CrossTrack: Robust 3D Tracking from Two Cross-Sectional Views

Mohamed Hussein[1] Fatih Porikli[1] Rui Li[2] Suayb Arslan[3]
[1]Mitsubishi Electric Research Labs [2]Mass. General Hospital [3]Univ. of California at San Diego
{hussein,fatih}@merl.com rli4@partners.org sarslan@ucsd.edu

Abstract

One of the challenges in radiotherapy of moving tumors is to determine the location of the tumor accurately. Existing solutions to the problem are either invasive or inaccurate. We introduce a non-invasive solution to the problem by tracking the tumor in 3D using bi-plane ultrasound image sequences. We present CrossTrack, a novel tracking algorithm in this framework. We pose the problem as recursive inference of 3D location and tumor boundary segmentation in the two ultrasound views using the tumor 3D model as a prior. For the segmentation task, a robust graph-based approach is deployed as follows: First, robust segmentation priors are obtained through the tumor 3D model. Second, a unified graph combining information across time and multiple views is constructed with a robust weighting function. For the tracking task, an effective mechanism for recovery from respiration-induced occlusion is introduced. Our experiments show the robustness of CrossTrack in handling challenging tumor shapes and disappearance scenarios, with sub-voxel accuracy, and almost 100% *precision and recall, significantly outperforming baseline solutions.*

1. Introduction

Research in Image Guided Radiation Therapy (IGRT) [19, 1] aims at deploying advanced medical image analysis techniques to maximize the effectiveness of radiation therapy. One of the challenges of IGRT is treatment of moving tumors. Uncertainties in determining the tumor location may cause radiation beams to overshoot or undershoot, thus, either damage the healthy tissue or fail to control the tumor growth. The current clinical practice is to track moving tumors using external or internal surrogates [7]. Tracking based on external surrogates, *e.g*. measuring lung volume, are not considered accurate due to changes in the relationship between theses measurements and the tumor location over time. On the other hand, using internal surrogates, *e.g*. surgically implanted fiducial markers, is highly invasive and involves the risk of medical complications. Recently, purely image-based tracking approaches have been actively researched using fluoroscopy [10], or 4DCT [16]. While these approaches are less invasive than using internal surrogates and more accurate than using external surrogates, they introduce extra imaging radiation to the patient.

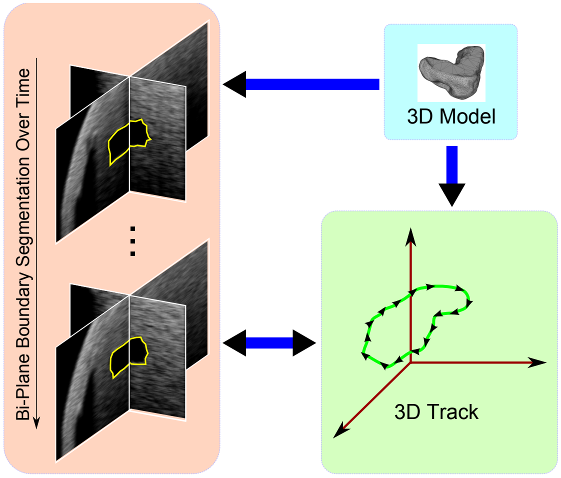

Figure 1. 3D tumor tracking from bi-plane ultrasound images by coupling segmentation and tracking, using the tumor 3D model.

Ultrasound imaging offers a noninvasive alternative to fluoroscopy and 4DCT. Being cost and time efficient, a high-frequency ultrasound system with 3-D imaging capabilities also achieves better resolution, discrimination and detection of abdominal metastases at a minimum size compared favorably with that of X-ray [5]. Ultrasound imaging depicts not only the center of the tumor but its whole volume and boundary for a large variety of high contrast neoplasms. It has already been used for detection and staging of tumors [11]. We believe that it can also be used in IGRT given that highly robust and accurate tracking algorithms are developed. In this paper we present CrossTrack, a tracking algorithm for ultrasound imaging that enjoys these features.

Yang *et al*. [20] introduced an algorithm for tracking the left ventricle in 3D ultrasound images. This algorithm is highly complex and relies on supervised learning on a training dataset, which may not be available in our case. Alternatively, 3D tracking in ultrasound can be considered as a sequential image segmentation problem where each pixel is labeled as foreground (tumor) and background (healthy tissue). To reduce the computational intensity and to eliminate the need for supervised training, we use two 2D ultrasound slices instead of full 3D volumes, as illustrated in Fig. 1. Our algorithm jointly segments the tumor boundary in the two slices. The tumor location in 3D is then inferred from the resulting segmentation given the tumor 3D model as a prior. We assume that the tumor becomes visible when it intersects with one of the ultrasound planes.

