TECHNICAL TRANSACTIONS | CZASOPISMO TECHNICZNE
FUNDAMENTAL SCIENCES | NAUKI PODSTAWOWE
2-NP/2014

MATEUSZ MATAN*

APPLICATION OF THE 3*D* OBJECTS SURFACE SHAPE ANALYSIS ALGORITHMS IN BIOMEDICAL ENGINEERING

WYKORZYSTANIE ALGORYTMÓW ANALIZY KSZTAŁTÓW POWIERZCHNI OBIEKTÓW 3*d* W INŻYNIERII BIOMEDYCZNEJ

Abstract

The paper describes a concept of measuring possible error estimation and later the decomposition of the predefined model object into convex areas ECD (Exact Convex Decomposition) in order to find a solution to the problem of cavities location with the use of three-dimensional μ–tomography image of the tooth. Such an approach will enable the improvement of automatic cavities detection methods in the future. The paper is also concerned with the problem of a precise object acquisition and estimation of the error value during execution of automatic detection methods.

Keywords: convex hull, error estimation, data acquisition, biomedical engineering, shape analysis, image processing

Streszczenie

Publikacja opisuje koncepcję pomiaru możliwego błędu wraz z dekompozycją predefiniowanego modelu obiektu do obszaru wypukłego z użyciem metod ECD (precyzyjnej dekompozycji wypukłej) w celu znalezienia rozwiązania problemu ubytków w zębach z wykorzystaniem trójwymiarowej tomogrfii obrazu zęba. Wykorzystanie tomografu i metod ECD pozwoli w przyszłości poprawić metody automatycznej detekcji ubytków. W publikacji również rozważany jest problem precyzyjnej akwizycji obiektu z szacowaniem błędu podczas wykonywania metod detekcji automatycznej.

Słowa kluczowe: otoczka wypukła, estymacja błędu, akwizycja danych, inżynieria biomedyczna, analiza kształtu, przetwarzanie obrazów

*Comarch S.A.; mateuszmatan@gmail.com

1. Introduction

The need for algorithms for the three-dimensional objects surface shape analysis arises directly from a desire to automatize real tasks based on extensive domain knowledge concerning issues in the field of medicine, biomedical engineering or physics. However, the complexity and irregularity of real structures (such as $3D$ images of teeth and bones) is a significant obstacle to the effective - in terms of both time and quality - processing and postprocessing of measurement data. One of the possible applications of shape detection algorithms in biomedical engineering is to solve the problem of cavities location with the use of three-dimensional μ–tomograpy image of the tooth [6]. In this paper there is described real (medical) object acquisition process with most common problems in this field (chapter 1). In the next chapter it is given a short survey of concepts in automatic cavities detection (chapter 2). In chapters 3 and 4 is is discussion about possible error estimation methods and expectations that will be shared with described methods. Chapters 5 and 6 are focused on a new convex hull algorithm, its implementation, efficiency and comparison with a commonly used programming library (CGAL). For more details about CGAL please refer to [2].

Precision, of both the measurement and each of the individual processing steps, is a very important aspect in biomedical engineering. Errors made in the early processing stages, accumulated during further processing, may result in falsified test results. Depending on the measured physical quantities, different methods, devices and data processing algorithms are used in the processing. The problem discussed here concerns the method of error estimation, and reducing the differences between the real object, its measurement using predefined model structure used for further processing.

1.1. Examined objects

A set of 20 previously prepared mandibular premolars were the examined objects. Examined objects were selected in terms of similarity of basic physical properties and features such as

- size,
- root size,
- damage degree.

All of the studied objects were previously adapted so that the measurement was in accordance with current dental procedures (removal of dental crowns, clogging of root canals, hardening of sealant).

1.2. Real object acquisition problems

This is the first of the performed measurements - the measurement of the current tooth structure [7]. It allows the acquisition of digital data in the form of $3D$ images of a

previously prepared object (i.e., the tooth). At this stage, the first errors which may affect the quality of further work with the object may occur. *Measuring uncertainties (errors)*, being an integral part of the measurement process, are generally being understood as deviations from the reference value. The main factors affecting the measurement uncertainty include, among others,

- environmental influence,
- imperfect measurement of environmental conditions,
- accuracy class of measuring instruments,
- imperfection of the measurement method.

