Ophthalmologic Image Segmentation and Surface Visualization

Abstract

In this paper we present a practical approach to interactive segmentation of human eye magnetic resonance images (MRI) and human optical system surface reconstruction. A prototype software has been implemented and therefore this paper describes all the necessary aspects of the proposed method, including the use of preprocessing, user interaction, snakes \ active contour model and 3d visualization. Practical results are analyzed and further improvements are reasoned out.

1 INTRODUCTION

Image segmentation software designed specifically for the needs of ophthalmologists can become a great aid in semi-automating the analysis of magnetic resonance images (MRI) of human eye for every day medical practice. That is why it is of an immediate interest to introduce an effective approach for segmentation and surface reconstruction of human optical system.

The approach consists of two main steps: the segmentation step, which includes the extraction of image elements (pixels) belonging to an eye ball and the surrounding anatomical structures (e.g. optical nerve) and the reconstruction step, which includes the integration of the extracted elements into the representation of a surface.

The core logic of segmentation step is based on widely spread Active Contour Models (Snakes) first introduced by Kaas (1987). A generic boundary-searching snake functions by iteratively deforming its contour in an effort to minimize the contour energy, which is based on combination of internal and external forces. A contour is composed of a finite number of connected snake points (also called 'snaxels') and can be parametrically described as

$$v(s) = (x(s), y(s)),$$

where s[0...1] and the contour is closed if

$$v(0)==v(1).$$

But, instead of the original snake we choose a 'greedy snake' (Lam 1994) variation as a starting point. There are two main problems associated with basic greedy snake algorithm: it does not work well with noised images and there's a tendency for a contour to be unstable. Both problems have known solutions and can be easily dealt with. It is possible to greatly reduce noise susceptibility of the greedy snake by implementing a gaussian pyramid. We take the image and create a number of blurred (locally averaged) and scaled down copies of it, then the snake begins its spatial and temporal evolution starting from the image copy with the lowest resolution and over a given number of iterations propagates to higher resolution. As the result the snake is able to pass high frequency noise parts of the image and more easily reach the desired shape. The desired shape is reached more easily also because the gradient capture range is increased after we blur the image. Contour instability takes place when the desired shape is found, but at least one point twitches a little and - because all snake point are connected and affect their neighbours – a chain reaction is started, which leads to an inefficient long-term contour twitching. This can be fixed by processing the points not in a sequential order, but in a random one – which eliminates the problem, but produces slightly different results each time with identical starting conditions.

As the prototyped software is aimed at interaction with users it is important to provide a streamlined workflow and an array of tools to facilitate extracting anatomical features. Active contour model heavily depends on initialization, so having several modes which allow the snake to be initialized – both manual and automatic – is an absolute requirement. Medical three-dimensional images often come in stacks, so having the ability to switch between different slices and projections is necessary. Also the reconstruction step implies means to manipulate the three-dimensional view.

It is evident that anatomical structures can be explicitly defined by a surface or boundary representation given by a mesh of polygons. It is possible to construct such mesh from contours registered on each slice in MRI stack, but this is not always convenient and also introduces several topological issues: we have to make sure that there's only one contour per anatomical structure on each slice and that the number of snaxels in each neighbouring contour is the same. One more option would be to extend the greedy snake algorithm to three-dimensional space, where a contour would be represented as a balloon.

2 METHOD

The following phases for interaction with the system are provided: preprocessing, 2d\3d snake initialization and evolution, polygon surface reconstruction.

Preprocessing includes filtering the image just before the initialization step to facilitate better results.

A snake can be initialized automatically (snake points are positioned to form an ellipse or some other predefined contour in the center of an image) and manually by the user. The user plots a contour, places individual points and pulls them around with the help of a mouse until a desired contour is formed.

An interactive system consists of 4 viewport: 3 two-dimensional views (axial, coronal, sagittal projections) and a 3d view. The user can scroll through slices in a stack, rotate the 3d view and pan it.

