Marginal Space Deep Learning: Efficient Architecture for Volumetric Image Parsing

Florin C. Ghesu, Edward Krubasik, Bogdan Georgescu, *Member, IEEE,* Vivek Singh,
Yefeng Zheng, *Senior Member, IEEE,* Joachim Hornegger, and Dorin Comaniciu, *Fellow, IEEE*

***Abstract*—Robust and fast solutions for anatomical object detection and segmentation support the entire clinical workflow from diagnosis, patient stratification, therapy planning, intervention and follow-up. Current state-of-the-art techniques for parsing volumetric medical image data are typically based on machine learning methods that exploit large annotated image databases. There are two main challenges that need to be addressed, these are the efficiency in processing large volumetric input images and the need for strong, representative image features. When the object of interest is parametrized in a high dimensional space, standard volume scanning techniques do not scale up to the enormous number of potential hypotheses and representative image features are subject to significant efforts of manual engineering. We propose a pipeline for object detection and segmentation in the context of volumetric image parsing, solving a two-step learning problem: anatomical pose estimation and boundary delineation. For this task we introduce Marginal Space Deep Learning (MSDL), a novel framework exploiting both the strengths of efficient object parametrization in hierarchical marginal spaces and the automated feature design of Deep Learning (DL) network architectures. Deep learning systems automatically identify, disentangle and learn explanatory attributes directly from low-level image data, however their application in the volumetric setting is limited by the very high complexity of the parametrization. More specifically 9 parameters are necessary to describe a restricted affine transformation in 3D (3 for each location, orientation, and scale) resulting in a prohibitive number of scanning hypotheses, in the order of billions for typical sampling. The mechanism of marginal space learning provides excellent run-time performance by learning classifiers in clustered, high-probability regions in spaces of gradually increasing dimensionality, for example starting from location only (3D) to location and orientation (6D) and full parameter space (9D). Given the structure localization, we estimate the 3D shape through non-rigid, DL-based boundary delineation in an Active Shape Model (ASM) framework. In our system we learn sparse adaptive data sampling patterns which replace manually engineered features by automatically capturing structure in the given data. This is also a type of model simplification, ensuring significant computational improvements and preventing overfitting. Experimental results are presented on detecting and segmenting the aortic valve in ultrasound using an extensive dataset of 2891 volumes from 869 patients, showing significant improvements of up to 45.2% over the current methods. To our knowledge, this is the first successful demonstration of the DL potential to detection and segmentation in full 3D data with parametrized representations.**

***Index Terms*—Deep learning, sparse representations, marginal space learning, three-dimensional (3D) object detection and segmentation, image parsing.**

I. Introduction

THE performance of machine learning algorithms depends on the underlying data representation and implicitly on the quality of the extracted features [1]. Designing strong and robust features that are able to compactly capture the information encoded in the given data is a particularly difficult task [2], [3], [4], [5]. In practice, this requires complex data preprocessing pipelines that do not generalize well between different image modalities or learning tasks. The reason for that is that most of these systems are manually engineered for specific applications and rely exclusively on human ingenuity to disentangle and understand prior information hidden in the data in order to design the required features [1], [6].

Specifically in the context of volumetric image parsing, machine learning is used to estimate the pose and nonrigid shape deformation of arbitrary 3D objects [5]. Here, the task of feature engineering becomes increasingly complex. A solution is required for efficient feature extraction, especially under challenging transformations such as arbitrary orientations, in order to support the efficient scanning of parameter spaces. In addition, these features need to be powerful and distinctive, regardless of the image modality and data complexity. For robust parameter estimation, scanning the parameter space exhaustively is not feasible, since in a volumetric setting the object pose is defined in a 9D parameter space. Such a task surpasses the capabilities of current consumer machines, creating also a need for a solution to effectively explore such high-dimensional spaces.

In this work we overcome these challenges by proposing a feature-learning-based framework to support the efficient 3D segmentation of arbitrary anatomical structures. For this we formulate a two-step approach using deep learning (DL) as a powerful solution for joint feature learning and task learning in each step: object localization and boundary estimation.

Manuscript received October 02, 2015; revised December 30, 2015; accepted February 26, 2016. *Asterisk indicates corresponding author.*

*F. Ghesu is with Medical Imaging Technologies, Siemens Healthcare, Princeton, NJ 08540 USA and with the Pattern Recognition Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg, 91058 Erlangen, Germany (e-mail: florin.c.ghesu@fau.de)

B. Georgescu, V. Singh, Y. Zheng and D. Comaniciu are with Medical Imaging Technologies, Siemens Healthcare, Princeton, NJ 08540 USA.

E. Krubasik is with the Robotics and Embedded Systems Department, Technische Universität München, 85748 München, Germany. He contributed to this work while he was with Medical Imaging Technologies, Siemens Healthcare.

J. Hornegger is with the Pattern Recognition Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg, 91058 Erlangen, Germany.

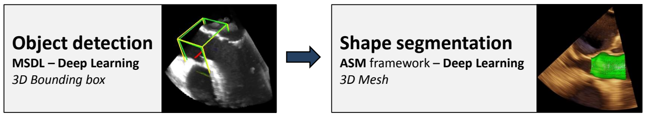

Fig. 1. Schematic overview of the complete proposed pipeline. The information about the object location is used to initialize a mean shape which in the second stage is deformed using learning to describe the true boundary.

To tackle the 3D object detection problem, we propose *Marginal Space Deep Learning* (MSDL), an efficient architecture that exploits the advantages of both deep learning and Marginal Space Learning (MSL) [5]. Marginal space learning reduces the estimation of the rigid transformation parameters to learning in parameter spaces of increasing dimensionality, focused on high probability regions. We propose a unified representation learning approach based on deep learning in each marginal space. First the location is estimated (3D space), then the location and orientation (6D space) and finally the complete transformation with the anisotropic scale (9D space). Since positives are usually clustered in dense regions of each space, the underlying sample set is very unbalanced. To account for this, we introduce a novel filtering cascade to balance the training data.

Given the object pose we can compute an initial estimate of the nonrigid shape. We then propose an deep-learning-based active shape model (ASM) [7] to guide the shape deformation (see Figure 1).

Focusing on the learning engine proposed in each of these two steps, representation learning through deep learning serves as a powerful solution against the limitations of handcrafted features [1], [5]. This type of system is based on a different learning paradigm, modeling the underlying task and the feature extraction as a joint automatic process, compared to traditional approaches which decouple the feature extraction task as an independent, complex, manual prerequisite. Hierarchical representations modeled by deep neural networks (DNN) [1] such as *Deep Convolutional Neural Networks* (CNN) [8], *Deep Belief Networks* (DBN) [9], *Restricted Boltzmann Machines* (RBM) [10] or *Stacked Denoising Autoencoders* (SDAE) [11] are very popular in this context, outperforming state-of-the-art solutions for a wide range of problems [9], [12], [13], [14].

However, anchored in the field of computer vision, the current applications of these architectures are focused on pixel(voxel)-wise classification in 2D or 2.5D data, with no generic extension supporting the parsing of large volumetric images. Capturing the complex appearance of 3D structures and ensuring the efficient scanning of high-dimensional parameter spaces are not straightforward, given the increased number of parameters to describe the pose and nonrigid deformation of an object. To account for this we propose novel *sparse adaptive deep neural networks* (SADNN) for learning parametrized representations from 3D medical image modalities and supporting the efficient scanning of large parameter spaces in the context of image parsing. In our system we learn sparse, adaptive sampling patterns which simplify the learning model, replacing the typical, manually engineered feature patterns [2], [3], [4], [5]. Through the permanent elimination of connections during training in the first layer of the network, the data sampling pattern is gradually structured around input voxels that are important for the learning task.

We validate our approach on the problem of detecting and segmenting the aortic heart valve in 3D ultrasound data. Several solutions for this problem have been presented in the literature: non-learning-based approaches for 3D echocardiographic images [15] and CT angiography [16], and a machine learning driven approach - the MSL framework [5] for 3D ultrasound [17] and 4D cardiac CT [18]. We selected this framework as a state-of-the-art reference in the evaluation, given the extensive experiments presented in previous works and excellent results achieved on large patient sets. In this sense we provide a comprehensive quantitative comparison between our framework and this reference, using a dataset of 2891 volumes from 869 patients. For completion, we also present an indirect comparison to [16], [19].

This paper represents an extended version of our initial work [20]. In summary, our contributions from [20] and presented here in greater detail, are the following:

- We present a novel method for injecting sparsity in deep neural networks to learn sparse, adaptive sampling patterns which enable the computational efficiency necessary for scanning large spaces. Note that this is different from typical weight-dropping methods. We learn sparse, adaptive structures in the data by dropping up to 90% of the input.
- Based on this method, we present MSDL, a novel framework combining the computational benefits of MSL with the potential of DL technology, to estimate the pose of arbitrary 3D anatomical structures.