One of the most visible aspects affecting the quality of measurement may include environmental influence and accuracy class of measuring instruments. It is very important that the measurement at this stage (sampling, quantization) has been performed using a device of the lowest possible quantization noise. While converting a continuous analogue signal (image) to a digital form, each value is rounded to the nearest integer. This is linked to the inevitable and irreversible loss of information. The input signal is approximated by the quantization values and the difference between quantized value and actual signal value appears as the quantization error. The reduction of the quantization error is possible by increasing the number of bits required to store the quantization levels. For example, increasing the number of bits of 1 doubles the number of quantization levels and causes quantization noise reduction by a factor of

$$20\log(2) = 6.02dB(SNR).$$

In the above formula and throughout the paper *log* denotes the decimal logarithm.

1.3. Acquisition and preprocessing of the real object (*ex vivo*)

High-resolution μ–tomograph *X-tek (Nikon) Benchtop CT160Xi* has been used for the measurement. Scanner settings were identical for all of the objects. The original image was recorded using 32-bit floating-point numbers in accordance with *IEEE*–754 and stored as

$$x = S * M * B^E$$

where

- S – sign of a number – 1 bit,
- M – normalized mantissa – 23 bits,
- B – base of the number system (2),

- E – exponent – 8 bits.

The next processing step is to convert each element of the set of measurements to the 8–bit greyscale. This procedure reduces the size of the measurement from 32 bits per point, to 8 bits per point. Conducting a colour image conversion (RGB) to the greyscale for a picture with vibrant colours can have a significant impact on the result obtained. There are many variants of algorithms for image conversion from RGB to greyscale. The quality of the conversion depends on the accuracy of mapping luminance (value) in greyscale, using the luminance of the colour image. The two most commonly used mapping standards are *YUV* (also known as *YIQ*) and *HDTV*, in which the luminance is described as

$$Y_1 = 0.299R + 0.587G + 0.114B \quad \textit{(YUV, YIQ)} \tag{1.1}$$

$$Y_2 = 0.2126R + 0.7152G + 0.0722B \quad \textit{(HDTV)} \tag{1.2}$$

where

- Y_1 – greyscale luminance according to *PAL, SECAM, NTSC*,
- Y_2 – greyscale luminance according to *ATSC* as a standard one for *HDTV*,
- R, G, B – luminances of the red, the green and the blue component.

In the case of colour images with high red, green and blue component intensity, conversion of the image to greyscale may result in a significant readability decrease, as well as the loss of information about the edges.

For photos taken with μ–CT, the colours contained in the image do not carry significant information, as far as the object processing is concerned. These colours are caused by ambient noise or artefacts (noise spots) caused by the device. Hence, the information about colours can be safely ignored here as one of the steps of measurement uncertainties filtering.

During the tests, the luminance model given by (1.1) was used as the most common one. Also the choice of a luminance model for teeth measurements with the use of μ–CT does not have much influence in the context of object shape processing algorithms and the construction of a model.

By using greyscale other information can be interpreted as the intensity of each pixel of the image (values from 0 - the lowest - black, to 255 - the highest - white). This procedure allowed to reduce the size of the measurement by three times.

Fig. 1: Visualization of the three-dimensional structure of a tooth extracted from a list of sequential measurements of cross–sections

$3D$ object structure obtained in this way is a set of two-dimensional images following one another from the top to the crown of the tooth (see Fig. 1). In the measurements set the dimensions of individual objects vary in the following ranges:

- length – from $210px$ to $430px$ (px – pixel),
- width – from $242px$ to $528px$,
- number of cross-sections (height) – from $1187px$ to $1338px$.

The measurement obtained in such a way is the basis for further measurements and hence, its quality is extremely important. Any uncertainties and measurement errors

at this stage remain without any possibility of their reduction or removal at the later processing stages.

2. The problem of automatic cavities detection and availability of solutions

The solution to the problem of automatic cavities detection with the use of μ–CT described in [6, 5] shows that the problem, from a technical point of view, is far from trivial. Solution to this problem requires the application of a series of transformations to simplify the model and to adapt the measurement data to the form in which calculations on the computer will be feasible.

Cavities in the structure of a tooth can be identified in a number of ways. A large amount of thematically varied algorithms (from shape analysis and analysis of the intensity of the pixels in areas with cavities to the construction of mathematical models to simplify the calculation) does not guarantee proper detection of the cavity with an acceptable error value. It is therefore a very important issue to be able to estimate detection error, and to develop complex detection methods by using a new approach and setting it with the current ones in order to compare the quality of algorithms. It is likely that the high-quality automatic detection requires developing a combined approach method based on a segmented object model and its preprocessing and then correcting the result from the previous step by applying, e.g., the pixel intensity analysis.

