Vito – A Generic Agent for Multi-Physics Model Personalization: Application to Heart Modeling

Dominik Neumann[1,2], Tommaso Mansi[1], Lucian Itu[3], Bogdan Georgescu[1], Elham Kayvanpour[4], Farbod Sedaghat-Hamedani[4], Jan Haas[4], Hugo Katus[4], Benjamin Meder[4], Stefan Steidl[2], Joachim Hornegger[2], and Dorin Comaniciu[1]

[1] Imaging and Computer Vision, Siemens Corporate Technology, Princeton, NJ
[2] Pattern Recognition Lab, FAU Erlangen-Nürnberg, Germany
[3] Imaging and Computer Vision, Siemens Corporate Technology, Romania
[4] Department of Internal Medicine III, University Hospital Heidelberg, Germany

Abstract. Precise estimation of computational physiological model parameters from patient data is one of the main hurdles towards their clinical applicability. Designing robust estimation algorithms is often a tedious and model-specific process. We propose to use, for the first time to our knowledge, artificial intelligence (AI) concepts to learn how to personalize a computational model, inspired by how an expert manually personalizes. We reformulate the parameter estimation problem in terms of Markov decision process and reinforcement learning. In an off-line phase, the artificial agent, called Vito, automatically learns a representative state-action-state model through data-driven exploration of the computational model under consideration. In other words, Vito learns how the model behaves under change of parameters and how to personalize it. Vito then controls the on-line personalization by exploiting its automatically derived action policy. Because the algorithm is model-independent, personalizing a completely new model would require only adjusting some simple parameters of the agent and defining the observations to match, without the full knowledge of the model itself. Vito was evaluated on two challenging problems: the inverse problem of cardiac electrophysiology and the personalization of a lumped-parameter whole-body circulation model. Obtained results suggested that Vito could achieve equivalent goodness of fit than standard methods, while being more robust (up to 25% higher success rates) and with faster (up to three times) convergence rate. Our AI approach could thus make model personalization algorithms generalizable and self-adaptable to any patient, like a human operator.

1 Introduction

For the past decade, computational models of heart function have been explored to improve clinical management of patients with cardiomyopathies, from stratification to therapy planning [1, 2]. Yet, the high model complexity and the often noisy and sparse clinical data still hinder their personalization; i.e. the estimation of their parameters such that they capture the observed physiology (e.g. cardiac motion, electrocardiogram, etc.) and can predict outcome. A wide variety of parameter estimation approaches have been explored to personalize cardiac models

from clinical data [3, 4]. They all aim to iteratively reduce the misfit between model output and measurements using automatic optimization algorithms (e.g. variational or filtering approaches). Applied blindly, those techniques could easily fail on unseen data, if not supervised, due to parameter ambiguity. Therefore, complex algorithms have been designed combining cascades of optimizers in a very specific way to achieve the required robustness [5]. However, such methods are not generic and their generalization to varying data quality cannot be guaranteed. Reversely, an experienced human can almost always succeed in manually personalizing a model for any subject. One reason is that an expert is likely to have an intuition of model behavior from his prior knowledge on physiology and model design, and past personalization experience. This intuition definitely helps to solve the personalization task more effectively, even on unseen data.

Instead, we propose to address personalization from a learning perspective, inspired by the "human expert". Based on neuroscience theories of animal learning, reinforcement learning (RL) encompasses a set of approaches to make a virtual agent learn by interacting with the environment [6]. RL was first applied to game or simple control tasks. However, the past few years saw tremendous breakthroughs in RL for more complex, real-world problems [7]. In [8], the authors combine RL with deep learning to train an agent to play with 49 Atari games, yielding better performance than an expert in the majority of them thanks to an outstanding generalization property of the RL algorithm.

