Interactive Separation of Segmented Bones in CT Volumes Using Graph Cut

Lu Liu, David Raber, David Nopachai, Paul Commean, David Sinacore,
Fred Prior, Robert Pless, and Tao Ju⋆

Washington University in St. Louis, St. Louis MO 63130, USA
taoju@cs.wustl.edu
http://www.cs.wustl.edu/~taoju

Abstract. We present a fast, interactive method for separating bones that have been collectively segmented from a CT volume. Given user-provided seed points, the method computes the separation as a multi-way cut on a weighted graph constructed from the binary, segmented volume. By properly designing and weighting the graph, we show that the resulting cut can accurately be placed at bone-interfaces using only a small number of seed points even when the data is noisy. The method has been implemented with an interactive graphical interface, and used to separate the 12 human foot bones in 10 CT volumes. The interactive tool produced compatible result with a ground-truth separation, generated by a completely manual labelling procedure, while reducing the human interaction time from a mean of 2.4 hours per volume in manual labelling down to approximately 18 minutes.

1 Introduction

Segmenting bone tissues from CT volumes is a common and important operation in various applications, including the measurement of bone mineral density (BMD). A number of different methodology have been reported for bone segmentation (see a recent survey in [1]). Due to the relatively higher tissue density of bones (especially cortical and trabecular bones) than other surrounding tissues, intensity thresholding [2,3] or edge-detection [4,5] are among the most common approaches in segmenting cortical and trabecular bones in a CT volume. A typical result of segmentation is shown in Figure 1 (b) for a CT scan of a foot in (a), segmented by edge-filtering on each transverse slice [5].

Unfortunately, the segmented CT volume using thresholding or edge detection often cannot be directly used for bone analysis, such as BMD measurement, which requires delineation of the complete boundary for each individual bone. The bones in this segmentation (Figure 1 (b)) often exhibit incomplete interiors, due to bone marrow and the lower density trabecular bone, and neighboring bones are often connected.

A number of methods can be used for filling the bone interior in a segmented CT volume, notably morphological closing followed by contour filling
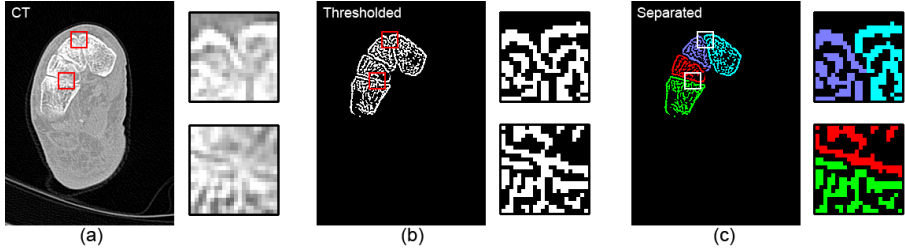

⋆ Corresponding author.

Fig. 1. Bone separation in a segmented CT volume. (a): A transverse slice of a CT volume. (b): The slice after an edge-detection-based segmentation. (c): The slice after individual bones (colored differently) are separated in the segmented volume.

[3] or connected-component analysis [2]. In contrast, methods for separating out each individual bone has been scarce at best. Westin et. al. [2] applied tensor-filtering to the grayscale CT volume before performing intensity thresholding. While the number of connections between bones are reduced using this approach, connections may still exist. Kang et. al. [3] proposed to use a special bone mask customized to the shape of the bone, such as the femoral head. However, creating masks for bones with less distinctive shapes (such as cuneiform) and large inter-subject variability can be as challenging as the separation task itself. In our experience, bone separation in the human foot where many connections exist (see Figure 1 (b)) is a labor-intensive process, even with the help of existing software that provide automated image processing capabilities such as finding connected-components. In our clinical setting, separating 12 bones in a human foot from a CT volume by going through each transverse slice and manually separate voxels belonging to different bones takes an average of 2.4 hours.