Many researchers approached the problem of coupling segmentation and tracking. Kohli and Torr [8] presented an algorithm for segmentation of a moving object using Dynamic Graph Cuts (DGC). DGC quickly segments a new frame using Graph Cuts (GC) [2] taking advantage of the final flow values of the last frame. In contrast, CrossTrack is not tailored to a specific segmentation algorithm. In fact, it can be applied to any graph-based segmentation algorithm. Ren and Malik [14] propagate segmentation masks over time using matching of super-pixels across frames. We believe that super-pixel segmentation is not suitable for ultrasound images due to high image noise and low contrast. Liang and Davis [21] addressed the problem of changing appearance of the tracked object over time by jointly performing segmentation and appearance modeling. This approach is very effective, but, highly computationally intensive. To the best of our knowledge, none addressed the problem of tracking from two cross-sectional views using the object's 3D model as a prior. Moreover, we present an effective method to recover from partial and total occlusion (disappearance), a common difficulty in coupled segmentation/tracking algorithms.

The main contributions of this paper are:

- A framework for tumor tracking in 3D from bi-plane ultrasound image sequences.
- An algorithm for 3D tracking of a volumetric object from two intersecting cross-sectional views. The algorithm uses the volumetric model prior of the tracked object by recursive coupling of 3D location estimation and 2D boundary segmentation.
- A novel graph construction that joins multiple-view and temporal information for graph-based segmentation, with a robust and easily-tunable link weighting function.
- A method for track recovery after tumor disappearance due to respiration-induced motion.

2. Algorithm Overview

CrossTrack couples two tasks: 3D tumor tracking, and 2D tumor boundary segmentation in the two ultrasound planes, as illustrated in Fig. 1. The two tasks assist one another and the tumor 3D model assists both of them. The interaction between segmentation and tracking can be viewed as a recursive inference process, as illustrated in the graphical model in Fig. 2. The tumor 3D model is used as a prior for the two inference directions. In one direction, given the current 2D segmentation and the 3D model, the current 3D location is inferred. In the other direction, given the last 3D location and the 3D model, the current 2D segmentation is inferred.

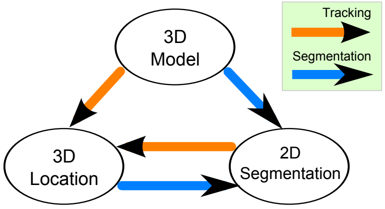

Figure 2. Graphical models for segmentation and tracking.

Tracking For inferring the 3D location, we use a simple model that proved effective in our framework. The new location is assumed to be sampled from a uniform distribution in a small sphere surrounding the last estimated location. Particles are sampled from that distribution, where each particle represents a hypothesis for the new location. Given the tumor 3D model, a segmentation mask is synthesized for each particle. By matching these segmentation hypothesis with the result of the segmentation task, a score is computed for each particle. We finally pick the particle that provides the best matching score (Maximum A Posteriori estimation). Occlusion resulting from respiration-induced motion is effectively handled via a simple update rule. For this paper, we only consider the coordinates of the 3D location as the state variables to track, *i.e.* we only consider translation motion. Extension of CrossTrack to a handle more general motion models is straightforward, in principle. However, tracking longer state vectors may require a more sophisticated probabilistic filtering scheme.

Segmentation Inferring the 2D segmentation using the latest 3D location estimation is the main focus of this paper. We start by giving a brief overview here. Given the latest estimated location (initial location for the first frame), a set of location particles are sampled and segmentation hypotheses are constructed for each, as described in the tracking task above. Using the segmentation hypotheses, a prior probability map for the segmentation is constructed. From this prior, segmentation seeds for the new frame are marked. Intensity distributions for foreground and background are learned from the intensity values of the marked seeds. This leads to another data-driven prior probability map for the segmentation. A single joint graph is constructed that combines graphs for two frames for each plane. A graph-based segmentation algorithm is run on the joint graph. The segmentation outcome is refined by replacing it with the best matching segmentation hypothesis.

The details of graph construction and computing prior probability maps are explained in Sect. 4. Occlusion handling is explained in Sect. 5. Before these topics, we start off with a brief background on graph-based segmentation in Sect. 3.

3. Graph-Based Segmentation

The image segmentation problem is one of the oldest and most studied problems in computer vision. For a comprehensive review of the segmentation literature, the reader is referred to dedicated survey papers [18, 3, 12]. Of particular interest to us are graph-based segmentation algorithms. In this category of algorithms, an input image is represented as a graph $G = (V, E, w)$, where $V = \{v_i | i = 1..N\}$ is a set of nodes, each corresponds to one of the N pixels in the image, $E = \{e_{ij} = (v_i, v_j) | i, j = 1..N\}$ is a set of links in the graph connecting neighboring image pixels, and $w : E \to \mathbb{R}$ is a weighting function that measures the similarity between the two nodes incident on a link. The segmentation problem is defined as finding an optimal discrete labeling function $l : V \to 1..K$, where K is the number of desired segments, with respect to some energy function. The energy function is designed to indicate the overall similarity among the nodes that are assigned the same label (smoothness), and the compliance of the labeling function to our prior knowledge (data). The simplest form of the energy function includes terms defined over singleton (data terms) and pairwise (smoothness terms) label assignments as

$$\mathbb{E}(l) = \sum_{v_i \in V} \mathbf{E}_1(l_i) + \sum_{e_{ij} \in E} \mathbf{E}_2(l_i, l_j) \quad , \tag{1}$$

where $l_i = l(v_i)$, and $\mathbf{E}_1$ and $\mathbf{E}_2$ define the singleton and pairwise energy terms.