Motivated by these recent successes, we propose a novel RL-based personalization approach, henceforth called *Vito*, with the goal of designing a framework that can, for the first time to our knowledge, learn by itself how to estimate model parameters from clinical data while being model-independent. First, like an expert, Vito assimilates the behavior of the model in an off-line, one-time only data-driven exploration phase. From this knowledge, Vito learns the optimal strategy encoded by the MDP state-action-state tuples using RL [6]. The goal of Vito is to choose an action that maximizes future rewards, and therefore bring it to the state representing the solution of the personalization problem. To setup the algorithm, the user just needs to define what observations need to be matched and the agent state space discretization. Then everything is learned automatically. The algorithm does not depend on the underlying model. Vito is evaluated on two different tasks: the inverse problem of cardiac electrophysiology and the personalization of a lumped-parameter model of whole-body circulation. Obtained results suggest that Vito can achieve equivalent goodness of fit as standard optimization methods, is more robust and has faster convergence rate.

2 Method

2.1 Markov Decision Processes for Modeling Agent Behavior

An MDP (Fig. 1) is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, where $\mathcal{S} = \{s_1, \ldots, s_{|\mathcal{S}|}\}$ is a set of states that describe the agent, $\mathcal{A} = \{a_1, \ldots, a_{|\mathcal{A}|}\}$ is the set of actions, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0; 1]$ and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ describe the probability of transitioning from one state to another upon action a_t; and the immediate reward after doing

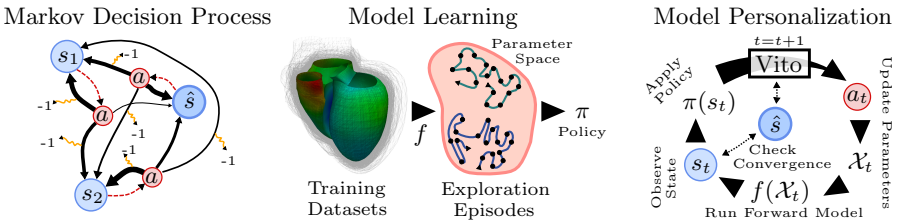

Fig. 1. Left: Simplified MDP (3 states, 1 action). Line width signifies $\mathcal{T}$. Negative rewards are given for all transitions, except when reaching state $\hat{s}$ (model personalized). **Mid**: Off-line model learning from training datasets. **Right**: On-line personalization.

so, respectively. γ is the discount factor that controls the importance of future versus immediate rewards [6]. The solution of an MDP is a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ that maximizes the discounted expected reward $\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t, s_{t+1})$.

2.2 Reformulation of the Model Personalization Problem into RL

Any computational model f is governed by a set of parameters $\mathcal{X} = \{x_1, \ldots, x_{|\mathcal{X}|}\}$ and characterized by state variables, some of them, $\mathcal{Y} = \{y_1^c, \ldots, y_{|\mathcal{Y}|}^c\}$, are observable and can be used to estimate $\mathcal{X}$. The goal is to minimize a set of objectives $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$, where $c_i = d(y_i^c, y_i^m)$, y_i^m is the measured data and d a distance function. Personalization is mapped to an MDP as follows. First, we define the MDP **states** $\mathcal{S}$ as $|\mathcal{C}|$-dimensional vectors with components $s_{t,i} = c_i$, encoding a combination of objective values. Hence, a state is characterized by its "distance" to the solution, which is patient-invariant. Continuous objective values are discretized into K_i bins, yielding K_i states per dimension i. The problem is therefore characterized by $\prod_i^{|\mathcal{C}|} K_i$ states. Next, mimicking a human operator, Vito will learn how to change the parameters $\mathcal{X}$ to fulfill the objectives $\mathcal{C}$. As such, the **actions** $\mathcal{A}$ modify the parameters, and consist in either in- or decrementing a parameter $x \in \mathcal{X}$ by $1\times$ or $10\times$ a user-specified reference value Δx. Inspired by the Mountain Car benchmark [6], we define the **reward** as always being equal to $\mathcal{R}(s_t, a_t, s_{t+1}) = -1$ (i.e. punishment) except when the agent performs an action that reaches the solution, i.e. when s_{t+1} is the state representing the smallest error $\hat{s}$. In that case, $\mathcal{R}(s_t, a_t, s_{t+1} = \hat{s}) = 0$. The **discount factor** $\gamma = 0.99$ encourages finding a policy that favors future over immediate rewards, as the latter could prefer finding local over global optima. Finally, the stochastic **transition function** $\mathcal{T}$ is learned automatically from f as described below.