In this paper, we propose a fast, interactive method for separating segmented bones using a graph cut approach. Although graph cut algorithms have been successfully employed for interactive object/background segmentation [6] and multiple-objects labelling [7] in grayscale images, the success has largely relied on the change in image intensity or texture on the interface between the object and background or between neighboring objects. These assumptions do not hold for bones in a grayscale CT volume, which exhibit an inhomogeneous texture that is often indistinguishable from the texture at their interfaces (Figure 1 (a)). Here we show how graph cut, when applied to a segmented CT volume represented as a weighted graph, can yield accurate bone separation with a small amount of user input. The key component of this algorithm is a novel graph weighting function that captures the density of connections among neighboring voxels in a binary volume. An example result of our algorithm is shown in Figure 1 (c).

Based on the algorithm, we developed an interactive user-interface for bone separation. The tool simplifies the tedious manual process of bone-labeling on each slice to only placing one or more seed points on each bone. In our experiment with separating the 12 bones in a human foot, we observed that the use of the tool yields compatible results with our previous, completely manual

labelling approach, while reducing the time needed for human intervention from 2.4 hours to approximately 18 minutes per volume. Such reduction is a significant improvement in practice.

2   The Method

We consider a segmented CT volume as a binary volume where each voxel is classified as either an object or background. The goal is to identify disjoint sets of object voxels that belong to each individual bone. Our key observation is that the connections between a bone and its neighboring tissues are usually fewer and sparser than those that constitute the bone itself (Figure 1 (b)). To identify each bone, we represent the collection of all object voxels as nodes in a graph where neighboring voxels are connected by graph edges. Given a set of seed voxels each labelled as one of the bones, we set up and solve a multi-way graph cut problem whose solution is a partitioning of the graph into multiple subsets, one for each individual bone.

   In the following, we first review the multi-way graph cut problem and classical solutions to this problem. Then we present graph construction from a segmented CT volume, and in particular, two ways to improve an initial, un-weighted graph with the goal that the computed cut in the graph is likely to be placed at bone-interfaces. Finally, we present a user-interface that utilizes the algorithm for interactive bone separation.

2.1   The Multi-way Graph Cut Problem and Solutions

Here we consider a weighted undirected graph $G(V, E)$, where V is a set of nodes and E is a set of edges, each connecting two nodes in V and associated with a weight. Given a set $T \subseteq V$ of k *terminal* nodes, a *multi-way cut* is a subset of edges $C \subseteq E$ such that no path exists between any two nodes of T in the residue graph $G(V, E \setminus C)$. The multi-way cut problem aims at finding the cut C with the minimal size $|C|$, computed as the sum of edge weights in the cut. Intuitively, a multi-way cut disconnects all terminal nodes in the way that involves the least amount of change to the graph.

   When $k = 2$, the 2-way cut problem can be solved by a polynomial time method, notably the Ford-Fulkerson algorithm [8]. However, the general multi-way cut problem when $k \geq 3$ is known to be NP-hard [9]. Several polynomial-time algorithms have been proposed to compute a near-minimal cut with varying approximation accuracy [9,10,11]. In our implementation, we adopt the combinatoric isolation algorithm of Dahlhaus *et. al.* [9] that computes the multiple-way cut as the union of 2-way cuts, each disconnecting one terminal node from the others. The heuristic, while simple, achieves a provable $2 - \frac{2}{k}$ approximation ratio. To compute each 2-way cut, we adopt an efficient implementation of the Ford-Fulkerson algorithm [12] that maintains a breadth-first search tree.
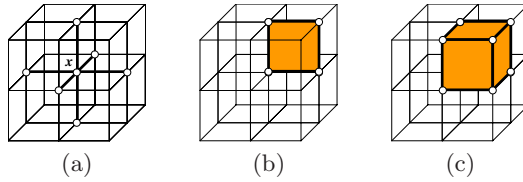
Fig. 2. (a): The 6 neighbors of a voxel x. (b): A voxel face. (c): A voxel cell.

