

# Reconstruction of the Human Brain from MRI-T1 Using 3-D Morphology and Snake

Chih-Yang Lin<sup>a</sup>, Yu-Tai Ching<sup>a</sup>

<sup>a</sup>Department of Computer and Information Science, National Chiao Tung University, Hsin Chu, Taiwan

## ABSTRACT

Accurate reconstruction of the human brain in MRI-T1 images is valuable and important to clinical needs. In this paper, the morphology and snake techniques are proposed to reconstruct a human brain model. First step in our method is to preprocess the volumetric image to remove skull, muscle, fat, and other non-brain tissue. We use a method of 3-d region growing. It has the advantage over thresholding that the resulting objects will be spatially connected, since brain has the connected property. Second, we use clustering method, and then use them to produce an initial estimate of the cortical surface. Third, we propose a novel active contour algorithm to move the snake toward the cortex. Thus we can use the snake to segment the brain. We use a wavelet method to model the external force that significantly increases the capture range of a traditional snake. Afterwards, we render the volumetric image to display the brain from multiple views. Both simulated data and patient data have been used to test the proposed techniques. The proposed method combines various techniques of 3-D morphology, clustering, active contour, wavelet, and volume rendering to accurately, robustly, and automatically reconstruct brain from MRI-T1 images.

Keywords: Brain, MRI-T1, Active Contour, Snake, Wavelet.

## 1. INTRODUCTION

The major difficulty in automatic segmentation techniques fail dramatically because of two disturbing factors; (1) the MR acquisition suffers from image gradients and/or FR coil inhomogeneities, and (2) are often linked to each other owing to partial volume effects, noise, imaging artifacts, or by connecting tissue. Although several approaches can be considered powerful tools for segmentation, rarely is fully automatic [1]-[2]. In this paper, a method is proposed to automatically segment the brain from MR-T1 images and reconstruct them using volume rendering to display volumetric data as a two-dimensional image. We use 3-D morphology and a region growing process to roughly segment the brain. An active contour algorithm is used to move the snake toward the cortex. We use a wavelet method to model the external force that significantly increases the capture range of a traditional snake. Thus we can use the snake to accurately segment the brain. Afterwards, we render the volumetric image to display the brain from multiple views. Our method consists of three major processes: (1) skull removal (2) active contour and wavelet, and (3) volume rendering, which are detailed in the next section. Several experimental results are demonstrated in Section 3 to show the robustness of the proposed techniques.

## 2. METHODS

### 2.1 Skull removal

In this step, we use 3-D morphology and region growing techniques to remove skull. The first and highest peak refers to background voxels in the histogram of MRI brain image. Thus we use the gray value that between the first two peaks in the histogram as threshold to remove the background voxels. The highest peak is selected as initial threshold value  $T_s$ . A volume of possible seed points is defined by thresholding with  $T_s$ . Then this seed mask volume is eroded several times to assure it is not containing skull. A region growing procedure is initiated from this seed mask volume, and grows one layer of voxels at each of iteration. Since the seed volume contains two most important anatomies. The region growing process is iterated when the difference between the growing voxels and the class centers are within certain threshold  $T_h$ . We can obtain approximate cortex and its estimated contour. The details are shown in Fig. 1.

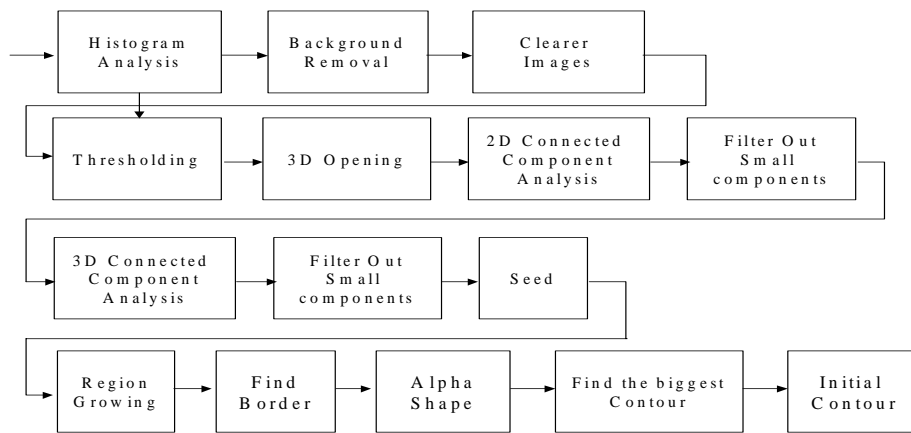


Figure 1. The flow chart to remove the skull and get initial contour.