2.3 Transition Function as Probabilistic Model Representation

The MDP transition probabilities $\mathcal{T}$ encode the agent's knowledge about the computational model f. Like a human operator, the agent first learns how the model "behaves" through self-guided "sensitivity analyses" (Fig. 1). First, we collect a batch of sample transitions from model exploration episodes $\mathcal{E} =$

$\{e_1, \ldots, e_{|\mathcal{E}|}\}$, i.e. sequences of state-action-state transitions, where each episode contains a fixed number of m (=100 in this work) consecutive transitions. An episode e is initiated by generating random model parameters $\mathcal{X}_0^e$ within physiologically plausible ranges. From the output of a forward model run $f(\mathcal{X}_0^e)$, we derive the initial state s_0^e as described in Sec. 2.2. Next, we employ a random exploration policy π_{rand} that randomly selects an action $a_0^e = \pi_{\text{rand}}(s_0^e)$ (uniform distribution) and applies a_0^e to $\mathcal{X}_0^e$, yielding $\mathcal{X}_1^e$. From $f(\mathcal{X}_1^e)$ the next state s_1^e is determined. We then select the next action $a_1^e = \pi_{\text{rand}}(s_1^e)$ and repeat this process $m-1$ times. Hence, each episode can be seen as a set of state-action-state tuples: $e = \{(s_t^e, a_t^e, s_{t+1}^e), t = 0 \ldots m-1\}$. Transition probabilities $\mathcal{T}$ are made patient-independent by exploring the model using different patients and combining the episodes computed on all of them in one big episode set $\mathcal{E}$. Finally, the transition probabilities $\mathcal{T}$ for each possible state-action-state transition are estimated. To this end, for each action $a \in \mathcal{A}$ and state $s \in \mathcal{S}$, we compute the relative frequency of (s, a, s') tuples in $\mathcal{E}$ among all $(s, a, \cdot)$ in $\mathcal{E}$, for all $s' \in \mathcal{S}$.

2.4 Learning How to Personalize a Model

Now that all components of $\mathcal{M}$ are defined, we compute a policy π that Vito will later use to decide which action to take given any possible state of a model personalization. To this end, value-iteration, a traditional MDP solving technique based on dynamic programming [6] is used. Value-iteration iteratively refines the state-value function V, which represents the expected sum of accumulated discounted rewards for any given state in the MDP: $V(s) = \sum_{s' \in \mathcal{S}} \mathcal{T}(s, \pi(s), s') \left[\mathcal{R}(s, \pi(s), s') + \gamma V(s')\right]$. The algorithm is guaranteed to converge to the optimal value function V^* for the given MDP. The optimal policy is given by $\pi(s) = \arg\max_{a \in \mathcal{A}} \sum_{s'} \mathcal{T}(s, a, s') \left[\mathcal{R}(s, a, s') + \gamma V^*(s')\right]$.

2.5 On-line Model Personalization

Once trained, Vito personalizes the computational model as follows. Starting from a default parameter set (e.g. normal values), Vito decides from the learned optimal policy π the first action to take, and walks through state-action-state sequences, guided by π, to personalize the computational model f. As observed in previous RL works [7], Vito could start oscillating between states. Since π is deterministic, oscillations can be automatically detected and Vito restarts the personalization from a randomly-selected state (attained by randomly sampling the model parameters). The personalization terminates when either Vito reaches the optimum $\hat{s}$, or when a maximum number of iterations N=100 is reached.

3 Experiments and Results

Two experiments were conducted to evaluate Vito: personalization of a cardiac electrophysiology (EP) model from ECG-derived parameters and personalization of a lumped whole-body circulation (WBC) model from volume and pressure

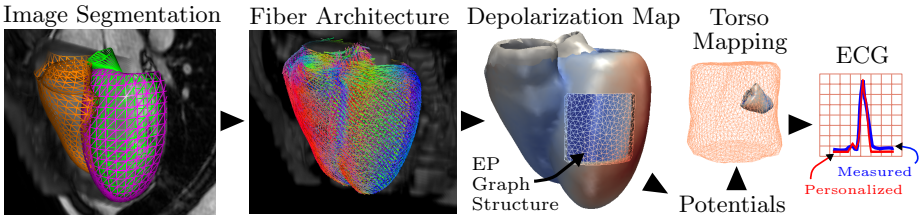

Fig. 2. EP simulation pipeline illustrated on a typical case (see text for details). The rightmost plot shows a personalization result provided by Vito against the ground-truth. As one can see, Vito was able to perfectly match the observed QRS signal.

