A novel method for skeletal age estimation based on cranial suture analysis

Andrey Evgeniev Gizdov
National High School of Mathematics and Natural Sciences
Sofia, Bulgaria

Under the direction of:

Assoc. Prof. Stanislav Harizanov
Institute of Mathematics and Informatics
Bulgarian Academy of Science

Assist. Prof. Diana Toneva
Institute of Experimental Morphology, Pathology and Anthropology with Museum
Bulgarian Academy of Science

Assist. Prof. Silviya Nikolova
Bulgarian Academy of Sciences, Institute of Experimental Morphology, Pathology and Anthropology with Museum
Bulgarian Academy of Science

Contents

Abstract

The accurate age estimation is a substantial part of the integral biological profile, but quite complex in cases of unidentified decomposed and skeletonized human remains, especially in adults. Commonly, the skull is well-preserved and due to the assumption that the cranial sutures close in conjunction with age, the patency of contact between adjacent calvarial bones has been used for an age-at-death prediction in the bioarchaeological and forensic expertises.

This study, is aimed to elaborate an algorithm for objective automatic assessment of the suture closure degree in cross-section and to assess its relation to aging. For this purpose I used volumetric images (.TIFF series and .DCOM series) of dry skulls of adult males with known age-at-death generated by industrial μCT\CT system. The obtained spatial resolution (voxel size of 97.5 μm for μCT) was high enough to allow precise detection of the contact between the adjacent calvarial bones in each of the three bone layers: external table, diploë and external table.

With the use of different algorithms and deep learning, the purpose of the project is to achieve a better cross-sectional view of sutures from a skull CT scan, and to create an automatically executed metric system for estimating the age of an individual, based on this cross-sectional view.

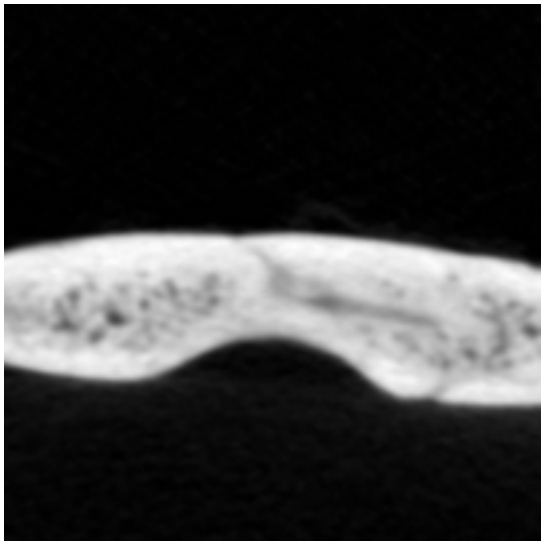
1 Introduction

The neurocranium is composed of flat bones fastened together by calvarial sutures, which represent bands of fibrous connective tissue. The sutures allow no active motions, but act as flexible joints and permit adjustive overlap of the calvarial bones as the head becomes compressed during the childbirth. Throughout the development they function as intramembranous bone growth sites in response to the intrinsic separating forces from the growing brain requirements. When the brain reaches its optimal shape and size during the early adulthood, the process of suture maturation usually begins with a series of morphological changes. The reorganization in the micro-structural finally leads to complete remodeling and obliteration of the sutural area.

Until now, the correlation between suture closure and age-at-death has been examined by conventional quantifying methods based on a superficial subjective assessment of the suture patency on the endo- and exo-cranial surfaces together or individually, using descriptive rank- ordered scoring scales of various grades.

Methods for endo- and exo-cranial surface assessment have existed for hundreds of years. No special equipment is required to extract data from a skull using such methods, therefor the concluded age estimates are not very accurate.

Nowadays, the availability of technology like the CT scanner allows an inside view of a skull's structure, which can further be used to not only explore the endo- and exo-cranial surface, but the cross-sectional depth of the suture as well. This type of in-depth analysis of a suture is a new approach, and only three studies have been made on the topic.
In two of the studies a human is responsible for evaluating the level of bone fusion along the suture, which is incredibly time consuming, subjective and therefore, sometimes inaccurate. There has been only one attempt to automatize this process with the use of a simple algorithm, but the results were not promising, as the algorithm lacked the ability to consider enough factors in the suture image. This image is a good example of a miss-classified suture image.



Using the classification algorithm from [8], the only information we'll receive about the suture is that it is "closed" on this image. While the suture is very vague, it can still be seen that the bones aren't completely fused as the color of the suture isn't completely white. How far they are from complete fusion directly depends on how close to white the color of the suture is. Computationally, this is relatively easy to calculate on a grayscale image but can hardly be done along the whole length of the suture by a human. Those bits of extra information on a certain image, like the color of the suture or the depth of the suture, are neglected by all currently known techniques for age estimation. Considering this extra information can be a key to improving the accuracy of age estimation based on cranial sutures, with real world applications in multiple fields including archaeology, criminology and anthropology. With the use of different algorithms, the purpose of the project is to achieve a better cross-sectional view of sutures from a skull CT scan. The produced cross-sectional images are then used to create a training database for a Semantic Segmentation network which detects and classifies the suture's degree of fusion in a particular image. Based on this information for every provided image, an overall age is predicted for the individual.

2 Problem setting

2.1 Image generation

The CT scans of the skulls for this project are provided by the Bulgarian Academy of Sciences. The scanning was performed using an industrial µCT

system Nikon XT H 225, developed by Nikon Metrology.
All of the skulls belonged to Bulgarian soldiers who died in the First and the Second Balkan Wars and the First World War. Their skeletal remains were preserved in the Military Mausoleum Ossuary, at the National Museum of Military History (Bulgaria). The individuals were fit for service which means they were without severe disorders and malformations. Information about the AAD was taken from the museum's archive, where it was kept.

A typical skull dataset consists of multiple horizontal slice images at different heights along the whole skull. Looking at those images from the top of the skull to the bottom can be seen in `dataset_video.mp4`. At the beginning of the video, a very long crack going through the middle of the image can be noticed. This crack is the sagittal suture, looked at from different horizontal slices.

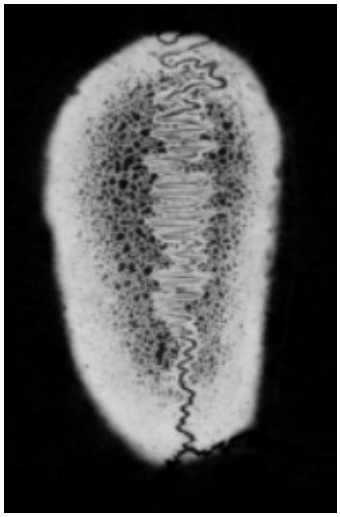


Figure 1

Now take a look at `support_materials\skull_verticall_view.png`. Considering the orientation of the bone on the image and imagining a horizontal slice going through the skull at each horizontal level, the images at the beginning of the video start to make more sense.

