TECHNISCHE UNIVERSITÄT MÜNCHEN

Fakultät für Informatik
Computer Aided Medical Procedures & Augmented Reality / I16

Hough Voting Strategies for Segmentation, Detection and Tracking

Fausto Milletari

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr.-Ing. Darius Burschka
Prüfer der Dissertation:
1. Prof. Dr. Nassir Navab
2. Sen. Lecturer Dr. Tom Vercauteren

Die Dissertation wurde am 10.11.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 19.01.2018 angenommen.

Abstract

Object detection, segmentation and visual tracking are extremely important problems in both computer vision and medical image analysis. In the last few years a plethora of methods leveraging a wide range of techniques such as optimization, integration of statistical priors, variational methods, feature extraction and matching have been proposed. The most recent scientific efforts focused on proposing machine learning based approaches that can tackle and solve these problems appropriately. Methods that are based on handcrafted features, the so called shallow approaches, have been widely used and explored until very recently and were often employing machine learning algorithms such as boosting, support vector machines, random forests coupled with a careful choice of manually engineered features which were designed in a specific manner for each task. Other researchers, who focused their attention on sparse sensing and dictionary learning, have achieved notable results proposing methods that leverage sparse coding to discover sets of basis functions which can be sparsely combined to reconstruct signals. These basis functions capture salient characteristics of the data at hand without requiring any manual efforts. Most recently these approaches have been replaced by deep learning methods which are as well capable of learning features directly from raw data and can capture semantically meaningful information in a hierarchical and structured fashion. Such approaches, which are particularly suited for vision tasks, deliver in some cases superhuman performances when applied to challenging problems. Although machine learning approaches as such delivered outstanding performances on a number of challenging tasks, many methods – especially in the field of medical image analysis – cannot still be applied in a straightforward manner. The lack of large amounts of annotated training data, the presence of noise and artifacts, the low inter-class versus the high intra-class variability of the samples, and other domain-specific factors, often limit the performances of the models. In the same way, computer vision problems such as visual tracking and pose estimation have proven more challenging than others, in the first case, due to the limited knowledge of the appearance of the object of interest beforehand, and in the second case due to the need to retrieve a precise 6 DoF pose from unconstrained RGBd frames.

In this thesis I will show how voting strategies can be used to tackle detection, segmentation and pose estimation problems relying on voting strategies which look only at image parts and assemble the resulting knowledge into a global decision. This approach overcomes the limitation of current machine learning methods in all those cases where, due to the nature of the data and despite appropriate training, the uncertainty of the decision over previously unseen data remains high. These cases include the situations, often encountered in medical image analysis, when the anatomy of interest cannot be easily distinguished from its surrounding and when only part of it is visible; in a similar manner, in the field of computer vision, we aim to manage uncertainty when high background clutter and extensive occlusions are present.

Zusammenfassung

Objekterkennung, Segmentierung und visuelle Verfolgung sind bei der Computer Vision und der medizinischen Bildanalyse äußerst wichtige Probleme. In den letzten Jahren wurde eine Vielzahl von Methoden, die eine breite Palette von Techniken wie Optimierung, Integration von statistischen Priors,Variationsmethoden, Merkmalsextraktion und Matching nutzen, vorgeschlagen. Die jüngsten wissenschaftlichen Bemühungen konzentrierten sich darauf, maschinell lernende Ansätze vorzuschlagen, die diese Probleme angemessen anpacken und lösen können. Methoden, die auf handgefertigten Merkmalen basieren, sogenannte flache Ansätze, werden bis heute erforscht und sind weit verbreitet. Oft sind diese mit maschinellen Lernalgorithmen wie Boosting, Support-Vektor-Maschinen, Random Forests gekoppelt, die eine sorgfältige Auswahl von manuell konstruierten Features verarbeiten, welche für jede Aufgabe spezifisch entworfen wurden. Andere Forscher, die ihre Aufmerksamkeit auf Sparse Sensing und Dictionary Learning konzentriert haben, haben bemerkenswerte Ergebnisse vorgestellt anhand von Methoden, die Sparse Coding nutzen, um Gruppen von Basisfunktionen zu entdecken, welche dünnbesetzt kombiniert werden können, um Signale zu rekonstruieren. Diese Basisfunktionen erfassen markante Merkmale der Daten, ohne dass manuelle Handlungen erfordert werden. Zuletzt wurden diese Ansätze durch Tiefe Neuronale Lernmethoden ersetzt, die ebenso gut fähig sind, Merkmale direkt aus Rohdaten zu erlernen und semantisch sinnvolle Informationen hierarchisch und strukturiert zu erfassen. Solche Ansätze, die sich besonders für visuelle Perzeption eignen, liefern in manchen Fällen übermenschliche Leistungen, wenn sie auf anspruchsvolle Probleme angewendet werden. Obwohl das maschinelle Lernen als solches hervorragende Leistungen bei einer Reihe von anspruchsvollen Aufgaben erbracht hat, können viele Methoden - vor allem im Bereich der medizinischen Bildanalyse - nicht immer einfach angewendet werden. Das Fehlen großer Mengen an annotierter Trainingsdaten, das Vorhandensein von Rauschen und Artefakten, die niedrige Inter-Klassen- gegenüber der hohen Intra-Klassen-Variabilität der Daten und andere domänenspezifische Faktoren beschränken oft die Leistungen der Modelle. In gleicher Weise haben sich Computer Vision Probleme wie visuelle Verfolgung und Posen-Schätzung als schwieriger herausgestellt als andere Probleme, im ersten Fall aufgrund der eingeschränkten Kenntnis des Erscheinungsbildes des interessierenden Objekts im Vorfeld und im zweiten Fall, aufgrund der Notwendigkeit, eine präzise Pose mit sechs Freiheitsgraden aus unbeschränkten RGBd-Frames erkennen zu müssen. In dieser Arbeit werde ich zeigen, wie Voting Strategien verwendet werden können, um die Probleme in Erkennung, Segmentierung und Posen-Schätzung zu lösen. Die Voting-Strategien stützen sich lediglich auf Teilen des Gesamtbilds und setzen das daraus resultierende Wissen zu einer globalen Entscheidung zusammen. Dieser Ansatz überwindet die Begrenzung aktueller maschineller Lernmethoden in all jenen Fällen, in denen aufgrund der Art der Daten und trotz entsprechenden maschinellen Trainings die Unsicherheit der Entscheidung über bisher nicht sichtbare Daten hoch bleibt. Diese Fälle beinhalten Situationen, die oft in der medizinischen Bildanalyse auftreten, wenn die interessierende Anatomie nicht leicht von ihrer Umgebung

zu unterscheiden oder wenn nur ein Teil davon sichtbar ist. In ähnlicher Weise wollen wir Detektions-Unsicherheiten im Bereich der Computer Vision bewältigen, wenn ein hoher Grad an Unordnung im Hintergrund und umfangreiche Verdeckungen vorhanden sind.

Acknowledgments

I would like to thank all the people who have enabled this work and have supported me during my PhD: my co-authors and colleagues who have been providing essential insights, knowledge and support to my research; my PhD advisor Dr. Ahmad Ahmadi that has taught me everything I know about ultrasound imaging and has been a great source of suggestions and support during these years; my supervisor Prof. Nassir Navab who has been providing exceptional guidance for my research yet granting all the necessary freedom to explore new topics, applications and techniques and has enabled me to pursue a successful career in this field.

My gratitude goes also to my family who has supported me during these years especially in the difficult moments. This help has been invaluable and has allowed me to reach my goals.

Lastly I would like to acknowledge my colleagues and friends Wadim Kehl, Dr. Vasilis Belagiannis, Robert Di Pietro, Mira Slavcheva, Marco Esposito, Nicola Rieke, Dr. Christoph Hennersperger, Dr. Shadi Albarqouni, Dr. Pascal Fallavollita, Sailesh Conjeti, Dr. Maximilian Baust, Dr. Ilic Slobodan and all the other people associated with the chair of Computer Aided Medical Procedures at TUM and JHU who have been interacting with me on a day to day basis in these years.

Fausto Milletari

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Machine learning has recently emerged as a valuable tool to aid humans to accomplish tasks. In the last few years we have witnessed significant progress and several innovations in the field of computer vision and pattern recognition. Machine learning fueled multiple of these advancements. Low level vision tasks such as segmentation, tracking and object detection as well as high level applications such as self driving cars, autonomous flying robots, self-taught robotic object manipulation, pervasive augmented reality and systems capable of reliably recognizing people or places have been enabled by recent efforts of the machine vision community. Similarly, in medical field, approaches aiming to solve diagnostic, interventional, and surgical planning problems have been proposed and have enabled a number of novel techniques. This corresponds to most of the research around computer aided medical procedures which focuses on lesion detection, organ segmentation, computer aided diagnosis, visual tracking, motion analysis and compensation, surgical robotics, etc.

Although there is a junction point between computer vision and computer aided medical procedures, which is represented by the use in both cases of images to accomplish or aid tasks, the two fields are inherently different.

Computer vision deals with images captured with cameras, or depth sensors, which exhibit, in general, a limited amount of noise and artifacts, are easily understandable by human observers, and whose interpretation is challenging for computer algorithms due to the complexity of the world they depict. Main challenges are represented by the presence of clutter, occlusions and illumination changes which are often irrelevant to human observers but are hard to be properly handled by computer algorithms. Moreover, the behaviour of objects, people or animals present in the scene is motivated and determined by the context, the semantic relationships that can be built between different entities and, ultimately, by physical laws. Pictures depicting scenes with multiple objects and entities are often open to multiple interpretations depending on the settings where the picture is taken, the characteristics of the people involved, and the role that the objects might have in the specific context being considered. In other words, a paramount need in computer vision is to include a deep knowledge of the world in computer algorithms such that complex and non-trivial situations can be interpreted.

Computer aided medical procedures, which corresponds largely to medical image analysis, focuses on the interpretation of medical data such as volumetric scans of the human body obtained with techniques such as Magnetic Resonance Imaging (MRI), Computed Tomography (CT) and ultrasound (US). This data is acquired by imaging processes that are very different from the ones involved in computer vision, since they rely on effects such as variations of magnetic momentum of atoms, the interaction of X-rays with matter and their attenuation, and the propagation of sound in mediums of variable density. This results in images that do not represent the world as our eyes would perceive it and that are often corrupted by noise and artifacts which render their interpretation not trivial for untrained observers. The image acquisition process can produce very different data depending on the equipment, on the experience and preferences of the person who acquires the data, and on a huge number of other parameters such as the size of the patient, his voluntary or involuntary motions, and, of course, by anatomical variations that can be normally observed among both healthy and diseased subjects.

This thesis mainly focuses on problems related to medical image analysis, and in particular segmentation of different organs and structures of interest in MRI and freehand ultrasound images. Some of these solutions and findings have been applied also to the field of computer vision for applications such as visual tracking and object pose estimation, always in the spirit of obtaining superior performances by incorporating more knowledge about the world and by managing the unavoidable uncertainty of machine learning approaches through voting strategies. Additionally other novel approaches which do not make use of Hough voting are introduced in this thesis to solve specific tasks or as means of comparison to highlight the advantages and disadvantages of voting based approaches.

1.1 Contributions

The main contributions presented in this thesis can be summarized as follows:

- We propose to address the problem of simultaneous localization and segmentation of various anatomies in freehand ultrasound images by employing a Hough voting based framework relying on random forests and handcrafted features. We show how volumetric segmentation can be obtained by employing the voting strategy in conjunction with an atlas of manually segmented volumes.
- We extend the previous idea to exploit features that are automatically discovered from the data at hand through the application of a sparse auto-encoder.
- We show, through our Hough-CNN, that deep neural networks can be employed at the core of a Hough voting based framework for the segmentation of various regions of the human brain in both MRI and freehand ultrasound images. A large study on different network architectures and different amounts of training data is performed to show the robustness

of voting approaches versus traditional ones. We show that the main idea behind this method can be also applied to 6 degrees of freedom (DoF) object pose estimation in colour images acquired through a depth camera.

- We propose V-Net, a fully convolutional neural network (FCNN) which uses a loss function based on the Dice overlap coefficient and therefore is specifically tailored for segmentation tasks. Beside comparing the performances of this method to the the results of our Hough-CNN approach, we use it to demonstrate prostate segmentation in MRI.

- We propose to use dictionary learning to detect and track electrophysiology (EP) catheters in X-Ray fluoroscopy images. We extend such approach to a voting based technique that uses dictionary learning to encode and capture generic world knowledge and ultimately employ it for the task of visual object tracking.

These, and other minor contributions are discussed as outlined in the following.

1.2 Thesis Outline

This thesis follows the structure presented below. Some items in this list contain references to published work that is directly connected to the specific topics discussed in the relative chapter.

Chapter 1 is this chapter, it contains a brief introduction to the thesis.

Chapter 2 is devoted to introduce some of the most relevant theoretical notions that are useful to gain a better understanding of this work. The foundations of machine learning are briefly described and notions related to image representation in computer systems are discussed. Basic notions about the medical data acquisition modalities used in this research, such as freehand US, MRI and X-Ray will be also briefly introduced.

Chapter 3 discussed the work relative to the topic of medical image segmentation. A brief analysis of the state of the art in this field and clinical motivation is presented. In particular, we discuss here our approaches exploiting random forests (RF), sparse auto-encoders (SA) and convolutional neural networks (CNN) together with a Hough voting strategy for anatomy delineation. Additionally we discuss our recent approach exploiting fully convolutional neural network (FCNN) for the same task. Important references for this chapter are:

- Milletari, F., Yigitsoy, M., Navab, N.: Left ventricle segmentation in cardiac ultrasound using hough-forests with implicit shape and appearance priors pp. 49–56 (2014)

- Milletari, F., Ahmadi, S.A., Kroll, C., Hennersperger, C., Tombari, F., Shah, A., Plate, A., Boetzel, K., Navab, N.: Robust segmentation of various anatomies in 3d ultrasound using hough forests and learned data representations. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015, pp. 111–118. Springer (2015)

- Milletari, F., Ahmadi, S.A., Kroll, C., Plate, A., Rozanski, V., Maiostre, J., Levin, J., Dietrich, O., Ertl-Wagner, B., Bötzel, K., et al.: Hough-cnn: Deep learning for segmentation of deep brain regions in mri and ultrasound. Computer Vision and Image Understanding (2017)

- Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 3D Vision (3DV), 2016 Fourth International Conference on, pp. 565–571. IEEE (2016)

- Zettinig, O., Shah, A., Hennersperger, C., Eiber, M., Kroll, C., Kübler, H., Maurer, T., Milletarì, F., Rackerseder, J., zu Berge, C.S., et al.: Multimodal image-guided prostate fusion biopsy based on automatic deformable registration. International journal of computer assisted radiology and surgery **10**(12), 1997–2007 (2015)

- Bortsova, G., Sterr, M., Wang, L., Milletari, F., Navab, N., Böttcher, A., Lickert, H., Theis, F., Peng, T.: Mitosis detection in intestinal crypt images with hough forest and conditional random fields. In: International Workshop on Machine Learning in Medical Imaging, pp. 287–295. Springer (2016)

Chapter 4 introduces our approaches to detect surgical tools and objects in images. In particular, we discuss how electrophysiology catheters can be detected in X-Ray images and how their 3D spatial arrangement can be reconstructed from two views. Additionally we discuss an approach exploiting Hough voting to detect objects in RGB-D images. As customary, the state of the art is briefly analyzed and the motivation of the work is presented in this chapter.

- Milletari, F., Navab, N., Fallavollita, P.: Automatic detection of multiple and overlapping ep catheters in fluoroscopic sequences. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 371–379. Springer (2013)

- Milletari, F., Belagiannis, V., Navab, N., Fallavollita, P.: Fully automatic catheter localization in c-arm images using â„“1-sparse coding. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 570–577. Springer (2014)

- Kehl, W., Milletari, F., Tombari, F., Ilic, S., Navab, N.: Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation. In: European Conference on Computer Vision, pp. 205–220. Springer (2016)

- Baur, C., Milletari, F., Belagiannis, V., Navab, N., Fallavollita, P.: Automatic 3d reconstruction of electrophysiology catheters from two-view monoplane c-arm image sequences. International journal of computer assisted radiology and surgery **11**(7), 1319–1328 (2016)

Chapter 5 presents our work relative to tracking of objects in videos. This method is based on dictionary learning and sparse coding to solve the problem of object tracking in computer vision domain.

- Milletari, F., Kehl, W., Tombari, F., Ilic, S., Ahmadi, S.A., Navab, N.: Universal hough dictionaries for object tracking. In: BMVC, pp. 122–1 (2015)

Chapter 6 is devoted to conclusions and discussion about future works.

Chapter 2

Background

In this chapter we introduce concepts fundamental to the rest of this work. In particular we show how images are represented in computer systems, how they are encoded and how to extract features to describe them. We additionally discuss concepts relative to probability theory and machine learning fundamentals such as classification and regression. In order to keep this work self-contained we provide an explanation of concepts relative to random forests, Hough voting, convolutional neural networks and sparse coding. Finally we focus our attention on medical images modalities, restricting ourselves to the modalities that have been employed in this thesis: ultrasound, MRI and X-Ray.

2.1 Foundations

Notions about the representation of images in a computer system and feature extraction are introduced in this section, followed by a brief explanation of well understood concepts in machine learning such as probability and its application for tasks such as classification and regression.

2.1.1 Image representation

Images are usually obtained by sensors whose role is to pick up signals resulting from physical interactions happening at a microscopic level in the observed specimen. In computer vision, for example, interactions between light and objects are commonly observed to create pictures. The source of contrast, in this case, is the behaviour of different materials when exposed to light. Most materials absorb some parts of the light spectrum and appear with different colors. When appropriate sensors are used, even subtle differences in this signals can be acquired and stored in digital format.

The imaging modalities discussed in this work are all relying a specific physical phenomenon which is sensed by dedicated hardware to form images. In computer vision we use most often visible or infrared light; in X-Ray and CT we use higher energy photons that are absorbed and scattered while they traverse a medium; in ultrasound we rely on the interactions between a sound

Figure 2.1: Schematic representation of how images are encoded in computer systems in color (RGB) and grayscale format.

wave and materials with different density as modeled by the Snell's law; in MRI we sense the nuclear atomic spin after it has been perturbed by a strong magnetic field.

Digital images are usually obtained by converting the electric signal coming from the sensors to digital data through analog-to-digital converters. An interface for digital data transfer is then used to be store or transfer the images to other systems. Depending on the modality this process can be different or more complex.

Regardless of the physical imaging process, the information resulting from the imaging process is stored into a computer system numerically into a tensor.

Grayscale images

Grayscale digital images are represented in computer systems as matrices (two dimensional tensors). Each element of the matrix represents the intensity value of one pixel. Differently than black-and-white images, grayscale ones can contain many shades of gray. Depending on the image format each pixel is represented using a fixed number of bits. Images making use of 8 bits encoding for their pixels represent the majority. Using 8 bits it is possible to encode 256 gray level. In some cases other encodings, often using 12 or 16 bits, are preferred in order to increase the fidelity of the digital information to the underlying analog raw signal. A schematic representation is show in Figure 2.1.

Color images

Color images are represented as a tensor in a computer system. A color image has 3 dimensions, the first two are spatial representing height and width, and the third is devoted to channels, which are used to represent the color information. Images can be represented using three channels in the red-green-blue (RGB) encoding, which renders color by fusing red, green and blue by addition. Each channel carries the intensity information for each pixel of the red, green and blue color component of the image. A variable number of bits can be used to encode RGB information. A popular choice is to use 8 bits per channel, in order to obtain approximately 16 millions possible combinations of colors. A schematic representation of this is shown in Figure 2.1. Using the same representation other color spaces such as LAB and HSV have been proposed.

Depth images

Depth images are acquired with special hardware. Popular choices for such systems are represented by Kinect® sensors which made use of "light-coding" to detect depth in indoor settings by projecting an infrared light pattern which is then recognized by a camera. A newer version of the device has been recently proposed making use of a time-of-flight sensor which allows for outdoor use. The ability of acquiring these kind of images has fueled and motivated an incredible amount of research in the last few years in the fields of robotic vision, human pose estimation and other detection or recognition tasks where the capability of sensing depth has a crucial role. Depth is represented as an additional channel of an RGB image, which motivates the name of RGBd that is often used to indicate this kind of data.

Volumes and beyond

Medical images obtained with imaging modalities such as MRI, CT and with 3D or freehand ultrasound (Section 2.3.1) have a dimensionality that goes beyond 2D. Tomographic images have at least three dimensions and are stored as a collection of voxels, that is volumetric pixels which carry an intensity value representing physical characteristics of the portion of tissue the represent. With recent technologies it is possible to acquire collections of tomographic images in a short amount of time. 3D ultrasound is capable of producing volumetric data in real time, while both CT and MRI can be driven to acquire "cine" sequences at a few frames per second. These images can be seen as a 4D (3D + time) data. In ultrasound, it is also possible to acquire 3D Doppler data over time, which adds another dimension (due to the color information used to encode flow in color Doppler scans) to the data.

2.1.2 Pre-processing and Feature extraction

Pre-processing is often essential to standardize data in order to facilitate further processing. Computer vision tasks can be adversely influenced by

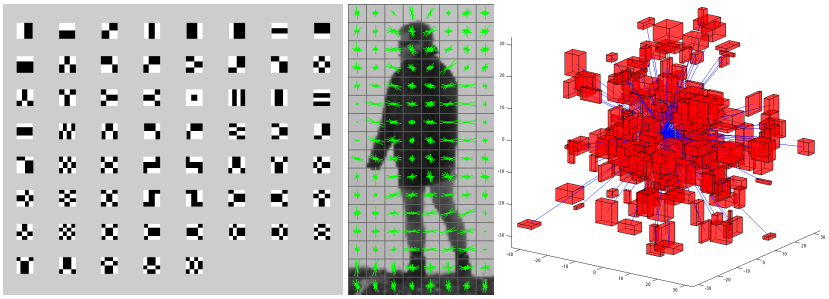

Figure 2.2: From left to right: examples of schematic representation for HAAR, HOG and box features.

varying lighting conditions, different camera gain, the presence of reflection and the image statistics are often subject to significant shifts when imaging conditions change. These problems do not affect quantitative image modalities, whose intensities have an absolute meaning, but are fairly common in many tasks using standard cameras. To solve the first problem it is possible to build invariance towards contrast and illumination changes using zero mean unit variance normalization. In this case the mean intensity $\mu = \frac{1}{K}\sum_{k=1}^{K} p_k$ computed from all the pixels of the image is subtracted from the intensity of each pixel, and the resulting values are divided by the variance $\sigma^2 = \frac{1}{K}\sum_{k=1}^{K}(p_k - \mu)^2$.

Another technique aiming to standardize all the statistics of the intensities of an image such that all the moments take predefined values is histogram equalization. A non-linear transformation is applied in order to obtain an histogram having approximately the same value for all the bins. This concept can be extended to histogram matching where the histogram of an image can be made similar to the one of another using the same reasoning.

Once the images have been pre-processed, features can be extracted. According to the definition, features must possess the following characteristics:

- Local - A feature occupies a small area of the image, thus it is robust to clutter and occlusions
- Repeatable - The same feature can be found across the images regardless the geometric and photometric changes
- Distinctive - For each feature an unique description can be created
- Robust - noise, blur, quantization, compression etc. do not destroy feature description

These features consist often of corner points, blobs, or edges in an image. They are often accompanied by descriptors which can be used to capture information about the surrounding of the region the feature has been extracted from and to make feature matching possible across different images. Also the descriptors need to be well behaved when we consider invariance and robustness. Examples of such descriptors are SIFT [103], SURF [16] BRIEF

[28] and many others that have been proposed in the last few decades by the computer vision community.

Although these features and the relative descriptors can be used for machine learning purpose, the very term "feature" has a slightly different and broader meaning in this case. Features are quantities that describe the data at hand. They don't necessarily need to possess the characteristics listed above and the aspects of the data captured by them are not always obvious. In machine learning it is desirable to have a data-set where each example is described by a collection of features which are sufficient to provide means to solve the task at hand. For example a problem where we need to recognize objects of a certain color from others of other color it might be possible to rely only on histograms as a feature sufficient to accomplish the task. In the past few years a number of handcrafted features have been proposed to solve problems such as face detection [165] and pose estimation [150]. Features such as HAAR features and box features have been successfully in a number of work, especially when they could be coupled with classifiers having feature selection capabilities, such as random forests and boosting-based approaches (Figure 2.2). Other features such as histograms of gradient orientations (HOG) have also been used in conjunction with a support vector machine to perform pose estimation. Recently deep convolutional neural networks have been employed in computer vision and have demonstrated their ability to learn hierarchical features which are especially optimized for the task at hand. Given enough training data and a well behaved and smooth loss function, optimization based on back-propagation is capable of discovering features directly from the data in order to capture both low and high level image content. In particular, early layers of a deep neural network usually capture low level vision cues, while deep layers can distinguish objects and specific patterns [178].

2.1.3 Probability

A random variable x represents an uncertain quantity. Every time x is observed its value may be different. Random variables can be continuous or discrete, bounded or unbounded. The tendency of x to assume different values, $P(x)$ can be summarized by the probability density function (pdf) in case of a continuous random variable or by a discrete density function often expressed with an histogram over the possible values in case of a discrete random variable. Both discrete and continuous representation of probability density functions must sum to one. Random variables can be considered in groups. In this case we express their tendency of assuming certain values at the same time as $P(x, y, ..., z)$. For simplicity we consider the case where two random variables are considered together, and we write $P(x, y)$. Knowing this quantity it is possible to recover the probability of x or y separately by marginalization. Marginalization is the process of retrieving the probability of one variable by aggregating its probability over every possible value of the others. In this case we write

$$P(x) = \int P(x, y)dy.$$

When we consider, for example, a couple of random variables together we may be interested in knowing the tendency of one of them to take some values when the other is fixed. In other words we are interested in knowing the probability of x given a specific value of $y = y^*$. We write $P(x|y = y^*)$ the conditional probability modeling this case. This corresponds to slicing the probability $P(x, y)$ by selecting a certain slice of values over x taken at $y = y^*$. We can therefore write $P(x|y = y^*) \propto P(x, y = y^*)$ and more precisely, by normalizing the slice such that its integral sums to one,

$$P(x|y = y^*) = \frac{P(x, y = y^*)}{\int P(x, y = y^*)dx} = \frac{P(x, y = y^*)}{P(y = y^*)}.$$

In more compact notation we write $P(x|y) = \frac{P(x,y)}{P(y)}$.

The terms of the equation above can be re-arranged to obtain

$$P(x, y) = P(x|y)P(y)$$

and by symmetry

$$P(x, y) = P(y|x)P(x).$$

Since $P(x|y)P(y) = P(y|x)P(x)$ we can write that

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

which is the Bayes rule which gives us a way to compute the posterior probability of y given what we know about x through the terms $P(x|y)$, which is the likelihood of x given y, $P(y)$ which models the prior knowledge we have about y, and $P(x)$ which is the evidence and normalizes the distribution.

When two random variables are independent the identity $P(x|y) = P(x)$ holds. The same identity can be derived also for y. One can then write that $P(x, y) = P(y|x)P(x)$ which translates into $P(x, y) = P(y)P(x)$ if the variables are independent.

When we consider a function f of a random variable x, it is possible to obtain quantities which summarize its behaviour in terms of expected value. This can be computed using the expectation operator

$$E[x] = \int f(x)P(x)dx$$

The expectation operator is a linear operator. The expectation of a deterministic quantity k is the quantity itself; the expectation of the sum of two different functions of x is the sum of their expectations; the expectation of $E[kf(x)]$ is $kE[f(x)]$ and if two variables x and y which are independent are considered, the expectations $E[f(x)g(x)]$ is the product of the expectations of the two functions. Quantities such as variance, kurtosis, etc, can be computed using this operator as they can be all defined as functions of x.

Although probability distribution functions can take arbitrary shapes and values as long as they are positive and integrate to one, there are some common distributions that are often used to model phenomena which exhibit

convenient properties when they need to be manipulated and employed in complex computations. We discuss here the Bernoulli, the categorical and the Gaussian distributions. The Bernoulli distribution can be used to model binary classification, the categorical distribution to model multi-class problems and the Gaussian distribution is important for various methods in this work.

The Bernoulli distribution is used to model phenomena that can have only a binary outcome. That is, the random variable x can only assume the value 0 or the value 1. Therefore it is a discrete, finite and uni-variate distribution. It can be written as

$$P(x) = \lambda^x (1-\lambda)^{(1-x)}$$

where λ is the probability of $x = 1$. When looking at Bayesian approaches it is useful to be able to compute a distribution for the parameter λ governing the Bernoulli distribution. For this purpose we use the Beta distribution. In this way we can model classification as a maximum a posterior estimation problem.

The categorical distribution is employed to model uni-variate random variables that can only assume K discrete values. It is similar to the Bernoulli distribution and can be expressed as

$$P(x = k) = \lambda_k$$

for each possible value of $x = 1 \ldots k$. And $\sum_{k=1}^{K} \lambda_k = 1$. In cases where it is useful to get a distribution for the hyper-parameters λ_k, the correct distribution to use is the Dirichlet distribution.

The Gaussian distribution is a uni-variate or multi-variate distribution modeling continuous random variables having unbounded values $\in \mathbb{R}$. It depends on two parameters the mean μ and the variance σ^2. It can be expressed, in the uni-variate form, as

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{(-\frac{1}{2}\frac{(x-\mu)}{\sigma^2})}$$

in case there is uncertainty about the parameters μ and σ^2 their distribution can be modeled as a normal-scaled inverse gamma distribution.

2.1.4 Classification

Classification consist of distinguishing between data instances belonging to different classes using a function that is learned in a supervised fashion from ensembles of feature vectors and corresponding labels. Such approaches rely on a trained dataset containing multiple samples organized in a matrix $\mathbf{X}$ and respective ground truth $\mathbf{Y}$. An example of classification consists of distinguishing normal medical images from those exhibiting abnormalities.

More formally, we can learn a discriminative model implemented by the function $f(\mathbf{x})$ capable of associating feature vectors $\mathbf{x}$ to probabilities $P(l|\mathbf{x})$ over the set of discrete labels $l \in \{1...K\}$.

Logistic regression is a discriminative model often used in classification. It enforces a linear decision boundary in feature space. In the following we will restrict ourselves to the case where we want to classify n-dimensional features

vectors $\mathbf{x_i} \in X$ in two classes $l = \{0, 1\}$. This situation can be modeled using the Bernoulli distribution introduced previously

$$P(l|\mathbf{x}, \theta) = \lambda_\theta(\mathbf{x})^l(1 - \lambda_\theta(\mathbf{x})))^{(1-l)},$$

where

$$\lambda_\theta(x) = \frac{1}{1 + exp(\theta^T \cdot \mathbf{x})}.$$

The set of parameters θ is learned through optimization by relying on the training set and the corresponding ground truth annotation. The training pairs are assumed to be independent. The solution to the learning problem amounts to finding the best parameters θ and can be sought, for example, via Maximum Likelihood Estimation (MLE).

In Maximum Likelihood Estimation we seek to maximize the likelihood $P(l|\mathbf{x}, \theta)$ with respect to θ over the whole training set $\{\mathbf{X}, \mathbf{L}\}$. Under the assumption of sample independence this corresponds to

$$P(\mathbf{L}|\mathbf{X}, \theta) = \prod_{i=1}^{N} \lambda_\theta(\mathbf{x}_i)_i^l(1 - \lambda_\theta(\mathbf{x}_i)))^{(1-l_i)}.$$

We can at this point use the expression for λ_θ and turn the product into a sum by taking the logarithm of the whole expression. Although this operation changes the value of the function over its domain, it doesn't change the location where its maximum is achieved. In this way we can obtain a more tractable expression for the derivative of the MLE expression having replaced the product with a sum.

The derivative can be written as

$$\frac{\partial L}{\partial \theta} = -\sum_{i=1}^{N}(\frac{1}{1 + exp(-\theta \mathbf{x}_i)} - l_i)\mathbf{x}_i$$

and the solution to the learning problem in a MLE sense can be achieved by gradient descent until convergence.

More complex classification approaches have been proposed in the past decades and although no single classifier is capable of solving every problem optimally, it is possible to achieve level of accuracy that surpass those of humans on specific tasks [78]. Approaches such as Logistic Regression are seen as naive when compared with more recent classification methods based on random forests [82], boosting [147] or multi-layer neural networks [45]. These approaches, which have been extensively studied in scientific literature, are capable of enforcing non-linear decision boundaries without resorting to kernel tricks [4] which are necessary when using Support Vector Machines (SVM) [37] or Logistic regression itself. Selected approaches are further discussed in Section 2.2.

2.1.5 Regression

The aim of regression is to learn a function to associate an observation $\mathbf{x}$ consisting of features organized in a vector of real numbers, with one or

more real numbers $\mathbf{w}$. Regression falls in the category of supervised machine learning approaches where a training set of data samples $\mathbf{x}_i$ organized in a matrix $\mathbf{X}$ and corresponding labels $\mathbf{w}$ contained in $\mathbf{W}$ is employed. An example of regression problem is the task of predicting the expected survival time of a patient from medical data. In this case the prediction is a continuous rather than a discrete value.

In the following we will restrict to the case where we want to predict a single real number w in correspondence of each sample $\mathbf{x}$. We want to learn the parameters θ governing the uni-variate posterior distribution $P(w|\mathbf{x})$. Since the labels are uni-variate and continuous we can resort to using the Normal distribution introduced previously in this chapter. As an example, we model this task as a linear regression problem.

$$P(w|\mathbf{x}, \theta, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp((-\frac{1}{2}\frac{(w - \mu_\theta(\mathbf{x}))}{\sigma^2}))$$

where $\mu_\theta(\mathbf{x}) = \theta^T \cdot \mathbf{x}$.

Also in this case the solution can be sought by employing Maximum Likelihood Estimation (MLE) with respect to both θ and σ and in particular by taking the logarithm of

$$P(\mathbf{W}|\mathbf{X}, \theta, \sigma^2) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} exp((-\frac{1}{2}\frac{(w_i - \mu_\theta(x_i))}{\sigma^2})).$$

The solution to this linear regression problem can be sought using gradient descent and the derivatives with respect to σ and θ of the expression shown above.

Similarly to classification, regression methods have seen considerable advancements in the last few decades and multiple approaches which go beyond linear regression have been proposed and have been demonstrated to be capable of solving complex regression problems.

2.2 Methods

In this section we discuss methods from prior art that are relevant to the approaches presented in this thesis.

2.2.1 Random trees and forests

Random trees are branching logistic regression models [137]. This model has activation

$$a_i = (1 - g(\mathbf{x}_i, \omega))\phi_0^T \mathbf{x}_i + g(\mathbf{x}_i, \omega)\phi_1^T \mathbf{x}_i$$

The function g is a (typically binary) gating function. Depending on the outcome of $g(\mathbf{x}_i, \omega)$ either the linear function $\phi_0^T \mathbf{x}_i$ or $\phi_1^T \mathbf{x}_i$ is used. The final classification outcome is produced using the parameters ϕ_0 or ϕ_1 depending on a decision made by g. Because of this formulation, when g can be learned during training, the two classifiers are specialized on a subset of the data rather than the whole set. In this way non-linear decision boundaries can be enforced.

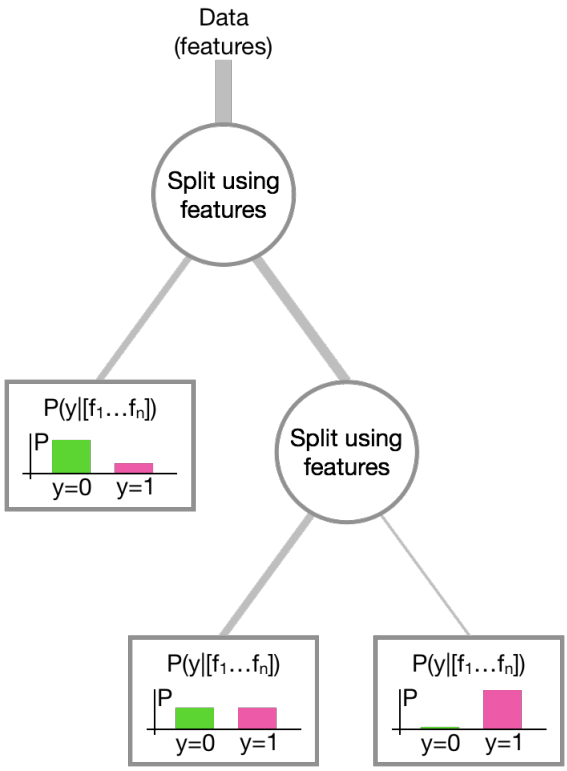

Figure 2.3: Schematic representation of the implementation of a random tree using a binary tree data structure. Line thickness represents number of data-points being routed in the tree structure. Leaves store class probabilities.

One possibility for g is to take the form of another logistic regression model having parameters ω. During learning we perform maximum likelihood estimation of the parameters $[\phi_0, \phi_1, \omega]$ using the training data pairs $\mathbf{x}_i, y_i{}_{i=0}^{N}$.

It is possible to nest gating functions and obtain more and more complex classifiers that, although have more parameters and are more difficult to optimize, can produce better and better decision boundaries.

In practice, random decision forests are constructed as groups of random decision trees whose implementation is based on the binary tree data structure [40] (Figure 2.3). In many cases a gradient-free optimization strategy is chosen.