measurements. For both experiments, the same set of 28 consecutive patients with no severe cardiac arrhythmias (QRS duration $\leq$ 120 ms) was used. For each patient, the bi-ventricular anatomy was segmented (Fig. 2) and tracked from short-axis MRI stacks using shape-constraints, learned motion models and diffeomorphic registration [5], from which ventricular volume curves were derived. At end-diastole, a tetrahedral anatomical model including myofibers was estimated and a torso atlas affinely registered to the patient based on MRI scout images [9]. We then randomly selected eight cases to determine Vito's hyper-parameters. More precisely, various state space discretization settings (K and the bin sizes) were manually tested, and the set yielding the best personalization was selected. It should be noted that this procedure could be easily automatized. Once the hyper-parameters were set, Vito was evaluated on the remaining 20 cases using a leave-one-out strategy: the off-line estimation of the transition probabilities $\mathcal{T}$ was carried out excluding the patient that was being personalized during the on-line phase. For fair comparison, just like Vito, the parameter estimation algorithms of reference terminated once all convergence criteria were met, and the maximum number of forward model runs was set to 100.

3.1 Personalization of Cardiac Electrophysiology Model

Forward Model Description: The depolarization time at each node of the tetrahedral anatomical model was computed using a shortest-path graph-based algorithm [10] (Fig. 2). Tissue anisotropy was modeled by modifying the edge costs to take into account fiber orientation. A time-varying voltage map was then derived according to the depolarization time: at a given time t, mesh nodes whose depolarization time was higher than t were assigned a trans-membrane potential of -70 mV, 30 mV otherwise. The time-varying potentials were then propagated to a torso model and QRS duration (QRSd) and electrical axis (EA) were computed [11]. The model was controlled by the conduction velocities (in m/s) of myocardial tissue, left and right Purkinje network ($\mathcal{X}_{\text{EP}} = \{v_{\text{Myo}}, v_{\text{LV}}, v_{\text{RV}}\}$), the latter two domains modeled as fast endocardial conducting tissue. The goal of EP personalization was to estimate $\mathcal{X}_{\text{EP}}$ from the measured QRSd and EA. Accounting for uncertainty in the measurements, the model was considered personalized if QRSd and EA misfits were below 5 ms and 10° respectively.

States and Actions: During hyper-parameters selection, large variations in the number of states (9 to 121) were tested. Yet, the observed variability in personalization outcomes was relatively low (standard deviation of ≈1 ms and 4° for mean QRSd and EA errors, respectively), suggesting that Vito's performance could be robust w.r.t. state discretization settings. The best results on eight datasets were obtained with $|\mathcal{S}|$=49 (K=7), with bin borders at $\{\pm 45, \pm 15, \pm 5\}$ ms for QRSd and $\{\pm 60, \pm 30, \pm 10\}$ degrees for EA. Reference increment values (Sec. 2.2) to build the action set of size $|\mathcal{A}|$=12 were set to Δx=5 m/s for all three $x \in \mathcal{X}_{\text{EP}}$.
Evaluation: $|\mathcal{E}|$=100 exploration episodes were generated per patient. Vito's personalization results were compared to BOBYQA, a standard, gradient-free optimization method where the objective function is the sum of absolute QRSd and EA errors. Vito achieved mean errors of 3.0±2.0 ms and 9.0±10.5°. The average goodness of fit yielded by BOBYQA was slightly better in terms of QRSd (1.4±2.6 ms) and slightly worse in terms of EA (9.9±14.7°). It required less forward model evaluations until convergence (31±7 versus 37±33 for Vito), however, this came at the cost of poor robustness: the personalization criteria could not be met in 7/20 cases. In comparison, Vito produced 70% less fail-cases. Despite the inherent ambiguity of states (several conduction velocity combinations could yield the same state), Vito was still able to personalize 18/20 cases.