2.2   Multi-way Graph Cut for Bone Separation

To construct the graph from a segmented CT volume, we create one node for each object voxel and connect two nodes representing neighboring object voxels by an edge. We adopt the classical 6-connectivity from digital topology [13], where each 3D voxel may have maximally 6 spatial neighbors (as illustrated in Figure 2 (a)). To apply the multi-way cut, we ask the user to provide a small number of object voxels (called seeds), each labelled as one of the k desired bones. Note that for each bone, the user may specify more than one seeds with that label (which are useful for dealing with noisy inputs, see Section 3). For each label $i \in [1, k]$, we create one terminal node t_i and connect it to all seeds with that label (Figure 3 (a)). Denoting the graph as $G(V, E)$ and the multi-way cut as C, voxels belonging to the ith bone are identified as the set of non-terminal nodes in V that are connected to t_i in the residual graph $G(V, E \setminus C)$ (Figure 3 (b)).

In order for multi-way cuts to be placed at bone-interfaces, the initial graph structure needs to be further adjusted and weighted. Below we present two specific modifications to the initial, un-weighted graph constructed above:

Seed expansion: Note that if only a single seed voxel is provided for each bone, the resulting cuts will most likely enclose just the seed itself due to the small number of edges (≤ 6) connected to a voxel. To resolve this problem without requiring the user to specify a large number of seeds, we automatically expand from a seed voxel x provided by the user to label all voxels connected to x via a path in the graph of length $\leq \alpha$ as seeds of the same label, where α is a user-defined parameter.

Edge weighting: The weighting of the graph edges directly influences the placement of the optimal cut. To ensure that each seed retains its label after the cut,
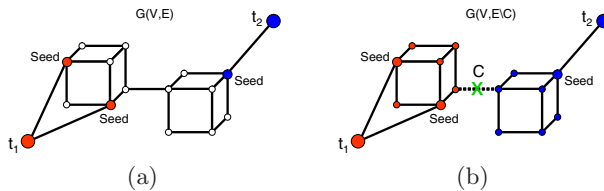

Fig. 3. (a): A graph with 3 labelled seeds (2 red and 1 blue) and 2 augmented terminals t_1, t_2. (b): Removing cut edge C partitions the graph into two labelled components.
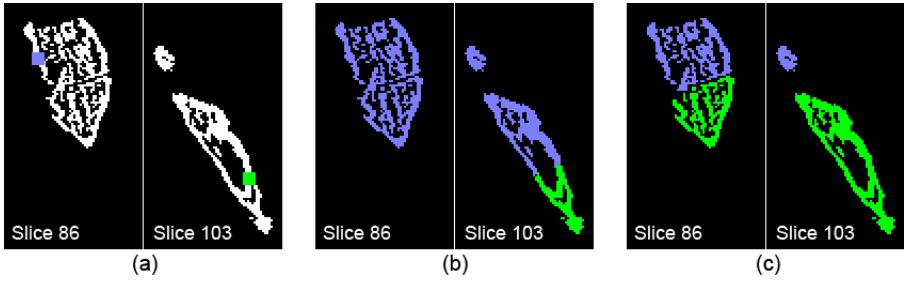
Fig. 4. The effect of edge-weighting on graph cut: (a) The input volume with two seeds on two slices. (b) Separation as a result of graph cut on a uniformly weighted graph. (c) Improved cut using non-uniform weighting that correctly separates the metatarsal 2 (green) from cuneiform 2 (blue).

edges connecting seeds and terminal nodes will be associated with an infinitely large weight. For the remaining edges in the graph that connect neighboring voxels, a straight-forward approach is to associate each edge with a same positive weight. The resulting cuts of this uniform weighting scheme essentially minimize the *number* of edges in the cut, and therefore likely to reside at the bone-interfaces when the connections between bones are few. However, we have observed that it is not uncommon for the number of connections between two bones to exceed those at a cross-section of one of the bones, causing the cut to be misplaced in the middle of that bone, as shown in Figure 4 (b).