Building upon these techniques, our additional contributions in this paper are the following:

- We propose the integration of MSDL with a novel, DL-based active shape model, enabling the automatic, nonrigid shape segmentation of arbitrary anatomical structures in the context of volumetric image parsing.
- We include a comparison of our proposed architecture with the state-of-the-art convolutional neural network. In this context we also present experiments highlighting the differences between these two architectures.
- We provide a comprehensive performance evaluation for the detection and segmentation of the aortic valve using an extensive dataset from 869 patients, with images acquired from different vendors.

This complete pipeline represents, to the best of our knowledge, the first DL-based approach in literature, that is focused towards parametrized detection and segmentation of arbitrary shapes in the context of volumetric image parsing.

The remaining paper is organized as follows. In Section II we review previous work on object localization and nonrigid shape estimation methods and also provide a motivation. In Section III we present our approach for generic 3D detection and segmentation using deep learning. Section IV shows the experiments we perform for validation. Finally, Section V concludes our paper.

II. Background and Motivation

Parsing volumetric medical image data is a challenging problem, approached only marginally in literature [5], [21]. It subsumes the robust detection, segmentation and recognition of objects contained in a volume, a particularly difficult task for arbitrary 3D anatomical structures considering the variance in location, the nonrigid nature of the shape as well as the differences in anatomy among different cases [5]. Many solutions focus mainly on the segmentation task, proposing methods for nonrigid boundary delineation based on Active Shape Models [7], [22], Active Appearance Models [23], Active Contour Evolution and Level Sets [24], [25], Markov Random Fields [26] and deformable models [19], [27], [28]. For the automatic parsing of volumetric data, in particular for the segmentation of arbitrary anatomical structures, a robust and efficient solution is required for localizing the object of interest.

A. Object Localization: Challenges

Given the complexity of nonrigid 3D shapes parametrized in high-dimensional spaces, fitting the shape model without any prior information extracted from the pose of the object, is not always feasible. As such, an essential step towards an accurate segmentation is the robust localization of structures (of the objects directly or of important anatomical landmarks). Initially introduced in the 2D context [3], [4], machine learning can be used for the efficient and robust localization of objects. In these approaches, object localization is formulated as a patch-wise classification problem. A parametric space is defined based on the parameters encoding the pose of the object. The space is then quantized to a large set of discrete hypotheses, which are used for learning. In the detection phase, the trained classifier is used to scan the parametric space and assign a score for each hypothesis, regarding the highest scoring hypothesis to be the detection result. The main advantage of such an approach is the robustness towards local optima which comes at a high computational cost associated with the space scanning. However, extending the logic to 3D is not straightforward since the number of hypotheses increases exponentially with respect to the dimensionality of the parameter space. This space becomes nine-dimensional when associated with a 3D restricted affine transformation, three parameters to define the position, three to define the orientation, and three to define the anisotropic scale of the object. Even with a very coarse discretization of $d = 10$ possible outcomes for each parameter, the number of hypotheses residing in that space will be as high as $d^9 = 1,000,000,000$, virtually impossible to evaluate on any current consumer machine. In consequence, a solution is needed to efficiently explore such high-dimensional spaces.

Considering the learning task itself in this high dimensional space, we emphasize the limitations of handcrafted features in capturing the variability of such complex data. For an accurate detection, the used features need to be powerful and robust to effectively represent the underlying phenomena. This property should hold regardless of the image modality, with no prior assumptions based on the appearance of the anatomy. Moreover, turning the focus on the feature extraction in the context of parametric space scanning, computational efficiency becomes critical. Features need to be efficiently computed also under challenging transformations such as arbitrary orientations or scales, without explicitly transforming the data. For example standard features such as local scale-invariant features [2], Haar wavelet features [4] or gradient based features [3] are not feasible in such a complex setup, lacking the required efficiency. On the other hand, steerable features as proposed by Zheng *et. al.* [5] can be efficiently evaluated under the assumed transformations, but are subject to the limitations of manual engineering, relying exclusively on human ingenuity, regardless of the underlying data. As such, there is a clear need for a mechanism to develop representative features which overcome the limitations of handcrafting methods and are fast to evaluate under any transformations.

B. Nonrigid Segmentation: Challenges

Even with an accurate object localization, the model fitting, i.e. the segmentation of the target structure, remains a challenging task given the nonrigid nature of the shape. Non-learning based boundary delineation methods represent a strong alternative (see for example [7], [15], [16], [19], [22], [23], [26], [28]) since they do not require large image databases for the model estimation and can also achieve accurate results both in 2D [29] and 3D [19]. However, they also suffer from a series of limitations in handling the frequent noise present in the data and in generalizing among different image modalities or anatomical structures. Machine learning is proposed as a solution to these challenges [5], [30]. In the 3D case, steerable features [5] have been proposed as a powerful and efficient handcrafted solution, supporting the learning-based shape segmentation using image evidence. Nonetheless, as argued in the context of object localization challenges, we can identify also here a need to address the limitations of handcrafted features.

III. Method

In this section we present a solution to all the aforementioned challenges, a novel feature-learning-based framework for parsing volumetric images split in a two-stage approach: anatomical object localization and nonrigid shape estimation. For the first task, we present *Marginal Space Deep Learning* (MSDL), a solution exploiting the computational benefits of Marginal Space Learning (MSL) [5] and the automated, self-learned feature design of Deep Learning (DL). For the segmentation task we propose a learning based Active Shape Model (ASM) using a deep-learning-based system to guide the shape deformation.

A. Deep Learning: An Overview

Deep Learning (DL) is a rapidly developing technology in the machine learning community addressing the limitations of handcrafted features by proposing an automated feature design learned directly from the data [12]. In the last years DL has shown a remarkable impact on a wide range of applications like speech recognition, natural language processing, transfer learning and in particular object recognition.

The breakthrough started in the computer vision community on the classification of natural images, where DL solutions [9], [12], [14] significantly surpassed the performance of the well-established state-of-the-art support vector machines. Further improvements were made through the introduction of the *dropout* regularization technique [13] or the *OverFeat* framework [31], an efficient multiscale, sliding-window approach for scanning. In the context of face recognition and classification the performance-boost of the *DeepFace* [32] platform over the state-of-the-art is by more the 27%, closely approaching human-level performance. These advancements also echoed in the medical image processing field, where transfer learning through DL enabled the application of this technology, given the overall, very limited availability of medical images [33]. For example, DL models pre-trained on natural images achieve accurate results for the localization of structures in fetal ultrasound data [34] or the identification of different types of pathologies in chest X-ray images [35]. Moreover, in the context of medical image segmentation, recent DL-based solutions for pixel-wise classification [29], [36] have significantly outperformed the state-of-the-art.

To the best of our knowledge no solutions are focused yet on the 3D context. This task is either approached through fusion of 2D features [37], sampling of random 2D planar observations [38] or through hypothesized extensions of readily available 2D methods [29] based on voxel-wise classification. In this paper we make a first step towards applying DL for efficient detection and segmentation in 3D with parametrized representations.

B. Deep Neural Networks

At the core of deep learning systems are deep neural networks (DNN) - powerful, automated feature-learning engines, built on hierarchies of data representations [8], organized as a series of inter-connected neural layers. From a functional point of view, the network is designed to emulate the functionality of the brain, building with every layer of neurons more abstract data representations, which are useful for better understanding the hidden structure and semantics of the input [1]. An essential step towards successfully training deep neural networks comes with the introduction of the *unsupervised layer-wise pre-training* algorithm [9]. Pre-trained layers can also be stacked to build deep, unsupervised models used to generate insightful representations of the data, for example Deep Autoencoders [39], Deep Belief Networks [9] or Deep Boltzmann Machines [40].

In our model we reduce both the detection and segmentation to a patch-wise classification task described by a set of m parametrized input patches $\vec{X}$ (i.e. observations) with a corresponding set of class assignments $\vec{y}$, specifying whether the sought anatomical structure is contained in the patch or not. In a representation learning approach such inputs are processed to abstract higher-level data representations using the inter-neural connections, defined as kernels under non-linear mappings. In this work, we focus on fully connected neural networks, meaning that the size of the filters is equal to the size of the underlying representations. From the perspective of one arbitrary neuron, this means that it is connected to all neurons in the previous layer (see Figure 2). Using this knowledge, we can define a deep fully connected DNN with the parameters $(\vec{w}, \vec{b})$, where $\vec{w} = (\vec{w}_1, \vec{w}_2, \cdots, \vec{w}_n)^\top$ represents the parameters of all n concatenated kernels over the layers of the network, i.e. the weighted connections between neurons, and $\vec{b}$ encodes the biases of the neurons. We mention that in this case n represents also the number of neurons in the network, in other words there is a one-to-one association between neuron and kernel (proof is intuitive). For computing the response or so called activation of one arbitrary neuron, a linear combination is computed between the weights of all incoming connections and the activations of all neurons from where the incoming connections originate. The bias of this neuron is added on this value, which is then transformed by a nonlinear mapping to obtain the activation value. In mathematical terms, from the perspective of the k-th neuron in the network, its activation value o_k is given by:

$$o_k = \delta\left(x_k^\top w_k + b_k\right), \tag{1}$$

where δ represents a non-linear activation function, w_k the weights of incoming connections, x_k the activations of the connected neurons from the previous layer and b_k the bias of the neuron. If the neuron is part of the first layer, x_k is given by the voxel values, i.e. the input data.