In this paper, it is written "concave area" instead of "non-convex (plane or $3D$) area". The trivial observation that an area is concave if and only if it is different from its convex hull allows to identify such areas by means of convex hull computer algorithms. There is proposed the solution to the problem of automatic cavities detection which consists of finding all concave areas of the volume greater than an appropriate δ, between the filling of the root canal and the dentin of the tooth. Sample cavities shown in Fig. 2.2, 2.4 and 2.6 can be seen as the dimming between the filling of the root (white colour in the middle of the structure) and the dentin (grey colour) in the form of black or dark grey spots.

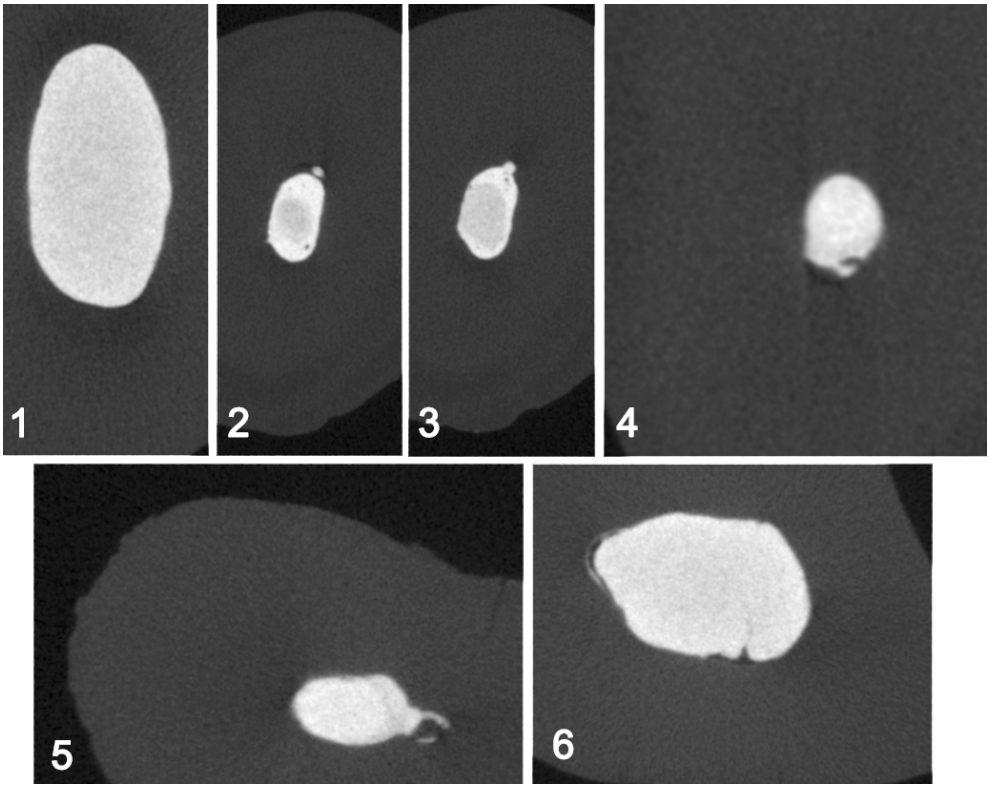

Fig. 2: Images of cross–sections through the tooth structure taken with μ–CT

As one can easily see, inherent limitations (due to the volatility of the real objects) which are not cavities in the filling of the root are among the basic problems of automatic cavities detection. Those include

- natural curvature and concavity (Fig. 2.1, 2.5),
- natural extensions or isolated areas (Fig. 2.3).

Currently studied automatic cavities detection methods with the use of μ–CT [6, 5] focus on the object shape analysis by using numerical techniques and image segmentation tools. The efficiency of these methods still does not allow for their practical use in dentistry, so work on this area focuses on improving the effectiveness of detection methods comparing the effects achieved using dedicated algorithms with results achieved using manual detection.

