# A prioritized and adaptive approach to volumetric seeded region growing using texture descriptors

Nathan J. Backman[a], Brian W. Whitney[b], Jacob D. Furst[c], Daniela S. Raicu[c]
[a]Whitworth College, 300 W. Hawthorne Road, Spokane, WA, 99251, USA
[b]Northern Kentucky University, Nunn Drive, Highland Heights, KY, 41099, USA
[c]Intelligent Multimedia Processing Laboratory, School of Computer Science, Telecommunications, and Information Systems, DePaul University, Chicago, IL, 60604, USA

## ABSTRACT

The performance of segmentation algorithms often depends on numerous parameters such as initial seed and contour placement, threshold selection, and other region-dependent *a priori* knowledge. While necessary for successful segmentation, appropriate setting of these parameters can be difficult to achieve and requires a user experienced with the algorithm and knowledge of the application field. In order to overcome these difficulties, we propose a prioritized and adaptive volumetric region growing algorithm which will automatically segment a region of interest while simultaneously developing a stopping criterion. This algorithm utilizes volumetric texture extraction to establish the homogeneity criterion by which the analysis of the aggregating voxel similarities will, over time, define region boundaries. Using our proposed approach on a volume, derived from Computed Tomography (CT) images of the abdomen, we segmented three organs of interest (liver, kidney and spleen). We find that this algorithm is capable of providing excellent volumetric segmentations while also demanding significantly less user intervention than other techniques as it requires only one interaction from the user, namely the selection of a single seed voxel.

Keywords: volumetric, region growing, segmentation, texture extraction

## 1. INTRODUCTION

Image segmentation is the isolation of a region of interest from a visual dataset and is commonly used for content-based image retrieval, face and hand recognition, object tracking, and as a preliminary step for medical image analysis. Segmentation algorithms must, therefore, have different strengths depending on the purpose for segmentation. For instance, object tracking and hand or gesture recognition algorithms must operate very quickly, often sacrificing accuracy to do so, in order to function in real-time or within the constraint of the frame-rate of an output device. Many medical fields necessitate a much greater demand for accuracy[8], for instance surgical planning, and therefore the speed of the algorithm becomes second to precision.

There are many segmentation algorithms available, but few are able to perform with the accuracy required by the medical domain. Active contours and deformable models are unable to segment regions with complex boundaries, threshold techniques are not designed to deal with textured regions of interest, and edge finding segmentation algorithms do not isolate only a single region of interest within an image. Furthermore, it must be recognized that if a segmentation algorithm will be used in the medical domain there must be a significant ease of operation with little user intervention as the users of the algorithm, whose expertise will likely lie in a medical field, should not be burdened with the task of modifying parameters, selecting thresholds, placing initial contours, or adjusting other factors that require region-dependent *a priori* knowledge[9]. Current unsupervised segmentation algorithms are not able to consistently match the accuracy of supervised segmentation algorithms so there is still a need to consider the degree of user intervention when comparing algorithms for medical image segmentation.

It is also important to recognize that there are multiple 3-dimensional image modalities in the medical domain. With this additional information comes the need to segment complex volumetric regions. Volumetric segmentation results can be used for 3-dimensional visualization purposes as well as a preprocessing step for other classification and analysis algorithms which can benefit from the additional data to analyze.

Texture, according to Smith and Chang[11], refers to a visual pattern that contains properties of homogeneity. Previous studies have shown that it is possible to perform pixel-level texture extraction in 2-dimensional CT images[5]. With CT studies now isotropic in three dimensions, it is all the more desirable to extend the previous work to take into

Computational Imaging IV, edited by Charles A. Bouman, Eric L. Miller, Ilya Pollak,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6065, 60651A, © 2006 SPIE-IS&T · 0277-786X/06/$15

account the extra dimension when extracting texture features. We therefore implemented voxel-level texture extraction using Haralick's[3] co-occurrence model on a volumetric co-occurrence[6] matrix derived from a 3-dimensional neighborhood around a voxel. The resultant texture features were then used to formulate a homogeneity criterion. With such a homogeneity criterion it is possible to define a region of interest.