For more robust separation, we adopt a different, non-uniform weighting scheme to prevent cuts in the middle of a bone. The key observation is that the voxels in the cortical bone usually form a contiguous, shell-like shape, while connections between bones are typically sparsely located. As a result, a weighting scheme that gives larger weights to edges that are more densely surrounded by other edges would favor the cuts between the bones. In practice, we characterized the local density of edges by the presence of voxel faces and voxel cells, as shown in Figure 2 (b,c). A voxel face is a group of 4 neighboring voxels that form a square, and a voxel cell is a group of 8 neighboring voxels that form a cube. To compute the weight at a graph edge that connects two voxels x, y, we use a simple linear sum $w_0 + w_1 * f + w_2 * c$ of the number of voxel faces, f, and voxel cells, c, that contain both voxels x, y, where w_j are pre-defined parameters. In our tests, the non-uniform weighting scheme effectively avoids cuts through the middle of cortical bones. In the example of Figure 4 (c), the improved cut correctly separates the metatarsal 2 (green) from cuneiform 2 (blue).

2.3   Interactive Tool

Using the proposed method, we have developed an interactive tool for simultaneous separation of multiple bones in a segmented CT volume (Figure 5 left). Given an input volume, the user first places one or more seeds for each bone, by
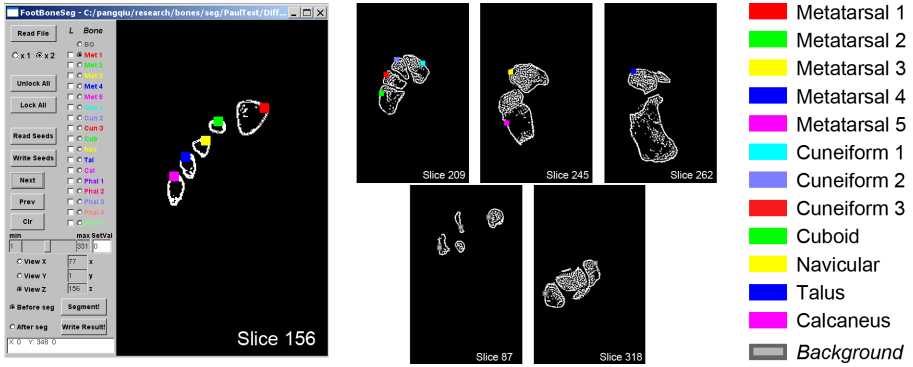
Fig. 5. The interactive tool for bone segmentation (left), where 5 seeds are placed on one transverse slice, other slices (middle) on which the user placed the seeds, and the color utilized to delineate each bone (right)
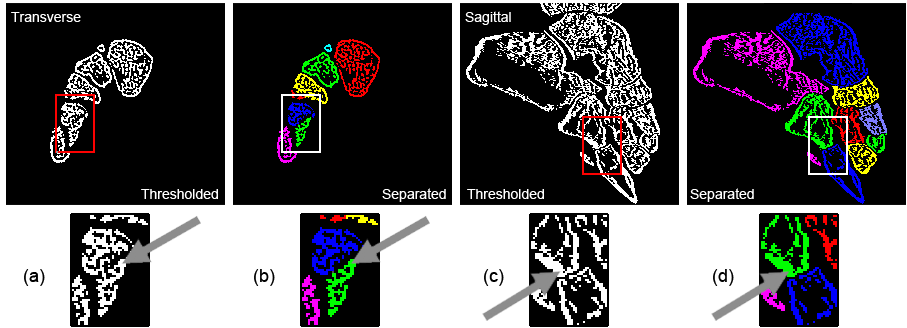

Fig. 6. A transverse slice (left) and a sagittal slice (right) before and after interactive separation using the seeds shown in Figure 5

"painting" the voxel on a 2D slice with a color associated with that bone. The tool then automatically computes a separation between the bones and displays the voxels colored by their corresponding labels (Figure 6 (b,d)). The user can further adjust the result by labelling more voxels on the input volume, and the separation will be updated using the expanded set of seeds.

