Hough Forests for Real-Time, Automatic Device Localization in Fluoroscopic Images: Application to TAVR

Charles R. Hatt[1,2], Michael A. Speidel[1,2], and Amish N. Raval[2]

[1] Department of Medical Physics, University of Wisconsin, Madison, USA
[2] Division of Cardiovascular Medicine, University of Wisconsin, Madison, USA

Abstract. A method for real-time localization of devices in fluoroscopic images is presented. Device pose is estimated using a Hough forest based detection framework. The method was applied to two types of devices used for transcatheter aortic valve replacement: a transesophageal echo (TEE) probe and prosthetic valve (PV). Validation was performed on clinical datasets, where both the TEE probe and PV were successfully detected in 95.8% and 90.1% of images, respectively. TEE probe position and orientation errors were 1.42 ± 0.79 mm and $2.59^\circ \pm 1.87^\circ$, while PV position and orientation errors were 1.04 ± 0.77 mm and $2.90^\circ \pm 2.37^\circ$. The Hough forest was implemented in `CUDA C`, and was able to generate device location hypotheses in less than 50 ms for all experiments.

1 Introduction

Detection and pose estimation of devices in x-ray fluoroscopic (XRF) images is a challenging but important task for enabling multimodal image fusion in cardiac interventional procedures. For example, catheter detection and tracking can be used to provide motion compensation of anatomical roadmaps used to help guide electrophysiology procedures [1]. Another application which has recently gained interest is transesophageal echo (TEE) to XRF registration [2]. TEE/XRF registration allows anatomical information from echo to be combined with device imaging from XRF.

A procedure that may benefit from a smart integration of XRF and TEE imaging is transcatheter aortic valve replacement (TAVR). For example, obtaining the optimal 3D echo cut-planes for anatomical and device visualization is non-trivial, even for experienced echocardiographers. Furthermore, once the optimal echo view is obtained, the device is not necessarily easy to visualize. By registering the two modalities, a prosthetic valve (PV) can be detected in XRF, and its position and orientation may be used to compute the optimal echo cut-planes for visualization. The PV can then potentially be rendered within the 3D echo volume (Fig. 1) as an alternate imaging tool for guiding PV deployment.

A key component of the clinical workflow is automatic localization of the devices at the beginning of an image sequence. In this paper, we describe a common framework for TEE and PV localization in XRF images. A Hough

N. Navab et al. (Eds.): MICCAI 2015, Part I, LNCS 9349, pp. 307–314, 2015.
DOI: 10.1007/978-3-319-24553-9_38

forest (HF) detector was trained that can detect multiple parts of each device, allowing for estimation of in-plane pose parameters. The data was validated on 1077 clinical images of the TEE probe and 388 of the PV.

Previous Work. In [3], the TEE probe was detected using the probabilistic boosting-tree approach with Haar wavelets and steerable features. Out-of-plane rotations were estimated using an oriented gradient binary template library. Average detection time was 0.53 seconds. In [4], the work from [3] was extended by focusing on a framework for adapting a classifier generated with *in silico* training data to perform better on *in-vivo* test data. Impressive results for detection of in-plane TEE pose parameters were obtained in terms of localization accuracy, low false positive rate, and detection speed.

In [5], the PV was manually segmented and then automatically tracked using template matching. To eliminate the need for manual interaction during computed-aided interventions, the method presented in this paper focuses on automatic device localization using a HF framework. Previously, this framework was used for anatomy localization in CT volumes [6]. To the best of our knowledge, our work is the first to employ a real-time HF for device localization during image guided interventions.