The approach to segmentation described in this paper involves the implementation of a seeded region growing algorithm, initially proposed by Adams and Bischof[1], that has been modified by the integration of an adaptive and prioritized search as well as voxel-level texture feature extraction that is used to determine a homogeneity criterion. By extracting voxel-level texture features we can progressively determine the texture features that characterize the region average as the region average is updated with the addition of each voxel to the region. This process differs from traditional region growing algorithms as voxels adjacent to the region are kept in a prioritized list, ordered by similarity to the region average, in which the voxel most similar to the region average is at the front[2]. The algorithm operates by taking voxels at the front of the list and adding them to the region first. This ensures that the region expands to the most similar voxels first while preserving the integrity of the region average. By keeping a list of visited voxels we maintain a histogram that expresses the similarity of each voxel to the region average they were initially compared to. In analyzing this histogram we can visualize the rate at which the similarity between newly visited voxels and the region average increases or decreases. Thus when filling a region of interest, the histogram will denote a bell-shaped structure, representing a region of similarity. The algorithm then continues filling the region until the most similar remaining voxels do not lie within the general curve representing the region of interest. That is to say, the most similar voxel is no longer within a certain cut-off threshold defined by the bell-curve.

Due to the fact that the algorithm is capable of finding texture features that quantify the region of interest and is able to automatically determine a stopping criterion, our algorithm requires very little user-intervention. The user is required to do nothing more than identify the seed.

## 2. METHODOLOGY

### 2.1. Data Compression

The CT studies used contained 3D image data consisting of consecutive 2D DICOM (Digital Imaging and Communications in Medicine) slices where each slice was 512 by 512 pixels with a 12-bit gray-level resolution. Extracting voxel-level texture features requires computations on co-occurrence matrices whose dimensions are directly related to the gray-level resolution. Therefore, in order to greatly reduce computational costs, we decreased the number of gray-levels by a factor of 16 resulting in 256 intensities. A study evaluating the method of image compression known as clipped binning has shown that a reduction from 4096 gray-levels to 256 gray-levels still provides satisfactory discriminatory value between abdominal organs when evaluating soft-tissues with the use of texture features[10].

In order to reduce the number of gray-levels in the data by a factor of 16, we implemented clipped binning. Clipped binning linearly reduces a specified range of gray-levels to a smaller range, while allocating everything above

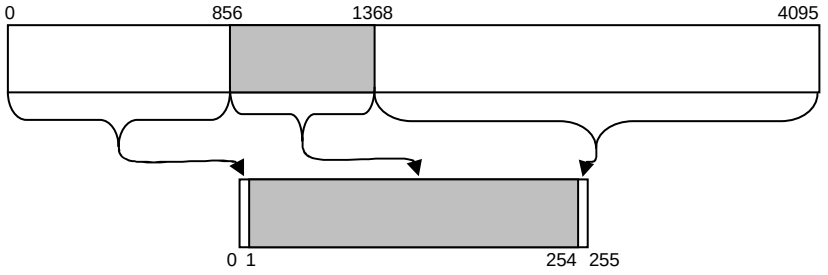


Figure 1: Clipped binning

and below the original range to one gray-level above and below the resultant range, respectively[7]. Radiologists from Northwestern Memorial Hospital claimed that soft-tissues in DICOM images were represented by gray-levels in the range [856, 1368], which was affirmed by Lerman[7] with the use of a k-means clustering algorithm. This technique was then used to effectively reduce all non soft-tissue elements in the data to one of two gray-levels while distributing the soft-tissue range to the remaining gray-levels. We started with 4096 gray-levels (12-bits, [0, 4095]), and compressed the range to 256 gray-levels whereas gray-levels [0, 855] were mapped to gray-level 0, gray-levels [856, 1368] were linearly distributed to gray-levels [1, 254], and gray-levels [1369, 4095] were mapped to gray-level 255.

### 2.2. Voxel-level texture extraction

A region growing algorithm requires a criterion of homogeneity in order to evaluate voxels and determine if they should be included in a region[1]. We used voxel-level texture extraction to calculate 3D texture features for the voxels in order to compare them to the texture features that characterize the region average. This technique is similar to pixel-level texture extraction, explained by Kalinin[5], although projected onto an additional dimension.