Figure 2: Horizontal slice

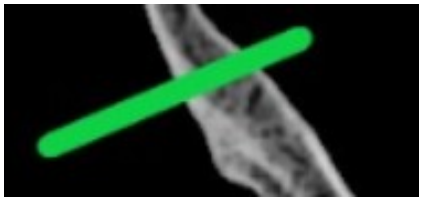


Figure 3: Cross- sectional slice

A cross- sectional slice of the bone at a certain point is defined as an image, perpendicular to the surface at that point. A good example of such slice can be seen on the above pictures. Once we have that in mind, we can conclude that to see an image of the depth (thickness, cross-section) of the suture at a particular point, we would need a slice that is perpendicular to the surface at that point. This is clearly not the case with the top of the skull and the horizontal slices we are provided with. In fact, those horizontal slices are perfectly perpendicular to the surface at very few points on the skull, so we can't directly extract our cross-sectional suture slices from the dataset. For this reason, the first part of the project is to generate images perpendicular to the surface of the skull, in order to get a view of the bone cross-section. This is achieved by firstly generating a 3D model of the skull from the provided image volume. The suture's start and end points are selected by the user on the 3D model. If the shape of the suture is more complex than just a straight line, the software allows for an unlimited number of points to define the length of the suture. The program then automatically generates images perpendicular to the surface of the skull along the length of the suture (between the start and end point). A good example of this ability can be seen in `suture_points.png`. Preferences like the number of images wanted, or the size of the generated images can also be specified by the user. The generated images are then exported to a specified folder, where they're stored for future analysis. The existing studies ([8], [9]) are only able to consider cross-sectional suture images at the top of the head, as the surface of the skull there is relatively flat. This allows for a vertical plane to generate multiple relatively cross-sectional images. Even at this stage, the project already provides a better opportunity to analyze a suture, as it can generate cross-sectional images, not only at the top of the skull, but at any point of the skull.

2.2 Image segmentation

The generated images, provided by the firstly described part of the project, are stored and used to train a Semantic Segmentation network. For the gen-

eration of training images, a sample of 25 male skulls of people from various age have been examined. So far 10 000 cross- sectional images of sutures have been generated, with a possibility to generate 16 000 more if accuracy performance demands it. Currently only the coronal and sagittal sutures are considered as subjects of interest, as the lambdoid suture has a different cross sectional structure and may require a separate classifier.
Each image can have custom dimensions, but for the purpose of training the network, a size of $201x201$ has been chosen for every picture. This size is optimal as it allows for some uncertainty in the position of the suture (sometimes the suture is not exactly in the middle of the image).
The dataset is currently undergoing labeling, with 2000 labeled images so far.
Once the dataset is fully labeled, I'll use a VGG-16 Semantic Segmentation network architecture for training a classifier to detect sutures on new images.

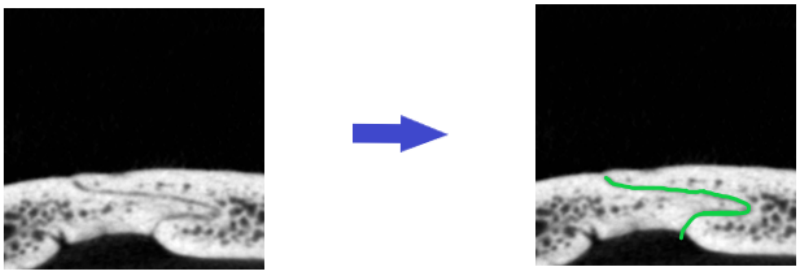


Figure 4: Semantic Segmentation goal

2.2.1 Suture processing

A number of quantifying characteristics for the detected suture region will be measured by an algorithm. The width of the suture on a number of equidistant locations, as well as the color will be recorded.

Based on the results produced from those measurements on each suture image from a particular skull, an overall age estimate will be produced.

2.3 Image processing

2.3.1 Development status

This aspect of the project is independent of the Image extraction and Suture segmentation algorithms. Currently, first priority is to achieve suffi-

cient results using datasets from an industrial μCT system. Integration and improvement of the Image processing capabilities will be implemented as a second priority once the main goal (Image segmentation, Suture processing) is achieved using the μCT datasets.

2.3.2 Description

A characteristic feature of μCT systems, is the very high quality of the produced images. This makes the extraction of data about the degree of suture fusion relatively easy, and not requiring any additional image processing to improve the quality of the under laying data.
Applying the μCT on large sample, however, is expensive, difficult to access and entirely inapplicable *in vivo*. On the other hand, diagnostic imaging of patients generates large datasets of CT scans, which could be used for scientific purpose. The main shortcomings, however, are the low resolution and the artificial blur for denoising purposes, which disable the adequate examination of the contact between the adjacent cranial bones. This is a comparison of suture images produced by a industrial μCT and a medical CT:

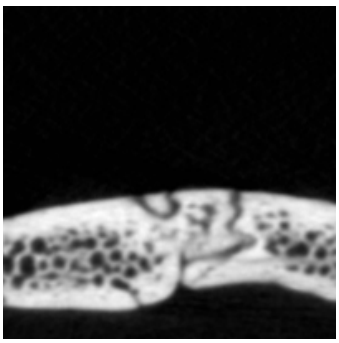


Figure 5: Industrial CT

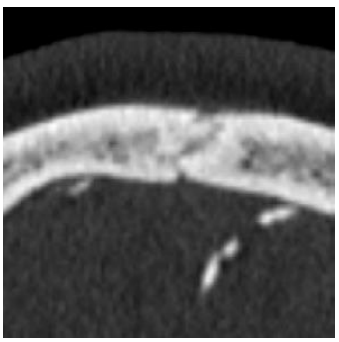


Figure 6: Medical CT

A clear lack of quality in the medical CT image can be noticed. For a good cranial suture analysis, it is important to have a clear view of the edges and a measurable distance between the two ends of the suture. This can be very hard to achieve even by a trained professional, given such a low quality blurred image. Thus, the project also aims to apply theoretical and experimental scientific techniques to artificially improve the quality and adapt the degraded CT data to the described above functionality for automatic suture assessment. This can further broaden the spectrum of situations in which the project is applicable. More specifically, when age estimation is needed

for non-skeletal body remains (*in vivo*).

2.3.3 Problem setting

The described in *Image generation* procedure for generating a 3D model and extracting suture images from it, after some fine tuning, can also be used to generate cross- sectional suture images from a medical CT scan. The image bellow is a medical CT image of a suture:

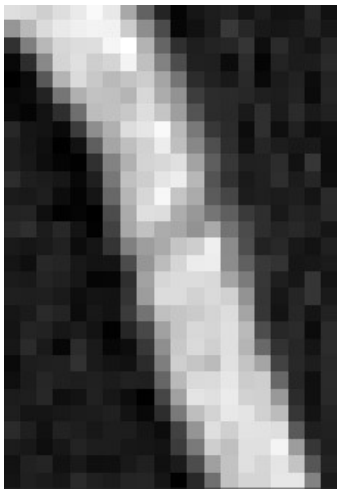


Figure 7: 26 x 18

Doing any sort of image processing on this level of picture quality does not give too satisfying results. For this reason the first step is to increase the resolution of the region we're interested in, which brings the ability to work with any sort of image processing technique on a more local level. A set of different methods have been evaluated for the purpose with details described bellow. The chosen method is subdivision.