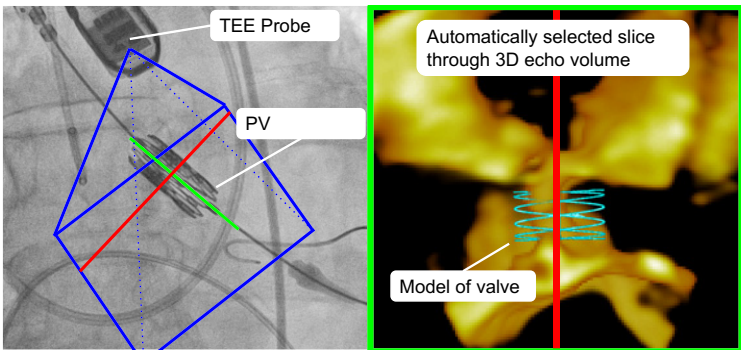

Fig. 1. Potential workflow enabled by TEE/XRF registration and PV detection. In the XRF image, the red line perpendicular to the PV corresponds to a plane through the echo image. The green line matches the viewing plane of the echo image.

2 Methods

2.1 Algorithm

We employ the HF framework for object localization [7]. A key component of our implementation is the simultaneous detection of multiple object parts, which allows for estimation of device pose under varying orientations. In the following section, a review of the HF object detection framework is presented in context of our application.

Hough Forest Detector. A HF is a specific type of random forest that is designed for object detection. A random forest is a collection of decision trees that perform classification and/or regression. A HF takes image patches as input, and simultaneously performs both classification (is it part of an object?) and regression (where is the object?). The term "Hough" comes from the idea that each input image patch classified as part of the object votes for the object center. Votes are added in an accumulator image ("Hough" image, Fig. 3), and peaks are considered as object detection hypotheses. In our implementation, we designed a HF that locates two ends of a device, referred to as the "tip" and "tail" (Fig. 3).

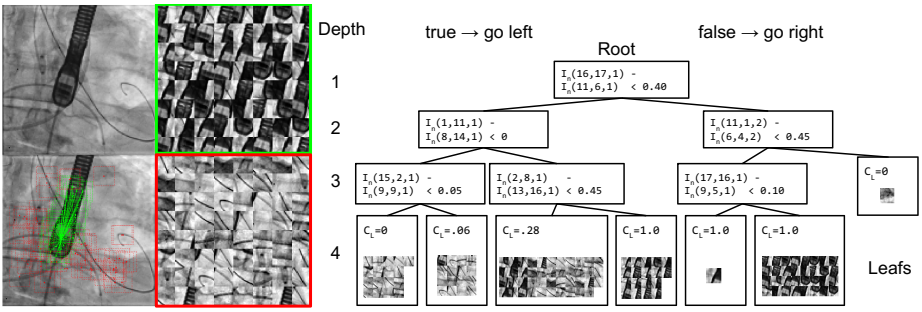

Fig. 2. A simple example of a decision tree trained on a single image of the TEE probe. Left: Example simulated TEE probe image, with locations of background (red) and device (green) training patches. Right: Example of a simple decision tree. Input data traverses the nodes based on binary test results and arrives at leaf nodes. In this example, all of the patches from the training image are shown in their destination leaf nodes.

A decision tree is an acyclic directed graph where each node contains a single input edge (except the root node) and two output edges (except the terminal nodes). During testing, data is input into the root node, and rules based on binary tests (aka features) determine which edge to travel down. For image patches, these binary tests typically encode patch appearance. Eventually the data will arrive at a terminal "leaf" node. The leaf node contains data, learned during training, about how to classify (or regress) the input data.

Each tree is trained by computing a set of binary tests on labeled training data, which are used to establish splitting rules. The splitting rules are chosen to maximize class discrimination at each node. In this work, binary pixel comparison tests are used due to their computational efficiency. Multi-channel image patches are used as input data, where a channel can be the raw pixel intensities or some operation computed on the intensities, e.g. gradient magnitude, blobness filter, etc... For each multi-channel input training patch I_n, a set of K binary tests are computed as follows:

$$F_{k,n}(p_k, q_k, r_k, s_k, \tau_k, z_k) = I_n(p_k, q_k, z_k) - I_n(r_k, s_k, z_k) < \tau_k \qquad (1)$$

Where (p, q) and (r, s) are patch pixel coordinates, τ is a threshold used for detecting varying contrast, and z is the channel index. Image channels used in this work were image intensity, the x-gradient and the y-gradient. Each channel of each patch is normalized to have a range of 1 ($I_z(u,v) = \frac{I_z(u,v)}{max(I_z)-min(I_z)}$) , I_z is the patch for channel z).

