

CHASM

Computer-Human Assisted Segmentation of Medical Structures

Manuel Cajada Coelho Lopes

Thesis to obtain the Master of Science Degree in

Biomedical Engineering

Supervisor: Prof. Joaquim Armando Pires Jorge

Supervisor: Dr. Maurício Lança Tavares de Sousa

Examination Committee:

Chairperson: Prof. João Miguel Sanches

Supervisor: Prof. Joaquim Armando Pires Jorge

Members of the Committee: Prof. Abel João Padrão Gomes

October 2021

Preface

The work presented in this thesis was performed at the Institute for Systems Engineering and Computers of Instituto Superior Técnico (Lisbon, Portugal), during the period March-October 2021, under the supervision of Prof. Joaquim Jorge.

Abstract

This thesis aims to implement a semi-automatic edge contouring method capable of segmenting high resolution cryosections of a cadaver available at the Visible Human Project, with minimal necessary user input. Initially, different image segmentation algorithms were explored, however, the use of graph cut based algorithms was the preferred method. An algorithm was then developed on top of OpenCV's Grabcut, which receives a manually drawn contour, corrects the segmentation and propagates it across an interval of slices. Pre and post processing methods were developed in order to improve computation time and segmentation performance. The contours were then saved in order to perform a reconstruction of the anatomical structure.

The results obtained were promising, total reconstruction of the femur was possible, along with other structures, such as the aorta and the liver. The time it took to achieve a full reconstruction depends on the size of the structure, computing around four slices per second for the aorta and one slice per two seconds for the liver. The resulting contours could be improved, as the color differentiation between some tissues is minimal. Some solutions were suggested that could enhance the contrast such as the using the additional radiological images available in the data set.

Keywords: Visible Human Project, OpenCv GrabCut, Python, Edge Contouring Algorithm, Computer Vision, Medical Imaging

1 Introduction

The use of human bodies for anatomical classes has been present in Europe since Late Middle Ages, and relied on jails and poorhouses as a source for bodies. Only later, in the 1960s, did body donation with informed consent became an alternative and has since been the primary method used for teaching gross anatomy to medical students[1]. It is still considered one of the most important methods of delivering fundamental anatomical knowledge to students in order to ensure safe clinical practice and also to cultivate humanistic values among the students. Nearly all clinicians still remember their first dissection of a human cadaver[2].

However, the whole process can be quite expensive, cadavers require a large laboratory and usually have short shelf lives if not prepared correctly. Body parts are hard to find and are often destroyed in the process. Not only that, but there has also been a shortage in cadavers for medical study due to higher demand. The number of medical programs is growing, organs are being used for transplants, and some bodies may end up in pharmaceutical companies or exhibits. Not only that, but also with the evolution of communication there are fewer unclaimed bodies for use[3][4].

As such, there has been a shift toward other teaching methods recently, such as the use of Virtual Reality. New tools are being developed that allow students to learn anatomy in new innovative ways, with 3D Reconstructions from anatomical slices (3DRAS). This has been made possible due to the large data sets being available, which allow for a total reconstruction of the body[5]. This brings many advantages, such as multiple viewpoints of anatomical structures and re-usability of the model, which were limitations from using real cadavers[6].

1.1 Motivation

As stated above, several tools are available that allow the user to manually segment data sets and provide the reconstructed object in 3D. Some use automatic/semi-automatic segmentation with edge contouring algorithms or artificial intelligence, by building and training machines that can perform the segmentation, while others rely heavily on manual delineation. The rising availability of cadaver data sets, such as the Visible Human Project, has also provided important resources for research on image segmentation.

Most user-oriented methods require heavy user interaction. However, pure manual contouring of the structures in a series of images is very time consuming. Moreover, user interfaces to interact with these models are still precarious and often in 2D. Lastly, the results obtained so far can have segmentation inaccuracies and are still not considered good enough to be used as a reliable tool in medical environments. Therefore, there is still a gap in the market for a reliable contouring algorithm that can be used with ease.

In this paper, existing segmentation methods are explored and an edge contouring algorithm is implemented in order to facilitate this interaction between the user and the model, removing the need to manually contour every slice of a data set. The algorithm takes a contour and propagates it across cryosections of the Visual Human Project, rebuilding the structure with little input from the user.

