HIST - AN APPLICATION FOR SEGMENTATION OF HEPATIC IMAGES

Daniel Reska, Marek Krętowski

Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: HIST (Hepatic Image Segmentation Tool) is a Java-based application for segmentation and visualization of medical images, specialised for hepatic image analysis. This paper contains an overview of the application features, a description of adapted segmentation algorithms and their experimental validation. The application provides two main segmentation tools, based on region growing and active contour model methods, adapted for the case of liver segmentation. HIST also offers data visualization tools, including multiplanar reconstruction, volume rendering and isosurface extraction.

Keywords: liver segmentation, active contour, region growing, volume rendering, multiplanar reconstruction, isosurface extraction

1. Introduction

Medical imaging [17] is one of the fundamental tools of modern medicine. The ability of non-invasive exploration of internal aspect of various body parts is invaluable in research and clinical practice. One of the basic tasks in medical image analysis is the segmentation of interesting structure for further evaluation, such as diagnosis or surgery planning [16].

Image segmentation consists in partitioning an image into a set of separated regions that differ by some specific characteristic, such as intensity or texture. It is one of the most necessary but difficult tasks in computer vision, especially in the analysis of medical images. In this case, the variety of imaging methods and their applications imposes a necessity of customisation of the methods for each specific task, which makes the development of more robust techniques challenging.

Liver segmentation is a particularly difficult task, even for an expert [6]. Segmentation tools have to deal with the irregularity of the organ shape and of their boundaries, the intensity variation due to anatomical complexity, the pathologies and

Zeszyty Naukowe Politechniki Białostockiej. Informatyka, vol. 7, pp. 71-93, 2011.

neighboring of other organs - mainly the heart, stomach or rib cage structures (see Fig. 1). The interpersonal variance of the liver shape is also a problem for statistical model-based algorithms. All these factors make liver segmentation especially challenging.

In this paper, we present HIST (Hepatic Image Segmentation Tool) - an application for segmentation and visualization of medical images with tools adapted specifically for liver segmentation tasks. Many other software frameworks and complete applications with similar functionalities are available [19],[24],[4]. Most of them, however, implements only general-usage tools that are not suited for any specific purpose and usually require significant customisation. In the case of liver segmentation, many advanced methods have been developed and are proven effective [10]. In practical evaluation the methods should be examined in a consistent environment which would simulate their real-life usage.

HIST was developed with emphasis on providing out of the box tools for liver segmentation and further visualization and evaluation of the results. Furthermore, the application is aimed at being used by medical doctors, and therefore user experience in practice was also an important factor.

The rest of the paper is organised as follows. Section 2 describes the hepatic segmentation methods, visualization tools and other features of the application. Section 3 contains the results of experimental validation of the tools. Finally, Section 4 presents conclusions and directions of future research.

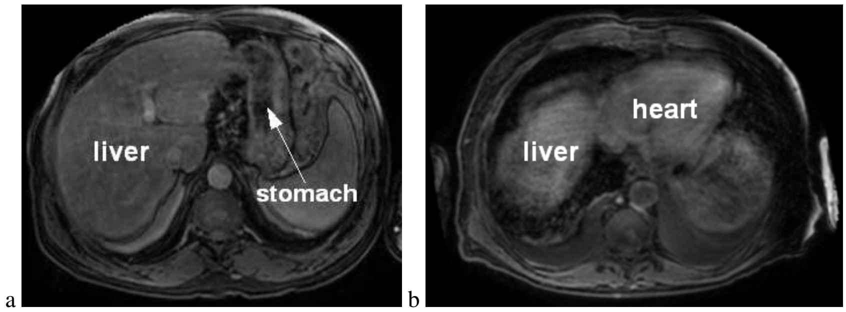


Fig. 1. Problematic liver position near other organs: stomach (a) and heart (b)

2. Hepatic segmentation and visualization

The main segmentation tools implemented in the application are based on two methods: region growing and active contour model. These techniques, commonly used in medical image analysis, were adapted to the specific task of hepatic MRI segmentation. The main goal in the adaptation process was to create usable real-time segmentation tools, that could be used and evaluated in a consistent environment.

2.1 Region growing

The first segmentation tool is seeded region growing [1] and merging algorithm. The base idea of this method is to initialise a set of pixels in the image domain and expand it by adding new pixels that meet specific criteria. Similar resulting regions are merged and post processed. Seed points initialisation can be performed manually or with specialised automated methods.

Seed points initialisation Region growing algorithms are particularly sensitive to the initial location of the start points. Manual initialisation is usually a tedious task, especially in the case of large data sets, therefore an automatic initialisation method was created. The method is based on split and merge algorithm [11], which divides the image into regions with uniform intensity and generates seed points from their centres. The dark regions around the body volume are excluded (see Fig. 2). These points can be used for segmentation of the whole image, but in the case of liver segmentation other constraints are also applied. The user can place a bounding box around the body volume and mark a uniform liver region with a cursor (see Fig. 3). The bounding box contains two regions with the highest probability of where the liver is located. The size and position of the region are based on proportions proposed in [6], although they were slightly modified as a result of experimental validation. A seed point of an area located in these regions and similar to the area pointed by a user is accepted as a potential liver point. This technique not only speeds up the initialisation process, but also increase its reproducibility.

Growing and merging The next phase of the method is growing of the seed points. Original regions (composed of a chosen pixel and its 3x3 neighbourhood) are expanding by adding new pixels that meet specific criteria. A set of new pixels P_{new}, added in a single grow cycle, can be described as:

$$P_{new} = \{p \in N_{region} : |I(p) - \mu| \leq k\sigma\},$$

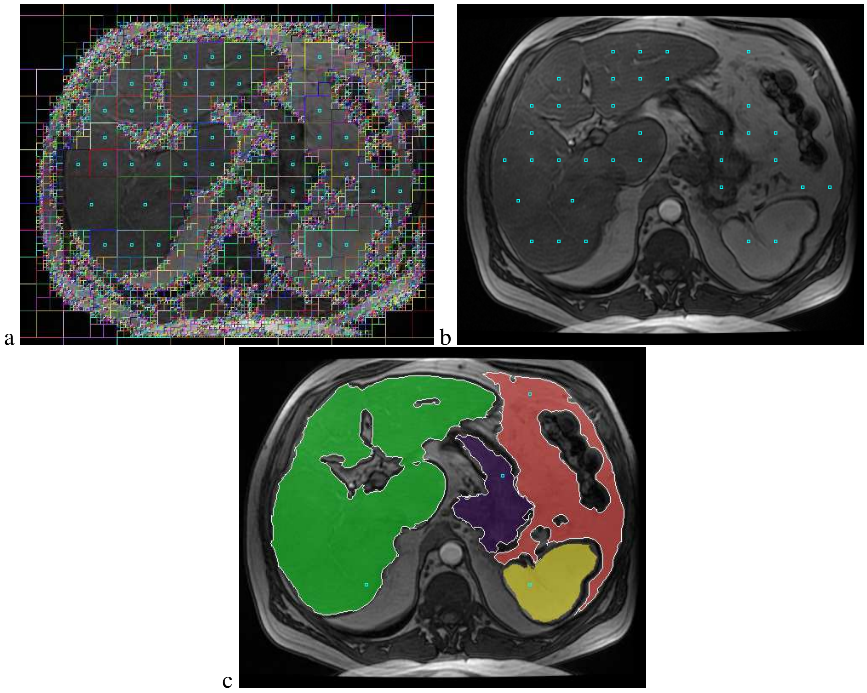


Fig. 2. Automated seed point initialisation: image divided into homogenous regions (a), generated seed points (b) and segmentation result (c)