Training begins by inputing a $K \times N$ training matrix with N training patches and K tests into the root node (Fig. 2). For classification, a metric is computed for each test k over all samples. In this work, the metric used for classification is the information gain:

$$G_k^c = H(S) - \frac{|S_1|}{|S|}H(S_1) - \frac{|S_0|}{|S|}H(S_0) \tag{2}$$

$$H(S) = -\sum_{c \in C} p(c)log(p(c)) \tag{3}$$

Where S is the entire set of training data, S_0 is the set of training data where F_k is false and S_1 is the set of training data where F_k is true, and $H(S)$ is the Shannon entropy over all classes (device or background) in the set S.

Alternatively, for regression of continuous variables, the metric is:

$$G_k^r = |S|var(S) - |S_1|var(S_1) - |S_0|var(S_0) \tag{4}$$

Where $var(S)$ is the variance of continuous data describing the device orientation or offset vectors within each set (non-device patches are ignored for this calculation).

A random decision is made at each node on which attribute to base the splitting rule on: class, offsets, or device orientation. If the offsets are chosen, a random choice about which offsets to regress ("tip" or "tail") is made. The test that gives the maximum value of G_k^c or G_k^r is stored as the splitting rule for that node, and the training data is passed onto the left or right child node according to the splitting rule. The same process is completed until a maximum tree depth D is reached or all of the samples in a node belong to the background class. The terminal node is termed a "leaf" node, and it stores the classes labels and offsets associated with all of the training data that arrived at that node. In order to speed up run-time, offsets in each leaf node are partitioned into 16 clusters using k-means and the cluster means replace the original offsets.

A key feature of HFs is the use of randomness during training, which helps prevent over-fitting the classifier to the training data. This is accomplished by only generating a small random subset of binary pixel tests for each tree, as well as randomizing whether each node will build a splitting rule based on class, offset vector, or device orientation. For example, in our implementation for the TEE probe, 8192 out of over 1 million binary tests are available to each tree.

During testing, a new image patch centered on (u_p, v_p) is fed into the root node of each tree and traverses the tree according to the splitting rules established

during training. When it arrives at a leaf node, each offset (u_o, v_o) in the leaf node votes for the device parts in the Hough image accordingly:

$$I_H(u_H, v_H) \rightarrow I_H(u_H, v_H) + \frac{C_L}{|D_L|} \tag{5}$$

Where $(u_H, v_H) = (u_p, v_p) + (u_o, v_o)$, C_L is the proportion of device samples in the leaf node, and $|D_L|$ is the number of offsets in the leaf node.

This process is then repeated at every patch and for every tree in the HF. The final I_H is blurred with a gaussian kernel and peaks are classified as tip and tail detection hypotheses (Fig. 3).

HF input patches can be sampled densely at random locations or sparsely at salient key-points. For our application, we found that device detection was faster and more reliable using densely sampled patches at random locations.

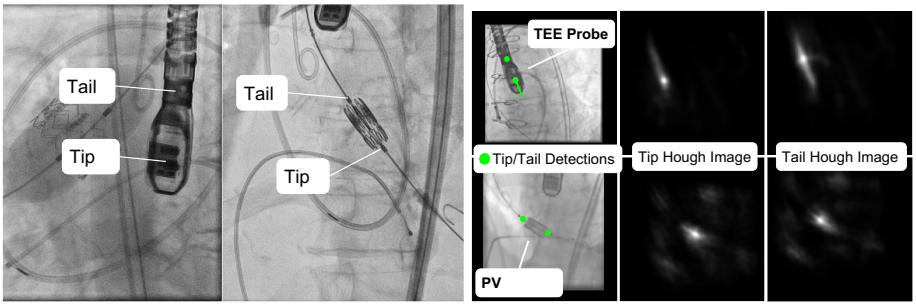

Fig. 3. Left: TEE probe and PV, with tip and tail labeled. Right: TEE probe and valve detection hypotheses with corresponding Hough images showing clearly defined peaks at the tip and tail of the devices.