To perform voxel-level texture extraction, we select a voxel and define a 3x3x3 neighborhood about the voxel in order to develop a 2-dimensional co-occurrence matrix from the volume of gray-level intensities. This co-occurrence

| Texture Feature | Formula | Description |
|---|---|---|
| Entropy | $-\sum_{i}^{M}\sum_{j}^{N} P[i,j]\log P[i,j]$ | Provides a measurement of the randomness of a gray-level distribution |
| Mean | $\frac{1}{2}\sum_{i}^{M}\sum_{j}^{N}(iP[i,j]+ jP[i,j])$ | Provides the mean of the gray-levels in the region |
| Variance | $\frac{1}{2}\sum_{i}^{M}\sum_{j}^{N}\left((i-\mu)^2 P[i,j]+(j-\mu)^2 P[i,j]\right)$ | Provides the variance of the distribution of gray-levels in the region |
| Cluster Tendency | $\sum_{i}^{M}\sum_{j}^{N}\left(i-\mu_i + j-\mu_j\right)^2 P[i,j]$ | Provides a measurement of groupings of voxels with similar gray-levels |

Table 1: Texture features

matrix P, is of size *n* by *n*, where *n* is the number of gray-levels in the CT study. The volumetric co-occurrence matrix accrues voxel-pair intensities such that P[*i, j*] records the number of occurrences of voxel-pairs with intensities *i* and *j*[6]. We then find all voxel-pairs by iterating through the 3x3x3 cube in an ordered manner to insert values into the co-occurrence matrix.

From the volumetric co-occurrence matrix we calculate four texture features proposed by Haralick, those being entropy, mean, variance, and cluster tendency[3]. These texture features are defined in Table 1. Since the texture features extracted from a voxel are compared to the texture features that characterize the region average, it was necessary to find the distribution of each texture feature to determine how much two texture features differ. By taking an initial sampling of 10,000 voxels from the CT study, we determined the distribution of the values of each texture feature in relation to our data. In our samplings, the distributions of the variance and mean represented rather normal distributions while cluster tendency and variance resulted in exponential distributions. In order for all texture features to exhibit normal

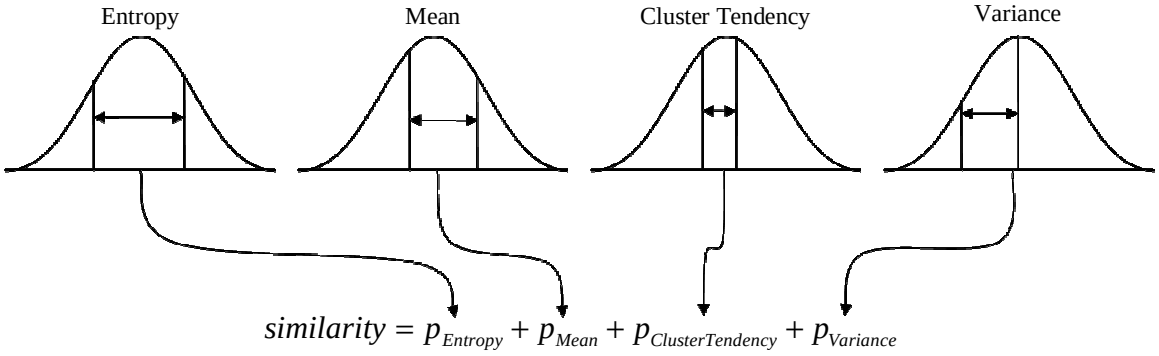


Figure 2: Mahalanobis distance

distributions we took the logarithmic value of the cluster tendency and entropy values which resulted in normal-like distributions. By doing this we were able to compare texture features taken from any voxel with the entropy, mean, cluster tendency, and variance of the running region average.

It was then necessary to provide a quantitative measurement defining the similarity between the texture features of a voxel and the texture features of the region average. Since sampling the dataset provides distributions of the texture

features, it is possible to find the probability between values in the distributions. By taking the Mahalanobis distance of the corresponding values of each texture feature, we found the degree to which each texture feature of a voxel deviates from the corresponding texture feature of the region average. We then took the Manhattan distance of the four resultant values providing a quantitative measurement of the similarity between any voxel and the region average, which can be seen in Figure 2.