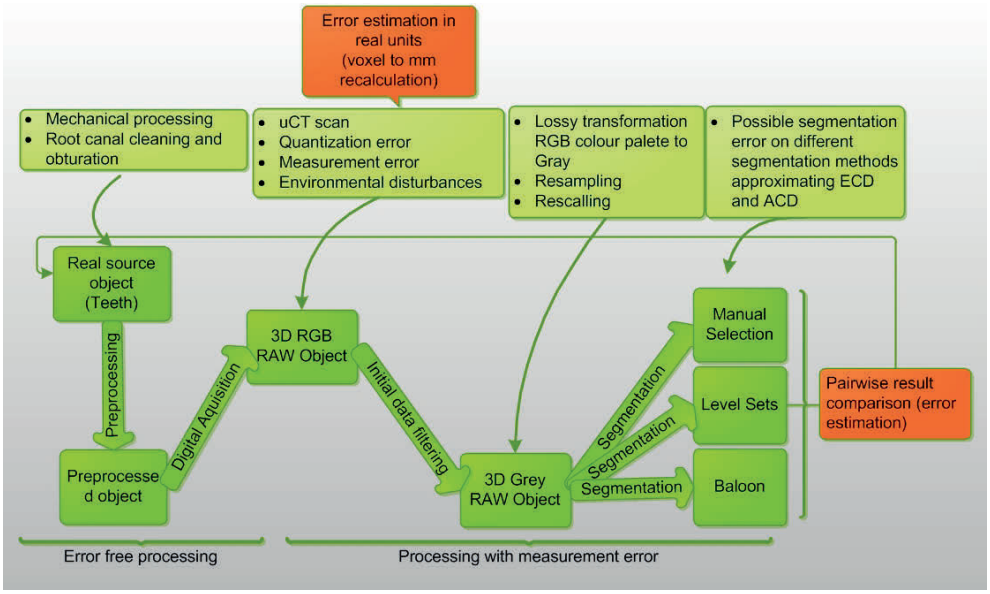

Fig. 3: Stages in the acquisition of medical images using μ–CT in terms of estimating measurement errors

While comparing the effects of algorithms in biomedical engineering it is worth considering the choice of an appropriate comparison method. *MSE* (*Mean Squared Error*), the widely used method for image comparison, gives less than satisfactory results, mainly due to the even distribution of energy (error) on the entire structure of the measured image. Medical image processing algorithms are often aimed at extraction (e.g. an edge of the area), strengthening or segmentation of particular features or areas of an image while leaving the majority of the image in its original form. The use of standard mechanisms for error estimation becomes not only inefficient but can also lead to erroneous conclusions. For this reason, in most medical imaging algorithms, dedicated algorithms are used both to the problem itself and to estimate the accuracy of the measurement. Also for this reason, The latter part of this paper will contain an analysis of the algorithms comparing the results of the segmentation of images made with μ–CT. Error estimation algorithms will be selected in terms of specificity of the current study, which includes all the previously described steps as each of them affects the total error of the test.

3. PSNR - Peak Signal to Noise Ratio

PSNR is expressed as the ratio of the maximum possible signal power to the measured power of the signal disturbed by noise and measurement uncertainties. Since the PSNR is a standard and universal error measure applied to the wide variety of signals, a number of measurement methods (including the methods of measuring the similarity of images created during transformation of a measure signal to a $2D$ or $3D$ structural representation) can be compared with PSNR. More detailed description of PSNR and other error estimation standards is available in [8]. Because of the specification of the measured signals and a wide range of their changeability, PSNR is often in a logarithmic scale. PSNR is most commonly used to compare the effects of the reconstruction of images and sounds by means of lossy compression algorithms such as MP3, JPEG. In this case, the initial measurement is the image or sound before compression and the measured object is the compressed image or sound. In the case of such an application, what is interesting is an information about the general average signal distortion across the studied object.

PSNR can be expressed most simply using MSE. Being given a monochrome noiseless image I of an $m \times n$ size and its approximation K (e.g., reconstruction using lossy compression), MSE is defined as follows.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I(i,j) - K(i,j))^2 \tag{3.1}$$

PSNR can be then expressed as

$$PSNR = 10 \log \left(\frac{M^2}{MSE} \right) \tag{3.2}$$

where $M = \max I(i,j)$.

In practical use, both PSNR and MSE are powerful tools for the study of various processing parameters of different objects, but using the same algorithm, such as JPEG. As it turns out, comparing outputs of different algorithms applied to the same image can sometimes provide subjectively better results for the object whose PSNR had lower value. This is mainly because PSNR returns information about an average error value between source and measured object, in contrast to the human eye, for which features such as preserved edges and pixel intensity are more important than their average values. The situation is very similar in the problem discussed in this paper. As far as cavities detection is concerned, information about the correct edge detection and contour retention is more important than the average accuracy (error) of the total measurement. In the case of a three-dimensional object model, it starts with a fully specified set of points [1, 3]. This collection will be a link between the root filling and the cavity. The error estimation using PSNR and MSE should therefore be

limited to the stage of data acquisition and conversion of the object to greyscale (8 bits per pixel). These stages (Fig. 3) can be treated as an acquisition of the original signal and the preprocessing stage of fixed noise resulting from a specific character of devices and conversion algorithms used.

While modeling the processing stage in terms of data accuracy control one can specify the following stages:

- stage initial - which consists of the measurement accuracy and precision of the preprocessing stage,
- stage processing - which consists of all the algorithms of detection, segmentation and modeling of the digital structure,
- stage feedback loop - in which the results obtained in the processing stage are converted from the pixel value to metric units together with error estimation at each stage.