Hypothesis Scoring. A Hough image peak was considered a valid hypothesis if it was $> 0.8 * max(I_H)$ following non-maximum suppression. At most, the top 10 peaks were retained as part hypotheses, but in practice usually only a few peaks survived the first criteria. All L tail and M tip hypotheses are combined to form $L \times M$ tip-tail pair hypotheses.

Next, unfeasible tail-tip pair hypotheses were removed. This was done by creating tail-tip pair distance and orientation matrices, and removing pair hypotheses that fell outside of the ranges of distance and orientation seen in the training datasets. Remaining tip-tail hypotheses are then given a score $S_{lm} = I_{H_{tip}}(u_l, v_l) \cdot I_{H_{tail}}(u_m, v_m)$. The tip-tail pair with the highest score is selected as the detected device.

2.2 Experiments

Computer Hardware and Software. All experiments were run on a Dell Precision T7500 work station running Ubuntu Linux with a 3.47 GHz Intel Xeon

processor and a NVIDIA Tesla K20 GPU. HF code was written in CUDA C. Retrospective clinical dataset processing was approved by the local institutional review board. The Philips X2-7t probe and the Edwards Sapien valve were used in this study.

Training Datasets. For the TEE probe, the classifier was trained on simulated XRF images. Similar to the method from [4], hybrid images were created by blending anatomical background images from TAVR cases with digitally reconstructed radiographs (DRRs) of the TEE probe. For the PV, 389 clinical images from TAVR cases were manually annotated and used for training. In order to increase the size of the training dataset for the PV detector, each training image was randomly rotated and re-used as if it were a new image. The PV was only trained and detected in the pre-deployment state.

Table 1. HF parameters for the TEE and PV. N = number of training samples. K = number of tests per tree. T = number of trees. D = tree depth.

	Patch size	N	K	T	D	Image resolution	Patches at run-time
TEE	17×17	65536	8192	32	10	1.0 mm	16384
PV	25×25	16384	8192	64	8	0.5 mm	16384

Validation. The TEE and PV detector were tested on 1077 and 388 clinical XRF images, respectively. Ground truth data for the TEE images was obtained by manually registering a model of the TEE probe to the image, followed by 2D/3D registration based refinement using the method from [2], which reported sub-millimeter in-plane position accuracy. The PV ground truth was obtained by manual annotation of the tip and tail in the test images.

For validation, we measured the rate of successful detections, the mean localization error for successful detections, and the orientation error for successful detections. HF run-time was also reported, which was the amount of time it took for the HF to process all patches for each tree and create the Hough images. A detection was considered successful if the distance error was less than 5 mm and the orientation error was $< 10°$. Localization error was the Euclidean distance between the true device center and the measured device center computed at the detector (i.e. projection magnification was not considered.)

3 Results

Results are summarized in Table 2. The rate of successful detections was 95.8% for the TEE probe and 90.1% for the PV. This was competitive with previously reported results for the TEE probe [3,4], especially when considering that the HF was trained on simulated images. For successful detections, both devices resulted in localization errors less than 1.5 mm on average, and orientation errors less than 3.0°.

Table 2. Detection results for the HF device detector.

	# Test images	Successful Detection Rate (%)	Localization Error (mm)	Orientation Error (°)	Run-time for HF (ms)
TEE	1077	95.8	1.42 ± 0.79	2.59 ± 1.87	38.8 ± 5.00
PV	388	90.1	1.04 ± 0.77	2.90 ± 2.37	37.0 ± 2.29

4 Discussion

The presented method was able to accurately detect both the TEE probe and the PV in over 90% of images. Most of the failed detections were due to occlusion from x-ray contrast during aortagraphy. The success rate for the PV was higher than expected, because a large percentage of the PV test images were recorded during contrast infusion. Furthermore, the PVs in the training and testing images varied greatly in size and appearance due to different patient sizes and valve models. This indicates that the HF classifier is robust to appearance variation and that greater detection performance may be possible using a classifier trained on single specific valve size and model.

