 1.18
Direct nearest neighbours	—	9.21 ± 1.84
Greedy cross-validation	RF	12.81 ± 2.32
Greedy cross-validation	SVM	15.17 ± 2.71

Table 3.7 – Measures of the Levenshtein distance for each sequence method. The measures are realised on 50 sequences computed from the patient dataset. 'Greedy cross-validation' is evaluated with two different classifiers. The other methods do not require the use of a classifier.

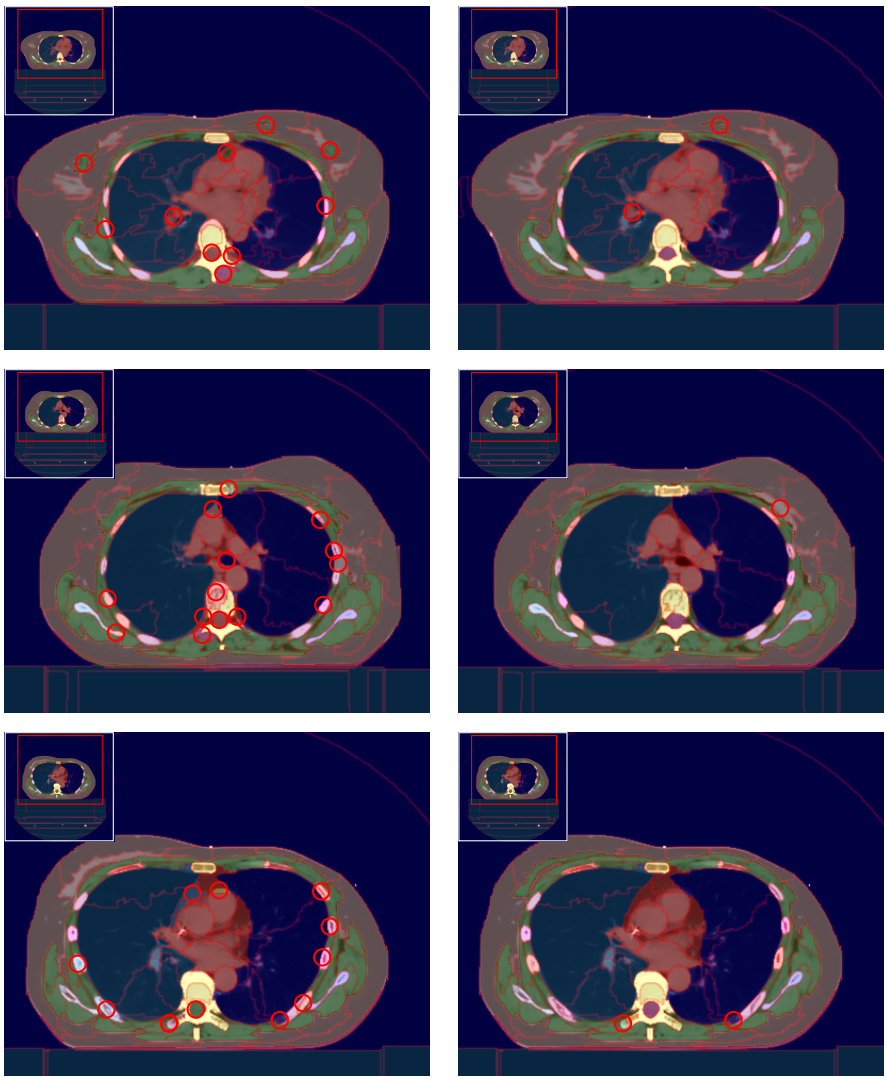


Figure 3.11 – Left: blind segmentations with RF – Right: incremental segmentations, sequencing by cross-validation with RF. Red circles highlight the mistakes. Blind classification makes a lot of various mistakes.

Figure 3.12 shows the map of dependencies between objects with the different methods and classifiers. Patterns can be observed in the sequence of classification.

For example, with ‘direct nearest neighbours’, the ribs are most of the time identifed at the end of the sequence. Nevertheless, there is no strict order for the identification of the ribs with that method. The same phenomenon can be observed for the scapulae, which are classified one after the other but not always in the same order. All those weak dependencies between objects decrease the stability of the sequences.

The determination of the sequence of classification is completely automatic and derived from the training set. Prior knowledge acquired from the training set is used to establish a sequence. Depending on the training set, several sequences can lead to equivalently good delineation. Nevertheless, it is preferable to guide the construction of the sequence by simple heuristics in order to avoid inefficient sequences.

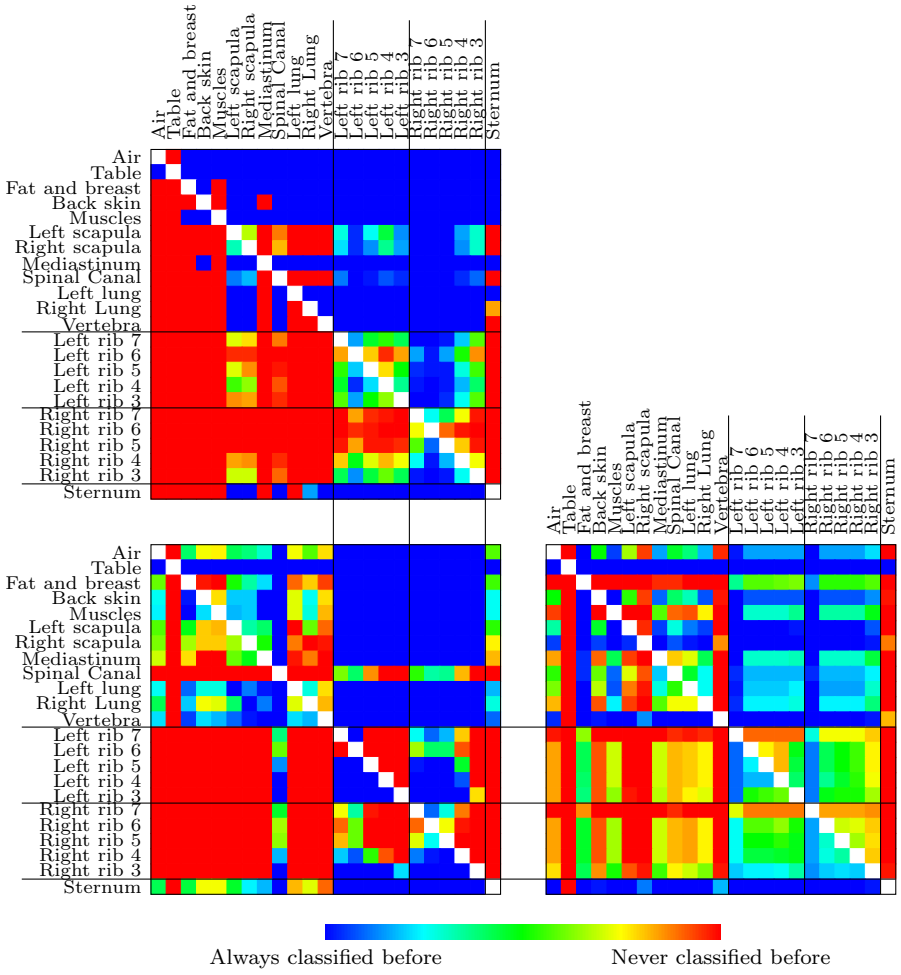


Figure 3.12 – Patients dataset. Upper left: Sequencing by direct nearest neighbours (same results with all the classifiers). Lower left: Sequencing by greedy cross-validation with RF. Lower right: Sequencing by cross-validation with SVM. Representation of the dependencies between the different objects. The number of times an object was identified after another was counted over the 50 runs. An object often classified after another is considered as being directly or indirectly dependent on that object.

3.6 Parameters of the method

Even if the method is automatic, some parameters and meta-parameters need to be adjusted by the user. Some of those parameter values are already determined automatically, some others can be learnt from the training set, and a few ones cannot be learnt. This section lists the parameters and discusses whether they can be learnt or not.

Number of superpixels. This number is currently fixed manually by the user. In the future, this number may be learnt from the training set. Finding the optimal number of superpixels can be seen as an optimisation problem where the number of superpixels must be minimized under the constraint that each super pixel belongs to only one organ. As manual delineation is not always accurate, the constraint can be modified so that each superpixel contains a given ratio of pixels from the same organ.

Contrast enhancement. It consists of a piecewise linear transformation of the gray levels. Its values are currently set manually. In the future, the parameters of the contrast enhancement may be learnt from the training set. Indeed, the areas that require some contrast enhancement can be identified by looking at the neighbourhood of the boundaries of the object.

Feature definition. Features are currently defined by the user. In practice, they can be validated by using the oracle method on the training set. The set of features is specific to the domain of application. Nevertheless, in the future, a large number of features (intrinsic and extrinsic) could be extracted. Next, feature selection techniques can be automatically used at each classification step in order to keep only the most important ones.

Classifier. The type of classifier is chosen by the user. In the future, the best classifier may be selected among several alternatives by using cross-validation. The best classifier can be selected at each step of the iterative process. This would lead to a solution where several types of classifier are used during the delineation process.

Meta-parameters of the classifier. Those meta-parameters are currently determined by a cross-validation technique each time that a classifier is built.

Classification sequence. The classification sequence is determined automatically from the training set.

Method used to establish the classification sequence. Two methods determining the sequence of classification have been proposed. The user need to select one of them. In the future, the best method may be selected by cross-validation over the whole process. By doing so, the best method is always chosen to build the final model.

Cross-validation parameters. When a cross-validation technique is used, it requires to adjust the number of folds to use. This number has to be set manually. Too many folds will reduce the number of images used for the evaluation. Too few folds will reduce the number of images used to built the model to evaluate. Usually, the number of folds is set between 5 and 10.

Our method still contains some parameters that required to be manually fixed. Nevertheless, most of those parameters can be evaluated and selected by a cross-validation technique. By using cross-validation, the computation time required for the construction of the model will increase but it will not impact the time required to perform the delineation.

3.7 Conclusion

In this chapter, we proposed an automatic method whose main characteristic is to involve a training set, which is used by the model to learn prior knowledge. Rich information is extracted from the images to make the prior knowledge as useful as possible. Only a very small amount of prior knowledge is directly hard-coded in the method. In other words, our method makes very few assumptions about the images and their content. Thanks to this limited quantity of static prior knowledge, the method is designed to be generic and easily adaptable to any part of the human body, just by changing the training set. The proposed method can be decomposed into four different modules.

- The first module segments the image in homogeneous areas (superpixels). In our case, this module relies on a specific watershed algorithm.
- The second module corresponds to the extraction of the features for each superpixel. The current version of the module can only consider features requiring the knowledge of no or one object. In the future, this module may be able to extract features depending on more than one object at a time.
- The third module establishes the sequence in which the objects have to be identified. We already suggest two options for this module, both of them working with no other information than that contained in the training set.
- Finally, the fourth module is dedicated to the learning of the identification and its use. It learns the submodels necessary to the organ delineations. Those submodels rely on well studied classification techniques inherited from machine learning.

Each of those modules tries to mimic the behaviour of the physicians. The modules can be changed and adapted separately with no repercussion on the others. For example, the watershed algorithm could be replaced with any other method providing a segmentation of the image in homogeneous areas. The modification of the segmentation method will not trigger any change in the other modules. This makes the method easy to improve as it does not require to know every module of the process to optimise the others.

Superpixels

Images (especially 3D ones) are voluminous and take much memory space. Segmenting the images in superpixels and keeping only a limited set of features

reduce substantially memory assuption. Only relevant information for the segmentation is kept. The gain in space allows working with dataset containing a lot of images.

Currently, the objects are represented by a union of several superpixels. Due to the use of superpixels, the boundaries of the organs can sometimes look a bit rough. Adequate post-processing of the results can smooth the boundaries and improve the visual quality of delineation. Straightforward post-processing, using for example mathematical morphology, can also generate margins, e.g. around target volumes, in order to get geometrical, non anatomical volumes (CTV and PTV), useful in radiotherapy.

One of the advantage of using superpixels is the ability to easily correct some delineation mistakes. Indeed, if it is wrong, the membership of a group of pixels to an organ can easily be changed. In the case where a superpixel is poorly defined (e.g. when it contains pixels from two different objects), it can be refined and subdivided by locally applying the watershed algorithm. This creates smaller superpixels that can be labelled independently.

Learning

Learning of the model is usually long. Indeed, the best sequence of classification needs to be evaluated and all the submodels, binary and multiclass classifiers, have to be learnt. Nevertheless, in practice, those operations can be carried out beforehand. When a query image needs to be delineated, the incremental method is fast. As a matter of fact, most of the time is spent segmenting the image with the watershed algorithm (less than two minutes for a 3D CT scan). By comparison, atlases and statistical models require much more computation while delineating a new image. With atlases, computations occur mainly in the registration step with the query image. For the statistical model, the search of the shape in the query image can require more or less time, depending on initialisation.

The main drawback of the method is the requirement of a training set containing images where all pixels are tagged with an organ label. In contrast, atlases and statistical models can use images where only a few pixels are labelled. The complete delineation of all organs and objects in an image takes hours of work. Therefore, the construction of training sets with enough images to be representative of the global population is rather expensive. A possible solution for this issue is discussed as a perspective in Chapter 4.

Robustness

By construction, the model is strongly related to the training set. Indeed, almost all parameters are automatically learnt from it. The model is therefore specific to the patient population depicted in the training set. If this population is very particular (same sex, same age, same origin, etc.), the model will be very specific and will not be able to generalise to other populations. Increasing the size and the variability of the training set can lead to a more robust model. Similarly, if the model is specific to images acquired from a single imaging device, it may difficultly be applied to other images. In this case, some pre-processing of the images can somehow normalise them.

As the model is directly derived from the training set, its performance can be degraded by the potential errors present in the training set. Physicians are humans, and they sometimes make mistakes. The errors can be of two types, systematic and random. Moreover, they can be specific to a physician or a population of physicians. In the case of a physician, the systematic errors correspond to errors that are systematically repeated when delineating one image several times. It can be caused by the tools used for the delineation or erroneous knowledge about the area delineated. The random errors correspond to errors that vary if the physician segments several times the same image. For each segmentation, he should not repeat the same mistakes. Those errors are usually caused by a lack of concentration. Some of those mistakes can be avoided by using superpixels, which place consistently the boundaries of the object on peaks of the gradient magnitude. Nevertheless, some errors may remain. Increasing the size of the training set may reduce the effect of the random errors on the model. Indeed, if the number of images is big enough, some random errors can be detected when learning the model. Those outliers are therefore not taken into account later. Systematic errors are more problematic as they cannot be detected and can be learnt by the model as a property of the training set. A solution to reduce the effect of the systematic error realised by a physician is to build a training set from a population of physicians. Indeed, part of the individual systematic errors can be seen as random errors for the population. Nevertheless, it may remain some systematic errors for the population. This somehow corresponds to misconceptions in the entire population. Unfortunately, such errors cannot be detected and overcome.

Chapter 4

Summary and perspectives

Radiation therapy has significantly improved in recent decades. The advent of new treatment methods and the creation of more accurate imaging techniques offer the possibility to deliver better treatments. Nevertheless, progress in treatment delivery techniques makes the accuracy of delineation more and more critical. Organ delineation requiring considerable medical expertise is indeed difficult to formalise and thus to automate. Moreover, variability of manual delineation is still significant despite the use of guidelines. Time required to perform the delineation of the target volumes (TVs) and organs at risk (OARs) remains long and repetitive. Most OARs and other healthy tissues are part of an anatomy that is very regular across patients. In this context, automation of OARs delineation could give the physicians more time to focus on TVs, which are patient- and/or disease-specific.

Nowadays, the atlas and statistical models are the most recent techniques to perform automatic delineation. Nevertheless, they suffer from slow adoption by the physicians. On one hand, the atlases remain difficult to parametrise and regularise. They are technical and often limited to precise locations. On the other hand, statistical models are only able to learn and delineate one organ at a time. Moreover, the initialisation of the search for the shape is difficult to automate.

The objective of this work was to propose an alternative to those existing methods. In particular, we suggest an organ delineation method based on machine learning techniques. This method attempts to reproduce the way physicians delineate organs manually, starting with those that are straightforward to identify and then going progressively to more difficult ones. Compared to the

other segmentation methods, our proposal is able to extract rich information during the delineation process. All information required to learn the organ delineation is acquired from images where organs are delineated by physicians.

In its current version, the method we propose requires that all objects are delineated in the image. This is an important limitation, as the construction of a big set of images to train the method is very expensive. At the same time, such a complete data set also conveys potentially much information, i.e. prior knowledge that is often not present in atlases. One might then rightfully wonder whether the investment is worth the while and gives our method some advantages over atlases. If not, it will be necessary to work with images that are not entirely segmented, in order to remain competitive with atlases. This would then call for the development of methods that automatically discover intermediate objects and/or organs for the delineation. Those methods can use the images provided by the physicians to learn useful intermediate objects. Those objects are not required to have an anatomical sense as long as the objects of interest, required by the physicians, are well delineated.

In Chapter 3, we propose to enhance the contrast in the image by converting the pixel intensities in a nonlinear way. The suggested technique is quite simple and completely independent of our incremental delineation process and can be used in other image segmentation methods. It increases the contrast in relevant areas of the image (mainly soft tissues in our case). The use of this technique with the atlases and statistical models might improve and accelerate the delineation. Indeed, both categories of methods are driven by the pixel intensities. In the cases of atlases, the increase of the contrast in areas of interest will give them more importance in the registration metric, leading hopefully to more accurate results. For the statistical models, the boundaries of the organ are easier to find when contrast is enhanced. In its current version, the contrast enhancement requires human interaction to define the transformation function. In the future, this function may be learnt from already delineated images. Indeed, the area that required a contrast enhancement can be identified by looking at the neighbourhood of the boundaries of the object.

Concerning organ classification, we have also observed that if no error is made during the incremental procedure, our method would give a nearly perfect delineation. Information extracted from the image is therefore sufficient to delineate all objects in an image. Nevertheless, some errors can occur during the identification process. If those errors are made at the beginning of the incremental identification, they can propagate in the following steps and have a

strong impact on the final result. We think that some of those errors can be detected before their propagation, by performing some simple checks on the images. For example, in our set of chest images, if two ribs are touching each other in the delineation process, some issue can be expected. Potential errors can also be discovered by doing some sanity checks on the used classifier. Most classifiers provide a score associated with the prediction. If the score is low when identifying a superpixel, the risk of error is high. The integration of a system detecting the potential errors would be of great help to improve the delineation. This system could even be coupled with the supervision by a physician. When a potential error is detected, the physician is requested to judge the case. By stopping the errors before their propagation in the incremental delineation, a better delineation should be achieved. Nevertheless, it is important to limit as much as possible the amount of interaction with the physicians. Physicians can introduce errors in the delineation process and those errors can be even more difficult to correct. In the future, more advanced methods may be built to detect and automatically correct the potential errors.

In its current version, the method is developed to work alone. In practice, it can also be combined with atlases or statistical models. The atlases can be used as an initialisation step for the method. This would provide the positions of some organs and extrinsic information can be extracted at the very beginning of the incremental process. At the end, the statistical models can use the segmentation obtained with our approach to search for the objects in the image. The combination of the different methods, may lead to more robust methods.

In this thesis, the incremental method has been tested on images of the chest. On those images, the breast tumour has been removed surgically. This property allows us to better evaluate the potential of our method. Indeed, in other cancerous diseases, the tumour is still present during radiation therapy. It introduces an additional object in the image whose properties (location, shape) are highly unpredictable. It can also deform the patient body and somehow reduce the capability of the iterative segmentation. In the future, it will be important to be able to detect automatically the position of the tumour. By using this information, as a new feature or combined with an adaptation of the existing features, our method may be able to deal with more complex images.

Medical images sometimes contain artefacts. In a real-life application, their detection in the image will be required in order to improve the method robustness. Indeed, artefacts deteriorate the image and can have an important effect on the superpixels obtained by the watershed algorithm. In practice, artefacts

usually have recognisable forms and tend to appear often at the same part of the body. Their detection should not be too difficult. Nevertheless, interpolating the objects through the artefact remains a challenging problem.

In this thesis, we have proposed a method mainly based on information extracted from computed tomography (CT) images. This approach is applicable to other imaging modalities. Yet, since CT images are always acquired to plan the treatment and compute the dose distribution, we suggest to combine several modalities instead of replacing CT with another. The combination of CT and positron emission tomography (PET) can be used to automatically detect the position of the tumour in the image by adding, for example, information about the pixel intensities in the PET image for each superpixels. Providing a higher contrast in soft tissues than CT, magnetic resonance imaging (MRI) can be used jointly with CT to improve the segmentation with the watershed algorithm and to facilitate the differentiation of organs. Whatever the source, any relevant information can help the incremental segmentation.

To conclude, we have proposed a novel approach that automatically adapts its parameters by learning prior knowledge from images. The modularity of the method opens the way to many opportunities of additional developments.

Author's contributions

Bernard, Guillaume, Michel Verleysen and John A Lee (2012). 'Incremental feature computation and classification for image segmentation'. In: *20th International Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2012)*, pp. 157–162.

— (2013). 'Segmentation with Incremental Classifiers'. In: *Image Analysis and Processing–ICIAP 2013*. Springer, pp. 81–90.

— (2014a). 'Automatic Organ at Risk Delineation with Machine Learning Techniques'. In: vol. 41. 6. American Association of Physicists in Medicine, pp. 101–101.

— (2014b). 'Incremental classification of objects in scenes: application to the delineation of images'. In: *Neurocomputing*. Forthcoming.

— (2014c). 'Organ delineation with watersheds and machine learning'. In: *29th Belgian Hospital Physicists Association Annual Meeting (BHPA 2014)*.

Lee, John A, Emilie Renard, Guillaume Bernard, Pierre Dupont and Michel Verleysen (2013). 'Type 1 and 2 mixtures of Kullback–Leibler divergences as cost functions in dimensionality reduction based on similarity preservation'. In: *Neurocomputing* 112, pp. 92–108.

Bibliography

Adams, Rolf and Leanne Bischof (1994). 'Seeded region growing'. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 16.6, pp. 641–647.

Aljabar, Paul, Rolf A Heckemann, Alexander Hammers, Joseph V Hajnal and Daniel Rueckert (2009). 'Multi-atlas based segmentation of brain images: atlas selection and its effect on accuracy'. In: *Neuroimage* 46.3, pp. 726–738.

Allène, Cédric, Jean-Yves Audibert, Michel Couprie, Jean Cousty, Renaud Keriven et al. (2007). 'Some links between min-cuts, optimal spanning forests and watersheds'. In: *Mathematical Morphology and its Applications to Image and Signal Processing*, pp. 253–264.

Bay, Herbert, Tinne Tuytelaars and Luc Van Gool (2006). 'Surf: Speeded up robust features'. In: *Computer Vision–ECCV 2006*. Springer, pp. 404–417.