4. Structural Similarity – SSIM

SSIM (Structural Similarity) just like PSNR and MSE is a method of similarity measurement between two images. Measurement of image quality uses as a reference an uncompressed and undisturbed (by any noise) image. This method is an improvement of traditional signal comparison methods which, as mentioned, do not fully reflect the perception of the human eye. Compared to other error measurement methods, such as MSE or PSNR, which estimate the size of the error, SSIM describes the degradation of the image as a preserved change in the information about the structure. The information about the structure is a concept in which the individual pixels or voxels (in $3D$ space) possess mutual dependencies. These dependencies bring important information, from a global perspective, about the examined object.

The original form of SSIM algorithm described by Z. Wang in [9] and [4] starts with the decomposition of the study area and reference area into $N \times N$ squares. Let us denote such a square by I. The following procedure is applied to (every) I:

$$SSIM(i,j) = \frac{(2\mu_i\mu_j + c_1)(2\sigma_{ij} + c_2)}{(\mu_i^2 + \mu_j^2 + c_1)(\sigma_i^2 + \sigma_j^2 + c_2)} \tag{4.1}$$

where

- μ_i – the average value of pixels in the examined square with the ordinate i,
- μ_j – the average value of pixels in the examined square with the abscissa j,
- σ_i^2 – the variance of pixels in the examined square with the ordinate i,
- σ_j^2 – the variance of pixels in the examined square with the abscissa j,

- σ_{ij} – the covariance of pixels in the examined square defined by

$$\sigma_{ij} = \frac{1}{N}\sum_{k}(I(i,k) - \mu_i)(I(k,j) - \mu_j),$$

- $K_1 = 0.01$, $K_2 = 0.03$ – default values *a priori*,
- L – dynamics of the variables used, expressed as their maximum possible value minus one (for the 8-bit greyscale it is $2^8 - 1$),
- $c_1 = (K_1 L)^2$, $c_2 = (K_2 L)^2$ – two constants to avoid the division by 0.

Computed values can be seen from two different perspectives due to this algorithm. As an example, SSIM algorithm with cluster size $10px \times 10px$ will divide the image of size $1000px \times 1000px$ into $2D$ structures of size $100px \times 100px$. These structures will be called "error boxes". Each pixel in error box will contain SSIM value computed based on different area of size $10px \times 10px$. Now, to compare two $2D$ images there can be used two approaches.

- Compare independently precomputed error boxes (each pair of corresponding pixels on both images). This approach is more precise because it will give an information about "area similarity" which is much more detailed than one summary value from one image.
- Treat the first error box as the first stage of SSIM computing. Repeat SSIM computing until the error box will contain only one value. In this case, two images can be compared using summary computed SSIM values.

Compared to the MSE this measure is more consistent with the perception of the human eye, which can be easily seen in Fig. 4 (From a) to f)). With identical values of MSE (144), SSIM values vary depending on the distortions of the image. The least readable image (bottom right corner) has the lowest SSIM value.

Similarly to SSIM, it can be defined a measure of lack of similarity referred to as *DSSIM* (*Structural DisSimilarity*) [4]. It is derived from SSIM in the following way.

$$DSSIM(i,j) = \frac{1}{1 - SSIM(i,j)} \tag{4.2}$$

DSSIM can be also used as an image distance metric which corresponds better to the human perception than MSE and PSNR.

Error estimation methods are the very beginning part of real objects surface estimation. Algorithms such as SSIM and DSSIM can be effectively applied to problems mentioned in this paper, in contrast to more common MSE and PSNR.

