

Efficient Automated Detection and Segmentation of Medium and Large Liver Tumors: CAD Approach

Toshiro Kubota

Mathematical Sciences
Susquehanna University
Selinsgrove PA 17870, USA

Abstract. In this paper, we present a fully automated system that detects and segments potential liver cancer tumors from a thin slice CT data. The system is targeted toward a tumor whose volume is larger than 1cm^3 , and is efficient as the average computation time per volume in our experiment is roughly 3.5 minutes.

The system first reduces the volume size by $4 \times 4 \times 4$ to reduce the computation and memory requirements. It then detects candidate locations as local minima of the intensity fields after a variant of *elastic quadratic* smoothing. It then provides a rough segmentation at each candidate by fitting a plane at sampled points near the periphery of the concave region in the intensity profiles. The rough segmentation is used to estimate the range of intensity values in the tumor, which is used to obtain a more accurate segmentation by a method originally developed for pulmonary nodules. The result of the second segmentation is interpolated at the resolution of the original data.

keywords: *computer aided detection, classification, smoothing, diffusion*

1 Introduction

Automated detection and segmentation of a liver tumor is a difficult image processing problem due mainly to the wide intensity and morphological variations of the tumor. Past works mainly focused on 2D approaches in a semi-automated manner[5, 7, 4, 1].

In this paper, we present a fully-automated detection and segmentation of liver tumors. The work is conducted as a part of 2008 3D Segmentation in the Clinic: A Great Challenge II. The algorithm was developed through 10 tumors in four CT volumes provided for the competition. The data come with manual segmentation as the ground-truth. The small size data set and limited amount of time and resources for development were two major factors in directing our design strategy, which can be stated as follows.

First, the method needs to be training free; We need to be able to set reasonable values to the system parameters based on our 'intuition' rather than the

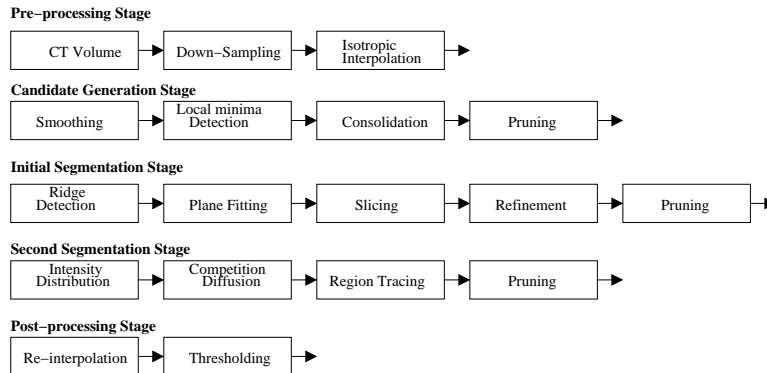


Fig. 1. A system diagram of the proposed method.

given data set. Second, the method needs to be efficient and computationally fast. Third, the system provides a general framework, rather than a solution tailored specific to available data at hand.

Ideally, the system takes in a 3D CT volume and provides a list of segmentations of liver tumors found in the data. This idealistic view, however, is not possible at the current state of our research. Therefore, we allow the system to produce a reasonable number of false positive segmentations. We do not set a concrete requirement of the maximum number of false positives at this time, and leave it our future goal to remove them more aggressively.

In short, the system employs a typical process flow of Computer Aided Detection (CAD) systems, in which a set of candidate locations are generated first and then a more elaborate segmentation step is applied to each candidate location. The segmentation presented in this paper is partially motivated by the segmentation algorithm targeted to pulmonary nodules recently developed by the author and his collaborators[2]. Some adjustments are necessary as liver tumors appear to be more diverse in their shapes and intensity distributions.

2 System

The system consists of 5 stages: pre-processing, candidate generation, initial segmentation, second segmentation, and post-processing. A system diagram of the processing flow is shown in Figure 1.