Beauchemin, Steven S. and John L. Barron (1995). 'The computation of optical flow'. In: *ACM Computing Surveys (CSUR)* 27.3, pp. 433–466.

Bernard, Guillaume, Michel Verleysen and John A Lee (2012). 'Incremental feature computation and classification for image segmentation'. In: *20th International Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2012)*, pp. 157–162.

— (2013). 'Segmentation with Incremental Classifiers'. In: *Image Analysis and Processing–ICIAP 2013*. Springer, pp. 81–90.

— (2014a). 'Automatic Organ at Risk Delineation with Machine Learning Techniques'. In: vol. 41. 6. American Association of Physicists in Medicine, pp. 101–101.

— (2014b). 'Incremental classification of objects in scenes: application to the delineation of images'. In: *Neurocomputing*. Forthcoming.

— (2014c). 'Organ delineation with watersheds and machine learning'. In: *29th Belgian Hospital Physicists Association Annual Meeting (BHPA 2014)*.

Besl, Paul J and Neil D McKay (1992). 'Method for registration of 3-D shapes'. In: *Robotics-DL tentative*. International Society for Optics and Photonics, pp. 586–606.

Beucher, Serge and Christian Lantuéjoul (1979). 'Use of watersheds in contour detection'. In:

Beucher, Serge and Fernand Meyer (1992). 'The morphological approach to segmentation: the watershed transformation'. In: *OPTICAL ENGINEERING-NEW YORK-MARCEL DEKKER INCORPORATED-* 34, pp. 433–433.

Bleau, André and L Joshua Leon (2000). 'Watershed-based segmentation and region merging'. In: *Computer Vision and Image Understanding* 77.3, pp. 317–370.

Blezek, Daniel J and James V Miller (2007). 'Atlas stratification'. In: *Medical Image Analysis* 11.5, pp. 443–457.

Bloch, Isabelle (1999). 'Fuzzy relative position between objects in image processing: a morphological approach'. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 21.7, pp. 657–664.

Boser, Bernhard E, Isabelle M Guyon and Vladimir N Vapnik (1992). 'A training algorithm for optimal margin classifiers'. In: *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, pp. 144–152.

Boykov, Yuri and Gareth Funka-Lea (2006). 'Graph cuts and efficient ND image segmentation'. In: *International Journal of Computer Vision* 70.2, pp. 109–131.

Boykov, Yuri and Vladimir Kolmogorov (2004). 'An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision'. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 26.9, pp. 1124–1137.

Breiman, Leo (2001). 'Random forests'. In: *Machine learning* 45.1, pp. 5–32.

Breiman, Leo, Jerome Friedman, Charles J Stone and Richard A Olshen (1984). *Classification and regression trees*. CRC press.

Brighton, Henry and Chris Mellish (2002). 'Advances in instance selection for instance-based learning algorithms'. In: *Data mining and knowledge discovery* 6.2, pp. 153–172.

Brown, Martin, Hugh G Lewis and Steve R Gunn (2000). 'Linear spectral mixture models and support vector machines for remote sensing'. In: *Geoscience and Remote Sensing, IEEE Transactions on* 38.5, pp. 2346–2360.

Caldwell, Curtis B, Katherine Mah, Yee C Ung, Cyril E Danjoux, Judith M Balogh, S.Nimu Ganguli and Lisa E Ehrlich (2001). 'Observer variation in

contouring gross tumor volume in patients with poorly defined non-small-cell lung tumors on CT: the impact of 18FDG-hybrid PET fusion'. In: *International Journal of Radiation Oncology*Biology*Physics* 51.4, pp. 923–931. ISSN: 0360-3016.

Cazzaniga, Luigi Franco, Maria Antonella Marinoni, Alberto Bossi, Ernestina Bianchi, Emanuela Cagna, Dorian Cosentino, Luciano Scandolaro, Marica Valli and Milena Frigerio (1998). 'Interphysician variability in defining the planning target volume in the irradiation of prostate and seminal vesicles'. In: *Radiotherapy and Oncology* 47.3, pp. 293–296. ISSN: 0167-8140.

Chambolle, Antonin (2004). 'An algorithm for total variation minimization and applications'. In: *Journal of Mathematical imaging and vision* 20.1-2, pp. 89–97.

Commowick, Olivier and Grégoire Malandain (2007). 'Efficient selection of the most similar image in a database for critical structures segmentation'. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2007*. Springer, pp. 203–210.

Commowick, Olivier, Simon K Warfield and Grégoire Malandain (2009). 'Using Frankenstein's creature paradigm to build a patient specific atlas'. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009*. Springer, pp. 993–1000.

Cootes, Timothy F, Gareth J Edwards and Christopher J Taylor (1998). 'Active appearance models'. In: *Computer Vision—ECCV'98*. Springer, pp. 484–498.

— (2001). 'Active appearance models'. In: *IEEE Transactions on pattern analysis and machine intelligence* 23.6, pp. 681–685.

Cootes, Timothy F, Christopher J Taylor, David H Cooper and Jim Graham (1995). 'Active shape models-their training and application'. In: *Computer vision and image understanding* 61.1, pp. 38–59.

Cortes, Corinna and Vladimir Vapnik (1995). 'Support-vector networks'. In: *Machine learning* 20.3, pp. 273–297.

Cousty, Jean, Gilles Bertrand, Laurent Najman and Michel Couprie (2009). 'Watershed cuts: Minimum spanning forests and the drop of water principle'. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31.8, pp. 1362–1374.

Cover, Thomas and Peter Hart (1967). 'Nearest neighbor pattern classification'. In: *Information Theory, IEEE Transactions on* 13.1, pp. 21–27.

Criminisi, Antonio, Jamie Shotton and Stefano Bucciarelli (2009). 'Decision forests with long-range spatial context for organ localization in CT volumes'. In: *Proc MICCAI PMMIA*, pp. 69–80.

Dalal, Navneet and Bill Triggs (2005). 'Histograms of oriented gradients for human detection'. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on.* Vol. 1. IEEE, pp. 886–893.

Davies, Rhodri H, Carole J Twining, Tim F Cootes, John C Waterton and Christopher J Taylor (2002). '3D statistical shape models using direct optimisation of description length'. In: *Computer Vision—ECCV 2002.* Springer, pp. 3–20.

Dawant, Benoit M, Steven L Hartmann, Jean-Philippe Thirion, Frederik Maes, Dirk Vandermeulen and Philippe Demaerel (1999). 'Automatic 3-D segmentation of internal structures of the head in MR images using a combination of similarity and free-form transformations. I. Methodology and validation on normal subjects'. In: *Medical Imaging, IEEE Transactions on* 18.10, pp. 909–916.

Delaney, Geoff, Susannah Jacob, Carolyn Featherstone and Michael Barton (2005). 'The role of radiotherapy in cancer treatment'. In: *Cancer* 104.6, pp. 1129–1137.

Donner, Rene, Michael Reiter, Georg Langs, Philipp Peloschek and Horst Bischof (2006). 'Fast Active Appearance Model Search Using Canonical Correlation Analysis'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.10, pp. 1690–1694.

Enzweiler, Markus and Dariu M Gavrila (2009). 'Monocular pedestrian detection: Survey and experiments'. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31.12, pp. 2179–2195.

Felzenszwalb, Pedro F and Daniel P Huttenlocher (2004). 'Efficient graph-based image segmentation'. In: *International Journal of Computer Vision* 59.2, pp. 167–181.

Ferrant, Matthieu, Benoit Macq, Arya Nabavi and Simon K Warfield (2000). 'Deformable modeling for characterizing biomedical shape changes'. In: *Discrete Geometry for Computer Imagery.* Springer, pp. 235–248.

Fornefett, Mike, Karl Rohr and H Siegfried Stiehl (2001). 'Radial basis functions with compact support for elastic registration of medical images'. In: *Image and Vision Computing* 19.1, pp. 87–96.

Fortunati, Valerio, René F Verhaart, Fedde van der Lijn, Wiro J Niessen, Jifke F Veenland, Margarethus M Paulides and Theo van Walsum (2013). 'Tissue segmentation of head and neck CT images for treatment planning: A multiatlas approach combined with intensity modeling'. In: *Medical physics* 40.7, p. 071905.

Frangi, Alejandro F, Wiro J Niessen, Daniel Rueckert and Julia A Schnabel (2001). 'Automatic 3D ASM construction via atlas-based landmarking and volumetric elastic registration'. In: *Information Processing in Medical Imaging*. Springer, pp. 78–91.

Friedl, Mark A and Carla E Brodley (1997). 'Decision tree classification of land cover from remotely sensed data'. In: *Remote sensing of environment* 61.3, pp. 399–409.

Fripp, Jurgen, Stuart Crozier, Simon Warfield and Sébastien Ourselin (2005). 'Automatic initialization of 3D deformable models for cartilage segmentation'. In: *Digital Image Computing: Techniques and Applications, 2005. DICTA'05. Proceedings 2005*. IEEE, pp. 74–74.

Geremia, Ezequiel, Olivier Clatz, Bjoern H Menze, Ender Konukoglu, Antonio Criminisi and Nicholas Ayache (2011). 'Spatial decision forests for MS lesion segmentation in multi-channel magnetic resonance images'. In: *NeuroImage* 57.2, pp. 378–390.

Gini, Corrado (1921). 'Measurement of inequality of incomes'. In: *The Economic Journal*, pp. 124–126.

Gislason, Pall Oskar, Jon Atli Benediktsson and Johannes R Sveinsson (2006). 'Random forests for land cover classification'. In: *Pattern Recognition Letters* 27.4, pp. 294–300.

Goodall, Colin (1991). 'Procrustes methods in the statistical analysis of shape'. In: *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 285–339.

Gower, John C (1975). 'Generalized procrustes analysis'. In: *Psychometrika* 40.1, pp. 33–51.

Han, Xiao, Mischa S Hoogeman, Peter C Levendag, Lyndon S Hibbard, David N Teguh, Peter Voet, Andrew C Cowen and Theresa K Wolf (2008). 'Atlas-based auto-segmentation of head and neck CT images'. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008*. Springer, pp. 434–441.

Heimann, Tobias and Hans-Peter Meinzer (2009). 'Statistical shape models for 3D medical image segmentation: A review'. In: *Medical image analysis* 13.4, pp. 543–563.

Helleputte, Thibault and Pierre Dupont (2009). 'Partially supervised feature selection with regularized linear models'. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. Montreal, Quebec, Canada: ACM, pp. 409–416.

Hill, Andrew, Timothy F Cootes and Christopher J Taylor (1992). 'A generic system for image interpretation using flexible templates'. In: *BMVC92*. Springer, pp. 276–285.

Hill, Andrew and Christopher J Taylor (1992). 'Model-based image interpretation using genetic algorithms'. In: *Image and Vision Computing* 10.5, pp. 295–300.

Ho, Tin Kam (1998). 'The random subspace method for constructing decision forests'. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 20.8, pp. 832–844.

Hubel, David H (1995). *Eye, brain, and vision*. Scientific American Library/Scientific American Books.

Hurkmans, Coen W, Jacques H Borger, Bradley R Pieters, Nicola S Russell, Edwin P.M Jansen and Ben J Mijnheer (2001). 'Variability in target volume delineation on CT scans of the breast'. In: *International Journal of Radiation Oncology*Biology*Physics* 50.5, pp. 1366–1372. ISSN: 0360-3016.

Jemal, Ahmedin, Freddie Bray, Melissa M Center, Jacques Ferlay, Elizabeth Ward and David Forman (2011). 'Global cancer statistics'. In: *CA: a cancer journal for clinicians* 61.2, pp. 69–90.

Jia, Hongjun, Guorong Wu, Qian Wang and Dinggang Shen (2010). 'ABSORB: Atlas building by self-organized registration and bundling'. In: *NeuroImage* 51.3, pp. 1057–1070.

Ke, Yan and Rahul Sukthankar (2004). 'PCA-SIFT: A more distinctive representation for local image descriptors'. In: *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. Vol. 2. IEEE, pp. II–506.

Kelemen, András, Gábor Székely and Guido Gerig (1999). 'Elastic model-based segmentation of 3-D neuroradiological data sets'. In: *Medical Imaging, IEEE Transactions on* 18.10, pp. 828–839.

Kimpe, Tom and Tom Tuytschaever (2007). 'Increasing the number of gray shades in medical display systems—how much is enough?' In: *Journal of digital imaging* 20.4, pp. 422–432.

Klein, Stefan, Uulke A van der Heide, Irene M Lips, Marco van Vulpen, Marius Staring and Josien PW Pluim (2008). 'Automatic segmentation of the prostate in 3D MR images by atlas matching using localized mutual information'. In: *Medical physics* 35.4, pp. 1407–1417.

Knutsson, Hans and Mats Andersson (2005). 'Morphons: Segmentation using elastic canvas and paint on priors'. In: *Image Processing, 2005. ICIP 2005. IEEE International Conference on*. Vol. 2. IEEE, pp. II–1226.

Kotcheff, Aaron CW and Christopher J Taylor (1998). 'Automatic construction of eigenshape models by direct optimization'. In: *Medical Image Analysis* 2.4, pp. 303–314.

Kubat, Miroslav and Martin Cooperson Jr (2001). 'A reduction technique for nearest-neighbor classification: Small groups of examples'. In: *Intelligent Data Analysis* 5.6, pp. 463–476.

Kullback, Solomon and Richard A Leibler (1951). 'On information and sufficiency'. In: *The Annals of Mathematical Statistics*, pp. 79–86.

Landis, Daniel M., Weixiu Luo, Jun Song, Jennifer R. Bellon, Rinaa S. Punglia, Julia S. Wong, Joseph H. Killoran, Rebecca Gelman and Jay R. Harris (2007). 'Variability Among Breast Radiation Oncologists in Delineation of the Postsurgical Lumpectomy Cavity'. In: *International Journal of Radiation Oncology*Biology*Physics* 67.5, pp. 1299–1308. ISSN: 0360-3016.

Langerak, Thomas Robin, Uulke A van der Heide, Alexis NTJ Kotte, Max A Viergever, Marco van Vulpen and Josien PW Pluim (2010). 'Label fusion in atlas-based segmentation using a selective and iterative method for performance level estimation (SIMPLE)'. In: *Medical Imaging, IEEE Transactions on* 29.12, pp. 2000–2008.

Lee, John A (2010). 'Segmentation of positron emission tomography images: some recommendations for target delineation in radiation oncology'. In: *Radiotherapy and Oncology* 96.3, pp. 302–307.

Levenshtein, VI (1966). 'Binary Codes Capable of Correcting Deletions, Insertions and Reversals'. In: *Soviet Physics Doklady* 10, p. 707.

Li, X. Allen et al. (2009). 'Variability of Target and Normal Structure Delineation for Breast Cancer Radiotherapy: An RTOG Multi-Institutional and Multiobserver Study'. In: *International Journal of Radiation Oncology*Biology*Physics* 73.3, pp. 944–951. ISSN: 0360-3016.

Lienhart, Rainer and Jochen Maydt (2002). 'An extended set of haar-like features for rapid object detection'. In: *Image Processing. 2002. Proceedings. 2002 International Conference on*. Vol. 1. IEEE, pp. I–900.

Lowe, David G (1999). 'Object recognition from local scale-invariant features'. In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Vol. 2. Ieee, pp. 1150–1157.

Luo, Suhuai, Qingmao Hu, Xiangjian He, Jiaming Li, Jesse S Jin and Mira Park (2009). 'Automatic liver parenchyma segmentation from abdominal CT images using support vector machines'. In: *Complex Medical Engineering, 2009. CME. ICME International Conference on*. IEEE, pp. 1–5.

Meyer, Fernand (1994). 'Topographic distance and watershed lines'. In: *Signal processing* 38.1, pp. 113–125.

Mohan, Anuj, Constantine Papageorgiou and Tomaso Poggio (2001). 'Example-based object detection in images by components'. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23.4, pp. 349–361.

Mountrakis, Giorgos, Jungho Im and Caesar Ogole (2011). 'Support vector machines in remote sensing: A review'. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 66.3, pp. 247–259.

Nadeau, Claude and Yoshua Bengio (2003). 'Inference for the generalization error'. In: *Machine Learning* 52.3, pp. 239–281.

Najman, Laurent and Michel Couprie (2006). 'Building the component tree in quasi-linear time'. In: *Image Processing, IEEE Transactions on* 15.11, pp. 3531–3539.

Osuna, Edgar, Robert Freund and Federico Girosi (1997). 'Training support vector machines: an application to face detection'. In: *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. IEEE, pp. 130–136.

Otsu, Nobuyuki (1975). 'A threshold selection method from gray-level histograms'. In: *Automatica* 11.285-296, pp. 23–27.

Pal, M (2005). 'Random forest classifier for remote sensing classification'. In: *International Journal of Remote Sensing* 26.1, pp. 217–222.

Pal, Mahesh and Paul M Mather (2003). 'An assessment of the effectiveness of decision tree methods for land cover classification'. In: *Remote sensing of environment* 86.4, pp. 554–565.

Papageorgiou, Constantine and Tomaso Poggio (2000). 'A trainable system for object detection'. In: *International Journal of Computer Vision* 38.1, pp. 15–33.

Pappas, Thrasyvoulos N (1992). 'An adaptive clustering algorithm for image segmentation'. In: *Signal Processing, IEEE Transactions on* 40.4, pp. 901–914.

Petersen, Ross P., Pauline T. Truong, Hosam A. Kader, Eric Berthelet, Junella C. Lee, Michelle L. Hilts, Adam S. Kader, Wayne A. Beckham and Ivo A. Olivotto (2007). 'Target Volume Delineation for Partial Breast Radiotherapy Planning: Clinical Characteristics Associated with Low Interobserver Concordance'. In: *International Journal of Radiation Oncology*Biology*Physics* 69.1, pp. 41–48. ISSN: 0360-3016.

Pitiot, Alain, Arthur W Toga and Paul M Thompson (2002). 'Adaptive elastic segmentation of brain MRI via shape-model-guided evolutionary programming'. In: *Medical Imaging, IEEE Transactions on* 21.8, pp. 910–923.

Pitkänen, M.A., K.A. Holli, A.T. Ojala and P. Laippala (2001). 'Quality assurance in radiotherapy of breast cancer: Variability in planning target volume delineation'. In: *Acta Oncologica* 40.1, pp. 50–55.

Pizer, Stephen M, P Thomas Fletcher, Sarang Joshi, Andrew Thall, James Z Chen, Yonatan Fridman, Daniel S Fritsch, A Graham Gash, John M Glotzer, Michael R Jiroutek et al. (2003). 'Deformable m-reps for 3D medical image segmentation'. In: *International Journal of Computer Vision* 55.2-3, pp. 85–106.

Qazi, Arish A, Vladimir Pekar, John Kim, Jason Xie, Stephen L Breen and David A Jaffray (2011). 'Auto-segmentation of normal and target structures in head and neck CT images: a feature-driven model-based approach'. In: *Medical physics* 38.11, pp. 6160–6170.

Quinlan, John Ross (1986). 'Induction of decision trees'. In: *Machine learning* 1.1, pp. 81–106.

— (1993). *C4. 5: programs for machine learning*. Vol. 1. Morgan kaufmann.

Rangarajan, Anand, Haili Chui and Fred L Bookstein (1997). 'The softassign procrustes matching algorithm'. In: *Information Processing in Medical Imaging*. Springer, pp. 29–42.

Rohlfing, Torsten, Robert Brandt, Randolf Menzel and Calvin R Maurer Jr (2004). 'Evaluation of atlas selection strategies for atlas-based image segmentation with application to confocal microscopy images of bee brains'. In: *NeuroImage* 21.4, pp. 1428–1442.

Rohlfing, Torsten, Robert Brandt, Randolf Menzel, Daniel B Russakoff and Calvin R Maurer Jr (2005). 'Quo vadis, atlas-based segmentation?' In: *Handbook of Biomedical Image Analysis*. Springer, pp. 435–486.

Rohr, Karl, H Siegfried Stiehl, Rainer Sprengel, Wolfgang Beil, Thorsten M Buzug, Jürgen Weese and MH Kuhn (1996). 'Point-based elastic registration of medical image data using approximating thin-plate splines'. In: *Visualization in Biomedical Computing*. Springer, pp. 297–306.

Rueckert, Daniel, Alejandro F Frangi and Julia A Schnabel (2001). 'Automatic construction of 3D statistical deformation models using non-rigid registration'. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2001*. Springer, pp. 77–84.

Rueckert, Daniel, Luke I Sonoda, Carmel Hayes, Derek LG Hill, Martin O Leach and David J Hawkes (1999). 'Nonrigid registration using free-form deformations: application to breast MR images'. In: *Medical Imaging, IEEE Transactions on* 18.8, pp. 712–721.

Sabuncu, Mert R, BT Thomas Yeo, Koen Van Leemput, Bruce Fischl and Polina Golland (2010). 'A generative model for image segmentation based on label fusion'. In: *Medical Imaging, IEEE Transactions on* 29.10, pp. 1714–1729.

Sauvola, Jaakko and Matti Pietikäinen (2000). 'Adaptive document image binarization'. In: *Pattern recognition* 33.2, pp. 225–236.

Shen, D, EH Herskovits and C Davatzikos (2001). 'An adaptive-focus statistical shape model for segmentation and shape modeling of 3-D brain structures.' In: *IEEE transactions on medical imaging* 20.4, p. 257.

Shi, Jianbo and Jitendra Malik (2000). 'Normalized cuts and image segmentation'. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.8, pp. 888–905.

Soler, Luc, Herve Delingette, Grégoire Malandain, Johan Montagnat, Nicholas Ayache, Christophe Koehl, Olivier Dourthe, Benoit Malassagne, Michelle Smith, Didier Mutter et al. (2001). 'Fully automatic anatomical, pathological, and functional segmentation from CT scans for hepatic surgery'. In: *Computer Aided Surgery* 6.3, pp. 131–142.

Song, Mingjun and Daniel Civco (2004). 'Road extraction using SVM and image segmentation'. In: *Photogrammetric Engineering & Remote Sensing* 70.12, pp. 1365–1371.

Steene, Jan Van de, Nadine Linthout, Johan de Mey, Vincent Vinh-Hung, Cornelia Claassens, Marc Noppen, Arjan Bel and Guy Storme (2002). 'Definition of gross tumor volume in lung cancer: inter-observer variability'. In: *Radiotherapy and Oncology* 62.1, pp. 37–49. ISSN: 0167-8140.

Stegmann, Mikkel B, Rune Fisker and Bjarne K Ersbøll (2001). 'Extending and applying active appearance models for automated, high precision segmenta-

tion in different image modalities'. In: *in Scandinavian Conference on Image Analysis*.