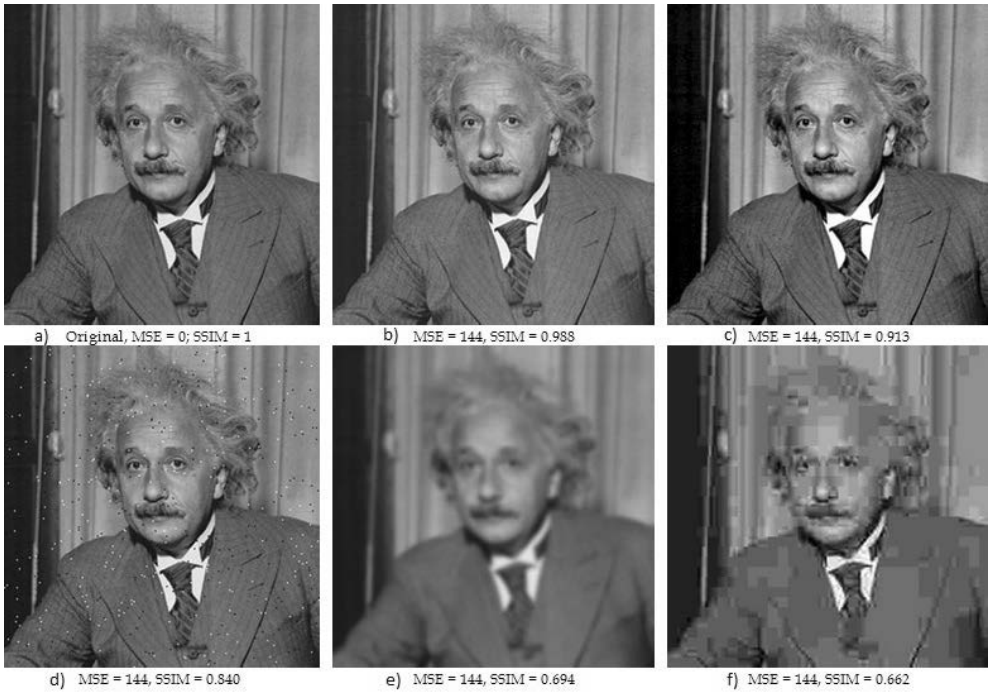

Fig. 4: Juxtaposition of MSE and SSIM with images of different noise levels

5. An algorithm for constructing a segmented object model based on the performed measurement

The most important topic in this paper is focused on ECD (Exact Convex Decomposition). In our problem finding a segmented object model allows to create processing feasible object (models based on RAW data are not feasible). This model can be created using a current measurement $3D$ image (μ–tomography scan). To create a computationally desirable model there should be analyzed some convex hull algorithms, because of a few important features:

- less computationally demanding (in postprocessing stage),
- stable (of known polynomial or logarithmic complexity),
- convex surface is the expected shape for a tooth root,
- convex hull algorithms can be easily adapted to new approaches (surfaces, point selection algorithms, parallelization).

As it was shown can see above, our goal is to design an efficient and problem adapted convex hull algorithm, which will produce the convex hull of a tooth root cavity. Optimal parameters for a convex hull algorithm in the $3D$ space, which is worth considering, are:

- computational complexity (both memory and time - the time of the CPU) and its dependence on points (their order, density and number),
- possibility to parallelize,
- the dependence on the current relation between the number of points in the hull and the total number of points (output sensitive algorithms).

A natural consequence of the parallelized convex hull algorithm design for a given problem is incorporation of elements of the problem into the formula of the algorithm. This considerably increases the time for processing, but simultaneously, it allows the optimization of the time necessary to construct a cavity model based on the measurement data. The starting point was therefore a $3D$ version of the commonly used incremental algorithm that was merged into the algorithm for searching qualifying points. The final algorithm version was parallelized at the end to speed up calculations. The algorithm for searching qualifying points is based on a high pass pixel intensity filter. For each pixel in the image algorithm decides if intensity is above or below a hardcoded intensity value. If intensity is above, the pixel is qualified for further processing. This simple algorithm can be easily parallelized. Using constant thread pool (i.e. 8 threads, depending on the number of processor cores), each thread will simply analyse a surface in the $3D$ image structure.

ParallelIncrementalConvexHull(*R*, *alpha*, *thd*)