where $I(p)$ is the intensity of pixel p, μ and σ are the mean and standard deviation of the intensity of the start region, k is a user-defined constant and N_{region} is a set of pixels adjacent to original region. In this method, a different approach was used. New points are added by analysing the neighbourhood of every existing region pixel p_{reg}. A new pixel p_{new}, adjoining to p_{reg}, can be added to the region only if it meet two conditions: the intensity difference between p_{reg} and p_{new} cannot exceed a given threshold and the intensity of p_{new} must be sufficiently similar to the start region intensity mean:

$$P_{new} = \{p_{new} \in N_{p_{reg}} : |I(p_{reg}) - I(p_{new})| \leq T_{adj} \ \wedge \ |I(p_{new}) - I_{start}| \leq T_{start}\},$$

where $I(p_{new})$ and $I(p)$ are the intensities of p_{new} and p_{reg}, T_{adj} is the intensity difference threshold, I_{start} is the original 3x3 region intensity mean, T_{start} is the start mean threshold and $N_{p_{reg}}$ is the neighbourhood of p_{reg}. The second condition prevents the "leakage" on a low gradient boundary of two regions, where the intensity difference

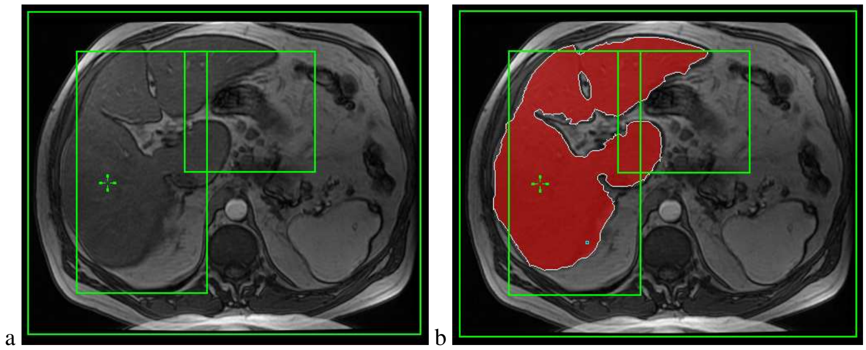


Fig. 3. Automated seed point initialisation with liver positioning constraint: position of bounding box and marked region (a) segmentation result (b)

of adjacent pixels is not sufficient to stop the growing. Algorithm iteration count can also be limited (see Fig. 4).

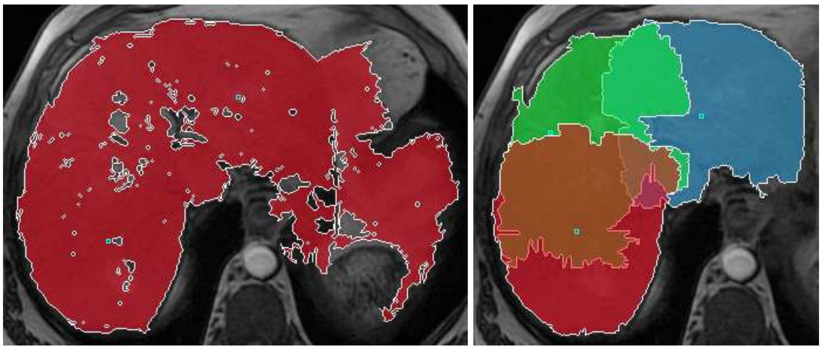


Fig. 4. Iteration limitation used to prevent leakage of a region: result of grow and merge of three seed without limitation (left) and result with decreased iteration count (right)

Merging is performed by calculating the overlap ratio. Two regions are merged if they contain a sufficient percent of shared pixels (see Fig. 5). The default merge ratio is 90%.

Postprocessing The regions segmented with region growing algorithms usually contain many internal discontinuities and border irregularities, therefore final enhance-

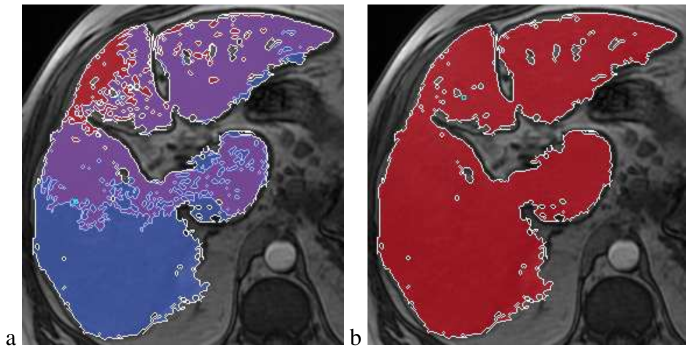


Fig. 5. Region growing and merging: result from three seed points (a) and merged regions (b)

ments are crucial. Apart from manual editing tools, three postprocessing methods were implemented:

- simple sealing by finding the pixels between the two outermost ones in every row and column of the region and filling the points present in both of these scanline runs - the simplest, fastest but least accurate method;
- morphological closing - more accurate, although with a tendency to make undesirable conjunctions in the region;
- classification of each pixel by analysing the count of region points in the pixel neighbourhood - the slowest but the most accurate technique.

The results of these methods are presented in Fig. 6.

2.2 Active contour

Active contour (snake) [12] is the second main segmentation tool. Its original representation is a parametric curve, which deforms under influence of internal and external forces. The goal of the evolution process is to minimise the total energy of the snake. With the contour defined as $v(s) = (x(s), y(s))$ where $s \in [0,1]$, total snake energy could be written as:

$$E^*_{snake} = \int_0^1 E_{snake}(v(s))ds = \int_0^1 E_{int}(v(s)) + E_{image}(v(s)) + E_{con}(v(s))ds \quad (1)$$

where $E_{int}$is the internal energy (controlling bending and stretching), E_{image} is the image force (moving the snake towards desired features) and E_{con} represents other possible constraints.