After having achieved a higher resolution, a more flexible pattern of processing algorithms can be used for the next steps. One downside of the subdivision method, is the edge smoothing effects it has on the picture. As a clear view of the suture's edges is a crucial factor for the success of the project's task, the next step would be to return or in most cases create, a better view of the edges. This is achieved by a *blind deconvolution* deblurring algorithm. The resulting output image has a much better view of the edges, but the described above noise removal-blurring relationship works both ways,

which creates an excessive amount of noise. A denosing method which uses spheres intersections is used afterwards to remove the present image degradation. The resultant image has a significantly improved quality of edges and a very little amount of noise.
In the processed image bellow, you can clearly see, that there is no noise present around the bone:

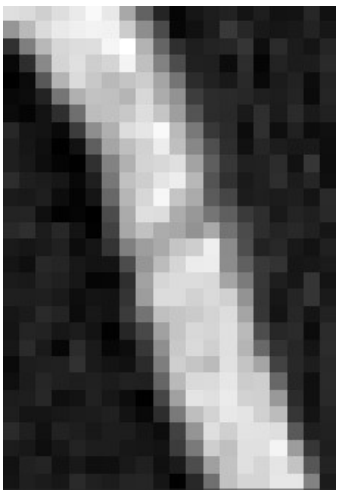


Figure 8: Unprocessed image

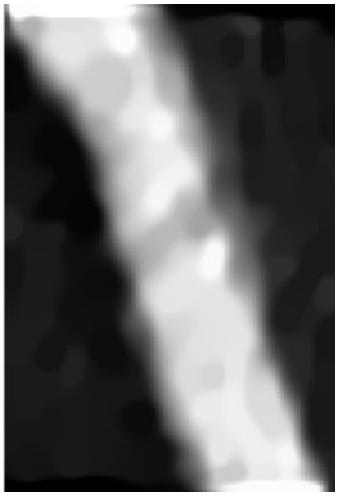


Figure 9: Processed image

3 Project process

3.1 Image generation

3.1.1 Loading an image volume

To do any sort of reading or manipulating on the skull dataset, I first had to load the image volume in the Matlab workspace. The workspace uses the RAM memory of the PC to store the loaded variables, in this case the loaded image volume. The first problem encountered was that the RAM memory required to completely load a typical skull dataset was about 50 GB. My PC has only about 14 GB of usable RAM memory.

What I figured out as a solution to this problem was to load a downscaled version of the volume instead of a full sized one.
Take every image of the image volume to have dimensions:

$$Width \times Height$$

$z \in (0, 1]$, where z is a variable depending on user preferences

Resultant image with dimensions:
$$(Width \times z) \times (Height \times z)$$

I also took the scale factor in consideration with the number of images in the dataset. Instead of loading every single image, I chose to load every $\frac{1}{z}$ th image. This way, the dimensions ratio of the produced downscaled 3D model remain consistent with the full-sized version.
Typical dimensions of a fully sized skull volume are 2190 x 1840 x 2130. The first two dimensions represent the size of each image, and the third is the number of images in the dataset. Displaying such large volumes as graphical object on the screen can be a computationally heavy task, so this downscaling improved the smoothness and significantly reduced any lagging while working with the 3D model.

3.1.2 3D model generation

After having loaded the volume of images, I make a copy of the same volume and run it trough the *binarizeVolume()* function. This function binarizes and fills the holes in every image. Results of the process can be seen bellow.

Figure 10: Unprocessed image

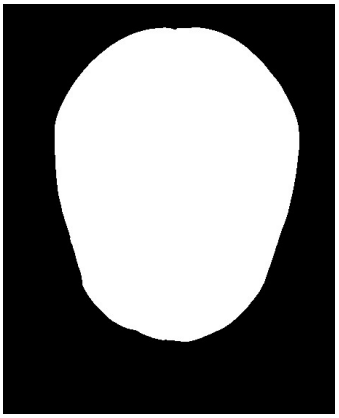


Figure 11: Processed image

The processed volume is then used to generate the 3D model in the *generateVolume()* function.

The advantage of processing the whole volume like this, is that unnecessary details are removed from each image. In this case, the only thing we're interested in seeing on the model, is the very outside surface of the skull. After processing, this very outside surface on an image, is the only preserved detail, as can be seen in the above photo. This suits the needs of the project perfectly and significantly reduces the time taken to generate the 3D model, as the used volume is very simplified after processing. Having generated the 3D model of the skull, the program relies on the user to define the places on the surface trough which a suture passes. Suture defining points can be seen on `suture_points.png`

3.1.3 Path finding

The next task is to generate the path that the suture follows from the given defining points.
Each point on the surface, on the very low level, is a part of a triangular mesh.

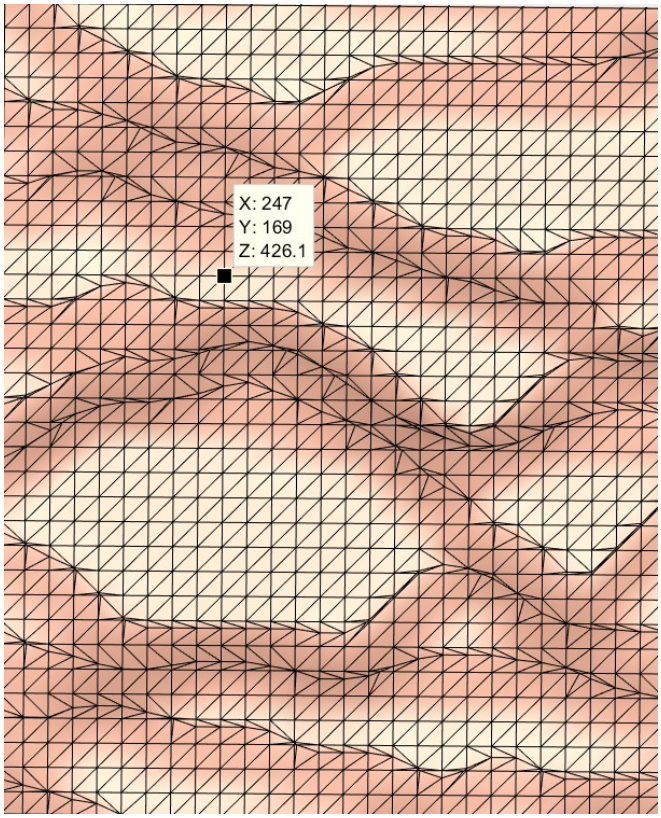


Figure 12: Volume triangular mesh

The question is, which path of connected points on this mesh do we choose to get from the start to the end point. Now this is a question with multiple solutions. Ultimately, this triangular mesh can be taught of as a weighted graph with the weights corresponding to distance between connected vertices. A number of graph search algorithms can be used to find the smallest cost path between two points. Most algorithms like Djikstra or the A* search algorithm may need to go through a big part of the graph before finding an optimal solution. Considering the fact that the number of points on the triangulated mesh of the skull is in the millions range, this process can be computationally and time expensive. For this reason, a computationally cheaper, geometric approach has been used instead.
Let's first look at the simplest case with only three defining suture points. A line is draw between the end point and the start point. Then, the foot of the normal (the blue line) to the start- end line from the middle point is found.