Each node in the tree can be either a splitting node or a leaf. Similarly to the example above, making use of the function g, splitting nodes route data to more specialized classifiers consisting of their left or right child. Routing is performed according to the outcome of a comparison between one or a group of features, selected at random during training, and a threshold. That is, in each splitting node we implement the behavior of a very simple linear classifier that is very similar to g. Once the tree has reached the desired depth or the number of data-samples resulting from the previous split has fallen below a threshold, a leaf is instantiated. A Leaf typically stores the proportion of labels from different classes of the data-points that reached it. Of course more

complex information can be stored. This information may consist of Hough votes as shown in [62]. A single random tree can enforce non-linear decision boundaries during classification. These boundaries are not always optimal and sometimes can overfit the training data. In this sense, a random forest exhibits better performances by fusing the outcome of multiple random trees to obtain smoother and more precise boundaries. Further explanations about random forests can be found in Section 3.3.

Random forests represent a leap forward in the performances of computer vision approaches based on machine learning [40]. They are able to do feature importance selection in an automatic way, since splits are often based on features that can best categorize the data at hand, and to obtain very complex decision boundaries using a simple and efficient implementation. Some drawbacks of this technique are the necessity to process the whole dataset at once, which can be limiting in cases where the number of data-points is extremely high, and the limited ability to perform online training.

2.2.2 Hough transform

Hough transform is a well known technique [75] which allows detection of object instances from a certain class of shapes even in presence of noise and artifacts. In computer vision, Hough transform has been used in its most classic formulation to detect lines, circles and ellipses from images. These shapes can be expressed analytically using a few parameters that describe them completely. Images are usually processed through convolutional operations to extract gradients which convey information about edges and object boundaries in the scene [54]. The resulting information is then used to accumulate Hough votes in parameter space and therefore discover instances of these shapes in the image plane.

For example, a line can be parametrized as $y = mx + b$ or, for better behaviour during computation, as $r = x * cos\theta + y * sin\theta$ using the Hesse normal form [54]. The parameter space, in this case, is the two dimensional space of either the parameters m and b or, for the Hesse normal form, the parameters r and θ. The Hough transform works by analyzing each pixel of the image (or feature-map). If the gradients extracted from the image show enough evidence of the presence of a portion of a line at that pixel, the parameters r and θ are computed. These parameters are accumulated in a vote map (or accumulator), at the position r and θ. Normally each vote is associated with a weight that conveys information about the strength of the evidence supporting it. In practice the parameters space is discretized into "bins" in order to represent the continuous parameters r and θ in the computer system. After each pixel of the image has been analyzed, voting is complete. At this point the vote map is processed and local maxima (peaks) are identified. These positions in parameter space correspond to line instances in the image.

The Hough transform in its classic implementation has several limitations. First of all, it can only detect basic shapes that can be expressed analytically. When a larger number of parameters is used to express more complex analytic shapes, a good trade-off between the size of the vote map and bin granularity

may be hard to find. When the vote map is very large, votes will accumulate more sparsely and peaks in the parameters space might become harder to distinguish from the background noise. Large vote maps occupy more memory, which can be problematic when the limit of available computational resources is reached. Coarse vote maps are smaller, since they contain less bins, but may determine imprecise results or a high number of false positives.

A "generalized Hough transform" has been introduced in [10] with the aim of allowing detection of arbitrary shapes which cannot be expressed analytically. In this case, a template shape can be detected in presence of translations, rotations and shape changes by making use of a parametrization relying on a correspondence table, called R-table, whose rows contain votes extracted from the template edges. The votes are grouped by orientation of the template edges. The parameters space is 5 dimensional, since it needs to model 2D translations, rotations, and scale changes in 2 orthogonal directions.

A reference point $\mathbf{y}$ is chosen within the template that needs to be recognized. The edges of the template are extracted, and in particular their orientation ϕ_i at each point is taken into consideration. The vector $\mathbf{r}_i$ joining every point on the edges of the template and y is then expressed with respect to ϕ_i and added to the R-table at the ϕ_i-th row. Detection is performed by extracting edges and orientations from the image and hand, and by using the r-table to cast votes in parameter space. Votes accumulate in a clear peak only when the correct scale and rotation is considered and only for objects whose edges are similar to the ones extracted when analyzing the reference template.

This approach is robust to partial occlusions and slightly deformed shapes. It can also tolerate noise and additional structures in the images. Multiple occurrences of the same object can be retrieved by processing the vote-map appropriately. The computational and storage requirements for this algorithm may be very high.

More recently the generalized Hough Transform has been paired with random forests [62]. This technique has proven to deliver better performances than other previous works refining the idea presented in [10]. The discriminative power of random forests, applied to both pixel classification and vote grouping, have further increased the effectiveness of generalized Hough voting and its robustness towards occlusions, deformations and other challenges in computer vision.

2.2.3 Convolutional neural networks

Neural networks have been introuced in [108]. A neural network makes use of building blocks called neurons which implement the linear function $y = f(\mathbf{w}^T\mathbf{x} + b)$ that produces an output y through the activation function f having as input the dot product between the parameters $\mathbf{w}$ and the inputs $\mathbf{x}$ plus the bias term b. When the activation f is a sigmoid, this corresponds to a linear classifier similar to the one described in 2.1.4.

The function implemented by an artificial neuron remotely resembles the behavior of biological neurons which activate their output synapse when the sum of the activation of their inputs surpasses a certain threshold. When

one or more hidden layers of neurons are present they have the potential to approximate any non-linear function. As a result, Neural networks are universal function approximators [44]. This kind of architecture takes the name of "fully connected" structure because each input is connected to each of the neuron of the first layer, and every neuron of any subsequent layer is connected to all the outputs of the previous, until the output of the network is reached.

Multi-layer neural networks can be trained using gradient back-propagation, which is a technique discussed in [98]. The first step of this stategy is the forward pass. The operations implemented by the layers of the neural network are performed in a cascaded fashion (layer after layer), and the objective function is computed. At this point the gradient of the loss with respect to the parameters of the network is back-propagated throughout the layers and applied to the parameters of the model.

As different layers of neurons are stacked together the theoretical learning capacity of the network increases but the number of total parameters of the model grows quickly. This becomes unpractical for problems such as computer vision and image understanding where the input dimensionality is extremely high already, and thus number of neurons required to tackle the task is very large. As a result, non-convolutional neural networks cannot be used to end-to-end image recognition tasks due to practical problems related to number of parameters of the model, overfitting, difficulties during optimization, etc.

In order to limit these issues, convolutional neural networks have been proposed [98]. The connections of the neural network architecture have been organized such that the only a small number of model parameters would be used in each layer and shared among different portions of the inputs. In convolutional neural networks, each convolutional layer convolves the activations from the previous with a small kernel containing model parameters. This operation is repeated layer after layer until a different kind of layer or the output of the network is reached.

Convolutional neural network, due to their structure, have the capability of learning hierarchical features directly from the training data [178]. This is directly linked with the concept of receptive field. Let us suppose that the convolutional kernels used in subsequent convolutional layers throughout the network have a size of $K \times K$ pixels. The first layer will be able to perceive a $K \times K$ region of the input image, while the subsequent layer will be able to perceive a region as big as $K + \left\lfloor \frac{K}{2} \right\rfloor \times K + \left\lfloor \frac{K}{2} \right\rfloor$. That is, the receptive field, and therefore the kinds of patterns that can be recognized, grow with the depth of the network (Figure 2.4). This is also the reason why current network architectures employ a larger number of convolutional kernels in deeper layers compared to shallower ones: those kernels are more specialized and recognize more complex patterns, therefore they are needed in greater number.

Current neural architectures do not only employ convolutional layers. Often fully connected and pooling layers are used. The first type of layer implements the same operations that early fully connected neural networks used to implement throughout the whole architecture. The second kind of operation consists of a decimation task where the size of the activation tensor resulting from the computations of the previous layer is reduced by a factor p. One of

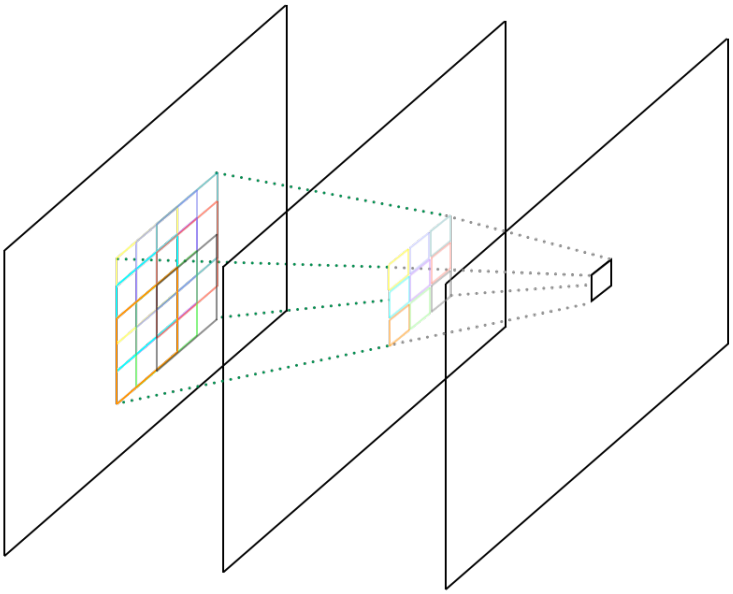

Figure 2.4: Schematic representation of the receptive field of two 3×3 kernel applied one after the other in a convolutional neural network architecture.

the most common strategy for this task is "max-pooling" which forwards the maximum value that can be retrieved by looking at a $K \times K$ window over the image applied every p pixels in each direction.

When a neural network is instantiated (both in the convolutional and fully connected cases) the parameters are initialized at random. This is very important as it serves the purpose of parameters symmetry breaking which in turn allows for a meaningful learning procedure where most convolutional kernels learn to recognize a different pattern.

Another crucial role is represented by the non-linearities employed to obtain the activation of each convolutional or fully connected layer. It is important to ensure that the behavior of the gradient of the non-linearities does not cause a gradient vanishing issue during back-propagation.

Convolutional neural networks have proven to be effective for end-to-end computer vision and image understanding tasks, for applications such as classification [93], regression [169] and segmentation [145]. Due to their formulation as a massively parallel task, it's possible to obtain efficient implementation of convolutional neural network on Graphic Processing Units (GPUs).

2.2.4 Sparse coding

We introduce here the main concepts of sparse coding [55]. Sparse coding is a technique to reconstruct a signal as a sparse combination of an over-complete set of basis functions called dictionary. Each basis functions takes the name of word or atom. This is particularly useful in fields such as compressed sensing and has proven also beneficial for computer vision applications such as visual

tracking.

Let us suppose a signal $\mathbf{y} \in \mathbb{R}^n$ and a dictionary $\mathbf{D} \in \mathbb{R}^{n \times m}$ whose columns, also called atoms, approximately span $\mathbf{y}$. The signal $\mathbf{y}$ is reconstructed as a linear combination of the words through the weights $\boldsymbol{\alpha} \in \mathbb{R}^m$ by solving the optimization problem

$$\min_{\alpha} \frac{1}{2} \|\mathbf{D}\boldsymbol{\alpha} - \mathbf{y}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_0 . \tag{2.1}$$

The weight λ controls the sparsity of the solution establishing a trade-off between least squares optimality and the number of words employed for its computation. When the weights α are constrained to be positive, the signal $\mathbf{y}$ can be reconstructed only as a conical combination of atoms.

The solution can also be sought for the dictionary $\mathbf{D} \in \mathbb{R}^{n \times m}$. Given a data-set of observations we are interested in finding $\mathbf{D}$ which can approximately reconstruct all the examples in the data-set in a sparse manner.

In practice, when the objective is to reconstruct images through sparse coding, the column of D capture low level visual features, such as edges at multiple orientations, blobs and corners.

2.3 Medical modalities

Modern medical procedures strongly rely on visual data acquired from patients using imaging techniques that are able to provide views of the inside of the body. In this way it is possible to correctly diagnose diseases, see problems or malformations and assess the progress of a treatment or the outcome of a procedure. Different medical imaging modalities have different capabilities. This is due to the fact that contrast, which is at the base of image formation, is obtained in different ways depending on the physical effect exploited by the modality. Some modalities use ionizing radiations which, depending on the dose, can be harmful for the patients but allow to obtain 2D or 3D images having high resolution and containing invaluable information for clinicians. Other techniques, such as ultrasound, use sound beams to obtain contrast from tissue interfaces and form images in real time and in an inexpensive and safe way. MRI uses a more complex physical process which exploits high magnetic fields to perturb the orientation of the atomic spins of living tissue. Although this modality is implemented in machines whose cost is extremely high and that require special facilities to be operated, the information provided by MRI is of paramount importance for clinical applications. Other modalities such of opto-acoustic tomography have been recently developed but are beyond the scope of this thesis.

2.3.1 Ultrasound

Ultrasound image formation is based on the interaction of sound waves with materials having different density and therefore different acoustic impedance. Any sound beam traveling through a medium experiences attenuation as it

travels away from its source and reflection or scattering when it encounters regions having different acoustic impedance in the medium. By varying the frequency of the sound beam it is possible to travel further distances in the medium (lower frequency) or obtain reflections and scattering for increasingly small structures (higher frequency).

When imaging the body, echoes are produced at the interfaces between different kind of tissue, due to their different density, and interfaces between tissue and air.

In order to obtain images at high enough resolution to allow clinical considerations it is necessary to apply high frequency sound beams through specifically designed transducers which require good coupling with the body. Coupling is usually provided by applying ultrasound gel to the body part that is being scanned which propagates sound at the speed of 1540 m/s. The frequency and power of the sound beam should be varied when imaging different body parts. Deep structures are often imaged at frequencies between 1 and 6 MHz, while shallow structures such as vessels and nerves are imaged at higher resolution using frequencies in the range 8 - 25 MHz.

The waves are usually produced using piezo-electric elements which are manufactured in arrays whose elements can be activated one by one or in groups. When electric tension is applied to the elements of the array they undergo deformation and therefore they can create a mechanical wave (sound wave) by repeatedly changing their shape. Piezo-electric elements can also act as sensors for sound waves. When a sound wave is applied to them they change their shape due to the mechanical force applied by the wave, and produce an electric tension at their extremities.

Image formation in ultrasound heavily relies on timings, computations and the assumption that the sound beam travels at a fixed speed through the body. This speed is assumed to be similar to the speed of sound in water as the human body consists largely of water. We give an intuitive description of ultrasound image formation. When a sound beam is produced, it propagates radially from the piezoelectric element outwards. When it interacts with an interface between media having different density an echo is produced. This echo is sensed by the elements in the transducer and a signal is recorded. Depending on *when* the echo reaches the transducer it is possible to retrieve the spatial location *where* it has occurred and by exploiting and refining this mechanism it is possible to associate raw signal to spatial locations during scanning. The image is retrieved by looking at the envelope of the raw signal, therefore taking into account the strength of the sensed echos.

In practice it is necessary to implement complex techniques in order to retrieve ultrasound images. Multiple elements of the transducer can be fired together in order to create a sound beam that has a shape and a focus in order to better trace the echos back to their spatial locations and obtain better images. This process is called beam-forming. In the past, focus has also been achieved using sound lenses placed in front of the transducer or by using transducer having a specific shape.

Ultrasound can be used in multiple different ways that are conventionally indicated by A-mode, B-mode, M-mode and Doppler. A-mode, where A stands

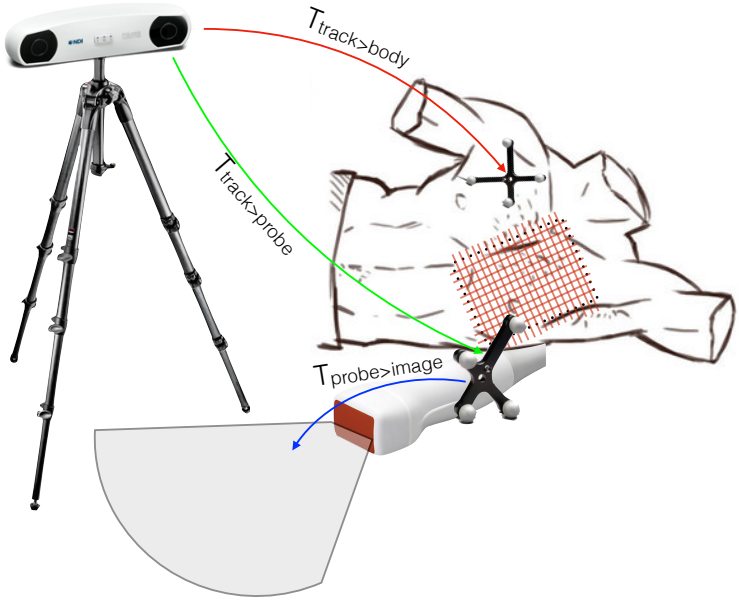

Figure 2.5: Schematic representation of freehand ultrasound volume acquisition.

for amplitude, shows a plot of the sensed echoes along a single scanning line through the body as a function of depth. B-mode stands for brightness mode and consists of grayscale images in 2D or 3D acquired through either a mechanically swept transducer or a 1D (producing 2D images) or 2D (producing 3D volumes) transducer array. M-mode stands for motion mode which usually produces high temporal resolution scans along a specific scan line which are assembled into a grayscale image showing the motion of the structures encountered along the line over time. Doppler uses frequency shifts due to Doppler effect to sense motion inside the body. This is especially useful when acquiring vascular images in order to assess the blood flow. Due to the motion of the blood in the vessels the echoes produced by the liquid are shifted in frequency. This is normally rendered with a color overlay on the grayscale B-mode images produced by the scanner.

Freehand ultrasound

In the previous section 2D and 3D B-mode ultrasound was briefly discussed. It has been stated that B-mode images can be acquired using mechanically sweeping transducers. A single transducer can be mechanically swept to create a 2D image by assembling the signal relative to different scan lines into an image due to the fact that its motion is known. 3D images can be also assembled by mechanically wobbling a 1D transducer.

From these considerations it seems intuitively clear that as long as we can put in relation to each other different 2D scans, we can also build a 3D image. This is the concept behind freehand ultrasound.

If we have means to track the pose of the transducer with respect to the patient, we can acquire a collection of 2D images by swiping through the whole region of interest, assess the physical boundaries of the volume that we have scanned, and interpolate brightness values for the whole area of interest in a voxel-by-voxel fashion. It is necessary to note though that when we image different structures from different angles we may obtain very different brightness signals in ultrasound. This is due to the fact that echoes will be reflected differently by differently shaped interfaces in tissue and maximum brightness will be achieved when the transducer is placed perpendicular to an interface. Moreover, when bones or air is present sound waves do not propagate anymore through the medium as they are completely reflected by the interface and instantly absorbed thereafter. This creates signal drop and shadow regions in 2D ultrasound which have different shape and characteristics depending on the imaging angle of the probe. Both these factors create additional challenges when it comes to 3D volume reconstruction.

We usually track the probe and the body of the patient using a high precision optical tracking and assemble our volumes using off-the-shelf software such as the PLUS framework [95]. This is summarized in a schematic form in Figure 2.5. By capturing the spatial transformation between the tracker and the body and the tracker and the probe it is possible to find the position and rotation of the probe with respect to the body at any given time. The relationship between the probe and the image it produces needs to be discovered by calibration of the data acquisition system prior to scanning. In this way the voxels of the reconstructed volume can be expressed in patient coordinate frames where both an origin and a meaningful metric for measurements are defined.

Some anatomies are more suited than others to be scanned via freehand ultrasound. Images of organs that do not undergo voluntary or involuntary motion are easier to acquire. Recently [135] freehand acquisition of transcranial ultrasound images of the brain has been demonstrated. In that case and in a handful of similar situations images can be acquired in a straightforward manner.

In other cases, deformations induced by the probe swiping motion itself can be present [179]. In those cases it is often possible to ignore these imprecision as long as the underlying model is robust enough.

When dealing with anatomies that are subject to frequent voluntary or involuntary motion it is necessary to either account for this effect using prior information and optimization, in a sort of "bundle-adjustment" strategy [166], or, when the motion is periodic or known, gate the acquisition process in order to obtain images that do not exhibit motion [79].

2.3.2 MRI

Magnetic resonance imaging (MRI) is a widespread modality used by clinician to acquire tomographic images of the body. This modality does not use any ionizing radiation and relies on a strong static magnetic fields, an oscillating magnetic field and radio frequency signals to obtain images. MRI images have

high soft tissue contrast and relatively high resolution in arbitrary orientation. Moreover MRI is applicable in children and has hardly any side effects. The source of contrast in this modality is the concentration of hydrogen atoms contained in the molecules making up the human body.

Hydrogen atoms nuclei have magnetism and, in particular, an atomic spin vector. These vectors behave similarly to compass needles when they are placed in the static magnetic field of the MRI machine. That is, they align with the field. The spin vectors have also an angular momentum. When the orientation of the atomic spin vectors is temporarily perturbed it is possible to observe the effect of the angular momentum as they re-align with the static magnetic field to equilibrium. This is a phenomenon called precession which can be observed only while the spin vectors re-align towards equilibrium. Precession can be sensed using coils which pick up the signal produced while the spin vectors re-align. By applying an oscillating magnetic field orthogonal to the static magnetic field at the appropriate resonance frequency, it is possible to temporarily perturb the orientation of the spin vectors and bring them out of equilibrium. At that point precession is induced and a signal can be obtained.

Hydrogen nuclei have two equilibrium states, spin up and spin down. The angular momentum of the atoms having spin up and down cancel each other out. It is therefore necessary to have a different proportion of atoms having different spins in order to create a net atomic spin axis during scanning. When operating at room temperature there is only a slight preponderance of the spin up state which results in a visible spin axis pointing upwards. Increasing the magnitude of spin axis can be achieved lowering the temperature of the specimen towards absolute zero or by increasing the static magnetic field of the MRI apparatus. The magnitude of the spin axis determines the magnitude of the signal sensed by the RF coil in the machine. For some applications it is desirable to image using higher magnetic fields or lower specimen temperatures. The magnetic field strength also influences the precession frequency which is proportional to the static field strength B_0 through the Larmor equation $\omega = \gamma * B_0$, where γ is a quantity specific to the chemical element (hydrogen) whose atomic spins are being exploited during imaging. At equilibrium we have a cumulative magnetic vector which is made up by the contributions of all the individual spin vectors.

In order to produce a tomographic image it is necessary to be able to distinguish the signal coming from different portions of the specimen at hand. This can be done through precession frequency shifts induced by gradient coils in the machine. These coils slightly perturb the static magnetic field B_0 influencing ω. The gradient coils can perturb the field in three orthogonal directions. This influences the Larmor frequency as a function of space. This spatial encoding can be retrieved by observing the signal produced by precession over time, through the concept of k-space.

When we tip the cumulative magnetic vector away from equilibrium (90 degree pulse) using the transverse magnetic field, precession will be observed. All the spin vectors across the specimen being scanned will precess at the same frequency and without phase differences. Once the gradient coils are active, the magnetic field in z direction will be perturbed and the spin vectors

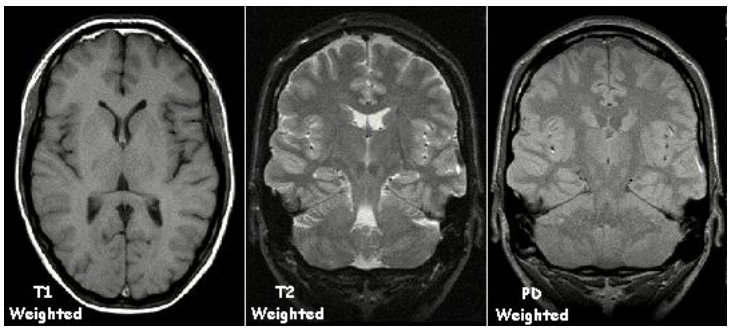

Figure 2.6: Example of MRI images. Image source: [180].

making up the magnetic vector will spin at different Larmor frequency ω_z. The spin vectors will gradually go out-of-phase with each other due to the frequency shift. As time passes this phase difference will become bigger and bigger. If we plot the phases as a function of z we obtain a sinusoidal wave whose frequency k increases with time. The obtained signal $S(k)$ therefore corresponds to the summation over the whole length z of the effects of all the spins (proportional to the number of atoms present in the specimen) times their phases. The phases can be encoded in k-notation as $exp(ikz)$ and the signal $S(k)$ (with k proportional to time) can be written $S(k) = \int \rho(z)exp(ikz)dz$. The function $\rho(z)$ can be obtained via inverse Fourier transform. These concepts can be generalized for imaging in three dimensions.

These principles have been deeply studied and refined during the last 50 years and MRI scanners have been continuously improved in order to be able to deliver better images at increased resolutions. Different imaging sequences involving complex pulse sequences have been developed by introducing different ways to drive the oscillating magnetic field of the MRI scanner. In this way different information could be obtained from the images. Very important sequences are "T1 weighted", "T2 weighted" and "PD weighted" which produce very different results using the same hardware as shown in Figure 2.6.

2.3.3 X-Ray

X-Ray imaging is a widespread diagnostic modality employed all around the world by physicians to image the inside of the human body. In this modality high energy photons are generated, through an X-Ray generator, beamed through the body part being scanned and detected by an X-Ray detector. As X-Rays travel through a medium they are scattered and absorbed. These effects lie at the core of image formation. The source of contrast for this modality is the opacity of the materials being scanned to X-Ray.

X-Ray has been discovered by the German scientist Wilhelm Roentgen who has introduced a technique to produce high energy photons at very short wavelengths (10^-11m) which, although cannot be perceived by the human eye, are able to penetrate thick solid objects.

X-Rays are generated by X-ray tubes (Figure 2.7). Although different

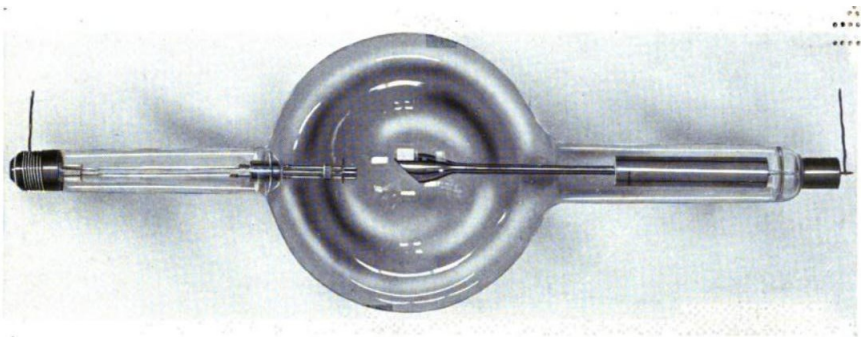

Figure 2.7: Picture of an X-Ray tube. Image source [34].

versions of such device exist, the functioning principle is common. A cathode and an anode are placed in a vacuum tube. A large difference of electric potential is enforced between the anode and the cathode which causes a current of highly accelerated electrons to flow in between. These electrons collide with the material of the anode and decelerate due to the presence of other charged particles. This effect, called Bremsstrahlung, consists of a loss of kinetic energy of the moving electrons which results in a high energy X-Ray photons being emitted.

X-Rays traveling through any medium are absorbed and scattered. The dominant effects being observed at energy windows that are typically used in diagnostic settings are the Photoelectric absorption, Compton scattering and Rayleigh scattering. The first effect consists of an interaction between the X-Ray photons and electrons belonging to the material. In this case the photon is absorbed completely and the interacting electron is kicked off its atom. Vacancy is subsequently filled by another electron of the outer shell of the atom which leads to an energy emission called X-Ray fluorescence. During Compton scattering, the X-Ray photon collides with electrons of the outer shell, loses energy, and continues to travel on a different course while the interacting electron is kicked off the atom. In Rayleigh scattering the photon deviates its course without an energy transfer. The accumulated effect of these interactions between X-Ray photon and matter is the source of contrast in X-Ray imaging and can be approximately summarized by the transmission equation $I = I_0 exp(-\mu x)$ where I_0 is the initial beam intensity, μ is an absorption coefficient specific for each material, x is the distance traveled by the beam.

Detection of X-Ray can be done using X-Ray films, even though new techniques capable of acquiring digital images have been recently employed. Image intensifiers or flat panels detectors are currently employed in most of the X-Ray systems around the world with a bias towards image intensifiers which are more artifacts prone but have a lower cost.

X-Ray images are currently used both during diagnosis and intervention. Applications range from orthopedy to angiography and include interventional uses such as X-Ray fluoroscopy which is essential to provide guidance during minimally invasive procedures such as trans-catheter surgeries.

Chapter 3

Segmentation

3.1 Introduction

Segmentation is important for both computer vision and medical image analysis applications. Segmenting an image or a volume consists of delineating regions of interest that have a specific meaning to the user in relation to the task at hand. For example, in computer vision, researchers are often interested in delineating all the objects that occur in an image, in order to distinguish them from the background and enable further tasks such as robot interaction or accurate object localization and classification.

Similarly, in medical image analysis we are often interested in delineating anatomical detail that are relevant to a diagnostic or surgical or treatment planning task. More specifically, segmentation enables:

- Extraction of quantitative measurements that are indicative of the presence or absence of a medical condition;
- Pre-operative planning, where the precise delineation of regions of interest allows surgeons to intervene more accurately and more effectively on the patient;
- Computer aided interventions, where segmentation is used for tasks such as navigation, real-time motion compensation and deformable registration;
- Extraction and enhancement of specific information from the image in order to enable further tasks.

Formally, segmentation can be seen as the problem of assigning a label to each pixel or voxel in the image [73]. Having N labels, we obtain an exhaustive N-partitioning P of the image I consisting of N regions R_n, where each pixel is assigned exactly to one region and there are no pixels that have not been assigned. Therefore

$$P = \{R_1, ..., R_N\} : \cup_{i=1}^{N} R_i = I.$$

Manual segmentation of medical images is tedious, time consuming and subjective. When different experts are asked to trace the boundary of a certain organ or structure of interest, their response can exhibit variations that are sometimes very significant. These variations depend on the training that each individual received, his or her particular area of speciality and the guidelines followed by their hospital or organization.

Despite the fact that the subjectiveness of manually traced segmentations poses challenges to automated delineation methods, by observing how experts react to ambiguous situations, we determine that prior anatomical knowledge is crucial to obtain high-quality results. As previously stated, medical images are often noisy and corrupted by artifacts, and portions of anatomical details are often not visible or their appearance is corrupted. Both human and computer aided approaches must be aware of this issue and react appropriately. Doctors apply the knowledge they have been incorporating through extensive training, while machines make use of shape models which have been implemented in a number of different ways in virtually all the most successful segmentation methods introduced recently.

In particular, segmentation has been implemented using:

- Methods incorporating explicit shape priors, via principal component analysis (PCA), produce contours by linear combination of the principal modes of variation of the training data. In this scenario the coefficients of the linear combination are obtained via optimization, taking into account the image content. The resulting shapes are therefore plausible. Methods based on "snakes" or level-sets belong to this group.

- Methods based on atlas where one or multiple manually annotated images are employed to segment novel data by deformation and label propagation from the atlas itself to the new scan. Most brain segmentation approaches belong to this category.

- Methods based on implicit shape priors, where it is not necessary to explicitly parametrize the contour and capture its modes of variation, but a shape prior is imposed by forcing the method to produce plausible boundaries by imposing the need for symmetry for example or the need to respect some distance constraints between points locally. Voting based approached impose these kind of constraints.

Another important aspect is the nature of the data. Medical data can be volumetric, when dealing with tomographic modalities (PET, MRI, CT, 3D ultrasound), or 2D when dealing with X-Ray, 2D ultrasound, endoscopy data and other modalities.

When dealing with 2D ultrasound data the main problem is represented by the fact that only a cut-through the anatomy of interest can be seen in one image and therefore only part of the anatomy might be visible simply because other parts are not in the current scan plane. This poses huge challenges when accurate measurements are needed for example as the accuracy of the measurements depends on the skills of the sonographers who need to find a suitable sonic window that guarantees a correct view. In X-Ray we see a

projection of all the structures of the body onto a plane, and therefore we face challenges when we need to discern structures at different depths. On the other hand 2D data is easier to process due to its limited dimensionality, compared to volumetric data, and therefore many researchers have proposed to treat 3D data as a collection of 2D planes and process each plane separately.

Volumetric data should be processed taking its nature into account. Main challenges in volumetric data is the limited resolution that is often a problem in MRI or 2D freehand ultrasound, and the data dimensionality which demands for higher computational resources and processing times. Finally, 3D data is sometimes more difficult or expensive to acquire and most importantly annotate than 2D images.

In the works presented in this chapter we will mainly focus on volumetric segmentation using approaches that scale well both in terms of computational resources, need for annotated training examples and processing time.

3.2 Related Work

Recent techniques to perform computer aided segmentation rely on a number of different methodologies such as deformable models, graph based models, single and multi atlases, and machine learning methods. Both automatic and semi-automatic methods have been proposed although, due to the potential advantages of the firsts, major efforts have been spent in the last few years to eliminate any need for human interaction from segmentation algorithms.

Approaches based on deformable models rely on optimization-based mathematical formulations and, often, on statistical prior knowledge to ensure robustness of the results. The cost functions being utilized by such methods are usually based on local intensity gradients, texture, region intensities or speckle statistics [125]. Exemplary works in this category are [1, 35, 70, 119] where the parameters defining the segmentation curve are optimized to fit the image content while taking into account prior information about the expected final shape, as well as [38, 39, 160] where a variational approach based on level sets is proposed.

Methods employing shape and appearance models often require a difficult and time-consuming training stage where the annotated data must be carefully aligned to establish correspondence across shapes in order to ensure the correctness of the extracted statistics. PCA can then be used to build a point distribution model (PDM) by finding the principal modes of variation of the shapes across the training dataset. Segmentation algorithms can therefore rely on both image data and prior knowledge to fit a contour that is in agreement with the shape model. The resulting segmentation is anatomically correct, even when the image data is insufficient or unreliable because of noise or artifacts. These approaches are referred to as active shape models (ASM) in literature [36] and were shown to be applicable to a variety of problems. For example in [70] a statistical shape model was used to segment the cerebellum of fetuses in volumetric ultrasound, in [71] the left ventricle of the heart was delineated making use of contour optimization and prior knowledge and in [1], a hardly visible portion of the brain imaged by ultrasound through the temporal bone window of the skull was reliably segmented using a 3D active contour.

Some models take advantage of both appearance and shape models obtained through PCA. In [119], volumetric ultrasound and MRI images of the heart were segmented using 3D active appearance models (AAM).

Graph based methods have also been proposed an studied by several groups. In these formulations the images or volumes are seen as graphs where the pixels or voxels play the role of nodes and neighboring nodes are interconnected with arcs whose weight is proportional to the intensity similarity. Although they often require extensive user interaction in order to identify the source and sink nodes and solve a max-flow/min-cut problem, they have proven to be successful in a number of applications, and in particular the method presented in [67] has demonstrated astonishing capabilities in a number of segmentation tasks, yet requiring only a reasonable amount of user interaction. When employed in conjunction with automatic approaches providing initialization, these methods are capable of delivering results without

requiring any human interaction [139].

Atlas based approaches rely on label propagation from annotated images to novel data. By the means offered by deformable registration, manually annotated images and previously unseen data can be brought into spatial alignment. At this point labels can be propagated across the images and a segmentation can be retrieved. Methods relying on this concept are popularly employed for MRI image segmentation where a number of atlas design choices were explored [92] such as the number of subject that need to be included in the atlas, the choice of deformable registration algorithm, the construction of a reference template and the strategy used for label fusion. Atlas based segmentation has been also successfully employed in ultrasound image segmentation of prostate [128] and left ventricle (LV) of the heart in volumetric echocardiography [129].

Machine learning approaches have been successfully employed to solve localization and segmentation tasks both in computer vision [63, 140] and medical image analysis [52]. Fast and accurate voxel-classification for multi-organ segmentation was achieved by Montillo et al. [120], through entangled Decision Forests. The power of Random Forests was also demonstrated in localization and bounding box detection for multiple organs simultaneously, for example in CT [41] or in MRI Dixon sequences [133] Compared to that, Riemenschneider et al. propose a joint single-object localisation and segmentation using Hough Regions and Bayesian labelling of a random field [142], implicitly modelling object shape. Rematas and Leibe [140] refined this approach and proposed a unified Hough Forest framework predicting object location and fuzzy segmentation in a streamlined manner. Random Forest lesion detection in 3D transcranial ultrasound was demonstrated in [132]. Ionasec et al. combine several Probabilistic Boosting Trees and Marginal Space Learning to fit a complex aortic-mitral valve model to 4D Cardiac CT and 4D trans-esophageal echocardiography data [84].

These work motivate our first contribution to the field of medical image segmentation, which is presented in Section 3.3. In this contribution we address the challenge of segmenting the left ventricle of the heart in 3D ultrasound images exhibiting noise and artifacts.

Previous approaches have tackled this problem by using either fully automatic or semi-automatic methods. Often hard constraints on the final shape of the LV needed to be imposed. In [22] a fast semi-automatic method based on graph cut and an implicit U-shape prior has been used. After asking the user to click at the location of three specific landmarks of the LV, the 3D volume was sampled and projected to a spherical-cylindrical coordinate system. Then, a graph was made in which each node was associated to a voxel. A gradient-based quantity was then assigned to each edge. Finally, a final delineation of the LV could be achieved by graph-cut. [51] proposed another semi-automatic approach. After manual initialization, each each short-axis slice was processed independently, via a Structured Random Forest (SRF) [50]. As a result the probability of each pixel belonging to the endocardium-blood interface was obtained. Model based surface estimation was then performed through a "model-to-data" approach followed by a "data- to-model" step. In

[129] a multi-atlas, semi-automatic, label propagation method was applied. In order to facilitate finding correspondences between ultrasound images, a spectral representation of the volumes through dictionary space was used. In [13] a fully automatic workflow was proposed. Initially, the ED frame was used to estimate the contour of the LV using an ellipsoid. This was further refined using a method based on explicit active surfaces. The information was then propagated through the whole 3D+t sequence using tracking, in order to estimate the contour of the LV through the cardiac cycle.

