https://doi.org/10.2352/ISSN.2470-1173.2021.1.VDA-331

Volume Data Segmentation Using Visual Selection

Shyh-Kuang Ueng; National Taiwan Ocean University; Keelung City, Taiwan
Hsin-Cheng Huang; National Taiwan Ocean University; Keelung City, Taiwan

Abstract

Segmentation is usually performed in the spatial domain and is likely hindered by similar intensity, intensity inhomogeneity, and partial volume effect. In this article, a visual-selection method is proposed to carry out segmentation in the intensity space such that the aforementioned difficulties are alleviated and better results can be produced. The proposed procedure utilizes volume rendering to explore the input data and builds a transfer function, encoding the intensity distribution of the target. Then, by using this transfer function and image processing techniques, a region of interest (ROI) is constructed in the intensity field. At the following stage, a texture-based region growing computation is conducted to extract the target from the ROI. Experiments show that the proposed method produces high quality results for a phantom which is composed of plates with similar intensities and textures. It also out-performs a traditional segmentation system in separating organs and tissues from a torso CT-scan data set.

Introduction

Segmentation is vital in medical image data processing. Tissue and organ models, segmented from 3D medical image data, are useful for health assessment, injury examination, surgical planning, and artificial implant manufacturing. Segmentation can be carried out in automatic and semi-automatic manners, depending on the degree of human involvement [1]. In an automatic segmentation process, the users give the computer program initial information at the beginning stage. Then, the program relies on predefined mathematical models, built-in pattern classifiers, and embedded anatomic rules to extract the target [2, 3, 4]. In a semi-automatic segmentation procedure, the users may need to create a region of interest (ROI) and place seeds inside the ROI before the computer program can start to extract the target. During the process, the users might have to guide searching directions, set thresholds, and make decisions for the computer program such that high-quality results could be produced [5, 6, 7].

Intuitively, automatic segmentation methods are more convenient than semi-automatic approaches. However, some intrinsic difficulties hinder segmentation processes, for examples, intensity overlapping, partial volume effects, and intensity inhomogeneity [1, 8]. When encountering these obstacles, obtaining feasible parameters for the mathematic models, training the recognizers to detect features, and acquiring precise anatomic rules for the classifiers pose challenges to the users and the automatic segmentation programs. On the other hand, in semi-segmentation procedures, the users can direct the computations on the fly based on their anatomic knowledge and experiences and the intermediate results. The segmented results are usually superior to those produced by automatic algorithms [2]. Hence, we prefer semi-automatic segmentation methods to automatic ones.

When conducting a semi-automatic segmentation task, any prior knowledge about the target is beneficial to the model extraction computation. In the proposed segmentation pipeline, we utilize a visual-selection method to extract prior knowledge about the target, including the intensity distribution, position, size, and shape of the target as well as the nearby structures. These data uncover the geometric and intensity features hidden in the data and help us to make feasible decisions in the subsequent computations.

Volume rendering is a powerful technique for exploring 3D medical image data [9]. We utilize volume rendering to preprocess the input data and generate images of the target and a transfer function, encoding the intensity distribution of the target. The transfer function helps us to classify voxels, remove unwanted parts from the input data, and create a ROI. Consequently, intensity overlapping and geometrical intertwining between the target and other objects can be reduced.

There are many efficient methods for segmenting volume data, for example thresholding, classification [10], morphological operations [11], and region growing [12]. Individually, these techniques are effective for extracting certain types of objects under specified conditions but may fail to generate good results in other situations. Our segmentation pipeline selectively combines these procedures to extract targets such that tissues and organs possessing different characteristics can be successfully segmented.

Overview

The flowchart of the proposed method is illustrated in Figure 1. The entire pipeline is composed of three stages: The aim of the first stage is to explore the input data and generate the transfer function and images of the target. At the second stage, the ROI of the target is constructed. The major computations include the Otsu classification [13] and morphological operations. At the third stage, the system extracts the target from the ROI by using a Seeded Region Growing (SRG) procedure.

We have performed several experiments to verify and analyze the capability of the proposed segmentation pipeline. In the experiments, we use the system to extract structures with similar intensities and textures from a phantom to study how well the system can do to alleviate these hazards. In other tests, the proposed procedure is utilized to segment organs and tissues from a complex torso CT-scan data. These tests are carried out to evaluate the performance of the proposed method under the influences of intensity overlapping, partial volume effects, and intensity inhomogeneity. Furthermore, the proposed segmentation pipeline is adopted to assist layered manufacturing in another set of experiments. We employ the system to extract targets from volume data and fabricate the physical models by using 3D printers. All these experiments are to be presented in this article.