ParallelIncrementalConvexHull(*R*, *alpha*, *thd*)
Input:
R – $3D$ picture of a tooth (vertical set of $2D$ pictures)
alpha – value from $[0, 255]$ of pixel brightness
thd – number of threads used to run algorithm, optimally as many as processor cores
Output:
The convex hull $C = CH(R)$ of a cavity in R.

```
1  Run the qualification algorithm with thd on R with alpha.
   This will produce a set P of qualified points (decreased
   global number of points from 50 to 5 mln) ;
2  Find 4 points that form the initial tetrahedron,
   C = CH({p_1, p_2, p_3, p_4});
3  Compute a random permutation {p_5, p_6, ..., p_n} of the remaining
   points;
4  foreach thread in thd do
5  |  Split P into thd sets such that every thread contains
   |  1/thd part of P;
6  |  Each thread will compute a different hull, ALL hulls
   |  will be glued together at the end.;
7  |  Initialize the conflict graph G with all visible pairs
   |  (p_t, f), where f is a facet of C and p_t, t > 4, is a
   |  non-processed point;
8  |  foreach r = 1 to n - insert p_r into C_p - where p_r is a
   |  point from the set P_thd (created at step 6) and C_p is a
   |  part of the convex hull - different for each thread -
   |  which will be glued at the end do
9  |  |  if F_conflict(p_r) is not empty - p_r is outside, any facet
   |  |  is visible then
10 |  |  |  delete all facets in F_conflict(p_r) from C_p - only
   |  |  |  from hull C_p, not from G;
11 |  |  |  walk around the visible region boundary, create
   |  |  |  the list L of horizontal edges;
12 |  |  |  foreach all e ∈ L do
13 |  |  |  |  connect e to p_r by a new triangular facet f;
14 |  |  |  |  if f is coplanar with its neighbor facet f'
   |  |  |  |  along e then
15 |  |  |  |  |  merge f and f', take conflict list from f';
16 |  |  |  |  else
17 |  |  |  |  |  - not coplanar ⟹ determine conflicts for
   |  |  |  |  |  the new facet f;
18 |  |  |  |  |  create a node for f in G;
19 |  |  |  |  |  let f_1 and f_2 be the facets incident to e
   |  |  |  |  |  in the old CH(C_{p_{r-1}});
20 |  |  |  |  |  P(e) = P_conflict(f_1) ∪ P_conflict(f_2) ;
21 |  |  |  |  |  foreach point p ∈ P(e) do
22 |  |  |  |  |  |  if f is visible from p then
23 |  |  |  |  |  |  |  add (p, f) to G - new edges;
24 |  |  |  |  |  |  end
25 |  |  |  |  |  end
26 |  |  |  |  end
27 |  |  |  end
28 |  |  |  Delete the node corresponding to p_r and nodes
   |  |  |  corresponding to facets in F_conflict(p_r) from G,
   |  |  |  together with their incident edges;
29 |  |  end
30 |  end
31 |  Merge C_p into C;
32 end
```

Algorithm 1: Parallel Incremental Convex Hull Algorithm

It will be shortly described the above version of Parallelized Random Incremental Convex Hull algorithm. Let $R = \{p_1, \ldots, p_n\}$ be a set of n points that form a $3D$ image. Each thread will execute the same code, so to simplify this description, parallelizing part was skipped. For $i = 5, \ldots, n$ define $R_i = \{p_1, \ldots, p_i\}$. Suppose that the convex hull $CH(R_{i-1})$ is already computed. The goal is to compute $CH(R_i)$. If p_i is inside $CH(R_{i-1})$, then result is simply $CH(R_i) = CH(R_{i-1})$. Otherwise, p_i is on the boundary of $CH(R_i)$. All edges of $CH(R_{i-1})$ between two straight line segments joining p_i to vertices of $CH(R_{i-1})$ should be deleted, and these two straight line segments should be added into $CH(R_i)$. Those few steps are executed sequentially for each point, and in the parallelized version the original problem is divided into *thd* number of subproblems executed by each thread independently.

This method can be parallelied but as an sequential algorithm it is nor the fastest one neither the less resources expensive. This is a good point to start affecting development of image processing in medical areas with deterministic and computationally feasible algorithms designed specially for these purposes.

The time complexity of the algorithm is $O(n \log n)$, and the expected memory usage is $O(n^2)$, where n stands for the number of points processed (i.e., the size of the source problem). What is more important, it is expected a very good time efficiency with results comparable to other popular $3D$ convex hull algorithms like D&C and Gift Wrapping.

6. The algorithm tests and comparison with CGAL

CGAL (The Computational Geometry Algorithms Library) is a programming library for C++ which provides a wide range of algorithms, including triangulation, Voronoi diagrams, mesh generation, the convex hull shape analysis and many others. During the measurement it was used to verify the accuracy of the proposed algorithm and in timing tests.

All the tests were performed on Windows 7. The computer used to compare the algorithms was equipped with $6GB$ of memory and 2 core (4 threading) Intel i7 – $2.2 - 2.8$ GHz. The source code was written in C++ and compiled with the GCC compiler (within the MinGW environment).

Test 1. The calculation of the convex hull of a million random points distributed in the $3D$ space.
CGAL static approach – 1,67s
CGAL dynamic approach – 9,76s
CGAL incremental algorithm – 12,01s
The suggested algorithm – 4,33s

Test 2. The calculation of the convex hull of three measurements of mandibular premolar (20 mln. points ±100000).
CGAL static approach – 38,43s
CGAL dynamic approach – 44,25s
CGAL incremental algorithm – 79,22s
The suggested algorithm – 36,94s

Test 3. The calculation of the convex hull of three measurements of cavities in mandibular premolars (5 mln. points ±50000), taking into account the time for qualifying points calculation.
CGAL static approach – 38,65s
CGAL dynamic approach – 81,34s
CGAL incremental algorithm – 93,64s
The suggested algorithm – 33,44s

The performance of the algorithm searching qualifying points takes 28, 32s. The idea of this algorithm is to find all the points of brightness above a specified threshold (inside the cavity). This algorithm is not the subject of this problem and will not be discussed.