### 2.3. Region growth

With a means of evaluating the similarity between voxels and the region average, we can start the region-growing process. When starting the algorithm, the region average will be based entirely on the seed voxel. That is, the entropy, mean, cluster tendency, and variance of the region average will be identical to that of the seed voxel. The algorithm will then proceed to evaluate voxels adjacent to the region, whereas the region is initially composed of only the seed, and then form a prioritized list of the adjacent voxels ordered by similarity to the region average[2]. Therefore the voxel most similar to the region will be at the front of the queue. The algorithm will then select the voxel at the front of the prioritized list and insert it into the region. The four texture features of the newly added voxel will then be used to update the texture features characterizing the region average allowing it to be an accurate representation of all voxels included thus far. The texture features of the voxels adjacent to the new addition to the region will then be calculated in order to add those voxels to the prioritized list so that they may be considered for the region as well. This process is repeated to allow region growth, whereas the region will spread to voxels most similar to the region average first.

Before the region growing process can start, the algorithm must first be initialized by the selection of a seed voxel as a starting point for growth as well as for a basis for the region average. It is then necessary to allow the algorithm to grow and spread to similar neighboring voxels in order to allow the region average to better represent the region of interest as opposed to just the area of the seed voxel. After an initial period of growth, the prioritized list, including voxels already added to the region, generates a histogram that accumulates voxels sorted by similarity to the region average. With this initial growth, the histogram represents a bell-shaped curve. This bell shaped curve essentially denotes the region of interest whereas the right-most voxels have yet to be added to the region due to their

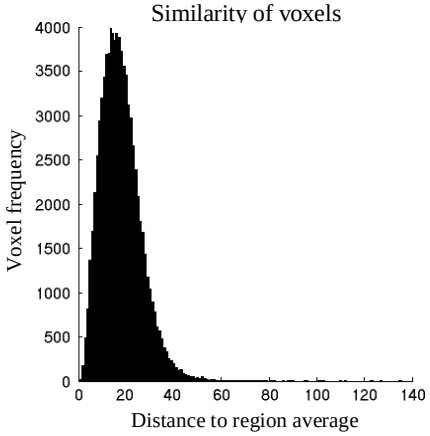


Figure 3: Histogram derived from prioritized list

dissimilarity. The right half of the bell-curve shows a notable increase in dissimilarity accompanied by an evident decrease in voxel occurrences. This change from frequently occurring voxels to less frequently occurring voxels of greater dissimilarity signifies a separation between voxels that are reasonably similar to the region average and those that are not. That is to say, the bottom of the right tail of the bell-curve represents the border of a region of interest.

Since the right tail of the histogram represents voxels adjacent to the region that are decreasing in similarity to the region average, it is desired to prevent region growth to these voxels. In order to completely fill the entire region it would be required to grow as close to the boundary as possible. In doing this, however, there is the potential problem of leaking and including voxels that are not actually within the region of interest. We claim, however, that it is possible to segment a region by filling most of it and then find the shape by filling in holes and softening edges with morphological operators[4]. To that extent, it is not our desire to add voxels to the region until we have reached the boundaries, but to find most of it. We then, relative to the right tail of the bell-curve, define a cut-off threshold such that voxels with a similarity less than such a value will not be added to the region.

Since it is possible, in early stages of region growth, for the bell-curve of the histogram to not yet fully represent the region of interest, we progressively redefine the cut-off threshold. After an initial period of growth, the first cut-off threshold is calculated as the similarity value that corresponds to the first bin of the histogram that is to the right of the peak bin as well as less than 90% of the height of the histogram. Since this initial value does not yet likely correspond to a cut-off threshold that would be used for the region of interest as a whole, we continue adding voxels to the region until a voxel is added whose similarity to the region average exceeds the cut-off threshold. We then take this, given additional growth of the region and correspondingly the histogram, to recalculate the cut-off threshold which