There are many graph-based segmentation algorithms in the literature. Among them, GC [2], and Random Walks (RW) [4] are among the most popular ones. Both of them are computationally efficient. RW's formulation guarantees a unique optimal solution regardless of the number of labels while GC's formulation guarantees a unique optimal solution in the case of two labels only and is further restricted in the type of energy functions it can minimize [9].

4. Graph Construction in CrossTrack

One of the main contributions of this paper is the robust graph-based segmentation. In this section, we explain the graph construction aspect of it. The main idea is to use a unified graph that combines multiple views and multiple time instances. Such unified graph construction makes segmentation robust by fusing all available information. Moreover, in some cases, as we further explain below, it helps in recovering from tumor disappearance.

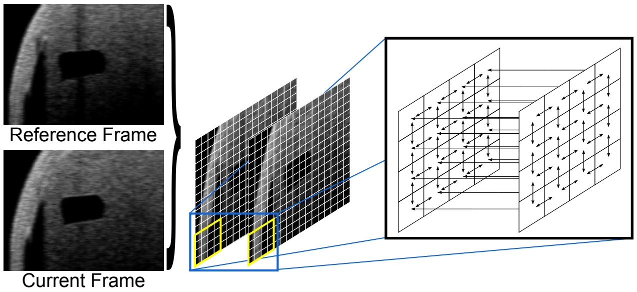

Figure 3. Dual-frame graph construction.

In supervised segmentation algorithms, such as GC and RW, the user must designate at least one pixel belonging to each label. These pixels, and their corresponding nodes in the graph are called *seeds*. The segmentation algorithm is guaranteed not to change the label assignment for seeds. In a tracking framework, seeds can be determined in the first frame given the tumor 3D model and its initial 3D location. The challenge is in finding seeds for each upcoming frame. In this section, we explain one solution to this problem. A complementary solution is explained later in Sect. 4.3.

4.1. Combining Frames Across Time

Despite the incremental change of the segmentation boundary over time, it is not straightforward to use the segmentation result of one frame to harvest seeds for the next. In order to ensure availability of seeds without enforcing erroneous label assignments, we segment two frames together. One frame is the new frame, and the other is a reference frame. Namely, we construct the segmentation graph with two parallel grids, one for each frame. Each pixel in one grid is connected to the corresponding one in the other grid, Fig. 3. For now, we consider the reference frame to be the preceding frame in time. We assume the preceding frame has both foreground and background labels in its segmentation. In Sect. 5 we handle the general case. The labeled pixels of the reference frame serve as the seeds of the combined graph. By using labeled nodes in the reference frame as seeds, no node in the new frame is forced to take a particular label (unless a strong evidence avails, Sect. 4.3). Nevertheless, due to the similarity between the two frames, the segmentation can still produce meaningful results. This dual-frame graph construction is useful also to maintain temporal consistency in the segmentation.

4.2. Combining Multiple Views

Thus far, we have not used the fact that the two ultrasound sequences in our setup correspond to two intersecting planes. The images corresponding to the two planes should have similar intensity values along the lines of intersection and should be segmented consistently along these lines. To make use of this fact, we use a bi-plane graph construction. In this construction, each plane is represented as a grid graph and the two grids are connected along the intersection lines. In addition to maintaining segmentation consistency between the two planes, such construction is also useful in some cases of tumor disappearance, as explained in Sect. 5. Note that the dual-frame construction, Sect. 4.1, is used with the bi-plane construction presented here. However, the bi-plane connections are made only between the grids corresponding to the new frames since the reference frames may correspond to different time instances, as explained in Sect. 5.

4.3. Probabilistic Priors From 3D Model

Thus far, our graph construction did not make use of the tumor 3D model. In this section, we show how to use the tumor 3D model to harvest seeds for a new frame (in addition to the seeds in the reference frame), construct a probabilistic prior, and use it to compute a robust link weighting function.