3   Results and Validation

We tested the method and the graphical tool's ability to separate the 12 human mid-foot bones that have been collectively segmented from CT scans. The 12 bones and their corresponding color labels are shown on the right of Figure 5. We used parameters $\alpha = 8$ and $\{w_0, w_1, w_2\} = \{1, 4, 16\}$ for graph construction. These parameters were selected empirically using one CT volume, and used for all volumes. All tests were performed on a PC with 2GHz CPU and 2GB memory.
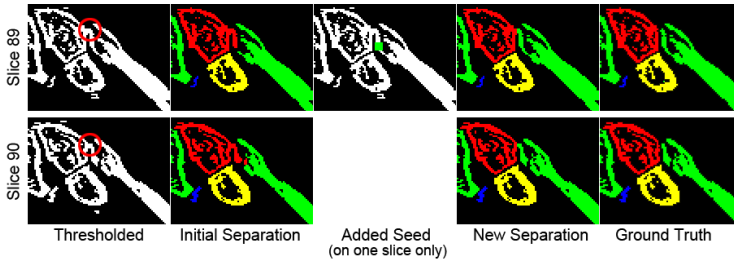
Fig. 7. The tool allows the user to interactively add new seeds when the initial separation is unsatisfactory. See Section 3.

We found that a reasonable separation of the majority of the bones can be achieved in most volumes by using only 1 seed per bone plus a number of background seeds (gray-colored in Slice 87 and 318 of Figure 5) that indicate the non-bone tissues. The top images in Figure 6 show the separation result, viewed from a transverse and a sagittal slice, using just the seeds shown in Figure 5. In the close-up views of the bone interface between metatarsal 4 (blue) and cuboid (green) at the bottom of Figure 6, the arrows on the transverse and sagittal slice point to the same voxel in 3D. While it is difficult even for a human user to determine what bone this voxel belongs to by only examining the transverse slice, the algorithm makes a correct decision by considering the full 3D connectivity.

Figure 7 demonstrates incremental improvement of separation using the tool. Here, the second-to-left column shows two consecutive slices of the initial separation computed from the seeds shown in Figure 5. Note that the cut between metatarsal 2 (green) and cuneiform 3 (red) has been incorrectly placed in the proximal end of metatarsal 2, due to a strong connection between the two bones there in the input volume and the weak connection within the cortical shell for the proximal end of metatarsal 2. After providing one more seed (middle column), the new cut is placed at the desired location (middle-right column).

The tool was tested with two users in 3 different sessions on 10 CT volumes (e.g., one of the users processed the volumes twice at different times), and the results were compared with that of a prior process where a user manually separated bone voxels on each transverse slice (Table 1). The error between an interactively separated volume, I, and the ground truth volume, M, is computed using the Dice's coefficient [14] $1 - \frac{2|I \cap M|}{|I|+|M|}$, where $|I|$ is the number of labelled voxels in I and $I \cap M$ is the number of object voxels that have the same bone label in both I and M. The maximum error among all 10 CT volumes is less than 1.5% with no more than 0.25% differences between sessions. The main source of this error arises from the fact that the separation of neighboring bones can be achieved by either removing the voxels between them (Figure 1 (c) bottom right) or labelling these voxels to be one of the bones (Figure 1 (c) top right). Such choices are automatically determined by the graph-cut algorithm in our interactive tool, but have been made subjectively by the human operator when creating the ground-truth volume.

Table 1. Accuracy of interactive separation of 12 bones in 10 CT volumes performed by 2 users in 3 different sessions (T1,T2,T3), compared to ground truth, slice-by-slice manual separation

ID	Dimension	# Object Voxels	# Seeds T1	Error T1	# Seeds T2	Error T2	# Seeds T3	Error T3	Max Diff.
1	180×352×204	478788	30	0.90%	28	0.82%	29	0.85%	0.08%
2	176×383×215	515706	28	0.78%	28	0.94%	31	0.95%	0.17%
3	169×407×176	463207	25	0.71%	27	0.72%	32	0.68%	0.04%
4	179×373×169	430627	32	0.72%	36	0.85%	44	0.77%	0.13%
5	193×450×153	598157	29	1.32%	31	1.26%	35	1.32%	0.06%
6	193×336×216	480429	40	0.99%	29	0.74%	27	0.91%	0.25%
7	160×361×214	518369	41	0.94%	29	0.99%	32	1.16%	0.22%
8	216×388×177	468017	24	0.74%	29	0.79%	27	0.75%	0.05%
9	184×379×165	807342	42	1.43%	41	1.46%	47	1.30%	0.16%
10	166×447×172	602184	46	1.04%	34	1.20%	43	1.24%	0.20%