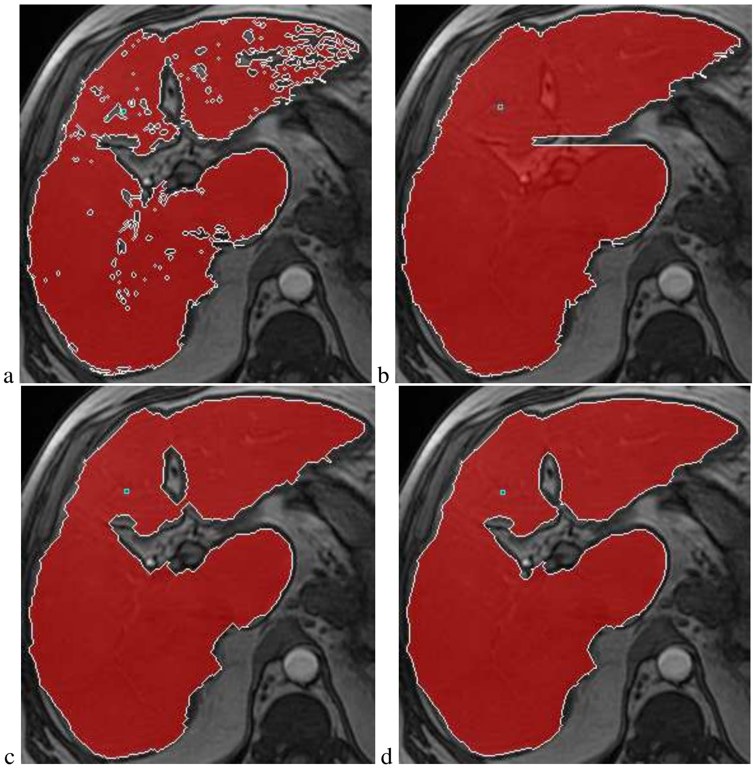


Fig. 6. Postprocessing of segmentation result: original region (a), scan fill with undesirable inner region inclusion (b), morphological closing with undesirable conjunction (c) and neighbourhood analysis (d)

The implemented model is a discreet form of the curve, composed of a set of points (snaxels). The total energy of the snake is minimised by moving each snaxel to a position of minimum local energy. The three main elements of the model are:

- balloon force, based on image properties and contour shape;
- image energy, minimising snake energy in the image domain;
- internal energies, responsible for flexibility, tension and topology of the curve.

Apart from a manual segmentation mode on a single image, this tool also has the ability to perform fast, semi-automatic segmentation on a series of images by contour propagation. The snake can operate directly on the source image or can use its gradient amplitude, calculated with custom liver-adapted filter.

Balloon force The fist main element of the developed model is a balloon force [7] that inflates the contour and pushes it towards desired image elements. This method overcomes the limitations of the original snake model [12], which has a limited scope and have to be placed close to the segmented object boundary. A snake with the inflation force can be easily placed inside the segmented area without the need of precise initialisation. This feature is particularly helpful in liver segmentation, because of the its large surface and boundary length (see Fig. 7).

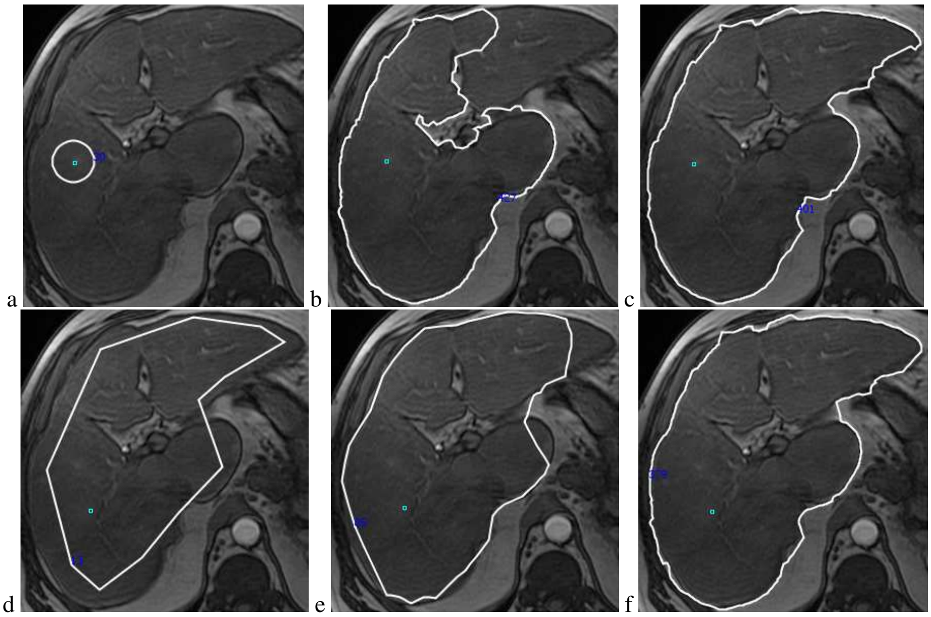


Fig. 7. Initialisation insensitivity: small snake placed inside the region (from a to c) gives similar result to more accurate initialisation (from d to f)

The balloon force needs a direction for moving the snake points. One of the most common approaches is to use the normal vector of each snaxel [7]. In the developed model, a different method was used. For a simple shape with a small points count (less than 30) the central point of the curve is used to calculate the growing directions. For more complicated curves, the vectors are based on the skeleton points [14] of an approximated simplified version of the curve. The vector of each point is constructed from its position and the closest skeletal points (see Fig. 8). In comparison with the normal vectors calculation, this approach results in a more uniform points distribution during the growth and helps in preserving the correct topology. Along with the dynamic topology modification, it also speeds up the process of adaptation to complicated shapes of segmented regions (see Fig. 9).

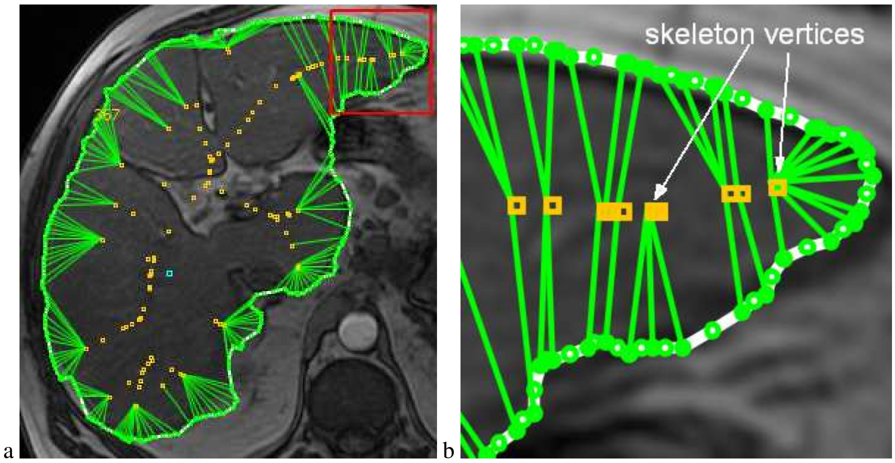


Fig. 8. Visualization of balloon force vectors: expansion directions are calculated from positions of each snaxel and the closest skeleton vertex (visible unconnected vertices were generated as a result of topology modification and will have their vectors calculated in the next iteration)

Movement of the snake points is constrained with conditions similar to the region growing method described earlier. Again, a snake point can advance along its directional vector when it meets the criteria of intensity difference between its current position, destination point and the start region. Additionally, contour points can not be moved into the area already covered by the snake and the maximal movement distance in one iteration can be limited. This force was customised to achieve fast expansion behaviour, taking into account the relatively large area of the liver. The high rate of the inflation can result in irregularities in the curve shape, therefore several topology optimisation procedures were developed.