Our machine learning based technique, relying on the power of random forests and handcrafted features, overcomes the limitations of classical active shape model (ASM) and active appearance model (AAM). We propose a novel methodology where the anatomy of interest is simultaneously localised and segmented using a voting strategy and an atlas of annotated volumes which implicitly impose appearance and shape priors.

Although handcrafted features can exhibit robustness towards the presence of noise and artifacts [85] and have been often employed to deliver automatic segmentations, they can rarely adapt well to a range of different tasks such as segmentation of different anatomies. For this reason, recent work [19] in the machine learning community focused on approaches leveraging single [33] or multi-layer [96] auto-encoders to automatically discover appropriate features from large amount of data. In particular, sparse auto-encoders with a single-layer have been proven to learn more discriminative features compared to multi-layer ones [33] when a sufficiently large number of hidden units is chosen. One of the main advantages of these approaches is the fact that they don't need supervision to accomplish the task.

These findings motivate our second contribution, presented in 3.4 which extends the first by proposing a way to overcome the limitations of handcrafted features therefore being able to adapt to different anatomies by learning adequate features from exemplary data. In this case a sparse auto-encoder is trained from a set of ultrasound volumes in order to create a bank of 3D features, which are specific and discriminative to the anatomy at hand.

In the last few years convolutional neural networks (CNNs) became very popular tools among the computer vision community. Classification problems such as image categorization [93, 154], object detection [65] and face recognition [60] as well as regression problems such as human pose estimation [17], and depth prediction from RGB data [56] have been addressed using CNNs and unprecedented results have been presented to the community. In order to cope with the challenges present in natural images, such as scale changes, occlusions, deformations different illumination settings and viewpoint changes, these methods needed to be trained on very large annotated datasets and required several weeks to be built even when powerful GPUs were employed. In medical imaging, however, it is difficult to obtain even a fraction of this amount of resources, both in terms of computational means and amount of annotated training data.

Prior to the introduction of extensive public datasets [9], at the point in time when our contribution was made, most approaches [31, 32, 76, 124, 136, 26] could only be trained on a few dozen training examples. Most networks were

applied to tasks that could be solved by interpreting the images patch-wise in a sliding window fashion. In this case, several thousands of annotated training examples could be obtained from just a few images. Dataset augmentation techniques, such as random patch rotation and mirroring, were also applied when the analyzed objects were invariant to these transformations [146, 31, 32, 76]. This is the case for cell nuclei, lymph nodes, tumor regions, and highly deformable organs such as prostate, but not for anatomic structures with regular size and local context, such as regions of the brain.

Another way to deal with little training data is to embed CNNs as core components into other methods which are well known to the community and have been previously applied to the same class of problems. A deep variational model is proposed in [139]. Their CNN is embedded into a global inference model, i.e. the CNN outputs are treated as unary potentials on a graph and the segmentation is solved via minimum s-t cuts on the predicted graph. In [161] the CNN performs 3D regression to predict an affinity graph, which can be solved via graph partitioning techniques or connected components in order to segment neuron boundaries. Active shape models are realized with CNNs in [100] via regression of multi-template contributions and object location. Variational Deep Learning was realized in [124] by combining shape-regularized level-set methods with Deep Belief Networks (DBN) for left ventricle segmentation in cardiac MRI.

In Section 3.5, we propose a novel Hough-CNN detection and segmentation approach which utilizes CNNs at its core to efficiently process medical volumes patch-wise and obtains voxel-wise classifications along with high level features –used to retrieve votes – that are descriptive of the object of interest.

In a similar spirit, works such as [141, 173] have employed Hough voting using a CNN. Their respective aim is to obtain head poses and cell locations in 2D by using the network to perform simultaneous classification and vote regression.

Fully convolutional networks (FCNN) trained end-to-end have been recently applied to 2D images both in computer vision [126, 102] and microscopy image analysis [145]. These models, which served as an inspiration for our fourth contribution in the scope of medical image segmentation, employed fully convolutional network architectures and were trained to predict a segmentation mask for the whole volumetric image at once. In [126], a pre-trained VGG network architecture [151] was used in conjunction with its mirrored, deconvolutional equivalent to segment RGB images, by leveraging the descriptive power of the features extracted by the innermost layer. In [102], three fully convolutional deep neural networks, pre-trained on a classification task, were refined to produce segmentations while in [145], a brand new CNN model, especially tailored to tackle biomedical image analysis problems in 2D, was proposed. More recently, this approach was extended to 3D and applied to segmentation of volumetric data acquired from a confocal microscope [30]. The method was trained using partially annotated data, by optimization of a weighted multinomial logistic loss layer.

In imaging modalities that are not strongly affected by shadows and signal drop regions, we found appropriate to apply methods based on FCNN to

perform segmentation. We introduce a novel FCNN architecture making use of an innovative loss layer specifically designed for segmentation tasks of MRI volumes. We propose to optimize the Dice coefficient, one of the most common measures of region overlap in medical image analysis [43], instead of more traditional loss functions. We show that a direct optimization of this objective yields better segmentation accuracy than the commonly used multinomial logistic loss function, as used e.g. in [30].

3.3 Hough segmentation forests

We propose a learning-based approach to perform automatic segmentation of 3D ultrasound images (3D-US). The segmentation contour is estimated through the use of a variant of Hough forests whose object localization capabilities are coupled with a patch-wise, appearance driven, contour estimation strategy. Our Hough-forest-based method, which builds upon [140], neither relies on complex construction of a SSM, nor on manual initializations to obtain results. An important difference in our main contribution is that we incorporate, in order to deal with the characteristics of the US images, an appearance prior enhancing the implicit shape model with constraints on the appearances of the region of interest. The performance of the proposed method is evaluated on a dataset of 60 volumes acquired from 30 patients using different equipment and settings.

3.3.1 Motivation

We apply our approach to segmentation of the left ventricle (LV) of the human heart in 3D-US volumes. This kind of images, depicting the anatomy of interest in real time, have proven to be extremely valuable for the assessment of the functionality of the left ventricle. The value of such assessment plays a central role in the prediction of outcomes, long term patient survival and for patient management [168, 127]. One of the main indicators used for evaluation of the functionality of the left ventricle of the heart is the ejection fraction (EF) which indicates the fraction of blood present in the left ventricle at the end of diastole (ED) that gets ejected through the aortic valve at the end of the systole (ES) due to the contraction of the heart. An accurate segmentation of the blood pool is extremely valuable since it allows an automatic, yet accurate assessment of the end systolic and end diastolic volume (ESV and EDV respectively) and therefore the ejection fraction.

Previously, imaging modalities such as computed tomography (CT), magnetic resonance imaging (MRI), and positron emission tomography (PET) have been employed for heart functionality assessment. Several recent studies indicated that 3D-US, as well as MRI, is particularly indicated to acquire images with a high diagnostic value. 3D-US is indeed cheaper, real time, has high temporal resolution and can be performed at the bed-side and for these reasons it might become the modality of election for this clinical application in the near future [121].

One of the main limitations of 3D-US, as well conventional 2D ultrasound, is that it produces images that are often corrupted by noise and artifacts and are not easy to interpret. One of the main reasons of these limitations seems to be the low contrast between the blood and the endocardial tissue and the presence of typical ultrasound artifacts. Our method addressed some of these issue by implementing an accurate, fully automatic approach for volumetric segmentation of the LV. Our approach consists of two steps. In the first step a random forest is used to do a initial estimation of the region assignment for each voxel. That is, voxels are classified and assigned to either background or

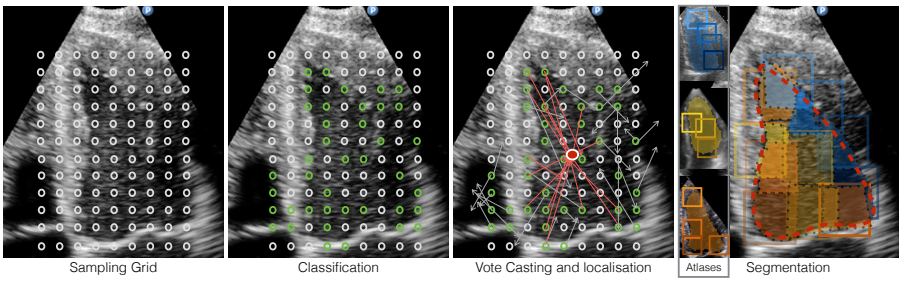

Figure 3.1: Schematic representation of our segmentation approach shown in 2D.

foreground class. In the second step, voxels that are part of the foreground cast votes in order to determine the location of the centroid of the anatomy and proceed to a implicitly-shape-constrained segmentation of the LV.

3.3.2 Method

Our approach leverages the object localization capabilities of Hough Forest [63] to obtain accurate segmentations of different organs in ultrasound volumes. Our approach comprises a training and a test phase. The localization of the left ventricle in the images is obtained in a fully automatic manner, using the Hough forest voting strategy. The contour is estimated by making use of a code-book of binary surface patches associated with the votes. That is, each vote is associated with a binary patch modeling the 3D boundary of the LV as observed during training around the location the vote originated from. These patches are used during segmentation to estimate the final contour as a superimposition of their contributions. Each contribution to the contour is weighted considering local appearances and added to the final contour. A schematic representation of this approach is show in Fig. 3.1.

Preprocessing

We normalize the spatial resolutions of the 3D-US volumes by re-sampling them to a common spacing and then equalize their intensity histograms via standard histogram equalization. In ultrasound images, high contrasts are produced locally by tissue interfaces such as the ventricular wall. Therefore, we define the foreground region as a narrow band around the annotated contour boundary and the rest of the image as background, represented with the class labels $c \in C = \{f, b\}$.

Hough forests differ from simple random forests because, beside the classification outcome, they provide means to localize object instances through a voting strategy. In our approach, we couple the voting strategy with a code-base of segmentation patches and associated intensity patches, enabling the forest to natively estimate a segmentation contour [140]. During testing the ventricle center will be retrieved through the voting strategy. After its location is estimated, all votes that didn't accumulate in its neighborhood will

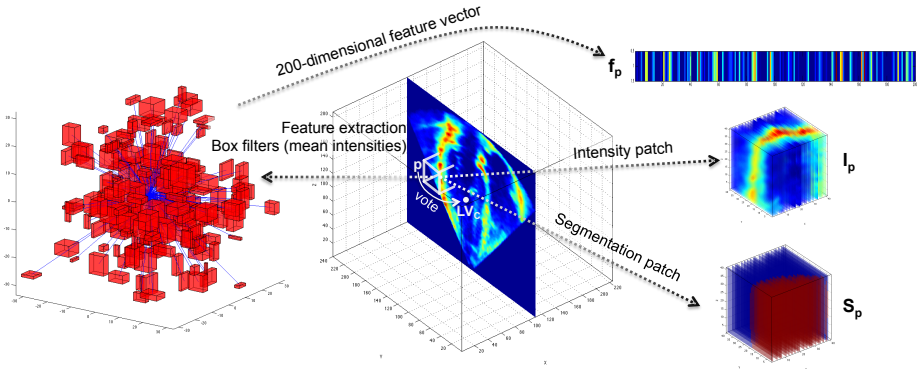

Figure 3.2: Schematic representation of the information carried by each training data-point.

be filtered out, making the subsequent segmentation step faster and more effective.

Feature extraction

After re-sampling and intensity standardization, we extract features from volume patches extracted from a regular grid imposed over the LV ultrasound volume. The same kind of features are used both during training and testing.

N-dimensional feature vectors $\mathbf{F} \in \mathbb{R}^K$ associated with each point in the regular grid are extracted by applying a bank of K box filters within a radius R_f around each sampling position, obtaining mean intensities over randomly displaced, asymmetric cuboidal regions similar to Criminisi et al. [42] (see Fig. 3.2 left). Only one component F_k of the vector $\mathbf{F}$ is used in each of the splitting nodes of the random trees employed in this algorithm. This corresponds to the case where axis aligned splits are performed.

Training the Hough Forest

Our implementation of Hough Forests (HF) combines the classification performance of Random Forests (RF) with the capability of carrying out organ localization and segmentation. Differently from the classical Hough Forest framework [63], our method retrieves segmentations enforcing shape and appearance constraints.

We consider a training set composed of N data samples, $\mathbf{d}_{1...N}$, where each sample $\mathbf{d}_i = [d_x, d_y, d_z]^\top$ corresponds to a voxel of an annotated volume V_t belonging to the training set T. The annotation G_t, obtained in the form of a 3D binary mask associated with the volume V_t, determines the binary labels $l_i = \{f, b\}$ that characterize each data-point as belonging to the foreground or to the background. Foreground data-points are associated with a vote $\mathbf{v}_i = \mathbf{c}_t - \mathbf{d}_i$, which is expressed as a displacement vector connecting $\mathbf{d}_i$ to the centroid of the annotated anatomy $\mathbf{c}_t = [c_x, c_y, c_z]^\top$, obtained from G_t. (Fig.

3.2)

During training, the best binary decision is selected in each node of the Hough Forest, either maximizing the Information Gain (IG) or maximizing the Vote Uniformity (VU). The criterion is chosen at random. In each node, we compute M random features and we determine S candidate splits through the thresholds $\tau_{1...S}$. Each split determines a partitioning of the data D_p reaching the parent node in two subsets $D_l = \left\{\mathbf{d}_i \in D_p : F_k(\mathbf{d}_i) \leq \tau_s\right\}$ and $D_r = \left\{\mathbf{d}_i \in D_p : F_k(\mathbf{d}_i) > \tau_s\right\}$ reaching the left and right child nodes, respectively. The Information Gain is obtained as:

$$IG(D_p, D_l, D_r) = H(D_p) - \sum_{i \in \{l,r\}} \frac{|D_i|}{|\bar{D}|} H(D_i),$$

where the Shannon entropy $H(D) = \sum_{c \in \{f,b\}} -p_c log\left(p_c\right)$ is obtained through the empirical probability $p_c = \frac{|D_c|}{|D|}$ using $D_c = \{d_i \in D : l_i = c\}$.

The Vote Uniformity criterion requires the votes $\mathbf{v}_j^{\{l,r\}}$ contained in D_l and D_r to be optimally clustered around their respective means $\bar{\mathbf{v}}^{\{l,r\}}$:

$$VU(D_l, D_r) = \sum_{i \in \{l,r\}} \sum_{\mathbf{v}_j} \left\| \mathbf{v}_j^i - \bar{\mathbf{v}}^i \right\|.$$

Once (i) the maximum tree depth has been reached or (ii) the number of data points reaching the node is below a certain threshold or (iii) the Information Gain is zero, the recursion terminates and a leaf is instantiated. The proportion of foreground versus background points $p_{\{f,b\}}$ is stored together with the votes $\mathbf{v}_i$ and the associated original positions $\mathbf{d}_i$. The coordinates $\mathbf{d}_i$, in particular, refer to training volumes which will be used as atlases during segmentation. The reason to use both vote uniformity and information gain criteria is that we want each of the leaves of our tree to convey both high confidence during classification and votes that point into approximately the same direction.

Each tree is trained with a random subset of T (circa 70%), which is recursively split in each node until a termination criterion is met and leaf nodes are established.

Segmentation via Hough Forests

Given an ultrasound volume I of the test set, we first classify its voxels into foreground or background, then we allow foreground data-points to cast votes in order to localize the target anatomy, and finally, we obtain the contour by projecting 3D segmentation patches from the atlases associated with each vote that correctly contributed to localization (Fig. 3.1).

The data-points processed in the Hough Forest are obtained through a regular grid of sampling coordinates $S = \left\{\mathbf{s}_1 ... \mathbf{s}_{N_d}\right\}$. In this way, we can reduce the computational load during testing without significantly deteriorating the results. Each data-point $\mathbf{s}_i$ classified as foreground in a specific leaf l of the Hough trees is allowed to cast the n_v^l votes $\mathbf{v}_{1...n_v^l}$ stored in that leaf during training.

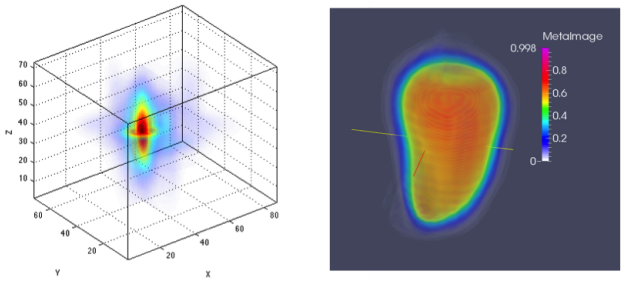

Figure 3.3: Left: votes cast by data-points classified as foreground during testing. Right: probabilistic segmentation contour.

Each vote determines a contribution, weighted by the classification confidence, at the location $\mathbf{s}_i + \mathbf{v}_j$ of a volume C having the same size as I and whose content is initially set to zero.

The target anatomy is localized retrieving the position of the highest peak in the vote map. All the votes $\hat{\mathbf{v}}_j$ falling within a radius r around the peak are traced back to the coordinates $\hat{\mathbf{s}}_i$ of the data-points that cast them. Each vote $\hat{\mathbf{v}}_j$ is associated with the coordinates $\hat{\mathbf{d}}_j$ of a specific annotated training volume.

We retrieve the 3D appearance patch A_j and the segmentation patch P_j associated to each vote by using the coordinates $\hat{\mathbf{d}}_j$ to sample the appropriate training volume and its annotation. The segmentation patches are projected at the positions $\hat{\mathbf{s}}_i$ after being weighted by the Normalized Cross Correlation (NCC) between the patch A_j and the corresponding intensity patterns around $\hat{\mathbf{s}}_i$ in the test volume. The fusion of all the re-projected segmentation patches forms the final contour, which implicitly enforces shape and appearance constraints. An exemplary qualitative result can be seen in Figure 3.3.

Using this strategy, not only we obtain a fully automatic method that does not rely on manual initialization, but we can limit the influence of false positives. In contrast to methods that perform segmentation relying on pixel-wise classification, our approach can easily discard misclassified data-points since they are unlikely to cast votes that accumulate around the actual object's center. Additionally, the intensity patches associated to each vote model the expected appearance in correspondence of the segmentation patch stored in the code-book, further helping in rejecting false-positive segmentation votes.

3.3.3 Experimental evaluation

We evaluated our algorithm on the MICCAI echocardiography segmentation challenge datasets [21]. Our method was mainly implemented in Matlab®, while some parts, like the segmentation stage, were implemented in C++. We trained a forest containing 20 Hough trees, having a maximal height of 16 and a minimum leaf population of 10 data-points. We performed two experiments, where in the first we perform a leave-one-out-cross validation on the training set, and in the second we do the training on the entire training set, and testing on the test volumes one-by-one.

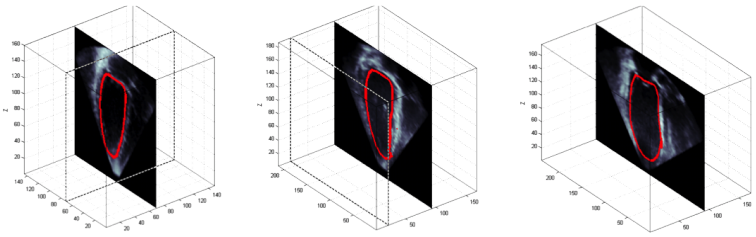

Figure 3.4: Typical contour estimates. In these pictures just one slice of the contour is depicted although the contour is estimated in 3D.

More details about the evaluation framework and the data used in this study are presented in [21].

In the first experiment, leave-one-out cross validation, in order to maintain computational efficiency, the data-points were sampled from a coarsely spaced grid, where the data-points were located at every $\Delta d = 8mm$ circa. The segmentation and intensity patches S_p and I_p stored in the code base had a size $(p_x \times p_y \times p_z)$ of approximately $18mm \times 16mm \times 13mm$. The features were extracted using a bank of $N = 200$ random box-filters that could be displaced at most $R_f = 15mm$ from the grid-points. All the votes in a radial neighborhood of $R_{LV_C} = 4mm$ from the detected centroid were considered for back-projection. The results of the cross validation, in terms of Dice coefficient, are provided in Table 3.1. In this experiment, training required circa 5 minutes while segmenting each volume took circa 6 seconds.

	MaD	HD	Dice	Min Err.
Mean ED	2.66 mm	9.01 mm	0.871	0.09 mm
Mean ES	2.43 mm	8.09 mm	0.869	0 mm
	ED Volume	ES Volume	Ejection Fraction	
Corr. Coeff	0.986	0.974	0.861	
Bias	38.42	-2.57	19.33	

Table 3.1: Evaluation of our approach on training data via leave-one-out cross validation. In the upper half of the table we present the statistics in terms of mean absolute distance (MaD), Hausdorff distance (HD), dice coefficient and minimum surface error, respectively. In the second half of the table we report the correlation coefficient, bias and limit of agreement that we achieved with respect to the clinical indices; ED volume, ES volume, ejection fraction and stroke volume.

In the second experiment, on the test set, we sampled the volumes more densely, reducing the grid point distance to $\Delta d = 6mm$ circa. All other parameters remained constant. We also compared the performances of our algorithm with the ones of the other teams participating to the challenge. In particular, we ranked among the best from the point of view of Dice overlap coefficient. These results together with a comparison with other methods is

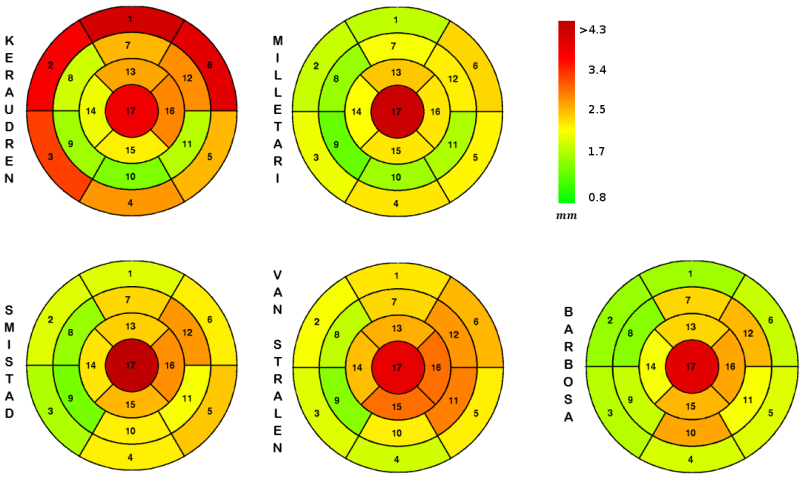

Figure 3.5: Bull eye graphs graphically representing errors attained by the methods compared in Table 3.2 in different regions (numbered following convention in [21]) of the left ventricle of the heart.

provided in Table 3.2.

Additionally, in Figure 3.5, the accuracy of our algorithm with respect to different locations of the LV is graphically rendered. The bull eye graph reported in Figure 3.5 shows 17 different regions of the left ventricle, projected onto the axial plane cutting through the LV long axis, and the error of the algorithms in those regions. We can conclude, from this data, that our algorithm is least precise in the apical region of the LV (region 17 in the bull eye graph).

The training stage required 15 minutes and one testing image could be segmented in circa 30 seconds.

3.3.4 Further applications

The technique explored in this work is particularly well suited for ultrasound volume segmentation and was employed without any modification for another work, presented in [179], whose goal was to segment the prostate in trans-rectal ultrasound scans acquired while performing computer assisted biopsies.

Currently, prostate cancer diagnosis procedures rely and sampling a number of random tissue regions through biopsy. The areas that are targeted are suspected to be affected by cancer according to pre-operative imagery. Such imagery is often acquired through MRI and PET and shows suspicious regions with reasonable contrast. During biopsy, real time guidance is be obtained by performing trans-rectal ultrasound (TRUS) scans which deliver intra-operative images of the prostate as the samples are collected. Unfortunately TRUS does not offer any contrast towards the detection of cancerous areas, therefore doctors must refer to pre-operative volumes (MRI, PET) and mentally register the images in order to target the correct areas.

In [179] we propose to automatize this process by i) obtain MRI and PET scans of the patient such that they are registered in the same coordinate

	End Diastolic (ED)		
Method	Dist.	Hausdorf Dist.	Dice
Barbosa [13]	2.26 ± 0.73	$\mathbf{8.10 \pm 2.66}$	$\mathbf{0.894} \pm 0.041$
Keraudren [91]	2.44 ± 0.95	8.98 ± 3.09	0.870 ± 0.048
Milletari [118]	$\mathbf{2.14 \pm 0.68}$	8.25 ± 3.87	$0.893 \pm \mathbf{0.031}$
Smistad [152]	2.62 ± 0.95	8.26 ± 2.98	0.885 ± 0.038
Van Stralen [163]	$2.44 + 0.91$	8.45 ± 3.50	0.879 ± 0.054
	End Systolic (ES)		
Barbosa [13]	$\mathbf{2.43} \pm 0.91$	$\mathbf{8.13} \pm 3.08$	$\mathbf{0.856} \pm 0.057$
Keraudren [91]	$2.54 \pm \mathbf{0.75}$	9.15 ± 3.24	0.842 ± 0.057
Milletari [118]	2.91 ± 1.01	$8.53 \pm \mathbf{2.30}$	0.838 ± 0.062
Smistad [152]	2.92 ± 0.93	8.99 ± 2.98	$0.844 \pm \mathbf{0.050}$
Van Stralen [163]	2.79 ± 1.24	8.65 ± 2.85	0.835 ± 0.079

Table 3.2: Evaluation of our and other approaches on the test set of 60 3D-US volumes depicting the left ventricle of the heart. Further results, including ejection fraction and volume correlations, can be retrieved from the official CETUS challenge website at address https://miccai.creatis.insa-lyon.fr/miccai.

frame, ii) obtain accurate segmentation of the prostate in MRI offline, iii) acquiring free-hand ultrasound volumes of the prostate intra-operatively, iv) segment the prostate in those volumes though our Hough forest framework, v) bring the MRI and US images into spatial agreement by relying only on the segmentations of the prostate in the respective volumes and a bio-mechanical of the organ model to improve robustness.

For the latter step, at first the surface meshes are registered together by minimizing the distance between surface vertices and computing their displacement. This alignment is performed elastically using coherent point drift between the MRI and US prostate contours, instead of relying on image data; then, the displacements are interpolated to obtain a dense deformation field that is used to warp the MRI volume onto the 3D-US scan. At this point the MRI, US and PET images can be fused together and accurate guidance can be achieved.

The freehand ultrasound volume is acquired through an external high-precision tracking system, in a coordinate frame that is aligned to the one of the patient. The biopsy needle is as well tracked by the same system. In this way it is possible to display to the user (Fig. 3.6):

- The MRI and PET scans fused together and accurately aligned via elastic deformation with the true boundaries of the prostate at the time of the intervention
- The expected trajectory of the needle on-screen, together with the images that show the suspicious regions at their true location.

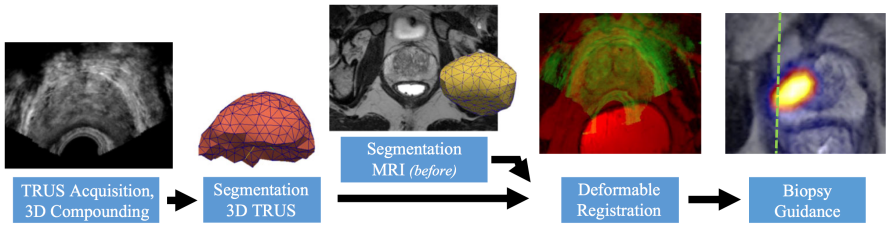

Figure 3.6: Overview of the system proposed in [179]. The pre-operative and pre-segmented MRI scan is brought into spatial alignment with the intra-operative TRUS scan by bringing into spatial alignment the respective segmentation contours. Once the images are fused together both MRI and PET images can be displayed together with the expected trajectory of the biopsy needle (greed dashed line).

3.3.5 Discussion

Our Hough forest based approach to segment 3D-US volumes in short time and without requiring any human interaction constitutes the foundation of other further approaches based on voting which have proven to be successful in a wide range of applications. This algorithm was applied to two important tasks in medical image analysis, namely the segmentation of the left ventricle of the heart and of the prostate and has been shown being suitable for real-time, intra operative usage. The main limitation of this approach can be traced back to the choice of the features used to describe the data at hand and process it through the random forest. Although random forest can perform feature selection automatically, we observed that the features used in this work were often inadequate for other applications, such as segmentation of different anatomies [179]. This motivated the work presented in chapter 3.4

3.4 Learned data representations

In this contribution, we propose a highly adaptive learning-based method for fully automatic segmentation of ultrasound volumes that leverages anatomy-specific features which are obtained through a sparse auto-encoder. This sparse auto-encoder is trained from a set of ultrasound volumes in order to create a bank of 3D features, which are specific and discriminative to the anatomy at hand.

The extracted features are employed in a Hough Forest based framework to retrieve the position of the target anatomy and its segmentation contour. Similarly to the previous contribution, presented in Section 3.3, the position of the region to be segmented is assessed through a voting strategy and the contour is obtained by patch-wise projection of appropriate portions of a multi-atlas. Again, in order to enforce both shape *and* appearance constraints, each contribution to the contour is weighted by a factor dependent on the appearance pattern of the region that it was collected from.

The resulting method is not only (i) fully automatic, and (iii) capable of enforcing shape and appearance constraints to ensure sufficient robustness, but also (ii) highly adaptive to different kinds of anatomies. We demonstrate the performance of the method for three different applications: segmentation of midbrain, left ventricle of the heart and prostate. The experiments show that our approach is competitive compared to state-of-the-art anatomy specific methods and that, in most cases, the quality of our segmentations lies within the expected inter-expert variability for the particular dataset.

3.4.1 Method

Our approach comprises a training and a test phase. During training, we discover anatomy-specific features that are employed to learn a Hough Forest. During testing, we perform simultaneous object localization and segmentation. A schematic representation of this method can be seen in Figure 3.2.

Feature Learning

Sparse Auto-Encoders are feed-forward neural networks designed to produce close approximations of the input signals as output (Fig. 3.7 - a). By employing a limited number of neurons in the hidden layer and imposing a sparsity constraint through the Kullback-Leibler (KL) divergence of the neurons firing rate with respect to a desired firing frequency, the network is forced to learn a sparse lower-dimensional representation of the training signals [33, 96].

The network has N inputs, K neurons in the only hidden layer and N outputs. The biases $b_i^{(1,2)}$ are integrated in the network through the presence of two additional neurons in the input and hidden layer having a constant value of 1. The weights of the connections between the j-th neuron in one layer and the i-th neuron in the next are represented by $w_{ij}^{(1,2)} \in \mathbb{R}$, that are grouped in the matrices $\mathbf{W}_1 \in \mathbb{R}^{K\times(N+1)}$ and $\mathbf{W}_2^\top \in \mathbb{R}^{N\times(K+1)}$. Network outputs can be

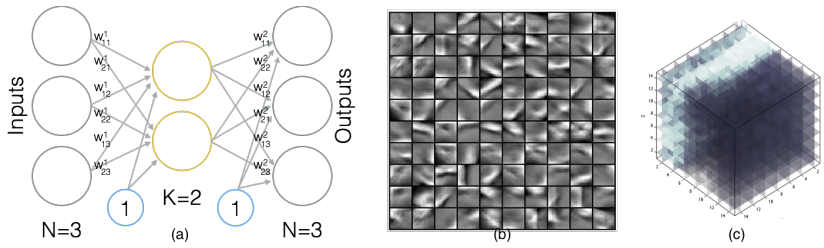

Figure 3.7: a) Schematic Illustration of a Sparse Auto-Encoder (SAE); b) Bank of feature extraction filters obtained from 2D ultrasound images of the midbrain; c) One filter obtained from 3D echocardiographical data through the SAE.

written as $h_{\mathbf{W}^{(1,2)}}(\mathbf{X}) = f\left(\mathbf{W}_2^\top f\left(\mathbf{W}_1\mathbf{X}\right)\right)$, where $f(z) = \frac{1}{1+exp(-z)}$ is the sigmoid activation function.
The matrix $\mathbf{X}$ is filled with M un-labeled ultrasound training patches arranged column-wise. After a normalization step to compensate for illumination variations through the dataset, the network is trained via back-propagation. The network weights are initialized with random values. The objective function to be minimized comprises of three terms, enforcing the fidelity of the reconstructions, small weights magnitude and sparsity respectively:

$$C(\mathbf{X}, \mathbf{W}_{1,2}) = \frac{1}{2}\left\|h_{\mathbf{W}^{(1,2)}}(\mathbf{X}) - \mathbf{X}\right\|^2 + \frac{\lambda}{2}\sum_{l=1}^{2}\sum_{k=1}^{K}\sum_{n=1}^{N}\left(w_{nk}^{(l)}\right)^2 + \beta\sum_{j=1}^{K} KL(\rho\|\rho_j).$$

In the third term, we indicate as $\rho_j = \frac{1}{M}\left(\mathbf{1}^\top f\left(\mathbf{W}_1\mathbf{X}\right)\right)$ the average firing rate of the j-th hidden neuron, and we define the KL divergence, which enforces the sparsity constraint [123] by penalizing deviations of ρ_j from ρ, as:

$$KL(\rho\|\rho_j) = \rho\, log\left(\frac{\rho}{\rho_j}\right) + (1-\rho) log\left(\frac{1-\rho}{1-\rho_j}\right).$$

The parameter ρ represents the desired firing rate of the neurons of the hidden layer, and must be set prior to training together with λ, β and K, which control the weight of the two regularization terms and the number of neurons of the hidden layer respectively.

After optimization, the rows of the weight matrix $\mathbf{W}_1$ can be re-arranged to form a set of 3D filters $\Xi = \{\xi_1 ... \xi_K\}$ having the same size as the ultrasound patches collected during training (Fig. 3.7 - b,c).

Training and Testing

The training and testing procedure in this approach is very similar to the one detailed in Section 3.3.2. The only exception, in this case, is that the features used to describe the volumetric patches are not handcrafted, but are obtained by the auto-encoder, as explained above.

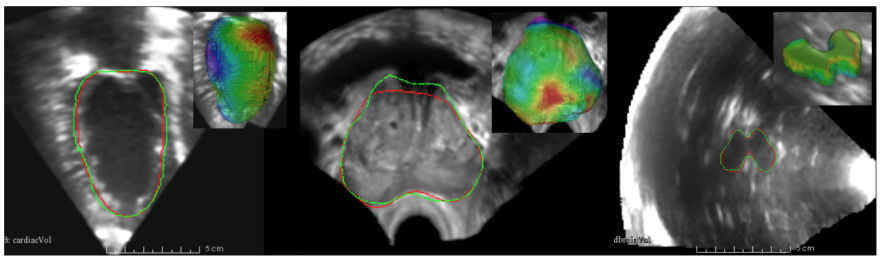

Figure 3.8: Exemplary segmentation results (green curves) Vs. ground-truth (red curves). Mesh color encodes distances from ground truth in the range -3mm (red) to $+3$mm (blue), with green indicating perfect overlap.

Each data point is therefore described by K features $F_{1\ldots K}$, which are computed by applying one of the filters ξ from the set Ξ obtained via Sparse Auto-Encoders as described in the previous step. Specifically, we write

$$F_k(\mathbf{d}) = \sum_{i=-r_x}^{r_x} \sum_{j=-r_y}^{r_y} \sum_{k=-r_z}^{r_z} V_t(d_x+i, d_y+j, d_z+k) * \xi(i,j,k).$$

3.4.2 Experimental Evaluation

We demonstrate the segmentation accuracy and the flexibility of our algorithm using three datasets of different anatomies comprising in total 87 ultrasound volumes. A brief description of each dataset and the relative state-of-the-art segmentation approach being used for comparison is provided below.

1. The left ventricle of the heart is segmented and traced in [14] using an elliptical shape constraint and a B-Spline Explicit Active Surface model. The dataset employed for our tests, comprising 60 cases, was published during the MICCAI 2014 "CETUS" challenge. Evaluations were performed using the MIDAS platform[1].

2. The prostate segmentation method proposed in [138] requires manual initialization. Its contour is retrieved using a min-cut formulation on intensity profiles regularized with a volumetric size-preserving prior. We test on a self-acquired trans-rectal ultrasound (TRUS) dataset comprising 15 subjects. All the volumes were manually segmented by one expert clinician via 'TurtleSeg'. Our results are obtained via cross-validation.

3. Segmentation of the midbrain in trans-cranial ultrasound (TCUS) is valuable for Parkinson's Disease diagnosis. In [1], the authors employed a discrete active surface method enforcing shape and local appearance constraints. We test the methods on 12 ultrasound volumes annotated by one expert using 'ITK snap' and acquired through one of the skull bones of the patients. Our results are obtained via cross-validation.

[1]Documentation under: `http://www.creatis.insa-lyon.fr/Challenge/CETUS`

Table 3.3: Overview of Dice coefficients and mean absolute distance (MAD) achieved during testing. Inter-expert-variabilities (IEV) are also reported. MAD was not provided by the authors of the algorithms used for comparison.