2.1 Pre-processing

The main purpose of this stage is to reduce the CT data of interest and interpolate it to form an isotropic voxels. First, the data is sub-sampled at $4 \times 4 \times 4$ voxels with a linear smoothing filter. To do so, we employ dyadic decimation twice using a linear separable filter whose 1D component is $[1/4, 1/2, 1/4]$. We chose this smoothing as it can be implemented with integer arithmetic without

multiplication. The goal of this step is two-fold: to reduce the amount of computation and to smoothen the data. A risk associated with the smoothing is a loss of information. As a result, we miss some small lesions in the competition data. We think that we can improve the performance by re-segmenting those small segmented regions at the original resolution. After the sub-sampling, the volume undergoes tri-linear interpolation so that each voxel becomes isotropic. We use the minimum length of the voxel among three dimensions as the new isotropic voxel size. In the following, we call the volume after this pre-processing stage *input volume*.

2.2 Candidate Generation

The purpose of this stage is to collect a number of points that are likely to be inside a tumor. By visual inspection of competition data set, we decided to characterize a tumor as a concave region in an intensity field. Hence, we select local minima of the intensity field as candidate points. To do so directly on the input volume results in a prohibitively large number of candidates. To reduce the number of candidates, we do the following. First, voxels whose intensity values are less than 1000 or larger than 1200 are masked out and cannot become candidate voxels. Second, we apply further smoothing to the input volume. We experimented two smoothing: Gaussian smoothing and a 3D extension of the elastic quadratic smoothing (EQS)[6]. The former is a well-known method of smoothing. The latter is a rather recent technique developed by the author. For brevity, we describe the exact computation of EQS without going into underlying theory and motivation. First, each voxel is represented with the intensity value and six additional attributes. Denote the intensity value by V and the attributes by A, B, C, D, E and F . We then visit each voxel in order and update its value and attributes by the following. In the below, subscripts specify the location of the voxel. Furthermore, $n(i), s(i), w(i), e(i), b(i),$ and $f(i)$ give indices of north, south, west, east, front, and back neighbors of the i th voxel, respectively.

$$\left(\hat{V}^1, \hat{A}, \hat{B}\right) = \left(V_{w(i)}, A_{w(i)}, B_{w(i)}\right) \mathbf{H}^T + \left(V_{e(i)}, A_{e(i)}, B_{e(i)}\right) \mathbf{H} \quad (1)$$

$$\left(\hat{V}^2, \hat{C}, \hat{D}\right) = \left(V_{n(i)}, C_{n(i)}, D_{n(i)}\right) \mathbf{H}^T + \left(V_{s(i)}, C_{s(i)}, D_{s(i)}\right) \mathbf{H} \quad (2)$$

$$\left(\hat{V}^3, \hat{E}, \hat{F}\right) = \left(V_{b(i)}, E_{b(i)}, F_{b(i)}\right) \mathbf{H}^T + \left(V_{f(i)}, E_{f(i)}, F_{f(i)}\right) \mathbf{H} \quad (3)$$

$$\left(V_i, A_i, B_i, C_i, D_i, E_i, F_i\right) = \left(\hat{V}^1 + \hat{V}^2 + \hat{V}^3, \hat{A}, \hat{B}, \hat{C}, \hat{D}, \hat{E}, \hat{F}\right) \mathbf{G} \quad (4)$$

where

$$\mathbf{H} = \begin{pmatrix} -2 & -1 & 1 \\ -1 & 0 & -2 \\ -1 & 2 & -2 \end{pmatrix}, \mathbf{G} = \frac{1}{270} \begin{pmatrix} 25 & -5 & 0 & -5 & 0 & -5 & 0 \\ -5 & 28 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 45 & 0 & 0 & 0 & 0 \\ -5 & 1 & 0 & 28 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 45 & 0 & 0 \\ -5 & 1 & 0 & 1 & 0 & 28 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 45 \end{pmatrix}. \quad (5)$$

The update operation is repeated for five sweeps of the entire volume. Values of V_i give the smoothed input volume. After experiments with limited data, we decided to use the EQS smoothing as it reduces the number of candidates more effectively than the Gaussian smoothing without losing candidates inside tumors in the training data.

It turns out that if we only consider three-dimensional local minimum (i.e. points where derivatives in x, y and z directions all vanish), then we will miss some tumors that are located near the boundary of the liver, due mainly to the partial volume effect. To prevent these tumors from being lost, we first painstakingly collect two-dimensional local minimum points in each axial plane (i.e. points where derivatives in x and y directions vanish). These points are then clustered into groups by the following procedure; for each point, it traces the downhill direction in the intensity field. If the trace reaches another two-dimensional local minimum and if their X-Y coordinates differ by at most 2, then they are clustered into the same group. After the clustering, we choose a representative point from each cluster by the following. If one of the points in the cluster is a three-dimensional local minimum, then the point becomes the representative point. Otherwise, the median-point along the Z-direction becomes the representative point. The former corresponds to a normal case where a tumor region contains a three-dimensional local minimum, and the latter corresponds to a partial volume case. Note that there cannot be more than one three-dimensional local minimum in each cluster.