3.2 Personalization of Whole-Body Circulation Model

Forward Model Description: The WBC model to personalize (Fig. 3) contained a heart model (left ventricle (LV) and atrium, right ventricle (RV) and atrium, valves), the systemic circulation (arteries, capillaries, veins) and the pulmonary circulation (arteries, capillaries, veins) [12]. Time-varying elastance models were used for all four chambers of the heart. The valves were modeled through a resistance and an inertance. A three-element Windkessel model was used for the systemic and pulmonary arterial circulation, while a two-element Windkessel model was used for the systemic and pulmonary venous circulation. The inputs of the model were the MRI-derived end-diastolic (ED) and end-systolic (ES) LV volumes, the heart rate and the systolic, diastolic, and average aortic pressures, as measured during catheterization. In this experiment, Vito learned how to personalize the systemic circulation to match five objectives: ED and ES LV volumes and the systolic, diastolic and average aortic pressures. To that end, we asked Vito to estimate five parameters $\mathcal{X}_{\text{WBC}}$: the initial LV volume, LV maximum elastance, LV dead volume, total arterial resistance and arterial compliance. To account for measurement noise, a personalization was considered successful if the final misfit was below ≈10% of average measured values in our population: 10 mmHg for pressure-, and 20 mL for volume-based objectives.
States and Actions: Like for EP, Vito appeared to be robust with respect to tested discretization settings, as quantified by the standard deviations of the mean errors after personalization (on average < 2.3 mmHg (or mL) for the five objectives). The best results were achieved with $|\mathcal{S}|$=675 (K=3 for pressures and K=5 for volumes), with bin borders at $\{\pm 10\}$ mmHg and $\{\pm 40, \pm 20\}$ mL. Based on the physiological ranges of parameter values, the reference increment values to

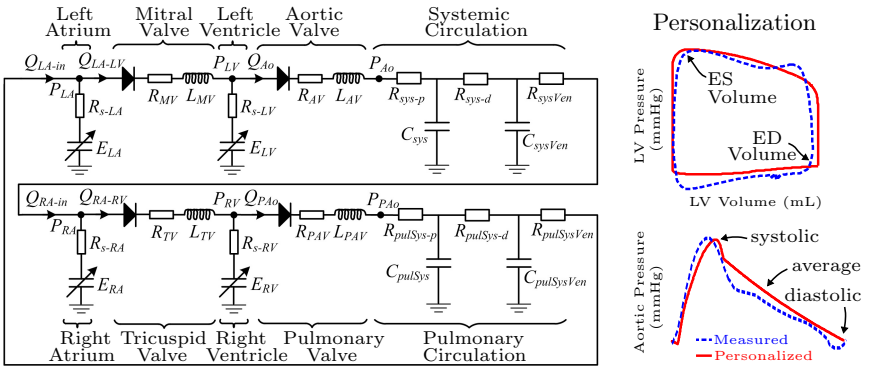

Fig. 3. Left: Whole-body lumped-parameter closed loop model of the cardiovascular system. P, Q, R, E, C and L represent pressures, flow, resistance, elastance, compliance and inertance, respectively. **Right**: Personalization objectives and results.

build the action set of size $|\mathcal{A}|$=20 were set to Δx={6 mL, 10^{-2} mmHg/mL, 4 mL, 8 g cm^{-4} s^{-1}, 1.5·10^{7}cm^{4} s^{2} g^{-1}} for the five parameters in $\mathcal{X}_{\text{WBC}}$ (see above). **Evaluation**: As one patient had no pressure data, evaluation was performed on the remaining 19 cases, for which $|\mathcal{E}|$=300 exploration episodes each were generated. Vito was compared to a standard optimization using the dogleg trust-region method [12]. Average errors over all pressure-based and all volume-based objectives after personalization were equivalent: 5±5 mmHg and 9±7 mL for Vito, and 5±5 mmHg and 9±6 mL for the comparison method. Vito was however more robust (2 versus 3 failed cases), and at the same time converged three times faster (16±30 versus 48±30 forward model runs), despite the large complexity of the model. Furthermore, preliminary experiments on ten randomly selected sets of initial parameter values suggested that Vito is robust w.r.t. initializations.