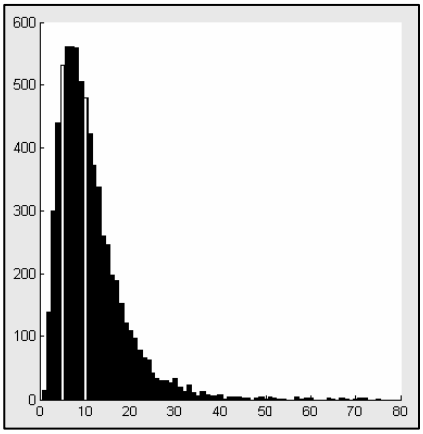


Figure 4: Histogram derived from prioritized list

eventually moves outwards. Should the cut-off threshold be exceeded immediately after recalculation, signifying that the voxel most similar to the region average in the prioritized list exceeds yet again the newly calculated threshold, region growth will be terminated. Figure 4 portrays a screenshot of the user-interface developed for the algorithm. The figure shows the histogram and two vertical bars within it. The right-most bar represents the current cut-off threshold and the left bar represents the last evaluated voxel. As additional voxels are added to the region, the left bar will reposition itself relative to the voxels added and eventually reach the cut-off threshold in which a new cut-off value will be calculated. We can note from this figure that the majority of voxels to the right of the cut-off threshold bar are not in the region, but adjacent to the region and waiting in the prioritized list.

The cut-off threshold is calculated as a direct result from the bins of the histogram. In order to generate a histogram from the prioritized list, it is necessary to define the number of bins that will be used to represent the histogram. If too many bins are chosen to create the histogram, precision is lost when calculating a cut-off value. If too few bins are used to create the histogram, the bell curve becomes distorted which can cause problems for generating a

$$bins = 75 + 10 * \left\lceil \frac{\log\left(\frac{numVoxels}{1000}\right)}{\log(2)} \right\rceil$$

Equation 1: Logarithmic bin-expansion

cut-off value. Initial histograms may also suffer if there is too little data spread across an excess of bins. It is therefore necessary for the number of bins to increase relative to the increasing region. We decided to make the number of bins representing the histogram increase logarithmically corresponding to the number of voxels in our region while starting with an initial amount of 75 bins. Therefore, every time the cut-off value must be recalculated, and thus a histogram recreated as well, the number of bins used is computed as shown in Equation 1. This ensures that the number of bins grows at a rate which accommodates increased precision, as the number of voxels in the region increases, while keeping a small amount of initial bins to maintain a the bell-shaped distribution at early stages. We chose to increase the number of bins logarithmically because the addition of bins at a linear rate would result in far more bins than would be necessary for large regions of interest such as the liver.

Since our goal, in region growing, was to not fill every voxel of the region of interest but instead a majority, we applied morphological operators to our segmented volume. We used dilation to fill in the dense portions of the segmented region and blurring allowed for the softening and rounding of edges in our segmentations.

## 3. RESULTS

The CT studies used to achieve our results were obtained from Northwestern Memorial Hospital and the National Library of Medicine. All segmentations were performed with a single user-intervention which called for the selection of a seed voxel from the user-interface shown in Figure 5. This only requires the user to scroll through