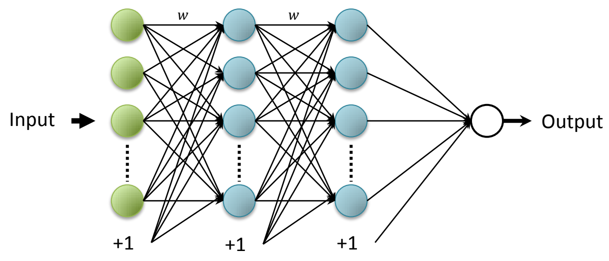

Fig. 2. Example of a fully connected neural network with 3 layers. Every neuron in a layer is connected to all neurons in the previous layer.

C. Training Deep Models

Turning the focus on the activation function δ, used to synthesize the input information, it can be shown that different functions relate to different learning problems. Possible functions are the identity function, rectified linear units (ReLU), the hyperbolic tangent or the sigmoid function. For our experiments we use the sigmoid function defined as $\delta(y) = 1/(1 + e^{-y})$, building through our network a multivariate logistic regression model for classification. Later in the experiment part we justify our choice and argue why the sigmoid activation function is more adequate in our setup.

Defining the network response function as $\mathcal{R}(\,\cdot\,; \vec{w}, \vec{b})$, we use it to approximate the probability density function over the class labels, given an input sample:

$$\mathcal{R}(x^{(i)}; \vec{w}, \vec{b}) \approx p(y^{(i)}|x^{(i)}; \vec{w}, \vec{b}), 1 \leq i \leq m. \tag{2}$$

Given the supervised setup and considering the independence of the input observations, we can use the Maximum Likelihood

Estimation (MLE) method to learn the network parameters in order to maximize the likelihood function:

$$\begin{aligned}\left(\hat{\vec{w}}, \hat{\vec{b}}\right) &= \arg\max_{\vec{w},\vec{b}} \mathcal{L}(\vec{w},\vec{b};\vec{X}) \\ &= \arg\max_{\vec{w},\vec{b}} \prod_{i=1}^{m} p(y^{(i)}|\vec{x}^{(i)};\vec{w},\vec{b}),\end{aligned} \tag{3}$$

where m represents the number of training samples. In other words we estimate the network parameters such that for every training sample $x^{(i)}$ the network predicts with high confidence its true class label $y^{(i)}$ ($1 \leq i \leq m$). This is equivalent to minimizing a cost function $\mathcal{C}(\cdot)$ quantifying how well the network prediction matches the expected output, i.e. the true label. We use the L_2 penalty function, reducing the maximization problem defined in Eq. 3 to the following minimization problem:

$$\left(\hat{\vec{w}}, \hat{\vec{b}}\right) = \arg\min_{\vec{w},\vec{b}} \left[\mathcal{C}(\vec{X};\vec{w},\vec{b}) = \|\mathcal{R}(\vec{X};\vec{w},\vec{b}) - \vec{y}\|_2^2\right]. \tag{4}$$

We solve this with the Stochastic Gradient Descent (SGD) method. Using a random set of samples $\tilde{X}$ from the training input, a feed-forward propagation is performed to compute the network response $\mathcal{R}(\tilde{X};\vec{w},\vec{b})$. Denoting $\vec{w}(t)$ and $\vec{b}(t)$ the network parameters in the t-th optimization step, they are updated according to the following rule:

$$\begin{aligned}\vec{w}(t+1) &= \vec{w}(t) - \eta\nabla_w C(\tilde{X};\vec{w}(t),\vec{b}(t)) \\ \vec{b}(t+1) &= \vec{b}(t) - \eta\nabla_b C(\tilde{X};\vec{w}(t),\vec{b}(t)),\end{aligned} \tag{5}$$

where ∇ denotes the gradient of the cost function with respect to the network parameters and η the magnitude of the update, i.e. the learning rate. To compute the gradient we use the backpropagation algorithm [41], which computes $\nabla_w C(\tilde{X};\vec{w}(t),\vec{b}(t))$ and $\nabla_b C(\tilde{X};\vec{w}(t),\vec{b}(t))$ layer by layer from the last layer to the first in a straightforward manner, given the chain structure of $\mathcal{R}$. We refer to $\tilde{X}$ as one batch of samples. One complete batch-wise iteration over the entire training data, with a parameter update at each step (see Eq. 5), is considered one training epoch. To train a powerful network based on this approach, many epochs are required (up to 300 in our experiments).

However, using this kind of technology straightforwardly to scan large parameter spaces for object localization and boundary delineation in the 3D context is not feasible. Ensuring the robustness of the network and also the training and testing efficiency represents a challenging task. Specifically in the volumetric setting, the parametrized input samples can become very large (boxes enclosing larger objects can reach sizes of 50^3 voxels). As such, a solution is required to handle the data-sampling/feature-computation task, especially under challenging transformations such as different orientations. Moreover, overfitting represents a common issue, given the large number of parameters in the low-level kernels.

D. Sparse Adaptive Deep Neural Networks

We address these challenges by proposing a novel method for layer sparsification inspired by the fundamental work of Lecun *et al.* [8], which conjectures that most neural networks are over-parametrized for their learning task. This especially holds for the non-convolutional fully connected deep networks we use, where kernels capture full representations in their field of view. As such we propose to apply *sparsity* as a means to simplify the neural network, aiming for computational efficiency and to avoid overfitting (see Figure 3). In other words, we permanently eliminate filter weights, while approximating as well as possible the original filter response. Viewing the system as a whole we aim to find a sparsity map $\vec{s}$ for the network weights $\vec{w}$, such that over T training rounds, the response residual ϵ given by:

$$\epsilon = \|\mathcal{R}(X;\vec{w},\vec{b}) - \mathcal{R}(X;\vec{w}_s,\vec{b}_s)\| \tag{6}$$

is minimal, where $\vec{b}_s$ denotes the biases of neurons in the sparse network and $\vec{w}_s$ denotes the learned sparse weights, determined by the sparsity map $\vec{s}$ with $s_i \in \{0,1\}, \forall i$. To achieve this, we propose a greedy, iterative learning process by gradually dropping neural connections which minimally impact the network response (see Algorithm 1).

The pre-training stage is designed to induce a certain degree of structure in the filters, prior to eliminating coefficients. Our sparsity injection method is applied over a predefined number of T training rounds. In each round t we greedily select a percentage of the absolute smallest active weights of the considered filters and permanently remove the corresponding neural connections from the network (set them to zero with the updated mask $\vec{s}^{(t)}$). We approximate the original response for any given filter i by preserving its L_1-norm (see Algorithm 1).

Algorithm 1 Learning algorithm with iterative threshold-enforced sparsity

```
Pre-training using all weights $\vec{w}^{(0)} \leftarrow \vec{w}$ (small # epochs)
Initialize sparsity map $\vec{s}^{(0)}$ with ones
$t \leftarrow 1$
for training round $t \leq T$ do
    for all filters $i$ with sparsity do
        $\vec{s}_i^{(t)} \leftarrow \vec{s}_i^{(t-1)}$ + remove smallest active weights
        $\vec{w}_i^{(t)} = \vec{w}_i^{(t-1)} \odot \vec{s}_i^{(t)}$
        Normalize active coeff. s.t. $\|\vec{w}_i^{(t)}\|_1 = \|\vec{w}_i^{(t-1)}\|_1$
    end for
    $\vec{b}^{(t)} \leftarrow \vec{b}^{(t-1)}$ (copy current biases)
    Train network on active weights (small # epochs)
    $t \leftarrow t + 1$
end for
Sparse kernels: $\vec{w}_s \leftarrow \vec{w}^{(T)}$
Biases: $\vec{b}_s \leftarrow \vec{b}^{(T)}$
```

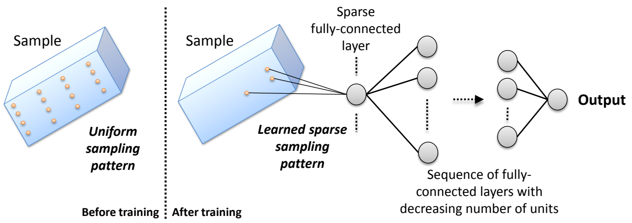
Sample
Uniform sampling pattern
Before training
After training
Sample
Sparse fully-connected layer
Learned sparse sampling pattern
Sequence of fully-connected layers with decreasing number of units
Output

Fig. 3. Visualization of the difference between uniform/handcrafted feature patterns and self-learned, sparse, adaptive patterns.