Dataset	Avg. our Dice	MAD (mm)	State-of-art Dice	IEV (Dice)
Left Ventricle	0.87 ± 0.08	2.90 ± 1.87	0.89 ± 0.03	86.1%[14]
Prostate	0.83 ± 0.06	2.37 ± 0.95	0.89 ± 0.02	83.8%[138]
Midbrain	0.85 ± 0.03	1.18 ± 0.24	0.83 ± 0.06	85.0%[1]

Table 3.3 shows the performance of our method in comparison to the other state-of-the-art approaches on the three datasets. Results are expressed in terms of Dice coefficients and mean absolute distance (MAD) from ground truth annotation. Typical inter-expert annotation variability is also shown for each anatomy.

Parameters of the Model

The Sparse Auto-Encoder was trained to obtain $K = 300$ 3D filters having size $15 \times 15 \times 15$ pixels, with parameters $\lambda = 10^{-4}$, $\beta = 10$ and $\rho = 10^{-3}$. The Hough Forest includes 12 trees with at most 35 decision levels and leafs that contain at least 25 data-points. During testing, the images were uniformly sampled every 3 voxels. All the votes accumulating in a radius of 3 voxels from the object centroid were reprojected. The size of the segmentation and intensity patches employed for reprojection during segmentation was different for the three datasets due to the variable size the object of interest. Values for left ventricle, prostate and midbrain were $35 \times 35 \times 35$, $30 \times 30 \times 30$ and $15 \times 15 \times 15$ pixels respectively.

Training time for the Auto-Encoder was approximately 24 hours per dataset, with 500,000 patches. The training time for the forest ranged from 20 minutes to 5 hours. The processing time during testing was always below 40sec. per volume.

In Fig. 3.9 we show the histogram of Dice scores observed during our tests. Its resolution is 0.05 Dice. Additional results can be found in Table 3.3 and Fig. 3.8.

3.4.3 Discussion

Localization of the target anatomy through a voting strategy, removes the need for user interaction while being very efficient in rejecting false positive data-points, whose votes could not accumulate in the vicinity of the true anatomy centroid. During our tests, only one out of 87 localizations failed, resulting in a wrong contour. A trade-off between appearance and shape constraints can be set choosing the size of the segmentation patches. Bigger patches force smoother contours, while smaller ones lead to more adaptation to local volume contents. The method is not suited for segmentation of elongated structures.

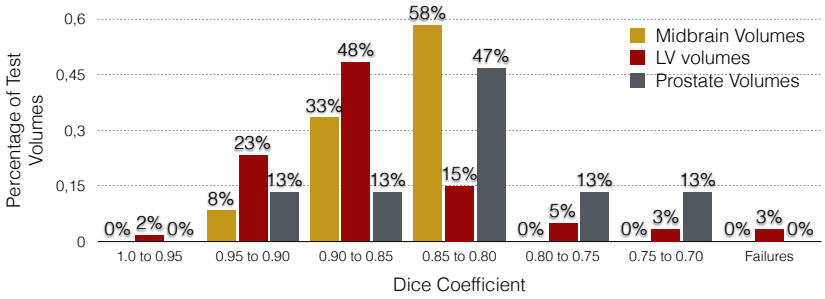

Figure 3.9: Percentage of test volumes vs. Dice coefficient. This histogram shows the percentage of test volumes falling in each Dice bin on the horizontal axis.

3.5 Convolutional neural networks and voting

In this contribution, we propose a flexible voting mechanism similar to the one proposed in Section 3.3 and 3.4 but based on neighborhood relations between the features computed by the CNN. On one hand, this allows us to cast, for each patch, a variable amount of votes that can be associated with additional information such as segmentation patches, therapeutic indications or diagnostic information which may be added or modified at any time without requiring re-training. On the other hand, by using votes collected from annotated training images and stored as displacement vectors instead of relying on regression, our method exhibits robustness to inputs that strongly differ from those observed during training.

3.5.1 Motivation

Recent research has shown the ability of convolutional neural networks (CNN) to deal with complex machine vision problems: unprecedented results were achieved in tasks such as classification [93, 154], segmentation, and object detection [155, 149], often outperforming human accuracy [78]. CNNs have the ability of learning a hierarchical representation of the input data without requiring any effort to design handcrafted features [97]. Different layers of the network are capable of different levels of abstraction and capture different amount of structure from the patterns present in the image [178]. Due to the complexity of the tasks and the very large number of network parameters that need to be learned during training, CNNs require a massive amount of annotated training images in order to deliver competitive results. As a consequence, significant performance increase can be achieved as soon as faster hardware and higher amount of training data become available [93].

In this work we investigate the applicability of convolutional neural networks to medical image analysis. Our goal is to perform segmentation of single and multiple anatomic regions in volumetric clinical images from various modalities. To this end, we perform a large study on parameter variations and network architectures, while proposing a novel segmentation framework based on Hough voting and patch-wise back-projection of a multi-atlas. We demonstrate the performance of our approach on brain MRI scans and 3D freehand ultrasound (US) volumes of the deep brain regions.

The better results delivered by CNNs in computer vision were in part accomplished with the help of extremely large training datasets and significant computational resources. Both of which may be often unrealistic in clinical environments, due to the absence of large annotated dataset and to data protection policies which often do not allow computation outsourcing. Therefore, in this study, we perform all training and testing of CNN networks on clinically realistic dataset sizes, using a high-performance, but stand-alone PC workstation.

Segmentation of brain structures in US and MRI has widespread clinical relevance, but it is challenging in both modalities.

In MRI, the segmentation of basal ganglia is a relevant task for diagnosis,

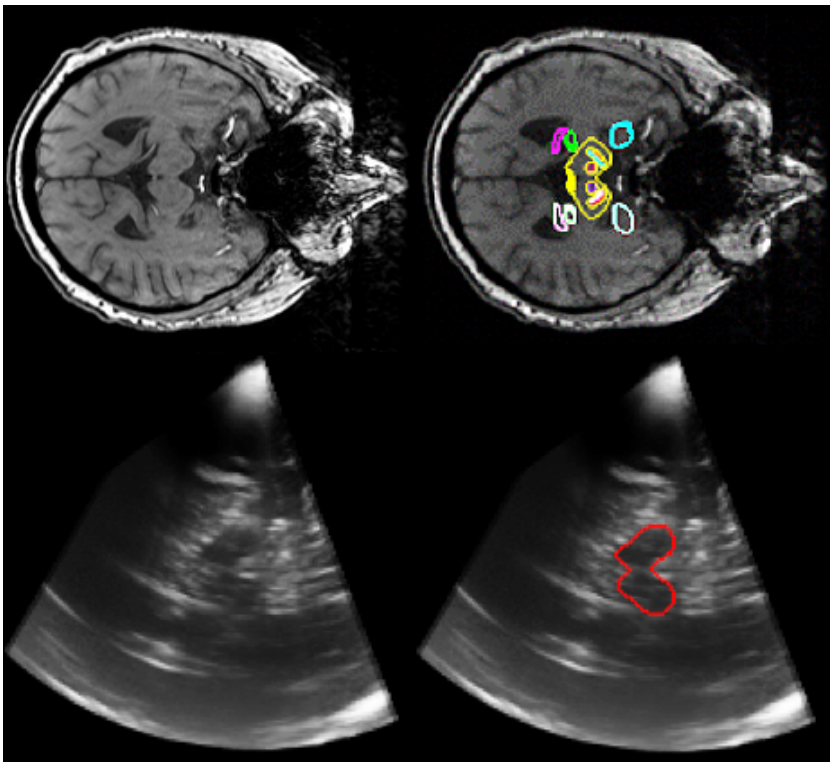

Figure 3.10: Example of MRI and ultrasound slices (left) and their respective segmentations (right) as estimated by Hough-CNN. Anatomies shown include midbrain in US (red) and in MRI (yellow). Further, in upper half of MRI slice: hippocampus (pink), thalamus (green), red nucleus (red), substantia nigra (green/red stripes within midbrain) and amygdala (cyan)

treatment and clinical research. A concrete application is pre-operative planning of Deep Brain Stimulation (DBS) neurosurgery in which basal ganglia, like the sub-thalamic nucleus (STN) and globus pallidus internal (GPi), are targeted for treatment of symptoms of Parkinson'ï¿½ï¿½s disease (PD) and dystonia, respectively [48]. Accurate localization and outlining of these nuclei can be challenging, even when performed manually, due to their weak contrast in MRI data. Moreover, fully manual labelling of individual MRIs into multiple regions in 3D is extremely time-consuming and therefore prohibitive. For this reason, both in research [48, 46] and in clinical practice [12], segmentation through atlas-based approaches is widely used.

Transcranial ultrasound (TCUS) can be used to scan deep brain regions non-invasively through the temporal bone window. Using TCUS, hyperechogenicities of the Substantia Nigra (SN) can be analysed, gaining valuable information to perform differential [167] and early [20] diagnosis of Parkinson's Disease (PD). A crucial step towards computer assisted diagnosis of PD is midbrain segmentation [1, 112]. This task is reportedly challenging even for human observers [135]. In order to penetrate the skull, low frequencies need to be applied resulting in an overall reduction of the resolution and in the presence of large incoherent speckle patterns. Scanning through the bone, moreover, attenuates a large part of the ultrasound energy, leading to overall reduction of the signal-to-noise ratio, as well as low contrast and largely missing contours at anatomic boundaries. Additionally, the higher speed of sound in the bone leads to phase aberration [86] and de-focussing of the ultrasound beam which causes further lowering of the image quality. A variety of image TCUS quality, anatomical visibility and 3D ultrasound fan geometry can be seen in Figure 3.12. Registration methods, in particular non-linear registration, are very difficult under these conditions. Therefore, atlas-building and atlas-based segmentation methods tend to fail in ultrasound.

In this work we evaluate the performance of our approach using an ultrasound dataset of manually annotated TCUS volumes depicting the midbrain, and an MRI dataset, depicting 26 regions including basal ganglia, annotated in a computer-assisted manner. Our method is fully automatic, registration-free and highly robust towards the presence of artifacts. Through our patch-based voting strategy, our approach can localize and segment structures that are only partially visible or whose appearances are corrupted by artifacts. This approach, published in [113], is the first work employing CNNs to perform ultrasound segmentation.

Our work features several contributions:

- We propose Hough-CNN, a novel segmentation approach based on a voting strategy similar to [112]. We show that the method is multi-modal, multi-region, robust and implicitly encoding priors on anatomical shape and appearance. Hough-CNN delivers results comparable or superior to other state-of-the-art approaches while being entirely registration-free. In particular, it outperforms methods based on voxel-wise classification.

- We propose and evaluate several different CNN architectures, with varying numbers of layers and convolutional kernels per layer. In this way

Name	Network Architecture	Act. function	Init.	Remarks
3-3-3-3-3	$I_{31} \cdot C_3^{64} \cdot P_3^2 \cdot C_3^{64} \cdot C_3^{64} \cdot C_3^{64} \cdot C_3^{64} \cdot F_{128} \cdot F_{128} \cdot F_{\#regions}$	PReLU	MSRA	F use drop-out (ratio 0.5)
3-3-3-3-3-3-3-3	$I_{31} \cdot C_3^{64} \cdot C_3^{64} \cdot C_3^{64} \cdot C_3^{64} \cdot C_3^{64} \cdot C_3^{64} \cdot C_3^{64} \cdot C_3^{64} \cdot F_{128} \cdot F_{128} \cdot F_{\#regions}$			
5-5-5-5-5	$I_{31} \cdot C_5^{64} \cdot C_5^{64} \cdot C_5^{64} \cdot C_5^{64} \cdot C_5^{64} \cdot F_{128} \cdot F_{128} \cdot F_{\#regions}$			
7-5-3	$I_{31} \cdot C_7^{64} \cdot P_3^2 \cdot C_5^{64} \cdot C_3^{64} \cdot F_{128} \cdot F_{\#regions}$			
9-7-5-3-3	$I_{31} \cdot C_9^{64} \cdot C_7^{64} \cdot C_5^{64} \cdot C_3^{64} \cdot C_3^{64} \cdot F_{128} \cdot F_{128} \cdot F_{\#regions}$			
Small Alex	$I_{31} \cdot C_{11}^{64} \cdot P_2^1 \cdot C_5^{64} \cdot P_2^1 \cdot C_3^{64} \cdot C_3^{64} \cdot C_3^{64} \cdot F_{128} \cdot F_{128} \cdot F_{\#regions}$			

Table 3.4: Six CNNs were designed and employed to process squared or cubic patches having size 31 pixels. Notation for architecture and CNN layers given in section 3.5.2. Activation functions follow all layers.

we acquire insights on how different network architectures cope with the amount of variability present in medical volumes and image modalities.

- Each network is trained with different amounts of data in order to evaluate the impact of the number of annotated training examples on the final segmentation result. In particular, we show how complex networks with higher parameter number cope with relatively small training datasets.
- We adapted the *Caffe* framework [87] to perform convolutions of volumetric data, preserving its third dimension across the whole network. We compare CNN performance using 3D convolution to the more common 2D convolution, as well as to a recent 2.5D approach [146].

In this work we propose and benchmark six network architectures, including one very deep network having 8 convolutional layers as shown in Table 3.4.

3.5.2 Method

We propose six different convolutional neural network architectures trained with patches extracted from annotated medical volumes. We optimize our models to correctly categorize data-points into different classes. The volumes were acquired in two different modalities, US and MRI, and depict deep structures of the human brain. Accurate segmentation of the desired regions has been achieved through a Hough voting strategy, inspired by [112], which was employed to simultaneously localize and segment the structures of interest.

Convolutional neural networks

A CNN consists of a succession of layers which perform operations on the input data. *Convolutional layers* (symbol C_s^k) convolve the images I_{size} presented to their inputs with a predefined number (k) of kernels, having a certain size s, and are usually followed by *activation units* which rescale the results of the convolution in a non linear manner. *Pooling layers* (symbol P_{size}^{stride}) reduce the dimensionality of the responses produced by the convolutional layers through downsampling, using different strategies such as average-pooling or max-pooling. Finally, *fully connected layers* (symbol $F_{\#neurons}$) extract compact, high level features from the data. The kernels belonging to convolutional layers as

well as the weights of the neural connections of the fully connected layers are optimized during training through back-propagation. The network architecture is specified by the user, by defining the number of layers, their kind, and the type of activation unit. Other relevant parameters are: the number and size of the kernels employed during convolution, the amount of neurons in the fully connected part and the downsampling ratio applied by the pooling layers. We propose six network architectures that are described in Table 3.4.

CNNs perform machine learning tasks without requiring any handcrafted feature to be engineered and supplied by the user. That is, discovering optimal features describing the data at hand is part of the learning process. During training the network parameters are first initialized and then the data is processed through the layers in a feed-forward manner. The output of the network is compared with the ground-truth through a loss function and the error is back-propagated [97] in order to update the filters and weights of all the layers, up to the inputs. This process is repeated until it converges. Once the network is trained, predictions can be made by using it in a feed-forward manner and reading out the outputs of the last layer.

In our approach we made use of parametric rectified linear units [78] (PReLU) as our activation functions.

$$PReLU(x) = \begin{cases} x \text{ if } x \geq 0 \\ \alpha x \text{ if } x < 0 \end{cases} \tag{3.1}$$

The parameter α in the PReLU activation function is learnt during training, along with other network weights. In this context we initialize the network parameters using MSRA [78] as it is an appropriate choice when employing PReLU activation units.

Many authors [93, 81] reported that the tendency of the network to overfit can be decreased by using a technique called "drop-out" during training which inhibits the outputs of a random fraction of the neurons of the fully connected layers in each iteration. In this way it is possible to limit their excessive specialization to specific tasks, which is believed to be at the origin of overfitting in CNNs.

Finally, we employ max-pooling layers to reduce the dimensionality of the data as it traverses the network. The input of the pooling layer is exhaustively subdivided into sub-patches having fixed size and overlapping by an amount controlled by the "stride" parameter. Only the maximal value in each sub-patch is forwarded to the next layer. This procedure is known to incorporate a spatial invariance to the network which contradicts the desired localization accuracy required for segmentation. For this reason we limit the usage of pooling layers to the minimum amount required to meet the existing hardware constraints.

Voxel-wise classification

A set $\mathbf{T} = \{\mathbf{p}_1, ..., \mathbf{p}_N\}$ of square (or cubic) patches having size p pixels is extracted from J annotated volumes V_j with $j \in \{1...J\}$ along with the corresponding ground truth labels (obtained from the annotation of the center voxel) $\mathbf{Y} = \{y_1, ..., y_N\} \in \mathbb{R}$. Based on this training set CNNs are optimized to

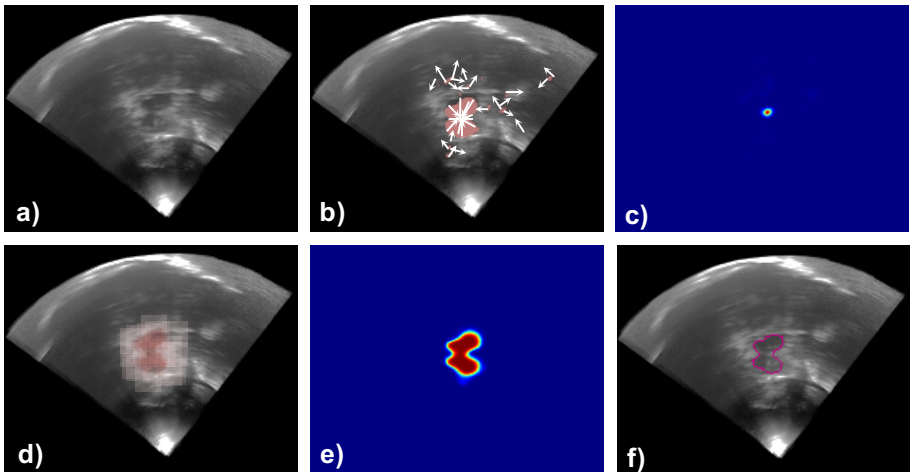

Figure 3.11: Schematic representation in 2D of the Hough-CNN segmentation approach. a) The volume is interpreted patch-wise and classified using the CNN. b) Every pixel of the foreground (red) casts one or multiple votes in order to localize the anatomy centroid. c) The votes accumulate in a vote-map, represented here in jet colormap, and the object centroid is found at the location of maximum vote accumulation. d) All the votes that accumulated close to the detected anatomy centroid contribute to the final contour by projecting a binary segmentation patch (here shown in red and white to indicate foreground and background respectively) at the location they were cast from. e) A contour confidence map is constructed by accumulating all the contributions associated to the votes. f) The resulting contour, depicted in purple, is retrieved by thresholding the confidence map.

categorize the patches correctly. The resulting trained networks are capable of performing voxel-wise classification, also called semantic segmentation, of volumes by interpreting them in a patch-wise fashion or in only one pass by employing the corresponding fully convolutional neural network (FCNN) formulation. However, due to the lack of regularization and enforcement of statistical priors this approach delivers sub-optimal results (Figure 3.16). For this reason we introduce a novel segmentation method that is based on simultaneous localization of the anatomy of interest and robust contour extraction (Figure 3.11).

Hough voting with CNN

We introduce a robust segmentation approach that is scalable to multiple regions and implicitly encodes shape priors. This method employs a Hough-voting strategy to perform anatomy localization and a database containing segmentation patches to retrieve the contour of the anatomy. Instead of relying only on categorical predictions produced by the CNNs we also make use of features extracted from their intermediate layers, in particular from the

second-last fully connected one. Several authors [93, 65, 60] have reported that these features (sometimes also called descriptors) can be used for tasks such as image retrieval by mapping images to the feature space and identifying their neighbours. These findings are employed at the core of our voting strategy.

To keep our notation as simple and understandable as possible we describe our approach for single region segmentation in the following.

During *training*, we make use of the dataset of training volumes $\mathbf{V}_j$ with $j \in \{1...J\}$, and respective binary segmentation volumes $\mathbf{S}_j$ with $j \in \{1...J\}$. We collect patches from both foreground and background and train a CNN for classification using the cross entropy loss. As a result, we obtain the parameters $\hat{\theta}$ that define the network. The CNN not only differentiates patches belonging to foreground and background through classification, but also associates each input to a feature vector obtained from its second-last fully connected layer. The macroscopic effect of the network can be summarized using two functions

$$f_1(\mathbf{p}_i, \hat{\theta}) = l_i \in \{0, 1\} \text{ and } f_2(\mathbf{p}_i, \hat{\theta}) = \mathbf{f}_i \in \mathbb{R}^d$$

respectively mapping each input patch $\mathbf{p}_i$ to its label l_i and to the feature $\mathbf{f}_i$, which has as many dimensions d as there are neurons in the fully connected layer it is collected from.

We exhaustively collect a dataset $\mathbf{T} = \{\mathbf{p}_1...\mathbf{p}_N\}$ of either 2D, 2.5D or 3D patches from the locations $\mathbf{X} = \{\mathbf{x}_1...\mathbf{x}_N\}$ of the foreground region of each of the training volumes $\mathbf{V}_j$, and we use the CNN to obtain the features $\mathbf{f}_i$ introduced before. Our goal is to create a database storing triples consisting of a feature vector $\mathbf{f}_i$, a vote $\mathbf{v}_i$ and a segmentation patch $\mathbf{s}_i$.

The vote $\mathbf{v}_i$ is a displacement vector joining the voxel $\mathbf{x}_i$, where the i-th patch was collected from, and the position anatomy centroid $\mathbf{c}_j$ in the training volume $\mathbf{V}_j$:

$$\mathbf{v}_i = \mathbf{x}_i - \mathbf{c}_j;\ \mathbf{c}_j = \frac{1}{|F_g|} \sum_{\mathbf{x}_i \in F_g} \mathbf{x}_i$$

where F_g is the set of all the voxels belonging to foreground. The binary segmentation patches assume values 1 or 0 respectively for foreground and background area since they are collected from the positions $\mathbf{x}_i$ of the binary annotation volumes $\mathbf{S}_j$.

During *testing*, in order to segment a previously unseen volume I, we make use of both the trained CNN and the database established before. We first obtain the classification label for each voxel $\mathbf{x}_i$ by processing the relative patch $\mathbf{p}_i$ through the CNN, which delivers also the features $\mathbf{f}_i$ for all the patches being classified as foreground. Each of such features is compared to those contained in the database in order to retrieve the K closest entries using Euclidean distance as criterion. This K-nearest neighbour search (K-nn) [122] is performed computing Euclidean distances $d^i_{1...K}$ between features, as previously done in [93] for image retrieval. In this approach we didn't find any need a Hough forest similar to the one one presented in Section 3.3, which would have increased the runtime of the approach and at the same time create opportunity for overfitting.

Once the neighbours are identified, their votes $\mathbf{v}^i_{1...K}$ and associated segmentation patches $\mathbf{s}^i_{1...K}$ from the database, are employed to respectively perform

localization and segmentation. The votes are weighted by the reciprocal of the Euclidean distance computed during K-nn search $w_{1...K} = \frac{1}{d^i_{1...K}}$ and contribute to a vote-map at positions

$$\hat{\mathbf{v}}^i_k = \mathbf{x}_i + \mathbf{v}^i_k;\ \forall k \in \{1...K\}$$

We repeat the steps described above for each of the patches that were classified as foreground (Figure 3.11b). Since the region of interest occurs only once in each volume, we smooth the final vote map with a small Gaussian filter and retrieve the region centroid by finding the location $\mathbf{c}$ where the maximal value of the vote map is reached (Figure 3.11c). Smoothing reduces the possibility of small localization mistakes due to "noise" in the vote map around the position where its maximum occurs.

The region of interest can now be segmented by re-projecting the votes $\mathbf{v}^i_k$ to the locations $\mathbf{x}_i$ where they have been originated from. However, not all the votes should be re-projected, since a relevant portion of them is erroneous, i.e. did not contribute to the vote-map anywhere close to the estimated anatomy location. Thus, only those that contributed to the vote-map within a certain range r from the predicted centroid are taken into consideration and are actually allowed to contribute to the final segmentation contour with their own segmentation patch $\mathbf{s}^i_k$. The segmentation patches $\mathbf{s}^i_k$ are centred at the location $\mathbf{x}_i$, weighted by w^i_k and accumulated in the segmentation map $\mathbf{S}$ (Figure 3.11d). Assuming that the segmentation patches $\mathbf{s}^i_k$ have been extended to an infinite spatial extent by zero-padding, we can write:

$$\hat{\mathbf{S}}(\mathbf{x}) = \sum_{\mathbf{x}_i} \sum_{k=1}^{K} Ind(\hat{\mathbf{v}}^i_k, \hat{\mathbf{c}})\ w^i_k\ s^i_k(\mathbf{x} - \mathbf{x}_i)$$

$$Ind(\mathbf{a}, \mathbf{b}) = \begin{cases} 1 & \|\mathbf{a} - \mathbf{b}\| < r \\ 0 & \|\mathbf{a} - \mathbf{b}\| \geq r \end{cases}$$

In this sense, the segmentation patches $\mathbf{s}^i_k$ can be seen as basis functions $s^i_k(\mathbf{x})$, which take binary values, that need to be scaled and re-centered at appropriate locations in order to produce the desired effect in the segmentation map. The segmentation map $\mathbf{S}$ is normalized to take only values comprised between 0 and 1 by dividing each of its voxels by the number of contributions that had been accumulated there. At this point it is thresholded in order to obtain the final binary contour.

The approach is summarized schematically in Figure 3.11. Extending this method to multiple regions requires little effort. In our implementation, we treated each region independently by creating region-specific databases as well as dedicated vote-maps and segmentations. The memory requirements of this approach can be decreased by retrieving the segmentation patches directly from the volumes $\mathbf{S}_{1...J}$ instead of storing them in the database. In this case, the database contains coordinates that are used to fetch contour portions from the $\mathbf{S}_{1...J}$.

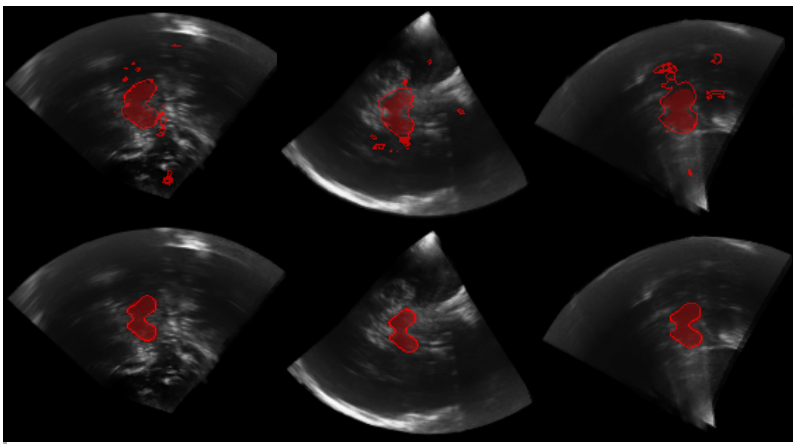

Figure 3.12: Visual comparison of semantic segmentation results (top) and Hough-CNN results (bottom) on the same ultrasound data using the best-performing CNN. Red areas represent ground truth annotation. Red contours represent segmentation outputs. Best viewed in digital format.

Efficient patch-wise evaluation through CNN

When dealing with images or volumes, patches are extracted in a sliding-window fashion and processed through a CNN. This approach is inefficient due to the high amount of redundant computations that need to be performed for neighbouring patches. In case no padding is used within the convolutional layers, the whole volume can be convolved with the respective kernels in one pass, instead of treating each patch separately, while achieving the same result. The same holds true for pooling layers whose pooling windows can be arranged to process the whole volume at once. However, as soon as fully connected layers are employed, the volume can no longer be processed in one pass due to the fact that the connections of this layer are limited to the size of the input patch.

To solve this issue we convert out CNN to a FCNN as proposed by Sermanet et al. in [149] in order to be able to process the whole volume at once, yet retrieving the same results that we would obtain if the data would be processed patch-wise.

3.5.3 Experimental evaluation

In this section we show that CNNs not only can be used to robustly segment medical volumes (Figure 3.12, Figure 3.13), but they also posses the ability of learning extremely effective features (outputs of upper layers) from the data. Even in ultrasound, where the structures of interest are often not clearly visible or the images are affected by artifacts, CNNs are able to focus on salient information and therefore recognize patterns. We demonstrate the superior performances of our Hough-voting-based segmentation algorithm by evaluating our method on two datasets of US and MRI volumes depicting the

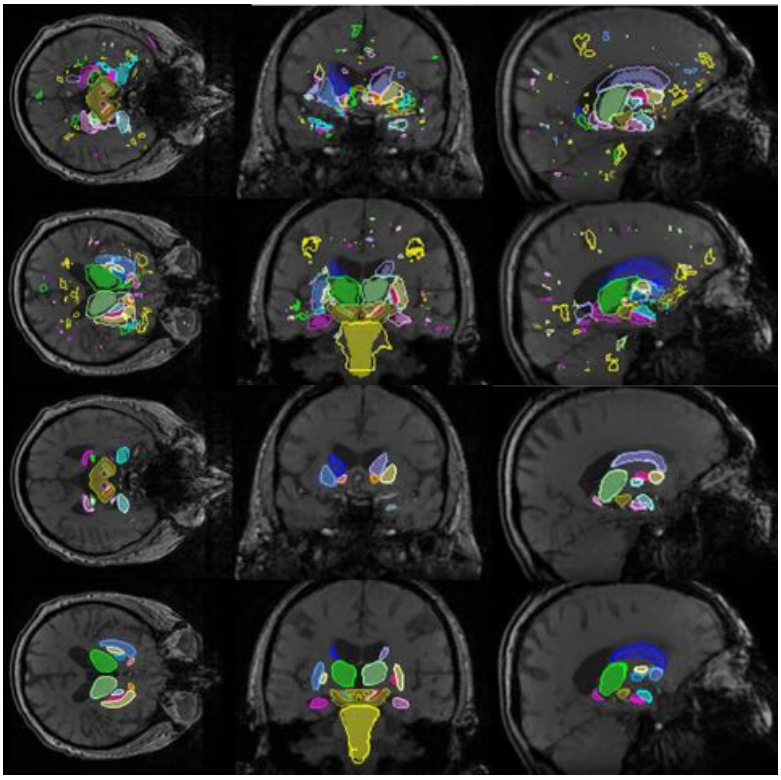

Figure 3.13: Visual comparison of semantic segmentation results (top two rows) and Hough-CNN results (bottom two rows) on same MRI volumes using the same trained CNN. Coloured areas represent ground truth annotation. Coloured contours represent segmentation outputs. Best viewed in digital format.

human brain. The two modalities provide complementary information, but are inherently different both from the point of view of the challenges they offer and the range of anatomy they can image.

Datasets and ground-truth definition

Our MRI dataset is composed of MRI volumes of 55 subjects, which were acquired using 3D gradient-echo imaging (magnitude and phase) with an isotropic spatial resolution of 1x1x1 mm. The sequence [49] is designed for quantitative susceptibility mapping (QSM) and sensitivity towards iron deposits. These are biomarkers for movement disorders like Parkinson's Disease and create visible contrast in relevant basal ganglia like SN and STN. For our study, basal ganglia and other deep-brain structures were annotated in an atlas volume in two ways. One set of bi-lateral atlas labels (brainstem, n. accumbens, amygdala, caudate, thalamus, hippocampus, pallidum, putamen) were annotated semi-automatically via a shape- and appearance-model segmentation (FSL FIRST [131]) plus manual correction of generated labels (one

neuroimage technician, verified by one expert neurologist). Another set of bi-lateral labels (separation of of pallidus into GPi and GPe, midbrain, red nucleus, substantia nigra pars compacta and substantia nigra pars reticulata) was annotated in a fully manual manner (neuroimage technician, verified by expert neurologist) based on visible contrast. The atlas labels were transferred using a state-of-the-art atlas approach [7]. As a summary, the list of structures of interest is also visible in Figure 3.15.

The US dataset was acquired transcranially on 34 subjects, with several freehand 3D sweeps recorded through the left and right temporal bone window each. Altogether, 162 volumes were acquired with slight variations in bone window positioning, and reconstructed at 1mm isotropic resolution. For all 162 TCUS volumes, midbrain outlines were annotated in 3D by a single human expert. Inter-rater agreement of the midbrain annotations, in terms of Dice coefficient, has been reported in [135] to be 0.85. CNN training was performed on data from 8 subjects (40 sweeps), and testing on data from 24 previously unseen subjects (114 sweeps), while validation data was performed on 8 sweeps from 2 subjects. Performing segmentation on more than 100 test volumes is a good indicator of actual clinical applicability of (Hough-)CNN-based segmentation. The experiments show that the method generalizes very well on previously unseen data, which is a highly desirable property in clinical settings.

In order to test our approach and to benchmark the capabilities of the proposed CNNs when they are trained with a variable amount of data, we establish, for each dimensionality (2D, 2.5D and 3D) two differently sized training sets in US and three in MRI respectively. For each of the 40 training volumes in US we collect either 2K or 10K patches per volume such that half of the training set depicts the background and the other half the foreground. The resulting training sets have respective sizes of 80K and 400K patches. A validation set containing 5K patches has been established for US using images of subjects that have not been used for training or testing and employed to assess the generalization capabilities of the models. From the 45 MRI training volumes, we extract either circa 100, 1K or 10K patches per volume *per region* (including background). The resulting training sets have respective sizes of 135K, 1.35M and 13.5M patches.

CNN parameters

We analyze six different network architectures, presented in Table 3.4, by training each of them for 15 epochs using Stochastic Gradient Descent (SGD) with mini-batches of 64 or 124 samples, learning rate varying between 10^{-2} and $5 \cdot 10^{-3}$ depending on the individual network architecture, momentum 0.9 and weight decay $5 \cdot 10^{-4}$. All our models converged after a few epochs, and often before the seventh epoch.

Each network is analyzed three times, with patches capturing the same amount of context from the neighbourhood, but having different dimensionality. That is, our networks process 2D data, 2.5D data and 3D data in order to investigate how the networks respond to the higher amount of information

Parameter Name	Value
Tolerance radius r for reprojection	$r = 3$ voxels
Amount of smoothing for vote-maps	$\sigma = 1$
Maximum number of neighbours K-NN	$K = 20$
Maximal distance of K-NN neighbours (US)	2.5
Maximal distance of K-NN neighbours (MRI)	6.0
Size of segmentation patch	$9 \times 9 \times 9$

Table 3.5: Parameters of the model utilized during the experiments.

carried by patches in 2.5D and 3D patches compared to 2D. During training, we randomly sample patches from annotated volumes and we feed them to the networks along with their ground truth labels. The patches of the $2D$ dataset are all square and have a size of 31×31 pixels; the 2.5D dataset is composed of patches having the same size and three channels consisting of 2D patches from the sagittal, coronal and transversal plane centred at the same location; the 3D dataset contains cubic patches having size $31 \times 31 \times 31$ voxels.

Some of the parameters supplied to our Hough-CNN algorithm are empirically chosen. Parameters names and respective values are reported in Table 3.5. These parameters remained constant throughout all experiments, both in ultrasound and MRI. All the trainings were performed on Intel i7 quad-core workstations with 32 gigabytes of ram and graphic cards from Nvidia, specifically "Tesla k40" or "Titan X" (12GB VRAM). All tests were made on a similar workstation equipped with a Nvidia GTX 980 (4 GB VRAM).

Experiments and results in ultrasound

We train our CNNs with different amount of data having different dimensionality, as explained in Section 3.5.3. Each of the six proposed architectures is trained six times (five for 3D due to the computational burden of some experiments) in order to cover all the possible combinations of dimensionalities (2D, 2.5D, 3D patches) and amount of data (training set sizes 80K, 400K). We test each CNN on 114 ultrasound volumes acquired from subjects whose scans have never been used during training or validation.

Table 3.6 shows the average performance in terms of Dice coefficients, mean distances of the estimated contours to the ground truth annotations and failure rates of the proposed Hough-CNN segmentation approach when different CNNs are employed. Since we segment one region per volume, the failure rate represents the percentage of volumes where the region of interest could not be segmented due to wrong localization (Dice 0). In Figure 3.14 we provide summary of the performances of each network, when various amounts of training data are used and patches of different dimensionality are supplied. Better networks produce Dice histograms whose higher values are occurring far away from the origin.

Visual examples of ultrasound segmentation results are visible in Figure 3.12. It is notable that the Hough-CNN segmentation is able to localize and segment the midbrain accurately, regardless of whether the scan was acquired

Dimensionality →	2D			2.5D			3D		
Averages →	Dice $[0,1]$	Distance (mm)	Failures	Dice $[0,1]$	Distance (mm)	Failures	Dice $[0,1]$	Distance (mm)	Failures
Training set size 80K patches									
3-3-3-3-3	**0.83**	0.92	3%	0.82	0.91	5%	0.79	0.95	6%
3-3-3-3-3-3-3-3	0.80	0.93	5%	0.80	0.94	4%	**0.82**	0.99	5%
5-5-5-5-5	0.77	1.07	9%	0.74	1.11	14%	**0.80**	1.02	6%
7-5-3	0.80	0.96	5%	**0.81**	1.00	5%	0.80	1.02	7%
9-7-5-3-3	0.79	0.96	7%	0.81	0.93	5%	**0.82**	0.99	7%
SmallAlex	**0.85**	0.81	1%	0.81	0.98	5%	0.80	0.98	3%
Training set size: 400K patches									
3-3-3-3-3	**0.84**	0.90	1%	0.83	0.95	3%	–	–	–
3-3-3-3-3-3-3-3	**0.85**	0.90	0%	0.83	0.99	3%			
5-5-5-5-5	**0.83**	0.94	2%	0.81	1.03	5%			
7-5-3	**0.83**	0.94	2%	0.81	0.99	5%			
9-7-5-3-3	**0.82**	1.01	2%	**0.82**	0.96	5%			
SmallAlex	**0.83**	0.91	3%	0.81	0.94	4%			

Table 3.6: Midbrain segmentation results in 114 previously unseen TCUS volumes, using Hough-CNN with variations of architectures (single rows), patch dimensionalities (column blocks) and training set sizes (row blocks). The best result for each architecture (across the data dimensionalities) are highlighted by using bold typeface. The best results for each dimensionality (across the architectures) are underlined.

through the left or right bone window. It is also robust to bone window quality and overall visibility of structures, as well as signal-drop regions and blurring.