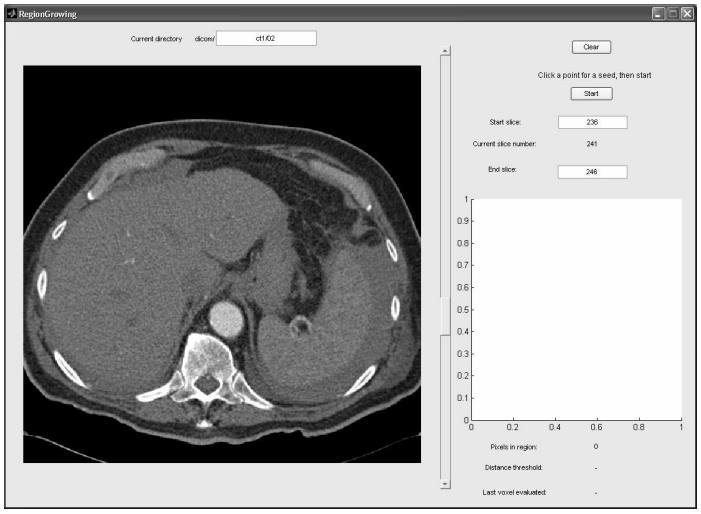


Figure 5: User interface

consecutive slices of CT images, click the cursor on a region of interest, and click on the start button. The following segmentations are of the kidney, liver, and spleen. Although the captured images only show a 2-dimensional segmentation, each organ was segmented volumetrically. Each figure of a segmentation shown here has three parts. The left part of each image shows a single slice of the volumetric region of interest before segmentation. The middle portion of the image shows a single slice of the volumetric region of interest and an outline defining the results of the segmentation. The right portion of the image contains a single slice of the isolated volumetrically segmented region of interest alone.

Figure 6 shows a single slice of the results of the volumetric segmentation of a right kidney. Notice that the result has a "U" shape. This gap, which was not segmented, is the collecting duct of the kidney and is not desired in the

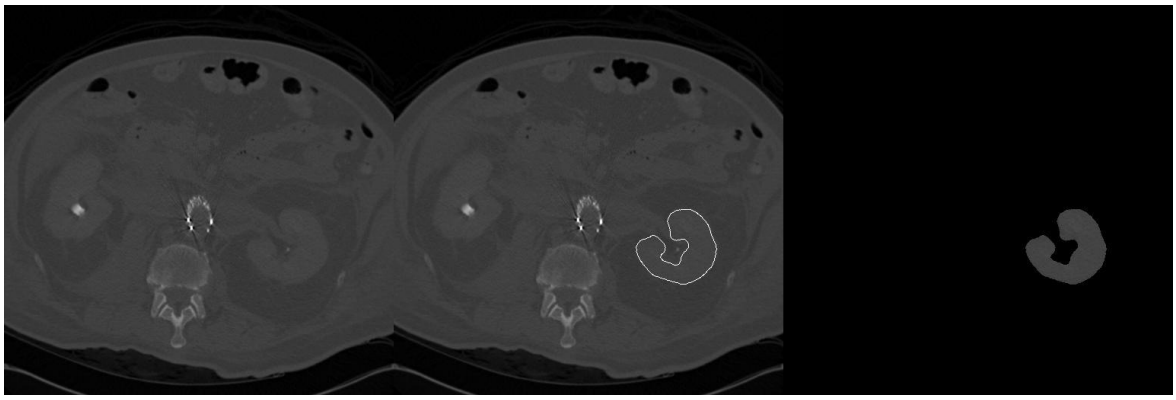

Figure 6: Segmentation of the right kidney

result, therefore the algorithm worked as intended. Thus, the texture of the kidney tissue and that of the collecting duct

were different enough to be classified as separate regions. This result shows that our algorithm is capable of segmenting complex volumetric structures such as the "U" shape that many active contour models are unable to properly segment.

The next volumetric segmentation, Figure 7, is of a liver and demonstrates the capability of our algorithm to segment donut-like regions of interest that have not only external boundaries but internal boundaries as well. These

Figure 7: Segmentation of the liver

kinds of phenomena prevent contours and deformable models from successful segmentations. It is also worth noting the visual similarity connecting the liver and the tissue just left of the liver and outside of the ribs, which texture features