4.3.1 Volume Prior

We use the tumor volumetric shape to obtain a probabilistic prior for foreground segmentation. Knowing the current 3D tumor location, we generate hypotheses for the segmentation mask corresponding to hypotheses of the next 3D tumor location (sampled particles for tracking). Each segmentation hypothesis is a binary mask with value 1 assigned to foreground pixels and 0 assigned to background pixels. We then compute the average of these masks to obtain a probabilistic prior. We call this prior the *volume prior*, and denote it as p_{V}.

$$p_{\mathrm{V}}(v) = \sum_i h_i(v) \ , \tag{2}$$

where h_i is the i's segmentation hypothesis. The volume prior contains useful information for enhancing the accuracy of the segmentation. Some pixels will have saturated probability values (either 0 or 1) in p_{V}. These are the pixels that have the same label in all hypotheses. Such pixels are used as seeds in the new frame, in addition to the seeds of the reference frame in the dual-frame graph construction, Sect. 4.1. Figure 4 illustrates the process of computing the volume prior and harvesting seeds from it.

4.3.2 Appearance Prior

Using the seeds from the volume prior, we create another probabilistic prior based on the foreground and background appearances. We use the intensity values at the foreground and background seeds to learn two probability distributions for the foreground and background intensity values, denoted as f_{fg} and f_{bg}. For ultrasound images, we use a Gaussian Mixture Model, with the number of modes empirically set to 2 for f_{fg} and 1 for f_{bg}. From these intensity distributions, we construct another probability map, the *appearance prior*, p_{A}.

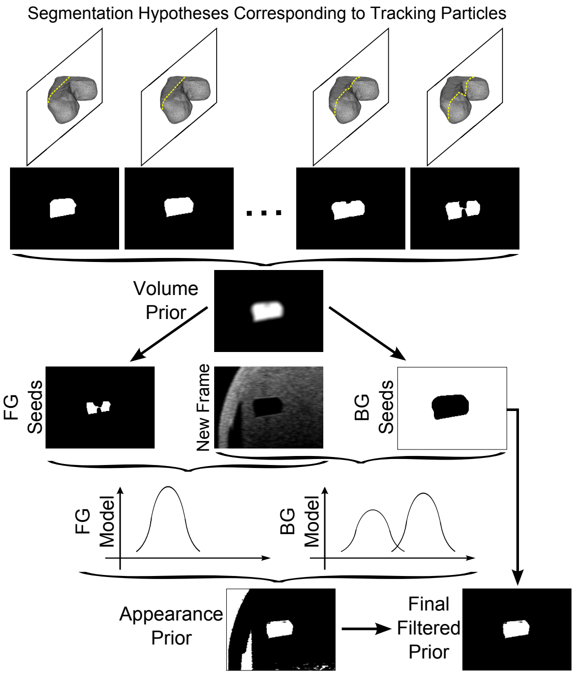

Figure 4. Constructing volume and appearance priors.

$$p_{\mathrm{A}}(v) = \frac{f_{\mathrm{fg}}(v)}{f_{\mathrm{fg}}(v) + f_{\mathrm{bg}}(v)} \ . \tag{3}$$

As shown in Fig. 4, the appearance prior may assign high foreground probabilities to pixels belonging to the background because of their similarities to the foreground distribution. However, based on the information in the volume prior, most of such pixels cannot be foreground. Therefore, we finally combine the two probability maps in a unified prior, p_{fg}, such that

$$p_{\mathrm{fg}}(v) = \phi\left(p_{\mathrm{V}}(v)\right) p_{\mathrm{A}}(v) \ , \tag{4}$$

where $\phi(x)$ is a step function that takes the value 1 if $x > 0$, and 0 otherwise.

4.4. Robust Link Weighting Function

Both GC and RW can be formulated in a way to incorporate priors in the singleton terms in (1). Grady [4] showed that including a probabilistic prior in GC and RW are closely related. In the case of two labels, they are both equivalent to adding a seed auxiliary node for each label, linking each auxiliary node to all nodes in the original graph, and setting the weight for such links to prior probability values multiplied by some constant ν. Properly setting the multiplicand ν is critical to adjusting the effect of the prior with respect to smoothness of the labeling function. Too high ν value would make the result a mere thresholding of the prior while too low value would make the prior ineffective. We argue that incorporating the probabilistic prior in the pair-wise energy terms in (1) is more intuitive to adjust and could make the segmentation algorithm more robust.

For intensity images, such as ultrasound images, the link weighting function is typically formulated as

$$w\left(e_{ij}\right) = e^{\alpha_I (F(v_i) - F(v_j))^2 + \alpha_D \|v_i - v_j\|} \quad , \tag{5}$$

where F is intensity map of the input image, $\|v_i - v_j\|$ is the Euclidean distance between the two vertices v_i and v_j, and α_I and α_D are two constants to adjust the relative importance of the two terms. The weighting function indicates how similar two pixels are, and hence, how likely they are to belong to the same segment. We introduce the incorporation of priors in the link weighting function (5). The underlying rationale is that two pixels are more likely to have the same label if their prior probabilities are similar. Therefore, the difference of prior probability values can be used in the weighting function. If the prior foreground probability map is p, we use the weighting function

$$w\left(e_{ij}\right) = e^{\alpha_I (F(v_i) - F(v_j))^2 + \alpha_P |p(v_i) - p(v_j)| + \alpha_D \|v_i - v_j\|} \quad . \tag{6}$$

where α_P is the weight of the prior-based distance. For a general K label segmentation, the absolute value difference in (6) can be replaced by another distance measure, such as the χ^2 statistic. Note that these changes to the weighting function still respect the regularity conditions required for GC [9].