Preliminary ROI Creation

Before starting the segmentation task, we preprocess the input data to uncover the target and its surroundings. This procedure produces 3D images of the target and a transfer function encoding the intensity distribution of the target. The 3D images help us to plant seeds for the region growing segmentation computation. The transfer function offers information for classifying the foreground and background voxels and building a preliminary ROI.

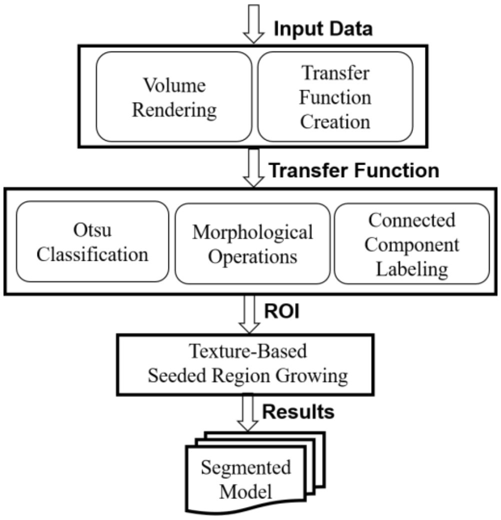


Figure 1. Flowchart of the segmentation pipeline, using visual selection and region growing.

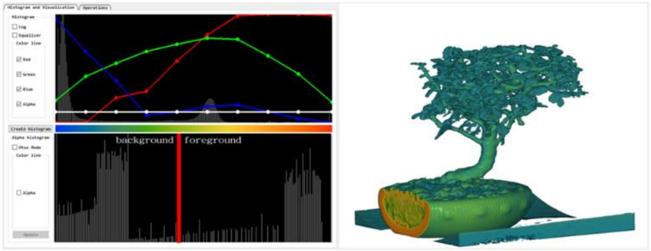


Figure 2. GUI of the volume rendering program, right: volume rendering image, left top: intensity histogram and the transfer function, left bottom: Otsu classification result.

In the proposed segmentation method, we use volume rendering [9] to explore the input data and search for the target. The data exploration procedure is conducted in an interactive way. The graphical user interface (GUI) of this program is shown in Figure 2. This GUI displays the histogram of the intensity field and offers tool-kits for the users to define four splines. These splines constitute a transfer function, which converts voxels' intensities into colors and opacities. We call these splines the R-, G-, B-, and A-spline.

We assign colors to voxels to differentiate structures by using the RGB-splines and give opacities to voxels via the A-spline. The voxels, potentially belonging to the target or being helpful to high-light the target, are given non-zero-opacities while the remaining voxels are associated with zero opacities. Once the transfer function has been defined, the input data is shaded and the resultant volume rendering image is shown in the right part of the GUI. If the results are satisfied, we stop this visual selection process and output the transfer function and volume rendering images. Otherwise, we adjust the RGBA splines via the GUI and repeat the process again.

At the first step, we collect voxels possessing non-zero opacities and form a preliminary ROI. In this article, this preprocessing computation is called the *visual-selection* since the ROI is produced via human visual-perception of the data.

A good ROI should tightly enclose the target such that the searching space is narrowed and the segmentation computation is thus sped-up [14]. Traditionally, ROI construction is carried out in the data space and assisted by users or predefined rules. The related techniques are well-developed and hard to improve. In this research, we offer an alternative way and create the ROI in the opacity field. The process is guided by the transfer function, which has been generated in the visual-selection process. The resultant ROI tightly surrounds the target in the intensity domain and reduces the influences of the nearby background voxels with similar intensities. It confines the target in the intensity space.

An example of the visual-selection process is shown in Figure 3. In this example, the input data is a torso CT-scan data and the target is the right lung. An image of the raw data is displayed in part (a). The target is totally obscured by other structures. We can't locate it, let alone extracting it. After performing the visual-section process, we successfully build a transfer function which filters out many irrelevant voxels and uncovers the target from the data set. One of the resultant images is shown in part (b). In this image, the two lungs emerge from the input data. Their locations and shapes are very comprehensible, though further computations are needed to extract the target.