In the last step of each iteration the supervised training is continued on the remaining active connections, guiding the recovery of the neurons from the missing information by minimizing the original network loss function (see step 11 of Algorithm 1):

$$\left(\hat{\vec{w}}^{(t)}, \hat{\vec{b}}^{(t)}\right) = \arg \min_{\substack{\vec{w}:\, \vec{w}^{(t)} \\ \vec{b}:\, \vec{b}^{(t)}}} \mathcal{C}(\vec{X}; \vec{w}, \vec{b}), \tag{7}$$

where $\vec{w}^{(t)}$ and $\vec{b}^{(t)}$ (computed from the values in round $t-1$) are used as initial values in the optimization step.

As such, our system learns highly sparse features with adaptive structure. On the lowest level we learn adaptive, sparse data patterns which capture essential structures in the data, explicitly discarding input with minimal impact on the network response function $\mathcal{R}$ (see Figure 3). In our approach special focus is dedicated to how many active weights are set to zero in one training round. This number decreases with the training rounds, more specifically, in later stages of the training exponentially less filter weights are set to zero. This is intuitive, since the fewer weights in one particular kernel, the harder it is for that kernel to recover after a new sparsity enforcement step. The resulting adaptive sparse patterns overcome the limitations of manually pre-defined sampling patterns used in handcrafted features [2], [3], [4], [5], eliminating all together the need for feature engineering. We emphasize here that in our experiments the computed patterns can reach **sparsity levels of $\mathbf{90-95}\%$**, meaning that only $5-10\%$ of the weights of the low-level, fully-connected kernels can approximate the original network response $\mathcal{R}$. Moreover, the sparse network even outperforms the original model on unseen data since the sparsity acts as regularization and reduces the likelihood of overfitting during training. We call this type of networks: sparse adaptive deep neural networks (SADNN).

1) Comparison to Deep Convolutional Neural Networks: Among existing deep learning architectures, deep convolutional neural networks show great computational efficiency using concepts like kernel weight sharing and pooling (see [8], [13] for more details). In the following we provide a brief theoretical comparison of our proposed SADNN architecture to the state-of-the-art CNN architecture, showing that the latter cannot straightforwardly address the computational challenges associated with scanning large volumetric spaces. For this we choose a set of criteria to highlight the differences:

Sampling-layer: We emphasize that the input to the network is particularly large in the volumetric setting, for example a $50 \times 50 \times 50$ patch contains $125,000$ input voxels. While a SADNN indexes only a small fraction of the voxels using sparse, adaptive patterns with up to 95% sparsity, the CNN indexes every single input voxel multiple times, proportionally to the size of the convolution kernel. This brings a considerable speed-wise increase in terms of sampling effort for the SADNN (around 2 orders of magnitude, depending on the sparsity of the patterns and the size of the convolution kernels).

Dropout/Activation-sparsity: While we did not apply the dropout technique [42] or any other regularization methods used in state-of-the-art models, the SADNN incorporates the benefits of connection dropping and activation-sparsity by explicitly enforcing sparsity in the network and guiding the remaining neurons to learn to cope with the missing connections. The sparsity acts as regularization and reduces the likelihood of overfitting (see evaluation).

Translation Invariance: At the core of the CNN architecture is the translation invariance of the learned kernels. Using the concept of weight sharing, a kernel can detect features at any position in the input field. SADNNs do not have this advantage at patch level, however when scanning the input space each sparse pattern is invariant to the location of the patch, detecting features at any position in the input (same as in the case of CNN kernels).

Pooling: An advantage of the CNN architecture compared to SADNN is the use of pooling operations, a translation invariant operation reducing the dimensionality of the data representations.

Kernel Structure: In terms of filter structure, both sparse adaptive patterns and CNN kernels are based on local correlation between pixels/voxels. The difference is that the latter maintains the squared structure throughout the learning process whereas the adaptive patterns change the structure by explicitly dropping connections.

In the experiments section we show a performance comparison of these two types of architectures which reveals that in practice the SADNN is around 2 orders of magnitude faster. However in our future work we will continue to seek solutions to address these computational limitations and integrate the CNN in our framework either as a standalone classifier or as part of a hybrid CNN-SADNN architecture.

E. Marginal Space Deep Learning

As briefly introduced in the motivation in Section II, we model the pose of the sought object by using a bounding box, defined by 9 parameters: $\vec{T} = (t_x, t_y, t_z)$ for the translation, $\vec{R} = (\phi_x, \phi_y, \phi_z)$ for the orientation and $\vec{S} = (s_x, s_y, s_z)$ for the anisotropic scaling of the object. Given a volumetric image *I*, we propose to find the pose of the sought object by maximizing the posterior probability defined as:

$$\left(\hat{\vec{T}}, \hat{\vec{R}}, \hat{\vec{S}}\right) = \arg \max_{\vec{T}, \vec{R}, \vec{S}} p(\vec{T}, \vec{R}, \vec{S} | I). \tag{8}$$

We estimate this probability using the introduced sparse, adaptive deep neural network $\mathcal{R}(X; \vec{w}_s, \vec{b}_s)$, where $\vec{w}_s$ and $\vec{b}_s$ are the weight and bias vectors of the sparse network. Scanning over the entire space of possible transformations is however not feasible, given the prohibitive number of hypotheses which grows exponentially with respect to the dimensionality of the parameter space. This number reaches the order of billions even for a very coarse discretization for each parameter. In this context we propose *Marginal Space Deep Learning*, a solution exploiting the computational benefits of Marginal Space Learning (MSL) [5] and the automated feature design and efficiency of the SADNN architecture. Instead of scanning the entire 9D space exhaustively, the search is performed in clustered, high-probability regions of increasing dimensionality, starting in the position space, extending to the position-orientation space and finally to the full 9D space, including

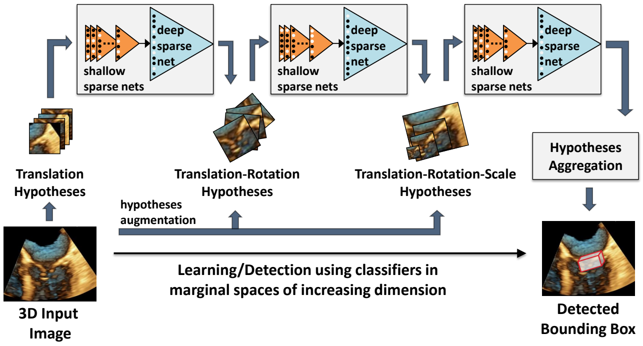

Fig. 4. Schematic visualization of the marginal space deep learning framework. The black/white dots encode the sparse sampling patterns.

also the anisotropic scaling information of the object. For this purpose we reformulate the original optimization problem presented in Eq. 8 by factorizing the posterior probability:

$$\begin{aligned}\left(\hat{\vec{T}}, \hat{\vec{R}}, \hat{\vec{S}}\right) &= \arg\max_{\vec{T},\vec{R},\vec{S}} p(\vec{T}|I)p(\vec{R}|\vec{T},I)p(\vec{S}|\vec{T},\vec{R},I) \\ &= \arg\max_{\vec{T},\vec{R},\vec{S}} p(\vec{T}|I)\frac{p(\vec{T},\vec{R}|I)}{p(\vec{T}|I)}\,\frac{p(\vec{T},\vec{R},\vec{S}|I)}{p(\vec{T},\vec{R}|I)},\end{aligned} \tag{9}$$

where the probabilities $p(\vec{T}|I)$, $p(\vec{T},\vec{R}|I)$ and $p(\vec{T},\vec{R},\vec{S}|I)$ are defined in the previously enumerated spaces of increasing dimensionality, also called *marginal spaces*. As such, we reduce our problem to learning classifiers in these marginal spaces and then scanning each space exhaustively to estimate in turn the position, orientation and scale of the object. This is possible by modeling each learning space as a set of sample hypotheses, positives and negatives used for training. For example, after learning the translation parameters in the translation space $\mathcal{U}_T(I)$, only the positive hypotheses with highest probability, clustered in a dense region are augmented with discretized orientation information, to build the joint translation-orientation space $\mathcal{U}_{TR}(I)$. The same principle applies when extending to the full 9D space $\mathcal{U}_{TRS}(I)$. In mathematical terms, the stage-wise optimization is defined by:

$$\begin{aligned}\left(\hat{\vec{T}}, \mathcal{U}_{TR}(I)\right) &\leftarrow \arg\max_{\vec{T}} \mathcal{R}\left(\mathcal{U}_T(I); \vec{w}_s, \vec{b}_s\right) \\ \left(\hat{\vec{T}}, \hat{\vec{R}}, \mathcal{U}_{TRS}(I)\right) &\leftarrow \arg\max_{\vec{T},\vec{R}} \mathcal{R}\left(\mathcal{U}_{TR}(I); \vec{w}_s, \vec{b}_s\right) \\ \left(\hat{\vec{T}}, \hat{\vec{R}}, \hat{\vec{S}}\right) &\leftarrow \arg\max_{\vec{T},\vec{R},\vec{S}} \mathcal{R}\left(\mathcal{U}_{TRS}(I); \vec{w}_s, \vec{b}_s\right),\end{aligned} \tag{10}$$

where $\mathcal{R}(\,\cdot\,; \vec{w}_s, \vec{b}_s)$ denotes the response of the sparse adaptive deep neural network, learned from the supervised training data $(\vec{X}, \vec{y})$. The same steps are performed also in the detection phase, using as input a single volumetric image. This type of approach brings a speed-up of 6 orders of magnitude compared to the exhaustive search in the 9D space (see proof in [5]), based on a dense discretization for each parameter. The pipeline is visualized in Figure 4.