The benefit of using the weighting function formulation in (6) is multi-fold. First, incorporating the prior with the intensity difference in a single formula makes it easier to adjust their relative weights to get the desired segmentation. Second, in noisy images, such as ultrasound, intensity difference is not a reliable similarity measure. Incorporating the prior probability difference in the similarity measure makes it more robust to image noise. Third, the segmentation becomes less sensitive to the weight given to the prior because the same prior value for a pixel is used multiple times, twice the number of neighbors, and is given a different weight each time.

5. Occlusion Handling

Occlusion handling is a main component in any tracking algorithm in the visible domain, *e.g.* [6]. The tracked object may become briefly invisible due to an occluding object in the scene, or due to lying outside the camera's field of view. The tracking algorithm has to detect such events and resume tracking afterwards. In ultrasound imaging, an equivalent phenomenon can happen when the tracked organ falls into the shadow caused by a highly reflective tissue interface. Another scenario is when the tracked organ moves so that it no longer intersects the ultrasound scanning plane. This case needs a special handling since the foreground region will be lost completely for a period of time.

We consider only respiration-induced motion, which is the most significant un-voluntary motion in the body. Fortunately, respiratory motion is highly periodic. Therefore, it is expected to cause an organ to approximately move along a fixed closed trajectory. For simplicity, we assume the organ to be approximately moving back and forth along a fixed path. Therefore, when a tracked organ moves off the scanning plane, it is expected to come back and intersect the plane at a location that is close to when it was last detected. Recall that in our algorithm, we use dual-frame graph construction that contains a reference frame along with the current frame to segment, Sect. 4.1. The reference frame has to be as close as possible in its appearance to the new frame to be helpful for the segmentation. Given the periodic motion pattern described above, considering the frame in which an invisible tumor starts to appear again, the closest frame in appearance would be the last frame in which the tumor was visible, i.e. right before leaving the field of view. Therefore, this frame can be the best choice as a reference frame for segmentation. It remains to set a criterion to determine whether the tumor is visible or not. When the tumor becomes invisible, the segmentation is expected to label no pixels as foreground. We rely on this desired segmentation behavior for detecting invisibility of the tumor.

In summary, invisibility of tumors due to respiratory motion can be effectively handled using the following simple update rule for the reference frame: *If the segmentation result for the current frame has a non-empty foreground region, update the reference frame to be the current frame. Otherwise, keep the reference frame as it is.* In practice, we stop updating the reference frame when the foreground region is smaller than a specific threshold (100 pixels in our implementation). This approach is illustrated by sample frames from one of our test videos in Fig. 5.

It is also worth noting here that the bi-plane graph construction, Sect. 4.2, helps in some situations when the simple occlusion handling scheme explained here does not. If the foreground seeds in the reference frame of one plane do not overlap with the tumor's boundary upon returning to view in that plane, the tumor can be missed. However, if the tumor is still visible in the other plane, seeds from the other plane can be of great help due to the interconnections between the two.

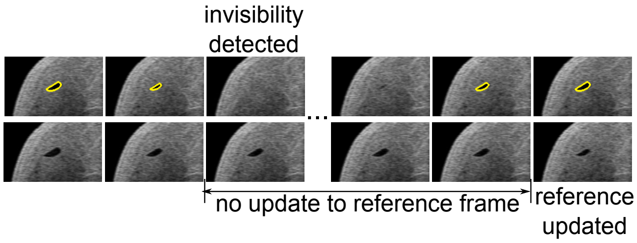

Figure 5. Occlusion handling. Top: segmentation result. Bottom: reference frames. Occlusion is detected and reference frame is not updated in third frame from left. Update is resumed in the right most frame.

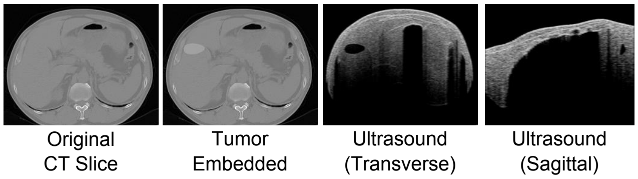

Figure 6. Data generation from a simulated 4DCT volume.