In greedy snake each snaxel is computed separately by operating in a neighbourhood of a given size (e.g. 3x3, 5x5 etc.). At first an image gradient is generated, which is then used in calculation of energy in every pixel in the neighbourhood. The energy is computed as shown by (Lam 1994):

$$E = alpha*E_{cont} + beta*E_{curv} + gamma*E_{grad},$$

where E_{grad} is an image gradient:

$$\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)$$

$$\frac{\partial I(x,y)}{\partial x} \approx \frac{I(x+1,y) - I(x-1,y)}{2}$$

$$\frac{\partial I(x,y)}{\partial y} \approx \frac{I(x,y+1) - I(x,y+1)}{2}$$

and *alpha,beta, gamma* are weighting coefficients determined by the user that show the contribution of E_{cont}, E_{curv}, and E_{grad}. E_{cont} is related to the distance between a pixel and its neighboring snaxels relatively to the rest of the snaxels. To calculate it we just subtract the total average distance from the distance between the pixel and the previous snaxel. E_{curv} is related to the amount of curvature in the given pixel. If it exceeds a threshold for the current pixel (and if the gradient also exceeds a threshold) – then the beta weighting coefficient for this particular pixel is set to 0.0 which allows to obtain sharp corners in an otherwise elastic contour. By estimating energies for all neighbouring pixels, the processed snaxel is transformed into position of a pixel with the smallest overall energy. It should be noted that snake iterating process terminates if either too many iterations have been performed (controlled by a variable set by the user to prevent infinite looping) or too little amount of snaxels have moved (also controlled by a special variable to have a way to notice that the active contour has reached a stable state).
Snaxels are drawn as a rectangular group of pixels and connected by segments plotted via Bresenham line algorithm. Every snaxel can leave a trail behind itself (if the user turns this options on), so it is easier to see how the snake is evolving in terms of time and space.

3d greedy snake (greedy balloon) as a surface model is able to approximate the data across all slices. This active contour model is quite similar to 2d greedy snake in many ways: it consists of a finite number of points, the initial balloon position is set manually by the user. An image gradient is also computed, but in case of a balloon it is 3d gradient which is calculated for each pixel in every slice, working not in 2 (x,y), but in 3 dimensions (x,y,z), where z is a slice id.

A good example of a 3d snake implementation is demonstrated in (Twente Univ. 2008).

Reconstruction step implementation depends on whether 2d or 3d snake was used during the segmentation step. In the first case, which is quite trivial, the resulting contour for each slice is connected with its neighbouring contours via triangle stripes. In the second case, a polygonal mesh of the resulting 3d baloon can be made to adjust its topological structure or smoothed via interpolating subdivision for better visual representation. A good reference on subdivision is (California Institute of Technology 1998).

"Good" (i.e. successfully extracted and approved by the user) contours are saved and an averaged contour is calculated, so it is possible to get a generalized contour for a shape being studied (e.g. an eyeball etc.).

3 RESULTS

A prototype software for ophthalmologic image segmentation and surface visualization has been developed and the proposed approach has been tested.
It should be noted that at least one of the possibilities for improvement in this project lies in the area of using the raw computing power of GPU to increase the speed of segmentation and reconstruction process. The other possibility may be to refine the implemented methods of surface reconstruction or include more sophisticated ones. The method can be used for pre-surgery planning. The work was partly supported by Belarusian Foundation of Fundamental Research, Project F09-152.

Reference

California Institute of Technology (1998) "Subdivision for Modeling and Animation".

Kaas, M., A. Witkins, D. Terzopolus (1987), "Snakes-Active Contour Models" International Journal of Computer Vision, Vol. (1), No. 4, pp. 312 – 330.

Lam, K.-M. and H. Yan (1994), "Fast greedy algorithm for active contours." *Electronics Letters.* v. 30, n. 1, pp. 21-23.

Twente Univ. (2008). http://www.sas.el.utwente.nl/open/research/medical_imaging/snakes.