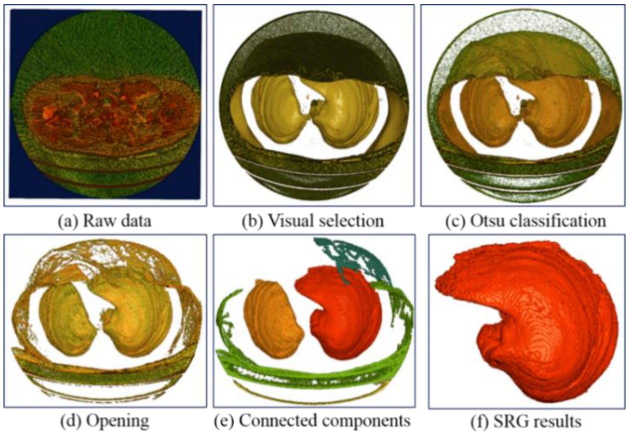

(a) Raw data
(b) Visual selection
(c) Otsu classification
(d) Opening
(e) Connected components
(f) SRG results

Figure 3. The intermediate and final results of the segmentation process.

ROI Improvement

In the visual selection process, the transfer function gives relatively higher opacities to the target's voxels to high-light the target in the resultant images. However, the transfer function may also assign non-zero opacities to other voxels to manifest the target's position, range, and orientation. These voxels enhance our cognition about the target but should not be included in the ROI. Thus, we establish a threshold in the opacity field to exclude more irrelevant voxels.

At first, we quantize the spectrum of the opacity field into k levels and create a table of k entries. In the i-th entry of the table, we record the number of voxels whose opacities are at the i-th level. Assume this number is M_i. Then, we divide M_i by M, the total number of voxels in the input data:

$$p_i = \frac{M_i}{M}. \tag{1}$$

Consequently, this table stores the probability density function of the opacity field. Then, we carry out the Otsu method presented in [13] to classify the opacity field. In the computation, we search for an opacity level *t,* which maximizes the *inter-class* variance defined as follows

$$\sigma_t^2 = \omega_0\omega_1(\mu_o - \mu_1)^2, \tag{2}$$

where $\omega_0, \omega_1, \mu_0$ and μ_1 are by

$$\begin{aligned}\omega_0 &= \sum\nolimits_{i=1}^{t} p_i,\\ \omega_1 &= \sum\nolimits_{i=t+1}^{k} p_i,\\ \mu_0 &= \sum\nolimits_{i=1}^{t} i(\frac{p_i}{\omega_0}),\\ \mu_1 &= \sum\nolimits_{i=t+1}^{k} i(\frac{p_i}{\omega_1}).\end{aligned} \tag{3}$$

The resultant level *t* splits the opacity field into two classes, as shown in the low-left part of Figure 2. (In this image, the gray bars show the probability density function of the opacity levels and the red bar represents the computed *t*.) Then, based on *t*, we keep those voxels whose opacities are greater than or equal to *t* and remove other voxels from the ROI.

Part (c) of Figure 3 shows a volume rendering image of the CT-scan data after undertaking the Otsu classification. Comparing part (b) and (c), it is obvious that more irrelevant voxels have been removed and the lungs are better revealed.

Morphological Operations

After undertaking the Otsu classification, the ROI may contain small dangling parts, narrow gapes, and tiny holes. To remove these noises, we selectively apply closing, opening, and connected component labelling operations [14] on the ROI. The closing operation dilates the ROI first and then erodes it. This operator fills small gaps and holes in the ROI. The opening operator performs an erosion followed by a dilation. It removes small and thin structures from the ROI. We monitor the filtered results by using the volume visualization module and selectively perform opening or closing. Excessive opening or closing may damage the ROI. To undo such mistakes, our system saves a copy of the ROI in a disk file before each operation. Hence, we can always recover the ROI if a morphological calculation degrades its quality.

After the above filtering step, we conduct connected component labelling to divide the ROI voxels into groups. Then, we count the number of voxels in each group. If the cardinality of a group is too small, all its voxels are converted into background voxels and removed from the ROI. Therefore, small dangling parts are deleted.

The ROI, shown in part (c) of Figure 3, contains some dangling parts and thin structures. We perform an opening operation to reduce these noises. The results are displayed in part (d) of the same figure. In the following step, we carry out connected component labelling to group the foreground voxels and delete small clusters. As part (e) shows, the ROI is greatly improved after this operation. It contains five separate clusters, shaded in different colors.