6. Experimental Results

In order to evaluate the performance of CrossTrack, we need ground truth data where the tumor 3D volume is known and the 3D location of the tumor corresponding to each pair of ultrasound frames is also known. Unfortunately, such data is extremely hard to obtain for real ultrasound data. Therefore, we evaluated CrossTrack on synthetic data obtained by simulation from a CT volume. We evaluate the algorithm based on the error in the estimated 3D location.

Given a CT scan, we first embed a volume at a given location to represent the tumor. The Hounsfield Unit (HU) values for the CT voxels covered by the embedded volume are randomly generated from a Gaussian distribution whose mean is slightly higher than the surrounding tissue with a small standard deviation. Next, we simulate the breathing motion and generate a 4DCT sequence. For each time instance, two ultrasound slices are simulated from the corresponding CT volume. The locations of the two slices are fixed. They are taken in the transverse and sagittal directions, and set to intersect at initial tumor location. The process is illustrated in Fig. 6 with sample images. Details of the simulation are beyond the scope of this manuscript. Interested readers are referred to equivalent simulation work [13, 17].

In our experiments, the tumor volume was generated by combining one or more of the basic shapes shown in Fig. 7. The breathing cycle length was set to 2 seconds[1]. Each sequence is 7 seconds with 30 fps. From a single CT scan, we generated 6 4DCT sequences while changing the tumor location and shape to create different levels of difficulties.

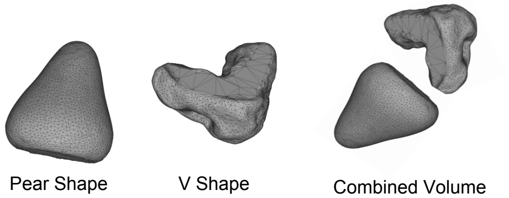

Figure 7. Making complex tumor shapes from basic ones.

In all the sequences, the tumor is placed in the abdomen area. The sequences cover the cases of continuous visibility in both planes, short and long tumor disappearance (8 and 19 frames, respectively) in each breathing cycle in one view only, and tumor disappearance in the two planes at the same time. In one sequence, the V shape volume is rotated so that the transverse plane cuts through the two arms of the V, which creates a disconnected tumor cross section. In another sequence, the tumor consists of two disconnected volumes. Figure 8 shows sample frames from the last two sequences.

We base our comparisons on quantitative measures for 3D tracking performance. We use the Euclidean distance (D) between the estimated 3D location of the tumor center and the ground truth location to represent the tracking error. The distance here is in terms of ultrasound voxels. To evaluate occlusion handling, we use the *precision* (P) and *recall* (R) metrics. To compute these metrics, we consider the tumor existing at a time instance if it is visible in at least one of the two planes, and non-existing if it is invisible in both. For each sequence, the P and R metrics and the average of the D metric over all frames are computed. The comparison is based on the average metrics over all sequences.

Table 1 shows the overall performance of our algorithm using both GC and RW as the base segmentation algorithms. The two algorithms are very close in terms of precision and recall while scoring more than 98% on average for both metrics. This highlights the robustness of the algorithm against different challenging cases of tumor disappearance. In terms of the distance measure, the RW delivers more than double the accuracy of GC, where RW's average error is less than a voxel, while GC's is 2.16 voxels. RW's subvoxel accuracy indicates that it is limited only by the image resolution and hence can deliver the maximum possible accuracy for a given imaging device. We believe that the difference between RW and GC is due to the tendency of GC to favor smaller foreground regions [15]. Since our goal is not to compare RW to GC, we just include the better performer, RW, in subsequent experiments.

Next, we show the significance of using bi-plane imaging for tracking over using a single plane. To conduct this experiment, we used our algorithm on one plane at a time. That is the segmentation is performed in one plane, and the 3D location is estimated by matching the segmenta-

[1] Slightly faster than average, but, more challenging.

	Distance	Precision	Recall
Random Walks	0.98	100.00	98.42
Graph Cuts	2.16	99.31	98.02

Table 1. CrossTrack's performance using Graph Cuts and Random Walks.

.

	Distance	Precision	Recall
CrossTrack	0.98	100.00	98.42
Single Plane	3.32	99.90	86.11
Baseline	5.80	100.00	54.98

Table 2. Performance comparison of CrossTrack vs. single plane and a baseline tracking algorithms.

.

tion boundary to the hypothesized segmentations. The second row of Table 2 shows the results for this experiment. Clearly, using bi-plane imaging outperforms single plane imaging. Using bi-plane imaging reduces the error by 70% and increases the recall by 14%. The tracking performance using a single frame is not acceptable for clinical deployment while the performance for bi-plane is excellent. Figure 8 shows the segmentation output using a single plane compared to bi-plane images. Especially, in Seq-5, top 4 rows, segmentation based on a single plane easily confuses the foreground with the background when the tumor falls under the shade of another organ and when it splits in two regions. Using bi-plane in our algorithm accurately handles this challenging case.