Image energy Image energy is the second main element of the developed model. It puts a snake point in the position of the lowest energy within its local neighbourhood. The energy is measured by the image gradient amplitude value: a pixel with the highest gradient value has the lowest energy. The thickness of the liver boundaries often results in the occurrence of many highest valued pixels in the snake point boundary. In this case the pixel with the closest or furthest distance from the snake point, de-

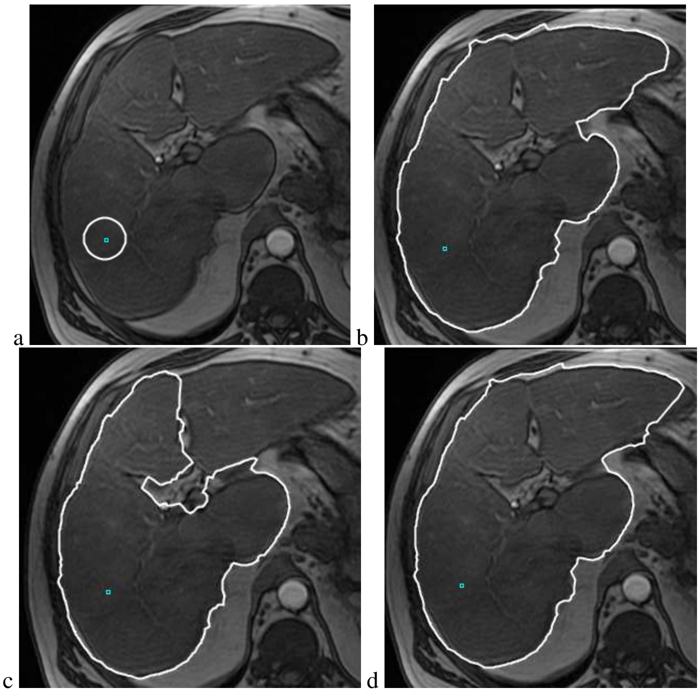


Fig. 9. Influence of the type of balloon force on the segmentation time: the evolution of the original snake (a) after 50 grow cycles using the skeleton-based balloon force (b) is significantly faster than the adaptation using the normal-based force (c), which needs another 50 iterations to equal the first method (d)

pending on user preference, will be chosen (see Fig. 10). This property allows the snake to contract or expand on the most significant boundaries, which is useful in the contour propagation process.

Topology optimisation Topology of the snake during the evolution process undergoes a constant optimisation. Internal constraints forbid the curve points from getting into undesirable locations. Points cannot move to a position already occupied by the snake surface, but can still share the same location. This could lead to snaxel redundancy and creation of unwanted loops, therefore the optimisation algorithm is searching for overlapping points and removes unnecessary snaxels between them.

Internal smoothing and tension forces also affect the contour shape. Fast inflation using the balloon force may cause significant irregularities in the shape of the curve and for that reason internal constraints are necessary. Specifically, these forces

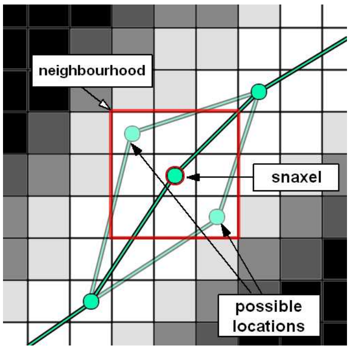


Fig. 10. Image force with possible snaxels positions: within a neighbourhood (marked by red square) with many image energy minimums, a snaxel can move to two outermost locations, resulting a small expansion or shrinkage of the curve

control the distance of a point from its neighbours points of the contour, preventing the point from moving too far, disrupting the snake smoothness and rigidity. These constraints can also be turned off for the points already positioned in their local minimum, affecting only the still-evolving snaxels and consequently speeding up the process.

New snake points are also added between existing snaxels, providing a simple subdivision scheme and allowing the curve to grow into further and more complex areas. Maximal point count can be limited, however usually the number of points quickly stabilises after the first few iterations and undergoes only minor oscillations during the evolution.

Contour propagation The presented model is capable of propagation over a series of images. This feature was used to implement a fast, semi-automatic segmentation tool. The user initialises a snake on one image. Then, the curve evolves and is copied to the next image, becoming the initial contour for the next snake. This process can be continued for the whole series of images (see Fig. 11). The deformation process in this case relies mainly on the image energy, because the pre-initialised snake from the previous image usually is already placed close to the boundaries. The process benefits from the image energy expansion/contraction preference, which can be set to correspond the actual change (growing/shrinking) of the liver region on consecutive slices.

The main limitation of this technique is the necessity of a small distance between slices in the data set. A relatively large spacing between subsequent images causes

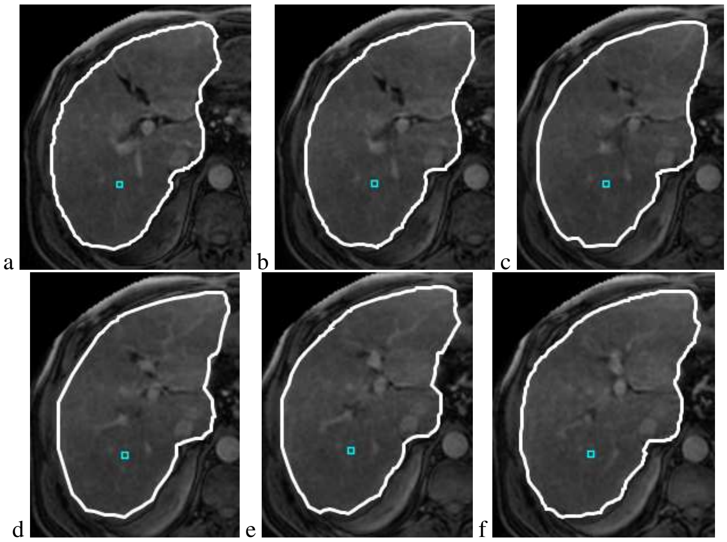


Fig. 11. Contour propagation over a series of images: manual initialisation on the first image (a) and a result of progression over 5 consecutive slices (from b to f)

abrupt changes in the segmented organ shape, making the reference contour less usable. In practice, this method was proven effective on sets with distance between two successive slices up to 3 mm.