After candidates are collected, we further reduce the number of them by simple rule based pruning to facilitate the computation. Two pieces of information are used: the location and intensity profiles. Pruning based on the location information is done as follows. First, candidates that are located within 5 slices at the both ends of the volume are pruned. Second, candidates that are located at the bottom-right quadrant in the axial plane are pruned. Pruning based on the intensity profiles is done as follows. First, we prune candidates whose intensity values at the locations are either less than 980 or more than 1170. These cut-off values are set rather conservatively so that we do not remove any true-positive candidates at this stage.

Next, we try to remove local minima that are too subtle visually. To do so, we trace the intensity fields in both west (left) and east (right) directions until it reaches a local maximum. If both local maxima are not sufficiently higher in intensity than the candidate, then the location is pruned. We set 5 as the cut-off value. Thus, for the candidate to be retained, one of the two local maxima has

to have a value larger than that of the candidate location by at least 5. The procedure is repeated for north and south directions, north-east and south-west directions, and north-west and south-east directions. Candidates have to pass all these tests for them to be retained.

2.3 Initial Segmentation

This stage computes a rough segmentation map at each candidate location. The main goal is to extract the core part of the tumor, which will be used to estimate the intensity distribution within the tumor and to consolidate multiple convex regions belonging to a single large tumor. The intensity distribution information is used in the next stage to more accurately delineate the boundary of the convex regions. A secondary goal is to furthermore prune candidates whose intensity profiles or segmentation maps do not meet simple heuristic rules.

First, we estimate the extent of a concave region represented by the candidate by tracing the intensity field from the candidate location. Since the candidate location typically corresponds to the bottom of the concave region, the trace climbs up the intensity field. We stop the trace when it reaches the top of the hill or a local maximum along the trace. We prepare 26 traces in 26 different directions. These directions are obtained by considering directions going into 26 neighbors. After 26 local maxima in the 26 directions around the candidate are obtained, we derive a plane that most closely cut these local maxima in the least square sense. We also derive a bounding box based on the 26 local maxima points. The plane is then used to cut out the convex region within the bounding box.

This method of segmenting a concave region often results in oversegmentation, as the plane equation may go over the top of the local maximum in some direction. We take two measures to reduce the degree of oversegmentation. First, the plane is shifted downward to the mid-way toward the intensity of the candidate location. Second, the cut-off region undergoes a morphological operation to aggressively remove parasitic branches from the segmentation using the following procedure. We visit each location in the segmentation in the order of ascending distance to the candidate location. In other words, the closer the location is to the candidate location, the earlier the location is visited. For each location, we examine its 26 neighbors. If any of the neighbors that are closer to the candidate location are not parts of the segmentation, then the location is removed from the segmentation.

Once the segmentation is obtained at each candidate location, we evaluate its shape and statistical properties to prune it if it is not deemed fit. We describe this part of our method in the following sub-section. The same strategy is applied again to the result of the second segmentation.

Segmentation Fitness Test The test consists of three sub-tests. First two are geometric ones based on approximation of the segmentation by an ellipsoid. The third measures the difference in distribution of intensity values between the

segmentation region and a region surrounding the segmentation. Below, a brief description of each sub-test is given.

– Test 1

We first fit an ellipsoid around the segmentation, compute the shortest and longest axes of the ellipsoid, and compute the "aspect-ratio" by dividing the length of the shortest axis by the length of the longest axis. The segmentation is rejected if the aspect ratio is below a threshold. We set the threshold at 0.09.

– Test 2

We compute the discrepancy between the fitted ellipsoid and the actual segmentation. We first define a hard-ellipsoid as the region where the Mahalanobis distance is less than 2. Then, we count the number of voxels that are within both segmentation and hard-ellipsoid. Denote the number as n_O . We also count the number of voxels that belong to either segmentation or hard-ellipsoid but not both. Denote the number as n_D . Now, we define a discrepancy measure as $1 - n_D/n_O$. If it is less than a threshold, then the segmentation is rejected. We set the threshold at 0.