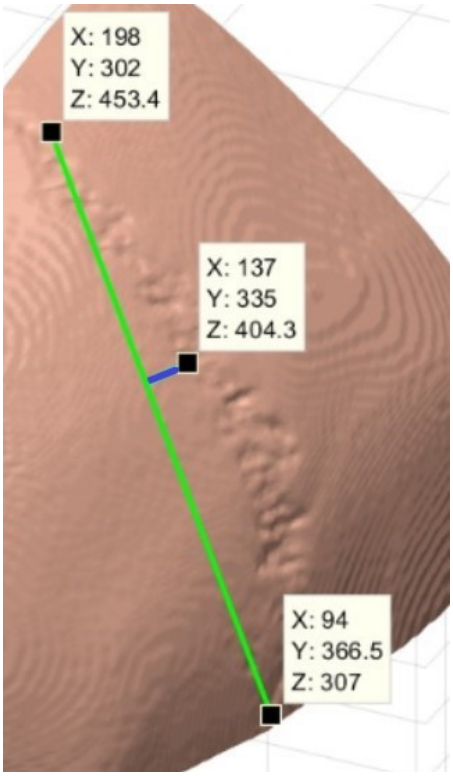


Figure 13

Let's call the vector connecting the foot of the normal and the middle point our *compare vector*.
Starting the path search from the top point, we are now looking at which of the connected to that point vertices to go to next. We then find the normal to the start-end points line for each connected vertex. The angle that the vertex normal vector makes with the "compare vector" is calculated by dividing the dot product of the vectors by their magnitudes. The next chosen point is the point which has the normal making the smallest angle with the "compare vector". The process is repeated until the end point is reached.

For more complicated cases with multiple defining points, every three neighboring points are taught of as separate cases of the described above. As the algorithm reaches the end point of one such triplet of points it starts to follow the normal of the next point triplet. The videos `path_finding.mp4` and `path_finding_action.mp4` show a good graphical representation of the algorithm.
After we find a path for the suture, we then take j equidistant point along the suture path. j is the number of cross-sectional images we want to generate

for the specific suture. The blue surface marks on the videos are such points.

3.1.4 Surface normal calculation

The next step is to determine at what angle to the skull surface we want to generate an image. Let's call the point on the surface for which we're calculating the normal to the surface a "***desired point***". For the purpose, we take the N closest to the "desired point" points. N is a variable and can be specified by the user, as the accuracy of the calculated normal and the time taken to do that calculation, tightly depend on the value of the variable. Experimentally, it has been determined that for most cases a value of:

$$N = 20$$

Is sufficient enough to produce an accurate surface normal estimate. The figure bellow shows a comparison between normals produced with different values of N.

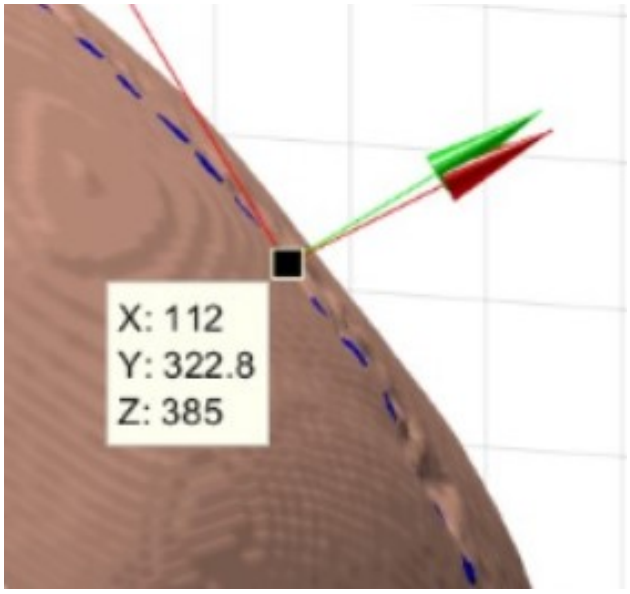


Figure 14: N = 20 (Green) N = 50 (Red)

The larger value of N, while being more reliable for the calculation of the normal vector, doubles the execution time of the function. The angle between the two vectors of different N has experimentally been determined to be:

$\alpha \in [1°, 7°]$, where α is a the angle between two vectors

This is a minimal uncertainty and unable to affect the produced images noticeably. Therefore, the smaller value of N is most commonly used. To find the N points, I've used a KNN search on the vertices tri- mesh graph. Next, we find all of the triangles from the mesh which the N points are part of. The normal vector of each triangle from the mesh is calculated by taking the cross- product of the triangle vectors and multiplying it by the area of the triangle. Every vector is ensured to be pointing outwards to the surface, by comparing the angle that the two possible normals of the triangle make with the vector to the volume mid point. The vector with larger angle to the volume mid point vector is the chosen as the normal vector for the triangle. All the normal vectors of the considered triangles are added together and divided by the number of normal vectors.

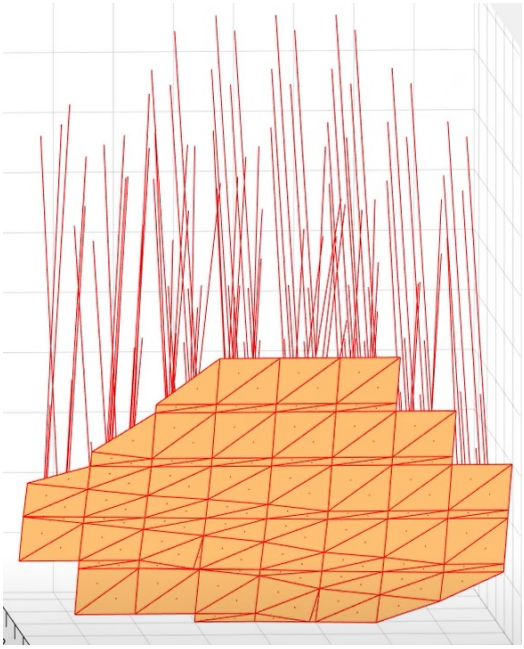


Figure 15: Triangular vector mesh for computing a single surface normal

The result, is an average vector which is a very close to being perpendicular to the skull's surface. A triangular mesh used to calculate the surface normal at one point can be seen in `tri_mesh.mp4` (This video is only available in the support_material folder on the provided GitHub link, as the video was too big to upload in the online form). The normal at each image point is calculated individually using the same method. You can see images of such normal vectors in the support materials folder, or check the video

`surface_normals.mp4`.
A plane parallel to the normal vector will be used to generate each cross-sectional image. The normal of such planes are the red lines coming from the bottom of each point. Planes with such normal are generated. Each plane extracts image data from a chunk of the fully sized volume, as if it extracted the data from the down-scaled volume, image quality would suffer. The function *generateSlices()* automatically loads the volume in chunks with size in respect to the currently available RAM memory.
Most of the times, the vertices of each image plane happen to not be integers. To estimate color values at non-integer vertex points, cubic interpolation is used. The generated images are exported to a specified folder.