Experiments and results in MRI

We train each of our networks nine times (eight for 3D) in order to explore all the possible combination of different data dimensionality and size of the training set as explained in Section 3.5.3. We test each of the models on 10 volumes, using their respective atlas-based annotations for evaluation. We verified, through visual inspection performed by a technician and an expert neurologist, that the annotation appropriately delineate the regions of interest.

Table 3.7 reports the average performance in terms of Dice coefficients, mean distances of the estimated contours to the ground truth annotations and failure rates of the proposed Hough-CNN segmentation approach when different CNNs are employed at its core. The failure rate, in particular, refers to the percentage of regions of the whole training set (total number: 26×10 regions), that were not segmented correctly by Hough-CNN due to the fact that they could not be correctly localized. The results are clustered by the size of the training set employed to train the model to improve readability and the possibility of making comparisons between CNNs employing data having different dimensionality (2D, 2.5D and 3D). From these results we observe that the best performing architecture is "7-5-3".

In Figure 3.15 we compare the results achieved by the architecture "7-5-3", on each of the 26 brain region of interest separately, when different data dimensionalities are used. The bar plot shows the results in terms of Dice coefficient, while the dashed line plot conveys the results in terms of average distance of the estimated contour to ground-truth delineation. We observe that Hough-CNN yields better Dice coefficients when bigger regions and high contrast area are segmented. Small and low contrast regions could be correctly localized but they were in general harder to segment.

Visual examples of MRI segmentation results are visible in Figure. 3.13. It is notable that the Hough-CNN segmentation is able to correctly localize and segment multiple structures, despite large anatomical variability, such as cortical atrophy and enlarged lateral ventricles.

3.5.4 Discussion

Training of CNNs requires a large amount of data in order to achieve satisfactory voxel-wise classification results and perform semantic segmentation. However, as described in the introduction, obtaining such large annotated datasets is rarely possible in clinical settings. By using a voting-based strategy, it is possible to localize the anatomy of interest with high precision, even when the rate of mis-classified voxels is very high. Additionally, our Hough-CNN approach implicitly enforces shape priors which facilitate segmentations in images where the anatomy of interest is poorly visible. Furthermore, when using 3D patches, only 1.35M training patches were required to surpass the performance obtained with datasets of 13.5 millions 2D and 2.5D patches. This marks a 90% reduction of required training data, which turns out to be useful

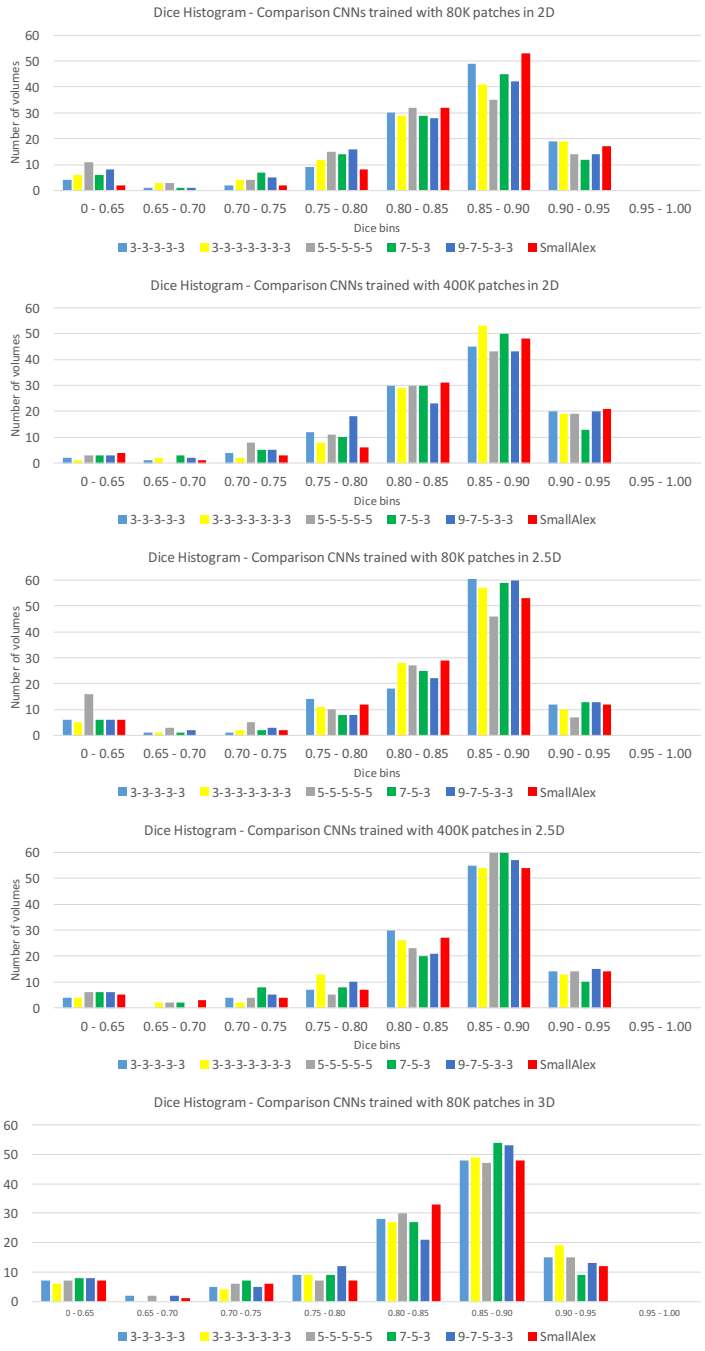

Figure 3.14: The midbrain segmentation performance of each network on 114 TCUS test volumes, under different training conditions, is summarised through histograms. The horizontal axis is subdivided in Dice bins having a width of 0.05 Dice. The vertical axis represents the number of volumes falling in each Dice bin. Each CNN architecture is depicted with its own colour.

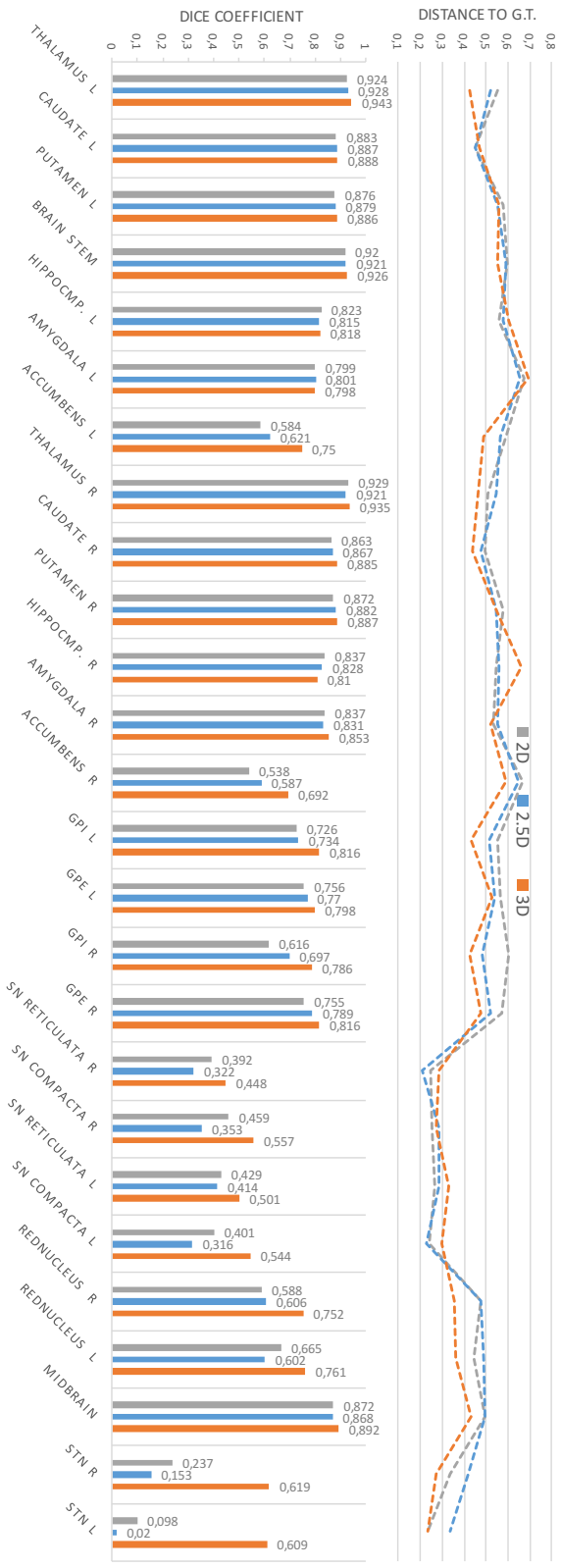

Figure 3.15: Average Dice coefficients (bar-plot) and distances to ground-truth delineation (dashed-lines plot), obtained segmenting the MRI test volumes using the best-performing network architecture "7-5-3". Dice coefficients are shown for each of the 26 target regions. Results obtained considering 2D, 2.5D and 3D data are represented in grey, blue and orange respectively. Best segmentation were delivered when 3D data was fed into the network, although the model was trained with only 1.35 millions 3D patches instead of the 13.5 million patches that were employed to train the models dealing with 2D and 2.5D data.

Dimensionality →	2D			2.5D			3D		
Averages →	Dice $[0,1]$	Distance (mm)	Failures	Dice $[0,1]$	Distance (mm)	Failures	Dice $[0,1]$	Distance (mm)	Failures
Training set size: 135K patches									
3-3-3-3-3	0.61	0.52	6%	0.62	0.51	3%	**0.70**	0.46	0%
3-3-3-3-3-3-3-3	0.61	0.52	8%	0.61	0.51	5%	**0.70**	0.45	0%
5-5-5-5-5	0.64	0.49	6%	0.63	0.52	1%	**0.71**	0.44	1%
7-5-3	0.67	0.48	4%	0.68	0.48	2%	**0.76**	0.45	0%
9-7-5-3-3	0.60	0.52	8%	0.61	0.52	3%	**0.68**	0.49	0%
SmallAlex	0.61	0.53	5%	0.62	0.52	5%	**0.71**	0.46	0%
Training set size: 1.35M patches									
3-3-3-3-3	0.63	0.51	3%	0.62	0.52	5%	**0.72**	0.45	0%
3-3-3-3-3-3-3-3	0.63	0.51	3%	0.62	0.52	2%	**0.70**	0.52	0%
5-5-5-5-5	0.64	0.51	3%	0.61	0.52	3%	**0.71**	0.44	0%
7-5-3	0.68	0.47	2%	0.68	0.47	2%	**0.77**	0.45	0%
9-7-5-3-3	0.63	0.53	4%	0.62	0.52	2%	**0.68**	0.47	1%
SmallAlex	0.64	0.51	4%	0.62	0.53	6%	**0.72**	0.46	0%
Training set size: 13.5M patches									
3-3-3-3-3	**0.64**	0.52	3%	**0.64**	0.52	3%	–	–	–
3-3-3-3-3-3-3-3	**0.65**	0.56	2%	**0.65**	0.54	0%			
5-5-5-5-5	**0.64**	0.51	2%	**0.64**	0.51	2%			
7-5-3	**0.68**	0.49	3%	0.67	0.48	3%			
9-7-5-3-3	**0.63**	0.52	5%	**0.63**	0.52	5%			
SmallAlex	**0.65**	0.52	4%	0.63	0.53	5%			

Table 3.7: Average segmentation results of 26 structures in 10 MRI test volumes, using Hough-CNN with variations of architectures (single rows), patch dimensionalities (column blocks) and training set sizes (row blocks). The best result for each architecture (across the data dimensionalities) are highlighted by using bold typeface. The best results for each dimensionality (across the architectures) are underlined. The best result is obtained using the architecture "7-5-3" and 3D data.

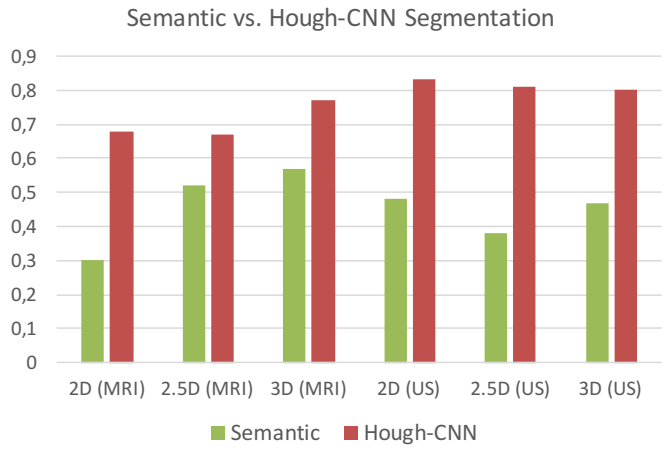

Figure 3.16: Comparison of mean Dice coefficients obtained in 2*D*, 2.5*D* and 3*D* on US and MRI data using Hough-CNN and semantic segmentation.

when big quantities of annotated data are not available. In all three dimensionalities, 2D, 2.5D and 3D, Hough-CNN outperforms voxel-wise segmentation (cf. Figure 3.16). Similar to related works [124, 139, 161, 100], we thus demonstrate that it may be beneficial to embed CNNs as powerful classifiers into higher-level methods which encode anatomic shape- and appearance priors.

The experiments performed on MRI highlight important aspects of both our CNNs and the modality itself. Most of the brain regions considered in this study (e.g. midbrain, STN, caudate) can be recognized by a human rater by clearly visible contrasts, while the position and boundaries of difficult regions with less contrast (e.g. GPi, GPe, SNpc, SNpr) can be inferred through anatomical knowledge and neighborhood context. Ultrasound volumes are much more challenging from this point of view. Human midbrain in TCUS can be difficult to discern and human observers can be mislead by artifacts and signal-loss areas having similar shape. The CNNs employed in this study had various architectures and therefore different pattern recognition capabilities. In MRI, where the most part of regions of interest have good contrast while the position of the others can be inferred by the context, the best performing network was "7-5-3". Although this architecture is the simplest, it delivered best results in all the MRI experiments. In US, which is a challenging modality, the networks that delivered best results were among the most complex. "SmallAlex" and "3-3-3-3-3-3-3-3" are deeper and therefore recognize more complex visual content than "7-5-3".

While we observed a strong performance advantage when segmenting MRI volumes considering 3D data (Table 3.7), we observed the opposite effect when segmenting ultrasound as shown in the bottom left of Table 3.6. In MRI, processing data in 3D brings additional useful information which improves the performance of both automated methods and human raters, who refer simultaneously to sagittal, coronal and axial views when establishing the ground truth. In US, we observed that experts segmenting the ground truth

used only the axial plane, since it is the only plane in which the characteristic shape of the midbrain can be recognized. Similarly, CNNs produce best results when they are not supplied with misleading information from sagittal and coronal planes.

Altogether, using Hough-CNNs, we segmented 10 previously unseen MRI volumes achieving very high Dice coefficients for large and high-contrasted regions, while some of the smallest and most challenging regions were almost always localized accurately and segmented with sub-voxel mean surface distance. Additionally, we achieved very robust midbrain segmentation in 3D-TCUS, in a test dataset of more than 20 subjects and 114 volumes, with a large variation of 3D sweep geometry, bone window qualities, midbrain appearance, location and orientation. Given the size and variety of the 3D-TCUS test set, we are confident to say that the method generalizes well to unseen patients.

Compared to atlas-segmentation, Hough-CNN is faster (30 seconds in US, and 3-4 minutes in MRI on the machine employed for testing) and entirely registration-free. This makes our approach applicable to TCUS data, in which registration-dependent methods like atlas-based segmentation would be extremely difficult, if not impossible, due to largely missing anatomical and structural context. Our approach is flexible since both votes and segmentation patches can be substituted without any need for re-training or augmented to include information from multiple experts. As a future work, we plan to investigate the extendability of the trained CNN classifier to other modalities via transfer learning, e.g. from our QSM sequences to T1 or T2. It is also noteworthy that in this work, we have only used the CNN method for segmentation. However, as other works have demonstrated [134], the learned data representations in the last layers of the CNN can be directly used for classification or regression of disease parameters. This can be interleaved with segmentation, which goes far beyond the capabilities of purely atlas-based methods.

3.6 Segmentation via fully convolutional neural networks

In this section we present a novel 3D segmentation approach that leverages the power of a fully convolutional neural network, trained end-to-end, for processing of volumetric medical images.

Compared to other recent approaches, the contributions of this approach is three-fold. First, instead of processing the input volumes in a 2D slice-by-slice fashion, we propose to directly use 3D convolutions. Second, we propose to maximize a novel objective function designed specifically for medical image segmentation, which is based on the Dice overlap measure. Third, we define our network architecture such that residual functions are learned by our convolutional layers. This improve convergence and performances. As our empirical observations confirm, this mechanism ensures also our novel architecture to converge in a fraction of the time required by a similar network that does not learn residuals.

Our CNN is trained end-to-end on MRI volumes depicting prostate, and learns to predict segmentation for the whole volume at once. Prostate segmentation from MRI can be challenging due to large appearance variation across different scans, e.g. in terms of deformations or changes of the intensity distribution. Moreover, MRI volumes are often affected by artifacts and distortions due to field inhomogeneity. To cope with the limited number of annotated volumes available for training, we augment the data applying random non-linear transformations and histogram matching. We show in our experimental evaluation that our approach achieves good performances on challenging test data while requiring only a fraction of the processing time needed by other previous methods[2].

3.6.1 Method

In Figure 3.17 we provide a schematic representation of our convolutional neural network. We perform convolutions aiming both at extracting features from the data and, at the end of each stage, reducing its resolution by using appropriate stride. The left part of the network consists of a compression path, while the right part decompresses the signal until its original size is reached. Convolutions are all applied with appropriate padding.

The left side of the network is divided into different stages that operate at different resolutions. Each stage comprises one to three convolutional layers. Similarly to the approach presented in [77], we formulate each stage such that it learns a residual function: the input of each stage is (a) used in the convolutional layers and processed through the non-linearities and (b) added to the output of the last convolutional layer of that stage in order to enable learning a residual function. As confirmed by our empirical observations, this architecture ensures convergence in a fraction of the time required by a similar network that does not learn residual functions.

[2]Detailed results available at `http://promise12.grand-challenge.org/results/`

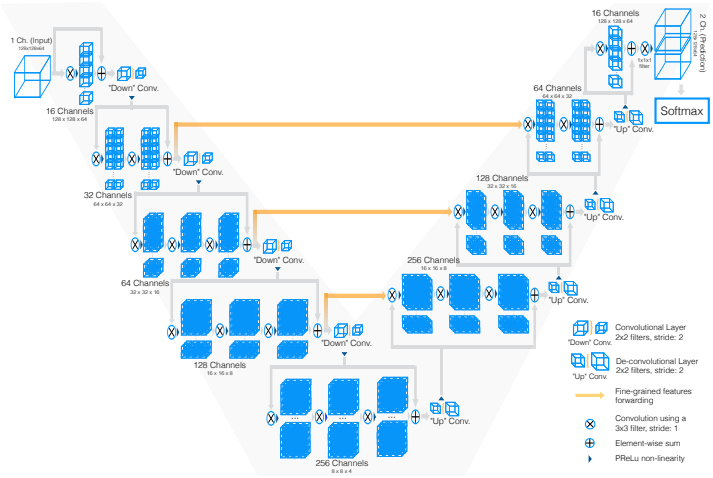

Figure 3.17: Schematic representation of our network architecture. Our custom implementation of Caffe [87] processes 3D data by performing volumetric convolutions. Best viewed in electronic format.

The convolutions performed in each stage use volumetric kernels having size $5 \times 5 \times 5$ voxels. As the data proceeds through different stages along the compression path, its resolution is reduced. This is performed through convolution with $2 \times 2 \times 2$ voxels wide kernels applied with stride 2 (Figure 3.18). Since the second operation extracts features by considering only non-overlapping $2 \times 2 \times 2$ volume patches, the size of the resulting feature maps is halved. This strategy serves a similar purpose as pooling layers which, motivated by [153] and other works discouraging the use of max-pooling operations in CNNs, have been replaced in our approach by convolutional ones. The number of feature channels doubles at each stage of the compression path of the V-Net. Due to the presence of the residual connections we need to increase the number of channels of the skip connection to match the output of the convolutional layers. We resort to using the down-sampling convolutions to do this. PReLu non linearities [78] are applied throughout the network.

Replacing pooling operations with convolutional ones also results in networks that, depending on the specific implementation, can have a smaller memory footprint during training. This is due to the fact that switches, which map the output of pooling layers back to their inputs, do not need to be stored for back-propagation. In particular, this can be analyzed and better understood [178] when applying only de-convolutions instead of un-pooling operations.

Downsampling allows us to reduce the size of the signal presented as input and to increase the receptive field of the features being computed in subsequent network layers. Each of the stages of the left part of the network, computes a number of features which is two times higher than the one of the previous layer.

The right portion of the network extracts features and expands the spatial support of the lower resolution feature maps in order to gather and assemble the necessary information to output a two channel volumetric segmentation.

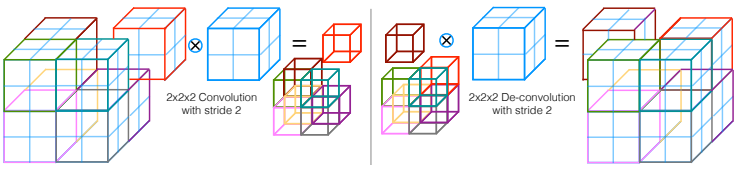

Figure 3.18: Convolutions with appropriate stride can be used to reduce the size of the data. Conversely, de-convolutions increase the data size by projecting each input voxel to a bigger region through the kernel.

The two feature maps computed by the very last convolutional layer, having $1 \times 1 \times 1$ kernel size and producing outputs of the same size as the input volume, are converted to probabilistic segmentations of the foreground and background regions by applying soft-max voxel-wise. After each stage of the right portion of the CNN, a de-convolution operation is employed in order increase the size of the inputs (Figure 3.18) followed by one to three convolutional layers involving half the number of $5 \times 5 \times 5$ kernels employed in the previous layer. Similar to the left part of the network, we resort to learn residual functions in the convolutional stages of the right part as well.

Similarly to [145], we forward the features extracted from early stages of the left part of the CNN to the right part. This is schematically represented in Figure 3.17 by horizontal connections. In this way we gather fine grained detail that would be otherwise lost in the compression path and we improve the quality of the final contour prediction. We also observed that these connections improve the convergence time of the model.

We report in Table 3.8 the receptive fields of each network layer, showing the fact that the innermost portion of our CNN already captures the content of the whole input volume. We believe that this characteristic is important during segmentation of poorly visible anatomy: the features computed in the deepest layer perceive the whole anatomy of interest at once, since they are computed from data having a spatial support much larger than the typical size of the anatomy we seek to delineate, and therefore impose global constraints.

Dice loss layer

The network predictions, which consist of two volumes having the same resolution as the original input data, are processed through a soft-max layer which outputs the probability of each voxel to belong to foreground and to background. In medical volumes such as the ones we are processing in this work, it is not uncommon that the anatomy of interest occupies only a very small region of the scan. This often causes the learning process to get trapped in local minima of the loss function yielding a network whose predictions are strongly biased towards background. As a result the foreground region is often missing or only partially detected. Several previous approaches resorted to loss functions based on sample re-weighting where foreground regions are given more importance than background ones during learning [145]. In this work, we propose a novel objective function based on Dice coefficient, a quantity

ranging between 0 and 1, which we aim to maximize.

The Dice coefficient D between two binary volumes can be written as

$$D = \frac{2\sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2} = \frac{2\sum_i^N p_i g_i}{\sum_i^N p_i + \sum_i^N g_i} \; s.t. \; p_i \in \{0,1\} \wedge g_i \in \{0,1\}.$$

As expressed in the formula above, the Dice coefficient can be written in two different yet equivalent ways as long as all the values p_i and g_i which represent voxels of, respectively, the predicted segmentation volume and the ground truth labeling are strictly binary. This equivalence does not hold for their gradient when the two expressions are differentiated with respect to any voxel p_j of the prediction volume P.

The derivative of the first expression is:

$$\frac{\partial D_1}{\partial p_j} = \frac{\partial}{\partial p_j}\left(\frac{2\sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2}\right) = 2\left[\frac{g_j\left(\sum_i^N p_i^2 + \sum_i^N g_i^2\right) - 2p_j\sum_i^N p_i g_i}{\left(\sum_i^N p_i^2 + \sum_i^N g_i^2\right)^2}\right];$$

the derivative of the second expression is:

$$\frac{\partial D_2}{\partial p_j} = \frac{\partial}{\partial p_j}\left(\frac{2\sum_i^N p_i g_i}{\sum_i^N p_i + \sum_i^N g_i}\right) = 2\left[\frac{g_j\left(\sum_i^N p_i + \sum_i^N g_i\right) - \sum_i^N p_i g_i}{\left(\sum_i^N p_i + \sum_i^N g_i\right)^2}\right].$$

Let's now study the behavior of these two gradients at optimum. In the optimal case we have that every voxel of the predicted volume P assumes a binary value $p_i \in \{0,1\}$ and that this value is equal to g_i which represents the corresponding voxel of the binary ground-truth volume.

$$\left.\frac{\partial D_1}{\partial p_j}\right|_{P=G} = 2\left[\frac{g_j\left(\sum_i^N p_i^2 + \sum_i^N g_i^2\right) - 2p_j\sum_i^N p_i g_i}{\left(\sum_i^N p_i^2 + \sum_i^N g_i^2\right)^2}\right]$$

because of the equivalence between the voxels of P and those of G,

$$\left.\frac{\partial D_1}{\partial p_j}\right|_{P=G} = 2\left[\frac{p_j\left(\sum_i^N p_i^2 + \sum_i^N p_i^2\right) - 2p_j\sum_i^N p_i^2}{\left(\sum_i^N p_i^2 + \sum_i^N p_i^2\right)^2}\right]$$

now we define $K = \sum_i^N p_i$ and we notice that, since p_i is binary, $p_i = p_i^2$. As a result

$$\left.\frac{\partial D_1}{\partial p_j}\right|_{P=G} = 2\left[\frac{2p_j K - 2p_j K}{(2K)^2}\right] = 0.$$

We have proven that the derivative of the dice coefficient as shown in the first part of the definition above, is zero at the optimal point when the segmentation matches the ground truth. Let us now follow the same procedure for the gradient of the second expression.

$$\left.\frac{\partial D_2}{\partial p_j}\right|_{P=G} = 2\left[\frac{g_j\left(\sum_i^N p_i + \sum_i^N g_i\right) - \sum_i^N p_i g_i}{\left(\sum_i^N p_i + \sum_i^N g_i\right)^2}\right]$$

Layer	Input Size	Receptive Field
L-Stage 1	128	$5 \times 5 \times 5$
L-Stage 2	64	$22 \times 22 \times 22$
L-Stage 3	32	$72 \times 72 \times 72$
L-Stage 4	16	$172 \times 172 \times 172$
L-Stage 5	8	$372 \times 372 \times 372$
R-Stage 4	16	$476 \times 476 \times 476$
R-Stage 3	32	$528 \times 528 \times 528$
R-Stage 2	64	$546 \times 546 \times 546$
R-Stage 1	128	$551 \times 551 \times 551$
Output	128	$551 \times 551 \times 551$

Table 3.8: Theoretical receptive field of the $3 \times 3 \times 3$ convolutional layers of the network.

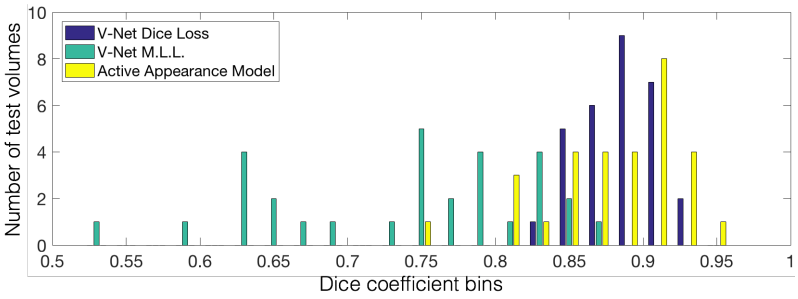

Figure 3.19: Distribution of volumes with respect to the Dice coefficient achieved during segmentation.

$$\left.\frac{\partial D_2}{\partial p_j}\right|_{P=G} = 2\left[\frac{p_j\left(\sum_i^N p_i + \sum_i^N p_i\right) - \sum_i^N p_i^2}{\left(\sum_i^N p_i + \sum_i^N p_i\right)^2}\right]$$

using the same definition we used above we can then write:

$$\left.\frac{\partial D_2}{\partial p_j}\right|_{P=G} = 2\left[\frac{2p_jK - K}{(2K)^2}\right] = 2\left[\frac{K(2p_j-1)}{4K^2}\right] = \frac{(2p_j-1)}{2K} \neq 0$$

which can create problems during optimization and neural network weights update.

We use the first formulation of dice in this work, exhibiting the squares of p_i and g_i at the denominator. When we optimize our network through this function we do not need to account for class imbalance between regions, e.g. by assigning loss weights to samples of different classes such as in [145]. In fact, we obtain results that we experimentally observed are much better than the ones computed through the same network trained optimizing a multinomial logistic loss with sample re-weighting (Fig. 3.21).

Algorithm	Avg. Dice	Avg. Hausdorff dist	Score Promise	Speed
V-Net + Dice-based loss	0.869 ± 0.033	5.71 ± 1.20 mm	82.39	1 sec.
V-Net + mult. logistic loss	0.739 ± 0.088	10.55 ± 5.38 mm	63.30	1 sec.
Imorphics [164]	0.879 ± 0.044	5.935 ± 2.14 mm	84.36	8 min.
ScrAutoProstate	0.874 ± 0.036	5.58 ± 1.49 mm	83.49	1 sec.
SBIA	0.835 ± 0.055	7.73 ± 2.68 mm	78.33	–
Grislies	0.834 ± 0.082	7.90 ± 3.82 mm	77.55	7 min.

Table 3.9: Quantitative comparison between the proposed approach and the current best results on the PROMISE 2012 challenge dataset.

Training

Our CNN is trained end-to-end on a dataset of prostate scans in MRI. An example of the typical content of such volumes is shown in Figure 3.20. All the volumes processed by the network have fixed size of $128 \times 128 \times 64$ voxels and a spatial resolution of $1 \times 1 \times 1.5$ millimeters.

Annotated medical volumes are not easily obtainable due to the high cost associated with one or more experts manually tracing a reliable ground truth annotation. In this work we found necessary to augment the original training dataset in order to obtain robustness and increased precision on the test dataset.

During every training iteration, we fed as input to the network randomly deformed versions of the training images by using a dense deformation field obtained through a $2 \times 2 \times 2$ grid of control-points and B-spline interpolation. These augmentations were performed "on-the-fly", prior to each optimization iteration, in order to alleviate the otherwise excessive storage requirements. Additionally, we vary the intensities of the data during training to simulate the variety of data appearance from the scanner. To this end, we use histogram matching to adapt the intensity distributions of the training volumes used in each iteration to the ones of other randomly chosen scans belonging to the dataset.

Testing

A previously unseen MRI volume can be segmented by processing it in a feed-forward manner through the network. The output of the last convolutional layer, after soft-max, consists of a probability map for background and foreground. The voxels having higher probability (> 0.5) to belong to the foreground than to the background are considered part of the anatomy.

3.6.2 Experimental evaluation

We trained our method on 50 MRI volumes, and the relative manual ground truth annotation, obtained from the "PROMISE2012" challenge dataset [101]. This dataset contains medical data acquired in different hospitals, using different equipment and different acquisition protocols. The data in this dataset is representative of the clinical variability and challenges encountered in clinical settings. As previously stated we massively augmented this dataset through

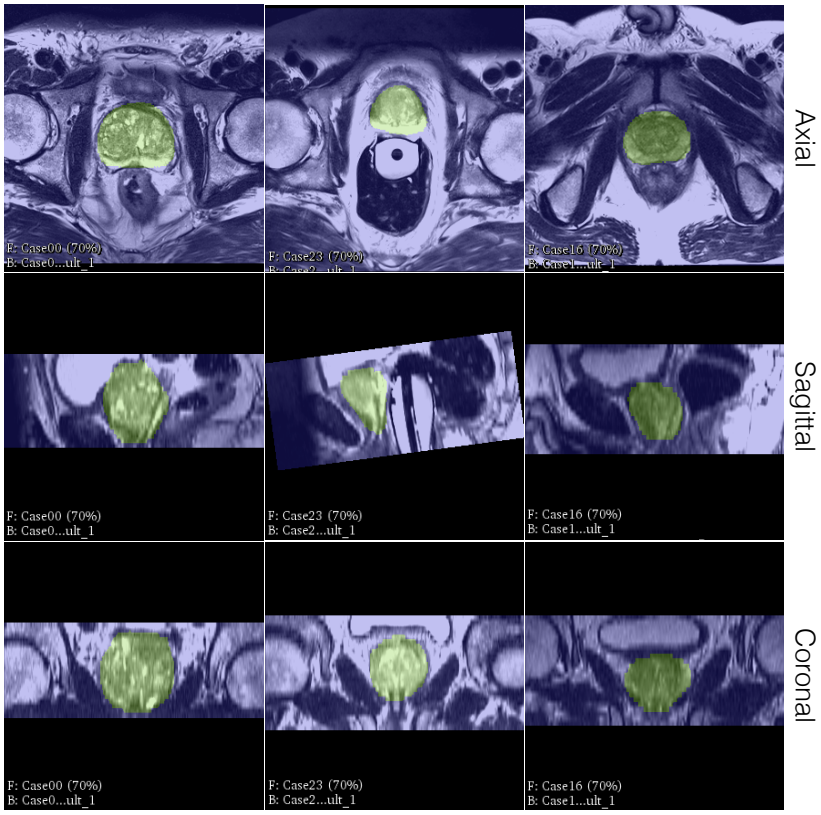

Figure 3.20: Qualitative results on the PROMISE 2012 dataset [101].

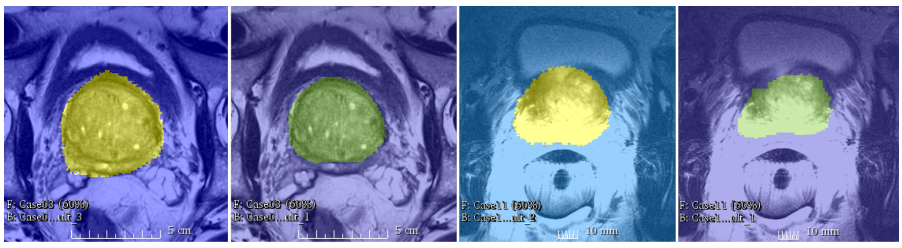

Figure 3.21: Qualitative comparison between the results obtained using the Dice coefficient based loss (green) and re-weighted soft-max with loss (yellow).

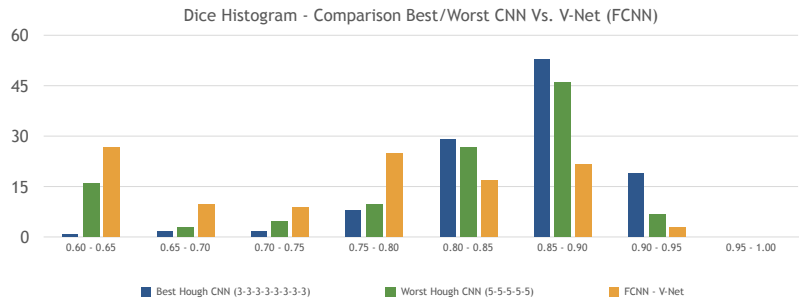

Figure 3.22: Comparison Dice coefficient distribution obtained by running our experiments on the ultrasound dataset using our best Hough CNN model (blue), our worst Hough CNN model (green), and V-Net (orange).

random transformation performed in each training iteration, for each mini-batch fed to the network. The mini-batches used in our implementation contained two volumes each, mainly due to the high memory requirement of the model during training. We used a momentum of 0.99 and an initial learning rate of 0.0001 which decreases by one order of magnitude every 25K iterations.

We tested V-Net on 30 MRI volumes depicting prostate whose ground truth annotation was secret. All the results reported in this section were obtained directly from the organizers of the challenge after submitting the segmentation obtained through our approach. The test set was representative of the clinical variability encountered in prostate scans in real clinical settings [101].

We evaluated the approach performance in terms of Dice overlap and Hausdorff distance between the predicted delineation and the ground truth annotation as well as the obtained challenge score, as computed by the organizers of "PROMISE 2012" [101] (cf. Table 3.9, Fig. 3.19).