– Test 3

We compute two metrics: Jensen-Shannon distance and Kolmogorov-Smirnov distance between two intensity distributions: one within the segmentation and the other within a shell of two-voxel thickness around the segmentation. If both Jensen-Shannon and Kolmogorov-Smirnov distances are less than some threshold, then we reject the segmentation. We set the threshold at 0.4.

Consolidation of Candidates A large tumor often has multiple local intensity minima, hence multiple candidates. We conjecture that segmentation maps resulted from these candidates at this stage have significant overlaps. We then explore this idea to group these candidates.

We use the same ellipsoid representation of a segmentation map as described above. We define the core of the segmentation as the region where the Mahalanobis distance is less than 1. We consider a pair of candidates equal if the two segmentation cores overlap. We then form clusters hierarchically based on this equivalent relationship. The results are a number of clusters, each possessing a set of candidates and union of segmentation cores. These clusters become the input to the next stage, which will be described next.

2.4 Second Segmentation

After the previous stage, we obtain a set of candidate clusters. Each cluster has a number of candidates and an associated core region. At this stage, we derive a more accurate segmentation for each cluster.

We apply competition-diffusion, which has been applied successfully to pulmonary nodule segmentation[3], to extract regions whose intensity values are

between minimum and maximum values found inside the core of the cluster. The competition-diffusion provides spatially more homogeneous (i.e. less holes and fragments) segmentation than thresholding, and appears suited to liver tumors as well. For brevity, we only describe the implementation of the method without references to the underlying theory and motivation.

First, each voxel undergoes the following pixel-wise transformation.

$$u_i = \max\left(0, \frac{|v_i - (v_H + v_L)/2|}{v_H - v_L}\right) \quad (6)$$

where v_i is the intensity at i th voxel as before, and v_H and v_L are, respectively, the highest and lowest intensity values found inside the current cluster core. Note that u_i take the largest value of 1 when $v_i = (v_H + v_L)/2$, decreases linearly as v_i deviates away from $v_i = (v_H + v_L)/2$, and capped at 0.

Next we visit each voxel in order and apply the following update.

$$u_i = \lambda \frac{u_i^2}{u_i^2 + (1 - u_i)^2} + \frac{1 - \lambda}{6} \sum_{j \in N_i} u_j \quad (7)$$

where N_i denotes a set of indices for 6-neighbors of i th voxel (i.e. $\{w(i), e(i), n(i), s(i), b(i), f(i))\}$ as in Section 2.2) and λ is a parameter balancing the competition and diffusion terms. The first term is the competition term and the second term is the diffusion term. We apply the update for 4-sweeps of the entire volume. In our experiments, λ is set to 0.5.

After the application of the above competition-diffusion operation, most voxel values (u_i) are near zero or near one. Near one voxels form the foreground and near zero voxels form the background. We draw a threshold at 0.5 to separate the two.

Next, we need to extract a region of interest indicated by a set of candidates from the rest of the foreground. We first compute a distance map from the foreground. Each voxel in the distance map stores the Euclidean distance to the nearest background voxel. Thus, the center of a convex region has a local maximum value in the distance map. From each candidate point, we trace up the distance map until it reaches a local maximum. We then grow a region from the set of local maxima by consuming every voxel adjacent to the current region, if its distance value is strictly less than that of an adjacent region voxel. The growth eventually stops and provides the segmentation result of this stage. We call this particular form of region growing (i.e. to grow in the down-hill direction on the distance map) *region tracing*.

Segmentation Fitness Test 2 Similar to the initial segmentation stage, we can evaluate a goodness-of-fit measure for each segmentation and reject those that are not deemed fit. We derive the same measures as described in Section 2.3 except that we apply different threshold values to segmentations of different sizes. We first categorize each segmentation into tiny, small, medium and large.

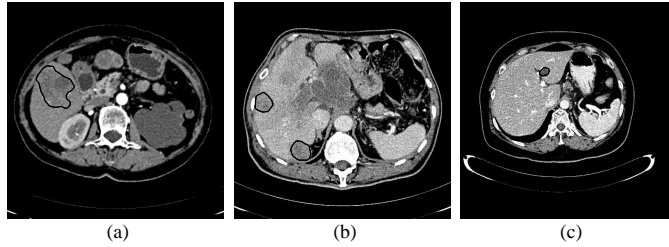


Fig. 2. Examples of manual segmentation.