3.2 Image processing

3.2.1 Resolution increasement

3.2.1.1 Machine learning

Gaining the largest amount of popularity and showing the most promising results, is not surprisingly the use of a deep learning neural network. Provided there is an abundant amount of training data appropriate for the particular Super-Resolution task, the use of such technique has the potential of achieving impressive results. If however, such data is in limited quantities, the effectiveness of the neural network drops sharply. A prior knowledge about the properties of the images is also required for the effective application. We face a few big problems regarding this idea;

1. During the imaging process of each CT picture of a skull, every X-ray devices produces a certain amount of degradation in the quality of the image. This degradation is often expressed as noise, which is afterwards removed by blurring the image with a certain algorithm. Each CT scan dataset can be originating from a wide range of different models CT scanners, with respectively noise with different characteristics. Besides the difference in the acquisition device, the amount of image degradation is tightly dependent on the scanning parameters. This wide range of factors results in a degraded image with very little to no knowledge about it's origin.

2. For a neural network to be able to construct the resolution of a low quality image, in the training process it needs to be given an example of the same image with a quality which it would aim to achieve in other similar pictures. The issue here is that simply, such good quality examples are just

not available. This becomes a major obstacle with no indication of an ability to overcome.

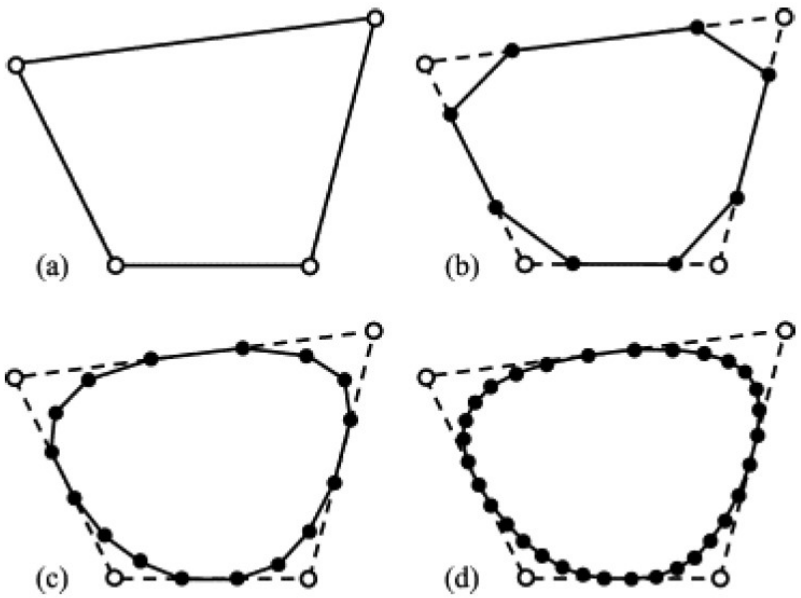
3.2.1.2 B-spline subdivision

Subdivision is a process of recursively refining discrete data using a set of subdivision rules to generate limits (curves, surfaces, height fields, etc.) with desirable properties such as continuity, smoothness, reproduction of shape features, and many more. In B-spline subdivision, the rules Sp are related to p-degree spline extrapolation and are defined in a recursive way, using the Lane-Riesenfeld algorithm: For the point-mass sequence Δ we have that

$$(S_0 x)_{2i} = (S_0 x)_{2i+1} = x_i, \qquad (S_p x)_i = \frac{(S_{p-1} x)_i + (S_{p-1} x)_{i+1}}{2}, \quad i \in \mathbb{Z}.$$

the limit function $S_p^\infty \ \Delta$ is the cardinal B-spline of degree p with support $[0, p+1]$ and thus Sp generates C^p limits. For the purposes of our research, we consider only even p, because those rules are interval-based, not point-based, which better corresponds to pixels. For p = 2, the process is known as "corner-cutting" and its explicit formulas are:

$$(S_2 x)_{2i} = \frac{3x_i + x_{i+1}}{4}, \qquad (S_2 x)_{2i+1} = \frac{x_i + 3x_{i+1}}{4}.$$



3.2.2 Deblurring algorithms

Let's introduce the following model:

h - original high quality image model, which we're attempting to reconstruct
n - added noise produced my imaging device
p - blur applied by imaging device to remove noise
i - initial image

$$i = (h + n) * p \tag{1}$$

This model represents a series of operations resulting to the initial input image. To achieve a closer estimate to the high quality model of the image, the first step would be to remove the blur generated by p.
The best estimation of the type of blur produced by a medical imaging device is a Gaussian blur. Based on result comparison from *Evaluation of Image Deblurring Techniques [1]*, this type of blurring is best dealt with by using a blind deconvolution algorithm. For the algorithm to work, a prior knowledge about the degree of blurring is need, which is expressed as the input argument *Point Spread Function - PSF* in the following function: *deconvblind(image, PSF).*
The PSF is the graphical representation of the blur applied.
Let's imagine that initially, there was a single point somewhere on the *(h + n)* model as shown bellow:

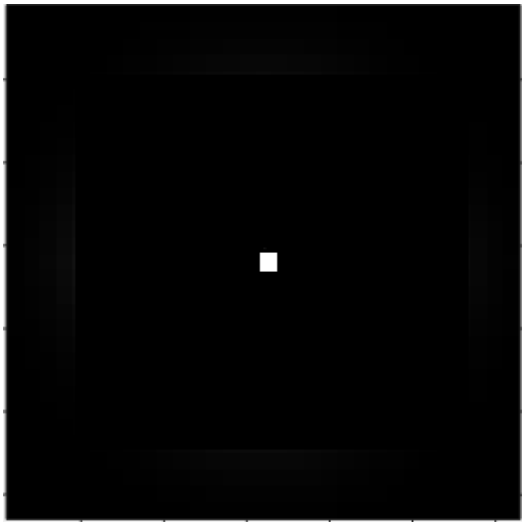


Figure 16: Original point with no blur

After blurring is applied, what before was a point, depending on the wights in the blurring kernel, would look somewhat like this:

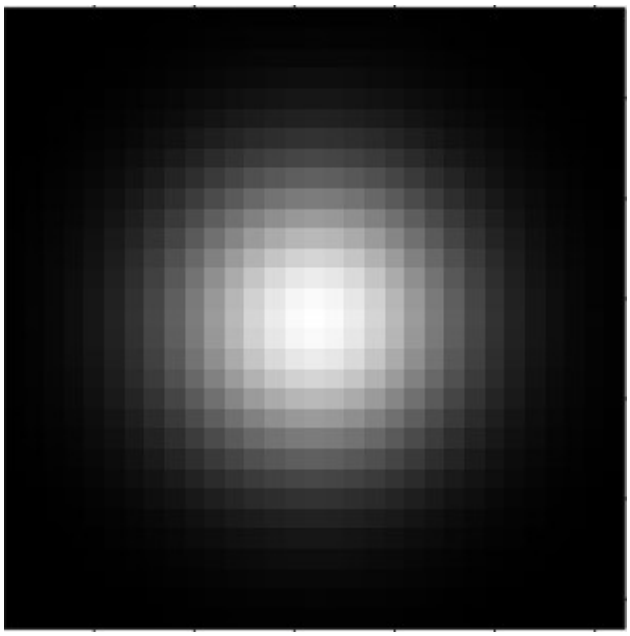


Figure 17: Point with blur applied

The transformation of a single point, after blurring is the best estimate of the PSF used for deblurring. Unfortunately, such single point elements are not marked on the input image, therefore in this case, such estimation of the PSF can not be used.
The BRISQUE and NIQE non-reference metric systems for image quality have been used experimentally to recognize the most sufficient *deblurring PSF standard deviation* used for the deconvolution of the image. The following graph is a comparison between the scores achieved by the different metric systems in respect to different values of this input parameter:

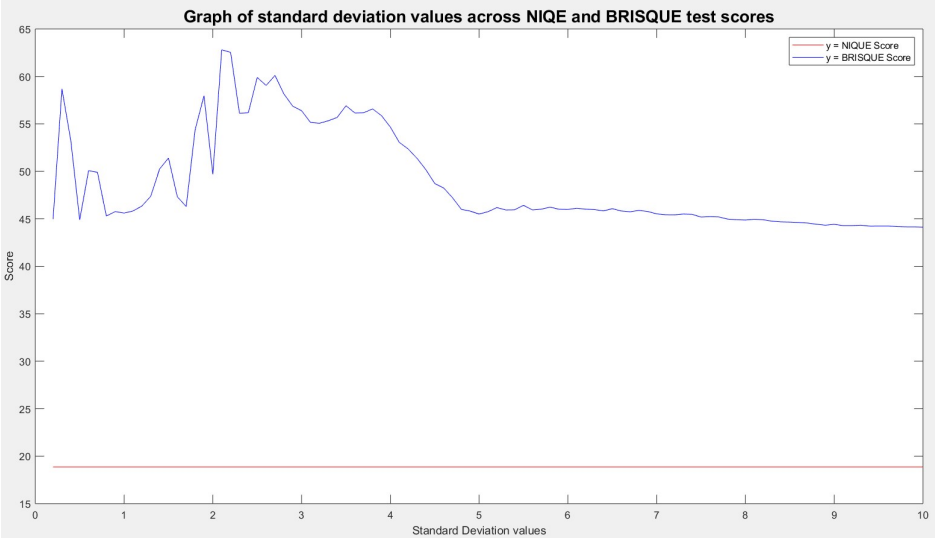


Figure 18: PSF Standart Deviation\Image quality

Based on conclusions drawn from this graphic, the most effective value for the *standard deviation* has been estimated to be around 2. Besides a standard deviation, the generation of the estimated PSF takes a size of the blurring kernel as an argument. For the value of the size, we use the fact that in a normal distribution, within a space of $3 \times Standard\ Deviation$ in both directions from the mean, about 96% of the distribution is contained. Since this distribution is infinitely continuous, this boundary for the size appears reasonable.

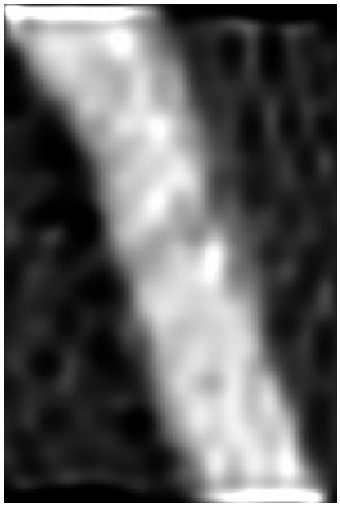


Figure 19: After Deblurring

3.2.3 Denoising algorithm

Deblurring the SR image, achieves a significantly better view of the structure's edges, as can be noted above, doing so creates an excessive amount of noise. To deal with this newly introduced type of degradation, a *Primal-Dual Hybrid Gradient Algorithm for Gaussian Noise Reduction* is used.
Experimentally, using a graph with scores of BRISQUE and NIQE metric systems, the most sufficient value for the denoising *sigma* has been estimated to be *0.002*, as can be noted bellow (The lower a score is, the better):

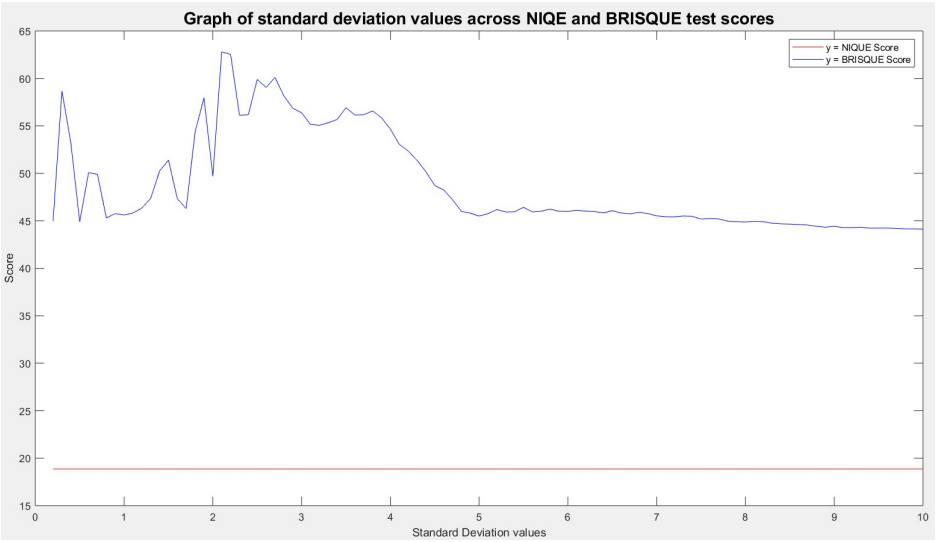


Figure 20: Denoising Sigma\Image quality metric systems

4 Technologies used

1. For the implementation of the cross-sectional image generator, the image processing and the Semantic Segmentation network, I've used *MATLAB R2018a*.