4 Discussion and Conclusion

This paper presented a novel personalization approach, Vito, based, for the first time, on AI concepts. We successfully applied it to two challenging personalization tasks in cardiac computational modeling. Inspired by the human approach, Vito first learns the underlying characteristics of the model under consideration using a data-driven approach. This knowledge is then utilized for automatically building a model-specific MDP. Vito is generic in the sense that beyond minimal user input (parameter ranges and authorized updates, discretization bins), it is able to learn by itself how to personalize a model. Setting up Vito thus does not require strong model knowledge. We showed that Vito can be faster and more robust than standard personalization methods, the same we would have expected from a human operator. As such, Vito could become a unified framework for personalization of any physiological model, potentially eliminating the need for an expert operator with in-depth knowledge to design complex optimization procedures. Important challenges still remain, like the definition of states and

their discretization. In this work we rely on manually defined bins, which could be seen as ordinal evaluation of the goodness of fit (e.g. "good", "satisfactory" and "bad" states). However, advanced approaches for continuous RL with value function approximators [6] could be integrated to fully circumvent discretization issues. At the same time, such methods could improve Vito's scalability towards higher-dimensional estimation tasks. Experience replay [8] or similar techniques could be employed to increase training data efficiency, which becomes important when computationally expensive models are considered. In the future, a thorough evaluation of convergence properties for both training and personalization will be carried out. Beyond these challenges, Vito showed promising performance and versatility, making it a first step towards an automated, self-taught model personalization agent.

References

1. Trayanova, N.A.: Whole-heart modeling applications to cardiac electrophysiology and electromechanics. Circulation Research **108**(1) (2011) 113–128
2. Nordsletten, D., Niederer, S., Nash, M., Hunter, P., Smith, N.: Coupling multiphysics models to cardiac mechanics. PBMB **104**(1) (2011) 77–88
3. Wallman, M., Smith, N.P., Rodriguez, B.: Computational methods to reduce uncertainty in the estimation of cardiac conduction properties from electroanatomical recordings. Med. Image Anal. **18**(1) (2014) 228–240
4. Wong, K.C., Sermesant, M., Rhode, K., Ginks, M., Rinaldi, C.A., Razavi, R., Delingette, H., Ayache, N.: Velocity-based cardiac contractility personalization from images using derivative-free optimization. J Mech Behav Biomed **43** (2015)
5. Seegerer, P., Mansi, T., Jolly, M.P., Neumann, D., Georgescu, B., Kamen, A., Kayvanpour, E., Amr, A., Sedaghat-Hamedani, F., Haas, J., et al.: Estimation of regional electrical properties of the heart from 12-lead ECG and images. In: STACOM Springer (2015) 204–212
6. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press (1998)
7. Kveton, B., Theocharous, G.: Kernel-based reinforcement learning on representative states. In: Association for the Advancement of Artificial Intelligence. (2012)
8. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540) (2015) 529–533
9. Neumann, D., Mansi, T., Georgescu, B., Kamen, A., Kayvanpour, E., Amr, A., Sedaghat-Hamedani, F., Haas, J., Katus, H., Meder, B., Hornegger, J., Comaniciu, D.: Robust image-based estimation of cardiac tissue parameters and their uncertainty from noisy data. In: MICCAI. Springer (2014) 9–16
10. Wallman, M., Smith, N.P., Rodriguez, B.: A comparative study of graph-based, eikonal, and monodomain simulations for the estimation of cardiac activation times. IEEE Transactions on Biomedical Engineering **59**(6) (2012) 1739–1748
11. Zettinig, O., Mansi, T., Neumann, D., Georgescu, B., Rapaka, S., Seegerer, P., Kayvanpour, E., Sedaghat-Hamedani, F., Amr, A., Haas, J., Steen, H., Meder, B., Navab, N., Kamen, A., Comaniciu, D.: Data-driven estimation of cardiac electrical diffusivity from 12-lead ECG signals. Med. Image Anal. **18**(8) (2014) 1361–1376
12. Itu, L., Sharma, P., Georgescu, B., Kamen, A., Suciu, C., Comaniciu, D.: Model based non-invasive estimation of PV loop from echocardiography. In: Engineering in Medicine and Biology Society (EMBC). (2014) 6774–6777