## 2.2 Active contour and wavelet

In this step, we use active contour algorithm [3] to move the snake, the estimated contour from previous step, toward the cortex. Thus we can use the snake to segment the brain. A wavelet based method is used to form the external force model that significantly increases the capture range of a traditional snake.

Active Contours describes an energy minimization problem for finding the contour of a local feature in an image. As the snake seeks to minimize its overall energy, its shape will converge on the image gradient contour. Energy in this active contour model is represented by two force terms:  $E_{in}$ ,  $E_{ex}$ . The total energy is therefore:

$$E_{snake} = \int E_{in}(v(s)) + E_{ex}(v(s)) ds, \quad (1)$$

where  $v(s)$  represents the contour coordinates as a function of arc length. Each of the terms that make up the force model can be intuitively realized in a physical sense.  $E_{in}$  represents the internal force of the snake. It is responsible for maintaining continuity between points by controlling segment length and vector curvature.  $E_{in}$  is modeled by the following form :

$$E_{in} = (\alpha(s) * |v'(s)|^2 + \beta(s) * |v''(s)|^2) / 2. \quad (2)$$

In the expression above, the  $\alpha$  coefficient controls the first-order term, which is continuity. The second-order term, controlled by  $\beta$ , represents curvature.

The second fundamental force term is the image energy that is so-called external force ( $E_{ex}$ ). External force is usually represented as the negative magnitude of the gradient. At points of high contrast, the external force term will create a large negative force. It will provide the external stimulus to mold the snake.

For this implementation, there were 3 explicitly defined energy terms: continuity ( $\alpha$ ), curvature ( $\beta$ ), and external force ( $E_{ex}$ ). For each time step, the list of the snake points is traversed from 0 to 1 (hitting node 0 once at the beginning and again at the end). When each snake point is visited, all possible moves within a neighborhood are considered. The move that results in the least total energy will become the new position of the snake point.

The most important part about understanding and implementing this algorithm is to get an intuitive sense about how the energy terms work together as well as independently. In the active model, all three terms coexist and respond to each other, hopefully causing the snake to converge to its rest contour.

The first term, continuity, likes to maintain equidistance between points in the snake. It is formulated by taking the difference between the average segment length and the current snake segment length. The minimum energy for continuity results when the current segment length is closest to the average.

$$E_{cont} = \alpha * (avgDist - |V_i - V_{i-1}|), \quad (3)$$

where  $V_i = (X_i, Y_i)$ .

Curvature is the other internal force term, which is a measure of the change in external angle with respect to arc length (segment length). From an intuitive standpoint, the curvature term should try to make the snake form a convex circular shape when there are no other forces applied. Unfortunately though, the introduction of continuity causes the impetus for circularity to result in rapid shrinking. This is both good and bad. It's good because in most cases, the snake needs to converge on a shape by shrinking. However, in the cases where a snake needs to expand, there needs to be significant local external force to pull on it.

The third and most important term is the force due to the image itself. This force results from the magnitude of the gradient. When this term is thought of in a negative sense, the image energy is minimal when the intensity differential is greatest. This means that edges will look very attractive to the snake. One problem with traditional snake formulations is capture range. If  $\sigma$  is small, the magnitude of image energy will die out quickly away from the object boundary. Increasing  $\sigma$  will increase this range, but the boundary localization will become less accurate, ultimately distorting the edge itself when  $\sigma$  becomes too large. To overcome this problem, we then used a wavelet transformation based method to form the external force model that significantly increases the capture range of a traditional snake. For completeness, we briefly describe the Discrete Wavelet Transform (DWT) in the follows. The details of the wavelet transformation can be found in [4]-[6].