2. For part of the data labeling, I've used *Labelbox* (https://labelbox.com/)

5 Project results

5.1 Image generation

The following is a series of consecutive actions resulting in the generation of an image.

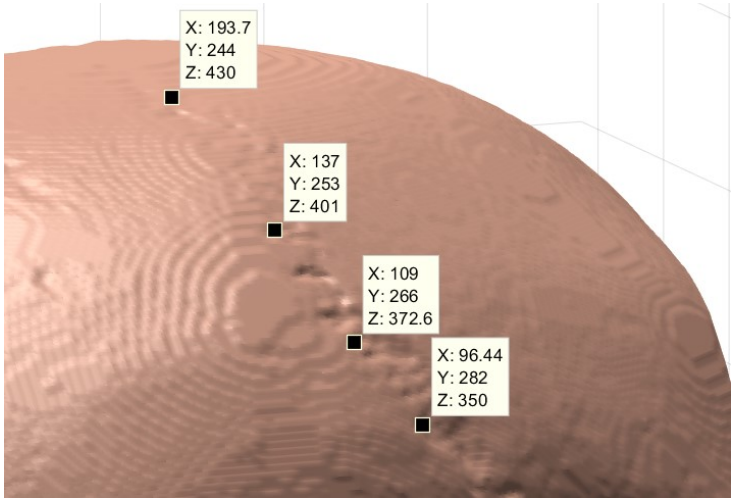


Figure 21: Define suture points

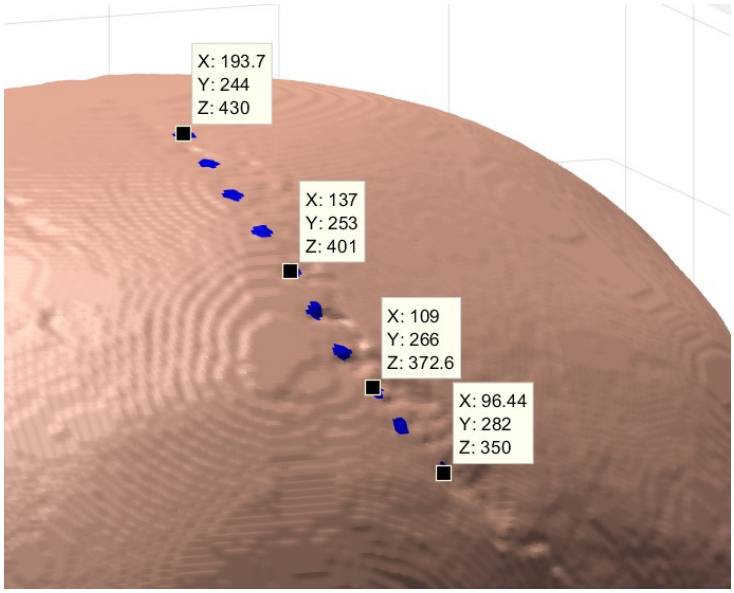


Figure 22: Find image generation points along the suture path

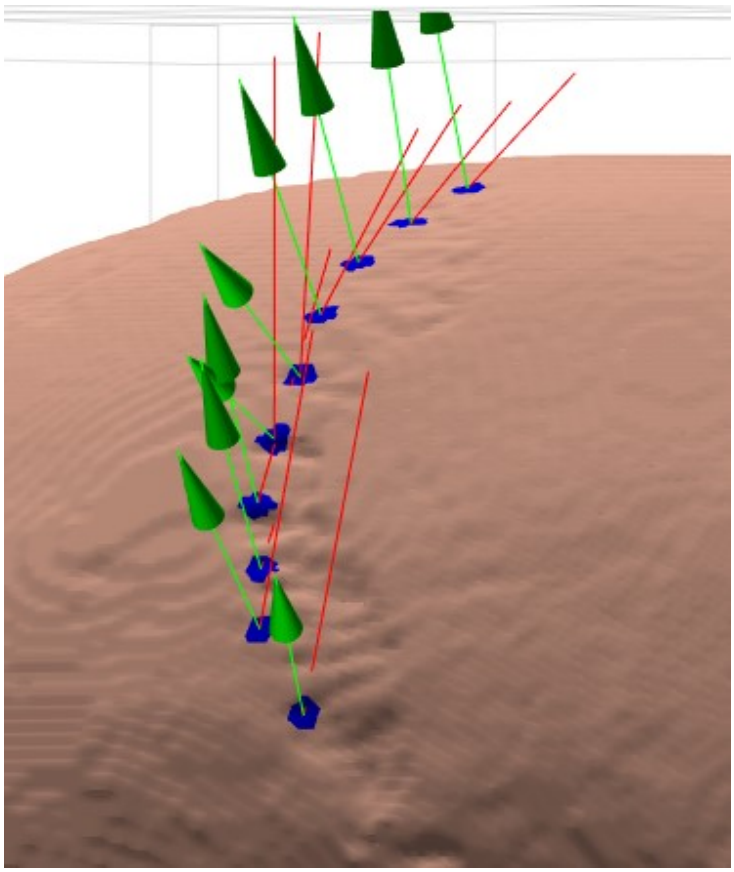


Figure 23: Find normal vectors for image planes

The vectors are then used to generate the specified images. Two folders of generated slice images can be found in the attached zip. Cross- sectional images can be generated for any point of the skull, which is something that has never been done automatically before. At the current stage of the project, even if a human were to analyze the cross sectional images, the results would be more accurate than ones produced by already existing studies, as cross sectional data is available for any point of the suture.
The Semantic Segmentation network is currently being worked on, and combined with a suture analyzing algorithm, has the potential to introduce a significantly more efficient and effective method for age estimation based on cranial suture analysis. The project has so far produced more than 10 000 images of cross-sectional suture images from more than 15 skulls. Part of the generated images can be seen in the `in_depth_sutures` folder.
After appropriate labeling, those images will be used to train an image segmentation neural network, to distinguish and mark a region in which, a part of a suture is present. The properties of the pointed-out suture (suture color, suture width) will further be analyzed. An overall statistic and conclusion about the age of the individual will be made based on the results produced on each image. An example of what the image segmentation neural network has to achieve can be found in the `deep_learning` folder.

5.2 Image processing

The following is a series of consecutively applied image processing techniques applied to a degraded CT medical image input.