Struikmans, Henk, Carla Wárlám-Rodenhuis, Tanja Stam, Gerard Stapper, Robbert J.H.A. Tersteeg, Gijsbert H. Bol and Cornelis P.J. Raaijmakers (2005). 'Interobserver variability of clinical target volume delineation of glandular breast tissue and of boost volume in tangential breast irradiation'. In: *Radiotherapy and Oncology* 76.3, pp. 293–299. ISSN: 0167-8140.

Subsol, Gérard, Jean-Philippe Thirion and Nicholas Ayache (1998). 'A scheme for automatically building three-dimensional morphometric anatomical atlases: application to a skull atlas'. In: *Medical Image Analysis* 2.1, pp. 37–60.

Tai, Patricia, Jake Van Dyk, Edward Yu, Jerry Battista, Larry Stitt and Terry Coad (1998). 'Variability of target volume delineation in cervical esophageal cancer'. In: *International Journal of Radiation Oncology*Biology*Physics* 42.2, pp. 277–288. ISSN: 0360-3016.

Thirion, Jean-Philippe (1998). 'Image matching as a diffusion process: an analogy with Maxwell's demons'. In: *Medical image analysis* 2.3, pp. 243–260.

van der Kogel, Albert and Michael C Joiner (2009). *Basic Clinical Radiobiology Fourth Edition*. CRC Press.

Viola, Paul and Michael Jones (2001). 'Rapid object detection using a boosted cascade of simple features'. In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Vol. 1. IEEE, pp. I–511.

Ward, JF (1988). 'DNA damage produced by ionizing radiation in mammalian cells: identities, mechanisms of formation, and reparability'. In: *Progress in nucleic acid research and molecular biology* 35, pp. 95–125.

Warfield, Simon K, Kelly H Zou and William M Wells (2004). 'Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation'. In: *Medical Imaging, IEEE Transactions on* 23.7, pp. 903–921.

Weiss, Elisabeth, Susanne Richter, Thomas Krauss, Silke I Metzelthin, Andrea Hille, Olivier Pradier, Birgit Siekmeyer, Hilke Vorwerk and Clemens F Hess (2003). 'Conformal radiotherapy planning of cervix carcinoma: differences in the delineation of the clinical target volume: A comparison between gynaecologic and radiation oncologists'. In: *Radiotherapy and Oncology* 67.1, pp. 87–95. ISSN: 0167-8140.

Wilson, D Randall and Tony R Martinez (2000). 'Reduction Techniques for Instance-Based Learning Algorithms'. In: *Machine Learning* 38.3, pp. 257–286.

Wong, Elaine K., Pauline T. Truong, Hosam A. Kader, Alan M. Nichol, Lee Salter, Ross Petersen, Elaine S. Wai, Lorna Weir and Ivo A. Olivotto (2006). 'Consistency in seroma contouring for partial breast radiotherapy: Impact of guidelines'. In: *International Journal of Radiation Oncology*Biology*Physics* 66.2, pp. 372–376. ISSN: 0360-3016.

Yamamoto, Masashi, Yasushi Nagata, Kaoru Okajima, Takashi Ishigaki, Rumi Murata, Takashi Mizowaki, Masaki Kokubo and Masahiro Hiraoka (1999). 'Differences in target outline delineation from CT scans of brain tumours using different methods and different observers'. In: *Radiotherapy and Oncology* 50.2, pp. 151–156. ISSN: 0167-8140.

Zhang, Jianguo, Kai-Kuang Ma, Meng-Hwa Er, Vincent Chong et al. (2004). 'Tumor segmentation from magnetic resonance imaging by learning via one-class support vector machine'. In: *International Workshop on Advanced Image Technology (IWAIT'04)*, pp. 207–211.

Zhu, Song Chun and Alan Yuille (1996). 'Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation'. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 18.9, pp. 884–900.

Appendices

Appendix A

Incremental feature computation and classification for image segmentation

Guillaume Bernard, John A. Lee, Michel Verleysen

Paper published in the proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (Conference) – 2012

Abstract

Image segmentation problems can be solved with classification algorithms. However, their use is limited to features derived from intensities of pixels or patches. Features such as contiguity of two regions cannot be considered without prior knowledge of one of the two class labels. Instead of stacking various classification algorithms, we describe an incremental classifier that works in a space where features are progressively evaluated. Experiments on artificial images demonstrate the capabilities of this incremental scheme.

A.1 Introduction

Various approaches can address the problem of image segmentation. Histogram thresholding (Otsu 1975), pixel or patch clustering (Shi and Malik 2000), gradient peak detection with active contours (Kass et al. 1988) or watersheds (Cousty et al. 2009) are only a few of them. Many of these methods are unsupervised, though they can take into account some a priori information, such as the expected region shape, size, and edge smoothness. As a matter of fact, supervised segmentation raises less interest, mainly because usual classification algorithm can only deal with *intrinsic* features, such as pixel coordinates, pixel intensities, or patch textures. On the other hand, *extrinsic* features that describe the relationships between two or more classes in the image are difficult to take into account. For instance, let us consider a feature such as the spatial distance to the region of class Y in the image. When training a classifier, this feature can be trivially computed since the labels of all regions are known in a pre-segmented image. In contrast, in a test image, measuring this distance requires some region to be already given the label Y. A pragmatic solution to take benefit of extrinsic features consists in stacking at least two classifiers. The first one involves only intrinsic features. The resulting partial classification can then serve to compute a first batch of extrinsic features, which are fed into a second classifier, and so on. This incremental process has been investigated in (Gould et al. 2008), for example.

In this paper, we suggest a more generic approach, where the multiclass problem is first divided into several binary classification problems (one class versus all others). Binary classifiers based on the principle of the K-nearest neighbors (KNN) tackle these problems repeatedly in an iterative way. At each iteration, precomputed feature relevance factors that reflect the ability of a given feature to discriminate a given class are used to select a binary classifier. Hence, this incremental process attempts to solve first the simplest binary classification problems, in order to enrich as quickly as possible the pool of known extrinsic features. Eventually, at the end of this incremental process, a multiclass classifier is used with all features. The efficacy of the approach is demonstrated in a few image segmentation tasks.

The rest of this paper is organized as follows. Section A.2 introduces the notations for intrinsic, extrinsic, and known features. Section A.3 describes the incremental procedure for feature computation and partial classification, as well as the final multi-class classification, which improves the accuracy. Section A.4 reports and discusses the experimental results. Finally, Section A.5 draws the

conclusions and sketches some perspective for future work.

A.2 Intrinsic, extrinsic, and known features

Let $\{\mathbf{X}, \mathbf{y}\}$ denote a data set where $\mathbf{X} = [x_{ij}]_{1 \leq i \leq D, 1 \leq j \leq N}$ contains the features and $\mathbf{y} = [y_j]_{1 \leq j \leq N}$ gives the corresponding labels. Labels y_j take their value in $\{Y_1, Y_2, \ldots, Y_C\}$, where C is the total number of classes.

In the training phase, the rows of data set $\mathbf{X}$ can be splitted into intrinsic and extrinsic features. As a reminder, intrinsic features are known at all times, independently of any classification, whereas extrinsic features require at least one class to be identified in the data set. Let $\mathcal{F}_{\text{in}}$, $\mathcal{F}_{\text{ex}}$ denote the non-intersecting sets of indices corresponding to intrinsic and extrinsic features. As extrinsic features represent relationships between objects of different classes, we assume that N is a multiple of C and that the data set consists of N/C groups of objects where all classes are instantiated once. Within the framework of image segmentation, this means that each image contains a single object of all kinds. This aasumption avoids undetermined or ambiguous relationships.

In the test phase, the unlabeled data set $\mathbf{X}'$ contains missing values for all extrinsic features. During the incremental classification process, blanks are filled in as soon as class labels are attributed. Let $\mathcal{F}_{\text{kn}}^{(t)}$ denote the set of indices corresponding to known features at iteration t of the incremental procedure. We have $\mathcal{F}_{\text{in}} = \mathcal{F}_{\text{kn}}^{(1)} \subseteq \mathcal{F}_{\text{kn}}^{(t)} \subseteq \mathcal{F}_{\text{kn}}^{(t+1)}$.

As the evaluation of yet unknown extrinsic features requires some precise class label to be attributed with reasonable certainty, it is more natural to use binary classifiers. Therefore, we must determine a running order for the binary classifiers. For this purpose, the already known features must be ranked according to their usefulness for binary classification. We suggest a ranking that assesses the overlapping of classes for a given feature. Let $\mathcal{N}_j^K(\mathbf{X})$ denote the set of indices corresponding to the K nearest neighbors of the jth column $\mathbf{x}_{\bullet j}$ of data set $\mathbf{X}$. Let $\mathcal{C}_k(\mathbf{y}) = \{p \text{ s.t. } y_p = Y_k\}$ be the set of indices associated with class Y_k. The usefulness of a certain feature to classify data inside or outside class Y_k can be measured as

$$s_{ik} = \frac{1}{K|\mathcal{C}_k(\mathbf{y})|} \sum_{p \in \mathcal{C}_k(\mathbf{y})} |\{q \text{ s.t. } q \in \mathcal{N}_p^K(\mathbf{e}_i^T \mathbf{X}) \text{ and } y_q = Y_k\}| \ ,$$

where $|A|$ denotes the cardinality of set A and $\mathbf{e}_i$ is a vector of zeros everywhere

except the ith element equal to 1. The value of s_{ik} can range from 0 to 1. The latter value indicates that class Y_k does not overlap with other classes along the axis of the ith feature. Each row of matrix $\mathbf{S} = [s_{ik}]$ can be computed quite efficiently by sorting vector $\mathbf{e}_i^T\mathbf{X}$ and sliding a $(2K+1)$-wide window. This leads to a computational complexity of $\mathcal{O}(N\ln(N) + NK\ln(K))$ for each feature.

A.3 Incremental feature computation and classification

The incremental procedure that we propose works as follows. First, we store the centered and normalized training set. Second, we compute matrix $\mathbf{S}$ and initialize $\mathcal{F}_{\text{kn}} = \mathcal{F}_{\text{in}}$. Next, we start the incremental iterations. At each iteration we go through the following steps:

1. Compute $\ell = \max_k s_{ik}$ with $i \in \mathcal{F}_{\text{kn}}$; set $s_{i\ell} = 0$.
2. Train the ℓth binary classifier on the reduced data set $[x_{ij}]_{i\in\mathcal{F}_{\text{kn}}^{(t)}, 1\leq j\leq N}$.
3. Attribute the class label Y_ℓ to the object in the test set having the highest probability to belong this class (make a random pick in case of a tie).
4. Compute all extrinsic features that involve a relationship with the object of class Y_ℓ and insert their index into $\mathcal{F}_{\text{kn}}^{(t)}$ to obtain $\mathcal{F}_{\text{kn}}^{(t+1)}$.
5. If $\mathcal{F}_{\text{kn}}^{(t+1)} = \mathcal{F}_{\text{in}} \cup \mathcal{F}_{\text{ex}}$, then stop, otherwise start a new iteration.

As features are ranked with matrix $\mathbf{S}$, the classifiers can be trained beforehand to increase the computational efficiency. At the end of the procedure, all features are known, but the classification might not be optimal. Once all features are known, we suggest the use of a multi-class classifier. This last step gives to the object to be classified their final class label. Moreover, this final global classification slightly improves the results, as shown in the experiments.

A.4 Experiments and results

As a proof of concept, we illustrate the principle of incremental classification with a simple image segmentation problem. The data set consists of artificial images of small worms or caterpillars. In each image, the caterpillar comprises

a bright head and 5 dark, almost equidistant body segments (Fig. A.1). The position, orientation, and twist of the caterpillars vary in each image. Beyond the rather easy segmentation of the patches corresponding to the caterpillar's head and body, the goal of the problem is to correctly label the first, second, ... and sixth segments in spite of their identical color. Incremental classification provides a non-imperative way to solve the problem. The features for each segmented patch (head or body) are the gray level (intrinsic) and the distance to all other patches (extrinsic). The gray level allows the head to be identified. Knowing where the head is then allows some distances to be measured, which help to progressively distinguish the body segments. In practice, the data

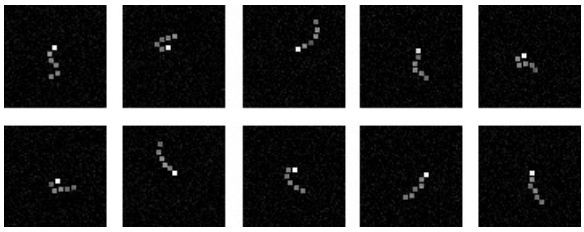


Figure A.1 – Some images of the caterpillar problem.

set contains 100 images. Six patches in each image can be segmented (hence, $N = 600$. Seven features characterize each patch (gray level plus the distance to the head, first body segment, etc.). Only the gray level is an intrinsic feature. Label 1 is associated with the head, label 2 with the first body segment, etc.

We randomly split this data set in two parts: 90 images serves as training set, whereas the remaining 10 images form the test set. The known features in the test set are centered and normalized by subtraction of the mean and division by the standard deviation computed on the training set. The binary and multi-class classifiers rely on the method of the large margin nearest neighbors (LMNN) (Weinberger and Saul 2009), which combines a usual KNN with metric learning. In our incremental procedure, the Mahalonobis distance of the LMNN is optimized on the training set reduced to the features that are known at each iteration. The extrinsic features are inferred for each image individually. In each image the patch that belongs with the highest probability to the class we wish to identify is used to compute the distance to this class in the considered image. The whole classification procedure is repeated with 20 different training sets for various values of K. The same K is used both in the computation of matrix S and in all LMNNs. Accuracy is defined as the number of data well

classified divided by the total number of data.

We analyzed the error rates at the end of the incremental procedure, just before the final multi-class classification, during each iteration and for each class. We also analyzed the order of feature extraction. Table A.1 reveals no significant difference in final classification accuracy for the different values of K.

K	3	5	7	9	15
Accuracy	0.959	0.968	0.961	0.961	0.969
Std.	0.054	0.030	0.049	0.036	0.035

Table A.1 – Accuracy at the end of the classification.

The final multi-class classification improves the accuracy (Table A.1 and A.2). Indeed, at the end of the incremental classification with binary LMNNs, some data might remain unlabeled and the final classification addresses this issue. Nevertheless, this last iteration cannot correct past classification mistakes.

k	3	5	7	9	15
Accuracy	0.952	0.957	0.954	0.953	0.967
Std.	0.054	0.026	0.044	0.037	0.036

Table A.2 – Accuracy before the last classification.

Table A.3 shows that the order of the features induced by **S** is stable across the 20 different test sets. For the first 4 iterations, it always extract the features that involve the knowledge of class 1, 2, 4, and 1. We enabled the possibility to recompute a feature: in the experiment, the extrinsic feature depending on classes 1 and 4 are modified after a first evaluation. Such a class relabeling allows feature computation errors to be corrected in the incremental procedure. The second classification involves more features than the first one and is thus expected to be more reliable.

Table A.4 shows that the third iteration is the less accurate. This iteration identifies class 4 (Table A.3). Even with the lower accuracy at iteration 3, the next iterations yield a good accuracy. Table A.5 reports that accuracy of the binary classifiers at the class corresponding to the first body segment next to the head is has the best accuracy. The class 3 is well classified despite the fact that its classification occurred at the end of the feature extraction process.

Class	1	2	3	4	5	6
Iteration 1	20					
2		20				
3				20		
4	20					
5			2			18
6					18	2
7				18	2	
8			18			

Table A.3 – Number of time a class is selected for binary classification in each iteration ($K = 15$). A blank cell means zero.

Iteration	1	2	3	4	5	6	7	8
Accuracy	1.000	0.994	0.960	1.000	0.988	0.982	0.976	0.994
Std.	0.000	0.011	0.023	0.000	0.010	0.021	0.028	0.012

Table A.4 – Binary classifier accuracy at each iteration ($K = 15$).

Figure A.2 shows that the images that lead to big classification mistakes are

Class	1	2	3	4	5	6
Accuracy	1	0.994	0.994	0.965	0.984	0.986
Std.	0	0.011	0.011	0.020	0.021	0.011

Table A.5 – Accuracy of the binary classifiers associated with each class ($K = 15$).

those depicting highly twisted or curled with large.

Figure A.2 – Caterpillars for which we have classification issues.

A.5 Conclusion

This paper describes a procedure for incremental classification. It can deal with problems where the value of some features requires a partial classification to be already known. The process of incremental classification aims at refining the partial classification in an iterative way. The procedure is generic and can solve the subproblems in each iteration with various classification techniques (e.g. naives Bayes, KNN, SVM, etc.). The final multi-class classifier can be changed as well. Depending on the problem at hand, the procedure must be adapted with appropriate definitions of features and relevance factors. Failure to do so increases the risk of error propagation in the incremental process. Experiments on artificial images show that the procedure is effective.

In the future, we will investigate the possibility to resort to a single classifier that deals with all intrinsic and extrinsic features at all times, thanks to the use of adaptive relevance factors.

A.6 Bibliography

Cousty, Jean, Gilles Bertrand, Laurent Najman and Michel Couprie (2009). 'Watershed cuts: Minimum spanning forests and the drop of water principle'. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31.8, pp. 1362–1374.

Gould, S., J. Rodgers, D. Cohen, G. Elidan and D. Koller (2008). 'Multi-class segmentation with relative location prior'. In: *International Journal of Computer Vision* 80.3, pp. 300–316.

Kass, M., A. Witkin and D. Terzopoulos (1988). 'Snakes: Active contour models'. In: *International journal of computer vision* 1.4, pp. 321–331.

Otsu, Nobuyuki (1975). 'A threshold selection method from gray-level histograms'. In: *Automatica* 11.285-296, pp. 23–27.

Shi, Jianbo and Jitendra Malik (2000). 'Normalized cuts and image segmentation'. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.8, pp. 888–905.

Weinberger, K.Q. and L.K. Saul (2009). 'Distance metric learning for large margin nearest neighbor classification'. In: *The Journal of Machine Learning Research* 10, pp. 207–244.

Appendix B

Segmentation with Incremental Classifiers

Guillaume Bernard, John A. Lee, Michel Verleysen

Paper published in the proceedings of the International Conference on Image Analysis and Processing (Conference) – 2013

Abstract

Radiotherapy treatment planning requires physicians to delineate the target volumes and organs at risk on 3D images of the patient. This segmentation task consumes a lot of time and can be partly automated with atlases (reference images segmented by experts). To segment any new image, the atlas is non-rigidly registered and the organ contours are then transferred. In practice, this approach suffers from the current limitations of non-rigid registration. We propose an alternative approach to extract and encode the physician's expertise. It relies on a specific classification method that incrementally extracts information from groups of pixels in the images. The incremental nature of the process allows us to extract features that depend on partial classification results but also convey richer information. This paper is a first investigation of such an incremental scheme, illustrated with experiments on artificial images.

B.1 Introduction

Cancer treatment with radiation beams amounts to a ballistic problem where the dose to the tumor must be maximized while the dose at surrounding healthy tissues must be minimized to avoid secondary effects. In order to achieve the best tradeoff, 3D images of the patients must be segmented to identify the tumor and the organs at risk. The physicians use an electronic pen or a mouse to delineate these volumes on each slice. Although it consumes a lot of time, delineation usually remains manual because it involves complex expertise. This explains why usual image segmentation methods such as histogram thresholding (Otsu 1975), pixel or patch clustering (Shi and Malik 2000), gradient peak detection with active contours (Kass et al. 1988), or watersheds (Beucher and Meyer 1992; Cousty et al. 2009) cannot solve the problem. Many of these methods are unsupervised, even though some of them can take into account some a priori information, such as the expected region shape, size, and edge smoothness. On the other hand, supervised segmentation remains difficult to apply, mainly because the encoding of expertise and a priori information is far from being trivial. The most successful approach is the use of atlases, which are (banks of) images that are segmented beforehand by experts. Atlases can be deformed to match any new image with a non-rigid registration algorithm (Bondiau et al. 2005). Once the two images are aligned, the contours or regions can be propagated from the atlas to the new image. This approach suffers from the shortcomings of the registration algorithms it relies on. Many of these algorithms regularize the deformation vector field in a simplistic or unrealistic way. This leads to segmentation results that are globally correct but often inaccurate near the region boundaries; the required corrections annihilate the expected gain of time.

From a theoretical point of view, the segmentation of several objects in an image amounts to a supervised multiclass classification problem. In practice, however, this alternative approach faces several obstacles, the most prominent being that usual classification algorithms can only deal with features that are class-independent and thus *intrinsic* to the image, such as pixel coordinates, pixel luminance, or patch textures. These features convey limited information about the objects depicted in the images. On the other hand, *extrinsic* features that describe the relationships between two or more classes in the image have a richer content but they are more difficult to take into account. For instance, let us consider a feature such as the spatial distance to the region of class Y in the image. When training a classifier, this feature can be trivially computed

since the labels of all regions are known in a pre-segmented image. In contrast, in a test image, measuring this distance requires at least some pixels to be already given label Y. A pragmatic solution to take benefit of extrinsic features consists in stacking at least two classifiers. The first one involves only intrinsic features. The resulting partial classification can then serve to compute a first batch of extrinsic features, which are fed into a second classifier, and so on. This incremental process has been investigated in (Gould et al. 2008), for example.

This paper suggests a generic approach, where the images are first over-segmented with a watershed transform. Information extracted from the watersheds are used for the multiclass problem, which is first divided into several binary classification problems (one class versus all others). Binary classifiers (k nearest neighbors (Cover and Hart 1967), support vector machine (Joachims 1999) and random forest (Breiman 2001)) tackle these problems repeatedly in an iterative way. Two methods are proposed to select the order in which the binary classifiers should be run. At the end of this incremental process, a multiclass classifier is used with all computed features, to improve the final results. The efficacy of the approach is demonstrated in a few segmentation tasks involving artificial images.

This paper is organized as follows. Section B.2 briefly describes the method used for the unsupervised over-segmentation of the images. Section B.3 introduces the notations for intrinsic, extrinsic, and known features; it also details the two proposed methods of feature ranking. Section B.4 describes the incremental procedure for feature computation and partial classification, as well as the final multiclass classification. Section B.5 reports and discusses the experimental results. Finally, Section B.6 draws the conclusions.

B.2 Unsupervised Over-segmentation

In order to obtain a first, unsupervised segmentation of the images, a watershed transform is used (Beucher and Meyer 1992; Cousty et al. 2009; Beucher and Lantuéjoul 1979). The principle is to consider the gradient magnitude image as a topographic relief where a flooding is simulated. The dam lines separating the catchment basins yield an over-segmentation of the image. This preprocessing step limits the computational complexity by working with consistent groups of similar pixels, called *super-pixels*, rather than with pixels themselves. To control the over-segmentation granularity, we use a reformulation of the watershed transform as a graph-cut problem, such as described in (Cousty et al. 2009)

and (Najman and Couprie 2006). This watershed-cut method works with both 2D and 3D images and it includes a graph filtering step that affects the number of super-pixels.