Our implementation[3] was realized in python, using a custom version of the Caffe[4] [87] framework which was enabled to perform volumetric convolutions via CuDNN v3. All trainings and experiments ran on a standard workstation (64 GB RAM, 3.30GHz Intel® Core™ i7-5820K CPU, NVidia GTX 1080 with 8 GB VRAM). Model training ran for 48 hours, or 30K iterations circa, while segmentation of a previously unseen volume took circa 1 second. Datasets were first normalized using the N4 bias field correction function [162] and then resampled to a common resolution of $1 \times 1 \times 1.5$ mm. We applied random deformations to the scans used for training by varying the position of the control points with random quantities obtained from gaussian distribution with zero mean and 15 voxels standard deviation. Qualitative results can be seen in Fig. 3.20.

[3]Implementation available at `https://github.com/faustomilletari/VNet`
[4]Implementation available at `https://github.com/faustomilletari/3D-Caffe`

Method	Dice [0,1]	Failures
Best Hough-CNN	0.85	0%
Worst Hough-CNN	0.74	14%
V-Net (FCNN)	0.71	1%

Table 3.10: Comparison between Hough-CNN and V-Net on the ultrasound dataset.

Comparison with Hough CNN

In order to put our results into perspective we compare the performances of V-Net with the results achieved by our Hough-CNN, presented in chapter 3.5. In our comparison we kept all the hyper-parameters of V-Net fixed and trained the model for 20 thousand iterations, until convergence, on the same training set we employed to train Hough-CNN. When we evaluated the method on our training set we noticed that although the rate of failure (cases with Dice equal 0) was slightly lower, the contours were often leaking into regions that didn't belong to the midbrain and in some cases their shape was not resembling any of the training shapes. As a result, the performance of V-Net on this dataset was much inferior to the one of Hough-CNN. This can be observed in Figure 3.22 and Table 3.10 where the distribution of dice coefficients across the test set and quantitative results and respectively shown. In particular, the results obtained on the ultrasound dataset by the best and the worst architectures employed in the Hough CNN study have been compared to V-Net and have clearly shown superior performances.

3.6.3 Discussion

We presented and approach based on a volumetric convolutional neural network that performs segmentation of MRI prostate volumes in a fast and accurate manner. We introduced a novel objective function that we optimize during training based on the Dice overlap coefficient between the predicted segmentation and the ground truth annotation. Our Dice loss layer does not need sample re-weighting when the amount of background and foreground pixels is strongly unbalanced and is indicated for binary segmentation tasks. Although we inspired our architecture to the one proposed in [145], we divided it into stages that learn residuals and, as empirically observed, improve both results and convergence time. Future works will aim at segmenting volumes containing multiple regions in other modalities such as ultrasound and at higher resolutions by splitting the network over multiple GPUs.

Chapter 4

Detection

4.1 Introduction

Object detection is a popular topic in the computer vision community as it enables various applications such as scene parsing, robotic interaction, robotic manipulation, navigation for autonomous systems and many others. In computer vision, it is often necessary to retrieve an accurate estimate not only of the location but also of the pose of the detected objects. This allows robotic applications that involve grabbing tools or operating machinery in an autonomous manner. These systems need to be robust to challenges such as clutter, occlusions, lighting changes, contrast changes, etc.

In medical field object detection assumes a slightly different connotation as it is often used to detect very specific patterns, for example in MRI or US, which demonstrate the presence or absence of diseases, or specific artificial objects, such as catheters, placed in the body by surgeons. In particular we are interested in finding specific objects by localizing the position of their center of mass, or bounding boxes. The main challenges that we must address in object detection for medical image analysis are related to the presence of noise and artifacts that are specific to each imaging modality. In general medical data is difficult to interpret for reasons such as the lack of depth perception, potential overlaps and noise in X-Ray, the presence of shadows, signal drop regions and noise in ultrasound, poor resolution in MRI and high data dimensionality (3D or 4D data) in both MRI and CT.

When dealing with 2D images, such as X-Ray, it may also be necessary to acquire scans from different points of view and fuse the information obtained after object detection to retrieve more clinically 3D information.

In this chapter two works are presented. The first deals with automatic electrophysiology catheter detection in X-Ray fluoroscopic images, while the second presents an approach to estimate a 6 degrees of freedom pose of known objects from depth images in a manner that is robust to occlusion clutter and the other challenges that are often present in realistic settings.

4.2 Related Work

In this section we provide literature review for the two works presented in this part of the thesis. Due to the profoundly different application of the two detection approaches related work is presented separately for each of them.

4.2.1 Catheter Detection

In recent research practice, the medical imaging community has focused its efforts to localize catheters directly in C-arm images. Fallavollita et al. developed a catheter tip detection algorithm based on thresholds of the fluoroscopic images; this failed in low contrast images [59]. A technique for tracking and detecting the ablation catheter in X-ray images was first proposed by Franken et al. but the computational cost was relatively high making the method not applicable in clinic [61]. Coronary Sinus and ablation catheter detections were first proposed in [106, 104]. Multiple user interaction and parameter fine-tunings were necessary to meet the quality of the X-ray image. Employing respiration and motion compensation methods may succeed in overcoming some of the above challenges. Recently, Schenderlein et al. proposed a catheter tracking method using snakes active contour models [148]. Brost et al. developed a model-based lasso catheter tracking algorithm in biplane X-ray fluoroscopy [27]. However, the tracking required re-initialization and user interaction. Wen et al. successfully tracked one catheter in a cardiac cycle and required user-initialization in selecting tip electrodes [170, 171]. Multiple catheter-tip detections are presented in [177]. There, authors require user interaction for their detections using a geodesic framework. Finally, methods including fast blob detections, clustering, shape-constrained searching and catheter model-based detection have been proposed [117, 105]. A limitation of these is that they assume fixed shape for the catheter and might not cope with different C-arm positions and catheter shape changes due to foreshortening.

4.2.2 6DoF pose estimation

Pose estimation is essential for robotic object manipulation and augmented reality. Novel research has been enabled by the commercialization of cost effective depth sensing devices from various manufacturers. Several works addressing the problem of 6DoF pose estimation have been proposed recently. In [80, 143, 90] a method using holistic representation of the objects of interest is proposed. In this case templates are matched to the scene using an efficient algorithm trained on synthetic views. The main limitation of this approach is introduced by the presence of occlusions which impairs its recall and by the computational burden of processing a large number of objects. Other approaches such as [110, 72, 6] are introducing mechanisms to improve the robustness of the detections in presence of occlusions or noise using carefully designed descriptors which introduce additional computation. Voting based approaches have been proposed in [25] and [157] where the objective of finding the 6DoF pose of an object was achieved by relying only on image patches which often contain only a small portion of the object and therefore provide

robustness to occlusions and presence of specular reflections while consistency and global context is guaranteed implicitly by the voting strategy. These work make use of random Hough forests which rely on handcrafted features.

With the recent advancements in the field of deep learning a plethora of methods has been introduced to perform image classification, regression, segmentation and also to obtain reliable and compact descriptors for images using strategies based on auto-encoders. Due to the capability of deep learning of capturing hierarchical features directly from the data at hand, handcrafted features have gradually been abandoned and replaced in most recent approaches. [69, 68] employed SVM to classify features computed through a convolutional neural network on candidate object region. In [169] the capability of CNNs of learning adequate descriptors for pose estimation from RGBd images is demonstrated. The main innovation of [169] is the fact that it uses a triplet loss for this particular application.

4.3 EP catheters detection and tracking

Catheter guidance is a vital issue for the success of electrophysiology interventions. It is usually provided through fluoroscopic images that are taken intra-operatively. The cardiologists, who are typically equipped with C-arm systems, scan the patient from multiple views rotating the fluoroscope around one of its axes. The resulting sequences allow the cardiologists to build a mental model of the 3D position of the catheters and interest points from both views. An approach to perform automatic detection and tracking of electrophysiology (EP) catheters in C-arm fluoroscopy sequences is proposed in this section. This method is fully automatic and can concurrently track an arbitrary number of overlapping catheters. After a pre-processing step, sparse coding is employed to first detect candidate catheter tips, and subsequently detect and track the catheters. The proposed technique is validated on 2835 C-arm images, which include 39,690 manually selected ground-truth catheter electrodes. Results demonstrated sub-millimeter detection accuracy and real-time tracking performances.

4.3.1 Motivation

Sudden cardiac death (SDC) is linked to severe disorders of the heart rhythm. In the United States alone, the incidence rate ranges up to 450,000 cases annually [47]. In some cases, patients affected by heart beat related diseases can be definitively treated with radio-frequency (RF) catheter ablation. The efficacy of catheter ablation is highly dependent on accurate identification of the site of origin of the arrhythmia. Once this site has been identified, an ablation catheter is positioned in direct contact with it and radio-frequency energy is delivered to ablate it.

Catheter ablation is often a long procedure requiring significant fluoroscopy exposure. It was proved recently [53], that 3D navigation systems contribute to the reduction of the exposure to patients and operators. The common mapping technologies that combine 3D anatomy and electrophysiological data are: CARTO and CARTOMerge (Biosense Webster), NavX (St.Jude Medical), and RPM (Cardiac Pathways-Boston Scientific). Other technologies that provide continuous data of all electrophysiogical events include Ensite 3000 (St. Jude Medical) and Basket (Cardiac Pathways-EP Technologies) [29]. Whether using mapping systems or conventional RF ablation techniques, clinicians still rely on C-arm images to position and guide catheters. Thus, exploiting C-arm image information is crucial for providing additional information to clinicians during cardiac ablation procedures. There are several reasons as to why detecting and tracking the position of ablation catheters relative to the patient anatomy is important. They are related to interventional guidance aspects: (i) accounting for heart motion compensation, (ii) easing positioning & navigation during cardiac ablation, (iii) planning the ablation procedure by (iv) registration to preoperative data such as CT and MRI.

We propose a unique method that considers all of the key challenges associated with catheter detections. Our method: (i) is fully automatic; (ii)

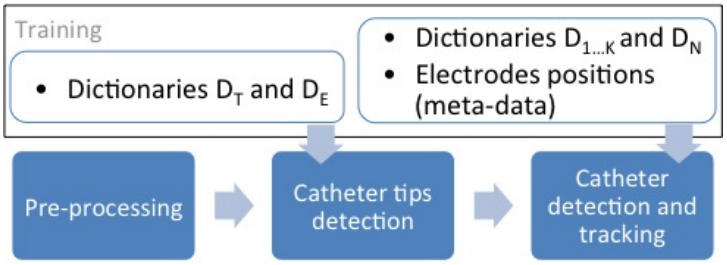

Figure 4.1: Proposed pipeline.

supports the presence of multiple, touching and overlapping catheters; (iii) can detect and track catheters appearing foreshortened or deformed; (iv) is robust to illumination variations and to the sudden motion of the catheters. Although implemented in a non-efficient manner, the algorithm achieves almost real-time performances.

4.3.2 Method

Our catheter tracking and detection pipeline is shown in *Figure* 4.1. The pre-processing step aims to improve the image signal to noise ratio and to reduce the search space. A further reduction of the search space is obtained in the catheter tips detection stage, where image locations corresponding to catheter tips are selected. In the final step, we detect and track the catheters by the means offered by sparse coding. Catheter hypotheses are formed and associated to a cost, the ones yielding the minimal global cost constitute the output of our algorithm.

In our approach, we use sparse coding, that was first introduced in Section 2.2.4, to solve our catheter tracking by detection task. We only allow conical combinations of dictionary atoms as, due to the nature of our problem, different atoms should be never subtracted from each other to obtain a better reconstruction. The dictionaries used in this work contain the appearance of the catheters and of the electrodes. We rely on the sparsity assumption to match the candidate appearances with a few, specific ones stored in the dictionaries.

Pre-processing

In order to cope with the presence of noise and improve the contrast of the fluoroscopic images, we apply to the images an homomorphic filter [130] followed by a bilateral filter [158], reducing noise artifacts while preserving edges. As a further pre-processing step, we use a determinant of hessian blob detector to obtain the accurate location of electrode-like structures appearing in the images. As demonstrated by [117, 105], the electrodes can in this way be localized with sub-millimeter precision, therefore enabling us to effectively limit the search space.

Training

In our method, we employ two sets of dictionaries to: (i) select image locations corresponding to the tips of the catheters, (ii) reconstruct and associate a cost to each candidate catheter. The dictionaries are obtained in a training stage that makes use of annotated data.

Training dictionaries for "tips" detection

In order to detect the catheter tips, we instantiate the dictionaries $\mathbf{D}_T$ and $\mathbf{D}_E$, respectively built from patches depicting catheter tips and electrodes at various orientations. The patches are normalized to have zero mean and unit standard deviation so that illumination invariance and uniform probability of being selected during reconstruction are ensured.

Training dictionaries for catheters detection

In our approach, detection and tracking are coupled tasks. Supposing we want to track K catheters, we train:

1. K dictionaries $\mathbf{D}_{1...K}$, one for each typology of catheter, of *positive templates* capturing the appearances of each catheter separately.
2. one dictionary $\mathbf{D}_N$ of *negative templates* capturing typical background appearances.

The words $\mathbf{d}_{jk}$ of each dictionary $\mathbf{D}_k$ are associated with the specific poses assumed by the $k-th$ catheter during training. We also instantiate *meta-data* matrices $\mathbf{M}_j$, whose purpose is to establish a correspondence between the 1D intensity profiles of the catheters and the expected locations of the catheter's electrodes. In this way it is possible to recover the position of electrodes that have been missed in the previous step. The coordinates stored in $\mathbf{M}_j$ are normalized to a common orientation and expressed with respect to the catheter's tip position. The negative profiles stored in $\mathbf{D}_N$ are used during tracking to penalize candidate catheters whose appearances resemble the background. All the appearances stored in the dictionaries consist of 1D intensity profiles of fixed length r, sampled from training images. The intensity profiles, which are implicitly rotation invariant (they stay the same regardless the orientation of the catheter), are normalized to have zero mean and unit standard deviation.

Tracking by detection

We want to detect and track K catheters through a fluoroscopic sequence. The output of the pre-processing step of our algorithm is a set of key-points $\mathbf{X} = \{\mathbf{x}_1 ... \mathbf{x}_p\}$ (*Figure* 4.2a). Once small image patches $\mathbf{y}_i$ are extracted around the $\mathbf{x}_i$ (*Figure* 4.2b), the ones that correspond to catheter tips can be discriminated by solving the following two problems:

$$\hat{\boldsymbol{\alpha}}_t = \min_{a_t} \|\mathbf{D}_T \boldsymbol{\alpha}_t - \mathbf{y}_i\|_2^2 + \lambda_1 \|\boldsymbol{\alpha}_t\|_1 , \; s.t. \; \boldsymbol{\alpha}_t \geq 0 \tag{4.1}$$

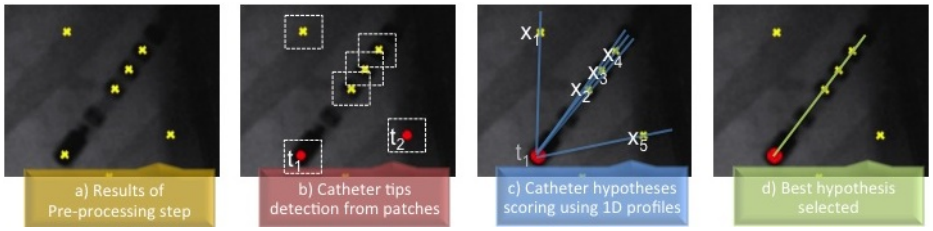

Figure 4.2: Main steps of our algorithm. The output of each step is fed into the next.

$$\hat{\boldsymbol{\alpha}}_e = \min_{a_e} \|\mathbf{D}_E\boldsymbol{\alpha}_e - \mathbf{y}_i\|_2^2 + \lambda_2 \|\boldsymbol{\alpha}_e\|_1 , \; s.t. \; \boldsymbol{\alpha}_e \geq 0. \tag{4.2}$$

Key-points associated to patches that have been reconstructed better with $\mathbf{D}_T$ than with $\mathbf{D}_E$, are regarded as catheter "tips" according to

$$\mathbf{T} = \{\mathbf{t}_1 ... \mathbf{t}_{N \geq K}\} = \left\{\mathbf{x}_i : \|\mathbf{D}_T\hat{\alpha}_t - \mathbf{y}_i\|_2^2 < \|\mathbf{D}_E\hat{\alpha}_e - \mathbf{y}_i\|_2^2\right\}. \tag{4.3}$$

In the final step of our pipeline, we aim to formulate and score catheter hypotheses (*Figure* 4.2c). Each catheter tip $\mathbf{t}_n$ yields as many catheter hypotheses as the number of neighboring key-point $\mathbf{x}_i \in \mathbf{X}$ falling within a distance r. The catheter hypotheses are intensity profiles $\mathbf{l}_{ni}$ extracted from lines of length r originated in $\mathbf{t}_n$ and intersected with each $\mathbf{x}_i$ in turn. For each $k = 1...K$ we aim to solve the following problems:

$$\hat{\alpha}_{ni}^k = \min_{\alpha_{ni}^k} \left\|\mathbf{D}_k\boldsymbol{\alpha}_{ni}^k - \mathbf{l}_{ni}\right\|_2^2 + \lambda_3 \left\|\boldsymbol{\alpha}_{ni}^k\right\|_1 , \; s.t. \; \boldsymbol{\alpha}_{ni}^k \geq 0 \tag{4.4}$$

$$\hat{\beta}_{ni}^k = \min_{\beta_{ni}^N} \left\|[\mathbf{D}_N, \mathbf{D}_{j\neq k}]\beta_{ni}^k - \mathbf{l}_{ni}\right\|_2^2 + \lambda_4 \left\|\beta_{ni}^k\right\|_1 , \; s.t. \; \beta_{ni}^k \geq 0. \tag{4.5}$$

We aim to assess, through (4.4), the similarity of each catheter hypothesis with the *k-th* catheter and, through (4.5), its similarity with the background or with catheters having label different than k.

Furthermore, we identify the biggest element of α_j of $\hat{\alpha}_{ni}^k$, and we retrieve the associated meta-data $\mathbf{M}_j = [\mathbf{m}_1 ... \mathbf{m}_Q]$, containing the expected, approximated and pose specific (in terms of out-of-plane rotation of the catheter) coordinates of the electrodes. When a catheter hypothesis corresponds to a true catheter, the coordinates $\mathbf{m}_j$ and $\mathbf{x}_i$ are spatially close. The minimal distances $d_i = \min_q \left(\|\mathbf{x}_i - \mathbf{m}_q\|\right)$ between each point $\mathbf{x}_i$ (after normalization to the orientation of $\mathbf{l}_i$) and the points stored in $\mathbf{M}_j$, are obtained.

The errors $E_P = \left\|\mathbf{D}_k\hat{\boldsymbol{\alpha}}_{ni}^k - \mathbf{l}_{ni}\right\|_2^2$ and $E_N = \left\|[\mathbf{D}_N, \mathbf{D}_{j\neq k}]\hat{\beta}_{ni}^k - \mathbf{l}_{ni}\right\|_2^2$, and the coefficient $d = \sum_i d_i$ determine the cost of a candidate catheter according to

$$E_{ni}^k = \begin{cases} d\, E_P & if \; E_P \geq E_N \\ d\, \frac{E_P}{E_N - E_P} & if \; E_P < E_N \end{cases}. \tag{4.6}$$

For each tip $\mathbf{t}_i$, the best catheter hypothesis that could be reconstructed using $\mathbf{D}_k$ is retained (*Figure* 4.2d) and its cost $\hat{E}^k_{ni}$ is stored in a matrix $\mathbf{C} \in \mathbb{R}^{K \times N}$ modeling associations between labels and catheter hypotheses. The hungarian method is employed to select K catheter hypotheses yielding the lowest total cost. Please note that the presence of the meta-data is not only beneficial to score the catheter hypotheses but can be used to effectively recover missed electrodes detections.

Mild temporal consistency can be enforced to favor catheter hypotheses occurring at similar position over time. This is realized by counting how many consecutive times a catheter k appears in a neighborhood (radius g) of its previous position and dividing the error E^k_{ni} by this number. If the *k-th* catheter moves abruptly, the counter associated with its previous position is decreased until it reaches zero.

4.3.3 Experimental evaluation

A total of 2835 C-arm images, belonging to 20 sequences acquired from two views were analyzed. A reference, a pacing and an 8-French ablation/mapping catheter are visible in the sequences. The image sizes are 512 × 512 with a pixel spacing of 0.44 mm. The X-Ray beam energy was varied between 70-92kV to ensure variability within the data. Ground truth annotation, which included the position of the 39690 electrodes appearing in the sequences, was provided by two observers. The model's parameters were fixed experimentally to be $\lambda_1 = 10$, $\lambda_2 = 150$, $\lambda_3 = \lambda_4 = 1$ for all the experiments. The scale of the blob detector was fixed to $\sigma = 4$. We enforced temporal consistency fixing the quantity g to $8px$ during all the experiments. Since our method requires a training phase, we assessed the performances of our approach when different amount of training data is used. The training images are selected from a sequence that is never used for testing.

Catheter detection and tracking

We assessed the performances of our method to detect and track the mapping, pacing and reference catheter respectively. The results are shown in *Table* 4.1. We evaluated, in particular, the impact of the number of annotated examples used during training on the performances. The pacing and reference catheters that experience little foreshortening and deformations are already well detected using a few training examples while the mapping catheter requires an higher number of training examples due to its frequent out-of-plane rotations. Incrementing the number of training examples the performances improve up to values close to 100%. The computation time increases with the dimension of the dictionaries. When 100 images are used during training, the processing time for one frame is 0.7 seconds using our MATLAB prototype and circa 0.08 seconds using our more optimized C++ implementation.

Table 4.1: Tracking and detection results. A different number of training examples was used in each test.

	A/P View (%)			Lateral View (%)		
Training set	Mapping	Pacing	Reference	Mapping	Pacing	Reference
3 examples	77.49	98.38	98.17	53.74	96.74	97.52
10 examples	87.13	99.79	98.38	78.86	98.02	99.08
20 examples	88.05	99.79	99.30	89.16	97.95	98.87
50 examples	93.46	99.79	99.36	89.31	**98.09**	99.01
100 examples	**93.95**	**99.79**	**99.51**	**90.02**	97.95	**99.36**

Table 4.2: Detection accuracy in pixels and millimeters.

	A/P View		Lateral View	
	Pixels	Millimeters	Pixels	Millimeters
mapping	1.17 ± 0.64	0.51 ± 0.28	1.28 ± 0.35	0.56 ± 0.15
pacing	1.48 ± 0.60	0.65 ± 0.26	1.29 ± 0.23	0.56 ± 0.10
reference	1.63 ± 0.75	0.71 ± 0.33	1.49 ± 0.22	0.65 ± 0.09

Detection accuracy

The accuracy of the catheters detections in terms of distance of the electrodes from the ground truth annotation was assessed. The achieved results are shown in *Table* 4.2.

4.3.4 3D multi-view catheter reconstruction

When multiple X-Ray sequences are acquired through a C-Arm from a different points of view, it is possible to reconstruct the 3D configuration of EP catheters by (i) detecting the catheters and their electrodes in each view separately, (ii) using epipolar geometry and the known transformation matrices relative to each view and the intrinsic parameters of the system. In [15] we develop both an approach based on epipolar geometry and a method that additionally incorporates prior knowledge about the catheter shapes. This helps increase robustness to deformations and other motion that might have happened between the two acquisitions from different viewpoints needed for the algorithm to run. In this way we are able to demonstrate the performance of our detection method on the complete surgical workflow and highlight the advantages of performing 3D multi-view catheter reconstruction with prior knowledge of their shapes.

4.3.5 Discussion

A novel method to detect and track linear EP catheters, that may appear foreshortened or occluded, in fluoroscopic images was presented. The approach, that is based on $\ell 1$-sparse coding is robust to catheter overlap and has great potential in correcting for patient motion when used in conjunc-

tion with anatomical overlays. Future work will focus on the development of unique methods to automatically reconstruct catheters from [57, 58] single or multi-view C-arm fluoroscopy images. The technique would rely on no user interaction, high clinical accuracy, and real-time performance. Alternatively the detection of catheter electrodes can be coupled with generative probabilistic models that optimizes correspondence and subsequent 3D reconstructions of the catheters.

Figure 4.3: Example of detection of multiple objects in a RGBd image. On the left, the representation of vote accumulation (red means high number of votes), in the center approximate segmentation resulting from the back-projection of votes and patch-masks scaled by the vote weight, on the right detection result with pose.

4.4 Pose estimation through voting

Six degrees of freedom pose estimation from RGBd images finds application in fields such as robotics and augmented reality. We propose a method to infer the pose of objects present in a scene in a scalable and robust manner through voting. We train a convolutional auto-encoder to perform reconstruction of a large dataset of RGBd patches and use the resulting compact representation of each patch as a descriptor. Descriptors are computed on a dataset of synthetic RGBd training patches annotated with pose and stored in a database. When a new image is supplied to the system the descriptors for its patches are computed and compared with the descriptors in the database. Votes are therefore cast and filtered to obtain detections and relative pose (Figure 4.3).

4.4.1 Motivation

In order to build robustness towards the presence of noise, occlusions, specular reflections, local illumination changes we propose to tackle 6DoF pose estimation using a patch-based approach and voting. Although the first step of the computation involve only local information and ignores the global context represented by the knowledge of the whole object at hand, this context is retrieved back after vote casting, as votes are able to accumulate correctly into a peak only if a high percentage of patches are recognized and cast consistent votes. Moreover, we put in place a refinement approach to avoid false positive detection and we perform hypothesis verification at the end of our processing pipeline. Moreover, during learning stage we use only synthetic data and we avoid learning background.

4.4.2 Method

This section is devoted to the description of the approach. We describe the way RGBd patches are extracted from a grid imposed over the image and at the appropriate scale. Then we give an overview of the convolutional neural network used in this work and last we present the voting strategy making use

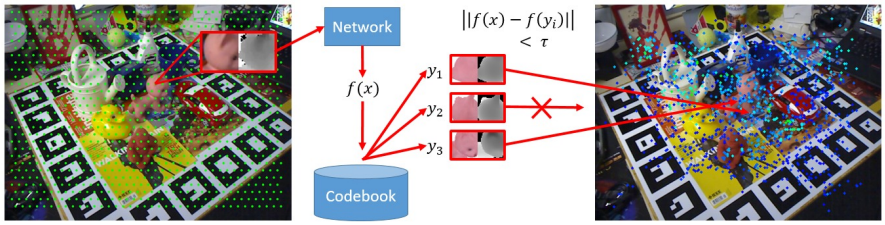

Figure 4.4: Detection and pose estimation pipeline. Patches are extracted from a grid over the image. Descriptors are computed and compared with the ones learned from synthetic images and contained in a database. Votes are cast using the information in the database, filtered and detections with pose produced.

of the learned descriptors. A schematic representation of our method is shown in Figure 4.4.

RGBd patch sampling

We extract training data by sampling patches from both synthetic renderings from CAD models of the objects of interest seen from viewpoints uniformly sampled on an icosahedron surrounding the 3D object model, and real images.

The patches are obtained at the same scale since we sampled them such that their side measures 5 centimeters. To do so we rely on the depth sensed at the pixel corresponding to the center of the patch. The depth data of the patch is then centered and clamped to confine it in a specific value range. Colors and depth are normalized and the patch is resized to a conventional size of 32 by 32 pixels.

By using patches we are able to avoid the background modeling that is necessary when holistic approaches are used. Very small portions of the background which can be present in the periphery of patches that lie on the silhouette of the object are eliminated. This is possible since we have a CAD model of the object, therefore we know the exact boundary of the object in the image. This is profoundly different from what other learning based method which look at the whole object at once do. Such approaches need to resort to particular strategies such as hard negative mining to be able to cope with this issue.

The patches extracted from real images, specifically belonging to the LineMod dataset, are used to trained our convolutional auto-encoder which is responsible for descriptor computation. These patches belong to both objects and background regions. The patches extracted from synthetic renderings are also associated to information about specific object pose. These patches along with the associated pose are used to produce a database of feature and 6DoF votes which is used during inference for detection and pose estimation.

A schematic representation of this process is provided in Figure 4.5.

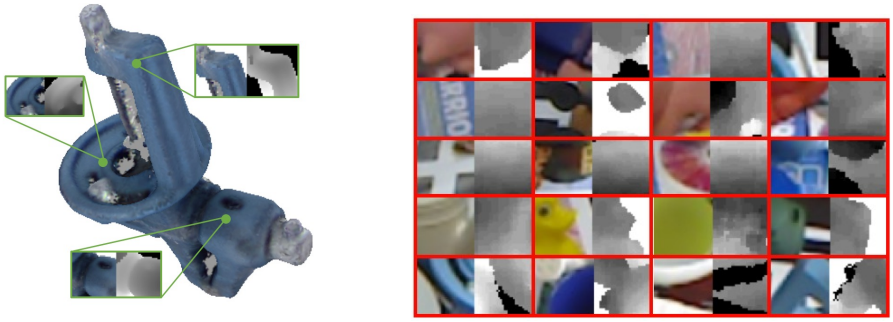

Figure 4.5: Schematic representation of training patch extraction from renderings.

Convolutional neural network

Our neural network needs to be trained on the data extracted from real scenes belonging to the LineMod dataset in order to produce descriptors that are suitable to be included in the voting strategy that lies at the core of our approach. We propose to use a convolutional auto-encoder (Figure 4.6) trained from scratch on the patches extracted in the previous step. The reason why we do not rely on a publicly available network pre-trained on a large dataset such as image-net and then refine it on our dataset are multiple.

- Although low level features of the first few layers of such networks capture common features that are most probably adequate for our task, higher level features that sense objects and complex patterns are most probably not useful in our approach.
- The presence of an additional channel in our data, namely depth, complicates the adoption of existing models.
- Using patches allow us to extract millions of training examples from each image of the LineMod dataset.

Multiple authors have highlighted and taken advantage of features extracted by deep layers of neural networks trained to solve classification tasks. In this case it is not possible to retrieve meaningful classes from our patch dataset and therefore we opt to train a convolutional auto-encoder on these images using a loss layer which enforces high quality reconstructions. The work presented in [169] learns features descriptive of the object pose pose **after it has been detected**. This goal is different from ours. While [169] proposes to enforce a correspondence between distance in feature space and pose space, our aim is to simply obtain a compact representation of object patches and then use a vote strategy to do **both** object detection and pose estimation.

Through the convolutional auto-encoder we aim to obtain descriptors that accomplish this task. An auto-encoder aims to minimize the reconstruction error $E = \|x - y\|_2^2$ where x are the input patches and y the reconstructions outputted by the network. Our neural network architecture employs two 5×5

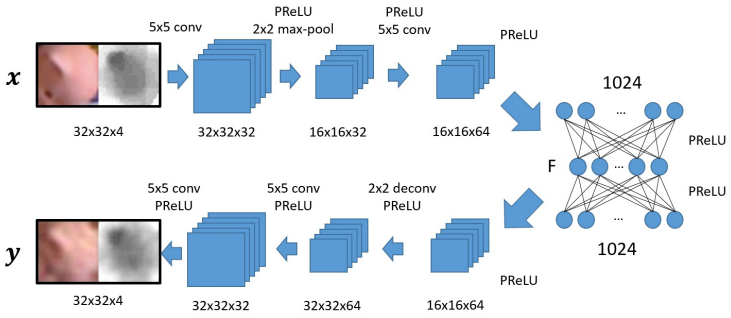

Figure 4.6: Schematic representation of the chosen convolutional auto-encoder architecture.

convolutional layers and 1 fully connected layer to produce the features F (having variable dimensionality as shown in the experiments) as a result of the compressing path of the network and a encoder-decoder-like arrangement of layers for the expanding path which produces the image output y. Pooling layers in the compressing path are replaced with 2×2 de-convolutions applied with stride 2.

After training, we use this neural network for (i) producing a database of object specific descriptors associated with 6DoF information in form of a vote relative to the pose and (ii) doing inference on previously unseen patches.

Pose estimation and voting

As previously mentioned, pose estimation is performed through a voting strategy that relies on an object-specific database containing 6D votes in the form $v = [t_x, t_y, t_z, \alpha, \beta, \gamma]$ and a feature vector $\mathbf{f}$ obtained from the auto-encoder network at the end of its compressing path. The vote can be understood as a mixture of two pieces of information: spatial information $[t_x, t_y, t_z]$ that encodes the spatial displacement between the center of each patch and the centroid of the object in 3D, and angular information $[\alpha, \beta, \gamma]$ which represents 3 angles expressing the pose of the object. Building these databases relies on synthetic renderings and patch sampling.

Object detection and pose estimation in a novel image is done by imposing a sampling grid on the image, extracting patches from each position and use these patches to obtain feature vectors similar to the ones included in the database from the auto-encoder. K nearest neighbors in feature space are selected from the database and their voting information is used. Their contributions are weighted by the reciprocal of the Euclidean distance in feature space. The spatial information $[t_x, t_y, t_z]$ is added to the sampling position of the patch $[s_x, s_y, s_z]$ to find the object centroid, and the angular information is accumulated at the centroid. In order to increase efficiency, we quantize all the votes and accumulate them in a 2D coarse grid. The votes contained in scarcely populated bins of the 2D grid are ignored. All the other votes are considered and mean shift filtering is performed both in spatial and quaternions space to find peaks in the vote-map which are understood as true detections.

Table 4.3: Results of our approach on the dataset used in [157]

Sequence	[80]	[157]	Ours
Camera	**0.589**	0.394	0.383
Coffee	0.942	0.891	**0.972**
Joystick	0.846	0.549	**0.892**
Juice	0.595	**0.883**	0.866
Milk	**0.558**	0.397	0.463
Shampoo	**0.922**	0.792	0.910
Total	0.740	0.651	0.747

It is possible to obtain also a coarse but reasonable segmentation of the object(s) at hand by back-projecting the votes that resulted in a true detection to the position they originated from and apply there a binary segmentation patch (which can be obtained when the database is built using synthetic renderings) scaled by the vote weight. An example of this can be seen in Figure 4.3, central panel.

4.4.3 Experimental evaluation

We evaluate the approach on challenging data in order to benchmark it and highlight its performances in comparison with other state-of-the-art approaches. We tested our approach on the dataset proposed by Tejani in [157], the LineMOD dataset [80] and on the challenge dataset used in [5] where different detectors multiple cues were employed to achieve robust results. These datasets are linked with state-of-the-art work in RGBd object detection and pose estimation which use different constraints and assumptions in their evaluation procedures. In order to obtain fair results during comparison we adopt the same conventions and assumptions of the original methods we use to benchmark ours. When we compare with [157] we follow their protocol of extracting N=5 stongest modes in the voting space and subsequently verify them via ICP and normal vectors check, with the goal of suppressing false positives. For what concerns the comparison with [80] we retain all the detections having a similarity score larger than 0.8. We have observed that this delivers, as expected, excellent results for LineMOD when the objects are not occluded. When we apply our approach we remove the last stage of the voting strategy described previously and we take the N=100 most confident votes to formulate our hypotheses.

The results achieved on the [157] dataset are summarized in Table 4.3 and put into perspective by comparing them with the ones achieved by other approaches. The results obtained on the LineMOD dataset are shown in Table 4.4. Finally, the results achieved on the challenge dataset associated to the work presented in [5], are reported in Table 4.5.

Table 4.4: Results of our approach on the dataset used in [80]

Sequence	[80]	[90]	[143]	[83]	Ours
ape	95.8	96.1	95.0	93.9	**96.9**
bench-vise	98.7	92.8	98.9	**99.8**	94.1
bowl	**99.9**	99.3	99.7	98.8	**99.9**
cam	97.5	97.8	**98.2**	95.5	97.7
can	95.4	92.8	**96.3**	95.9	95.2
cat	**99.3**	98.9	99.1	98.2	97.4
cup	97.1	96.2	97.5	99.5	**99.6**
drill	93.6	**98.2**	94.3	94.1	96.2
duck	95.9	94.1	94.2	94.3	**97.3**
eggb	99.8	99.9	99.8	**100**	99.9
glue	91.8	96.8	96.3	**98.0**	78.6
hole puncher	95.9	95.7	**97.5**	88.0	96.8
iron	97.5	96.5	98.4	97.0	**98.7**
lamp	97.7	98.4	97.9	88.8	96.2
phone	93.3	93.3	**95.3**	89.4	92.8
Average	96.6	96.5	**97.2**	95.4	95.8

Table 4.5: Results of our approach on the challenge dataset [5]

Method	Precision	Recall	F1-Score
GHV [5]	1.0	0.998	0.999
Tang [156]	0.987	0.902	0.943
Xie [174]	1.0	0.998	0.999
Aldoma [6]	0.998	0.998	0.997
Ours	0.941	0.973	0.956

4.4.4 Discussion

This approach demonstrates the capabilities of voting strategies for object detection and pose estimation tasks using RGBd data. The idea presented here is very similar to other works presented in this thesis and further demonstrates the flexibility of Hough voting strategies. Moreover, we demonstrate the usage of a deep convolutional auto-encoder for the crucial task of producing features driving the voting approach.

Chapter 5

Tracking

5.1 Introduction

Visual tracking has been a central topic in computer vision research since a few decades. Being able to track an object over time using only visual information allows a number of applications, which are interesting both under the academic and the industrial point of view.

Video surveillance, augmented reality, robotic applications, motion tracking and behavioral studies are just a few examples that involve visual tracking as a relevant part of the task. Also in medical field, tracking is necessary for a number of tasks. Its application in this field ranges from surgical tools visual tracking (applied for example in minimally invasive - image guided - surgical procedures) to work-flow modeling where the goal is to track the staff in the operating room during medical procedures, in order to analyze the intervention itself and be able to aid the team.