1.2 Objectives

This thesis aims to evaluate existing methods of edge contouring and produce an algorithm capable of reducing the amount of user input needed for a total reconstruction. The algorithm will then be used in Anatomy Studio, a project being developed by INESC-ID in partnership with the Unesco Chair of Digital Anatomy. Anatomy Studio consists of a collaborative Mixed Reality tool for virtual dissection that combines tablets with styli and see-through head-mounted displays to assist anatomists by easing manual tracing and exploring cryosection images.

The goal was to develop an algorithm that was able to take a slice from the Visible Human Project data set, along with a rough contour drawn on the platform and produce an adjusted segmentation of the image. The algorithm should then be able to take this contour and propagate it across a set of slices in order to gather information for a full reconstruction of the anatomical structure.

The developed algorithm should maximize the output with minimal input in order to reduce the amount of interactions required from the user. Several pre processing and post processing techniques were evaluated and applied to the algorithm to achieve better performance and efficiency. The user should be able to have a fully reconstructed object within minutes.

Lastly, since the algorithm will be incorporated in a platform that is under development, several adjustments should be made to accommodate the implementation in the platform.

2 The Visible Human Project

The data set used in this paper was gathered from The Visible Human Project, available at the National Library of Medicine of the USA. The processing of the two bodies available was done at the University of Colorado Health Sciences Center (UCHSC) and the radiological images consist of CT and MRI scans. Two cadavers are available, but for consistency purposes only the female body was used, which is also more complete than the male counterpart.

The CT data are axial scans of the body, taken at 1mm intervals with a resolution of 512 by 512 pixels with 12 BITS of grey tone. The MRI data are also axial scans of the head and neck, and longitudinal sections of the rest of the body, but taken at 4mm intervals with 256 by 256 pixel resolution and 12 bits of grey tone.

The cryosections come in different formats, there are a total of 5 189 slices, obtained at 0.33 mm intervals in order to match the sizing of the X-Y plane, making it easier to work with voxels. The images are available at a pixel resolution of 2048 pixels by 1216 pixels and 4096 pixels by 2700 pixels. [7]

Initially, the original data set was used, but after the algorithm was developed it was used to remove the ice in the background along with other unwanted objects. It is also important to note that some slices were damaged by the cryo-procedure along with others where parts of the body disappear, such as the fingers and/or toes, and the nose cartilage. The cadaver also contains several pathologies, including cardiovascular disease and diverticulitis. The damaged slices were not used. The slice numbers from the filenames do not match the numbers on the image. Since the background was removed, when referencing the number of the slice, the filename was used.

3 Existing Edge Detection Algorithms

3.1 Localized Histograms & Region Merging

In this technique, the segmentation is done based on pixel feature values. Clusters are formed with peak-valley analysis of an histogram of pixel intensity. It can also be coupled with region merging to further enhance the segmentation. This has many advantages, such as not being a purely local nor global algorithm, taking all variability into account. This is very important, since some images have huge amounts of features, such as textures, color, shadows, highlights and even fine geometric structures[8]. These systems are usually composed of two algorithms, a localized histogram-segmentation and a region-merging algorithm.

3.2 Gradient Based Algorithms

The Canny Edge Detector is widely used to extract structural information in computer vision, reducing the data load that needs to be processed. This algorithm receives an image and outputs the computed edges of the entire image. Even though the Canny Method is the most accurately defined, it is rather simple and often misses simple edges[9]. It cannot detect branching edges and, when the gradient slopes are very small, the computed direction might not be the normal direction to the contour which leads to missing edges[9]. In order to tackle these problems, several adjustments have been developed to try and improve this method, mainly changing the third step of the algorithm - where the pixel is marked as an edge or background.

Active contours, also known as snakes, are commonly used in shape recognition, segmentation and edge detection. This method requires previous knowledge of the contour, needing an input. It is very useful when the initial seed is close to the desired edge, resulting almost always in convergence[10]. Compared to traditional methods, snakes have many advantages, such as tracking of dynamic objects, the ability to adaptively search for the minimal state and scale sensitivity to the Gaussian smoothing in the energy function. However, the method is very sensitive to local minima, small details on the image are often ignored and the accuracy depends on the convergence policy. As such, there are many variations of this model that try to solve its limitations, but each of them have their own trade-offs.