The real-time performance of the method is contingent on the full image processing workflow. However, we expect that the bulk of processing is required by the HF, which we have shown has a maximum run-time less than 50 ms. The other steps, which comprise random patch location generation and extraction, can be implemented very efficiently on the GPU using texture reads. We expect that the full image processing workflow can be completed in less than 60 ms, which is sufficient for typical fluoroscopic imaging frame rates (15 fps)

The main application of these methods is to enable XRF/Echo image fusion, where the device will either be rendered in the echo image, or soft-tissue information from echo will be projected onto the XRF image. It is expected that these tools will minimize the need for use of x-ray contrast, which is not only healthier for the patient, but also decreases the risk of device detection failure. For the TEE probe, future work will focus on detection of the out-of-plane pose parameters, which is often a necessary step for fully automatic initialization of 2D/3D registration. For the PV, future work will focus on not only detecting the PV prior to deployment, but also during and after. This will allow a dynamic model of the PV to be rendered in echo images, potentially resulting in new image guidance tools for TAVR deployment.

5 Conclusion

A method for real-time, automatic detection of devices in fluoroscopic images is presented. Based on the Hough forest object detection framework, the method is fully automatic, and has the potential to operate at fluoroscopic frame rates. The percentage of successful device detections was 95.8% for the TEE probe and 90.1% for the prosthetic valve, despite the presence of x-ray contrast in many of the image frames. Future work will focus on detecting PV deformation during and after valve deployment for enhanced multi-modal guidance of TAVR.

References

1. Brost, A., Wimmer, A., Liao, R., Bourier, F., Koch, M., Strobel, N., Kurzidim, K., Hornegger, J.: Constrained registration for motion compensation in atrial fibrillation ablation procedures. IEEE Transactions on Medical Imaging 31(4), 870–881 (2012)
2. Gao, G., Penney, G., Ma, Y., Gogin, N., Cathier, P., Arujuna, A., Morton, G., Caulfield, D., Gill, J., Rinaldi, C.A., Hancock, J., Redwood, S., Thomas, M., Razavi, R., Gijsbers, G., Rhode, K.: Registration of 3D trans-esophageal echocardiography to x-ray fluoroscopy using image-based probe tracking. Medical Image Analysis 16(1), 38–49 (2012)
3. Mountney, P., et al.: Ultrasound and Fluoroscopic Images Fusion by Autonomous Ultrasound Probe Detection. In: Ayache, N., Delingette, H., Golland, P., Mori, K. (eds.) MICCAI 2012, Part II. LNCS, vol. 7511, pp. 544–551. Springer, Heidelberg (2012)
4. Heimann, T., Mountney, P., John, M., Ionasec, R.: Real-time ultrasound transducer localization in fluoroscopy images by transfer learning from synthetic training data. Medical Image Analysis 18(8), 1320–1328 (2014). Special Issue on the 2013 Conference on Medical Image Computing and Computer Assisted Intervention
5. Karar, M., Merk, D., Chalopin, C., Walther, T., Falk, V., Burgert, O.: Aortic valve prosthesis tracking for transapical aortic valve implantation. International Journal of Computer Assisted Radiology and Surgery 6(5), 583–590 (2011)
6. Criminisi, A., Shotton, J., Robertson, D., Konukoglu, E.: Regression forests for efficient anatomy detection and localization in CT studies. In: Menze, B., Langs, G., Tu, Z., Criminisi, A. (eds.) MICCAI 2010. LNCS, vol. 6533, pp. 106–117. Springer, Heidelberg (2011)
7. Gall, J., Lempitsky, V.: Class-specific hough forests for object detection. In: Criminisi, A., Shotton, J. (eds.) Decision Forests for Computer Vision and Medical Image Analysis. Advances in Computer Vision and Pattern Recognition, pp. 143–157. Springer, London (2013)