2.3 Visualization

HIST contains three visualization tools: multiplanar reconstruction [20],[13], volume rendering [21] based on texture mapping [8],[5] and isosurface extraction with marching cubes algorithm [15]. All these methods can be used to present both segmentation result and the entire data set. These methods are commonly used in medical purposes and provide fast, real-time interaction.

Multiplanar reconstruction module enables visualization of the data set slices in other directions than the default one. The user can point a cursor to indicate the planes position and the reconstruction view will be instantly updated. An example with a segmentation result is presented in Fig. 12. Bilinear interpolation was used to achieve sufficient quality of low resolution sets.

Volume rendering of a loaded data set was achieved with texture mapping. The main idea of this method is to display a set of polygons with cross-sectioned images of the original data set mapped on. All three series of planes (transversal, coronal

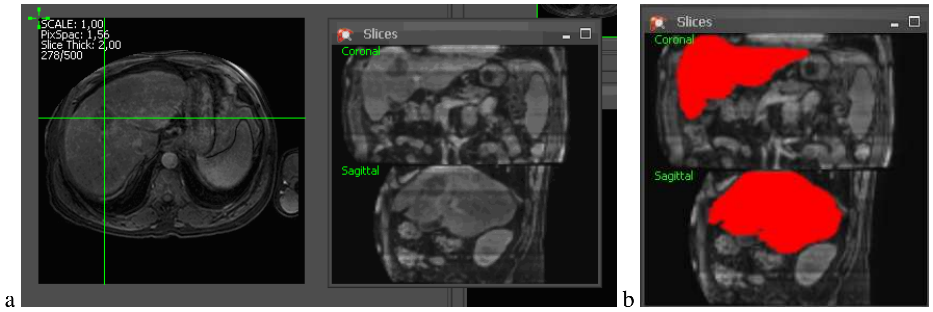


Fig. 12. Multiplanar reconstruction: original image with cross-section planes marker and reconstructed coronal and sagittal planes (a), reconstructed planes with marked segmentation (b)

and sagittal) are reconstructed with the multiplanar module. Combined planes are displayed using Java3D, which is an OpenGL wrapper. This enables fast hardware accelerated rendering and interaction. The set can be viewed from arbitrary angle and opacity characteristics can be instantly changed (see Fig. 13 and 14). Dynamic plane-switching was implemented to compensate the resolution differences between axes.

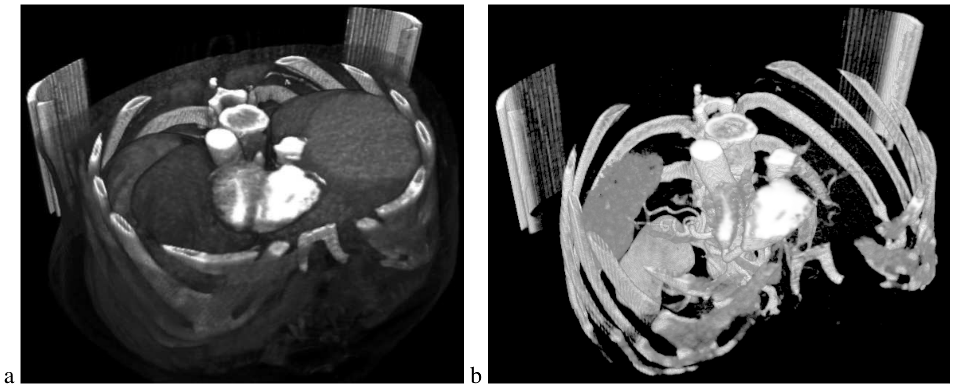


Fig. 13. Example of transparency modification usage: volume rendering of an entire abdominal CT (data courtesy of OsiriX [23]) (a) and a tissue separation achieved by dynamic transparency adjustment (b)

Isosurface extraction with marching cubes algorithm [15] is the last visualization method available in HIST. Unlike volume rendering, this method takes only a specific part of the data and converts it into a triangle mesh. The main challenge

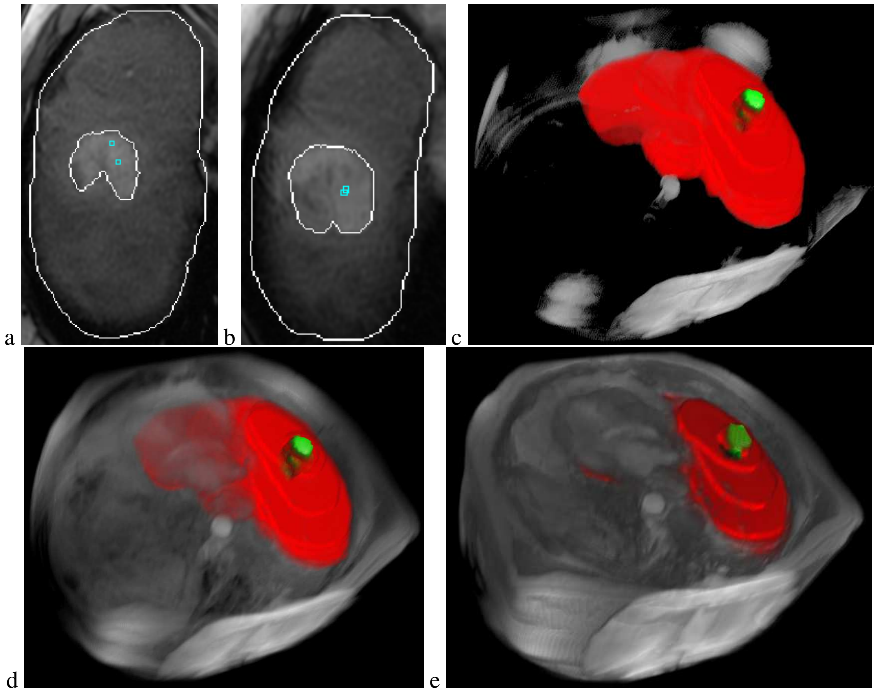


Fig. 14. Visualization of liver volume (red) and pathology (green): regions segmented on original images (a and b) and volume rendering in various transparency modes (from c to e)

of this method is to achieve a correct smoothing of the final mesh, which can be generated from the entire data set or only from the segmentation result. In the first case, smoothing can be easily performed by interpolating the values of each boundary voxels. Unfortunately, segmentation results are provided in the form of binary maps. Usage of specific smoothing algorithms is then necessary [18]. Furthermore, low spatial resolution of data sets can cause characteristic stairway artifacts in the final mesh. Moreover, the basic shape of the segmentation result has to be closely preserved and therefore simple smoothing algorithms are unacceptable from a medical point of view. The smoothing is performed with one iteration of morphological dilation, resulting in one pixel-wide smooth boundary around the region. Intensity $I_p(i)$ of each boundary pixel is calculated as:

$$I_p(i) = I_1 + I_2(s_i/s_{max}),$$

where I_1 and I_2 are predefined constants, s_i is the count of voxels adjacent to the i pixel, and s_{max} is the maximal count of possible adjacent voxels in 3x3x3 neighbourhood cube. The binary format of the region map imposes that the $I_p(i)$ should take a value between 0 and 1, therefore $I_1 + I_2 = 1$. Sampling factor of the region can be also altered to create more consistent map. The results of the developed method are presented in Fig. 15.