Next, we compare CrossTrack to the prior art. To the best of our knowledge, there is no close prior work to compare to. Nevertheless, we compare the tracking result using CrossTrack against RW with probabilistic priors. As shown by Grady [4], incorporating probabilistic priors in RW alleviates the need for seeds and enables segmentation of fragmented regions if no seeds available within each connected component. We consider a hypothetical baseline solution to our problem that uses RW with probabilistic priors as follows. From the initial frame, two probability distributions are estimated for the background and foreground intensity values. This should be enough in our data since the intensity distribution does not exhibit severe changes. These distributions are used to estimate an appearance prior for each subsequent frame, as in Sect. 4.3.2. The resulting segmentation is matched to the tumor model to estimate the 3D locations. Third row of Table 2 shows the results for this experiment. Clearly, the baseline algorithm fails, with six times higher tracking error, and only 55% recall. Despite the baseline algorithm's ability to handle disappearing tumors in the absence of seeds, due to similarity of the tracked region and a large area of the background, it erroneously segments this area as foreground leading to a significant tracking error. Figure 9 shows sample segmentation outputs for CrossTrack compared to this baseline algorithm on our least challenging sequence.

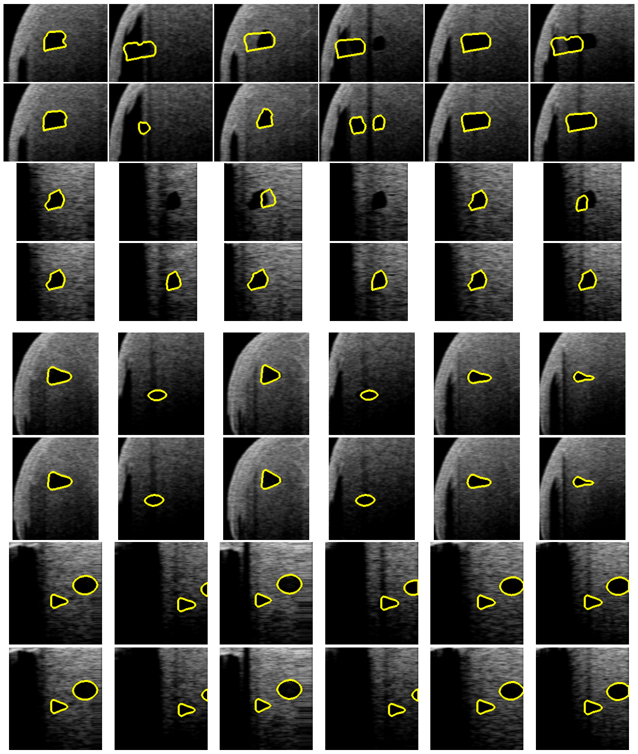

Figure 8. Sample frames from Seq-5 (top 4 rows) and Seq-6 (bottom 4 rows) in our dataset. Rows 1 and 3: tracking using single transverse plane. Rows 2 and 4: corresponding output using bi-plane. Rows 5 and 7: tracking using single sagittal plane. Rows 6 and 8: corresponding output using bi-plane. First two rows of each seq. are transverse plane, last two are sagittal.

7. Conclusion

We presented CrossTrack, a novel approach for recursive 3D location estimation and segmentation in multiple cross-sectional views of a volumetric object. The target application is 3D tumor tracking from bi-plane ultrasound imaging given the tumor 3D model. CrossTrack uses the 3D model and last estimated location to create priors for next frame, and to refine the segmentation result afterwards. Segmentation is performed jointly on two views and two time instances to ensure temporal and view consistency and to achieve robustness against image noise and occlusion. Tumor occlusion is handled in a simple way using the periodic nature of respiration-induced motion. Our experiments on synthesized ultrasound data show the effectiveness and robustness of CrossTrack. CrossTrack is general enough to be used with any graph-based segmentation. The performance is reasonably good for both GC and RW, with a significant lead to RW.

CrossTrack indeed delivers excellent performance on

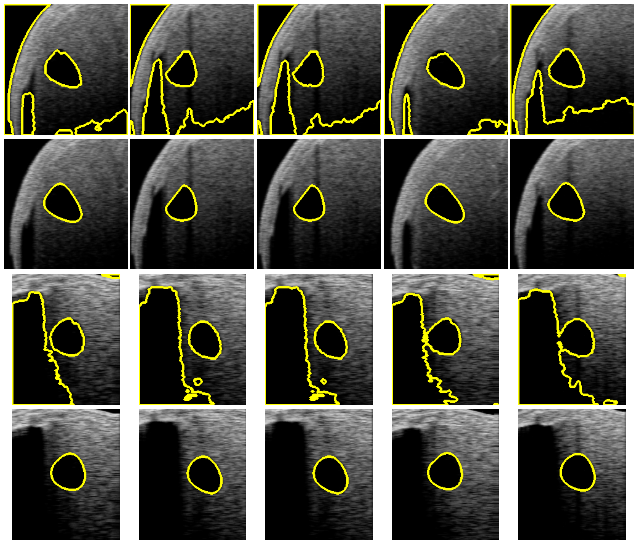

Figure 9. Sample frames from Seq-1 in our dataset. Rows 1 and 3: output of baseline algorithm (see text). Rows 2 and 4: output of our algorithm. First two rows are transverse plane, last two are sagittal.