Tiny segmentations are those with less than 15 voxels. These segmentations are immediately rejected. Small segmentations are those with the number of voxels between 15 and 100. The threshold values for this class of segmentations are 0.5 for both Test 1 and Test 2 and 0 for Test 3. Medium segmentations are those with the number of voxels higher than 100 but less than 20000. The threshold values for this class are 0.2 for Test 1, 0.3 for Test 2 and 0 for Test 3. Large segmentations are those with more than 20000 voxels. These segmentations are immediately rejected.

2.5 Post-processing

Finally, we need to rescale the segmentation obtained in the previous stage to the size of the original CT volume. We simply expand the segmentation by tri-linear interpolation and apply thresholding at 0.5. (Note that the segmentation at the previous stage is represented by 0 and 1 where 1 represents the foreground.)

3 Experiments

This section shows results of the method applied to data provided by 20 tumors found in 10 CT volumes. The 10 CT volumes are divided into a training set with 4 volumes and a test set with 6 volumes. Each set contains 10 tumors. Figure 2 shows some of tumors in the training set with manual segmentation superimposed. Our method is developed using the set.

Figure 3 shows a result at each step of the proposed segmentation process. Figure 3(a) shows an axial slice of the original volume. The slice actually contains two tumors. We show how the fully automated method segment one of them, which will be denoted as IMG04-04 in Table 1. Figure 3(b) shows the corresponding slice after 4x4x4 sub-sampling. Figure 3(c) shows the slice smoothing with candidate points near the slice (within 5 slices away) super-imposed by circles. Figure 3(d) is shows core regions of the 1st stage segmentation. Using the core segmentation of the tumor, we derive $v_L = 1081.1$ and $v_H = 1113.5$, respectively.

Next, we apply the competition-diffusion to the smoothed volume shown in Figure 3(c). Figure 3(e) shows the result of applying (6) to the data. After

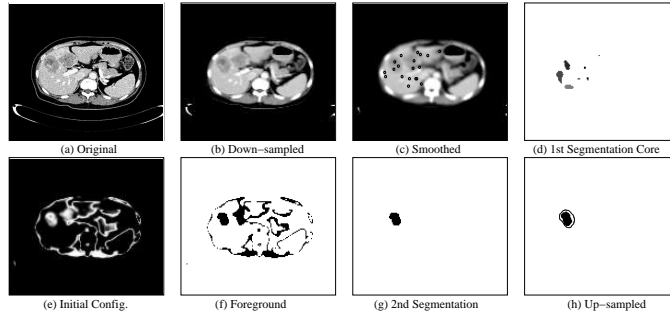


Fig. 3. Intermediate and final segmentation results.

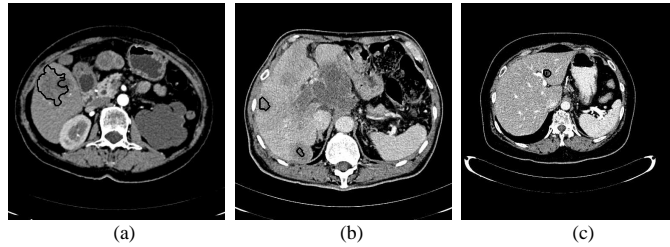


Fig. 4. Results of the proposed segmentation method.

five iteration of the competition-diffusion of (7), the foreground image shown in Figure 3(f) is obtained by thresholding u at 0.5 as described in Section 2.4. The foreground is then subjected to the region tracing as described in Section 2.4. The result is a segmented region of interest as shown in Figure 3(g). Finally, the segmented region of interest is interpolated back to the original resolution. The result of the interpolation is shown in Figure 3(h). The corresponding manual segmentation is super-imposed in the image.

Results of the segmentation method applied to tumors shown in Figure 2 are shown in Figure 4. Compared to the manual segmentation, the automated one tends to be smaller, implying the algorithm tends to undersegment the tumor.