1) Efficient Hypotheses Filtering: One important particularity of the learning task in each marginal space is the high class-imbalance. In parametric space this is explained by the limited range of possible positions, orientations or scales of the object of interest, assuming a predetermined patient pose or image acquisition protocol. In practice, this imbalance can reach ratios of $1:1000$ positive to negative samples, impacting both the training efficiency and stochastic sampling of the gradient during learning, resulting in a bias of the classifier towards the overrepresented negative class. While over/undersampling methods can be used in this context, the typical solution for this problem, as proposed by Ciresan *et al.* [43], is based on a re-weighting scheme of the network penalty to account for the imbalance ratio of the dataset. However, this type of approach does not address the computational challenges associated with processing such large amounts of training samples (in particular negative hypotheses), and also re-weighting the cost function can further worsen the vanishing gradient effect. Furthermore, most of these negative hypotheses are very easy to classify showing characteristics that are fundamentally different from the ones of positive examples. Using deep architectures with complex features to classify such simple hypotheses might lead to overfitting during training, affecting the performance of the classifier in difficult cases.

Algorithm 2 Negative sample filtering algorithm

```
$P$ - set of positive samples
$N$ - set of negative samples ($|P| << |N|$)
while $|N| \geq 1.5 \times |P|$ do
    Learn shallow SADNN using Algorithm 1
    $d \leftarrow$ largest decision boundary with FNR $= 0$
    Filter $N$ based on $d$ - eliminate true negatives
end while
```

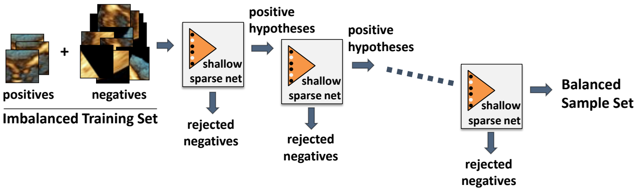

Fig. 5. Schematic visualization of the negative-sample filtering cascade used to balance the training set. The black/white dots encode the sparse sampling patterns of the shallow nets.

To account for all these issues, we propose a different approach based on a cascade of shallow neural networks used to efficiently and effectively filter the negative hypotheses. In each stage of the cascade we train a shallow, sparse neural network and adaptively tune its decision boundary to eliminate as many true negative hypotheses from the training set as possible. The remaining hypotheses, classified as positives, are propagated to the next stage of the cascade where the same filtering step is applied until the training set is balanced (see Figure 5 and Algorithm 2). Note that in each stage of the cascade, the underlying set of hypotheses used for training is unbalanced. We manually ensure that each batch of B samples used to estimate the gradient is balanced, by independently and randomly sampling $B/2$ positives respectively negatives from the training set.

Using this type of cascaded approach requiring only simple low-level features to filter the sample set represents an essential step towards a competitive run-time performance (both during training and testing). Reducing the size of the sample set processed by the main classifier in each stage from $|N|+|P|$ to approximately $3\times|P|$ improves the scanning performance by an additional 2 orders of magnitude (depending on the imbalance ratio).

F. Nonrigid Deformation Estimation

The automatic object localization using MSDL is followed in the second stage by the nonrigid deformation estimation of the object. The mean shape, computed based on the groundtruth data, is aligned to the estimated pose and then deformed to fit the object boundary. Based on the underlying image information, an active shape model can be used for the deformation process. However, the original approach based on energy deformation [7] is not feasible in a 3D setup where the image information and the boundary context are very complex. Learning based approaches have been introduced and been proven successful in solving this problem in 2D [30], [44] and 3D [5]. More specifically, a classifier labeled as boundary detector is trained at specific anatomical locations to detect the shape boundary based on local evidence, i.e. decide whether there is a boundary point at a given position, under a given orientation. To avoid the complexity associated with the image or feature pattern rotation, Zheng *et. al.* [5] propose steerable features combined with the *Probabilistic Boosting Tree* [45] to efficiently tackle the problem in the 3D context.

However, as argued in Section II, handcrafted features are based on strong prior assumptions and do not generalize well between different modalities or anatomical structures. We address this by proposing the SADNN (with negative filtering cascade) as a boundary classifier to automatically learn adaptive, sparse feature sampling patterns directly from low-level image data. Essentially, for a given control point of the warping mesh, we are dealing with the same problem as faced in the joint translation-orientation space, in the localization stage. Here we ask the question: Is there a boundary point at position $\vec{T}=(t_x,t_y,t_z)$ and orientation $\vec{R}=(\phi_x,\phi_y,\phi_z)$? The position $\vec{T}$ and orientation $\vec{R}$ is given by the current sample along the normal for the respective shape point. This is a classification problem and can be solved using the same mechanism used in the second stage (the joint translation-orientation learning space) of the MSDL framework. Training is performed by using positive samples on the current ground-truth boundary for each mesh point (aligned with the corresponding normal) and using negative samples at various distances from the boundary. The sparse adaptive patterns are essential in efficiently applying this classifier under arbitrary orientations, emphasizing relevant anatomical structures.

The iterative process is illustrated in Figure 6. The boundary estimation is followed by constraining the deformed shape to the space of shapes corresponding to the current object. We use statistical shape modeling for the constraint, where we estimate from the training set the linear shapes subspace through principal components analysis and online we project the current shape into this subspace using the learned linear projector. The process of boundary estimation and shape constraint enforcement are iteratively applied for a number of predetermined iterations or until there are no large deformations.

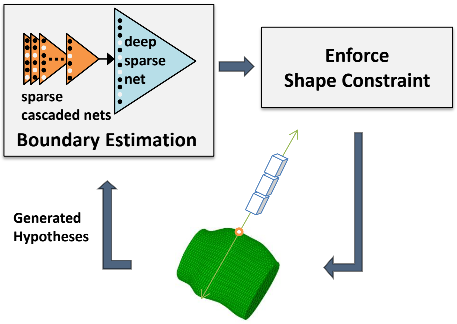

Fig. 6. Schematic visualization of the boundary deformation with SADNN. Starting from the current shape, the SADNN is aligned and applied along the normal for each point of the mesh, the boundary is deformed and projected under the current shape space. The process is iteratively repeated. The black/white dots encode the sparse patterns, learned in the cascaded shallow networks and deep boundary classifier.

IV. Experiments

In order to validate our approach we compare the performance against the state-of-the-art Marginal Space Learning solution [5] on detecting and segmenting the aortic valve (root) in 3D transesophageal echocardiogram (TEE) images. The aortic root connects the ascending aorta to the left ventricular outflow tract and is represented through a tubular grid (see Figure 9). This is a particularly challenging task, considering the high variation in anatomical appearance of the valve, the heart-motion and implicitly motion of the valve-leaflets and also the image quality limitations, i.e. noise in ultrasound images.

A. Dataset

The dataset used for evaluation stems from 869 patients and contains 2891 3D TEE volumes from different vendors. The size and resolution of the images vary from $100\times100\times50$ to $250\times250\times150$ voxels and 0.75 to 1 mm. In terms of preprocessing, we resampled all images to an isotropic resolution of 3 mm for the object localization task and mean shape initialization, and normalized the intensity of each volume to unit-range.

The validation setup is the same for both approaches and is based on a random split of the volumes in 2481 training and 410 testing volumes (about $84\%-16\%$). At patient level this results in 719 patients in the training set and 150 patients in the test set. Note that the split is made at patient-level, meaning that for any given patient, all corresponding frames are either in the training or the testing set. This ensures an unbiased comparison of the two approaches.