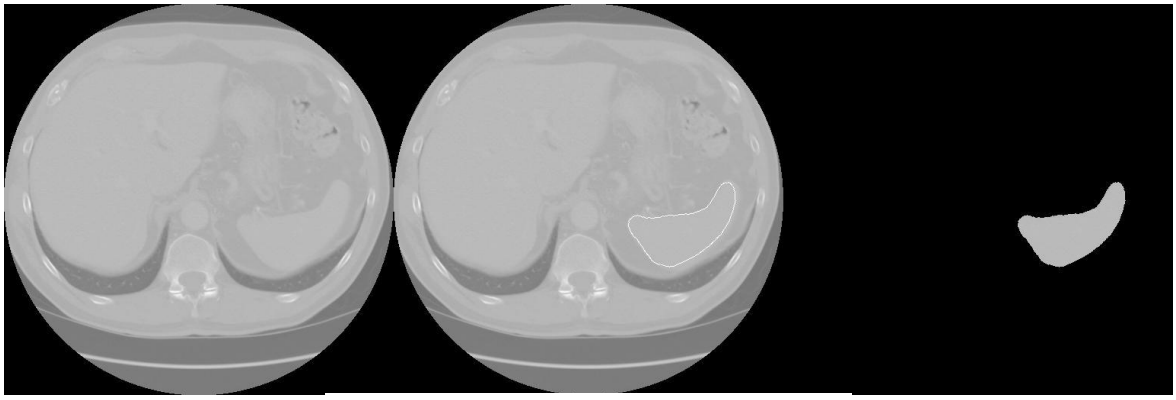

Figure 8: Segmentation of the spleen

were able to discriminate between. The final volumetric segmentation shown, Figure 8, is of the spleen.

## 4. CONCLUSIONS

The immediate benefits demonstrated by our algorithm are the production of quality volumetric segmentations and a parameter-less user-interface that takes only one user intervention, the selection of a single seed voxel. Our algorithm does, however come with drawbacks. Volumetric CT studies contain an extremely large amount of data. Regions of interest may contain potentially millions of voxels. It may, therefore, take quite a few hours to segment larger organs such as the liver. It must also be noted that the algorithm operates on a criterion of homogeneity. It is, therefore, required for a region of interest to be homogeneous in texture. Organs of a complex makeup, such as the heart would be much more difficult to segment given the varying amount of textures present.

For future work we are interested in examining alternate image compression strategies such as continuous clipped-binning[7] which is similar to clipped-binning but the target range of gray-levels would be distributed continuously to the resultant gray-levels instead of linearly. It is also worth looking into alternate texture features to see if there may be a better combination which may allow for greater discrimination between soft-tissues. We are also interested in the potential for unsupervised segmentation which would utilize a multi-seeded placement strategy in order to automatically accomplish segmentation.

## REFERENCES


1. R. Adams, L. Bischof, “Seeded Region Growing”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16 no. 6, pp. 641–647, 1994.
2. M. F. Cohen, J. Painter, M. Mehta, K. Ma, “Volume Seedlings”, *Proceedings of the 1992 Symposium on Interactive 3D Graphics*, pp. 139-145, 1992.
3. R. M. Haralick, K. Shanmugam, I. Dinstein, “Textural features for image Classification”, IEEE Transactions on Systems, Man, and Cybernetics, vol. Smc-3 no. 6, pp. 610-621, 1973.
4. R. Huang, K. Ma, “RGVIs: Region growing based techniques for volume visualizations”, *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, pp. 355-363, 2003.
5. M. Kalinin, D. S. Raicu, J. D. Furst, D. S. Channin, “A classification approach for anatomical regions segmentation”, *The IEEE International Conference on Image Processing (ICIP)*, 2005.
6. A. S. Kurani, D. H. Xu, J. D. Furst, D. S. Raicu, “Co-occurrence matrices for volumetric data”, *The 7th IASTED International Conference on Computer Graphics and Imaging - CGIM 2004*, 2004.
7. R. Lerman, D. S. Raicu, J. D. Furst, "Contrast enhancement of soft tissues in computed tomography images", Proceedings of SPIE Medical Imaging 2006.
8. T. S. Newman, N. Tang, S. Bacharach, P. Choyke, “A volumetric segmentation technique for diagnosis and surgical planning in lower torso CT images”, *Proceedings of the 1996 International Conference on Pattern Recognition*, pp. 553-557, 1996.
9. S. D. Olabarriaga, A. W. M. Smeulders, “Interaction in the segmentation of medical images: A survey”, Medical Image Analysis, vol. 5, pp. 127-142, 2001.
10. D. S. Raicu, S. Handrick, B. Naimipour, J. D. Furst, "Binning strategies evaluation for tissue classification in computed tomography images", Proceedings of SPIE Medical Imaging 2006.
11. J. Smith, S. Chang, “Automated binary texture feature sets for image retrieval”, *IEEE International Conference on Acoustics*, vol. 4, pp. 2239-2242, 1996.