B.3 Intrinsic, Extrinsic, and Known Features

Let $\{\mathbf{X}, \mathbf{y}\}$ denote a data set where $\mathbf{X} = [x_{ij}]_{1\leq i\leq D, 1\leq j\leq N}$ contains the features and $\mathbf{y} = [y_j]_{1\leq j\leq N}$ gives the corresponding labels. Label y_j takes its value in $\{Y_1, Y_2, \ldots, Y_C\}$, where C is the number of classes.

In the training phase, the rows of data set $\mathbf{X}$ can be distinguished between intrinsic and extrinsic features. Let $\mathcal{F}_{\text{in}}$, $\mathcal{F}_{\text{ex}}$ denote the non-intersecting sets of indices corresponding to intrinsic and extrinsic features. As extrinsic features represent relationships between objects of different classes, we assume that N is a multiple of C and that the data set consists of N/C groups of objects where all classes are instantiated at least once. Within the framework of image segmentation, this means that each image contains at least an object of each kind. This assumption avoids undetermined or ambiguous relationships.

In the test phase, the unlabeled data set $\mathbf{X}'$ contains missing values for all extrinsic features. During the incremental classification process, blanks are filled in as soon as class labels are attributed. Let $\mathcal{F}_{\text{kn}}^{(t)}$ denote the set of indices corresponding to known features at iteration t of the incremental procedure. We have $\mathcal{F}_{\text{in}} = \mathcal{F}_{\text{kn}}^{(1)} \subseteq \mathcal{F}_{\text{kn}}^{(t)} \subseteq \mathcal{F}_{\text{kn}}^{(t+1)}$.

As the evaluation of yet unknown extrinsic features requires a certain class label to be attributed with reasonable certainty, it is more natural to use binary classifiers that are specialized for the considered class. Therefore, an execution sequence of the binary classifiers has to be determined. For this purpose, the already known features must be ranked according to their usefulness for binary classification. This paper proposes two methods to rank the features.

B.3.1 Feature Ranking by Nearest Neighbors

As a first method, we suggest a ranking that assesses the overlap of classes for a given feature. Let $\mathcal{N}_j^K(\mathbf{X})$ denote the set of indices corresponding to the K nearest neighbors of the j^{th} super-pixel $\mathbf{x}_{\bullet j}$ of data set $\mathbf{X}$. Let $\mathcal{C}_k(\mathbf{y}) = \{p \text{ s.t. } y_p = Y_k\}$ be the set of indices associated with class Y_k. The usefulness of a certain feature to classify data inside or outside class Y_k can be measured

as

$$s_{ik} = \frac{1}{K|\mathcal{C}_k(\mathbf{y})|} \sum_{p \in \mathcal{C}_k(\mathbf{y})} |\{q \text{ s.t. } q \in \mathcal{N}_p^K(\mathbf{e}_i^T \mathbf{X}) \text{ and } y_q = Y_k\}| \ ,$$

where $|A|$ denotes the cardinality of set A and $\mathbf{e}_i$ is a vector of zeros everywhere except the ith element equal to 1. The value of s_{ik} can range from 0 to 1. The latter value indicates that class Y_k does not overlap with other classes along the axis of the ith feature. Each row of matrix $\mathbf{S} = [s_{ik}]$ can be computed quite efficiently by sorting vector $\mathbf{e}_i^T \mathbf{X}$ and sliding a $(2K+1)$-wide window. This leads to a computational complexity of $\mathcal{O}(N \ln(N) + NK \ln(K))$ for each feature. The following steps need to be realized to obtain the binary classification order:

1. $\mathcal{F}_{\text{kn}} = \mathcal{F}_{\text{in}}$; O is an empty 'first in first out' list (FIFO).
2. Compute $\ell = \max_k s_{ik}$ with $i \in \mathcal{F}_{\text{kn}}$; set $s_{\bullet \ell} = 0$.
3. Push ℓ into O.
4. Insert the indices of the extrinsic features that involve a relationship with the object of class Y_ℓ into $\mathcal{F}_{\text{kn}}^{(t)}$ to obtain $\mathcal{F}_{\text{kn}}^{(t+1)}$.
5. If $\mathcal{F}_{\text{kn}}^{(t+1)} = \mathcal{F}_{\text{in}} \cup \mathcal{F}_{\text{ex}}$, then stop, otherwise go to step 2.

At the end, O contains the sequence of the binary classifiers.

B.3.2 Feature Ranking by Cross-validation

The first method only takes into account the performance of a binary, kNN-like classifier for each class in each feature dimension. The second method is based on cross-validation: a cross-validation is performed at each step of the incremental classification process to select the best binary classifier with respect to the space of currently known features. By doing this, the classification error rate is minimized at each step. The order can be determined with the following steps:

1. $\mathcal{F}_{\text{kn}} = \mathcal{F}_{\text{in}}$. O is an empty FIFO list.
2. Only take into account known feature on the data set : $[x_{ij}]_{i \in \mathcal{F}_{\text{kn}}^{(t)}, 1 \leq j \leq N}$.
3. Split the data set into N groups.
4. For each group:

   (a) Use the data from the $N-1$ other groups as a training set and build a model for each class that are not present in O.
   (b) Measure the model performance on the validation set (data from the selected group).

5. Compute $\ell = \max_k p_k$, where p_k is the mean performance for the binary classifier identifying the class Y_k.
6. Push ℓ into O.
7. Insert the indices of the extrinsic features that involve a relationship with the object of class Y_ℓ into $\mathcal{F}_{\text{kn}}^{(t)}$ to obtain $\mathcal{F}_{\text{kn}}^{(t+1)}$.
8. If $\mathcal{F}_{\text{kn}}^{(t+1)} = \mathcal{F}_{\text{in}} \cup \mathcal{F}_{\text{ex}}$, then stop, otherwise go to step 2.

Like in the first method, O contains the sequence of the binary classifiers.

B.4 Incremental Feature Computation and Classification

The incremental procedure works as follows. First, the centered and normalized training set is stored; $\mathcal{F}_{\text{kn}}$ is initialized at $\mathcal{F}_{\text{in}}$ and the classification order O is computed. Next, the incremental iterations begins:

1. Pop the first element of O : $\ell = \text{pop}(O)$.
2. Train the ℓ^{th} binary classifier on the reduced training set $[x_{ij}]_{i \in \mathcal{F}_{\text{kn}}^{(t)}, 1 \leq j \leq N}$.
3. Give class label Y_ℓ to the objects in the test set having the highest probability to belong this class according to the classifier. At least one super-pixel has to belong to the class Y_ℓ.
4. Compute all extrinsic features that involve a relationship with the object of class Y_ℓ and insert their indices into $\mathcal{F}_{\text{kn}}^{(t)}$ to obtain $\mathcal{F}_{\text{kn}}^{(t+1)}$.
5. If $\mathcal{F}_{\text{kn}}^{(t+1)} = \mathcal{F}_{\text{in}} \cup \mathcal{F}_{\text{ex}}$, then stop, otherwise start a new iteration.

Once a feature order is determined, the classifiers can be trained beforehand to increase the computational efficiency. At the end of the procedure, all features are known, but the classification might not be optimal. Some super-pixel might not be classified, while others can be classified in several classes. A multiclass

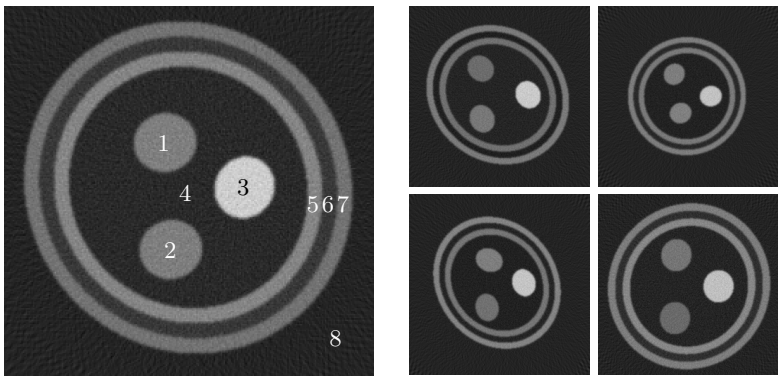


Figure B.1 – Left: Image with the position of the 8 different labels. Right: Some images picked in the data set, showing the variations in shape and position.

classifier can address these issues. The multiclass classifier is trained on the whole training set. The multiclass classifier is fed with all features to obtain the final class label for each super-pixel. This last step gives the object their final class label and slightly improves the results, as shown in the experiments.

B.5 Experiments and Results

As a proof of concept, the principle of incremental classification is illustrated with a simple problem of image segmentation. The data consists of artificial 2D images of crowns and discs encompassing each others, like organs or tissue layers. In each image, there are 8 labels, as shown in Fig. B.1. Noise is added to get realistic images. The position, orientation, size, and color of the depicted objects vary in each image. The disc labeled 3 is the only white circle. The discs and crowns with label 4, 6, and 8 are black, while those labeled 1, 2, 5, 7 are gray.

The data set contains 50 images. Each of them is over-segmented with the watershed-cut algorithm to obtain 20, 30, 40, 50, 75, 100, 125, 150 or 200 super-pixels (see Fig. B.2).

For the data set $\mathbf{X}^w$ (where w is the number of super-pixels), we have $N = 50 * w$. Six intrinsic features are extracted: the luminance, the mass center coordinates, the height, the width, and a binary feature that indicate if the super-pixel touches the border of the image. Three extrinsic features by class are also computed: the signed distance (or offset) to the center of the class and

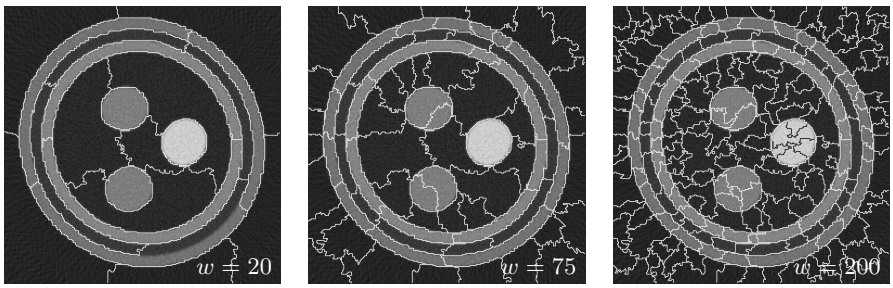


Figure B.2 – Exemple of different watershed segmentations. Left: image with 20 super-pixels. Center: 75 super-pixels. Right: 200 super-pixels.

a binary feature that indicates if the super-pixel is adjacent to the class. At the end, there are 24 extrinsic features. Altogether, there are 30 features.

This data set is randomly split into a training set with 45 images and a test set with the remaining 5 images.. The known features in the test set are centered and normalized by subtraction of the mean and division by the standard deviation computed on the training set. To compute the binary classification order, the whole training set is used for the ranking by nearest neighbors. For the cross-validation method, the training set is split in 10 groups. Each group serves as a validation set during the determination of the order while the rest is used as the training set in the cross-validation process. In both cases, the whole training set is used to build the binary and multiclass models. Our method is implemented with three algorithms: k nearest neighbors (kNN ((Cover and Hart 1967)), support vector machine (SVM (Joachims 1999)) and random forest (RF (Breiman 2001)). For the kNN classifier, we set $k = 3$ (other values give similar results). The SVM uses a Gaussian kernel. The RF grows 500 classification trees. The extrinsic features are inferred for each image individually. During the classification step, if no super-pixel can be identified by the binary classifier, we select the super-pixel that has the highest probability to belong to the class we wish to identify. For the kNN classifier, we choose the super-pixel that has the highest number of neighbors belonging to the considered class. For the SVM, we take the super-pixel with the shortest distance to the classification margin. For the RF, we use the super-pixel that has the highest number of trees classifying it to the considered class. The whole classification procedure is repeated with 20 different training sets.

As the classes are unbalanced, the accuracy is measured with the BCR

(balanced classification rate). The BCR is defined as $BCR = \frac{1}{l}\sum_{i=1}^{l}\frac{T_{Y_i}}{|Y_i|}$, where l denotes the number of class, $|Y_i|$ is the number of pixels that should be labeled Y_i and T_{Y_i} is the number of pixels well classified in class Y_i. The BCR is analysed at the end of the incremental procedure (solid line in Fig. B.3) and just before the final multiclass classification (dotted line in Fig. B.3). The results are compared with an incremental classification where the order is selected randomly (triangles in Fig. B.3) and with a baseline classification using only the intrinsic features (gray line in Fig. B.3).

Figure B.3 shows that SVM classifiers perform rather poorly in our incremental method. With SVMs, the multiclass classification decreases the BCR. Moreover, SVMs combined with the nearest neighbors ranking never goes beyond the baseline (classification with intrinsic features only). Eventually, the BCR also falls down when the number of super-pixels increases. The lower BCR can be explained by the fact that during the classification step we have to identify at least one super-pixel belonging to a class at each iteration. If we cannot find one we select the super-pixel with the smallest distance to classification margin. This measure does not identify a good candidate and the features extracted from the classification are wrong. Those mistakes are propagated during the classification.

Although it is the simplest, the kNN classifier yields better results. The final multiclass classification with the kNN improves the BCR. Indeed, at the end of the incremental process with binary classifiers, some super-pixels may remain unlabeled and the final classification addresses this issue. Nevertheless, this last iteration cannot correct all past classification mistakes. As expected, the incremental kNN classifiers pass the baseline and outperform the random feature order. The standard deviation is smaller with the order based on cross-validation (results not shown).

The results with the RF are as good as those with the kNN. The BCR is even more robust when the number of super-pixels increased.

The classification results for the kNN with the different feature ranking methods are shown in Fig. B.4.

A detailed analysis of the results, image per image, shows that very often all classifiers make identical errors in the same image of the data set. Errors typically happen in images that depicts the objects in unusual configurations, like when they are the biggest, the smallest, the most shifted, the brightest, etc. Results for those images are naturally poorer, since such configurations are seldom instantiated in the data set.

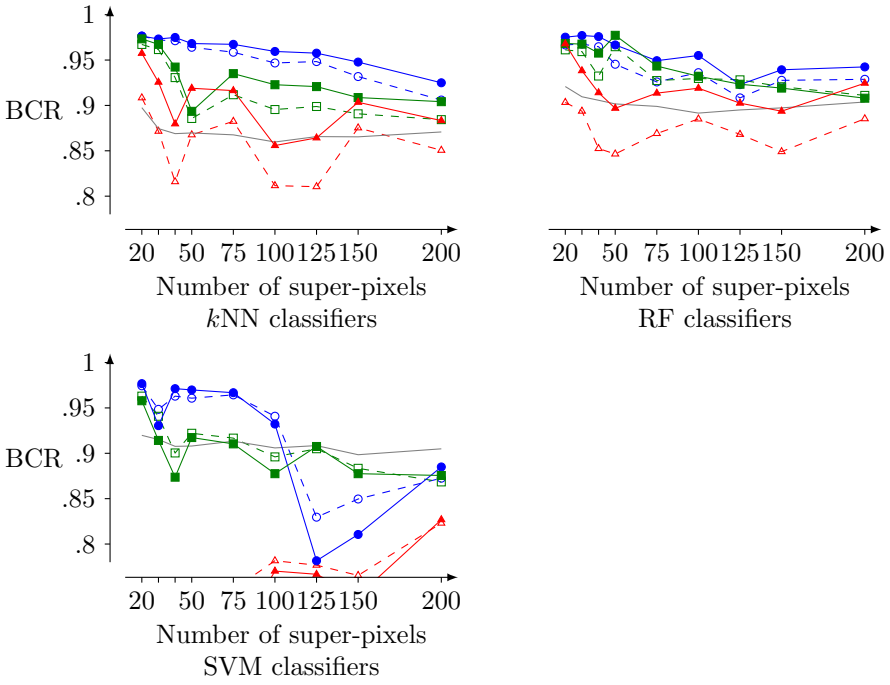


Figure B.3 – BCR with respect to the number of super-pixels, with kNN classifiers (upper left), RF classifiers (upper right) and SVM classifiers (bottom). Circles (●- and -○-) represent the measures with the order by cross-validation. Squares (■- and -□-) represent the measures with the order by nearest neighbors. Triangles (▲- and -△-) represent the measures with a random order of extraction. — is the measure while using only the intrinsic features. Filled lines (●-, ■- and ▲-) are used for the measure after the final multiclass classification. Dashed lines (-○-, -□- and -△-)are used for the measure before the final multiclass classification.

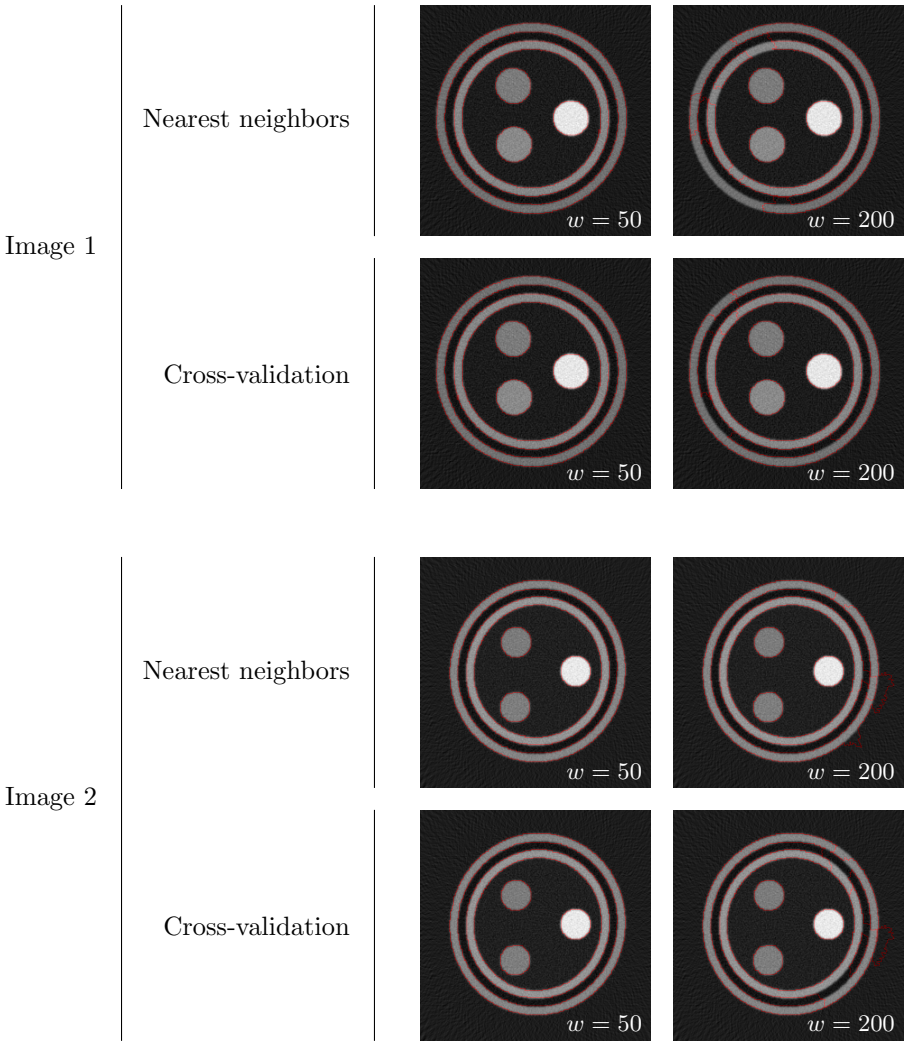


Figure B.4 – Result of the final classification with kNN for 2 different images. In each case, the top images are the resulting images with nearest neighbors and the bottom images are the resulting images with cross-validation. The left images are segmented in 50 super-pixels and the right images are segmented in 200 super-pixels.

We also compared the segmentation and labeling results of our incremental classifier to those obtained with atlas registration. For this purpose, we used MIRT (medical image registration toolbox[1]). For each of the 50 images in the data sets, we picked the one that correlates best, among the 49 others. Next, we non-rigidly registered the latter to match the former, using MIRT 2D. The deformation field is parameterized with B-splines, image similarity is measured with mutual information, and registration goes through 5 hierarchical levels with lower-resolution images. These settings allow nonlinear gray-to-gray mappings to be identified, as well as large deformations to be easily captured. Finally, the deformation field was applied to labels associated with the mobile image, in order to determine the labels on the fixed image. The relative simplicity of the images, the choice of the best-correlating image, and the quite powerful settings prevented any convergence failure. The average BCR reached 0.9672. Our methodology based on incremental classification was thus capable of performing better, with BCR values going up to 0.977 in Fig. B.3.

B.6 Conclusion

This paper describes a procedure for incremental classification with two methods of sequential feature extraction. It can deal with problems where the value of some features requires a partial classification to be already known. The process of incremental classification aims at refining the partial classification in an iterative way. The procedure is generic and can solve the sub-problems in each iteration with various classification techniques (e.g. naives Bayes, kNN, SVM, decision tree, random forest etc.). The final multi-class classifier can be changed as well. Depending on the problem at hand, the procedure must be adapted with appropriate definitions of features and relevance factors. Failure to do so increases the risk of error propagation in the incremental process. Experiments on artificial images show that the procedure is effective. Nevertheless, the results must be reproduced on real images for a full validation of the method.

In the future, we will investigate the possibility of using a single classifier that deals with all intrinsic and extrinsic features at all times, thanks to the use of adaptive relevance factors.

[1] `https://sites.google.com/site/myronenko/research/mirt`

B.7 Bibliography

Beucher, Serge and Christian Lantuéjoul (1979). 'Use of watersheds in contour detection'. In:

Beucher, Serge and Fernand Meyer (1992). 'The morphological approach to segmentation: the watershed transformation'. In: *OPTICAL ENGINEERING-NEW YORK-MARCEL DEKKER INCORPORATED-* 34, pp. 433–433.

Bondiau, P.Y., G. Malandain, S. Chanalet, P.Y. Marcy, J.L. Habrand, F. Fauchon, P. Paquis, A. Courdi, O. Commowick, I. Rutten et al. (2005). 'Atlas-based automatic segmentation of MR images: validation study on the brainstem in radiotherapy context'. In: *International Journal of Radiation Oncology* Biology* Physics* 61.1, pp. 289–298.

Breiman, Leo (2001). 'Random Forests'. English. In: *Machine Learning* 45 (1), pp. 5–32. ISSN: 0885-6125.

Cousty, Jean, Gilles Bertrand, Laurent Najman and Michel Couprie (2009). 'Watershed cuts: Minimum spanning forests and the drop of water principle'. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31.8, pp. 1362–1374.