tracking challenging tumor shapes. However, there are a number of limitations we would like to point out. Typically, the tumor 3D model is obtained by analyzing the patient's CT scan. A model obtained from a CT scan may not perfectly match the appearance in ultrasound due to the many distortion effects in the latter. We believe that converting a 3D model from the CT space to the ultrasound space is possible with a sophisticated physical simulation of the ultrasound imaging process. Another limitation is that tumors with multiple small volumes may violate the motion assumptions we use in recovering from occlusion. CrossTrack can be further enhanced by deploying a richer recursive Bayesian filtering model, and a more general tumor motion model. Finally, the algorithm can possibly be made to run in real time using a GPU implementation since the most time consuming part, segmentation matching, is highly parallelizable.

References

[1] T. Bortfeld, R. Schmidt-Ullrich, W. D. Neve, and D. E. Wazer, editors. *Image-Guided IMRT*. Springer, 2006. 1041

[2] Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary and region segmentation. In *Proc. IEEE International Conf. on Computer Vision*, 2001. 1042, 1043

[3] X. Cufa, X. Muoza, J. Freixeneta, and J. Marta. A review of image segmentation techniques integrating region and boundary information. *Advances in Imaging and Electron Physics*, 120:1–39, 2003. 1043

[4] L. Grady. Multilabel random walker image segmentation using prior models. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005. 1043, 1044, 1047

[5] K. Graham, L. Wirtzfeld, L. MacKenzie, C. Postenka, A. Groom, I. MacDonald, A. Fenster, J. Lacefield, and A. Chambers. Three-dimensional high-frequency ultrasound imaging for longitudinal evaluation of liver metastases in preclinical models. *Cancer Research*, 65:5231–5237, 2005. 1041

[6] A. Jepson, D. Fleet, and T. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(10):1296–1311, 2003. 1045

[7] S. B. Jiang. Radiotherapy of mobile tumors. *Seminars in Radiation Oncology*, 16:239–248, 2006. 1041

[8] P. Kohli and P. Torr. Dynamic graph cuts for efficient inference in markov random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29:2079–2088, 2007. 1042

[9] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26:147–159, 2004. 1043, 1045

[10] T. Lin, L. I. Cervi, X. Tang, N. Vasconcelos, and S. B. Jiang. Fluoroscopic tumor tracking for image-guided lung cancer radiotherapy. *Phys. in Med. and Biol.*, 54:981–992, 2009. 1041

[11] C. Mittelstaedt. Ultrasound as a useful imaging modality for tumor detection and staging. *Cancer Research*, 40:3072–3078, 1980. 1041

[12] D. L. Pham, C. Xu, and J. L. Prince. Current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 2:315–337, 2000. 1043

[13] T. Reichl, J. Passenger, O. Acosta, and O. Salvado. Ultrasound goes GPU: real-time simulation using CUDA. In *Medical Imaging: Visualization, Image-Guided Procedures, and Modeling*, 2009. 1046

[14] X. Ren and J. Malik. Tracking as repeated figure/ground segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007. 1042

[15] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000. 1046

[16] R. Tanaka, S. Mori, M. Endo, and S. Sanada. Volumetric tracking tool using four-dimensional CT for image guided-radiation therapy. *Radiological Physics and Technology*, 1:34–43, 2008. 1041

[17] P.-F. Villard, M. Beuve, and B. Shariat. Lung 4DCT scan generation. In *2nd Workshop on Computer Assisted Diagnosis and Surgery*, pages 47–50, 2006. 1046

[18] O. Wirjadi. Survey of 3D image segmentation methods. Technical report, ITWM, 2007. 1043

[19] L. Xing, B. Thorndyke, E. Schreibmann, Y. Yang, T.-F. Li, G.-Y. Kim, and G. Luxton. Overview of image-guided radiation therapy. *Medical Dosimetry*, 31:91–112, 2006. 1041

[20] L. Yang, B. Georgescu, Y. Zheng, P. Meer, and D. Comaniciu. 3d ultrasound tracking of the left ventricle using one-step forward prediction and data fusion of collaborative trackers. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008. 1041

[21] L. Zhao and L. Davis. Segmentation and appearance model building from an image sequence. In *Proc. IEEE International Conf. on Image Processing*, 2005. 1042