Textured-based Seeded Region Growing

Once the ROI has been constructed, a Seeded Region Growing (SRG) process [15] is triggered to extract the target. At first, one voxel is selected in the ROI by referring the volume rendering images produced in the visual-selection and ROI creation processes. This seed voxel is kept in the target set, which is initially empty. Then, all other ROI voxels, which are connected to the seed, are examined to extract the target. For each of these ROI voxels, the similarity between it and the seed is evaluated. If the similarity exceeds a predefined threshold, this voxel is added to the target set. Otherwise, it is converted into a background voxel.

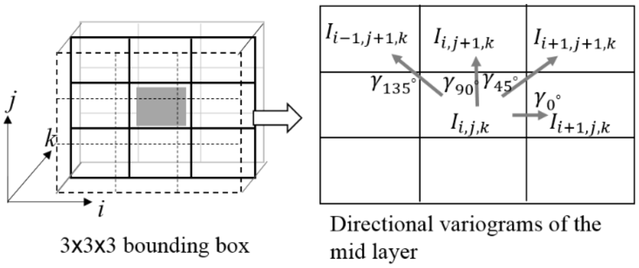


Figure 4. Directional variograms of a voxel (shaded in gray color), left: the bounding box, right: the four directional variograms of the middle layer.

Voxel Similarity Measurement

Similarity between voxels is measured by their semivariograms. The algorithm presented in [16,17] are revised in this research to compute voxels' semivariograms.

Before computing the semivariogram of a voxel, a 3x3x3 bounding box (BB) centered at this voxel is first created as shown in the left part of Figure 4. This BB contains three layers of 3x3 squares. Then, four directional variograms, in 0, 45, 90, and 135 degrees and at distance 1, are computed at the middle layer, as shown in the right part of Figure 4. The directional variogram in the direction θ is calculated by

$$\gamma(\theta) = \frac{1}{N}\sqrt{\sum_{i=1}^{N}\left|I(\bar{x}_i) - I(\bar{x}_i + \vec{h}(\theta))\right|}, \tag{4}$$

where N is the number of voxels having a neighbor in the direction θ within the box, and $I(x_i)$ represents the intensity of the voxel located at position x_i. The vector h is an offset vector decided by θ as follows

$$\begin{aligned}&\vec{h}(0') = \begin{bmatrix}1 & 0 & 0\end{bmatrix}, \vec{h}(45') = \begin{bmatrix}1 & 1 & 0\end{bmatrix},\\ &\vec{h}(90') = \begin{bmatrix}0 & 1 & 0\end{bmatrix}, \vec{h}(135') = \begin{bmatrix}-1 & 1 & 0\end{bmatrix}.\end{aligned} \tag{5}$$

These four directional variograms are combined to produce the semivariogram of the middle layer:

$$\gamma_{mid} = \log\left(\frac{\gamma(0')}{\gamma(90')} + \frac{\gamma(90')}{\gamma(0')} + \frac{\gamma(45')}{\gamma(135')} + \frac{\gamma(135')}{\gamma(45')}\right). \tag{6}$$

Then, the semivariograms of the upper and lower layers, γ_{top} and γ_{low}, are evaluated by using the same procedure. Finally, the semivariogram of the voxel is set to the average of the three semivariograms:

$$\gamma = (\gamma_{low} + \gamma_{mid} + \gamma_{top}) / 3. \tag{7}$$

In part (f) of Figure 3, the segmentation results are shown. The segmented lung is in good quality and free from the surroundings.

Layered Manufacturing

When the target has been segmented, we show its outer appearance and inter structures by using volume rendering. The resultant images are useful in many medical applications, including pathological assessment and surgical planning. As layered manufacturing technologies have made significant progressions, the physical model of the segmented target can be fabricated by using a 3D printer. The printed result is a real model, and the users can touch, measure, and examine it in various aspects. Thus, it is a better media for surgical planning and pathologic assessment than the volume rendering images. In some medical applications, the printed model can serve as an artificial implant in a surgery too.

However, most assistant software of layered manufacturing can only process polygonal models. They are unable to generate G-codes for volume models. Hence, the results produced by our segmentation pipeline cannot be manufactured by using 3D printers. We can transform the results into polygonal representations. Nonetheless, this format transformation may produce extra geometric errors. In recent researches, 3D printing assistant software have been upgraded and are able to generate G-codes for volume models [18, 19]. Thus, the format transformation is unnecessary. The segmented result can be directly manufactured by 3D printers.

Test Results

We implement the proposed segmentation pipeline by using C-language and OpenGL libraries. The volume rendering module is accelerated by the GPU so that the visual-selection and ROI construction processes can be monitored and adjusted in real-time. To verify the effectiveness of our system, three sets of experiments have been conducted. The results and analysis are presented in this section.