In the last few years the computer vision community proposed a number of high relevance works about visual tracking. These approaches were able to perform reliably in challenging situations and were tested on a number of sequences featuring one or multiple challenges. Even though the approaches are multiple, their goal is always the same: estimate the state of an object over time.

Object can be occluded or out of focus, can deform, change color, exhibit different surface characteristic such as reflections and therefore their properties cannot be captured fully by any analytic model. Moreover the object motion can be very different from sequence to sequence: some objects move slowly while others move very fast, their trajectory can be smooth or can be characterized by abrupt changes.

The imaging process itself reduces the amount of information that is exploitable by the tracker: noise is always present and corrupts the images in an unpredictable way while the analog to digital conversion operated by the components of the camera introduces quantization noise. In addition, when dealing with 2D images, algorithms must rely only on projections of the structures of the 3D real world onto the image plane.

As of today, there is not a single tracker that can address at once all the possible challenges and the variety of phenomena that can affect target objects in real life settings.

Despite this, there are a number of trackers that aim to cope with a very specific subset of these challenges and are designed to work in very specific environment. Trackers that are used in factories' production lines, for example, perform in a reliable way as a result of the availability of some prior knowledge about the object being tracked. The method presented in Section 4.3, belongs also to the category of object-specific trackers since it aims to track specific catheters having a specific electrode pattern in X-Ray sequences.

The aim of a generic tracker is more general. The goal that must be solved is to track objects of which no prior knowledge about the object is available. The only information provided to track the object is the initialization supplied by the user or some other mechanism in the first frame of the sequence. A tracker that is equally good in tracking humans, cars and animals at the same time in different scene setting is not straightforward to realize.

In this section, a generic tracking algorithm relying on Hough voting and dictionary learning will be presented. This approach was tested against a number of challenging video sequences and proved to surpass or match the performances of most recent approaches.

5.2 Related Work

Current approaches can be grouped into two classes [176]:

- discriminative trackers, which make use of a classifier to distinguish the object of interest from the background. This includes tracking methods based on segmentations and bounding boxes;
- generative trackers, which rely on appearance models to capture and match the visual characteristics of the object of interest across the frames. In this case, most approaches work by managing the uncertainty around the exact location of the object using strategies such as particle filter.

Recent examples of discriminative methods [66, 63] made use of a voting strategy to track deformable objects while ensuring robustness towards occlusion. Generative approaches such as [175, 181, 109] rely on a set of object templates stored in a dictionary to maintain the appearances of the object of interest and on sparse coding to robustly recognize it.

The state of an object can be parametrized using a number of parameters. This depends on the complexity of the motion of the object and on the desired precision of the tracking results. The state space can be parametrized using 2 parameters, which are able to model all the possible translation of the object onto the image plane, it can be 4 dimensional defining the parameters of a rigid transformation, it can be 6 dimensional modeling affine transformations or 8 dimensional when the aim is to model projective transformations.

Discriminative approaches to visual object tracking make use of classifiers to produce binary [8, 88] or structured predictions [63, 66, 74, 64] and therefore distinguish the object of interest from the background. The most relevant factors influencing the performance of the trackers are the discriminative capabilities of the features, the choice of the learning algorithm, and the online update strategy. In [8] a classifier based on boosting was updated online using multiple instance learning, while [88] proposed to integrate structural constraints in order to limit the impact of data-samples that are unlikely to be related to the target during update. More recently [74] employed a kernelized structured output Support Vector Machine (SVM) to regress the transformation of the bounding box between subsequent frames. In [64], Gaussian Process Regression (GPR) was employed to discriminate the target position from background points, by means of a semi-supervised approach that learns the discriminative model from both previously seen samples as well as unseen candidates directly extracted from the current frame. Hough forests have been used [63, 66] to jointly perform classification and localization of the target bounding box. Data-points are classified and, depending on their label, they are enabled to cast votes which localize the target. Recently, Convolutional Neural Networks (CNN) have also been employed to classify between target and background [99].

Generative approaches such as [18, 23] model the appearances of the object of interest using histograms and ensure robustness using a fast segmentation strategy which prevents the appearances of portions of the background from triggering failures. In [18] a Graph Cut-based segmentation method was

employed in each frame to obtain reliable histograms. In [23] a probabilistic framework was developed to obtain Maximum-A-Posteriori estimations (MAP) of both a level set-based segmentation contour wrapping the object of interest, as well as an affine transformation accounting for rigid object motion. Other generative methods [11, 181, 182, 175] make use of a dictionary of target object templates and sparse coding to score candidate bounding boxes positions. In each frame, patches are collected from the image and sparsely reconstructed through the dictionary. The reconstruction fidelity serves as a likelihood of candidate patches to depict the object of interest.

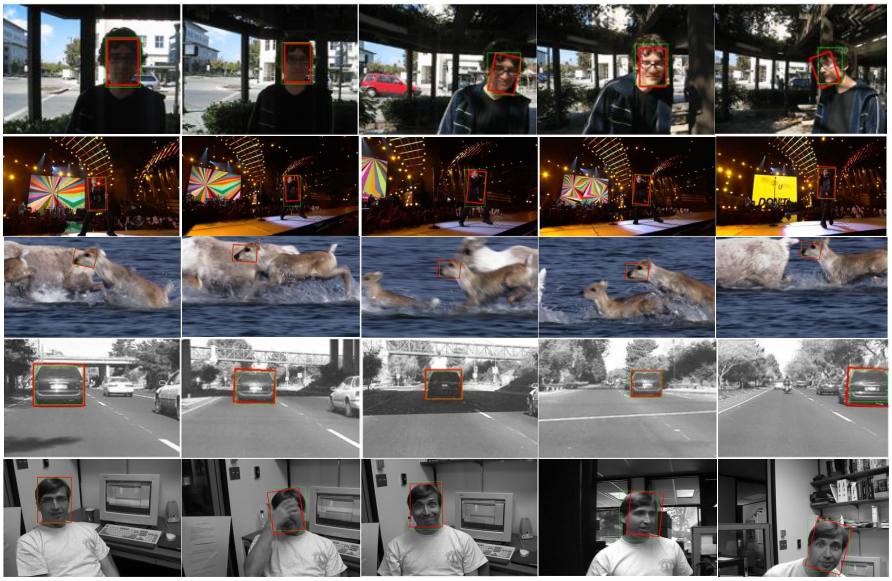

Figure 5.1: Qualitative results on the sequences 'Trellis', 'Singer2', 'Deer', 'Car4' and 'Dudek'. Our results are highlighted in red, manual annotation from the benchmark sequence is depicted in green.

5.3 Hough-dictionaries for visual tracking

We propose a novel approach to online visual tracking that combines the robustness of sparse coding with the flexibility of voting-based methods. Our algorithm relies on a dictionary that is learned once and for all from a large set of training patches extracted from images unrelated to the test sequences. In this way we obtain basis functions, also known as atoms, that can be sparsely combined to reconstruct local image content. In order to adapt the generic knowledge encoded in the dictionary to the specific object being tracked, we associate a set of votes and local object appearances to each atom: this is the only information being updated during online tracking. In each frame of the sequence the object's bounding box position is retrieved through a voting strategy. Our method exhibits robustness towards occlusions, sudden local and global illumination changes as well as shape changes. We test our method on 50 standard sequences obtaining results comparable or superior to the state of the art.

5.3.1 Method

We implement visual object tracking by means of a universal dictionary, learned offline, together with a specific voting strategy.

The algorithm comprises three steps: *Offline Dictionary learning* — Where we learn a dictionary of visual words from a large set of randomly sampled image patches, with the goal of obtaining a set of basis functions (i.e. atoms) capable of reconstructing a large variety of local image appearances.

Tracker Initialization — Aiming at adapting the generic knowledge captured in the dictionary to the target object. This is achieved by storing votes to the bounding box centroid, obtained by manual initialization, and associated local object appearances in correspondence to each dictionary atom.
Online Tracking — Whose purpose is to track the object across the sequence using a generalized Hough voting strategy. We reconstruct image patches through the dictionary and we cast the votes associated to each atom employed for the reconstructions in order to obtain the updated bounding box centroid position.

Our approach combines the advantages of both sparse coding and Hough voting-based strategies to reliably track unconstrained objects. Instead of using dictionaries containing object templates and relying on the reconstruction error to score candidate object positions as in [11, 181, 182, 175], we learn a generic, fixed, over-complete dictionary from small patches collected from images unrelated to the test sequences. Such resulting *universal* dictionary, which is estimated once and for all, is capable of reconstructing portions of the target object using a sparse combination of visual words selected with the awareness of the large range of appearances that can be found in real-world situations, as supported by the findings of [159].

A crucial step is the initialization, where the content of the manually placed bounding box is reconstructed patch-wise through the atoms of the dictionary and the notion of target shape and appearance is acquired by storing votes associated to each atom. The votes are stored as displacement vectors between the patches sampling positions and the center of the bounding box, while the appearances are represented by the reconstructions obtained through the dictionary. Due to the fact that the object is modeled locally by means of small patches, our approach is able to cope well with the presence of occlusion, noise, blur, sudden local and global illumination changes (all patches are individually normalized to account for illumination changes and are local) and background clutter. Furthermore, our approach can adapt to appearance changes of the tracked object by means of a specific update procedure of the votes and appearances associated to the dictionary atoms.

The intuition is that, as long as the appearances of the target do not radically change, its parts are always reconstructed using the same set of atoms. Therefore, the bounding box position in each frame can be retrieved using the proposed voting strategy. To cope with object appearance changes, the votes and appearances are updated in each frame to achieve robustness, while, conversely, the atoms of the dictionary are never modified.

Sparse coding

We make use of the theory shown in Section 2.2.4. In particular, the equation

$$\min_{\alpha} \frac{1}{2} \|\mathbf{D}\boldsymbol{\alpha} - \mathbf{y}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1$$

will be used in this work to learn the dictionary D and reconstruct the signal y. Here, to facilitate optimization and the enable the use of a fast implementation

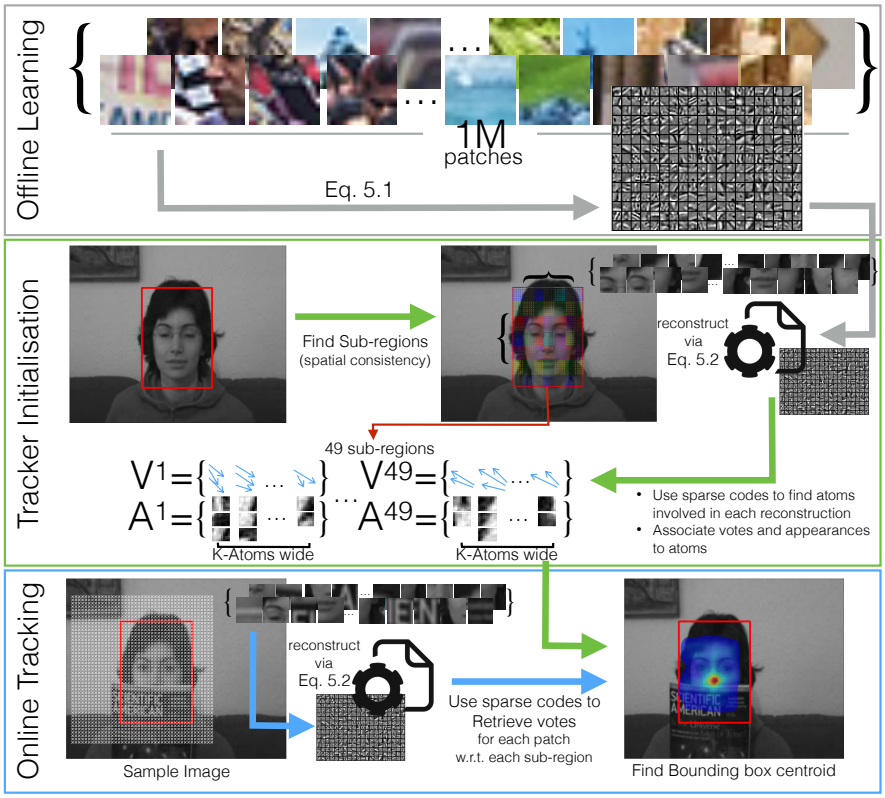

Figure 5.2: During offline learning we obtain a generic dictionary from image patches (Sec. 5.3.1). The initialization aims at collecting object-specific information in the form of votes and local appearances (Sec. 5.3.1). Online tracking is implemented using a voting strategy to retrieve the centroid of the bounding box (Sec. 5.3.1).

of the Lasso algorithm [107], the regularization term $\lambda \|\boldsymbol{\alpha}\|_1$ makes use of the l_1 norm as an approximation of the more principled l_0 norm.

Offline Dictionary Learning

Recent approaches demonstrated the capabilities of sparse coding to perform tasks such as denoising, texture synthesis, compression and audio processing [107]. In these approaches, a dictionary of non-orthogonal basis functions is employed to obtain sparse reconstructions of the input signals. We propose to reconstruct parts of the image using a limited number of basis patches, the atoms of the dictionary, which capture phenomena underlying real-world appearances. In our intuition we can retrieve sparse codes that are discriminative of the object of interest by deploying a dictionary capable of reconstructing a large range of different image patches. In contrast to previous methods based on l_1-sparse coding, we do not try to explain parts of the image using templates depicting the object of interest [182, 11, 181, 109], neither we employ the recon-

structions fidelity, which are potentially misleading, to score candidate object positions. In our approach, we employ a dictionary that can approximately reconstruct every possible image patch and which possesses knowledge about recurrent intensity patterns as seen in the training set. As a result, we encode the object of interest through combinations of dictionary atoms, each of which encodes the causes underlying intensity patterns occurring in real scenes [159]. This is possible because our dictionary is trained with an amount of data that goes well beyond that which is available in the first frame of the sequence. During the first step of our algorithm (Fig. 5.2, top), we collect a large set $\mathbf{T} = \{\mathbf{t}_1, ..., \mathbf{t}_n\}$ of grayscale image patches from generic images downloaded from the Internet and we learn a dictionary $\mathbf{D} = \{\mathbf{d}_1, ..., \mathbf{d}_k\}$ containing k atoms by optimizing the following problem with respect to $\mathbf{D}$ and $\boldsymbol{\alpha}_i$:

$$\arg\min_{\mathbf{D}} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} ||\mathbf{t}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 + \lambda ||\boldsymbol{\alpha}_i||_1. \tag{5.1}$$

We aim to minimize the sum of squared differences (SSD) between the patches contained in the data-set $\mathbf{T}$ and their sparse reconstructions obtained as a linear combination of the columns of $\mathbf{D}$ through the coefficients $\boldsymbol{\alpha}_i \in \mathbb{R}^k$. The strength of the sparsity constraint can be controlled through the parameter λ.

Tracker Initialization

Using the object bounding box provided in the first frame of every sequence, we initialize the method by capturing the shape and appearance of the object of interest: we rely, as previously stated, on a set of votes pointing to the bounding box centroid $\mathbf{c} = (c_x, c_y)$ together with a representation of the appearances of the region where each vote originated from.

Specifically, the initial bounding box is subdivided into $M \times N$ sub-regions $R_1, ..., R_{M \times N}$. Each region stores votes a separate list of votes and appearances of the regions of origin. This is necessary since patches from different sub-regions may be reconstructed using the same set of dictionary atoms, and if there would be just one common list of votes for the whole object, many unnecessary votes might be cast. By considering only small sub-regions within the object, we ensure votes that are always pointing in approximately the right direction. That is, they never induce a violation of the initial spatial configuration of the object's sub-regions during tracking. The absolute ordering of the sub-regions does not change during tracking. It is also worth pointing out that, in comparison with the case where no sub-regions are defined, a smaller amount of votes is stored in correspondence of each atom in every sub-region, leading to a reduction of processing time during tracking. The subdivision of each template into sub-regions is graphically depicted in Fig. 5.2, middle.

For each of the sub-regions we densely extract image patches $\mathbf{p}_1, ..., \mathbf{p}_s$ at locations $\mathbf{x}_1, ..., \mathbf{x}_s$ having the same dimensionality as the atoms in $\mathbf{D}$. Each patch $\mathbf{p}_i$ is reconstructed through $\mathbf{D}$ by solving the l_1-sparse optimization

problem

$$\arg\min_{\boldsymbol{\alpha}_i} \frac{1}{2}||\mathbf{p}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 + \lambda||\boldsymbol{\alpha}_i||_1 \tag{5.2}$$

yielding the sparse coefficients $\boldsymbol{\alpha}_i$, and the reconstructions $\hat{\mathbf{p}}_i = \mathbf{D}\boldsymbol{\alpha}_i$. Using these sparse coefficients $\boldsymbol{\alpha}_i$ we identify the indices of the atoms that contributed to the reconstruction of $\mathbf{p}_i$. Supposing that the i-th patch $\mathbf{p}_i$ belongs to the j-th sub-region and that it required the contribution of the k-th atom during its reconstruction, the vote $\mathbf{v}_i = \mathbf{c} - \mathbf{x}_i$ and the appearance $\hat{\mathbf{p}}_i$ are respectively added to the sets $\mathbf{V}_k^j$ and $\mathbf{A}_k^j$ (Fig. 5.2, middle). Importantly, storing sparse reconstructions $\hat{\mathbf{p}}_i$ as robust representations of region appearances is advantageous since it allows to reduce the effects of noise, and it implicitly encodes the configuration of the sparse coefficient vector $\boldsymbol{\alpha}$ characteristic of the patches used for initialization.

Our l_1-sparse optimization of Eq. 5.1 is almost instantaneous due to the fact that the dictionary atoms $\mathbf{d}_i$, as well as the signals $\mathbf{p}_i$, consist of very small patches with low dimensionality, thus yielding an average computational cost for the initialization step of typically just a few milliseconds.

Online Tracking

We track the object across the sequence by retrieving the position of the bounding box centroid in each frame through the voting strategy. We extract image patches $\mathbf{p}_1^j, ..., \mathbf{p}_N^j$ from the area surrounding the last known position of each sub-region R_j ($50px^2$ in our experiments) and we reconstruct them using the dictionary $\mathbf{D}$ solving the l_1-sparse optimization as stated in Eq. 5.2. The obtained sparse codes $\boldsymbol{\alpha}_i^j$ and the reconstructions $\hat{\mathbf{p}}_i^j = \mathbf{D}\boldsymbol{\alpha}_i^j$ are respectively employed to identify the atoms involved in each reconstruction and to obtain weights for the votes $\mathbf{V}_k^j$ by comparison with the learned appearances stored in $\mathbf{A}_i^j$ (Fig. 5.2, bottom). Let us suppose the i-th image patch belongs to the search area of the j-th sub-region and that is reconstructed through the k-th dictionary atom: we cast all the votes $\mathbf{v}_l^{(k,j)}$ stored in $\mathbf{V}_j^k$ after weighting their contributions with the weights $w_l^{(k,j)}$ obtained as the reciprocal of the SSD between the appearances $\mathbf{a}_l^{(k,j)}$ and the reconstruction $\hat{\mathbf{p}}_i^j$:

$$w_l^{(k,j)} = \frac{1}{(\mathbf{a}_l^{(k,j)} - \hat{\mathbf{p}}_l^j)^\top (\mathbf{a}_l^{(k,j)} - \hat{\mathbf{p}}_l^j)}. \tag{5.3}$$

The weighted votes contribute to a vote map. The bounding box centre is found by identifying the location of the highest peak in the vote map after it is smoothed by convolving it with a small Gaussian kernel.

Since the different search areas often overlap, an efficient implementation of the reconstruction can be achieved by solving Eq. 5.2 only once for all the patches in the global search area, regardless of the sub-regions they belong to. After the sparse codes are retrieved, they are interpreted using the information stored in the data structures of the specific sub-regions.

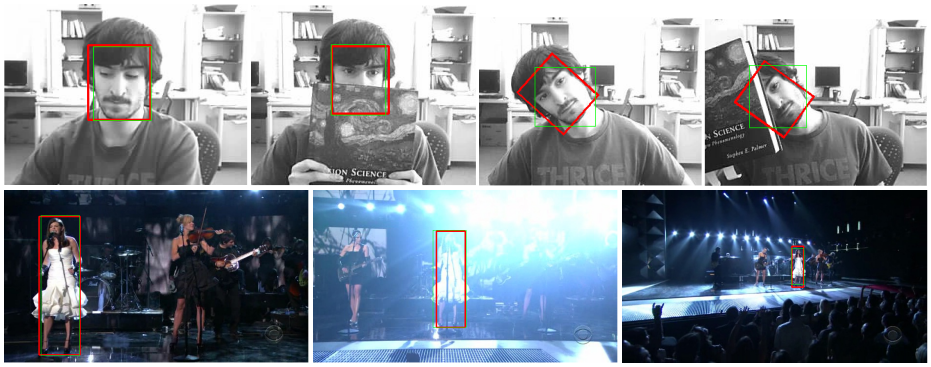

Figure 5.3: Our method, whose output is depicted using a red bounding box, is able to cope with large rotations and scale changes. Note that the manual annotation provided in the benchmark data-set [172], depicted in green, does not take into account rotations.

Update strategy

Once the bounding box is estimated, we select the atom of the dictionary that was employed the most for reconstruction of the background area and we prune its votes and appearances from the data structures of every sub-region. On the other hand, all the samples contained inside the estimated bounding box serve to update the voting structures through a procedure similar to the one used during initialization. In this way, we aim to keep information about the object until the moment it becomes misleading. This happens when votes and appearances get coincidentally associated to background structures.

Handling scale changes and rotations

The votes and appearances employed in our method are not invariant to rotation and scale changes. When the object changes orientation or size, the votes do not accumulate in clear peaks anymore. To handle scale changes and rotations of the target object, we create different versions of the input frame which are rotated and re-scaled by fixed quantities. We decide for the rotation and scale for which we obtain the vote map yielding the maximum peak. The inverse of the estimated parameters are then added to the current state of the tracker.

Since performing an exhaustive search by considering a large range of rotations and scale changes is a computationally intensive task, we rely on the assumption, commonly used in tracking, that the position, scale and rotation of the object changes smoothly from one frame to the other. In this way, as shown in Fig. 5.3, we can deal with those changes by only considering a small range of rotations and scale factors.

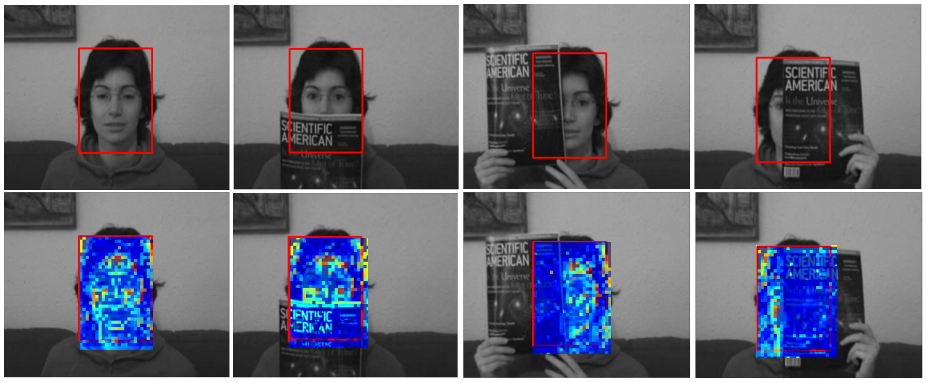

Figure 5.4: Backprojection of the Hough votes. Upper row: Output of our algorithm. Lower row: Votes having high weights (jet colormap) were generated only by patches belonging to the visible region of the object: the occlusion has a negligible effect on the vote map.

Figure 5.5: Robustness towards illumination changes is achieved by normalizing the patches extracted from the image. Even in sequences like 'David', where extreme illumination changes are present, our method performs correctly.

Robustness against occlusions

As previously stated, our method exhibits robustness to large amounts of occlusion. Since the reconstruction of the object is performed patch-wise and a few patches are already sufficient to cast a high number of votes with high confidence, we are able to localize the bounding box even when large portions of the target are not visible. In Fig. 5.4 we show the behaviour of our approach when the object undergoes occlusions. We re-project the votes that contributed to the estimation of the bounding box in each frame to the position of the patches that generated them and we observe that only visible parts of the object are able to effectively contribute to the estimation of the bounding box centroid.

Robustness against illumination changes

The patches extracted from the images both during initialization and tracking are normalized to zero mean and unit standard deviation to achieve illumination invariance. The same applies to the appearances stored in correspondence of the dictionary atoms. As briefly shown in Fig. 5.5, our method exhibits robustness against extreme illumination changes.

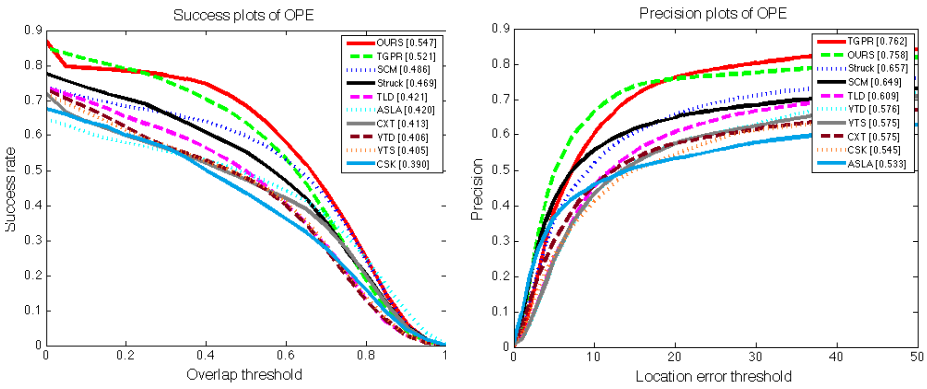

Figure 5.6: Results in terms of success and precision comparing our method with top performing algorithms on the 50 sequences (51 targets) of the CVPR13 Visual Tracking Benchmark[172]. Area under curve (AUC) is reported in brackets. All plots are color-coded according to performances. These images are obtained using the automatic scoring tool provided by the organizers of the challenge [172].

5.3.2 Experimental evaluation

To test our approach we employ the benchmark data-set published in [172] that consists of 50 annotated sequences (51 targets) including challenging situations such as illumination changes, deformations, occlusions, background clutter and motion blur. We compared with the most recent approaches having publicly available results on this benchmark, in particular 'L1APG' [11], 'MTT' [181], 'SCM' [182], 'Struck' [74], 'TGPR' [64] and all the others which have been evaluated in [172]. We follow the experimental protocols proposed in the benchmark [172] and evaluate our approach in terms of success and precision. All the sequences were converted to grayscale. The parameters of each algorithm are fixed for all the sequences and the bounding box used for initialization is provided in the first frame. Since the first frame of the sequence 'David' is very dark and unsuited for the initialization of many tracking algorithms, all the methods used for comparison were initialized at frame 300 while ours was initialized at frame 1. Although our approach yields better performance when initialized at frame 300, we want to demonstrate that we are able to track the object correctly even if the initialization frame is extremely dark as shown in Fig. 5.5. The average overlap and precision plots for all the experiments are depicted in Fig. 5.6. The performance of the trackers is expressed in terms of area under curve (AUC) and these values are enclosed in brackets in the plot of Fig.5.6. Qualitative results are shown in Fig. 5.1.

Parameters of the algorithm

The parameter λ, which controls the sparsity of both the dictionary and the sparse reconstructions is set to 0.1. The dictionary D consists of $k = 300$ 8×8 pixels atoms. During online tracking, candidate patches are collected using a

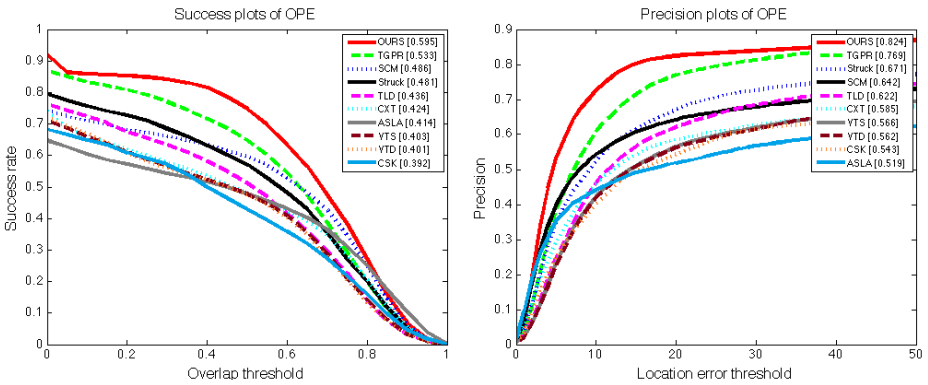

Figure 5.7: Results in terms of success and precision our method in comparison with top performing algorithms on 44 sequences (45 targets). Area under curve (AUC) reported in brackets. All plots are color coded according to performances. These images are obtained using the automatic scoring tool provided by the organizers of the challenge [172]

regular grid which has a 4 pixels spacing and which covers a 50 pixels wide area around the last known position of the bounding box. To handle scale changes and rotations, we transform each frame by considering each possible pair of scale and rotation from the set of possible rotation offsets, $\Delta r = \begin{bmatrix} -3 & 0 & 3 \end{bmatrix}$ degrees, and the set of possible scale offsets, $\Delta s = \begin{bmatrix} -0.03 & 0 & 0.03 \end{bmatrix}$. With these empirically selected parameters, our MATLAB implementation processes approximately 5 frames per second.

Results on selected sequences

From empirical observations we have noticed that the our tracking method tends to fail over low-resolution sequences depicting small target objects or objects that are hardly distinguishable from the surroundings (in grayscale). As a result, the algorithm performs unsatisfactorily in sequences such as 'Basketball', 'Bolt', 'Freeman3', 'Freeman 4', 'Girl' and 'CarDark'. We conclude that, failure over 'Freeman3', 'Freeman 4' and 'CarDark' sequences is due to the small size of the initial bounding box (with an area of resp. 156, 240, 667 px^2), which causes the number of votes stored during training to be low. Failure in the 'Basketball', 'Bolt', 'Girl' and 'CarDark' sequences are instead mostly determined by the additional presence of background clutter and lack of contrast between the objects and their surroundings in the grayscale images. Once these sequences are left out from the evaluation, we observe that the performance gap between our approach and the others remarkably increase to our favor, as witnessed by Fig. 5.7, which shows the results, in terms of success and precision, on 44 sequences (45 targets), where the 6 benchmark sequences having smallest resolution and target size were excluded.

5.3.3 Discussion

We presented a novel method for robust object tracking which uses dictionaries in a new fashion: generic, non object-specific information is learned from random images and it is used to reconstruct the object of interest patch-wise. The locality of these reconstructions coupled with the robustness of Hough voting allows the algorithm to perform in presence of large occlusions, illumination changes, motion blur and background clutter. Our approach outperforms the state of the art on every sequence apart from the ones that suffer from very low resolutions and depict very small, hard to distinguish, target objects. As a future work we plan to investigate a similar strategy using deep sparse auto-encoders instead of dictionaries.

Chapter 6

Conclusion and Outlook

Hough voting can be used in its generalized form to solve a wide range of problems in computer vision and medical image analysis. As shown in this thesis, voting methods find applications in fields such as detection, segmentation and tracking. Using voting it is possible to handle the limitation of most current machine learning approaches that often deliver imperfect results due to their generalization capabilities, the limited amount of available annotated training data and the characteristics of the images which can contain noise, artifacts and illumination changes. In other words, we have shown that we can accomplish through voting tasks that otherwise would have been impossible to solve by relying on more standard, end-to-end machine learning techniques. We can summarize here the findings of this thesis with respect to segmentation, detection and tracking.

In Chapter 3 we have shown how to segment one or more hardly visible structure in medical volumes with different voting based approaches. In particular, different methods were employed to extract meaningful features from image patches and fast nearest neighbors search allowed to retrieve votes and segmentation patches from a database assembled during training. This allowed for both more robust results and shape consistency of the final segmentation. Additionally it has been demonstrated that voting based approaches are much superior when compared to patch based classification or to the fully convolutional neural network approach presented in [116].

In Chapter 4 we have seen how voting based approaches can be used to perform RGBd object detection and pose estimation using the same principle introduced in Chapter 3 with some minor changes. The results achieved by the presented approach scale well when a great number of objects is present in the scene, and exhibit robustness to occlusions and other sources of performance decrease such as specular reflections etc.

In Chapter 5 the same finding has been demonstrated by employing dictionary learning and applying this technique in a truly unique fashion to the problem at hand. Differently than other recent approaches, and differently from the catheter detection work presented in Chapter 4, we learn the dictionary from small patches coming from unrelated images and we apply this knowledge to reconstruct patches from the objects that we want to track. Then

we associate the atoms used in these reconstructions with votes such that when similar atoms will be employed again in future reconstructions the votes will be cast and a robust detection result will be achieved.

6.1 Limitations

One of the main limitations of the presented approaches is their limited scalability. The works presented in Chapter 3 are all limited by the K nearest neighbors (K-NN) search algorithm efficiency and moreover require storage of segmentation patches, or at least an atlas of annotated volumes in order to work. Additionally, these approaches cannot be currently trained in a end-to-end fashion, due to the huge limitations that the voting strategy introduces computationally when we make it differentiable by replacing the K-NN selection strategy with "all-NN" where all the features extracted from the image are compared with the whole database and cast all the votes. Therefore, although this end-to-end training strategy is in theory possible, even simple experiment proving its effectiveness have not delivered interesting results so far.

Another limitation stems from the compute time required by the approaches presented in this thesis. Although some approaches are almost real time and are employed in a setting, such as medical volume segmentation, where wait times of a handful of seconds are acceptable our tracking algorithm and RGBd detection algorithm are still running at a reduced frame-rate.

6.2 Future Work

As a future work we would like to invesitgate how we can create more effective features to achieve better voting in a faster manner by selecting a smaller K in K-NN search. Moreover, a smart voting strategy that relies on voters that are not uniformely distributed in a grid would surely bring benefits to the runtime of the algorithm.

One of the main reasons to use Hough voting in segmentation is their capability to enforce an implicit shape prior of the anatomy at hand therefore improving the results by avoiding delivering anatomically implausible segmentations. As a future work we are currently looking at ways to include strong shape priors in methods that are more flexible such as the FCNN method employed in [116].

6.3 Epilogue

Extensive experimental evaluation and comparison of the approaches presented in this thesis with state of the art methods indicate that voting strategies are particularly effective in challenging problems and can improve the performances of current machine learning based methods. These findings motivate future research and the development of novel ideas which build on top of the approaches presented in this thesis.