Next, the same method was applied to 6 test volumes. We have been given the location of the tumors but the ground truth segmentation of these nodules are unknown. We then manually selected 10 segmentations from a collection of segmentations obtained by the algorithm. Using these manually selected ones, the same performance measures were computed by the organizer of the Grand Challenge competition. The result also contains scores derived from error measures. The scores are between 100 (best) and 0 (worst). These results are summarized in Table 2.

The algorithm missed completely two nodules: IMG06-L1 and IMG09-L1, which are quite small compared to those in the training set. They are give the total score of 5 by the organizer.

Table 1. Results on training data

Tumor	Overlap Error (%)	Volume Difference (%)	Ave. Surf. Dist. (mm)	RMS Surf. Dist. (mm)	Max Surf. Dist. (mm)
IMG01-L1	41.75	28.10	2.73	3.65	14.23
IMG01-L2	60.13	59.72	2.64	3.00	6.44
IMG02-L1	95.10	95.10	8.52	9.74	16.30
IMG02-L2	78.12	78.12	4.85	5.49	11.48
IMG02-L3	72.63	72.63	3.80	4.19	8.07
IMG03-L1	76.02	153.67	4.24	6.02	15.48
IMG04-L1	53.66	40.93	4.59	5.85	19.95
IMG04-L2	70.41	70.41	4.57	4.89	12.16
IMG04-L3	40.15	32.42	3.28	4.29	20.04
IMG04-L4	50.49	50.42	3.58	4.03	8.82
Average	63.85	68.15	4.28	5.11	13.30

Table 2. Results on test data

Tumor	Overlap Error		Volume Difference		Ave. Surf. Dist.		RMS Surf. Dist.		Max. Surf. Dist.		Total Score
	(%)	Score	(%)	Score	(mm)	Score	(mm)	Score	(mm)	Score	
IMG05-L1	41.03	68	38.97	60	3.32	16	4.33	40	11.07	72	51
IMG05-L2	63.13	51	58.44	39	2.83	28	3.55	51	8.89	78	49
IMG05-L3	58.58	55	15.95	83	2.70	32	3.28	54	8.76	78	60
IMG06-L1	N/A	23	N/A	0	N/A	0	N/A	0	N/A	0	5
IMG06-L2	88.11	32	84.34	12	4.16	0	5.12	29	10.36	74	29
IMG07-L1	45.82	65	38.94	60	6.02	0	7.70	0	27.23	32	31
IMG07-L2	31.85	75	25.42	74	1.79	55	2.43	66	10.34	74	69
IMG08-L1	26.67	79	21.35	78	3.08	22	4.17	42	21.73	46	53
IMG09-L1	N/A	23	N/A	0	N/A	0	N/A	0	N/A	0	5
IMG10-L1	74.56	42	74.56	23	5.92	0	6.22	13	12.14	70	30
Average	53.72	51	44.75	43	3.73	15	4.60	30	13.81	52	38

Finally, Table 3 shows the computation time of the method on each volume. It took between 96 seconds to 408 seconds. Note that the smoothing is done twice: one in the candidate generation stage and the other prior to the 1st stage segmentation stage. The second application is completely redundant and will be removed in the near future. The reason for this redundancy is to allow ourselves to test different smoothing techniques easily in our testbed. Thus, the computation time of the 1st stage segmentation should improve substantially.

The total computation time shown in the last column is the sum of the pre-processing time, candidate generation time, 1st stage segmentation time and the total amount of time spend to apply the 2nd stage segmentation to N_R , the number of regions survived after the 1st stage segmentation. It does not contain the computation time to re-interpolate the segmentation to the original resolution.

Table 3. Computation time of the method per volume

	N_C ¹	N_R ²	N_S ³	Preproc. (sec)	Cand. Gen. (sec)	1st Stage (sec)	Average 2nd Stage (sec)	Total (sec)
IMG01	155	74	31	9.9	9.7	18.8	2.5	225
IMG02	220	96	32	7.8	12.2	25.5	2.9	325
IMG03	90	42	15	12.9	8.9	16.1	3.7	191
IMG04	99	46	12	7.7	10.0	19.0	3.6	204
IMG05	529	110	22	8.7	14.3	69.0	2.9	408
IMG06	77	45	15	7.4	6.4	9.0	1.7	98
IMG07	199	82	23	6.6	10.0	23.3	2.6	255
IMG08	139	56	14	10.3	10.8	15.7	2.4	172
IMG09	190	68	18	6.8	10.2	22.7	2.5	213
IMG10	76	35	15	6.6	6.5	9.6	2.1	96

¹ N_C : the number of candidates.