Test Data Sets and Evaluation Methodology

Several data sets are employed in the experiments. The first one is a Credence Cartridge Radiomics (CCR) phantom [20]. It is a man-made multi-layer structure, including a plaster layer (layer 1), a shredded rubber particle layer (layer 2), an acrylic layer (layer 3), two cork layers (layers 4 and 5), and one sycamore wood layer (layer 6). These layers are shown in Figure 5. This phantom is used to measure the accuracy of the proposed segmentation method.

Assuming that *X* and *Y* represent the ground truths and the segmented results, the accuracy of the segmentation process is calculated by

$$Accuracy = \frac{\|X \cap Y\|}{\|Y\|}. \tag{8}$$

The similarity between the ground truths and the results is measured by using the Sørensen-Dice Coefficient (SDC) [21], defined as follows:

$$SDC = \frac{2\|X \cap Y\|}{\|X\| + \|Y\|}. \tag{9}$$

The ||.|| operator computes the cardinality of a set. The two metrics defined in Equation (8) and (9) are *1*, if the segmented results match the ground truths perfectly. On the other hand, if *X* and *Y* are totally exclusive, these metrics will be *0*.

The second data set is a CT-scan data of a human torso, containing two lungs, a spine, ribs, a heart, a liver, and other organs and tissues. The input data are stored in a DICOM file [22]. Intensities of these organs and tissues are measured by Hounsfield units (HUs) [23]. Besides our system, we also use the segmentation software ITK-SNAP [24] to extract targets from the CT-scan data. Since this software supports semi-automatic segmentation and its operational procedure is similar to ours, a fair comparison can be made. However, the kernel segmentation methods of these two systems are different. Our program employs the SRG method to generate targets while ITK-SNAP uses a SNAKE algorithm [25] to extract objects.

The third data set is the Bonsai CT-scan data, shown in Figure 2. We segment the tree from the data set and fabricate the physical object by using a 3D printer. Conventional 3D printing takes polygonal representations as input. However, the tree contains a lot of fine structures. It is hard to create a polygonal representation for the segmented object. Instead, we employ a 3D printing software, which is capable of directly converting volume models into G-codes. Thus, the segmented model can be physically built.

Segmentation Accuracy

In the first set of experiments, we segment the six plates hidden in the phantom by using our program. The segmented plates are displayed in Figure 5. These images reveal their shapes and surface patterns. The porosities and textures of these plates are well preserved in the segmentation process.

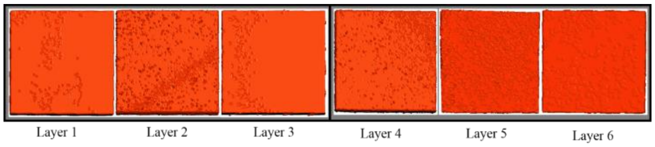


Figure 5. The six layers extracted from the phantom by using our program.

The accuracy and SDC are estimated based on Equation (8) and (9). These metrics are recorded in Table 1. Based on the data of Table 1, we can see that the accuracy and SDC metrics of the segmented plates are high, except the accuracy of plate 4 and the SDC of plate 5. These two plates are adjacent and made with the same material. The visual selection module cannot create clean ROIs for them. By examining Figure 5, we find that the granularities of holes and porous inside them are a little different, and the SRG procedure should be able to separate them. Nonetheless, the semivariograms of the interface voxels between these two plates are almost the same. When extracting plate 4, some voxels belonging to plate 5 are included. Thus, the accuracy of the segmentation is reduced.

Table 1. Accuracy of the proposed segmentation procedure

Layer	*Ground Truth X*	*Our results Y*	$X \cap Y$	*DSC*	*Accuracy*
1	*639,350*	*599,781*	*596,036*	*0.96202*	*0.99376*
2	*612,876*	*605,046*	*586,889*	*0.96375*	*0.96999*
3	*607,509*	*559,173*	*559,164*	*0.95855*	*0.99998*
4	*597,463*	*631,398*	*586,646*	*0.95478*	*0.92912*
5	*628,167*	*534,025*	*530,506*	*0.91234*	*0.99341*
6	*688,154*	*686,084*	*672,871*	*0.97926*	*0.98074*

CT-scan Data Segmentation

In another set of experiments, we use our program to extract three targets from the CT-scan data. Since the ground truths of these targets are not available to us, we utilize the ITK-SNAP program to segment the same targets and visually compare the results. The segmented objects include the lungs, spine (with ribs), and liver. The models generated by our system are volume models. They are displayed by using volume rendering. On the other hand, ITK-SNAP produces surface models. Hence, the results from ITK-SNAP are shaded by using surface rendering. In each test, the extracted object is portrait in four different view angles to reveal its outer-shape. The images of the segmented models are shown in Figure 6, 7, and 8.