Cover, Thomas and Peter Hart (1967). 'Nearest neighbor pattern classification'. In: *Information Theory, IEEE Transactions on* 13.1, pp. 21–27.

Gould, S., J. Rodgers, D. Cohen, G. Elidan and D. Koller (2008). 'Multi-class segmentation with relative location prior'. In: *International Journal of Computer Vision* 80.3, pp. 300–316.

Joachims, T. (1999). 'Making large scale SVM learning practical'. In:

Kass, M., A. Witkin and D. Terzopoulos (1988). 'Snakes: Active contour models'. In: *International journal of computer vision* 1.4, pp. 321–331.

Najman, Laurent and Michel Couprie (2006). 'Building the component tree in quasi-linear time'. In: *Image Processing, IEEE Transactions on* 15.11, pp. 3531–3539.

Otsu, Nobuyuki (1975). 'A threshold selection method from gray-level histograms'. In: *Automatica* 11.285-296, pp. 23–27.

Shi, Jianbo and Jitendra Malik (2000). 'Normalized cuts and image segmentation'. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.8, pp. 888–905.

Appendix C

Organ delineation with watersheds and machine learning

Guillaume Bernard, John A. Lee, Michel Verleysen

Abstract accepted at the Belgian Hospital Physicists Association Annual Meeting (Conference) – 2014

C.1 Introduction and purpose

Treatment planning in radiation oncology requires physicians to delineate target volumes and organs at risks. In most cases, this task is still performed by hand, although it is long, repetitive, and time-consuming. For organs at risk, some automatic segmentation tools exist, like atlases, in which pre-segmented images are non-rigidly registered to any new image and organ contours are propagated after deformation. Despite their elegance, atlases suffer from all shortcomings of current deformable registration algorithms. In particular, the regularization of the deformation field is often too simple or unrealistic, leading to small contour inaccuracies that must be corrected for and hence jeopardize the expected gain in time of automatic contouring.

In this work, we propose an alternative to atlases, which is aimed at seg-

menting and recognizing objects and organs in medical images, using watershed transforms and machine learning techniques. With this method, potential segmentation errors are easier to correct.

C.2 Material and methods

The method we propose addresses the problem of organ segmentation and recognition in two steps. First, the image is divided into small, connected regions with nearly uniform intensities, thanks to a watershed transform (Cousty et al. 2009). Each watershed is then considered as an elementary piece (or super-pixel) that has to be labeled in the second step with the appropriate organ tag or membership. Our approach consists in solving this recognition problem with classification algorithms. In machine learning, classification is the process of determining the class of an unlabeled object starting from examples for which the class is known. Therefore, classification problems are typically solved in two steps: first, a model is learned and, second, the model is used to predict the class of new data.

In the considered case, data consists of super-pixels, which can each be characterized with several features or attributes. We distinguish two kinds of attributes: intrinsic ones describe the super-pixel itself (size, position, mean intensity, etc.), while extrinsic attributes characterize its relationships with objects (distance to an already identified organ, contiguity to this organ, etc.). All these attributes can be fed into a classifier. The learned model can then be applied to a new image. However, in such a new image, no label is known and hence extrinsic attributes cannot be evaluated, while they convey much richer information than intrinsic ones. Our solution to this problem was to develop an incremental classification procedure, starting from the intrinsic attributes only and progressively extending the set of extrinsic attributes as soon as organs are identified. In practice, the whole multiclass classification problem is therefore divided into a series of smaller and simpler binary classification problems, each one of them being dedicated to the identification of a particular object or organ in the image. Two kinds of classifiers were used: k nearest neighbors (kNN) and random forest (RF).

In this preliminary phase, our method has been tested on artificial data. The images mimic a phantom with several inserts and layers that are similar in gray level, making their recognition impossible without using richer information such as extrinsic attributes. See Figure C.1 for an example. To validate the

method, the results of our method are compared with those obtained with an atlas. To measure the error without bias, it is recommended to use the balanced classification rate (BCR) (Helleputte and Dupont 2009). If all pixels are correctly classified, the BCR equals one. If it is null, no pixel is correctly classified.

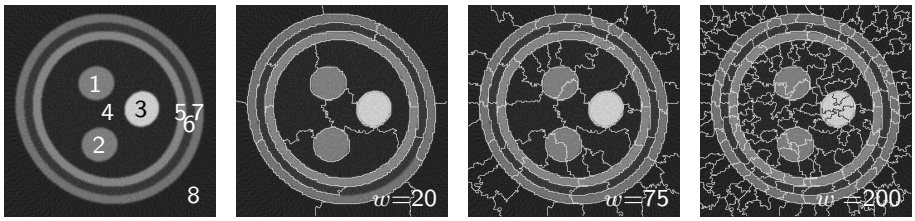


Figure C.1 – Left: One image of the dataset with 8 objects to delineate. Second to last: watershed transform with 3 different granularity presets (20, 75, and 200 watersheds)

C.3 Results and discussion

The sequence of classification is established using different strategies: random order, nearest neighbors, and cross-validation. (More details can be found in (Bernard et al. 2013)). The BCR is also computed with a strategy relying only on the intrinsic features. As we can see on Figure C.2, the results obtained with the incremental classifier are as good as or even better than those obtained with atlas registration. The incremental method with rich extrinsic attributes also improves the naive methods using only the intrinsic features or a random sequence. As the incremental method works with super-pixels (i.e. groups of pixel), it is easier to correct than atlas-based contours (only a label reassignment is needed, vs. redrawing for the atlas). The results depends on the number of superpixels created with the watershed transform. We need to have enough watersheds/superpixels to ensure that all pixels in a watershed consistently belong to a single object/organ. Figure C.2 shows that the BCR decreases if the number of watersheds is too big.

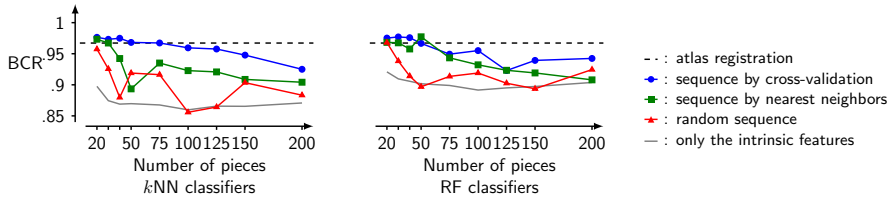


Figure C.2 – Left: One image of the dataset with 8 objects to delineate. Second to last: watershed transform with 3 different granularity presets (20, 75, and 200 watersheds)

C.4 Conclusions

Organ delineation and recognition is feasible with other approaches than registration with an atlas. The proposed method uses machine learning techniques to extract knowledge from previously delineated images to segment new ones. By decomposing the global classification problem into an incremental process, the proposed method segments and labels artificial images, with results that are competitive with atlas registration. Ongoing work aims at testing the presented methodology on real images (MR feet images).

C.5 Bibliography

Bernard, Guillaume, Michel Verleysen and John A Lee (2013). 'Segmentation with Incremental Classifiers'. In: *Image Analysis and Processing–ICIAP 2013*. Springer, pp. 81–90.

Cousty, Jean, Gilles Bertrand, Laurent Najman and Michel Couprie (2009). 'Watershed cuts: Minimum spanning forests and the drop of water principle'. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31.8, pp. 1362–1374.

Helleputte, Thibault and Pierre Dupont (2009). 'Partially supervised feature selection with regularized linear models'. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. Montreal, Quebec, Canada: ACM, pp. 409–416.

Appendix D

Automatic organ at risk delineation with machine learning techniques

Guillaume Bernard, John A. Lee, Michel Verleysen

Abstract and supplementary document accepted.
Abstract published in the proceedings of the American Association of Physicists in Medicine Annual Meeting (Conference) – 2014

D.1 Abstract

Purpose

Manual delineation of organs at risk (OARs) on CT images consumes much time. Automatic segmentation methods like atlases partly address this issue. However, atlases depend on deformable registration quality. This work proposes an atlas-like method that relies on machine learning techniques instead of registration.

Methods

First, a watershed algorithm segments filtered CT images into superpixels (images patches with similar intensity pixels). Next, two kinds of superpixel features are computed: intrinsic ones (known at all times, like superpixel size, position, and mean intensity) and extrinsic ones (to be inferred from partial delineation results, like the distances to other organs). To build the atlas, a binary classifier is associated with each organ. Training optimizes the classifiers' parameters as well as their sequence, to make the most useful extrinsic features available as soon as possible. After training, the sequence of binary classifiers can process any new image, tagging all superpixels incrementally with an OAR label.

The method was applied to 2D CT images of 49 breast-cancer patients (axial slice passing through the 7th thoracic vertebra). The balanced classification rate (BCR) measures the method's accuracy, by giving the percentage of correctly classified pixels per label.

Results

The proposed incremental method (BCR = 94%) is compared to two similar classification procedures, with either no extrinsic features (blind, BCR = 84%) or all of them known beforehand (cheating oracle, BCR = 98%). A preliminary comparison with a registration-based atlas on synthetic data led to 97% (registration) and 98% (proposed).

Conclusion

This abstract demonstrates the feasibility of atlas-like OAR delineation based on machine learning techniques instead of deformable registration. The proposed method relies on incremental classification (partial classification allows additional highly informative features to be inferred). Involving no elastic deformation, delineation can be easily corrected if needed, just by changing the erroneous superpixel labels.

D.2 Supplementary document

Innovation/Impact

This abstract proposes a method for organ segmentation and recognition based on machine learning. This method is a real alternative to traditional atlas based delineation. It relies on well-known classification algorithms such as k-nearest neighbors, support vector machine, random forest, etc. The process is iterative, fast and effective. As it works iteratively, a physicist or a physician can interfere with the delineation during its computation. This would lead to less error propagation and better image delineation. The use of superpixel (group of neighbor pixels of similar intensities) for the delineation makes the correction of the delineation easier than traditional atlas based delineation.

Metric

The number of pixel in each region of interest may vary a lot depending on the region. To correctly measure the error without bias, it is recommended to use the balanced classification rate (BCR). The BCR is defined as $\text{BCR} = \frac{1}{C}\sum_{i=1}^{C}\frac{|T_{Y_i}|}{|Y_i|}$, where C denotes the number of classes, $|Y_i|$ is the number of pixels that should be labeled Y_i and $|T_{Y_i}|$ is the number of pixels well classified in class Y_i.

Key results

The proposed method gives a BCR of 94%. This means that for each organ, on average, 94% of the pixels are correctly classified. The confusion matrix (Fig. D.1) shows the performance of the method. As we can see, the diagonal is very bright meaning that the superpixels are often correctly classified. The figure is more deeply analyzed in its legend.

Figures D.2 and D.3 show some comparisons between the result of the automatic segmentation and the manual segmentation. Figures D.2 illustrate the risk of error propagation while a rib is wrongly classified. In the case of error propagation between ribs, the error is not too problematic as a rib is still labellized as being a rib. Moreover, this kind of error will be reduced when working with 3D images as it will be easier to identify the rib from the vertebra. All those errors are easy to fix by changing the erroneous superpixel labels.

Currently, no post-processing method is used on the results of the automatic segmentation. In the future, some post-processing algorithm can be used to

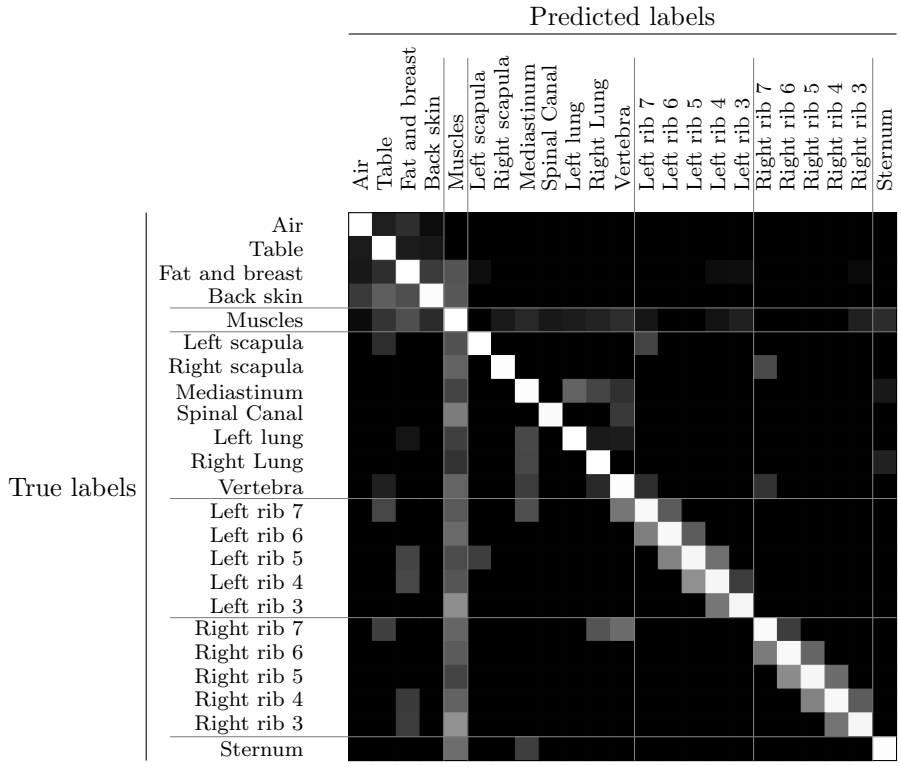
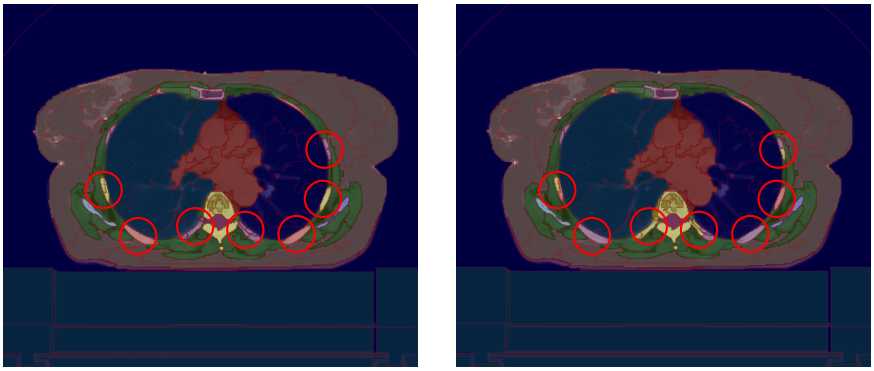


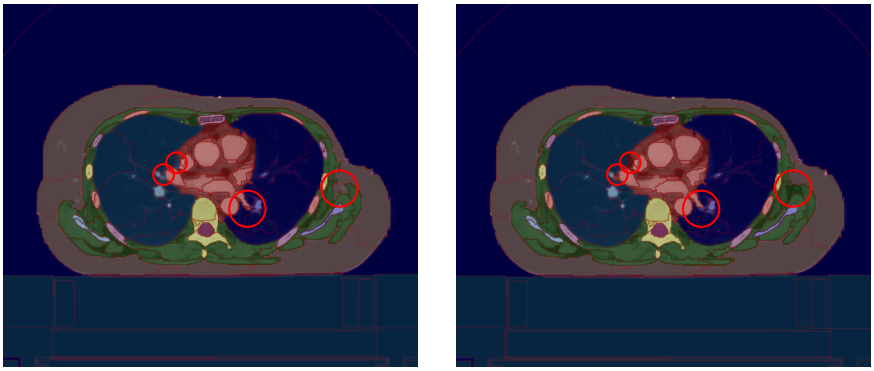
Figure D.1 – Representation of the confusion matrix. Each row was normalized so that it adds up to one. To improve visual contrast among the off-diagonal entries, the fourth root was applied. One can observe that many small portions of organs are sometimes misclassified as being muscle. Similarly, the series of numbered ribs is sometimes shifted by one position, the rib next to the vertebra being then merged with the latter.

identify possible mistake obtain during the delineation. For example, the case of two ribs touching each other comes certainly from a misclassification error.



Manual segmentation Automatic segmentation

Figure D.2 – Illustration of the ribs shifting. The red circles highlight the differences between manual and automatic segmentation. The 7th rib (closest to the vertebra) is labelled as belonging to the vertebra. The adjacent rib (6th rib) is labelled as being the 7th rib. The error propagate to the next ribs.



Manual segmentation Automatic segmentation

Figure D.3 – Illustration of small mistakes difficult to evaluate around the mediastinum. The red circles highlight the differences between manual and automatic segmentation. The border of the mediastinum area is different in 3 places. It cannot truly be said that the result obtained by automatic segmentation is worse than the manual one. It can be seen that a superpixel mainly containing fat is labelled as being a muscle. As the defined region *muscle* is very heterogeneous, superpixel that are surrounded by *muscle* are often labelled as being *muscle*.

Appendix E

Incremental classification of objects in scenes: application to the delineation of images

Guillaume Bernard, John A. Lee, Michel Verleysen

Paper accepted for publication in Neurocomputing (Journal) – 2014

Abstract

Usual multi-class classification techniques often rely on the availability of all relevant features. In practice, however, this requirement restricts the type of features that can be considered. Features whose value depends on some partial, intermediate classification results, can convey precious information but their nature hinders their use. A typical example is the identification of objects in a scene, where the distance from some yet unclassified object to some other that would already be identified earlier in the process. This paper proposes a generic method that solves classification problems involving such features in an incremental way. It proceeds by decomposing the multi-class problem into a sequence of simpler binary problems. Once a binary classifier gives an object

its class tag, all features depending on this object are computed and appended to the list of known features. Experiments with both synthetic and real data, comprised of tomographic images, show that the proposed method is effective.

E.1 Introduction

Object recognition in images is a long-standing problem in the computer vision literature. It entails both segmentation and classification aspects. The proposed methods greatly vary, depending on the problem at hand and the data specificities. For instance, one way to segment an object consists in identifying its pieces such as proposed in Mohan et al. 2001. In particular, the authors detect bodies in natural images by locating arms, heads, and legs. The method is effective but it aims at detection rather than actual segmentation (it yields a rectangular window encompassing the sought object). Several other methods provide tighter object contours and exploit image properties and features (Gould et al. 2009; Ion et al. 2011; Kuettel et al. 2012). These methods achieve both segmentation and recognition (they label the image segments). Most of them are intended to work with natural images. In Gould et al. 2009, a region-based energy functional is defined by individual segment potentials and inter-segment potentials. A two-steps hill climbing procedure minimizes the energy. The first step consists in giving a label to a group of pixel. The second step optimizes the region shape and updates its properties. The two alternate steps are repeated until convergence to a local minima of the energy. In Ion et al. 2011, the authors proposed to extract a bag of segments. Those segments are extracted at different scales and locations. Non-overlapping segments are used to build tilings of the image (graphs that connects adjacent segments). The segmentation and labels of a new image are based on parameters learnt from previously labeled images. The most probable tiling is selected and its associated labels are then copied and attached to the tiles of the new image. In Chen et al. 2014, the tiling of different images from the same scene is used to make a co-labelling of the image. All the images are jointly annotated, thereby giving more consistent values. In computer vision, labels are often given to (groups of) pixels but sometimes the whole image can also be labeled. This approach is investigated in Kuettel et al. 2012. Richer label information is expected to improve the results but such a method has higher requirements for the data collection process. Another method to improve the annotation of an image is label propagation, where the obtained labels are corrected by propagation. An interesting use of label

propagation can also be seen in Kazmar et al. 2013. To be able to identify several drosophila embryo stages on one image, the authors use the shape of the embryos to get a first label. Then, they use a similarity measure based on patterns in the embryos to make the label propagation.

This paper follows a different approach, which is intended to solve a very specific problem. The main assumption is that data consists of similar scenes, all including the very same set of objects. Medical image segmentation, for example, enters within this framework: all patients are imaged following mostly the same protocol, share the same anatomy, but differ in their size, weight, and morphology. In order to interpret medical images, physicians proceed step by step. They typically start by using the little available information to label a few first organs. By doing so, they can deduce new pieces of information, which were initially unavailable, allowing them to recognize new organs, and so forth.

This paper formalizes such an incremental classification process and provides two slightly different methods of solving it, both inspired by a 'divide and conquer' approach. They consist in solving a succession of usual, binary classification problems, which are fed with the available features at the time of their respective execution. The two proposed methods actually differ in the way they sequence the ordinary classifiers. As a proof of concept, the two methods are used to solve object recognition problems involving synthetic and real tomographic images.

The remainder of this paper is organized as follows. Section E.2 formalizes the problem of incremental classification. It also defines the various terms and symbols used throughout this paper. Section E.3 describes the proposed method of decomposing incremental multi-class problems into a sequence of simpler binary problems. In particular, it details two different ways to determine the sequence of these subproblems. Section E.4 presents the experiments and their results. Eventually, Section E.5 draws the conclusions and sketches some perspectives for future work.

E.2 Incremental classification: formalization of the problem

Classification is the task of labeling objects or data items, according to known features, and based on previously seen examples. A typical classification problem includes a learning set (examples of data items for which the class label is known) and a query (a set of instances for which the label has to be attributed). Both

the learning set and the query consist of feature vectors, that are supposed to be drawn from the same distribution, so that the learning set is representative of the query. All features are supposed to be known at the time of resolving the query. If this assumption has the merit to frame the classification problem, it can be constraining in practice. As an example, let us consider our visual system, when it analyses a scene. Our brain is able to recognize and tag objects of the scene, but not all features of the objects are known from the beginning. In particular, our eyes use complex features such as the spatial or geometrical relationships between the objects in the scene. Such features are not known if one of the objects taking part in the relationship has not been labeled yet. Therefore, solving the whole problem requires not only information collection, but also some reasoning: simpler subproblems must be solved in an incremental way to progressively build the missing pieces of information.

For instance, a child who does a jigsaw puzzle solves such a problem. Another example is a physician who looks at a radiographic or tomographic image. The diagnostic depends on the interpretation of the image and therefore on the sequential recognition of the depicted organs.

All these problems share several characteristics. Most of them are visual problems, in which objects must be recognized or differentiated. Sometimes the objects are intrinsically dissimilar, making the solution obvious. Sometimes the objects bear some confounding similarity and differentiating them requires additional, extrinsic information coming from their environment and their relationship with other objects.

In order to formalize the problem, a few terms are defined hereafter.

Scene ($\mathcal{S}$) A scene is a picture of objects living in a N-dimensional space. The scene can be encoded in various ways. Here we assume that an image of the scene is available, either as a projection (like a 2D picture taken by a digital still camera) or a full 3D image (like tomographic images in medical imaging). As an example, let us consider a scene composed of three objects: a green bike, a red ball, and the ground (see Fig. E.1).