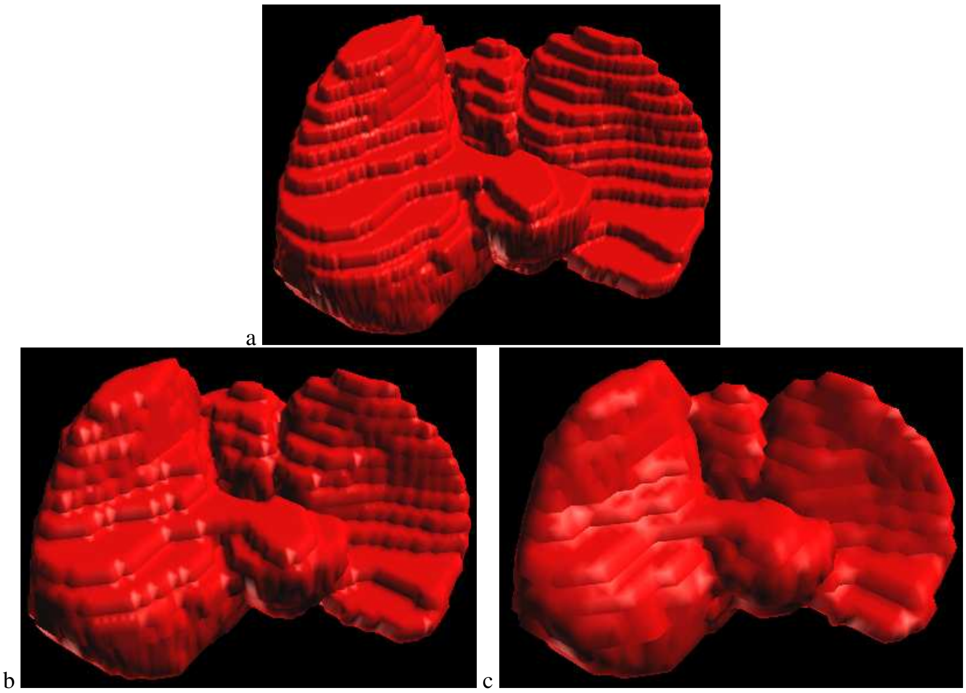


Fig. 15. Smoothing of a segmented liver isosurface: original result with visible artifacts (a), smoothing with factors of 4 (b) and 6 (c)

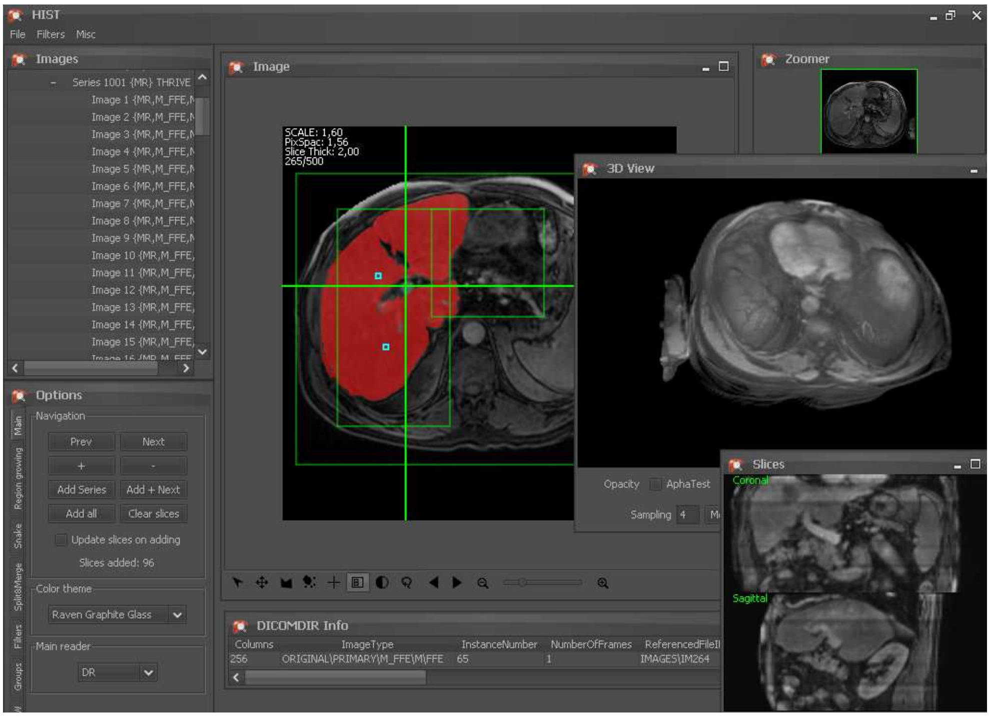
2.4 Application overview

The application was implemented in Java SE 6 platform, using Swing and Java3D libraries. This choice was based on availability of many useful, free libraries and cross-platform support.

The main tools proposed by our application are:

- browser for DICOM files (single/multi-frames, image directories and DICOMDIR files) and plain image files (BMP, JPEG, PNG);

- various image filters: bilateral filter [25], morphological closing, median filter, edge detection, histogram equalisation, thresholding, gray scale quantization, LUT colour palette applying and more;
- manual and automated segmentation with region growing, active contour and LiveWire [3] (on single and multiple images);
- segmentation results editing, grouping, saving, and merging;
- multiplanar reconstruction module;
- interactive 3D visualization of data sets and segmentation results with volume rendering and isosurface extraction;
- segmentation results analysis and comparison (described in Section 3);
- customisable user interface with many themes (see Fig. 16).



SCALE: 1,60
PixSpac: 1,56
Slice Thick: 2,00
265/500

☐ Update slices on adding

Slices added: 96

Color theme: Raven Graphite Glass

Main reader: DR

☐ AphaTest

Sampling 4

Columns	ImageType	InstanceNumber	NumberOfFrames	ReferencedFileI
256	ORIGINAL\PRIMARY\M_FFE\M\FFE	65	1	IMAGES\IM264

Fig. 16. Application main window with visible segmentation result, 2D multiplanar reconstruction and 3D visualization

2.5 User interface

The application had to be equipped with a comprehensible user interface. Many segmentation methods propose many parameters, making specific adjustments complex. Some works have shown that a user can simultaneously manipulate only up to four parameters [9] of solved task. This problem was taken into consideration and options of every implemented tool were divided into two sets: a small number of basic parameters and a set of advanced options, that could be understood during the usage of the application. Default values of these parameters were also adjusted to the specific task of liver segmentation.

3. Experimental validation

A preliminary validation of the implemented tools was performed, in which the segmentation time and quality was tested. Ideally, the usage of the tools should lead to a significant shortening of the segmentation time while maintaining the quality of the results.