In Figure 6, the lungs segmented by both systems are shown. Images of the results produced by ITK-SNAP are contained in the lower row while images of our results are presented in the upper row. Comparing these images, we find that the targets created by both systems are in good quality. Since the intensity distribution of the lungs does not overlap with the surrounding structures and occupies a narrow range in the HU scale, the segmentation processes do not suffer from overlapping intensity and intensity inhomogeneity problems. Thus, both the systems produce decent results, though our resultant model is smoother.

The segmented spines are illustrated in Figure 7. The models created by ITK-SNAP and our program are shown in the lower and upper rows respectively. The intensities of the spine (and ribs) are overlapped with those of the heart and arteries in the HU scale. It is difficult to separate the spine from the arteries based on voxels' intensities. The arteries and heart are included in the model produced by ITK-SNAP. In our system, the opening and connected-component labelling operators successfully separate the spine from the heart and arteries. Therefore, our results do not contain the heart and arteries.

The segmented liver is displayed in the images of Figure 8. The intensities of the liver overlap with the intensities of the surrounding structures. It is difficult to separate the liver from the backgrounds based on intensity contrast. Furthermore, the liver intertwines with the surrounding structures. Thus, it is hard to produce a clean ROI in the spatial domain. However, our system creates the ROI in the intensity space and excludes more surrounding structures than ITK-SNAP. Therefore, the liver produced by our system is much cleaner.

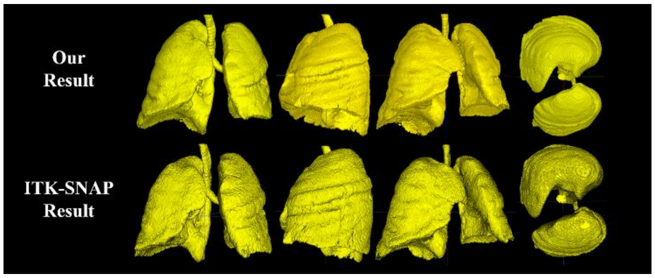


Figure 6. Segmented lungs, upper row: by our program, lower row: by ITK-SNAP.

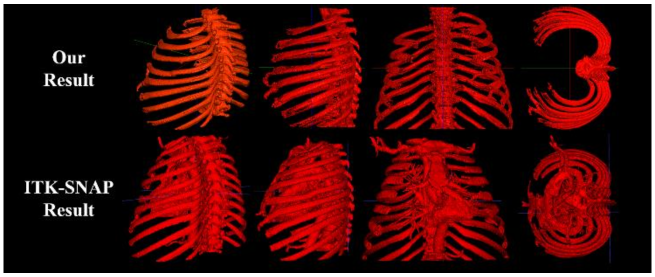


Figure 7. Segmented spine, upper row: by our program, lower row: by ITK-SNAP.

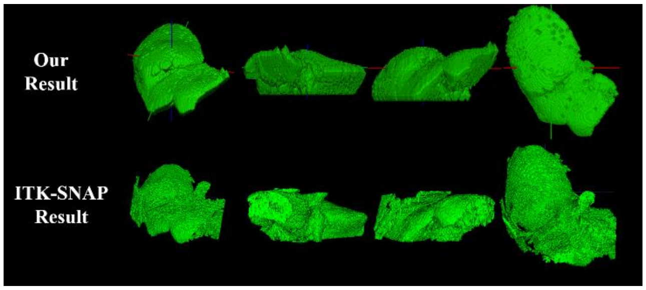


Figure 8. Segmented liver, upper row: by our program, lower row: by ITK-SNAP.

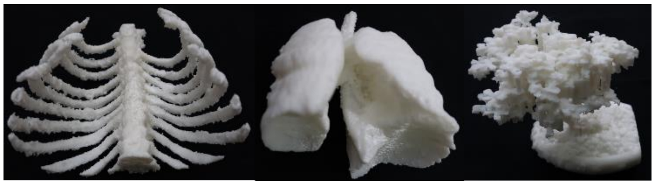


Figure 9. Physical models produced by layered manufacturing.