Appendix A

Authored and Co-authored Publications

Authored:

1. Milletari, F., Yigitsoy, M., Navab, N.: Left ventricle segmentation in cardiac ultrasound using hough-forests with implicit shape and appearance priors pp. 49–56 (2014)
2. Milletari, F., Ahmadi, S.A., Kroll, C., Hennersperger, C., Tombari, F., Shah, A., Plate, A., Boetzel, K., Navab, N.: Robust segmentation of various anatomies in 3d ultrasound using hough forests and learned data representations. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015, pp. 111–118. Springer (2015)
3. Milletari, F., Ahmadi, S.A., Kroll, C., Plate, A., Rozanski, V., Maiostre, J., Levin, J., Dietrich, O., Ertl-Wagner, B., Bötzel, K., et al.: Hough-cnn: Deep learning for segmentation of deep brain regions in mri and ultrasound. Computer Vision and Image Understanding (2017)
4. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 3D Vision (3DV), 2016 Fourth International Conference on, pp. 565–571. IEEE (2016)
5. Milletari, F., Navab, N., Fallavollita, P.: Automatic detection of multiple and overlapping ep catheters in fluoroscopic sequences. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 371–379. Springer (2013)
6. Milletari, F., Belagiannis, V., Navab, N., Fallavollita, P.: Fully automatic catheter localization in c-arm images using â„“1-sparse coding. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 570–577. Springer (2014)
7. Milletari, F., Kehl, W., Tombari, F., Ilic, S., Ahmadi, S.A., Navab, N.: Universal hough dictionaries for object tracking. In: BMVC, pp. 122–1 (2015)

Co-authored:

1. Kehl, W., Milletari, F., Tombari, F., Ilic, S., Navab, N.: Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation. In: European Conference on Computer Vision, pp. 205–220. Springer (2016)

2. Zettinig, O., Shah, A., Hennersperger, C., Eiber, M., Kroll, C., Kübler, H., Maurer, T., Milletarì, F., Rackerseder, J., zu Berge, C.S., et al.: Multimodal image-guided prostate fusion biopsy based on automatic deformable registration. International journal of computer assisted radiology and surgery **10**(12), 1997–2007 (2015)

3. Bernard, O., Bosch, J.G., Heyde, B., Alessandrini, M., Barbosa, D., Camarasu-Pop, S., Cervenansky, F., Valette, S., Mirea, O., Bernier, M., et al.: Standardized evaluation system for left ventricular segmentation algorithms in 3d echocardiography. IEEE transactions on medical imaging **35**(4), 967–977 (2016)

4. Ahmadi, S.A., Milletari, F., Navab, N., Schuberth, M., Plate, A., Bötzel, K.: 3d transcranial ultrasound as a novel intra-operative imaging technique for dbs surgery: a feasibility study. International journal of computer assisted radiology and surgery **10**(6), 891–900 (2015)

5. Bortsova, G., Sterr, M., Wang, L., Milletari, F., Navab, N., Böttcher, A., Lickert, H., Theis, F., Peng, T.: Mitosis detection in intestinal crypt images with hough forest and conditional random fields. In: International Workshop on Machine Learning in Medical Imaging, pp. 287–295. Springer (2016)

6. Kroll, C., Milletari, F., Navab, N., Ahmadi, S.A.: Coupling convolutional neural networks and hough voting for robust segmentation of ultrasound volumes. In: German Conference on Pattern Recognition, pp. 439–450. Springer (2016)

7. Ahmadi, S.A., Plate, A., Schuberth, M., Milletari, F., Navab, N., Bötzel, K.: P116. dbs electrode imaging using 3d transcranial ultrasound–a feasibility study with first quantitative results. Clinical Neurophysiology **126**(8), e106–e107 (2015)

8. Riva, M., Hennersperger, C., Milletari, F., Katouzian, A., Pessina, F., Gutierrez-Becker, B., Castellano, A., Navab, N., Bello, L.: 3d intra-operative ultrasound and mr image guidance: pursuing an ultrasound-based management of brainshift to enhance neuronavigation. International Journal of Computer Assisted Radiology and Surgery pp. 1–15 (2017)

9. Baur, C., Milletari, F., Belagiannis, V., Navab, N., Fallavollita, P.: Automatic 3d reconstruction of electrophysiology catheters from two-view monoplane c-arm image sequences. International journal of computer assisted radiology and surgery **11**(7), 1319–1328 (2016)

Bibliography

[1] Ahmadi, S.A., Baust, M., Karamalis, A., Plate, A., Boetzel, K., Klein, T., Navab, N.: Midbrain segmentation in transcranial 3D ultrasound for Parkinson diagnosis. Med Image Comput Comput Assist Interv **14**(Pt 3), 362–369 (2011)

[2] Ahmadi, S.A., Milletari, F., Navab, N., Schuberth, M., Plate, A., Bötzel, K.: 3d transcranial ultrasound as a novel intra-operative imaging technique for dbs surgery: a feasibility study. International journal of computer assisted radiology and surgery **10**(6), 891–900 (2015)

[3] Ahmadi, S.A., Plate, A., Schuberth, M., Milletari, F., Navab, N., Bötzel, K.: P116. dbs electrode imaging using 3d transcranial ultrasound–a feasibility study with first quantitative results. Clinical Neurophysiology **126**(8), e106–e107 (2015)

[4] Aizerman, M.: Theoretical foundations of the potential function method in pattern recognition learning. Automation and remote control **25**, 821–837 (1964)

[5] Aldoma, A., Tombari, F., Di Stefano, L., Vincze, M.: A global hypothesis verification framework for 3d object recognition in clutter. IEEE transactions on pattern analysis and machine intelligence **38**(7), 1383–1396 (2016)

[6] Aldoma, A., Tombari, F., Prankl, J., Richtsfeld, A., Di Stefano, L., Vincze, M.: Multimodal cue integration through hypotheses verification for rgb-d object recognition and 6dof pose estimation. In: Robotics and Automation (ICRA), 2013 IEEE International Conference on, pp. 2104–2111. IEEE (2013)

[7] Avants, B.B., Yushkevich, P., Pluta, J., Minkoff, D., Korczykowski, M., Detre, J., Gee, J.C.: The optimal template effect in hippocampus studies of diseased populations. NeuroImage **49**(3), 2457 – 2466 (2010)

[8] Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 983–990. IEEE (2009)

[9] Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J., Freymann, J., Farahani, K., Davatzikos, C.: Segmentation labels and radiomic features for the pre-operative scans of the tcga-lgg collection. The Cancer Imaging Archive (2017)

[10] Ballard, D.H.: Generalizing the hough transform to detect arbitrary shapes. Pattern recognition **13**(2), 111–122 (1981)

[11] Bao, C., Wu, Y., Ling, H., Ji, H.: Real time robust l1 tracker using accelerated proximal gradient approach. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 1830–1837. IEEE (2012)

[12] Barbe, M.T., Dembek, T.A., Becker, J., Raethjen, J., Hartinger, M., Meister, I.G., Runge, M., Maarouf, M., Fink, G.R., Timmermann, L.: Individualized current-shaping reduces dbs-induced dysarthria in patients with essential tremor. Neurology **82**(7), 614–619 (2014)

[13] Barbosa, D., Friboulet, D., D?hooge, J., Bernard, O.: Fast tracking of the left ventricle using global anatomical affine optical flow and local recursive block matching. Proceedings of the MICCAI Challenge on Endocardial Three-dimensional Ultrasound Segmentation-CETUS pp. 17–24 (2014)

[14] Barbosa, D., Heyde, B., Dietenbeck, T., Houle, H., Friboulet, D., Bernard, O., D'hooge, J.: Quantification of left ventricular volume and global function using a fast automated segmentation tool: validation in a clinical setting. Int J Cardiovasc Imaging **29**(2), 309–316 (2013)

[15] Baur, C., Milletari, F., Belagiannis, V., Navab, N., Fallavollita, P.: Automatic 3d reconstruction of electrophysiology catheters from two-view monoplane c-arm image sequences. International journal of computer assisted radiology and surgery **11**(7), 1319–1328 (2016)

[16] Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. Computer vision–ECCV 2006 pp. 404–417 (2006)

[17] Belagiannis, V., Rupprecht, C., Carneiro, G., Navab, N.: Robust optimization for deep regression. preprint arXiv:1505.06606 (2015)

[18] Belagiannis, V., Schubert, F., Navab, N., Ilic, S.: Segmentation based particle filtering for real-time 2d object tracking. In: Computer Vision–ECCV 2012, pp. 842–855. Springer (2012)

[19] Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. Pattern Analysis and Machine Intelligence, IEEE Transactions on **35**(8), 1798–1828 (2013)

[20] Berg, D., Seppi, K., Behnke, S., Liepelt, I., Schweitzer, K., Stockner, H., Wollenweber, F., Gaenslen, A., Mahlknecht, P., Spiegel, J., Godau, J., Huber, H., Srulijes, K., Kiechl, S., Bentele, M., Gasperi, A., Schubert, T., Hiry, T., Probst, M., Schneider, V., Klenk, J., Sawires, M., Willeit, J.,

Maetzler, W., Fassbender, K., Gasser, T., Poewe, W.: Enlarged substantia nigra hyperechogenicity and risk for Parkinson disease: a 37-month 3-center study of 1847 older persons. Arch. Neurol. **68**(7), 932–937 (2011)

[21] Bernard, O., Bosch, J.G., Heyde, B., Alessandrini, M., Barbosa, D., Camarasu-Pop, S., Cervenansky, F., Valette, S., Mirea, O., Bernier, M., et al.: Standardized evaluation system for left ventricular segmentation algorithms in 3d echocardiography. IEEE transactions on medical imaging **35**(4), 967–977 (2016)

[22] Bernier, M., Jodoin, P., Lalande, A.: Automatized evaluation of the left ventricular ejection fraction from echocardiographic images using graph cut. Proceedings MICCAI Challenge on Echocardiographic Three-Dimensional Ultrasound Segmentation (CETUS), Boston, MIDAS Journal pp. 25–32 (2014)

[23] Bibby, C., Reid, I.: Robust real-time visual tracking using pixel-wise posteriors. In: Computer Vision–ECCV 2008, pp. 831–844. Springer (2008)

[24] Bortsova, G., Sterr, M., Wang, L., Milletari, F., Navab, N., Böttcher, A., Lickert, H., Theis, F., Peng, T.: Mitosis detection in intestinal crypt images with hough forest and conditional random fields. In: International Workshop on Machine Learning in Medical Imaging, pp. 287–295. Springer (2016)

[25] Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., Rother, C.: Learning 6d object pose estimation using 3d object coordinates. In: European conference on computer vision, pp. 536–551. Springer (2014)

[26] de Brébisson, A., Montana, G.: Deep neural networks for anatomical brain segmentation. CoRR **abs/1502.02445** (2015)

[27] Brost, A., Liao, R., Strobel, N., Hornegger, J.: Respiratory motion compensation by model-based catheter tracking during ep procedures. Medical Image Analysis **14**(5), 695–706 (2010)

[28] Calonder, M., Lepetit, V., Strecha, C., Fua, P.: Brief: Binary robust independent elementary features. Computer Vision–ECCV 2010 pp. 778–792 (2010)

[29] Casella, M., Pelargonio, G., Russo, A.D., Riva, S., Bartoletti, S., Santangeli, P., Scarà, A., Sanna, T., Proietti, R., Di Biase, L., et al.: Near-zero fluoroscopic exposure in supraventricular arrhythmia ablation using the ensite navx mapping system: personal experience and review of the literature. Journal of interventional cardiac electrophysiology **31**(2), 109–118 (2011)

[30] Cicek, O., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. arXiv preprint arXiv:1606.06650 (2016)

[31] Ciresan, D., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. In: Advances in Neural Information Processing Systems 25, pp. 2843–2851 (2012)

[32] Cireşan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Mitosis detection in breast cancer histology images with deep neural networks. In: International Conference on Medical Image Computing and Computer-assisted Intervention, pp. 411–418. Springer (2013)

[33] Coates, A., Ng, A.Y., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: International Conference on Artificial Intelligence and Statistics, pp. 215–223 (2011)

[34] Comstock, D.F., Troland, L.T.: The Nature of Matter and Electricity: An Ouline of Modern Views. D. Van Nostrand Company (1917)

[35] Cootes, T.F., Edwards, G.J., Taylor, C.J., et al.: Active appearance models. IEEE Transactions on pattern analysis and machine intelligence **23**(6), 681–685 (2001)

[36] Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models-their training and application. Computer vision and image understanding **61**(1), 38–59 (1995)

[37] Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3), 273–297 (1995)

[38] Cremers, D.: Dynamical statistical shape priors for level set-based tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(8), 1262–1273 (2006)

[39] Cremers, D., Rousson, M., Deriche, R.: A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. International journal of computer vision **72**(2), 195–215 (2007)

[40] Criminisi, A., Shotton, J.: Decision forests for computer vision and medical image analysis. Springer Science & Business Media (2013)

[41] Criminisi, A., Shotton, J., Konukoglu, E.: Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. Tech. rep., Microsoft Research Cambridge, UK (2011)

[42] Criminisi, A., Shotton, J., Robertson, D., Konukoglu, E.: Regression forests for efficient anatomy detection and localization in ct studies. In: B. Menze, G. Langs, Z. Tu, A. Criminisi (eds.) Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging, *Lecture Notes in Computer Science*, vol. 6533, pp. 106–117. Springer Berlin Heidelberg (2011). doi: 10.1007/978-3-642-18421-5_11

[43] Crum, W.R., Camara, O., Hill, D.L.G.: Generalized overlap measures for evaluation and validation in medical image analysis. IEEE Transactions on Medical Imaging **25**(11), 1451–1461 (2006)

[44] Csáji, B.C.: Approximation with artificial neural networks. Faculty of Sciences, Etvs Lornd University, Hungary **24**, 48 (2001)

[45] Cybenko, G.: Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals, and Systems (MCSS) **2**(4), 303–314 (1989)

[46] D'Albis, T., Haegelen, C., Essert, C., Fernández-Vidal, S., Lalys, F., Jannin, P.: Pydbs: an automated image processing workflow for deep brain stimulation surgery. International journal of computer assisted radiology and surgery **10**(2), 117–128 (2015)

[47] Deo, R., Albert, C.M.: Epidemiology and genetics of sudden cardiac death. Circulation **125**(4), 620–637 (2012)

[48] D'Haese, P.F., Pallavaram, S., Li, R., Remple, M.S., Kao, C., Neimat, J.S., Konrad, P.E., Dawant, B.M.: Cranialvault and its crave tools: A clinical computer assistance system for deep brain stimulation (dbs) therapy. Medical Image Analysis **16**(3), 744–753 (2012)

[49] Dietrich, O., Ahmadi, S.A., Levin, J., Maiostre, J., Plate, A., Giese, A., Bötzel, K., Reiser, M.F., Ertl-Wagner, B.: Quantitative susceptibility mapping with superfast dipole inversion: Influence of regularization parameters on the susceptibility of the substantia nigra and the red nucleus. In: Proc. Intl. Soc. Mag. Reson. Med., vol. 23, p. 3325 (2015)

[50] Dollár, P., Zitnick, C.L.: Structured forests for fast edge detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1841–1848 (2013)

[51] Domingos, J., Stebbing, R., Noble, J.: Endocardial segmentation using structured random forests in 3d echocardiography. Proceedings MICCAI Challenge on Echocardiographic Three-Dimensional Ultrasound Segmentation (CETUS), Boston, MIDAS Journal pp. 33–40 (2014)

[52] Donner, R., Menze, B.H., Bischof, H., Langs, G.: Global localization of 3d anatomical structures by pre-filtered hough forests and discrete optimization. Medical image analysis **17**(8), 1304–1314 (2013)

[53] D'Silva, A., Wright, M.: Advances in imaging for atrial fibrillation ablation. Radiology research and practice **2011** (2011)

[54] Duda, R.O., Hart, P.E.: Use of the hough transformation to detect lines and curves in pictures. Communications of the ACM **15**(1), 11–15 (1972)

[55] Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al.: Least angle regression. The Annals of Statistics **32**(2), 407–499 (2004)

[56] Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Advances in Neural Information Processing Systems, pp. 2366–2374 (2014)

[57] Fallavollita, P.: Acquiring multiview c-arm images to assist cardiac ablation procedures. Journal on Image and Video Processing pp. Jg., S.3 (2010)

[58] Fallavollita, P.: Is single-view fluoroscopy sufficient in guiding cardiac ablation procedures? Journal of Biomedical Imaging pp. Jg., S.1 (2010)

[59] Fallavollita, P., Savard, P., Sierra, G.: Fluoroscopic navigation to guide rf catheter ablation of cardiac arrhythmias. In: Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE, vol. 1, pp. 1929–1932. IEEE (2004)

[60] Farfade, S.S., Saberian, M.J., Li, L.: Multi-view face detection using deep convolutional neural networks. CoRR **abs/1502.02766** (2015)

[61] Franken, E., Rongen, P., van Almsick, M., ter Haar Romeny, B.: Detection of electrophysiology catheters in noisy fluoroscopy images. In: MICCAI 2006, pp. 25–32. Springer (2006)

[62] Gall, J., Yao, A., Razavi, N., Gool, L.V., Lempitsky, V.: Hough forests for object detection, tracking, and action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(11), 2188–2202 (2011)

[63] Gall, J., Yao, A., Razavi, N., Van Gool, L., Lempitsky, V.: Hough forests for object detection, tracking, and action recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33**(11), 2188–2202 (2011)

[64] Gao, J., Ling, H., Hu, W., Xing, J.: Transfer learning based visual tracking with gaussian processes regression. In: Computer Vision–ECCV 2014, pp. 188–203. Springer (2014)

[65] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Computer Vision and Pattern Recognition, IEEE Conf. on, pp. 580–587 (2014)

[66] Godec, M., Roth, P.M., Bischof, H.: Hough-based tracking of non-rigid objects. Computer Vision and Image Understanding **117**(10), 1245–1256 (2013)

[67] Grady, L.: Random walks for image segmentation. IEEE transactions on pattern analysis and machine intelligence **28**(11), 1768–1783 (2006)

[68] Gupta, S., Arbeláez, P., Girshick, R., Malik, J.: Aligning 3d models to rgb-d images of cluttered scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4731–4740 (2015)

[69] Gupta, S., Girshick, R., Arbeláez, P., Malik, J.: Learning rich features from rgb-d images for object detection and segmentation. In: European Conference on Computer Vision, pp. 345–360. Springer (2014)

[70] Gutiérrez-Becker, B., Cosío, F.A., Huerta, M.E.G., Benavides-Serralde, J.A., Camargo-Marín, L., Bañuelos, V.M.: Automatic segmentation of the fetal cerebellum on ultrasound volumes, using a 3d statistical shape model. Medical & biological engineering & computing **51**(9), 1021–1030 (2013)

[71] Hamarneh, G., Gustavsson, T.: Combining snakes and active shape models for segmenting the human left ventricle in echocardiographic images. In: Computers in Cardiology 2000, pp. 115–118. IEEE (2000)

[72] Hao, Q., Cai, R., Li, Z., Zhang, L., Pang, Y., Wu, F., Rui, Y.: Efficient 2d-to-3d correspondence filtering for scalable 3d object recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 899–906 (2013)

[73] Haralock, R.M., Shapiro, L.G.: Computer and robot vision. Addison-Wesley Longman Publishing Co., Inc. (1991)

[74] Hare, S., Saffari, A., Torr, P.H.: Struck: Structured output tracking with kernels. In: Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 263–270. IEEE (2011)

[75] Hart, P.E.: How the hough transform was invented [dsp history]. IEEE Signal Processing Magazine **26**(6) (2009)

[76] Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A.C., Bengio, Y., Pal, C., Jodoin, P., Larochelle, H.: Brain tumor segmentation with deep neural networks. CoRR **abs/1505.03540** (2015)

[77] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)

[78] He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034 (2015)

[79] Hennersperger, C., Karamalis, A., Navab, N.: Vascular 3d+ t freehand ultrasound using correlation of doppler and pulse-oximetry data. In: International Conference on Information Processing in Computer-Assisted Interventions, pp. 68–77. Springer (2014)

[80] Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., Lepetit, V.: Gradient response maps for real-time detection of textureless objects. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(5), 876–888 (2012)

[81] Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. preprint arXiv:1207.0580 (2012)

[82] Ho, T.K.: Random decision forests. In: Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on, vol. 1, pp. 278–282. IEEE (1995)

[83] Hodaň, T., Zabulis, X., Lourakis, M., Obdržálek, Š., Matas, J.: Detection and fine 3d pose estimation of texture-less objects in rgb-d images. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pp. 4421–4428. IEEE (2015)

[84] Ionasec, R., Voigt, I., Georgescu, B., Wang, Y., Houle, H., Vega-Higuera, F., Navab, N., Comaniciu, D.: Patient-specific modeling and quantification of the aortic and mitral valves from 4-d cardiac ct and tee. Medical Imaging, IEEE Transactions on **29**(9), 1636–1651 (2010). doi: 10.1109/TMI.2010.2048756

[85] Ionasec, R.I., Voigt, I., Georgescu, B., Wang, Y., Houle, H., Vega-Higuera, F., Navab, N., Comaniciu, D.: Patient-specific modeling and quantification of the aortic and mitral valves from 4-D cardiac CT and TEE. IEEE Trans Med Imaging **29**(9), 1636–1651 (2010)

[86] Ivancevich, N., Dahl, J., Light, E., Nicoletto, H., Seism, M., Laskowitz, D., Trahey, G., Smith, S.: 2b-2 phase aberration correction on a 3d ultrasound scanner using rf speckle from moving targets. In: Ultrasonics Symposium, 2006. IEEE, pp. 120–123 (2006)

[87] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)

[88] Kalal, Z., Matas, J., Mikolajczyk, K.: Pn learning: Bootstrapping binary classifiers by structural constraints. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 49–56. IEEE (2010)

[89] Kehl, W., Milletari, F., Tombari, F., Ilic, S., Navab, N.: Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation. In: European Conference on Computer Vision, pp. 205–220. Springer (2016)

[90] Kehl, W., Tombari, F., Navab, N., Ilic, S., Lepetit, V.: Hashmod: a hashing method for scalable 3d object detection. arXiv preprint arXiv:1607.06062 (2016)

[91] Keraudren, K., Oktay, O., Shi, W., Hajnal, J.V., Rueckert, D.: Endocardial 3d ultrasound segmentation using autocontext random forests. In: Proc. MICCAI CETUS, pp. 41–48 (2014)

[92] Klein, A., Andersson, J., Ardekani, B.A., Ashburner, J., Avants, B., Chiang, M.C., Christensen, G.E., Collins, D.L., Gee, J., Hellier, P., et al.: Evaluation of 14 nonlinear deformation algorithms applied to human brain mri registration. Neuroimage **46**(3), 786–802 (2009)

[93] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)

[94] Kroll, C., Milletari, F., Navab, N., Ahmadi, S.A.: Coupling convolutional neural networks and hough voting for robust segmentation of ultrasound volumes. In: German Conference on Pattern Recognition, pp. 439–450. Springer (2016)

[95] Lasso, A., Heffter, T., Rankin, A., Pinter, C., Ungi, T., Fichtinger, G.: Plus: open-source toolkit for ultrasound-guided intervention systems. IEEE Transactions on Biomedical Engineering **61**(10), 2527–2537 (2014)

[96] Le, Q.V.: Building high-level features using large scale unsupervised learning. In: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pp. 8595–8598. IEEE (2013)

[97] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)

[98] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)

[99] Li, H., Li, Y., Porikli, F.: Robust online visual tracking with a single convolutional neural network. In: Computer Vision–ACCV 2014, pp. 194–209. Springer (2015)

[100] Liang, X., Liu, S., Shen, X., Yang, J., Liu, L., Dong, J., Lin, L., Yan, S.: Deep human parsing with active template regression. CoRR **abs/1503.02391** (2015)

[101] Litjens, G., Toth, R., van de Ven, W., Hoeks, C., Kerkstra, S., van Ginneken, B., Vincent, G., Guillard, G., Birbeck, N., Zhang, J., et al.: Evaluation of prostate segmentation algorithms for MRI: the PROMISE12 challenge. Medical image analysis **18**(2), 359–373 (2014)

[102] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)

[103] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)

[104] Ma, Y., Gao, G., Gijsbers, G., Rinaldi, C.A., Gill, J., Razavi, R., Rhode, K.S.: Image-based automatic ablation point tagging system with motion correction for cardiac ablation procedures. In: Information Processing in Computer-Assisted Interventions, pp. 145–155. Springer (2011)

[105] Ma, Y., Gogin, N., Cathier, P., Housden, R.J., Gijsbers, G., Cooklin, M., O'Neill, M., Gill, J., Rinaldi, C.A., Razavi, R., et al.: Real-time x-ray fluoroscopy-based catheter detection and tracking for cardiac electrophysiology interventions. Medical physics **40**(7), 071,902 (2013)

[106] Ma, Y., King, A.P., Gogin, N., Rinaldi, C.A., Gill, J., Razavi, R., Rhode, K.S.: Real-time respiratory motion correction for cardiac electrophysiology procedures using image-based coronary sinus catheter tracking. In: MICCAI 2010, pp. 391–399. Springer (2010)

[107] Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 689–696. ACM (2009)

[108] McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics **5**(4), 115–133 (1943)

[109] Mei, X., Ling, H.: Robust visual tracking using l1 minimization. In: Computer Vision, 2009 IEEE 12th International Conference on, pp. 1436–1443. IEEE (2009)

[110] Mian, A., Bennamoun, M., Owens, R.: On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. International Journal of Computer Vision **89**(2), 348–361 (2010)

[111] Milletari, F., Ahmadi, S.A., Kroll, C., Hennersperger, C., Tombari, F., Shah, A., Plate, A., Boetzel, K., Navab, N.: Robust segmentation of various anatomies in 3d ultrasound using hough forests and learned data representations. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015, pp. 111–118. Springer (2015)

[112] Milletari, F., Ahmadi, S.A., Kroll, C., Hennersperger, C., Tombari, F., Shah, A., Plate, A., Boetzel, K., Navab, N.: Robust segmentation of various anatomies in 3d ultrasound using hough forests and learned data representations. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015, pp. 111–118. Springer (2015)

[113] Milletari, F., Ahmadi, S.A., Kroll, C., Plate, A., Rozanski, V., Maiostre, J., Levin, J., Dietrich, O., Ertl-Wagner, B., Bötzel, K., et al.: Hough-cnn: Deep learning for segmentation of deep brain regions in mri and ultrasound. Computer Vision and Image Understanding (2017)

[114] Milletari, F., Belagiannis, V., Navab, N., Fallavollita, P.: Fully automatic catheter localization in c-arm images using â„“1-sparse coding. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 570–577. Springer (2014)

[115] Milletari, F., Kehl, W., Tombari, F., Ilic, S., Ahmadi, S.A., Navab, N.: Universal hough dictionaries for object tracking. In: BMVC, pp. 122–1 (2015)

[116] Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 3D Vision (3DV), 2016 Fourth International Conference on, pp. 565–571. IEEE (2016)

[117] Milletari, F., Navab, N., Fallavollita, P.: Automatic detection of multiple and overlapping ep catheters in fluoroscopic sequences. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 371–379. Springer (2013)

[118] Milletari, F., Yigitsoy, M., Navab, N.: Left ventricle segmentation in cardiac ultrasound using hough-forests with implicit shape and appearance priors pp. 49–56 (2014)

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

[120] Montillo, A., Shotton, J., Winn, J., Iglesias, J.E., Metaxas, D., Criminisi, A.: Entangled decision forests and their application for semantic segmentation of ct images. In: IPMI (2011)

[121] Mor-Avi, V., Jenkins, C., Kühl, H.P., Nesser, H.J., Marwick, T., Franke, A., Ebner, C., Freed, B.H., Steringer-Mascherbauer, R., Pollard, H., et al.: Real-time 3-dimensional echocardiographic quantification of left ventricular volumes: multicenter study for validation with magnetic resonance imaging and investigation of sources of error. JACC: Cardiovascular Imaging **1**(4), 413–423 (2008)

[122] Muja, M., Lowe, D.G.: Scalable nearest neighbor algorithms for high dimensional data. Pattern Analysis and Machine Intelligence, IEEE Trans. on **36** (2014)

[123] Ng, A.: Sparse autoencoder. CS294A Lecture notes **72**(2011), 1–19 (2011)

[124] Ngo, T.A., Carneiro, G.: Left ventricle segmentation from cardiac mri combining level set methods with deep belief networks. In: Image Processing (ICIP), IEEE Intl. Conf. on, pp. 695–699 (2013)

[125] Noble, J.A., Boukerroui, D.: Ultrasound image segmentation: a survey. IEEE Trans Med Imaging **25**(8), 987–1010 (2006)

[126] Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1520–1528 (2015)

[127] Norris, R., White, H., Cross, D., Wild, C., Whitlock, R.: Prognosis after recovery from myocardial infarction: the relative importance of cardiac dilatation and coronary stenoses. European heart journal **13**(12), 1611–1618 (1992)

[128] Nouranian, S., Mahdavi, S.S., Spadinger, I., Morris, W.J., Salcudean, S.E., Abolmaesumi, P.: A multi-atlas-based segmentation framework for prostate brachytherapy. IEEE transactions on medical imaging **34**(4), 950–961 (2015)

[129] Oktay, O., Shi, W., Keraudren, K., Caballero, J., Rueckert, D., Hajnal, J.: Learning shape representations for multi-atlas endocardium segmentation in 3d echo images. Proceedings MICCAI Challenge on Echocardiographic Three-Dimensional Ultrasound Segmentation (CETUS), Boston, MIDAS Journal pp. 57–64 (2014)

[130] Oppenheim, A.v., Schafer, R., Stockham, T.: Nonlinear filtering of multiplied and convolved signals. IEEE transactions on audio and electroacoustics **16**(3), 437–466 (1968)

[131] Patenaude, B., Smith, S.M., Kennedy, D.N., Jenkinson, M.: A bayesian model of shape and appearance for subcortical brain segmentation. NeuroImage **56**(3), 907 – 922 (2011)

[132] Pauly, O., Ahmadi, S.A., Plate, A., Boetzel, K., Navab, N.: Detection of substantia nigra echogenicities in 3d transcranial ultrasound for early diagnosis of parkinson disease. In: N. Ayache, H. Delingette, P. Golland, K. Mori (eds.) Medical Image Computing and Computer-Assisted Intervention MICCAI 2012, *Lecture Notes in Computer Science*, vol. 7512, pp. 443–450. Springer Berlin Heidelberg (2012). doi: 10.1007/978-3-642-33454-2_55. URL `http://dx.doi.org/10.1007/978-3-642-33454-2_55`

[133] Pauly, O., Glocker, B., Criminisi, A., Mateus, D., Martinez-Moeller, A., Nekolla, S., Navab, N.: Fast multiple organ detection and localization in whole-body mr dixon sequences. In: Proc. MICCAI (2011)

[134] Payan, A., Montana, G.: Predicting alzheimer's disease: a neuroimaging study with 3d convolutional neural networks. CoRR **abs/1502.02506** (2015)

[135] Plate, A., Ahmadi, S.A., Pauly, O., Klein, T., Navab, N., Bötzel, K.: Threedimensional sonographic examination of the midbrain for computeraided diagnosis of movement disorders. Ultrasound Med Biol **38**(12), 2041–2050 (2012)

[136] Prasoon, A., Petersen, K., Igel, C., Lauze, F., Dam, E., Nielsen, M.: Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network. Med Image Comput Comput Assist Interv **16**(Pt 2), 246–253 (2013)

[137] Prince, S.J.: Computer vision: models, learning, and inference. Cambridge University Press (2012)

[138] Qiu, W., Rajchl, M., Guo, F., Sun, Y., Ukwatta, E., Fenster, A., Yuan, J.: 3D prostate TRUS segmentation using globally optimized volumepreserving prior. Med Image Comput Comput Assist Interv **17**(Pt 1), 796–803 (2014)

[139] Ranftl, R., Pock, T.: A deep variational model for image segmentation. In: Pattern Recognition, vol. 8753, pp. 107–118 (2014)

[140] Rematas, K., Leibe, B.: Efficient object detection and segmentation with a cascaded hough forest ism. In: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, pp. 966–973 (2011). doi: 10.1109/ICCVW.2011.6130356

[141] Riegler, G., Ferstl, D., Rüther, M., Bischof, H.: Hough networks for head pose estimation and facial feature localization. Journal of Computer Vision **101**(3), 437–458 (2013)

[142] Riemenschneider, H., Sternig, S., Donoser, M., Roth, P., Bischof, H.: Hough regions for joining instance localization and segmentation. In: A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (eds.) Computer Vision â€“ ECCV 2012, *Lecture Notes in Computer Science*, vol. 7574, pp. 258–271. Springer Berlin Heidelberg (2012). doi: 10.1007/978-3-642-33712-3_19

[143] Rios-Cabrera, R., Tuytelaars, T.: Discriminatively trained templates for 3d object detection: A real time scalable approach. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2048–2055 (2013)

[144] Riva, M., Hennersperger, C., Milletari, F., Katouzian, A., Pessina, F., Gutierrez-Becker, B., Castellano, A., Navab, N., Bello, L.: 3d intra-operative ultrasound and mr image guidance: pursuing an ultrasound-based management of brainshift to enhance neuronavigation. International Journal of Computer Assisted Radiology and Surgery pp. 1–15 (2017)

[145] Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015, pp. 234–241. Springer (2015)

[146] Roth, H., Lu, L., Seff, A., Cherry, K., Hoffman, J., Wang, S., Liu, J., Turkbey, E., Summers, R.: A new 2.5d representation for lymph node detection using random sets of deep convolutional neural network observations. In: Med Image Comput Comput Assist Interv, vol. 8673, pp. 520–527 (2014)

[147] Schapire, R.E.: The strength of weak learnability. Machine learning **5**(2), 197–227 (1990)

[148] Schenderlein, M., Stierlin, S., Manzke, R., Rasche, V., Dietmayer, K.: Catheter tracking in asynchronous biplane fluoroscopy images by 3d b-snakes. In: SPIE Medical Imaging, pp. 76,251U–76,251U. International Society for Optics and Photonics (2010)

[149] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv:1312.6229 (2013)

[150] Shotton, J., Girshick, R., Fitzgibbon, A., Sharp, T., Cook, M., Finocchio, M., Moore, R., Kohli, P., Criminisi, A., Kipman, A., et al.: Efficient human pose estimation from single depth images. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(12), 2821–2840 (2013)

[151] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

[152] Smistad, E., Lindseth, F.: Real-time tracking of the left ventricle in 3d ultrasound using kalman filter and mean value coordinates. Medical Image Segmentation for Improved Surgical Navigation p. 189 (2014)

[153] Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806 (2014)

[154] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Computer Vision and Pattern Recognition, IEEE Conf. on (2015)

[155] Szegedy, C., Toshev, A., Erhan, D.: Deep neural networks for object detection. In: Advances in Neural Information Processing Systems 26, pp. 2553–2561 (2013)

[156] Tang, J., Miller, S., Singh, A., Abbeel, P.: A textured object recognition pipeline for color and depth image data. In: Robotics and Automation (ICRA), 2012 IEEE International Conference on, pp. 3467–3474. IEEE (2012)

[157] Tejani, A., Tang, D., Kouskouridas, R., Kim, T.K.: Latent-class hough forests for 3d object detection and pose estimation. In: European Conference on Computer Vision, pp. 462–477. Springer (2014)

[158] Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Computer Vision, 1998. Sixth International Conference on, pp. 839–846. IEEE (1998)

[159] Tosic, I., Frossard, P.: Dictionary learning. Signal Processing Magazine, IEEE **28**(2), 27–38 (2011)

[160] Tsai, A., Yezzi, A., Wells, W., Tempany, C., Tucker, D., Fan, A., Grimson, W.E., Willsky, A.: A shape-based approach to the segmentation of medical imagery using level sets. IEEE transactions on medical imaging **22**(2), 137–154 (2003)

[161] Turaga, S.C., Murray, J.F., Jain, V., Roth, F., Helmstaedter, M., Briggman, K., Denk, W., Seung, H.S.: Convolutional networks can learn to generate affinity graphs for image segmentation. Neural Comput **22**(2), 511–538 (2010)

[162] Tustison, N.J., Avants, B.B., Cook, P.A., Zheng, Y., Egan, A., Yushkevich, P.A., Gee, J.C.: N4ITK: improved N3 bias correction. Medical Imaging, IEEE Transactions on **29**(6), 1310–1320 (2010)

[163] Van Stralen, M., Haak, A., Leung, K., Van Burken, G., Bosch, J.G.: Segmentation of multi-center 3d left ventricular echocardiograms by active appearance models. In: Proc. MICCAI CETUS, pp. 73–80 (2014)

[164] Vincent, G., Guillard, G., Bowes, M.: Fully automatic segmentation of the prostate using active appearance models. MICCAI Grand Challenge PROMISE 2012 (2012)

[165] Viola, P., Jones, M.J.: Robust real-time face detection. International journal of computer vision **57**(2), 137–154 (2004)

[166] Wachinger, C., Wein, W., Navab, N.: Registration strategies and similarity measures for three-dimensional ultrasound mosaicing. Academic radiology **15**(11), 1404–1415 (2008)

[167] Walter, U., Dressler, D., Probst, T., Wolters, A., Abu-Mugheisib, M., Wittstock, M., Benecke, R.: Transcranial brain sonography findings in discriminating between parkinsonism and idiopathic Parkinson disease. Arch. Neurol. **64**(11), 1635–1640 (2007)

[168] White, H.D., Norris, R.M., Brown, M.A., Brandt, P., Whitlock, R., Wild, C.J.: Left ventricular end-systolic volume as the major determinant of survival after recovery from myocardial infarction. Circulation **76**(1), 44–51 (1987)

[169] Wohlhart, P., Lepetit, V.: Learning descriptors for object recognition and 3d pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3109–3118 (2015)

[170] Wu, W., Chen, T., Barbu, A., Wang, P., Strobel, N., Zhou, S.K., Comaniciu, D.: Learning-based hypothesis fusion for robust catheter tracking in 2d x-ray fluoroscopy. In: CVPR, pp. 1097–1104. IEEE (2011)

[171] Wu, W., Chen, T., Strobel, N., Comaniciu, D.: Fast tracking of catheters in 2d fluoroscopic images using an integrated cpu-gpu framework. In: ISBI, pp. 1184–1187. IEEE (2012)

[172] Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2013)

[173] Xie, Y., Kong, X., Xing, F., Liu, F., Su, H., Yang, L.: Deep voting: A robust approach toward nucleus localization in microscopy images. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015, pp. 374–382. Springer (2015)

[174] Xie, Z., Singh, A., Uang, J., Narayan, K.S., Abbeel, P.: Multimodal blending for high-accuracy instance recognition. In: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pp. 2214–2221. IEEE (2013)

[175] Xing, J., Gao, J., Li, B., Hu, W., Yan, S.: Robust object tracking with online multi-lifespan dictionary learning. In: Computer Vision (ICCV), 2013 IEEE International Conference on, pp. 665–672. IEEE (2013)

[176] Yang, H., Shao, L., Zheng, F., Wang, L., Song, Z.: Recent advances and trends in visual tracking: A review. Neurocomputing **74**(18), 3823–3831 (2011)

[177] Yatziv, L., Chartouni, M., Datta, S., Sapiro, G.: Toward multiple catheters detection in fluoroscopic image guided interventions. Information Technology in Biomedicine, IEEE Transactions on **16**(4), 770–781 (2012)

[178] Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Computer vision–ECCV 2014, pp. 818–833. Springer (2014)

[179] Zettinig, O., Shah, A., Hennersperger, C., Eiber, M., Kroll, C., Kübler, H., Maurer, T., Milletarì, F., Rackerseder, J., zu Berge, C.S., et al.: Multimodal image-guided prostate fusion biopsy based on automatic deformable registration. International journal of computer assisted radiology and surgery **10**(12), 1997–2007 (2015)

[180] Zhang, L., Zhong, J., Lu, G.: Multimodality mr imaging findings of low-grade brain edema in hepatic encephalopathy. American Journal of Neuroradiology **34**(4), 707–715 (2013)

[181] Zhang, T., Ghanem, B., Liu, S., Ahuja, N.: Robust visual tracking via multi-task sparse learning. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 2042–2049. IEEE (2012)

[182] Zhong, W., Lu, H., Yang, M.H.: Robust object tracking via sparsity-based collaborative model. In: Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on, pp. 1838–1845. IEEE (2012)