3.1 Data sets

Quality and efficiency of the implemented tools were tested on various hepatic MRI sets, gathered at the Pontchaillou University Hospital, Rennes, France. The available data sets where generally divided into two groups: a smaller series of higher resolution images (usually over a dozen of 512x512 px) and smaller resolution sets of about 100 of images in series. Two representative sets were chosen:

- Set 1, containing 18 512x512 px images with 0,74x0,74x8,5 mm voxel size;
- Set 2, containing 100 256x256 px images with voxel size of 1,56x1,56x2 mm.

3.2 Evaluation

Segmentation quality was measured with two commonly used [10] error measures. The first one is the Overlap Error (OE), defined as:

$$OE(A,B) = 100(1 - (|A \bigcap B| / |A \bigcup B|)) \tag{2}$$

where A and B are two segmented pixel sets. The 0 value indicates that the two sets are identical and 100 that the sets do not overlap.

The second measure is the Relative Volume Difference (RVD), defined as:

$$RVD(A,B) = 100((|A| - |B|)/|B|) \tag{3}$$

where A is the tested segmentation and B is the reference. This measure can indicate a tendency to over- or undersegmentation of the method. It must be used along with other measures, because the actual sets overlap is not considered.

3.3 Results

Reference segmentations, used in evaluation of the examined methods, were performed made by the application author using the implemented manual tools. The intrapersonal variation was also an important factor, therefore several manual segmentations of each set were performed with 24-hour interval. Thereafter, the sets were evaluated with each other using the described measures. Set 1 was segmented using the automated region growing method and the propagating active contour was tested on Set 2. The averaged comparison of the results are enclosed in Table 1. Table 2 contains average time of manual and tool-guided segmentation. Figures 17 and 18 presents sampled tool-guided segmentation results along with the manually outlined referential areas.

Table 1. Manual and tool-guided segmentation quality

	Manual		Tool-guided	
Data set	OE	RVD	OE	RVD
Set 1	7,26 ± 4,46	-2,35 ± 7,59	10,94 ± 8,56	-3,14 ± 10,68
Set 2	7,84 ± 3,98	-1,35 ± 5,41	10,1 ± 4,29	-4,49 ± 5,56

Table 2. Segmentation time (in minutes)

Data set	Manual seg.	Tool-aided seg.
Set 1	25-30	5-10
Set 2	40-45	6-8

3.4 Outcome

Experimental results show a significant decrease in segmentation time. For instance, in the case of Set 2 from 45 minutes for manual segmentation to less than 10 minutes

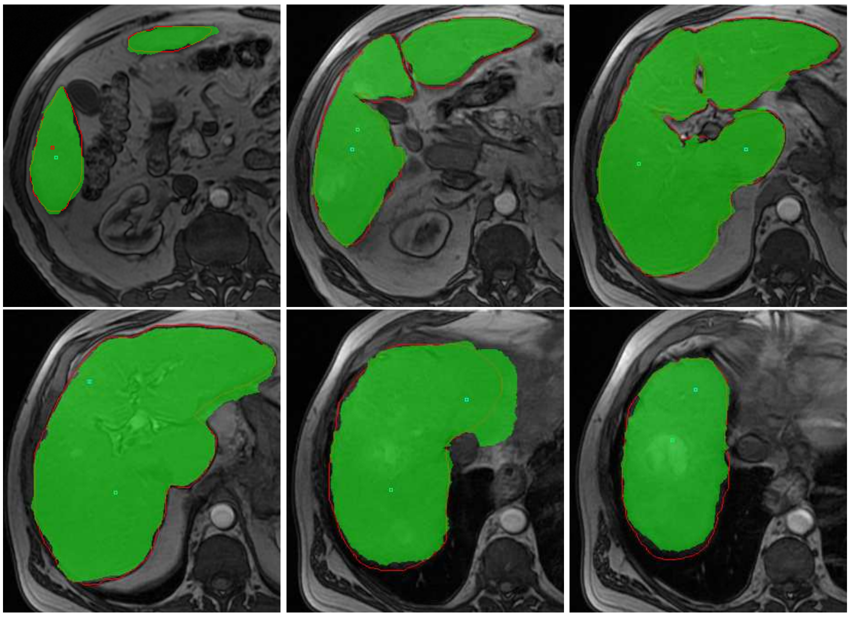


Fig. 17. Set 1 segmentation sample: the greed regions are the result of the automated region growing segmentation and the referential region boundaries are marked in red

using the implemented tools. It has to be noted that these results depend heavily on personal experience and manual abilities, especially in the case of manual segmentation. As for the quality of the segmentation, the average Overlap Error was about 10% and the Relative Volume Difference was -2,69%. High values of the standard deviation of these results was caused mainly by the ambiguous cases in the outermost images of series, where the organ segmentation was particularly difficult because of the partial volume effect [2]. Obtained results were compared to the results of hepatic CT segmentation from MICCAI 2007 Grand Challenge [10]. For interactive methods, OE was 8% ± 2,6% and RVD was equal -2,81% ± 3,62%. Intrapersonal variation results was also remarkable, with 7,7% for OE and 1,8% for RVD.

A general tendency to undersegmentation in the developed methods was noted. Fortunately, the results of semi-automated methods are never conclusive and in the most cases can be easily corrected with available enhancement tools, preserving the time reduction.

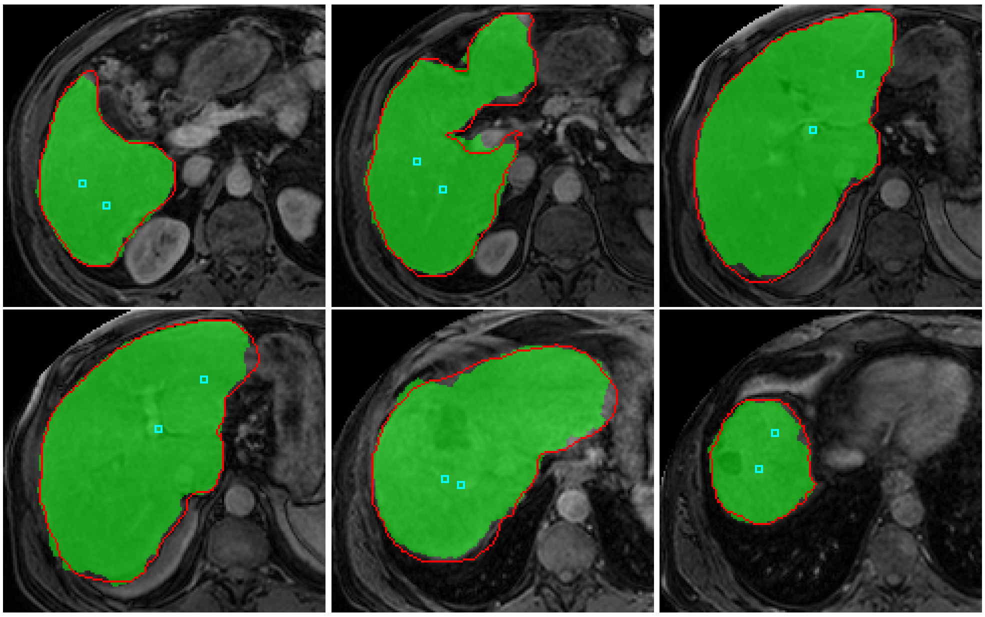


Fig. 18. Set 2 segmentation sample: the green regions are the result of the active contour segmentation and the referential region boundaries are marked in red