Objects (set $\mathcal{O}$ composed of o_j) The objects are the main elements of the scene. All the objects of the scene must be identified at the end of the iterative process. In the example, they are the bike, the ball, and the ground. In practice, to determine the border of the object we use segmentation algorithms Lim and Lee 1990; Shi and Malik 2000; Felzenszwalb and Huttenlocher 2004; Cousty et al. 2009. Due to noise of varying illu-

mination, those algorithms tend to over-segment the image. The object are therefore fragmented into several pieces.

Pieces (set $\mathcal{P}$ composed of p_i) The pieces are the over-segmented parts of the objects. $\mathcal{P}$ includes all pieces of the segmented image of the scene. In the example, the bike can be split in several pieces: two wheels, a frame, a handlebar, a saddle... The ball is quite uniform so it would only be composed of one piece. The ground can be split in two pieces: a piece is illuminated by sunlight while the other is shadowed.

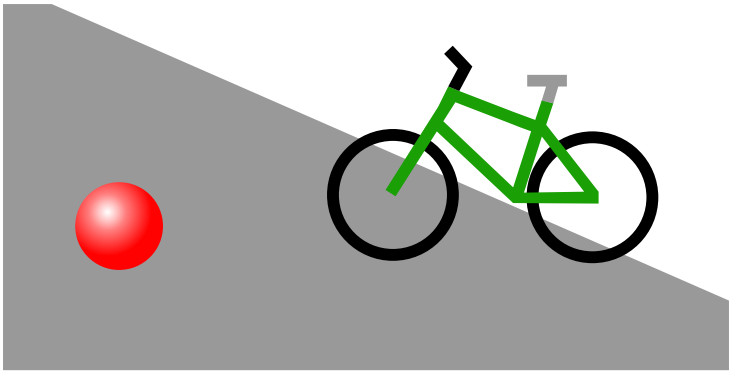


Figure E.1 – Example of scene, composed of a red ball, a green bike, and partially shadowed ground.

***O*-features and *P*-features** Pieces and objects have several features allowing their identification. In the example, the pieces have a color: the frame of the bike is green, the ball is red, the saddle and the ground in the shadow are both gray. The size of the objects (and pieces) can also be used as features. For this paper, $\mathcal{O}$-features refer to the features of an object and $\mathcal{P}$-features refers to the features of a piece. The features can be of any type: Boolean, real, categorical, etc. The objects, pieces, and their respective features are represented in Fig. E.2.

Those elements and features do not suffice to describe a scene. The relationships between pieces and objects must be defined as well.

Labels ($\mathcal{L}$: set of $\mathcal{L}$-edges) The labels give the membership of a piece to an object. In the example, the saddle belongs to the bike, the shadowed

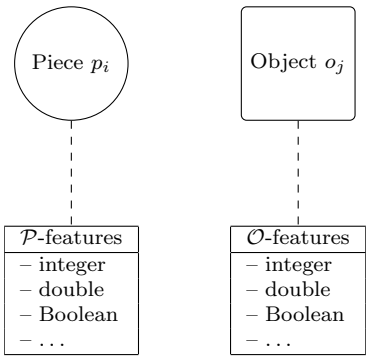


Figure E.2 – Pieces and objects are the two main elements of the scene. They are represented with there respective features.

ground belongs to the ground and the piece 'ball' belongs to the object 'ball.' The set of $\mathcal{L}$-edges is called $\mathcal{L}$ in this paper. An $\mathcal{L}$-edge is written $\mathcal{L}(p_i, o_j)$ to represent the membership of piece p_i into object o_j. If $\mathcal{L}(p_i, o_j) \in \mathcal{L}$, piece p_i has label o_j.

Piece-to-object relation ($\mathcal{PO}$: set of $\mathcal{PO}$-edges) In addition to the membership relations given by the labels, other features can be defined; those features are called $\mathcal{PO}$-features In the example, each piece of the bike (frame, saddle, etc.) has edges connecting to the ground, the ball but also the bike itself. There might be several features characterizing the relation between a piece p_i and an object o_j, The $\mathcal{PO}$-features of these edges are in the example the distance between the piece and the object, the gray-scale color difference, etc. These features are the numerical attributes of the specific $\mathcal{PO}$-edges between p_i and o_j The graph generated by the pieces, the objects and $\mathcal{PO}$ is a complete bipartite graph.

Geometry ($\mathcal{PP}$: set of $\mathcal{PP}$-edges) The scene has a geometry which is represented by the edges connecting the pieces. Each piece is linked to all other pieces. Those edges have numerical attributes called $\mathcal{PP}$-features characterizing the relation between two pieces. In the example, we have a Boolean $\mathcal{PP}$-feature corresponding to the contiguity between two pieces. The bike frame is contiguous to other of its constituting pieces like the saddle, the handlebar, and the wheels. The graph generated by the pieces and edge set $\mathcal{PP}$ is a complete graph.

A complete scene with two objects and two pieces is represented in Fig. E.3.

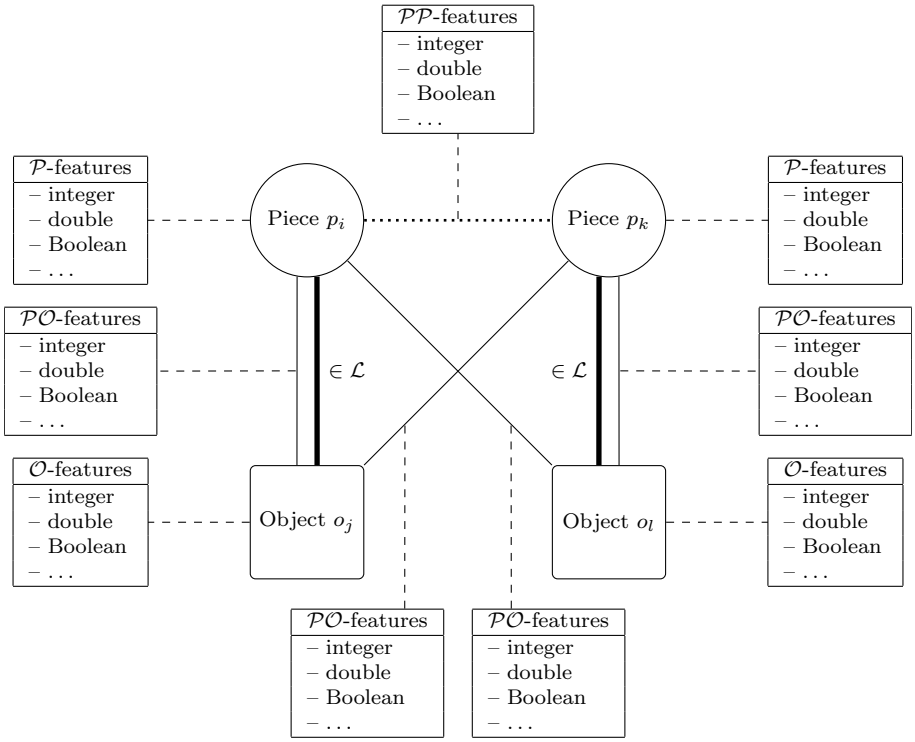


Figure E.3 – A scene composed of two pieces (p_i,p_k) and two objects (o_j,o_l). All pieces (resp. objects) have there own $\mathcal{P}$-features (resp. $\mathcal{O}$-features). The bold links represent the membership of a piece to an object. In this case, $\mathcal{L}(p_i, o_j) \in \mathcal{L}$ and $\mathcal{L}(p_k, o_l) \in \mathcal{L}$. Each piece is connected to all the objects. Those links are characterized by the $\mathcal{PO}$-features. The pieces are linked together and characterized by the $\mathcal{PP}$-features.

Within this framework, our classification problem can be stated as follows. Knowing a few similar scenes containing all the same objects in which all pieces are labeled, we must determine the piece labels in a new unknown scene. An unknown scene is a scene where we only have the objects $\mathcal{O}$ (common to all the scenes), the pieces $\mathcal{P}^{\text{query}}$ and the geometry $\mathcal{PP}$-edges$^{\text{query}}$. For the previously used example, the objects composing the scene are known (a bike, a ball, the ground). The pieces are the result of an unsupervised segmentation. The

arrangement of the pieces in the image, revealed by the contiguity indicator in the example, determines the geometry.

The training set is composed of several scenes. The n^{th} scene in the training set is written $\mathcal{S}^n = (\mathcal{O}^n, \mathcal{P}^n, \mathcal{PO}^n, \mathcal{PP}^n, \mathcal{L}^n)$. As all the objects are the same for all the scenes, we can replace $\mathcal{O}^n$ with $\mathcal{O}$. The query scene is defined by $\mathcal{S}^{\text{query}} = (\mathcal{O}, \mathcal{P}^{\text{query}}, \emptyset, \mathcal{PP}^{\text{query}}, \emptyset)$. All the $\mathcal{PO}$-features are unknown because we do not have any information on the object in the image. We know the objects are present but at the beginning of the process, we do not know anything about the values of their $\mathcal{O}$-features. Conversely, the $\mathcal{PP}$-features are computed from the segmentation and will never change during the classification process.

This paper describes a solution that proceeds incrementally to solve the classification problem. First, the $\mathcal{P}$-features are used to identify the first object in the scene. For example, we identify that the piece 'ball' belongs to the object 'ball' because this piece is red. So, $\mathcal{L}$(piece 'ball', object 'ball') is in $\mathcal{L}$. Next, $\mathcal{PO}$-edges linking to the discovered object are added to set $\mathcal{PO}$ and the value of the $\mathcal{PO}$-features related to the identified object are computed (sometimes with the help of the $\mathcal{PP}$-features). In the example, the distance between each piece and the object 'ball' is computed. $\mathcal{P}$-features and the newly computed $\mathcal{PO}$-features are used to identify a second object. In the example, the ground can be identified because the shadowed ground touches the ball. By repeating these simple identification steps, information about $\mathcal{PO}$-features is accumulated, allowing more objects to be labeled.

E.3 Iterative feature building and classification

Our approach to solve an incremental classification problem relies on a generic method that performs a sequence of classification tasks. In this Section, the generic method is firstly presented. Next, various methods that determine the classification sequence are explained.

E.3.1 Incremental $\mathcal{PO}$-feature computation and classification

The incremental procedure works as follows. The training set is

$$\{\mathcal{S}^n = (\mathcal{O}, \mathcal{P}^n, \mathcal{PO}^n, \mathcal{PP}^n, \mathcal{L}^n)\} \quad .$$

All the $\mathcal{PO}$-features and all $\mathcal{PP}$-features are known.

The query is

$$\mathcal{S}^{\text{query}} = (\mathcal{O}, \mathcal{P}^{\text{query}}, \emptyset, \mathcal{PP}^{\text{query}}, \emptyset) \ .$$

All the $\mathcal{PP}^{\text{query}}$-features are known. The $\mathcal{PO}^{\text{query}}$-features are completely unknown because no object has been identified in the scene yet.

Starting from the training set, the classification sequence σ is computed (this part is further developed in Subsection E.3.2). This sequence determines in which order the objects will be identified. For example, based on the bike and ball example used in the previous section, we can fix a sequence of classification 'bike, ground, ball.' Secondly, the $\mathcal{PO}^n$ are initialized to empty sets. Next, the incremental iterations starts. Each iteration can be divided in two parts (see Algorithm 5). The first part (lines 3 to 13) consists in identifying a new object based on the already known classes. The second part (lines 15 to 24) is a loop where the $\mathcal{PO}^{\text{query}}$-features are used to refine the classification and to update the values of the $\mathcal{PO}^{\text{query}}$-features. This loop stops when the binary classification is stable.

Sequence σ has initially a finite size and contains each object once. When σ is empty, each object has been identified and each $\mathcal{PO}$-feature has been computed. Fixing a limit to the number of times the second part of the algorithm (lines 15 to 24) can be repeated ensures that the algorithm ends.

Once the classification sequence is determined, the classifiers can be trained beforehand to reduce the computation time needed for each query. At the end of the procedure, all $\mathcal{PO}$-features are known, but the classification might not be optimal. Some pieces might not be classified. Others can be classified in several classes. The use of an ordinary multiclass classifier after the iterative procedure can address these issues. The multiclass classifier can be trained over the whole training set. By using this classifier on the query scene with all the $\mathcal{P}$-features and the $\mathcal{PO}$-features , it ensures that each piece is associated with a single object.

E.3.2 Classification sequence

To be able to use the algorithm proposed in Section E.3.1, the best classification sequence needs to be determined. At each step, the ideal situation would be to minimize the number of misclassified pieces to avoid the propagation of wrong class labels and hence the computation of wrong $\mathcal{PO}$-features.

Two methods for determining the sequence of classification are proposed. The first will not necessarily guarantee a decrease of the number of misclassified

Algorithm 5

Require: σ, $\{\mathcal{S}^n\}$, $\mathcal{S}^{\text{query}}$

1: $\{\mathcal{S}^n\} \leftarrow \{\mathcal{S}^n\} \setminus \{\mathcal{PO}^n\}$ ▷ $\{\mathcal{PO}^n\}$ is removed as there is no $\mathcal{PO}$ in $\mathcal{S}^{\text{query}}$ $\mathcal{PO}$ are iteratively added to $\{\mathcal{S}^n\}$ at line 12
2: **while** σ is not empty **do**
3: $o_\ell \leftarrow \text{pop}(\sigma)$ ▷ o_ℓ is the object to be identified
4: $\mathcal{C}_\ell \leftarrow \text{Train}(\{\mathcal{S}^n\}, o_\ell)$ ▷ $\mathcal{C}_\ell$ is a binary classifier identifying pieces of o_ℓ At the first iteration only $\mathcal{P}$-features are used
5: $\{p_i\} \leftarrow \mathcal{C}_\ell(\mathcal{S}^{\text{query}})$ ▷ $\mathcal{C}_\ell$ identifies pieces belonging to o_ℓ
6: **for all** $p_j \in \{p_i\}$ **do** ▷ Add label for all the pieces identified
7: $\mathcal{L}^{\text{query}} \leftarrow \mathcal{L}^{\text{query}} \cup \mathcal{L}(p_j, o_\ell)$
8: **end for**
9: Add $\mathcal{PO}$-edges between o_ℓ and all the pieces of $\mathcal{S}^{\text{query}}$
10: Compute the values of the $\mathcal{PO}$-features of the $\mathcal{PO}$-edges linked to o_ℓ
11: **for all** $\mathcal{S}^m \in \{\mathcal{S}^n\}$ **do** ▷ Add the $\mathcal{PO}$-edges linked to o_ℓ in each $\mathcal{PO}^n$
12: $\mathcal{S}^m \leftarrow \mathcal{S}^m \cup \mathcal{PO}^m_{o_\ell}$ ▷ The $\mathcal{PO}$-features of the $\mathcal{PO}$-edges are known
13: **end for**
14: $\mathcal{C}_\ell \leftarrow \text{Train}(\{\mathcal{S}^n\}, o_\ell)$ ▷ Use the updated training set to update $\mathcal{C}_\ell$
15: **while** $\mathcal{L}^{\text{query}}$ is changing **do** ▷ Iterate until the set of pieces belonging to o_ℓ do not change
16: $\{p_i\} \leftarrow \mathcal{C}_\ell(\mathcal{S}^{\text{query}})$
17: **for all** $p_j \in \{p_i\}$ **do** ▷ Add labels to the newly identified pieces
18: $\mathcal{L}^{\text{query}} \leftarrow \mathcal{L}^{\text{query}} \cup \mathcal{L}(p_j, o_\ell)$
19: **end for**
20: **for all** $p_k : (p_k \notin \{p_i\}$ and $\mathcal{L}(p_k, o_\ell) \in \mathcal{L}^{\text{query}})$ **do**
21: $\mathcal{L}^{\text{query}} \leftarrow \mathcal{L}^{\text{query}} \setminus \mathcal{L}(p_k, o_\ell)$ ▷ Remove previously detected labels
22: **end for**
23: Update the values of the $\mathcal{PO}$-features of the $\mathcal{PO}$-edges linked to o_ℓ in $\mathcal{S}^{\text{query}}$.
24: **end while**
25: **end while**
26: $\mathcal{C} \leftarrow \text{Train}(\{\mathcal{S}^n\}, \mathcal{O})$ ▷ Train a multiclass classifier from the complete data set
27: $\mathcal{L}^{\text{query}} = \mathcal{C}(\mathcal{S}^{\text{query}})$ ▷ Get the final labels for each piece

pieces, but is computationally affordable. The second is built to minimize the number of misclassified pieces. For each method, a computational time complexity is provided.

Sequence by nearest neighbors

The usefulness of a feature to identify a given object depends on its marginal distribution. Given a single feature, if a class does not significantly overlap the others, then this feature may be considered as useful to identify that class. As a first option, this paper suggests a ranking that assesses the overlap of classes if all pieces were characterized by only one $\mathcal{P}$-feature or $\mathcal{PO}$-feature. In the example of Section E.2, this method evaluates if by using only one feature (color, distance to the bike) we can easily classify an object (the ball, the ground, or the bike)

Let $\mathcal{N}_f^K(p_i)$ denote the set of the K nearest neighbors of piece p_i in the subspace space of feature f alone (f can be a $\mathcal{P}$-feature or $\mathcal{PO}$-feature) of data set $\mathcal{S} = (\mathcal{O}, \mathcal{P}^n, \mathcal{PO}^n, \mathcal{PP}^n, \mathcal{L}^n)$.

Let $\mathcal{P}_{o_c} = \{p_i \text{ s.t. } \mathcal{L}(p_i, o_c) \in \mathcal{L}\}$ be the set of pieces belonging to object o_c. The usefulness of a certain feature to classify pieces inside or outside the object o_c can be measured as the proportion of pieces belonging to object o_c among the K nearest neighbors (regarding only the considered feature) of each pieces of the object o_c. It can be written as

$$s_{fc} = \frac{1}{K|\mathcal{P}_{o_c}|} \sum_{p_i \in \mathcal{P}_{o_c}} \left|\left\{p_j \text{ s.t. } p_j \in \mathcal{N}_f^K(p_i) \text{ and } \mathcal{L}(p_j, o_c) \in \mathcal{L}\right\}\right| ,$$

where $|A|$ denotes the cardinality of set A. The value of s_{fc} can range from 0 to 1. The value s_{fc} indicates how much class o_c stands apart from other classes along the axis of feature f. The bigger s_{fc}, the more feature f can discriminate object o_c.

By representing the values of the $\mathcal{P}$-features and $\mathcal{PO}$-features of all pieces in a matrix $\mathbf{X}$ where rows are pieces and columns are features, each row of matrix $\mathbf{S} = [s_{fc}]$ can be computed quite efficiently by sorting vector $\mathbf{X}\mathbf{e}_f$ and sliding a $(2K+1)$-wide window. Vector $\mathbf{e}_f$ is a vector of zeros everywhere except the fth element that is equal to 1. This leads to a time complexity of $\mathcal{O}(P^2 K \ln(P) \ln(K))$ for each feature, where P is the number of pieces.

In order to obtain the binary classification sequence, the most discernable object, knowing the already computed feature, need to be identified at each step. This can be done by going through Algorithm 6 where $\mathcal{F}_{\text{kn}}$ is the set of known features (it can be $\mathcal{P}$-features and/or $\mathcal{PO}$-features).

As explained previously, at the beginning of the iterative classification, none of the objects is known. Therefore, the only known features are the $\mathcal{P}$-features.

Algorithm 6

Require: σ is an empty FIFO list. $\mathcal{F}_{\text{kn}}$ = set of $\mathcal{P}$-features.
1: **while** σ does not contain all objects **do**
2: $o_\ell \leftarrow \max_c s_{fc}$ with $f \in \mathcal{F}_{\text{kn}}$
3: $s_{\bullet\ell} \leftarrow 0$
4: Push o_ℓ into σ
5: $\mathcal{F}_{\text{kn}} \leftarrow \mathcal{F}_{\text{kn}} \cup \{\mathcal{PO}\text{-features related to object } o_\ell\}$
6: **end while**

The $\mathcal{PO}$-features are added afterwards, when the object to which they are related is known. It is important to start the sequence determination with $\mathcal{F}_{\text{kn}}$ equals to the set of $\mathcal{P}$-features to correctly evaluate the best sequence. At the end, σ contains the sequence of the objects to be identified with the binary classifiers. The time complexity of the whole process is $\mathcal{O}(FP^2K\ln(P)\ln(K) + K^2F)$ where F is the number of features. As FK^2 is smaller than FP^2, we have a time complexity of $\mathcal{O}(FP^2K\ln(P)\ln(K))$.

Sequence by cross-validation

The first method only takes into account the performance level of a binary kNN classifier for each class in each feature dimension. As a second option, this paper suggests a cross-validation at each step of the incremental classification process to select the best binary classifier with respect to the space of currently known features. By doing this, the minimization of classification error rate is ensured at each step. The sequence can be determined from the training set $\{\mathcal{S}^n = (\mathcal{O}, \mathcal{P}^n, \mathcal{PO}^n, \mathcal{PP}^n, \mathcal{L}^n)\}$ with Algorithm 7:

Like in the first method based on the nearest neighbors, σ contains the sequence of the binary classifiers. The drawback of this method is that it is much more time-consuming than the previous one. Let $K = |\mathcal{O}|$. At the first iteration, $K(Q-1)$ classifiers are computed and evaluated. At each iteration, one more class is identified. It leads to a total number of $(Q-1)\frac{K(K+1)}{2}$ classifiers. Therefore, the time complexity is $\mathcal{O}\left(QK^2 f(\mathcal{S})\right)$ where $f(\mathcal{S})$ is the time complexity of building the model and using it with the dataset $\mathcal{S}$.

E.4 Experiments and results

The methods developed in this paper find their motivation in an image segmentation problems encountered in the field of cancer treatment by radiotherapy.

Algorithm 7

Require: σ is an empty FIFO list.
1: $\forall n\ \mathcal{PO}^n = \emptyset$
2: **while** All the objects are not in σ **do**
3: Separate $\{\mathcal{S}^n\}$ into Q groups.
4: **for all** group $\{\mathcal{S}^q\}$ **do**
5: Use $\{\mathcal{S}^n\} \setminus \{\mathcal{S}^q\}$ as a training set and build a binary model for each object that is not present in σ.
6: Measure the model performance on the validation set ($\{\mathcal{S}^q\}$).
7: **end for**
8: Compute $o_\ell = \max_{o_c} p_{o_c}$, where p_{o_c} is the mean performance for the binary classifier identifying the object o_c.
9: Push o_ℓ into σ
10: Add the $\mathcal{PO}$-edges related to o_ℓ in $\mathcal{PO}^n$.
11: **end while**

To minimize side effects of the treatment, radiation oncologists try to maximize the irradiation of the tumor while avoiding organs at risk as much as possible. For this purpose, they rely on X-ray tomographic images, which reveal the three-dimensional morphology of the patient. Next, they manually delineate the tumor and the organs at risk. Starting from medical constraints that are specific to each of the delineated organs, physicists then determine the safest beam configuration to irradiate the tumor.