Physical Model Manufacturing

Besides evaluating segmented objects by using volume rendering, we fabricate them by using 3D printers. The resultant physical models can be used in surgical planning, pathological assessment, and artificial implant surgery [26].

We extract the spline model, the lung model, and the tree model by using the proposed segmentation pipeline. Then, the resultant models are sent to a 3D printing software to generate G-codes. At the following step, the physical models are built by using a Fused Deposition Modelling (FDM) 3D printer. The images of the printed results are shown in Figure 9. Comparing the physical models and the volume rendering images, we found that the physical models retain many details, which are filtered out by the volume rendering process. The physical models improve our cognition about these targets. This phenomenon is especially obvious in the tree model. Thus, segmentation augmented with layered manufacturing enhance the usability of medical image data.

Conclusion

The proposed pipeline utilizes volume rendering to explore the input data. Thus, the density distribution, position, shape, and size of the target can be revealed before the segmentation process starts, and we can make better decisions in the following computations by referring the resultant images. The Otsu classification, performed in the opacity space, and the morphological operators eliminate noises and delete unwanted background voxels such that geometrical and intensity intertwining between objects are alleviated in the ROI. Furthermore, the pipeline employs the texture-based SRG method to annex candidate voxels into the target set, and hence structures possessing similar intensities can be separated. Test results reveal that the proposed segmentation method accurately splits adjacent planar structures with similar intensities. Its performance may slightly decrease if these segmenting structures are not only adjacent but also having similar textures and intensities. In another test, we utilize the ITK-SNAP program and ours to segment tissues and organs from a CT-scan data. The results conclude that our program generates better models in all the tests than ITK-SNAP.

References

[1] D. L. Pham, C. Xu, J. L. Prince, *Current methods in medical image segmentation*, Annual Review of Biomedical Engineering, **2** (2000), pp. 315-337.

[2] T. Heimann, B. van Ginneken, M. A. Styner, et al., *Comparison and evaluation of methods for liver Segmentation from CT datasets*, IEEE Transactions on Medical Imaging, 28 (2009), pp. 1251-1265.

[3] C. Chu, M. Oda, T. Kitasaka, et al., “Multi-organ segmentation based on spatially-divided probabilistic atlas from 3D abdominal CT images,” Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention (Berlin, Germany 2013) pp. 165-172.

[4] M. Prastawa, J. H. Gilmore, W. Lin, G. Gerig, *Automatic segmentation of MR images of the developing newborn brain*, Medical Image Analysis, 9 (2005), pp. 457-466.

[5] R. Lu, P. Marziliano, C. H. Thng, *“*Liver tumor volume estimation by semi-automatic segmentation method,” Proc. IEEE Engineering in Medicine and Biology 27th Annual Conference (2005)*,* pp. 3296-3299.

[6] P. A. Yushkevich, J. Piven, H. C. Hazlett, et al., *User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability*, Neuroimage, 31 (2006), pp. 1116-1128.

[7] D. T. Lin, C. C. Lei, S. W. Hung, *Computer-aided kidney segmentation on abdominal CT images*, IEEE Transactions on Information Technology in Biomedicine, 10 (2006), pp. 59-65.

[8] J. G. Sled, G. B. Pike, S*tanding-wave and RF penetration artifacts caused by elliptic geometry: an electrodynamic analysis of MRI*, IEEE Transactions on Medical Imaging, 17 (1998), pp. 653-662.

[9] K. Engel, M. Hadwiger, J. Kniss, C. Rezk-Salama, D. Weiskopf, Real-time Volume Graphics (CRC Press, 2006).

[10] S. U. Lee, S. Y. Chung, R. H. Park, *A comparative performance study of several global thresholding techniques for segmentation*, Computer Vision, Graphics, and Image Processing, 52 (1990), pp. 171-190.

[11] I. Bankman, Handbook of Medical Image Processing and Analysis (Elsevier, 2008).

[12] R. Adams, L. Bischof, *Seeded region growing*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 16 (1994), pp. 641-647.

[13] N. Otsu, *A threshold selection method from gray-level histograms*, IEEE Transactions on Systems, Man, and Cybernetics, 9 (1979), pp. 62-66.

[14] G. Dougherty, Digital Image Processing for Medical Applications (Cambridge University Press, Cambridge, UK, 2009).

[15] J. Wu, S. Poehlman, M. D. Noseworthy, et al., “Texture feature based automated seeded region growing in abdominal MRI segmentation,” Proc. I*nternational Conference on BioMedical Engineering and Informatics* (2008), pp. 263-267.