Ground truth was obtained through extensive manual annotation using a smart editing process performed by experts (see [17] for more details). The aortic root contour is modeled with a pseudo-parallel slice based method using cutting planes distributed equidistantly along the centerline of the structure. To account for the bending of the shape, the normal of each

TABLE I
COMPARISON OF THE PERFORMANCE OF THE STATE-OF-THE-ART MSL [5] AND THE PROPOSED MSDL FRAMEWORK FOR AORTIC VALVE DETECTION. THE MEASURES USED TO QUANTIFY THE QUALITY OF THE RESULTS W.R.T TO THE GROUND TRUTH DATA ARE THE ERROR OF THE POSITION OF THE BOX AND MEAN CORNER DISTANCE (BOTH MEASURED IN MILLIMETRES). THE SUPERIOR RESULTS ARE DISPLAYED IN BOLD.

	Position Error [mm]				**Corner Error [mm]**			
	Training Data		**Test Data**		**Training Data**		**Test Data**	
	MSL	MSDL	MSL	MSDL	MSL	MSDL	MSL	MSDL
Mean	3.12	**1.47**	3.34	**1.83**	5.42	**2.80**	6.16	**3.72**
Median	2.80	**1.27**	3.05	**1.58**	4.98	**2.58**	5.85	**3.34**
STD	1.91	**0.99**	1.85	**1.31**	2.47	**1.23**	2.31	**1.74**

plane is aligned to the tangent of the centerline of the tubular structure. The defined nonrigid shape is guided to delineate the boundary of the aortic valve and then used to extract the ground truth for the global location model. Centered at the barycenter of the aortic valve, the ground truth bounding box is scaled to capture the complete underlying anatomy and aligned according to the orientation of the commissural plane and interconnection point of the left and right leaflets. As such, the pose of the 3D bounding box is fully determined. For a complete definition of the aortic model please see the work of Ionasec *et al.* [17].

B. Aortic Valve Detection and Segmentation in 3D US

Detecting the aortic valve resumes to finding the 9 parameters describing its position, orientation and scale in the 3D space. The detected bounding box is then used to initialize a mean shape which in both approaches is guided using learning to fit the true boundary. Given the aforementioned validation setup, both MSDL and MSL frameworks are optimized to achieve maximum performance.

1) Model Selection and Meta-parameters: We used a grid search to estimate both the meta-parameters related to the hypotheses generation and augmentation in each marginal space, as well as the network dependent parameters, i.e. network architecture, number of hidden units, learning rate and sparsity levels of the sampling patterns. To highlight the benefits of using sparse sampling patterns we compared against the original non-sparse network (in our experiments we denote this framework as MSDL-non-sparse). Across all 3 marginal spaces we use the same SADNN architecture for the cascade and main classifier, i.e. cascade: 2 layers = 5832 (sparse) $\times$ 60 $\times$ 1 and main classifier: 4 layers = 5832 (sparse) $\times$ 150 $\times$ 80 $\times$ 50 $\times$ 1 hidden units. Note that in each stage we managed to balance the training set using at most 3 shallow networks in the cascade. Although trained successively, during testing the cascade networks and the main classifier define one large network with 7 to 13 layers (depending on how many shallow networks are necessary to achieve the balancing).

In all networks we used sigmoid activations, observing that ReLU does not perform as well in our experiments. We argue this might happen for two possible reasons. Firstly, having a cascaded structure of networks with early reject stages reduces the depth of each individual network to at most 4 hidden layers, thus decreasing the likelihood of vanishing gradients for which ReLU usually represents a powerful solution.

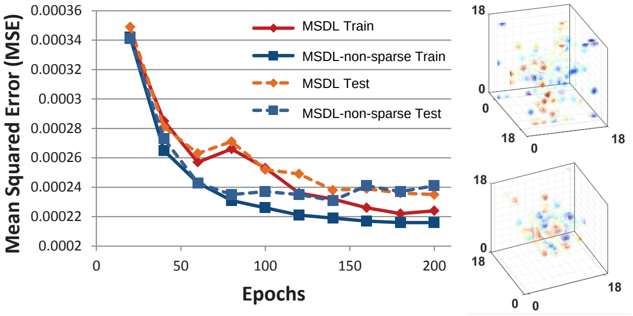

Fig. 7. **Left:** Training and testing error of the SADNN in the translation stage. The performance oscillation of the sparse model is explained by the enforcement of sparsity in the dense sampling layer of the network. **Right:** Example color-coded sparse patterns for translation (upper box) and full space (lower box). The latter representation is more compact because of the better data alignment.

Secondly, the sparsity effect of the ReLU activation (when applied on negative inputs) might not be that meaningful in our system, where sparsity is heavily enforced during training. We complete our model with a mean squared loss (MSE) instead of the cross-entropy loss which is usually suited for classification tasks. Given the fact that the target of the MSDL framework is the robust fusion / aggregation of hypotheses into a final result, we hypothesize that the MSE loss is more adequate for this task, given the increased smoothness of this type of loss function around the ground truth, compared to the cross-entropy loss.

2) Estimation Performance: We quantify the performance of the object localization step by measuring the position difference of the center of the detected box compared to the reference box and by computing the corner distance error, i.e. the average distance between the 8 corners of the detected box and the ground truth box. While the first value only measures the accuracy in terms of translation, the latter gives also an indication about the accuracy of the orientation and scale estimation. Table I shows the obtained results. The MSDL framework significantly improves the performance of the state-of-the-art MSL solution, decreasing the mean position error by 45.2% and the corner distance error by 39.6%. In this context, Figure 7(left) highlights the fact that using sparsity is an important step towards reaching this performance level. The error variation on the training data is caused by the iterative enforcement of sparsity in the low-

TABLE II
COMPARISON OF THE PERFORMANCE OF THE STATE-OF-THE-ART MSL [5] AND THE PROPOSED MSDL FRAMEWORK FOR AORTIC VALVE SEGMENTATION. THE MEASURE ILLUSTRATES THE DISTANCE OF THE DETECTED MESH TO THE GROUNDTRUTH FOR BOTH THE INITIALIZATION FROM THE DETECTED BOX AS WELL AS THE DISTANCE AFTER BOUNDARY REFINEMENT. THE SUPERIOR RESULTS ARE DISPLAYED IN BOLD.

	Training Data, distance to ground truth mesh [mm]				**Test Data, distance to ground truth mesh [mm]**			
	Initialization		**Final**		**Initialization**		**Final**	
	MSL	MSDL	MSL	MSDL	MSL	MSDL	MSL	MSDL
Mean	2.08	**1.16**	1.17	**0.89**	2.06	**1.21**	1.04	**0.90**
Median	1.94	**1.09**	1.05	**0.82**	1.95	**1.10**	0.98	**0.80**
STD	0.83	**0.40**	0.66	**0.35**	0.79	**0.55**	0.50	**0.48**

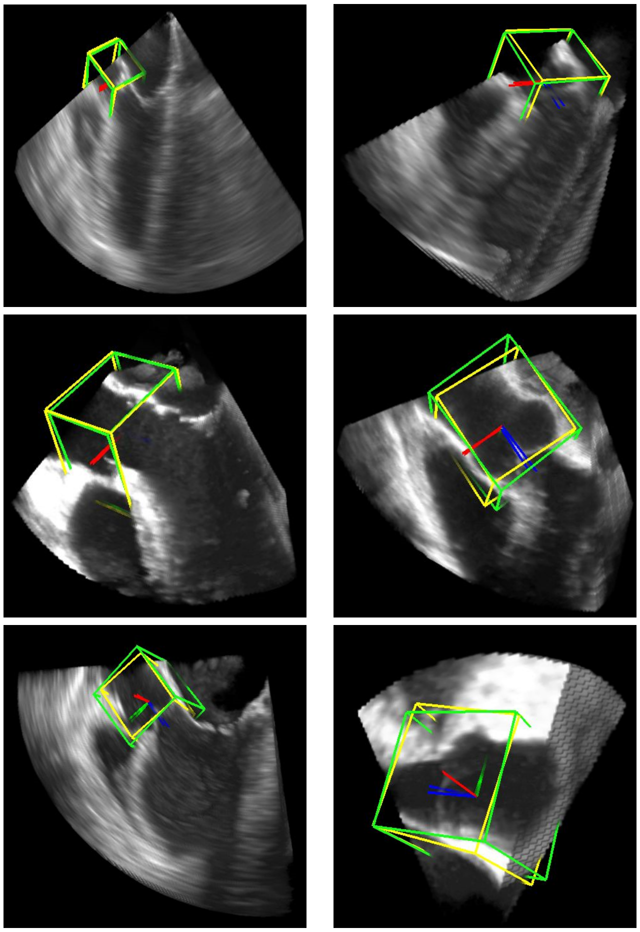

Fig. 8. Example images showing the detection results for different patients from the test set. The detected bounding box is visualized in green and the ground truth box in yellow. The segments with the origin in the center of each box define the corresponding coordinate system. Note that as specified in the text, the 3D pose of the aortic valve (position, orientation and scale) is fully determined by the anatomy.

level kernels, directly impacting the immediate performance of the network. The affected neurons however recover during the following training epochs to reach and even surpass the original performance. The reason is that by enforcing sparsity we introduce regularization in the network, which helps to prevent overfitting (see performance on the hold-out test set).