Several guidelines for the manual delineation of the organs at risk exist Wijers et al. 1999; V. Grégoire et al. 2000; Levendag et al. 2004; Vincent Grégoire et al. 2003. However, the task is long, repetitive, and subject to intra- and inter-expert variability (results can differ for the same experts repeating the delineation or across experts). For a given type of tumor, the X-ray images bear some structural similarity, in the sense that all include the same organs, with variations limited mainly to size or shape. In this perspective, the X-ray images are scenes and the depicted organs are the objects found in every scene.

As a proof-of-concept, the proposed incremental classification methods are applied to both synthetic and real tomographic images.

The metrics, datasets, and experimental protocols are explained in Subsections E.4.1, E.4.2, and E.4.3, respectively. Finally, the result are shown in Subsection E.4.4.

E.4.1 Metrics

When working with images, the number of pieces associated with an object may vary a lot depending on the object. To correctly measure the error without bias, it is recommended to use the balanced classification rate (BCR) Helleputte and Dupont 2009. The BCR is defined as $\text{BCR} = \frac{1}{C}\sum_{i=1}^{C}\frac{|T_{Y_i}|}{|Y_i|}$, where C denotes the number of classes, $|Y_i|$ is the number of pixels that should be labeled Y_i and $|T_{Y_i}|$ is the number of pixels well classified in class Y_i. In the case of multiple labels, a pixel classified in n classes counts for $\frac{1}{n}$ in $|T_{Y_i}|$ if its true label is Y_i.

In the experiments, the classification sequence of the objects in the scene may vary a lot. In order to assess this variation, the Levenshtein distance Levenshtein 1966 is used. Also known as edit distance, it computes the distance between two sequences (minimal number of insertion, deletion and substitution). The distances for all pairs of sequences that are obtained with the different training sets are computed. The bigger the average distance, the less stable the method is.

E.4.2 The datasets

The methods were applied on both artificial and real images. The first dataset is synthetic and simulates real tomographic images (noise is added in the projection space, before reconstruction with the inverse Radon transform). Each image depicts a scene that include several organ-like objects, shaped as disks and crowns (see Fig. E.4). Difficulty comes from the fact that many objects share the same gray level. Each image is segmented into several pieces with a watershed algorithm Beucher and Lantuéjoul 1979; Beucher and Meyer 1992; Cousty et al. 2009. The watersheds are sets of contiguous pixels and correspond to the catchment basins in the gradient-magnitude image. Therefore, all pixels in a piece have quite similar gray levels. In order to improve robustness to noise, the method proposed by Najman and Croupie Najman and Couprie 2006 was used to fix the number of watersheds beforehand (100 in this dataset). If this number is large enough, then each object in the scene is spread over at least one watershed/piece. Conversely, each watershed/piece belongs to a single object.

The nine $\mathcal{P}$-features of the pieces are their mean intensity, their minimum and maximum coordinates along the x and y axes, their size along the x and y axes, their surface, and a binary flag indicating whether they are contiguous to at least one of the image borders. The three $\mathcal{PO}$-features are the contiguity with an object and the distances along the x and y axes between the piece and

object centers. The three $\mathcal{PP}$-features are the center-to-center distance and the contiguity between the two pieces. One hundred scenes were generated. The $\mathcal{PO}$-features are computed from the $\mathcal{PP}$-features. The contiguity of a piece to a known object is established by evaluating the contiguity of the piece to all pieces composing the object. If the piece is contiguous to at least one piece in the object, it is contiguous to the object. To compute the distance from the center of a piece to the center of an object, a weighted sum of the distances between the piece and the pieces composing the object is computed. Each distance is weighted relatively to the number of pixels in the piece.

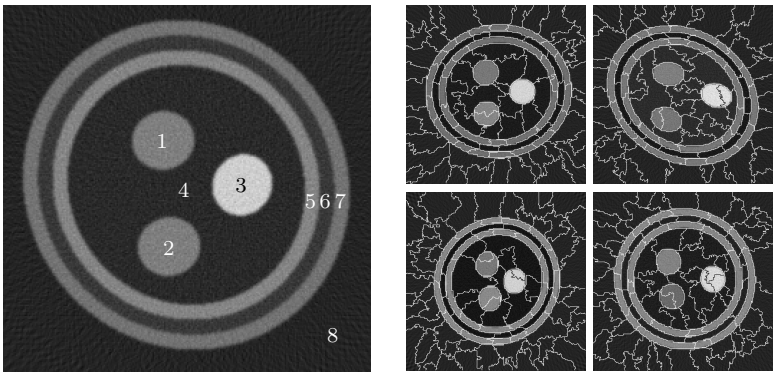


Figure E.4 – Examples of image used in the synthetic dataset. The object labels are indicated on left side. The border of the watersheds are in white (black for the clearest area).

The second and third datasets consist of real tomographic images. Their only difference is the number of watersheds (200 and 400). Forty nine 3D X-ray computed tomography images were collected in the radiotherapy service of the Saint-Luc university hospital (Brussels, Belgium). All these images were part of the routine protocol for female patients treated by radiotherapy after breast cancer surgery. All images were acquired on a Toshiba Aquilion LB CT scanner with varying slice thicknesses (2, 3, or 5 mm). In each tomographic acquisition, an axial slice was selected and extracted at the level of the seventh thoracic vertebra. Each slice contains 512^2 square pixels with edge length equal to 1.074 mm. The luminance of each pixel ranges from -2048 to 2048 Hounsfield unit [HU] and indicates the radiodensity of the depicted material.

The 49 2D slice were then preprocessed with a total variation filter Rudin et al. 1992; Chambolle 2004 in order to reduce the sensitivity of the watershed

algorithm to noisy textures. This filter can attenuate noise while preserving salient edges. The weight of the total variation regularisation term was manually adjusted in order to significantly smooth noise and textures, while avoiding a cartoon-like result.

A second preprocessing step aims at reinforcing the luminance contrast in soft tissues (muscle, fat, etc.). For this purpose, a continuous, monotonically growing, and piecewise linear transformation was applied the all luminance values. This allowed us to shrink unimportant segments of the luminance histogram, while stretching the most relevant ones for the problem at hand.

After preprocessing, the tomographic slices were segmented like the artificial images, with the same watershed algorithm but larger numbers of watersheds (200 and 400). Each scene/slice includes 23 objects that are shown, reported and commented in Figure E.5 and Table E.1. Each segmented piece is given a single label, determined by the object covering all (or most) of the piece.

The eleven $\mathcal{P}$-features of the pieces are their average luminance, their center along the x and y axes, their minimum and maximum coordinate along the x and y axes, their size along the x and y axes, their surface, and their contiguity with an image border. The three $\mathcal{PO}$-features are the contiguity and the center-to-center distances along the x and y axes. The three $\mathcal{PP}$-features are the center-to-center distance and the contiguity.

Key information about the datasets are summarized in Table E.2.

E.4.3 Experimental protocol

The following operations were repeated 50 times for each dataset. The dataset is split in two parts. Ninety percent of the scenes are included in the training set, whereas the remaining 10% form the test set. The training set determines the classification sequence. In the case of the cross-validation approach, the training set is further divided into in 5 equal parts (5-fold cross-validation). Cross-validation involves a grid-search method to adjust the parameters of the binary classifiers. After cross-validation, the final binary classifiers are built with the whole training set. As to the multiclass classifier, its parameters are tuned by cross-validation as well.

Incremental classification was tested with the two proposed methods of sequencing (nearest neighbors and cross-validation), as well as with random sequences. It was also compared to blind non-incremental classification involving either the $\mathcal{P}$-features only or all features. The former is called 'blind' whereas the latter is the 'oracle', since it knows all $\mathcal{P}$-features and $\mathcal{PO}$-features from

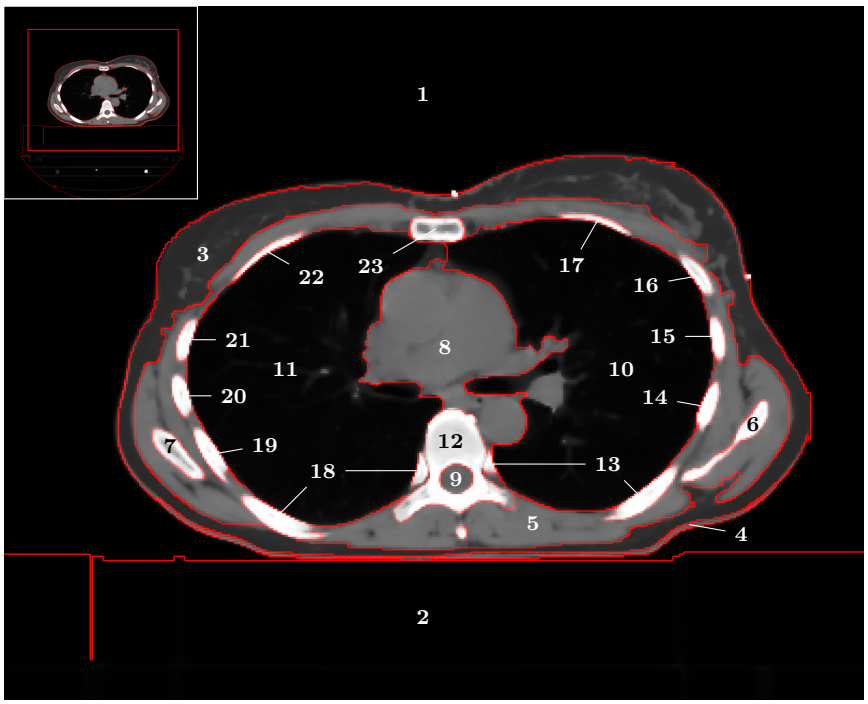


Figure E.5 – Image from the real dataset with all object labels. The objects are described in Table E.1. The image was cropped in order to improve his readability. The cropped area is represented on the upper left corner.

the start. This comparison allows us to rank the various sequencing methods. It also shows whether incremental classification outperforms a blind classifier with $\mathcal{P}$-features only and how close it can get to the oracle's optimal results.

The classifiers used in all approaches are of three kinds: k-nearest neighbors (kNN) majority vote, a support vector machine (SVM) Cortes and Vapnik 1995 and random forests (RF) Breiman 2001. For each binary classifier and for each class/object o, a 'recovery' function is defined in order to force at least one piece of the scene to belong to o. Such a function is necessary to compute the $\mathcal{PO}$-feature associated with o and thereby to increase the base of known features. The selected piece is trivially the one with the highest probability to belong to o. For the kNN classifier, it is the piece with the largest number of neighbors belonging to o in the training set. For the SVM classifier, it is the closest piece to the class separation boundary. For the RF classifier, it is the

1. Out of field and air[1]	13. Left rib 7[8]
2. Treatment table[2]	14. Left rib 6
3. Fat and breast[3]	15. Left rib 5
4. Back skin[4]	16. Left rib 4
5. Muscles[5]	17. Left rib 3
6. Left scapula	18. Right rib 7[8]
7. Right scapula	19. Right rib 6
8. Mediastinum[6]	20. Right rib 5
9. Spinal canal	21. Right rib 4
10. Left lung	22. Right rib 3
11. Right Lung	23. Sternum
12. Vertebra[7]	

Table E.1 – Real images – List of the objects.
[1]'Out of field and air' includes the air surrounding the patient in the cylindrical field of view of the scanner (-1000 HU), as well as the padded corners outside the field of view (-2048 HU).
[2]The low radiodensity of the table aims to minimize X-ray attenuation.
[3]If clips were used in breast cancer surgery, they are considered to belonging to the breast.
[4]Due to weak contrast with fat, only part of the skin could be identified in the patients' back.
[5]All muscles are gathered in the same object. Any small area with lower radiodensity between two muscles is considered to belong to the muscles.
[6]The mediastinum includes all organs between the lungs, namely, the heart, arteries, veins, and oesophagus.
[7]The vertebra can consist of one or several parts, depending on its orientation.
[8]The left and right seventh ribs are sometimes split in two parts, depending on their orientation, and one of these parts often lies near the vertebra (see Fig. E.5).

piece with the largest number of trees that assign it to o.

In order to compare the results obtained with the different methods, a

Dataset	#objects	#pieces	#$\mathcal{P}$-feat.	#$\mathcal{PO}$-feat.	#$\mathcal{PP}$-feat.
Synthetic	8	100	9	3	3
Real 200	23	200	11	3	3
Real 400	23	400	11	3	3

Table E.2 – Main characteristics of the two datasets. There are 100 synthetic images and 49 real ones.

modified t-test is used as suggest by Nadeau Nadeau and Bengio 2003. A 0.95 confident interval is used to establish whether differences are significant or not.

E.4.4 Results

Table E.3 shows that all methods performed well with the synthetic dataset. Since the sum in the BCR runs over all pixels, segmentation mistakes in the watershed algorithm can slightly degrade the BCR. The SVM classifier appears to be the best at solving this problem. The blind method with SVM gives a very good result. Nevertheless, the other blind methods are the only ones significantly outperformed by the blind method with SVM. Table E.3 also shows that the oracle method reaches a BCR of nearly 100% with all kinds of classifiers. This means that the set of considered features is sufficient to classify the scene objects. It is also noteworthy that the incremental method takes profit of non-random classification sequence, with fewer pieces having either no or several labels. Cross-validation seems to work finely with kNN and SVM classifiers, not so well with RF ones. But the RF with cross-validation method does not significantly differ from the other methods.

Table E.3 also indicates that the BCR increases after the final multiclass classification. Figure E.6 confirms this observation by showing a kernel-density approximation of the BCR distribution over all available scenes, before and after the multiclass classifier. Multiclass classification shifts the bulk of the distribution to right and also lifts the peak centred near BCR= 1. Similar trends are observed for the others classifiers/methods except for RF with cross-validation. Nevertheless no significant difference were observed between the classification before and after the final step.

Beyond classification accuracy, stability of the classification sequence can also be investigated. For this purpose, Table E.4 reports the Levenshtein distances between sequences. The Levenshtein distance naturally peaks if the classification sequences are random. As the sequence obtained with nearest neighbors does not

				#label per piece (%)		
		BCR_f(%)	BCR_0(%)	One	Zero	Several
kNN	Random	94.14 ± 11.99	91.99 ± 10.63	95.35	1.39	3.26
	Nearest	98.61 ± 6.00	97.94 ± 4.59	98.45	0.15	1.41
	XVal	98.91 ± 2.81	97.95 ± 3.51	99.99	0.01	0.00
	oracle	99.27 ± 3.13				
	blind	92.86 ± 2.87				
RF	Random	95.02 ± 9.68	93.52 ± 9.43	96.53	1.94	1.54
	Nearest	97.49 ± 6.31	96.62 ± 7.38	99.03	0.82	0.14
	XVal	91.81 ± 19.22	92.84 ± 15.54	97.94	1.54	0.52
	oracle	99.94 ± 0.46				
	blind	95.08 ± 4.02				
SVM	Random	96.27 ± 7.94	94.86 ± 9.08	96.15	1.53	2.32
	Nearest	99.44 ± 2.04	98.24 ± 3.90	98.39	0.33	1.28
	XVal	98.50 ± 4.50	97.00 ± 4.42	98.03	0.83	1.14
	oracle	99.98 ± 0.13				
	blind	98.59 ± 1.55				

Table E.3 – Synthetic images – BCR_f is the BCR computed for all pixels after the whole classification process. BCR_0 is the BCR computed at the end of the iterative process just before the final multiclass classification.

depend on the classifier type, the sequence remains always the same for a given training set, thus justifying why the Levenshtein distance keeps the same value with kNN, RF, and SVM in this particular case. With sequences determined by cross-validation, kNN and SVM yields identical sequences. Cross-validation with RF produces a different and likely suboptimal sequence, which lowers the BCR.

Concerning real data with 200 watersheds, Table E.5 reports the BCR for the organ classification in the 49 tomographic images. The incremental methods with kNN classifier yield rather poor results with this dataset and are significantly outperformed by all the other methods, even the blind one. Similarly, the incremental method using SVM more or less matches the mean BCR of the blind method. However, Figure E.7 shows that the BCR distribution is quite different. With the blind method, the BCR distribution shows that only a few scenes achieve a high BCR, while incremental classification with SVM produces many excellent results but also totally fails on a few scenes. In a clinical application for the delineation of organs at risk, the latter case is

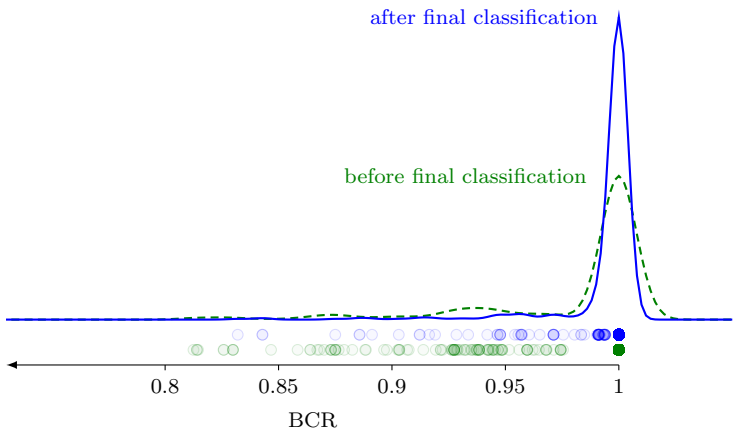


Figure E.6 – Synthetic images – Kernel density estimation of the BCR distribution before (dashed green) and after (solid blue) multiclass classification when using the SVM classifiers sequenced by the nearest neightbors.

		Levenshtein Distance
kNN	Random	6.35 ± 1.04
	Nearest	0.86 ± 0.99
	XVal	0.00 ± 0.00
RF	Random	6.35 ± 1.04
	Nearest	0.86 ± 0.99
	XVal	1.18 ± 1.14
SVM	Random	6.35 ± 1.04
	Nearest	0.86 ± 0.99
	XVal	0.00 ± 0.00

Table E.4 – Synthetic images – Levenshtein distance computed from the 50 sequences.

preferable, since only a few images need corrections. However, due to the higher variability of the incremental method, we cannot observe a significant difference of the BCR. As shown in Table E.5, the BCR can sometimes decrease between the end of the iterative process (BCR_0) and the final classification (BCR_f). It usually occurs when BCR_0 is low and therefore the proportion of wrong feature values is high, hence jeopardising final classification.

				#label per piece (%)		
		BCR_f(%)	BCR_0(%)	One	Zero	Several
kNN	Random	40.82 ± 29.89	54.45 ± 27.72	71.21	7.48	21.31
	Nearest	53.07 ± 33.29	75.32 ± 20.01	89.79	2.75	7.46
	XVal	53.80 ± 29.18	76.46 ± 14.96	90.90	2.38	6.72
	oracle	96.56 ± 3.44				
	blind	81.24 ± 9.75				
RF	Random	93.18 ± 8.64	88.42 ± 10.52	94.00	4.34	1.66
	Nearest	93.41 ± 8.49	88.90 ± 10.04	94.38	3.53	2.09
	XVal	95.22 ± 6.40	91.54 ± 8.11	94.66	3.66	1.68
	oracle	98.56 ± 2.09				
	blind	84.15 ± 8.87				
SVM	Random	78.55 ± 19.54	76.05 ± 20.47	86.99	4.85	8.16
	Nearest	85.03 ± 12.37	82.06 ± 13.39	90.41	2.09	7.50
	XVal	83.98 ± 16.01	84.14 ± 14.34	91.33	3.37	5.30
	oracle	98.49 ± 1.82				
	blind	85.17 ± 9.37				

Table E.5 – Real images (200 watersheds) – BCR_f is the BCR computed at the end of the whole process. BCR_0 is the BCR computed at the end of the iterative process just before applying the final multiclass classification.

Table E.5 indicates that incremental classification with RF works very well and much better than with kNN and SVM. Moreover, this combination significantly outperformed blind classification with kNN, SVM and RF with p-values smaller than 10^{-2}. This good performance of RF can be related to its low number of pieces with multiple labels, compared to kNN and SVM. Actually, each time a piece is included in an object o, it gets involved in the computation of the $\mathcal{PO}$-features associated with object o. If this piece is assigned to more than one object, it can lead to wrong feature values, and therefore to erroneous classification. The obvious conclusion is that wrong pieces information are much more of an issue than missing ones.

Figure E.7 illustrates the benefit of multiclass classification at the end of the process on the BCR distribution when using SVM classifiers sequenced by the nearest-neighbors criterion. Figure E.8 reports a similar effect with RF classifiers sequenced by cross-validation. Many objects of the scenes in the test sets are very well classified, with the main mode of the BCR distribution near 1.

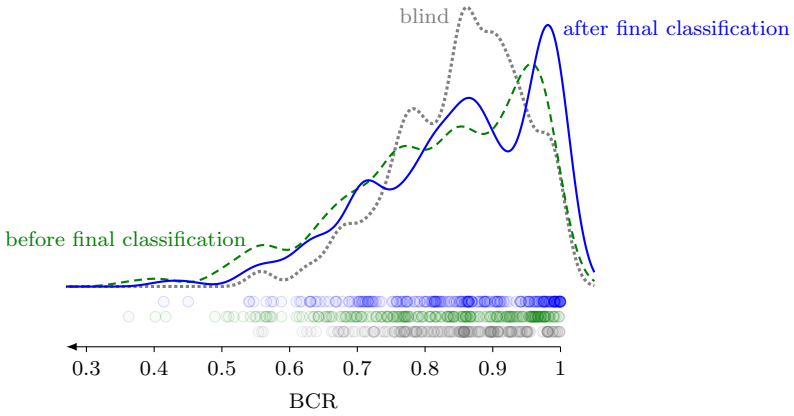


Figure E.7 – Real images (200 watersheds) – Kernel density estimation of the BCR distribution before (dashed green) and after (solid blue) multiclass classification when using SVM classifiers sequenced with sequencing by nearest neightbors. The dotted grey line is the kernel density estimation of the BCR distribution with the blind method.

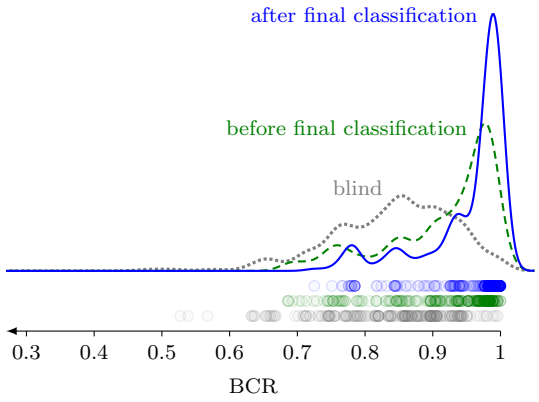


Figure E.8 – Real images (200 watersheds) – Kernel density estimation of the BCR distribution before (green dashed) and after (blue dotted) multiclass classification when using RF classifiers sequenced by nearest neightbors. The dotted grey line is the kernel density estimation of the BCR distribution with the blind method.