[16] C. E. Woodcock, A. H. Strahler, D. L. B. Jupp, *The use of variograms in remote sensing: I. Scene models and simulated images*, Remote Sensing of Environment, 25 (1988), pp. 323-348.

[17] P. Maillard, “Developing methods of texture analysis in high resolution images of the Earth,” Proc. Simpósio Brasileiro de Sensoriamento Remoto (2001), pp. 1309-1319.

[18] S. K. Ueng, H. K. Huang, H. C. Huang, *A G-code generator for volumetric models,* Applied Sciences, 9 (2019), pp. 1-15.

[19] S. K. Ueng, H. C. Huang, C. S. Chou, H. K. Huang, *Layered manufacturing for medical imaging data*, Advances in Mechanical Engineering, 11 (2019).

[20] Credence Cartridge Radiomics phantom, https://scidm.nchc.org.tw/dataset/ccradiomicsphantom, accessed June 2020.

[21] F. R. Zakani, K. Arhid, M. Bouksim, et al., “New measure for objective evaluation of mesh segmentation algorithms,” Proc. IEEE International Colloquium on Information Science and Technology (2016), pp. 416-421.

[22] DICOM standard, https://www.dicomstandard.org/current/ , accessed June 2020.

[23] Hounsfield units, https://en.wikipedia.org/wiki/Hounsfield_scale , accessed June 2020.

[24] ITK-SNAP segmentation program, http://www.itksnap.org/pmwiki/pmwiki.php , accessed June 2020.

[25] Y. Y. Wong, P. C. Yuen, C. S. Tong, *Segmented snake for contour detection,* Pattern Recognition, 31 (1998), pp. 1669-1679.

A. A. Giannopoulos et al., *3D printed ventricular septal defect patch: a primer for the 2015 Radiological Society of North America (RSNA) hands-on course in 3D printing*, 3D Printing in Medicine 1.1 (2015).

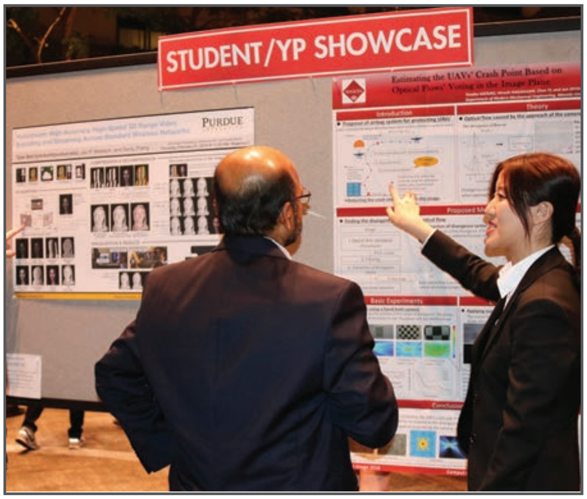
JOIN US AT THE NEXT EI!

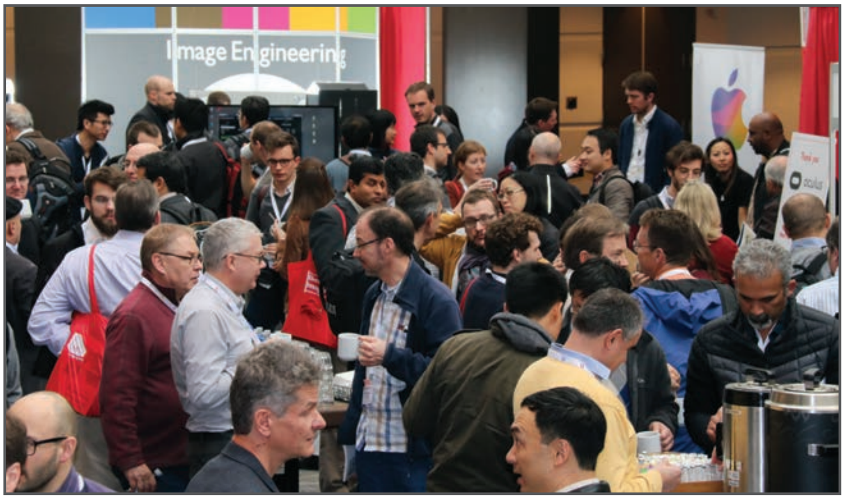
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





- AR is a fundamental and transformative interface and platform.
- 20+ years from my initial working demo to today's AR displays
- Optimistic that it won't take another 20 to reach ubiquitous consumer AR!

• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org