3.3 Graph Cut Based Algorithms

This method is quite different from the ones described above. It was initially developed by Boykov and Jolly in 2001 using a monochromatic image. The segmentation started with a trimap T and an array of grey values, then histograms were built using the respective trimap regions for background and foreground in order to calculate the opacity values and model. Later iterations of the algorithm start using RGB color space. The histograms are no longer used and Gaussian Mixture Models have been implemented.

When the intensity of the pixel is close to the foreground, the weight between this pixel and the s -node will be larger, meaning that the cut will occur at the edge with smaller weight. When two pixels have the same intensity, the weight is very big, therefore, they are not likely to be separated[11].

3.4 Comparison

The Canny Edge Detection algorithms while effective, are heavy in computation, rely heavily on proper parameterization and do not use global information. This last aspect is also true for other gradient-based algorithms. These algorithms are best suited for images with high contrast, such as CT and MRI scans.

Localized Histograms and Region Merging make use of global information to detect edges, which is an improvement, and there are multi-feature implementations to segment images of color. However, the results obtained are not always closed contours and the process is fully automatic, which is not what is needed for this project.

As for methods using graph cut, the fact that they can detect boundaries even when the color separation between foreground and background is not very clear makes it very useful to segment the cryosections available in the data set. The main problem with this method is the heavy user interaction needed to produce an image, however, this can be solved with a few bridges between the user and the algorithm.

Out of all the methods presented above, Graph Cut stands out, as later iterations of the algorithm start using RGB color space, the histograms are no longer used and Gaussian Mixture Models have been implemented. Also, based on the results from the other algorithms, the use of Graph Cut is the most suited for the objective of this thesis.

4 Methodology

4.1 Implementation

In order to implement the algorithm in python, the OpenCV-Python library was used. This library was developed to solve computer vision problems and supports a wide variety of algorithms, such as the Canny Edge Detection and the Watershed Segmentation Algorithm, along with tools for image processing and manipulation[12]. The GrabCut algorithm works by accepting an image and one of two user inputs: a bounding box with the desired foreground object inside or a mask with the approximated segmentation. Then it iteratively goes through the GMM estimation, the Markov field and the Graph Cut optimization.

The output consists of the modified *mask*, along with the *bgModel* and *fgModel*. The mask is marked with the same flags specified above (0,1,2 or 3) representing the background/foreground. Some adjustments are needed to perform the segmentation: all 0-pixels and 2-pixels are changed to 0 (ie. background pixels) and all 1-pixels and 3-pixels are changed to 1 (ie. foreground pixels). It can then be applied to the initial image to perform the segmentation.

4.2 Input

Even though the algorithm already has user input functionalities, they do not coincide with the input that will be used. Even though the algorithm accepts a mask, it needs to be generated in advance. At first, different forms of input were explored, such as the using polygons and free-hand drawing to label the background and foreground. Since the users will be able to give input through a tablet or in Virtual Reality, which allows for precise drawing, it was decided that a closed contour of the desired edge was the most appropriate for the problem at hand. A small algorithm was developed to solve this, where the output is a mask with three different pixel labels that can then be used with the OpenCV GrabCut algorithm. Represented in Figure 1, the red, yellow and green areas represent the sure background, the probable foreground and the sure foreground respectively. The width of the drawn contour can be adjusted in order to change the size of the yellow area (probable foreground). This allows for less specific inputs to be drawn, as long as the desired edge is inside the yellow zone.

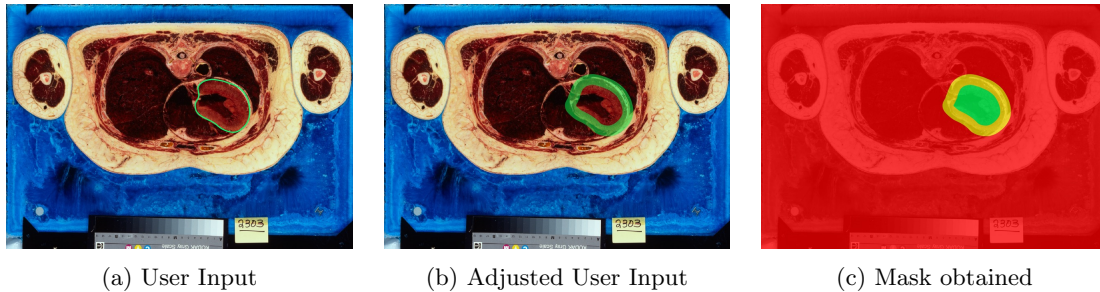


Figure 1: Illustration of the process in Algorithm 2 that starts with a 1-pixel contour in green (a), transforms it into a larger path in green (b) and the final mask obtained (c). In (c), the colors red, yellow and green represent sure background, probable foreground and sure foreground respectively. The object being contoured is the Left Ventricle of the Heart of slice number 2292.