From the above results of the measurements, it can be concluded that the proposed algorithm in two of the three generated cases is not optimal in terms of execution time (based on our experience, the difference of approx. 5% of the time size is negligible, because the testing environment influence can be bigger despite few test trials). For a real-life case (described in Test 3) it turns out to be better then the static approach (over 10% of static time). Due to its flexibility (multi-threading, integration of the algorithm searching qualifying points) and the proposed use (analysis of medical structures) it can be a good alternative to the commonly used methods.

7. Concluding remarks

The present paper highlights the need for graphics processing for the requirements of dentistry in biomedical engineering applications and the possibilities which are available this area. The first important aspect is the need to verify the quality of the measurement, or the reconstruction, by selecting the appropriate method of error verification and measurement uncertainty. Error indicators such as MSE and PSNR only provide information about the energy distribution of all components and properties of the compared images. In the case of biomedical issues, a more efficient method would be the comparison of images on the basis of contours (edges) and the brightness of each pixel (voxel), mainly due to the analogy with the human eye perception. Applying algorithms such as SSIM or DSSIM in biomedical engineering is not common solution and still has not been tested nor described in other science literature. On the basis of the performed tests juxtaposing MSE error indicators and SSIM it can be

concluded that SSIM algorithm considered by VQEG (video quality experts group) as one of the standard algorithms for image quality assessment, despite the much greater computational complexity, is a strong competitor for the PSNR and MSE in image processing for medical purposes. The second considered aspect is algorithms for the construction of models of areas of interesting features from the point of view of a given field (algorithm qualifying points on the basis of, e.g., their brightness). Here convex hull algorithms were considered. They are able to generate convex area, which meets the initial limitations, on the basis of $3D$ space analysis. The proposed algorithm was a parallelized version of the incremental algorithm. All the tests were compared (in terms of calculation speed and accuracy) to the algorithms implemented within CGAL library which is commonly used for scientific calculations. Modifications of incremental algorithm were limited to calculations scattering and preliminary points classification. The resulting code has proven to be competitive in terms of time for algorithms within CGAL library (QuickHull3D, DC3D, GiftWrapping3D, incremental static and dynamic). This allows to claim that in the future the application of a similar approach to the optimization of calculations for medical structures analysis will allow to create algorithms enabling to generate $3D$ models in real time with an acceptable resemblance to the original structure of the design model. As a result, the operations performed on the segmented object model will be less computationally complex, which will allow broader analysis of structures with computer-assisted methods.

References

[1] Borgefors G., Sanniti di Baja G., *Analyzing Nonconvex 2D and 3D Patterns*, Computer Vision And Image Understanding, Vol. 63, No. 1, 1996, 145—157.

[2] CGAL Open Source Project – *http://www.cgal.org/*.

[3] Lien J., *Approximate convex decomposition and its applications* (PhD dissertation), Texas A&M University 2006.

[4] Loza et al. A., *Structural Similarity-Based Object Tracking in Video Sequences*, Proc. of the 9th International Conf. on Information Fusion, Florence 2006.

[5] Metzger Z., Zary R., Cohen R. , Teperovich E., Paque F., *The Quality of Root Canal Preparation and Root Canal Obturation in Canals Treated with Rotary versus Self-adjusting Files: A Three-dimensional Micro-computed Tomographic Study*, JOE, Vol. 36, No. 9, 2010, 1569—1573.

[6] Petryniak R., Tabor Z., Kierklo A., Jaworska M., *Detection of voids of dental root canal obturation using micro–CT*, Computer Vision and Graphics, Lecture Notes in Computer Science, Vol. 7594, Springer, 2012, 549—556.

[7] Rhodes J. S., Pitt Ford T. R., Lunch J. A., Liepins P. J., Curtis R. V., *Micro-computed tomography: a new tool for experimental endodontology*, International Endodontic Journal, Vol. 32, 1999, 165—170.

[8] Salomon D., *Data Compression: The Complete Reference (4 ed.)*, Springer 2007.

[9] Wang Z., Bovik A. C., Sheikh H. R., Simoncelli E. P., *Image quality assessment: From error visibility to structural similarity*, IEEE Transactions on Image Processing, Vol. 13, No. 4, 2004, 600—612.