Let  $L(\omega) = \sum_k l_k e^{-jk\omega}$  and  $H(\omega) = \sum_k h_k e^{-jk\omega}$  be respectively a lowpass filter and a highpass filter that satisfy the orthogonality condition as show in the following equation

$$|L(\omega)|^2 + |H(\omega)|^2 = 1. \quad (4)$$

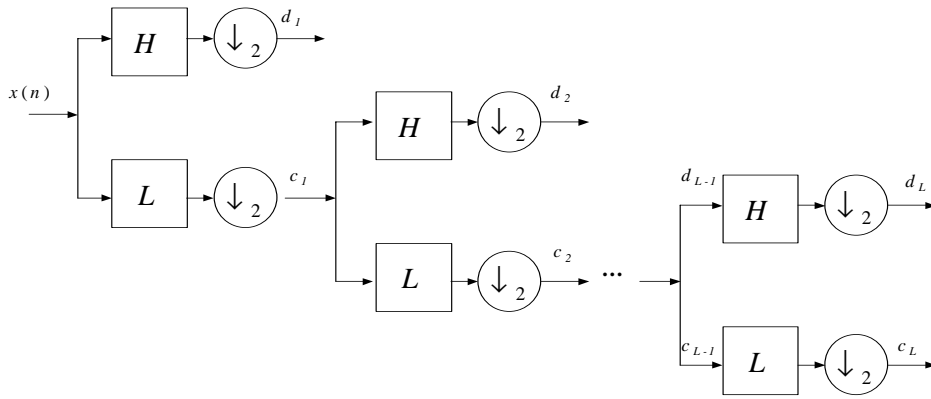
The DWT employs a highpass filter and a lowpass filter to decompose an input signal into high frequency and low frequency components in different resolutions according to the number of levels employed. In the one-dimensional case, a signal  $x(n)$  is decomposed recursively according to:

$$c_{j+1,k} = \sum_n l_{n-2k} C_{j,n}, \text{ and} \\ d_{j+1,k} = \sum_n h_{n-2k} C_{j,n}, \quad (5)$$

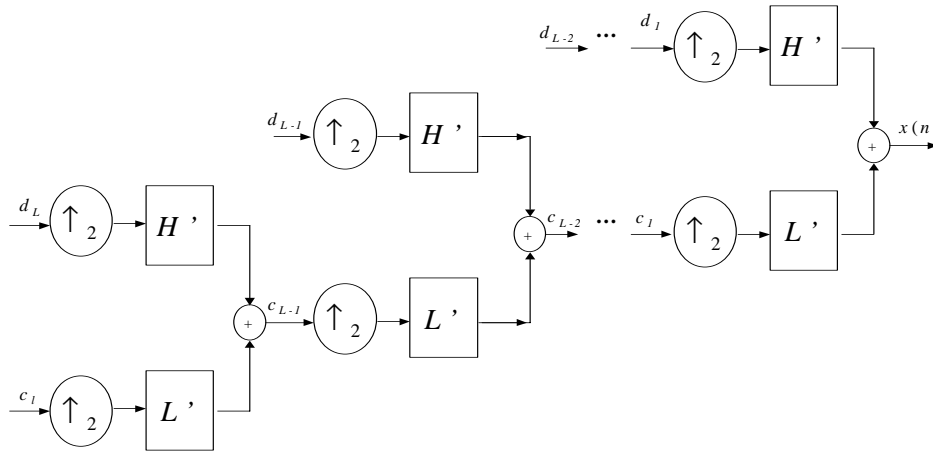
which correspond to a convolution followed by a process of down-sampling by 2 as shown in Fig 2.  $l_k$  and  $h_k$  are the impulse responses of the lowpass and the highpass filters. Index  $j$  indicates the number of decomposition levels that lies in the range  $[0, L]$ , where 0 represents the index of the highest resolution level and  $L$  represents the index of the lowest resolution level.  $C_{L,k}$  is equal to the input signal  $x(k)$ . The coefficients  $d_{1,k}, \dots, d_{L-1,k}, d_{L,k}$  are called the DWT wavelet coefficients of  $x(n)$ . The signal  $x(n)$  can now be reconstructed from its DWT coefficients by using the recursive formulation:

$$c_{j,n} = \sum_k l_{n-2k} C_{j+1,k} + \sum_k h_{n-2k} d_{j+1,k} \quad (6)$$

The reconstruction process is illustrated in Fig 2 and defines the inverse discrete wavelet transform (IDWT). This time, up-sampling precedes filtering at each level of the transform.



(a)



(b)

Fig. 2. Multi-level wavelet decomposition (a) and reconstruction (b) for 1-D.

A two-dimensional discrete wavelet transform and its inverse are the extension of the one-dimensional transform. They are simply implemented by using a one-dimensional DWT and IDWT along each of  $x$  and  $y$  coordinates. In other words, we apply a lowpass filter and a highpass filter along each of the two coordinates. We can then obtain four parts that are

$LL$ : obtained by applying lowpass filters on both coordinates,

*HL* and *LH*: obtained by applying highpass filter on one coordinate and lowpass filter on the other coordinate, and

*HH*: obtained by applying highpass filters on both coordinates.

In the proposed method, we applied an eight-stage discrete wavelet transformation as described in Eq. (7)

$$W(x, y, t) = DWT^8(f(x, y, t)) \text{ for } x, y \quad \forall 0 \leq t \leq T - 1, \quad (7)$$

The first stage of the transform splits the image into four quarter-size images, the upper left (*LL*<sub>1</sub>), the upper right (*HL*<sub>1</sub>), the lower left (*LH*<sub>1</sub>), and the lower right (*HH*<sub>1</sub>). The second stage splits the upper left image (*LL*<sub>1</sub>) into four images again. For the subsequent stage *j*, the upper left image (*LL*<sub>*j-1*</sub>) is decomposed in exactly the same way to form four smaller images. The resulted image is shown in Figure 3.

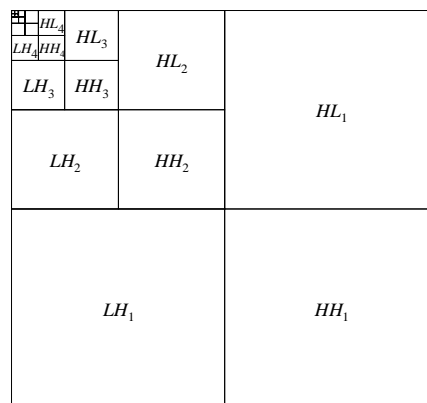


Figure 3. A 2-D eight-stage discrete wavelet transformation.

### 2.3. Volume rendering

In this step, we use the volume rendering to display volumetric data as a two-dimensional image. The steps of volume rendering consist of gradient calculation, resampling, classification, shading, compositing are a large series of computations. Here we only show the steps in fig. 5. The segmentation step is done in previous steps. The gradient computations find edges or boundaries between different materials. The gradient is a measure of how quickly the data in a data set changes. This information is used in the classification stage and the shading stage. As imaginary rays are passed through the block of voxels, samples are taken along each ray for accumulation. The samples are used to generate new values for samples that lie between the actual voxels. The process of generating new addressed into the voxel space and generating new values is called resampling or interpolation. The step of classification is to assign opacity values and colors to the voxels. You can think of classification as separating voxels into different feature classes, typically done by assigning different opacities and colors to voxels. Shading is used to highlight parts of the data that set by using an illumination model. Since one pixel on our display may represent hundreds of values we have sampled along a ray, we need a way to accumulate these values into one. Accumulation is accomplished via a composition function. The detail techniques can refer to [7].