4.3 Editing and Multiple Contours

After obtaining the mask, it may not have converged to the desired edge or contain the entirety of the foreground. Also, some structures might have multiple surfaces, such as hollow objects, where more than one edge is present. As such, some editing tools were added to the algorithm.

Firstly, the ability to directly edit the mask was added. This is done by using two input colors, where the user can draw over the undesired parts in black and unrecognized parts in white. A mask is then created from the input, where white points correspond to sure foreground pixels, and black points to sure background pixels. This mask is used in GrabCut and a new image is produced. The cycle can be repeated until the user is satisfied with the output. The illustration of the algorithm can be seen in Figure 2, where the left ventricle of the heart was not fully labeled as sure foreground by Grabcut. With the new editing tool, the user can now remove unwanted pixels and make sure the entire object is labeled as foreground.

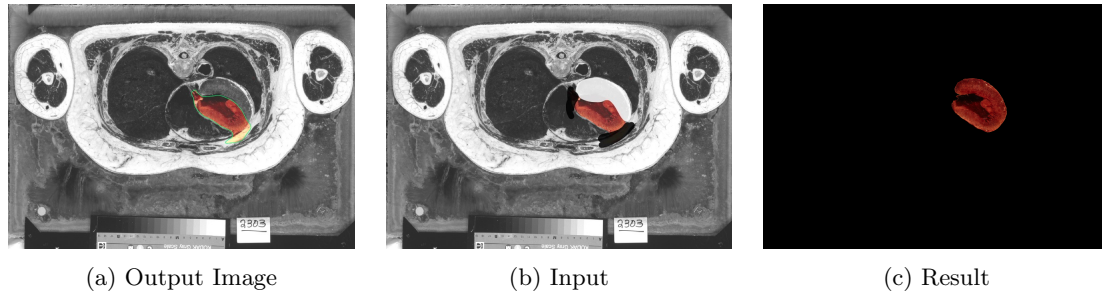


Figure 2: Illustration of the editing process that starts with a sub-optimal result, where only part of the left ventricle of the heart in slice 2292 was masked (a)(b). The editing process receives as input white and black strokes (c), that stand for sure foreground and sure background respectively. The input is then processed and the mask is adjusted (d).

Secondly, the ability to add multiple contours was implemented. In order to accomplish this, part of the code was redone to save more than one contour instead. When the user wants to add a new surface, a new window is opened and the process is the same. The user draws the new contour, the algorithm runs and a new mask is computed. Now, the user specifies whether the new object is to be subtracted or added to the original mask. On the other hand, if the object the user is contouring splits into two, for example, the user can contour the new structure, in addition mode.

4.4 Propagation

The main purpose of the algorithm is to reconstruct the entire object from a single contour. In order to do this, the algorithm is applied to an interval of slices, using the output from a slice and converting it into input for the next. This process consists of creating a dilated mask and a eroded mask. The area between the dilated and eroded masks is the read as probable foreground, the eroded mask is read as sure foreground and anything outside the dilated mask is sure background.

4.5 Improving Segmentation

The next step is to enhance the segmentation of the algorithm. One of the most visible problems is that the contours generated are rough. Also, the time it takes to compute the propagation of the contour along the slice interval is quite long and has room for improvement.

4.5.1 Pre & Post Processing

The first obvious step in preprocessing is to reduce the noise in the original image. The chosen filter was Bilateral which enhanced the smoothness of the final contour obtained from the algorithm while keeping the edges mostly intact. The original image was convolved using the OpenCV Bilateral Filter. The results can be seen below.

On the other hand, the post processing implemented was to reduce the number of points in the contour. The simplest way to achieve this is to reduce the number of points by a factor and using the mean to compute the substitution.

This can be done with a few lines of code, using numpy functionalities. The reshape function transforms the list into an array where the number of columns is the factor of reduction. Then, the mean function flattens the array back into a smaller list using the mean of each line. The results can be seen below.