The key advantage of the interactive tool versus manual separation is the efficiency, as the user only needs to provide < 50 seed voxels in the whole volume, in contrast to manually separating individual bones on every transverse slice. Using the tool, all 3 sessions took less than 3 hours for all 10 volumes (e.g., approximately 18 minutes per volume), while the manual labelling that generated the ground truth took approximately 1.5 to 3 hours for each volume.

4   Conclusion

We presented a graph-cut method for interactive separation of bones that have been collectively segmented from a CT volume. The core of the method is a novel construction of a weighted graph from a binary volume so that the computed multi-way cuts properly reside on the bone interfaces. The resulting tool has been shown to be effective in segmenting practical data while dramatically reducing the human labor in this process. The method can be used as a pre-process to existing approaches [2,3] that further fill the interior of individual bones.

Acknowledgement

References

1. Wang, L.I., Greenspan, M., Ellis, R.: Validation of bone segmentation and improved 3-D registration using contour coherency in ct data. IEEE Transactions On Medical Imaging 25(3), 324–334 (2006)

2. Westin, C.F., Warfield, S.K., Bhalerao, A., Mui, L., Richolt, J.A., Kikinis, R.: Tensor controlled local structure enhancement of ct images for bone segmentation. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) MICCAI 1998. LNCS, vol. 1496, pp. 1205–1212. Springer, Heidelberg (1998)
3. Kang, Y., Engelke, K., Kalender, W.A.: A new accurate and precise 3D segmentation method for skeletal structures in volumetric ct data. IEEE Trans. Med. Imaging 22(5), 586–598 (2003)
4. Elmooutaouakkil, A., Peyrin, F., Elkafi, J., Laval-Jeantet, A.M.: Segmentation of cancellous bone from high resolution computed tomography images: Influence on trabecular bone measurements. IEEE Trans. Med. Imaging 21(4), 354–362 (2002)
5. Commean, P.K., Ju, T., Liu, L., Sinacore, D.R., Hastings, M.K., Mueller, M.J.: Tarsal and metatarsal bone mineral density measurement using volumetric quantitative computed tomography. Journal of Digital Imaging (accepted, 2008)
6. Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In: ICCV 2001, p. 105 (2001)
7. Grady, L., Funka-Lea, G.: Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. In: ECCV Workshops CVAMIA and MMBIA, pp. 230–245 (2004)
8. Ford, L.R., Fulkerson, D.R.: Maximal flow through a network. Canadian Journal of Mathematics 8, 399–404 (1956)
9. Dahlhaus, E., Johnson, D.S., Papadimitriou, C.H., Seymour, P.D., Yannakakis, M.: The complexity of multiterminal cuts. SIAM J. Comput. 23(4), 864–894 (1994)
10. Călinescu, G., Karloff, H., Rabani, Y.: An improved approximation algorithm for multiway cut. In: STOC 1998: Proceedings of the thirtieth annual ACM symposium on Theory of computing, pp. 48–52 (1998)
11. Karger, D.R., Klein, P.N., Stein, C., Thorup, M., Young, N.E.: Rounding algorithms for a geometric embedding relaxation of minimum multiway cut. In: Proceedings, ACM Symposium on Theory of Computing, pp. 668–678 (1999)
12. Boykov, M.Y., Kolmogorov, M.V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. Pattern Anal. Mach. Intell. 26(9), 1124–1137 (2004)
13. Kong, T.Y., Rosenfeld, A.: Digital topology: introduction and survey. Comput. Vision Graph. Image Process 48(3), 357–393 (1989)
14. van Rijsbergen, C.J.: Information Retrieval. Butterworths, London (1979)