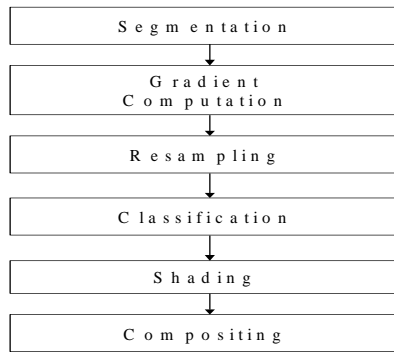


Figure 4. The steps of volume rendering

### 3. EXPERIMENTAL RESULTS

In this section, we present two results obtained by use of the proposed method. A sequence of brain images from MRI-T1 consisting of 128 images with the size of 256x256 pixels per image (Figure 5) is used in the experiment. The final results with different views from figure 5 are shown in Figure 6. The proposed methods were implemented on a PC with a Pentium III (700 MHz) CPU running on Windows NT operating system. The overall execution time took less than 25 minutes.

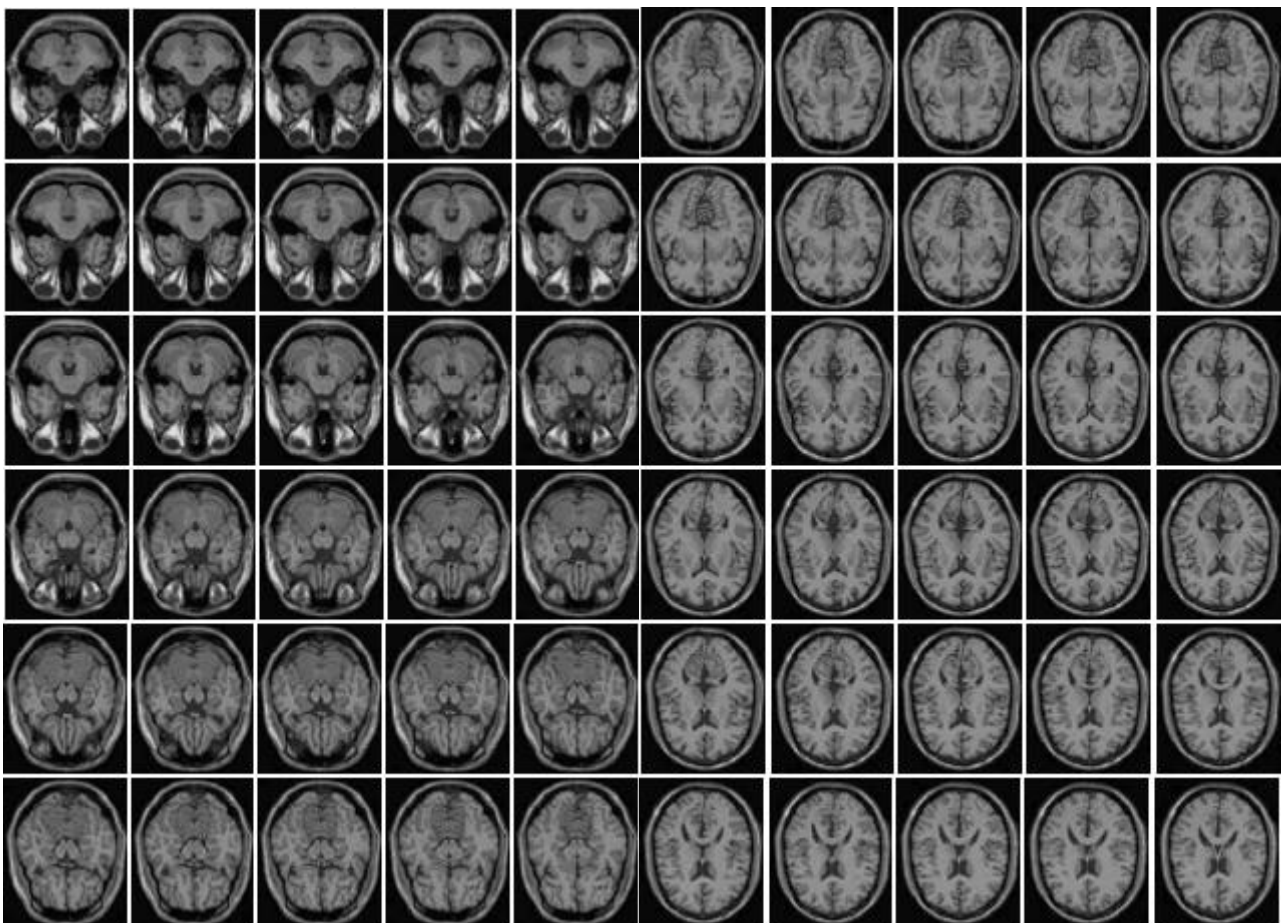


Figure 5. A part of a sequence of brain images from MRI-T1.

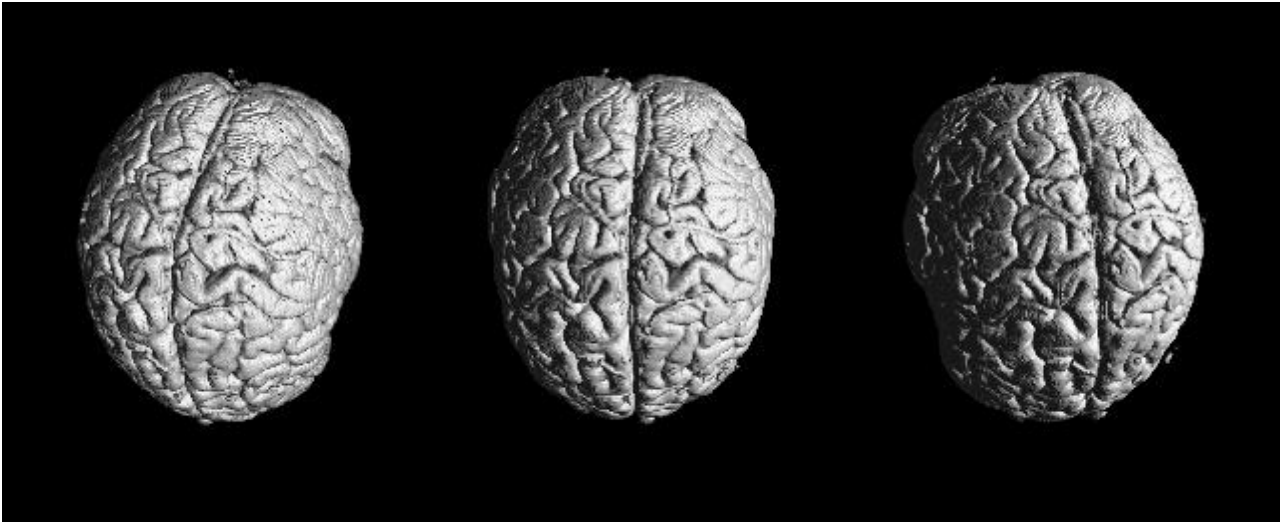


Figure 6. The final results with different views.

#### REFERENCES

1. K. H. Hohne, and W. A. Hanson, "Interactive 3D Segmentation of MRI and CT Volumes using Morphological Operations," *Journal of Computer Assisted Tomography*, vol. 16, no. 2, pp. 285-294, 1992.
2. C. Barillot, F. Lachmann, N. Gibaud, and J. M. Scarabin, "3-D Display of MRI data in neurosurgery: Segmentation and rendering aspects," *SPIE Medical Imaging V: Image Processing*, Vol. 1445, pp. 54-65 1991.
3. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *International J. Computer Vision*, 1987, pp.321-331
4. S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans.PAMI*, vol. 11, pp. 647-693, 1999.
5. P. P. Vaidyanathan, "Multirate Systems And Filter Banks," *Prentice Hall, Inc.*, 1993.
6. L. Cohen, "Time-Frequency Distributions - A Review," *Proc. IEEE*, Vol. 77, pp. 941-981, 1989..
7. A. Kaufman, "Volume Visualization," *IEEE Computer Society Press*, 1991.