Table E.6 shows the distance between the different sequences obtained. Like in the 'synthetic' dataset, the proposed methods improve the stability of the sequence. However, over the 50 runs, none of the sequence repeated more than once. It is noteworthy that except for the kNN classifier (which gives poor BCR), the cross-validation sequencing seems less stable than the nearest neighbors sequencing. The nearest neighbors sequencing is only driven by the data while the cross-validation sequencing is driven by the data and the classifier used. Selecting the cross-validation sequence by training on 80% of the training set (36 images) can explain the loss of stability. With such a small training set, the classifier is not able to extract general information from the dataset.

		Levenshtein Distance
kNN	Random	21.16 ± 1.18
	Nearest	9.21 ± 1.84
	XVal	9.42 ± 2.95
RF	Random	21.16 ± 1.18
	Nearest	9.21 ± 1.84
	XVal	12.81 ± 2.32
SVM	Random	21.16 ± 1.18
	Nearest	9.21 ± 1.84
	XVal	15.17 ± 2.71

Table E.6 – Real images (200 watersheds) – Levenshtein distance computed from the 50 sequences.

The number of times an object is classified before another is represented in Figure E.9. Somehow it shows the dependencies between objects. As it can be seen, for all methods, the object 'table' is always the first to be identified and the ribs depend on a lot of object and are always in the second half of the classification. In the case of the sequencing by nearest neighbors, the first objects to be classified are always the same: table, air, muscle, fat and breast, mediastinum, back skin. In the case of the RF, everything but the spinal canal is identified before the beginning the identification of the ribs. Rib shifts are rare with RF as there are more information for their identification. If all ribs are mislabelled, the BCR value drops.

Table E.7 shows the results obtained with the real dataset where each image is segmented in 400 watersheds. The results are very similar to those in table E.5. The incremental methods using cross-validation or nearest neighbors with RF are again significantly better than all the blind methods (p-value $< 2\times10^{-2}$).

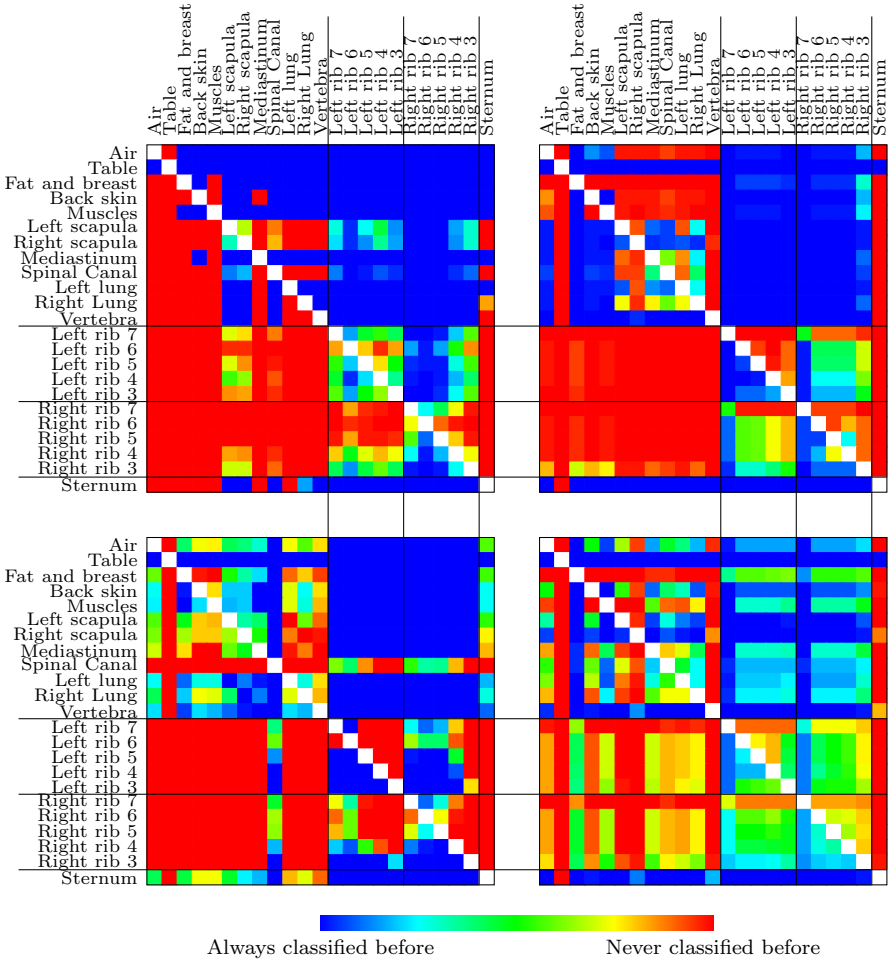


Figure E.9 – Real images (200 watersheds) – Upper left: Sequencing by nearest neighbors (same result with all the classifiers) – Upper right: Sequencing by cross-validation with kNN– Lower left: Sequencing by cross-validation with RF– Lower right: Sequencing by cross-validation with SVM– Representation of the dependencies between the different object. The number of times an object was classified after another was counted over the 50 runs. An object often classified after another is considered as being directly or indirectly dependent on that object.

Nevertheless, it can be seen that all the BCR values are a few percent lower when the number of watershed is increased. Apart for the incremental methods with kNN, the difference is not significant.

				#label per piece (%)		
		BCR_f(%)	BCR_0(%)	One	Zero	Several
kNN	Random	36.54 ± 28.91	51.67 ± 27.40	65.75	7.22	27.03
	Nearest	22.11 ± 21.86	43.61 ± 19.79	82.53	4.26	13.21
	XVal	13.77 ± 22.77	13.75 ± 22.74	99.80	0.20	0.00
	oracle	97.00 ± 2.24				
	blind	81.43 ± 9.48				
RF	Random	90.04 ± 10.49	86.46 ± 11.38	93.58	4.02	2.40
	Nearest	90.27 ± 9.59	86.43 ± 10.31	93.35	3.15	3.50
	XVal	93.53 ± 7.14	89.89 ± 8.50	94.56	3.33	2.11
	oracle	97.74 ± 2.33				
	blind	82.28 ± 9.53				
SVM	Random	76.85 ± 18.91	75.10 ± 18.62	85.96	5.22	8.82
	Nearest	74.79 ± 15.60	69.89 ± 17.53	82.37	3.25	14.38
	XVal	81.00 ± 15.94	81.81 ± 13.22	89.98	2.72	7.31
	oracle	98.51 ± 1.51				
	blind	82.37 ± 9.03				

Table E.7 – Real images (400 watersheds) – BCR_f is the BCR computed at the end of the whole process. BCR_0 is the BCR computed at the end of the iterative process just before applying the final multiclass classification.

Figure E.10 shows the normalised confusion matrix for RF classifiers sequenced with cross-validation for the real dataset with 200 watersheds. Each matrix entry is represented with a big pixel whose gray level actually corresponds to the fourth root of the considered value. This nonlinear transformation strongly increases the visual contrast among the really tiny off-diagonal entries. Doing so reveals two kinds of classification mistakes that can be related to the particularities of the problem at hand. First, many small bits of all organs can be wrongly classified as being muscle. Such confusion stems from the rather complicated shape and high heterogeneity of this region, which includes not only actual muscle but also some interstitial fat and cartilage between the ribs and the sternum. Second, it can happen that the series of numbered ribs is shifted by one position. Such mistakes partly stem from ambiguous organ delineation in the training set. For instance, the last rib (7th) is sometimes split in two

parts, one of them being contiguous to the vertebra. Figure E.5 illustrates this phenomenon (see labels 13 and 18). The risk is then high that the piece of rib close to the vertebra gets merged with it. Similarly, the classification method can difficultly determine whether the 7th rib tag has to be assigned twice. Mistakes at this stage then propagate to the other ribs and explain the shift. The result with the lowest BCR, obtained after the classification with RF and cross-validation, is shown on Figure E.11. This image illustrates the rib shifting effect as well as the wrong muscle identification.

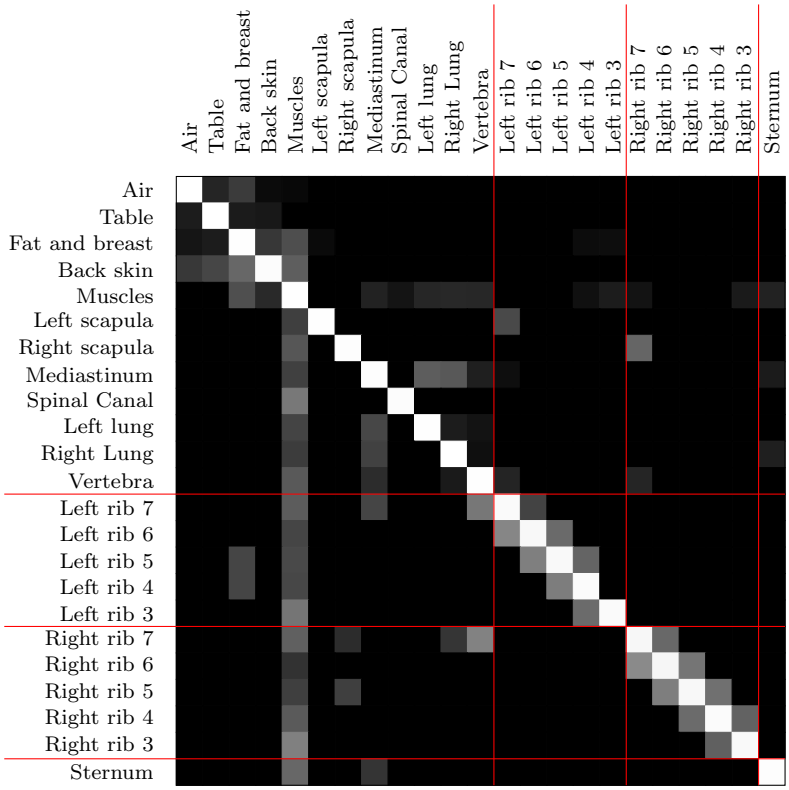


Figure E.10 – Real images (200 watersheds) – Representation of the confusion matrix. Each row was normalized so that it adds up to one. To improve visual contrast among the off-diagonal entries, the fourth root was applied. One can observe that many small portions of organs are sometimes misclassified as being muscle. Similarly, the series of numbered ribs is sometimes shifted by one position, the rib next to the vertebra being then merged with the latter.

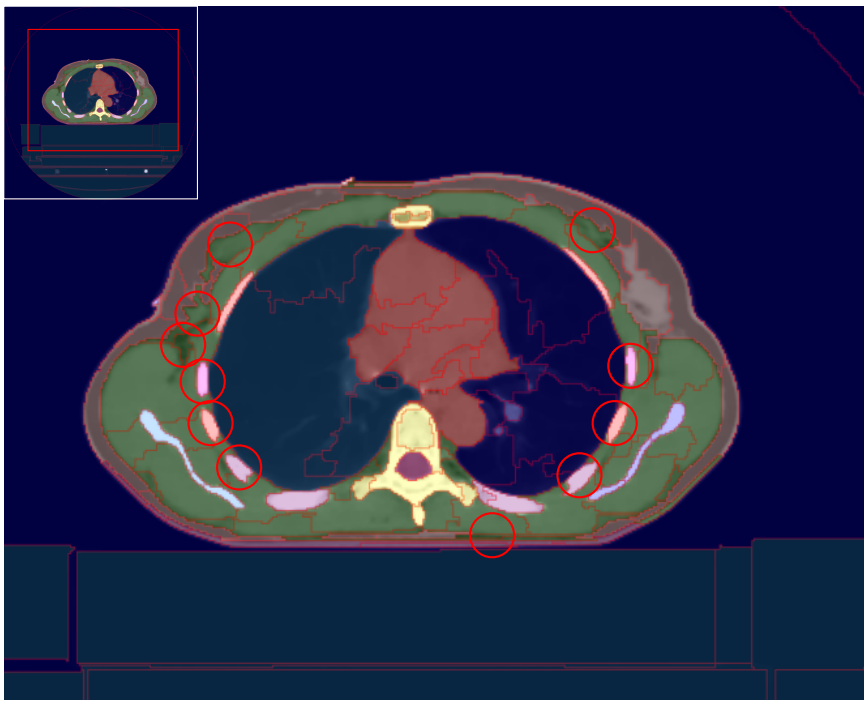


Figure E.11 – Real images (200 watersheds) – Bad classification: BCR=0.72. Red circles highlight the mistakes. Errors on the ribs: ribs 7 is identified twice, ribs 6 and 5 are shifted, rib 4 is missing. Confusion between muscle and fat-breast area: mammary gland are identified as being muscle, some of the fat near to the muscle is wrongly labelled as being muscle.

Heterogeneity of the muscle region and ambiguous rib definitions in the training data show that data quality is of paramount importance. Any imperfection directly reduces the maximal performance level that even the best classification method could reach. Other ambiguous cases are for instance some pieces between the heart and the sternum, which were defined as belonging to the mediastinum on some training images and to a lung on others. In this particular case, however, the incremental classifier overcame the ambiguity and behaved in a consistent way, producing always the same result. In our experiment, we observed that some images raised many issues (rib shifting, muscle identification...) independently of the training set, whereas some of the others led to nearly perfect labelling every time they were used in the test set. Those wrongly classified images may be outliers in our small population.

Blind and incremental classification can be compared qualitatively on the segmented images. Figure E.12 shows the segmentations obtained with 3 images. The blind method makes a lot of mistakes of various types. The spinal canal is often misclassified as part of the mediastinum or muscles. As the classification process does not use features like the relative position, confusion between the scapula and the ribs is unavoidable. In one case, part of the mediastinum is labelled as being the sternum. Moreover, ribs are often misclassified.

E.5 Conclusion

This paper describes a method of incremental classification with two different criteria to build the sequence of considered features. It can deal with problems in which the values of some features are not known from the beginning and depend on a partial classification. The incremental nature of the process aims at initiating and progressively enriching this partial classification in an iterative way. The method is generic and can solve the sub-problems in each iteration with various classifiers like kNN, SVM, random forests, etc. At the end of the procedure, when all features are known or estimated, a usual multi-class classifier refines the result. Like the binary classifiers, it can rely on any existing kind of classifier. Depending on the problem at hand, the procedure must be adapted with appropriate definitions of features. Failure to do so increases the risk of error propagation in the incremental process. Experiments on both synthetic and real images show that the proposed method is effective and significantly improves classification accuracy, compared to traditional non-incremental methods.

Future work will concentrate on designing a single, custom classifier that deals with all known and unknown features at all times, thanks to the use of adaptive relevance factors that indicate the reliability of each feature.

E.6 Bibliography

Beucher, Serge and Christian Lantuéjoul (1979). 'Use of watersheds in contour detection'. In:

Beucher, Serge and Fernand Meyer (1992). 'The morphological approach to segmentation: the watershed transformation'. In: *OPTICAL ENGINEERING-NEW YORK-MARCEL DEKKER INCORPORATED-* 34, pp. 433–433.

Breiman, Leo (2001). 'Random forests'. In: *Machine learning* 45.1, pp. 5–32.

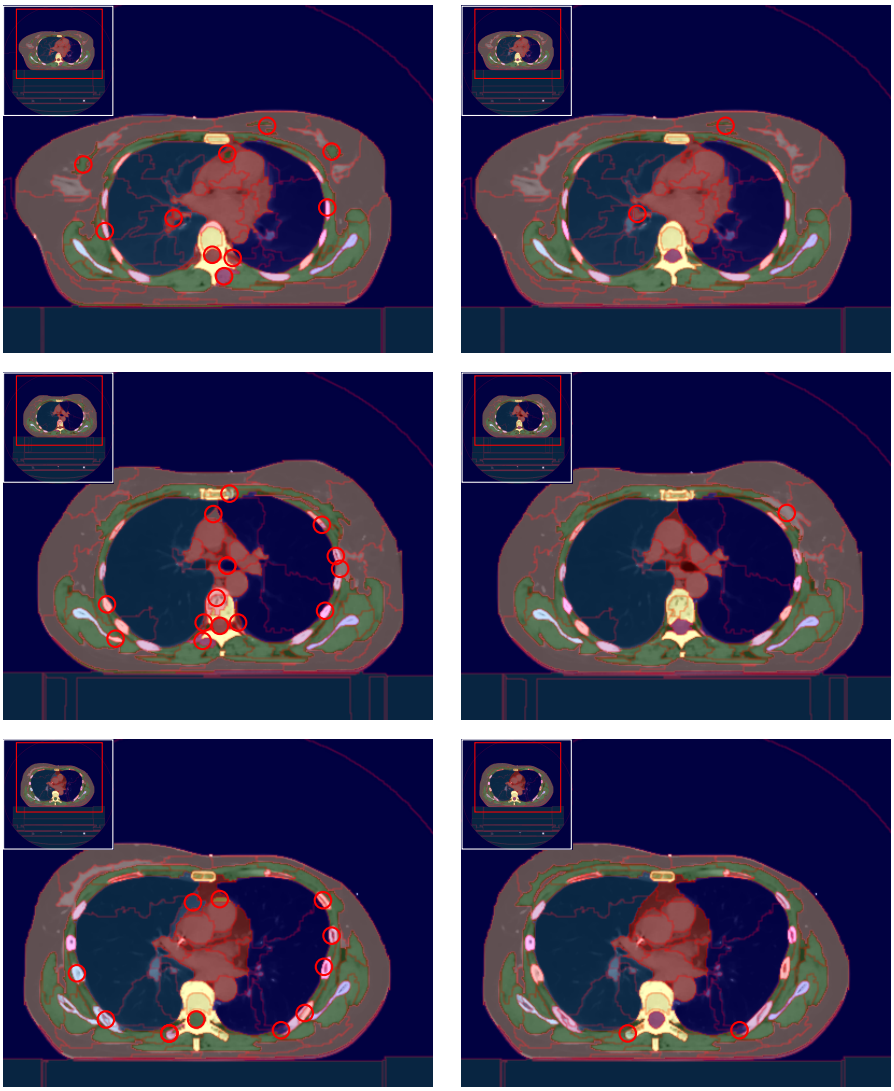


Figure E.12 – Real images (200 watersheds) – Left: blind segmentations with RF– Right: incremental segmentations, sequencing by cross-validation with RF. Red circles highlight the mistakes. Blind classification makes a lot of various mistakes.

Chambolle, Antonin (2004). 'An algorithm for total variation minimization and applications'. In: *Journal of Mathematical imaging and vision* 20.1-2, pp. 89–97.

Chen, Xi (Stephen), Arpit Jain and Larry Davis (2014). 'Object Co-labeling in Multiple Images'. In: *IEEE Winter Conference on Applications of Computer Vision (WACV) 2014*.

Cortes, Corinna and Vladimir Vapnik (1995). 'Support-vector networks'. In: *Machine learning* 20.3, pp. 273–297.

Cousty, Jean, Gilles Bertrand, Laurent Najman and Michel Couprie (2009). 'Watershed cuts: Minimum spanning forests and the drop of water principle'. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31.8, pp. 1362–1374.

Felzenszwalb, Pedro F and Daniel P Huttenlocher (2004). 'Efficient graph-based image segmentation'. In: *International Journal of Computer Vision* 59.2, pp. 167–181.

Gould, Stephen, Richard Fulton and Daphne Koller (2009). 'Decomposing a scene into geometric and semantically consistent regions'. In: *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1–8.

Grégoire, V., E. Coche, G. Cosnard, M. Hamoir and H. Reychler (2000). 'Selection and delineation of lymph node target volumes in head and neck conformal radiotherapy. Proposal for standardizing terminology and procedure based on the surgical experience'. In: *Radiotherapy and Oncology* 56.2, pp. 135–150.

Grégoire, Vincent et al. (2003). 'CT-based delineation of lymph node levels and related CTVs in the node-negative neck: DAHANCA, EORTC, GORTEC, NCIC, RTOG consensus guidelines'. In: *Radiotherapy and Oncology* 69.3, pp. 227–236.

Helleputte, Thibault and Pierre Dupont (2009). 'Partially supervised feature selection with regularized linear models'. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. Montreal, Quebec, Canada: ACM, pp. 409–416.

Ion, Adrian, Joao Carreira and Cristian Sminchisescu (2011). 'Probabilistic joint image segmentation and labeling'. In: *Advances in Neural Information Processing Systems*, pp. 1827–1835.

Kazmar, T., E.Z. Kvon, A. Stark and C.H. Lampert (2013). 'Drosophila Embryo Stage Annotation Using Label Propagation'. In: *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 1089–1096.

Kuettel, Daniel, Matthieu Guillaumin and Vittorio Ferrari (2012). 'Combining image-level and segment-level models for automatic annotation'. In: *Advances in Multimedia Modeling*. Springer, pp. 16–28.

Levendag, P., M. Braaksma, E. Coche, H. van Der Est, M. Hamoir, K. Muller, I. Noever, P. Nowak, J. van Sörensen De Koste and V. Grégoire (2004). 'Rotterdam and Brussels CT-based neck nodal delineation compared with the surgical levels as defined by the American Academy of Otolaryngology–Head and Neck Surgery'. In: *International Journal of Radiation Oncology* Biology* Physics* 58.1, pp. 113–123.

Levenshtein, VI (1966). 'Binary Codes Capable of Correcting Deletions, Insertions and Reversals'. In: *Soviet Physics Doklady* 10, p. 707.

Lim, Y.W. and S.U. Lee (1990). 'On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques'. In: *Pattern Recognition* 23.9, pp. 935–952.

Mohan, Anuj, Constantine Papageorgiou and Tomaso Poggio (2001). 'Example-based object detection in images by components'. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23.4, pp. 349–361.

Nadeau, Claude and Yoshua Bengio (2003). 'Inference for the generalization error'. In: *Machine Learning* 52.3, pp. 239–281.

Najman, Laurent and Michel Couprie (2006). 'Building the component tree in quasi-linear time'. In: *Image Processing, IEEE Transactions on* 15.11, pp. 3531–3539.

Rudin, Leonid I., Stanley Osher and Emad Fatemi (1992). 'Nonlinear total variation based noise removal algorithms'. In: *Physica D: Nonlinear Phenomena* 60.1–4, pp. 259–268. ISSN: 0167-2789.

Shi, Jianbo and Jitendra Malik (2000). 'Normalized cuts and image segmentation'. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.8, pp. 888–905.

Wijers, O.B., P.C. Levendag, T. Tan, E.B. van Dieren, J. van Sörnsen de Koste, H. van der Est, S. Senan and P.J.C.M. Nowak (1999). 'A simplified CT-based definition of the lymph node levels in the node negative neck'. In: *Radiotherapy and oncology* 52.1, pp. 35–42.