In Figure 7(right), we illustrate an example of the learned sparse sampling patterns, emphasizing that in the later stages, i.e. in the full 9D space, the patterns are more compact and structured around relevant anatomical parts (in our case around the aortic root). Qualitative results for different patients from the test set are depicted in Figure 8. The MSDL framework matches the excellent run-time performance of the MSL framework running in less than 0.5 seconds using only the CPU. This is greatly due to the use of sparse data sampling patterns which bring a speed-up of $\times 300$ to the MSDL-non-sparse, hence also the significant computational benefit of the network simplification. We also emphasize here that using a cascade for the early rejection of hypotheses is essential to reaching this competitive performance. Depending on the imbalance ratio in each marginal space, using a cascade brings around 2 orders of magnitude speed-wise improvement.

For the segmentation experiments, we initialize the boundary deformation by aligning the mean mesh with the detected box and refining the boundary. The performance is measured by computing the distance between the segmentation and the ground truth annotation mesh. Table II shows that the MSDL approach outperforms the MSL method by reducing the average mesh error from 1.04 mm to 0.9 mm, an improvement of 13.5%. Similar accuracy of around 1 voxel mean mesh error on the same image modality is also reported in [15], using a multi-atlas segmentation approach combined with deformable medial modeling. However there the image set used for validation contains only 22 cases. Also in an indirect comparison against methods evaluated not only on different patient sets, but also different image modalities, our framework shows a competitive accuracy. For example, Elattar *et al.* [16] present a method for aortic root segmentation in CT angiography data reaching a mean error of 0.74 mm on a test set of 20 cases, with a runtime of around 90 seconds/case. Adaptations of the MSL approach with which we directly compare in this work reach an accuracy of 1.08 mm on C-arm CT data [46], respectively 1.22 mm mean mesh error on 4D CT data [18]. In both these cases the segmentation of the aortic root is performed in around 0.8 seconds and the number of patients used for evaluation is in the order of hundreds.

We emphasize here that our DL-based approach shows competitive runtime performance, achieving the segmentation in under 1 second using only the CPU. Qualitative results can be seen in Figure 9.

3) Additional Experiments and Discussion: Given the fact that the accuracy of the nonrigid shape segmentation directly depends on the accuracy of the object detection, Table II is not an optimal indicator for the superiority of the DL-based

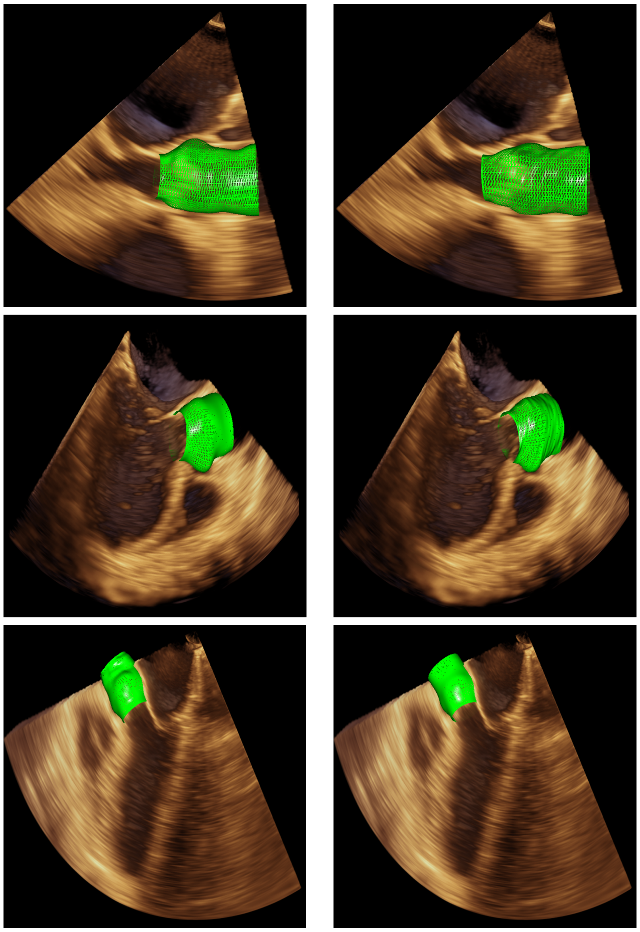

Fig. 9. Example images showing the aortic valve segmentation results for different patients from the test set, using our proposed method. Each row of images corresponds to a different patient, the **left** image represents the groundtruth mesh, while the **right** image shows the detected mesh.

segmentation alone. In this context we have investigated the performance of two mixed framework-versions: one combining our MSDL object detection with the original segmentation approach from Zheng *et al.* [5] and the other combining the MSL object detection solution [5] with our the DL-based segmentation. This gives a direct comparison between typical handcrafted features and adaptive sparse patterns in capturing the image context around the boundary. The experiment highlights the overall superiority of our end-to-end DL-based approach on the test set. Table III shows the results.

TABLE III
COMPARISON OF THE PERFORMANCE OF OUR DL-BASED FRAMEWORK WITH THE STATE-OF-THE-ART SOLUTION [5] AND THE TWO MIXED VARIANTS: COMBINING THE MSDL BOX DETECTION WITH THE ORIGINAL SEGMENTATION METHOD [5], RESPECTIVELY THE MSL BOX DETECTION [5] WITH OUR DL-BASED SEGMENTATION METHOD. THE SUPERIOR RESULTS ARE DISPLAYED IN BOLD.

	Segmentation error [mm]		
	Mean	Median	STD
Ours	**0.90**	**0.80**	**0.48**
Reference [5]	1.04	0.98	0.50
Detection (ours) + Seg. [5]	0.95	0.88	0.48
Detection [5] + Seg. (ours)	1.00	0.90	0.51

Replacing the current cascaded solution with a single end-to-end deep architecture (for example a convolutional neural network) and comparing the accuracy is not feasible - the main reason is the limited computational performance. We have experimented with using a CNN both in the cascade and as main deep classifier. Our implementation is based on direct calls to the latest cuDNN 4.0 library on top of a high-end GTX980 GPU architecture. We found that reasonable architectures with 1 convolution layer for the cascade, respectively 3 convolution layers (with pooling and fully-connected) for the main classifier are about $400 - 500$ times slower than our original solution. For example, in the orientation stage a deep 3D-CNN could process around 600 hypotheses/second compared to $420,000$ processed with a SADNN. When comparing shallow versions of these type of networks for cascade filtering, the SADNN is around $200\times$ faster, processing over $1,200,000$ hypotheses/second using only the CPU. In this context, removing the cascade and using a single end-to-end deep CNN can only further increase this performance deficit, both during training and testing. As such, training our framework with standard deep networks and investigating their accuracy becomes infeasible, requiring in the order of thousands of hours of training time.

The main reason for this speed difference is not only the cascade filtering but also the data-efficiency of the sparse sampling. Recall that most sparse patterns in our SADNN architecture reach sparsity levels of $90 - 95\%$, thus indexing only a small fraction of the data voxels during scanning and minimizing the memory-footprint in the large incoming stream of volumetric data. This is different from the CNN architecture which samples every single voxel multiple times, proportionally to the size of the convolution kernel. However integrating the CNN in our pipeline in a type of hybrid learning system is part of our ongoing work. For example, we are optimistic in managing to integrate such precise deep architectures into our pipeline to process only small subsets of difficult hypotheses, to further increase the accuracy of the system.

V. CONCLUSION

In this work we presented a novel framework supporting the efficient and robust detection and segmentation of arbitrary objects using parametrized representations. For the detection task our solution is the Marginal Space Deep Learning framework, which combines the inherent computational benefits of marginal space learning for efficient exploration of large parameter spaces, with the descriptive power of deep-learning-based data representations. The direct application of deep learning in this context is however not computationally feasible. For this we introduced a novel method for enforcing sparsity in the layers of deep neural networks, creating sparse, adaptive sampling patterns that replace the standard, pre-determined, handcrafted features. To further increase the evaluation speed and also address the high sample imbalance we proposed a novel technique to filter negative hypotheses using a cascade-like hierarchy of shallow neural networks.

Finally, given the object pose, we applied these techniques to build a deep-learning-based active shape model guiding the automatic segmentation of the object shape. Our method has been extensively tested on a challenging 3D detection and segmentation task, outperforming the state-of-the-art by a considerable margin. This represents, to the best of our knowledge, the first deep-learning-based method, focused on volumetric detection and segmentation with parametrized representations in the context of volumetric image parsing.