Another way to achieve this is using the Ramer–Douglas–Peucker algorithm developed for cartographic generalization. Its main purpose is to recursively reduce the number of points of a polyline using the maximum distance between the original curve and the simplified one as criteria. The result is a simplified line consisting only of points that were marked. To use this algorithm in Python, the shapely library was used.

The chosen method was the simpler version, using numpy, as the Ramer–Douglas–Peucker algorithm caused a significant increase in the computation time, not compensating the improved output.

4.5.2 Efficiency

After the algorithm was more robust and was producing good results, there was still one problem at hand. The images in the dataset have a large resolution and the algorithm takes a long time to compute. At this state, the algorithm would compute around 5 slices per minute.

One of the steps that takes a lot of time is using GrabCut. When propagating the contour, the entire image does not need to be read. The image can be cut around the contour and only include the area where the mask has foreground pixels. The only thing to take into account is to make sure that when propagated the window always contains the necessary information. A new window is computed with every cycle using the limits of the contour. This managed to drastically reduce the time it takes to perform a full segmentation of small structures. However, if the tissue/organ being segmented covers a big part of the slice, then the time reduction is less noticeable.

Another possible improvement is the use of parallel programming. There are different techniques that make better use of the CPU cores and python has some tools that allow the use of multi-threading and multiprocessing. The Visual Human consists of massive amounts of data that is loaded in memory and processed. Multiprocessing will considerably speed-up the algorithm and prevent bottlenecks in time and resources. Therefore, it was implemented where one process works on the even slices and the other on the odd ones, since it had good results and works in many scenarios. After implementation, the running time went down by almost a half. For example, when contouring the humerus, before parallel programming, the process took 260 seconds, whereas after it was reduced to 140 seconds.

5 Experimental Results

In order to evaluate the results, segmentation of different structures was attempted.

First of all, the whole data set was segmented in order to remove the ice background. The whole process took some time to compute (around 6 hours), since the full resolution (4096 x 2700 pixels) images needed to be processed. The whole body was segmented with only one, however, some body parts were missing due to the discontinuities on the data set, which were then segmented and incorporated.

Secondly, a complete segmentation of the femur was achieved (figure 3). The femur spans across an interval of 1248 slices, and a total of 9 inputs were provided during the process. The whole process took around 12 minutes to compute. The most problematic areas, which required additional user input, were the extremities, and the damaged slices on the data set. An additional contour was also added, as both the femur's head and greater trochanter were separated in some slices. The knee area is hard to segment due to the lack of clear borders between the bone and the surrounding tissue. Overall, the running time was adequate and the number of inputs was minimal. However, the contours could still be perfected.

Part of the aorta was also segmented, but since it is very homogeneous, it was successfully segmented in one go, which can be seen in Figure 4. The computed time was around 160 seconds for an interval of 630 slices, which is around 4 slices per second. This is possible due to the aorta being small in size, close to 200 by 200 pixels. The performed segmentation converged and managed to track the aorta with only one input, which is to be expected. However, the generated contours did not fully adapt to the rapid changes in the aortic wall in some slices, including pixels from surrounding tissues.

Lastly, a partial reconstruction of the liver was made from a single input. This structure is more heterogeneous and has multiple arteries and veins connected. The algorithm ran for 13 minutes in order to propagate the contour through 400 slices with a window of around 1 500 by 1 000 pixels. The final 3D model can be seen in Figure 5. The reconstruction looks promising, since the algorithm managed to contour most of the liver's ramifications from only one input. However, the model is sub-optimal, some veins were not included, and sections of the liver were left out. The liver itself is a difficult structure to contour as it changes a lot from slice to slice, so one solution would be to provide more input across slices. It is also important to note that the propagation took 13 minutes to compute, around two seconds per slice, which is substantially longer than other structures



Figura 3: Side view of the reconstruction of the femur using the developed algorithm. The discontinuity in the bone is due to the damaged slices present in the data set that were ignored.