4. Conclusion and future work

Segmentation tools implemented in the presented application where proven to be useful in the challenging case of liver segmentation. Despite the early stage of their development, these methods have given promising results that encourage further study. Contour propagation was particularly effective in the case of large data sets and region growing was useful in the case of larger resolution images, that required more precise handling. However, the complexity of liver segmentation makes stronger generalisation of this kind difficult and is forcing the study of different methods. Currently, the highest priority task is the evaluation of the implemented tools on real-life cases in medical environment.

HIST provides a stable base for further improvement of segmentation and visualization algorithms. Current state-of-the-art techniques provides a wide variety of potentially effective methods [10]. Apart from expanding the current single-image based tools, implementation of methods working on full 3D data set is also planned. The visualization techniques can also benefit from greater usage of hardware acceleration on modern GPUs [22].

Acknowledgements

The authors are grateful to Prof. Johanne Bézy-Wendling for inspiring discussion and useful comments.

References

[1] Adams R., Bischof L., Seeded region growing, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 16(6):641 –647, 1994.

[2] Ballester M.A., Zisserman A.P., Brady M., Estimation of the partial volume effect in MRI, Medical Image Analysis, 6(4):389 – 405, 2002.

[3] Barrett W., Mortensen E.N., Interactive live-wire boundary extraction, Medical Image Analysis, 1(4):331–341, 1997.

[4] Brigham and Women's Hospital, 3D Slicer, http://www.slicer.org Accesed at 2011-07-04.

[5] Cabral B., Cam N., Foran J., Accelerated volume rendering and tomographic reconstruction using texture mapping hardware, In Proceedings of the 1994 symposium on Volume visualization, VVS '94, pages 91–98, 1994.

[6] Campadelli P., Casiraghi E., Esposito A., Liver segmentation from computed tomography scans: A survey and a new algorithm, Artificial Intelligence in Medicine, 45:185–196, 2009.

[7] Cohen L.D., On active contour models and balloons, CVGIP: Image Underst., 53:211–218, 1991.

[8] Cullip T.J., Neumann U., Accelerating volume reconstruction with 3D texture hardware, Technical report, Chapel Hill, NC, USA, 1994.

[9] Halford G.S., Baker R., McCredden J.E., Bain J.D., How many variables can humans process?, 2004, University of Queensland, Brisbane, Australia, and Griffith University, Brisbane, Australia.

[10] Heimann T., van Ginneken B., Styner M.A., et al, Comparison and evaluation of methods for liver segmentation from CT datasets, IEEE Transactions on Medical Imaging, 28:1251–1265, 2009.

[11] Horowitz S.L., Pavlidis T., Picture segmentation by a directed split and merge procedure, In International Conference on Pattern Recognition, pp. 424–433, 1974.

[12] Kass M., Witkin A., Terzopoulos D., Snakes: Active contour models, International Journal of Computer Vision, 1(4):321–331, 1988.

[13] Kramer D.M., Kaufman L., Guzman R.J., Hawryszko C., A general algorithm for oblique image reconstruction, Computer Graphics and Applications, IEEE, 10(2):62 –65, 1990.

[14] Lee D.T., Medial axis transformation of a planar shape, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-4(4):363–369, 1982.

[15] Lorensen W.E., Cline H.E., Marching cubes: A high resolution 3D surface construction algorithm, Computer Graphics, 21(4):163–169, 1987.

[16] Meinzer H.P., Thorn M., Cardenas C.E., Computerized planning of liver surgery - an overview, Computer and Graphics, 26(4):569–576, 2002.

[17] Mudry K.M., Plonsey R., Bronzino J.D., Biomedical imaging, CRC Press, 2003.

[18] Neubauer A., Forster T.M., Wegenkittl R., Mroz L., .Bühler K., Interactive display of background objects for virtual endoscopy using flexible first-hit ray casting, VisSym (Joint EG - IEEE TCVG Symp. on Visualization):301–304, 2004.

[19] Rasband W., ImageJ - image processing and analysis in Java, http://rsbweb.nih.gov/ij/ Accesed at 2011-07-04.

[20] Rhodes M.L., Glenn W.V., Azaawi Y.M., Extracting oblique planes from serial ct sections, J. Comput. Assist Tomogr., 4(5):649–657, 1980.

[21] Sabella P., A rendering algorithm for visualizing 3D scalar fields, In Proceedings of the 15th annual conference on Computer graphics and interactive techniques, SIGGRAPH '88, pp. 51–58, New York, 1988.

[22] Shihao Ch., Guiqing H., Chongyang H., Rapid texture-based volume rendering, In Environmental Science and Information Application Technology, 2009, ESIAT 2009, International Conference on, volume 2, pp. 575–578, 2009.

[23] OsiriX Software, DICOM sample image sets, http://pubimage.hcuge.ch:8080/ Accesed at 2011-08-21.

[24] ITK-SNAP Team, ITK-SNAP home page, http://www.itksnap.org Accesed at 2011-07-04.

[25] Tomasi C., Manduchi R., Bilateral filtering for gray and color images. In ICCV '98: Proceedings of the Sixth International Conference on Computer Vision, IEEE Computer Society, 1998.

HIST - APLIKACJA DO SEGMETNACJI OBRAZÓW WĄTROBY

Streszczenie HIST (ang. Hepatic Image Segmentation Tool – narzędzie do segmentacji obrazów wątroby) jest napisaną w języku Java aplikacją do segmentacji i wizualizacji obrazów medycznych, wyspecjalizowaną w segmentacji obrazów wątroby. Artykuł ten zawiera przegląd możliwości aplikacji, opis zaadaptowanych algorytmów segmentacji i wizualizacji oraz ich eksperymentalną walidację. Aplikacja oferuje dwie główne metody segmentacji, oparte o algorytmy rozrostu regionów i aktywnego konturu, dostosowane do segmentacji wątroby. Narzędzia wizualizacyjne aplikacji wykorzystają rekonstrukcję multiplanarną, rendering wolumetryczny oraz ekstrakcję izopowierzchni.

Słowa kluczowe: segmentacja wątroby, aktywny kontur, rozrost regionów, rendering wolumetryczny, rekonstrukcja multiplanarna, ekstrakcja izopowierzchni.

Artykuł zrealizowano w ramach pracy badawczej S/WI/2/08