References

[1] Y. Bengio, A.C. Courville, and P. Vincent, "Unsupervised feature learning and deep learning: A review and new perspectives," *Computing Research Repository*, vol. abs/1206.5538, 2012.

[2] D.G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, vol. 2, 1999, pp. 1150–1157.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[4] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. 511–518.

[5] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, and D. Comaniciu, "Four-chamber heart modeling and automatic segmentation for 3-D cardiac CT volumes using marginal space learning and steerable features," *IEEE Transactions on Medical Imaging*, vol. 27, no. 11, pp. 1668–1681, 2008.

[6] J. Feulner, S.K. Zhou, M. Hammon, S. Seifert, M. Huber, D. Comaniciu, J. Hornegger, and A. Cavallaro, "A probabilistic model for automatic segmentation of the esophagus in 3-D CT scans," *IEEE Transactions on Medical Imaging*, vol. 30, no. 6, pp. 1252–1264, 2011.

[7] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham, "Active shape models - Their training and application," *Journal of Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] G.E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation Journal*, vol. 18, no. 7, pp. 1527–1554, 2006.

[10] P. Smolensky, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, 1986.

[11] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[12] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, U.D. Montral, and M. Qubec, "Greedy layer-wise training of deep networks," in *Conference on Neural Information Processing Systems*. MIT Press, 2007.

[13] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "ImageNet classification with deep convolutional neural networks," in *NIPS*. Curran Associates, Inc., 2012, pp. 1097–1105.

[14] J. Schmidhuber, "Multi-column deep neural networks for image classification," in *CVPR*, 2012, pp. 3642–3649.

[15] A.M. Pouch, S. Tian, M. Takabe, J. Yuan, R.G. Jr., A.T. Cheung, H. Wang, B.M. Jackson, J.H.G. III, R.C. Gorman, and P.A. Yushkevich, "Medially constrained deformable modeling for segmentation of branching medial structures: Application to aortic valve segmentation and morphometry," *Medical Image Analysis*, vol. 26, no. 1, pp. 217–231, 2015.

[16] M. Elattar, E. Wiegerinck, F. Kesteren, L. Dubois, N. Planken, E. Vanbavel, J. Baan, and H. Marquering, "Automatic aortic root landmark detection in CTA images for preprocedural planning of transcatheter aortic valve implantation," *The International Journal of Cardiovascular Imaging*, pp. 1–11, 2015.

[17] R.I. Ionasec, I. Voigt, B. Georgescu, Y. Wang, H. Houle, F. Vega-Higuera, N. Navab, and D. Comaniciu, "Patient-specific modeling and quantification of the aortic and mitral valves from 4-D cardiac CT and TEE." *IEEE Transactions on Medical Imaging*, vol. 29, no. 9, pp. 1636–1651, Sep 2010.

[18] S. Grbic, R. Ionasec, D. Vitanovski, I. Voigt, Y. Wang, B. Georgescu, N. Navab, and D. Comaniciu, "Complete valvular heart apparatus model from 4D cardiac CT," *Medical Image Analysis*, vol. 16, no. 5, pp. 1003 – 1014, 2012.

[19] A.M. Pouch, H. Wang, M. Takabe, B.M. Jackson, J.H.G. III, R.C. Gorman, P.A. Yushkevich, and C.M. Sehgal, "Fully automatic segmentation of the mitral leaflets in 3D transesophageal echocardiographic images using multi-atlas joint label fusion and deformable medial modeling," *Medical Image Analysis*, vol. 18, no. 1, pp. 118–129, 2014.

[20] F.C. Ghesu, B. Georgescu, Y. Zheng, J. Hornegger, and D. Comaniciu, "Marginal space deep learning: Efficient architecture for detection in volumetric image data," in *MICCAI*, 2015, pp. 710–718.

[21] L. Lu, J. Bi, S. Yu, Z. Peng, A. Krishnan, and X.S. Zhou, "Hierarchical learning for tubular structure parsing in medical imaging: A study on coronary arteries using 3D CT angiography," in *ICCV*, 2009, pp. 2021–2028.

[22] H.C. van Assen, M.G. Danilouchkine, A.F. Frangi, S. Ordas, J.J.M. Westenberg, J.H.C. Reiber, and B.P.F. Lelieveldt, "SPASM: A 3D-ASM for segmentation of sparse and arbitrarily oriented cardiac MRI data." *Journal of Medical Image Analysis*, vol. 10, no. 2, pp. 286–303, 2006.

[23] S.C. Mitchell, J.G. Bosch, B.P.F. Lelieveldt, R.J. van der Geest, J.H.C. Reiber, and M. Sonka, "3-D active appearance models: Segmentation of cardiac MR and ultrasound images." *IEEE Transactions on Medical Imaging*, vol. 21, no. 9, pp. 1167–1178, 2002.

[24] A. Mitiche and H. Sekkati, "Optical flow 3D segmentation and interpretation: A variational method with active curve evolution and level sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1818–1829, 2006.

[25] P. Pedersen, J. Quartararo, and T. Szabo, "Segmentation of speckle-reduced 3D medical ultrasound images," in *IEEE Ultrasonics Symposium*, 2008, pp. 361–366.

[26] F. Tombari and L. Di Stefano, "3D Data segmentation by local classification and Markov random fields," in *Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, 2011, pp. 212–219.

[27] L. Zhukov, Z. Bao, I. Gusikov, J. Wood, and D.E. Breen, "Dynamic deformable models for 3D MRI heart segmentation," *SPIE Medical Imaging*, vol. 4684, pp. 1398–1405, 2002.

[28] R.J. Schneider, N.A. Tenenholtz, D.P. Perrin, G.R. Marx, P.J. del Nido, and R.D. Howe, "Patient-specific mitral leaflet segmentation from 4D ultrasound," in *MICCAI*, 2011, pp. 520–527.

[29] W. Zhang, R. Li, H. Deng, L. Wang, W. Lin, S. Ji, and D. Shen, "Deep convolutional neural networks for multi-modality isointense infant brain image segmentation," *NeuroImage*, vol. 108, pp. 214 – 224, 2015.

[30] B. van Ginneken, A. Frangi, J. Staal, B. ter Haar Romeny, and M. Viergever, "Active shape model segmentation with optimal features," *IEEE Transactions on Medical Imaging*, vol. 21, no. 8, pp. 924–933, 2002.

[31] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," *Computing Research Repository*, vol. abs/1312.6229, 2013.

[32] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[33] Y. Sawada and K. Kozuka, "Transfer learning method using multi-prediction deep Boltzmann machines for a small scale dataset," in *Machine Vision Applications (MVA), 2015 14th IAPR International Conference on*, 2015, pp. 110–113.

[34] H. Chen, D. Ni, J. Qin, S. Li, X. Yang, T. Wang, and P. Heng, "Standard plane localization in fetal ultrasound via domain transferred deep neural networks," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 99, pp. 1627–1636, 2015.

[35] Y. Bar, I. Diamant, L. Wolf, S. Lieberman, E. Konen, and H. Greenspan, "Chest pathology detection using deep learning with non-medical training," in *Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium on*, April 2015, pp. 294–297.

[36] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention - (MICCAI)*, 2015, pp. 234–241.

[37] H.R. Roth, L. Lu, A. Seff, K.M. Cherry, J. Hoffman, S. Wang, J. Liu, E. Turkbey, and R.M. Summers, "A new 2.5D representation for lymph node detection using random sets of deep convolutional neural network observations," in *MICCAI*, 2014, pp. 520–527.

[38] A. Prasoon, K. Petersen, C. Igel, F. Lauze, E. Dam, and M. Nielsen, "Deep feature learning for knee cartilage segmentation using a tripla-

nar convolutional neural network," in *Medical Image Computing and Computer-Assisted Intervention*, vol. 8150, 2013, pp. 246–253.

[39] G.E. Hinton and R.R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[40] R. Salakhutdinov and G.E. Hinton, "Deep Boltzmann machines," in *Conference on Artificial Intelligence and Statistics*, 2009, pp. 448–455.

[41] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, 1986.

[42] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[43] D.C. Ciresan, A. Giusti, L.M. Gambardella, and J. Schmidhuber, "Mitosis detection in breast cancer histology images with deep neural networks," in *MICCAI*, vol. 2, 2013, pp. 411–418.

[44] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 530–549, 2004.

[45] Z. Tu, "Probabilistic boosting-tree: learning discriminative models for classification, recognition, and clustering," in *International Conference on Computer Vision*, vol. 2, Oct 2005, pp. 1589–1596.

[46] Y. Zheng, M. John, R. Liao, A. Nottling, J. Boese, J. Kempfert, T. Walther, G. Brockmann, and D. Comaniciu, "Automatic aorta segmentation and valve landmark detection in c-arm ct for transcatheter aortic valve implantation," *Transactions on Medical Imaging*, vol. 31, no. 12, pp. 2307–2321, Dec 2012.