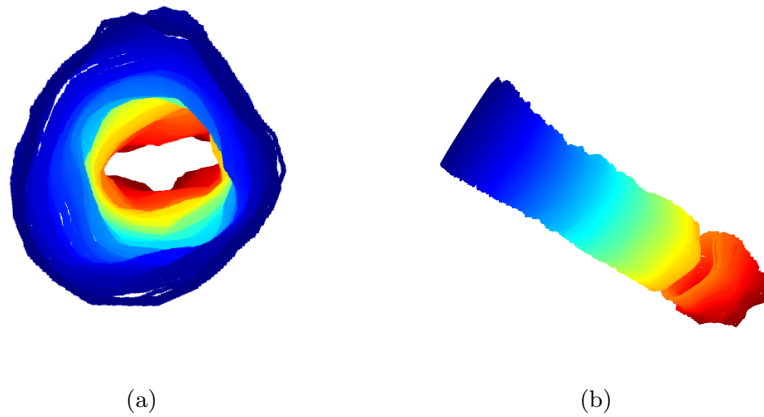


Figura 4: 3D reconstruction of part of the aorta. The discontinuity in the structure is due to the damaged slices present in the data set that were ignored. The top view is represented in (a) and a side view in (b).

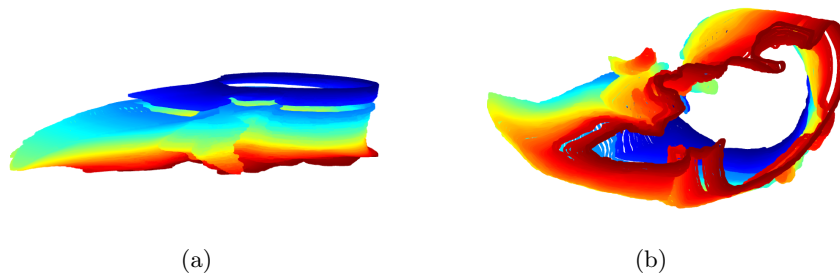


Figura 5: 3D reconstruction of part of the liver. The discontinuity in the structure is due to the damaged slices present in the data set that were ignored. The side view is represented in (a) and a top view in (b).

6 Conclusions

The availability of reliable 3D visualization and segmentation tools is currently a major goal in the medical field, not only for surgical planning but also for teaching anatomy. With the rise of anatomical data-sets of cadavers, more information has been available for researchers to investigate and use.

In this thesis, a study of edge contouring methods was made, such as the canny edge detector, localized histograms and region merging and gradient and graph cut based algorithms. After careful evaluation of the methods, the latter was chosen as it was best suited to segment images of color with the desired user input.

Python code was written to create a semi-automatic segmentation algorithm, which was built on top of the OpenCV GrabCut, capable of fully removing the background from the whole set of slices. Also, a full reconstruction of the femur, the aorta and part of the liver was made in order to evaluate performance.

Current limitations of the algorithm are tied to the lack of contrast between some tissues and the use of high resolution images. GrabCut relies on its color models to segment an image, when the segmented tissue and the background overlap in color-space, the propagation can diverge. This was observed in some structures, such as the lungs, where the segmentation would not converge after propagation.

One solution could be to use of the CT images and MRI for added contrast. This is can be done with Image Fusion, which is a process that combines two or more images. There has been a lot of research done on this method in medical imaging recently, using different techniques. These include convolutional neural networks (CNN)[13], stacked autoencoders [14], pixel-level fusion, convolutional sparse representation[15], and many others. The results from these methods look promising, and could be a solution to the segmentation problem. The only setback when applying this to the algorithm is the fact that the scans were taken at different intervals, the cryosections were done with 0.33mm intervals, the CT scans with 1mm and the MRI scans with 4mm. Also, the axial MRI data is only available for the head and neck, however, these can be reconstructed for the rest of the body using the longitudinal scans.

Another way to improve segmentation quality would be to implement another edge segmentation algorithm. SnakeCut is a technique that integrates the output of both active contours and Grab Cut and uses a probabilistic model to integrate both results[16]. Another technique that takes Grab Cut one step further is the integration of multi-scale decomposition to extract features and integrate into the segmentation process[17] However, these methods make the computation time longer, which is not ideal.

Additionally, the algorithm lacks sensitivity to small structures, such as veins and arteries that can be found on the data set. A second algorithm could be used as a separate tool for detailed segmentations. Also, if the data set used is always the same, then a deep learning machine could be implemented to learn features and improve computed contours.