² N_R : the number of regions after the first stage segmentation.

³ N_S : the number of segmentations after the second stage segmentation.

The table also shows the number of candidates at the end of the candidate generation stage (N_C), the number of regions that are retained after the initial segmentation stage (N_R), and the number of segmentations after the second segmentation stage (N_S). According to this combined result (both training and test sets), the sensitivity is $18/20 = 0.9$ as we missed two small tumors (IMG06-L1 and IMG09-L1), and the average number of false positives per volume is $179/10 = 17.9$.

4 Discussion

There are various limitations of the method in the current form. The foremost is that it misses small tumors due mainly to 4x4x4 sub-sampling. The sub-sampling was necessary to facilitate our development time and compensate limitation of the computing resources. Once the algorithm becomes stable, we can reduce the sampling rate, and make the system applicable to small tumors with some minor changes to the parameter settings. Probably, a better approach is to do in a multi-resolution fashion. At a coarse resolution, we detect non-small tumors with the current method while extracting suspicious regions where small tumors may exist. We can then inspect these regions further at the original resolution.

Another limitation is that the current form reduces false positives in a highly heuristic way. The rules are derived from a small number of training data set, and may not be applicable to unseen data. However, once we can segment tumors accurately, then more effective filtering of false positives is possible.

We should be able to improve the results shown in Tables 1 and 2 substantially with parameter tuning and additional post-processing. For example, some results contain holes in their segmentation, which increase the surface difference error. A simple post-processing can remove these holes. The results show that the automated method consistently produces smaller segmentation than the manual

one. To increase the size of the segmentation, we can increase the amount of diffusion in the second stage segmentation, shift v_L and v_H down and up, use the convex-hull of the segmentation, or a combination of these.

Finally, we use a simple tri-linear interpolation to scale up the segmentation to the original volume size. It may be possible to improve the accuracy by applying the same segmentation algorithm of Section 2 to the original data around the area of the interpolated segmentation. Appropriate amount of smoothing should be introduced to the original data prior to the segmentation process.

5 Summary

We presented a fully automated method for detection and segmentation of liver tumors from a thin-slice CT data. The core of the method is a pulmonary segmentation method developed previously. However, a considerable amount of modifications as well as new additional processing modules were necessary to achieve reasonable performance.

The average total score provided by the Grand Challenge organizer for the proposed method was 38. (See Table 2). We proposed various approaches to improve the performance. (See Section 4). For the system to gain truly a practical value, the number of false positives needs to be reduced substantially. These two issues constitute the major portion of our future work

References

1. C. Krishnamurthy, J.J. Rodriguez, and R.J. Gillies. Snake-based liver lesion segmentation. In *Southwest Symposium on Image Analysis and Interpretation*, pages 187–191, 2004.
2. T. Kubota, M. Jerebko, Salganicoff, M. Dewan, A., and A. Krishnan. Robust segmentation of pulmonary nodules of various densities: from ground-glass opacities to solid nodules. In *The First International Workshop on Pulmonary Image Analysis*, 2008.
3. T. Kubota and K. Okada. Estimating diameters of pulmonary nodules with competition-diffusion and robust ellipsoid fit. In *Computer Vision for Biomedical Image Applications*, pages 324–334, 2005.
4. Y.Z. Li, S. Hara, and K. Shimura. A machine learning approach for locating boundaries of liver tumors in ct images. In *International Conference on Pattern Recognition*, pages I: 400–403, 2006.
5. R. Lu, P. Marziliano, and C. H. Thng. Liver tumor volume estimation by semi-automatic segmentation method. In *27th Annual International Conference of the Engineering in Medicine and Biology Society*, pages 3296–3299, 2005.
6. S. Wang, T. Kubota, J.M. Siskind, and J. Wang. Salient closed boundary extraction with ratio contour. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(4):546–561, April 2005.
7. P. J. Yim and D. J. Foran. Volumetry of hepatic metastases in computed tomography using the watershed and active contour algorithms. In *16th IEEE Symposium on Computer-Based Medical Systems*, pages 329–335, 2003.