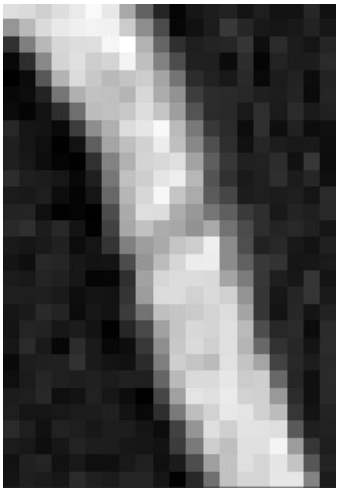


Figure 24: Initial input

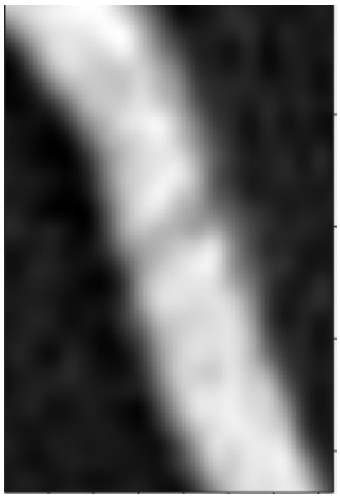


Figure 25: After Super Resolution

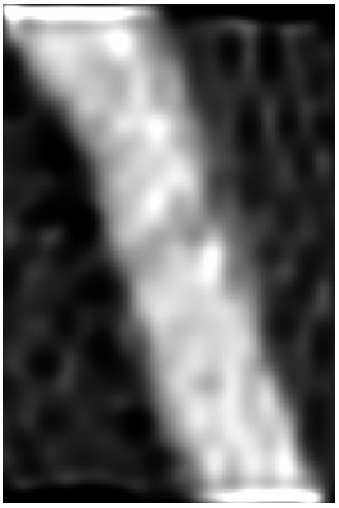


Figure 26: After Deblurring

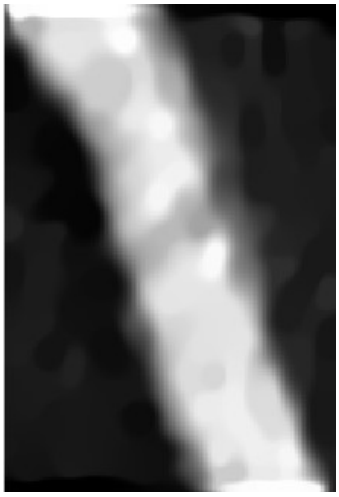


Figure 27: After denosing

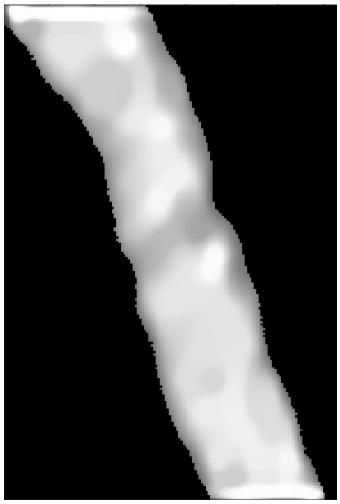


Figure 28: After bone segmentation

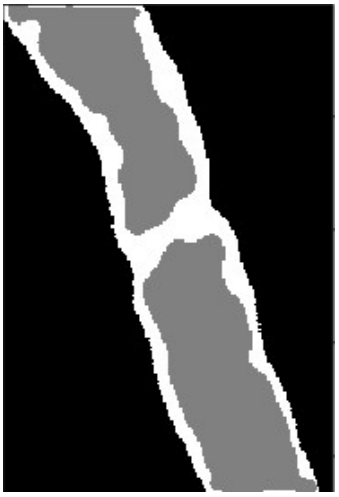


Figure 29: After color clustering

As can be noticed, the suture on the final images is much easier to observe, when it comes to measuring width, or color.

6 Future development

6.1 Automatic suture selection

Currently, the user has to define points trough which a particular suture passes. Based on those points, a path for the suture is generated, for the purpose of image generation.
The suture shape can clearly be seen on the 3D model. It's structure distinguishes from the rest of the surface by certain characteristics.

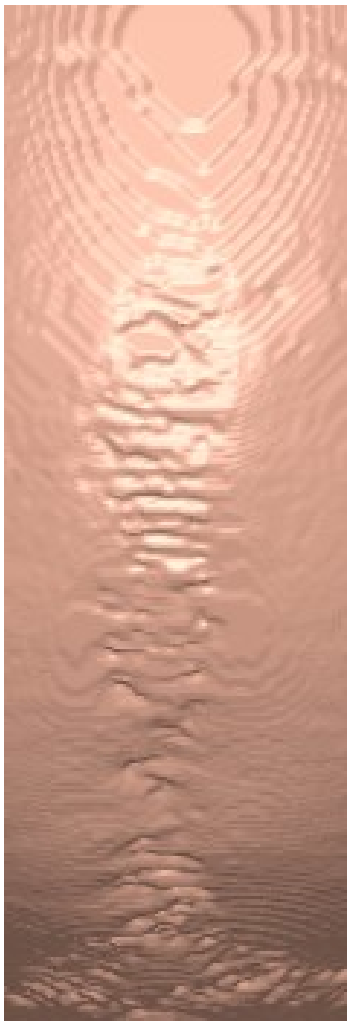


Figure 30: 3D model suture

One of those characteristics is the larger change in angle between connections of adjacent vertices. This results in the artificial light source, reflecting light differently and showing such suture paths in a darker color.
Such gradient and color change patterns can potentially be observed and noticed by an algorithm.
This extra feature will completely automatize the process of age estimation.

6.2 Adapting current algorithms for medical CT

Currently, the image extraction and image segmentation algorithms are tuned to work on high quality image data, as the one provided by a μCT system.
A medical CT provides a volume in a similar manner; a set of horizontal slices is used to represent the skull.
This similarity allows for the use of similar algorithms for medical CT data extraction. Having this in mind, there still are differences between the data format, as the volume size (a medical CT provides less horizontal slices for each volume), the file format ect. Some additional tuning of the existing algorithms might be needed to adapt to the new type of data.

7 Bibliography

References

[1] Evaluation of Image Deblurring Techniques.
https://pdfs.semanticscholar.org/45fc/b086db4631c16990caf21544ec3fb06bfd28.p

[2] Fibrous joint.
https://en.wikipedia.org/wiki/Fibrous_joint

[3] Point Spread Function estimation in X-Ray imaging with partially collapsed Gibbs sampling.
https://arxiv.org/pdf/1709.09105.pdf

[4] PSF of CT imaging devices.
https://aapm.onlinelibrary.wiley.com/doi/pdf/10.1118/1.4835515

[5] Darwinism.
https://en.wikipedia.org/wiki/Darwinism

[6] Definition and explanation of an image blur.
https://en.wikipedia.org/wiki/Defocus_aberration

[7] Image deblurring using neural networks.
http://www.iosrjournals.org/iosr-jece/papers/Vol.%2011%20Issue%205/Version-2/B1105020712.pdf

[8] High-resolution flat-panel volumetric CT images show no correlation between human age and sagittal suture obliteration—Independent of sex.
https://www.sciencedirect.com/science/article/pii/S0379073810001787?via%3Dihub

[9] Estimating age by assessing the ossification degree of cranial sutures with the aid of Flat-Panel-CT.
https://www.sciencedirect.com/science/article/pii/S1344622309001126

[10] Age estimation by multidetector CT images of the sagittal suture.
https://link.springer.com/article/10.1007/s00414-013-0883-y