Lastly, the computation time still has room to improve. Even though the algorithm is computing around 2 slices per second for small structures, when contouring large organs (such as muscles or skin) the velocity can decrease drastically. One solution could be a trade-off between quality and velocity, scaling down the images in order to speed up the process.

Referências

- [1] Ghosh Sanjib Kumar. «Human cadaveric dissection: a historical account from ancient Greece to the modern era». Em: *ACB* 48.3 (2015), pp. 153–169. DOI: 10.5115/acb.2015.48.3.153. eprint: <http://www.e-sciencecentral.org/articles/?scid=1071811>. URL: <http://www.e-sciencecentral.org/articles/?scid=1071811>.
- [2] Lawrence J. Rizzolo. «Human dissection: An approach to interweaving the traditional and humanistic goals of medical education». Em: *The Anatomical Record* 269.6 (2002), pp. 242–248. DOI: <https://doi.org/10.1002/ar.10188>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/ar.10188>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/ar.10188>.
- [3] Ghosh Sanjib Kumar. «Bodies for Anatomy Education in Medical Schools: An Overview of the Sources of Cadavers Worldwide». Em: *Academic medicine : journal of the Association of American Medical Colleges* 93.9 (2018), pp. 1293–130. DOI: 10.1097/ACM.0000000000002227.
- [4] R.McS. «Why there is a shortage of cadavers». Em: *The Economist* (19 de jan. de 2014). URL: <https://www.economist.com/the-economist-explains/2014/01/19/why-there-is-a-shortage-of-cadavers> (acedido em 10/10/2021).
- [5] M.J. Ackerman. «The visible human project: a resource for education.» Em: *Acad Med J Assoc Am Med Coll*, 74.6 (1999), pp. 667–670. DOI: 10.1097/00001888-199906000-00012.
- [6] Jannat Falah et al. «Virtual Reality medical training system for anatomy education». Em: *2014 Science and Information Conference*. 2014, pp. 752–758. DOI: 10.1109/SAI.2014.6918271.
- [7] National Library of Medicine. *The Visible Human Project*. URL: https://www.nlm.nih.gov/research/visible/visible_human.html. (accessed: 05.10.2021).
- [8] Beveridge et al. «Segmenting images using localized histograms and region merging». Em: *International Journal Of Computer Vision* 2 (1989), pp. 311–347. DOI: <https://doi.org/10.1007/BF00158168>.
- [9] Lijun Ding e Ardeshir Goshtasby. «The Canny edge detector revisited». Em: *Pattern Recognition* 34.3 (2001), pp. 721–725. DOI: <https://doi.org/10.1109/TMI.2005.850541>.
- [10] Michael Kass, Andrew Witkin e Demetri Terzopoulos. «Snakes: Active contour models». Em: *International journal of computer vision* 1.4 (1988), pp. 321–331.
- [11] Faliu Yi e Inkyu Moon. «Image segmentation: A survey of graph-cut methods». Em: *2012 International Conference on Systems and Informatics (ICSAI2012)*. 2012, pp. 1936–1941. DOI: 10.1109/ICSAI.2012.6223428.
- [12] *OpenCV: Introduction to OpenCV-Python Tutorials*. 2021. URL: https://docs.opencv.org/master/d0/de3/tutorial_py_intro.html. (accessed: 05.10.2021).
- [13] Yu Liu et al. «Multi-focus image fusion with a deep convolutional neural network». Em: *Information Fusion* 36 (2017), pp. 191–207. ISSN: 1566-2535. DOI: <https://doi.org/10.1016/j.inffus.2016.12.001>. URL: <https://www.sciencedirect.com/science/article/pii/S1566253516302081>.

- [14] Zhuyun Chen e Weihua Li. «Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network». Em: *IEEE Transactions on Instrumentation and Measurement* 66.7 (2017), pp. 1693–1702.
- [15] Yu Liu et al. «Medical image fusion via convolutional sparsity based morphological component analysis». Em: *IEEE Signal Processing Letters* 26.3 (2019), pp. 485–489.
- [16] Surya Prakash, R Abhilash e Sukhendu Das. «Snakecut: An integrated approach based on active contour and grabcut for automatic foreground object segmentation». Em: *Progress In Computer Vision And Image Analysis*. World Scientific, 2010, pp. 535–555.
- [17] Kun He et al. «An improved GrabCut on multiscale features». Em: *Pattern Recognition* 103 (2020), p. 107292.